Living with Robots: Interactive Environmental Knowledge Acquisition

Guglielmo Gemignani[a], Roberto Capobianco, Emanuele Bastianelli, Domenico Daniele Bloisi, Luca Iocchi, Daniele Nardi[b]

[a]*Corresponding author*
E-mail: gemignani@dis.uniroma1.it
[b]*Department of Computer, Control, and Management Engineering*
Sapienza University of Rome, Italy
E-mail: <lastname>@dis.uniroma1.it

Abstract

Robots, in order to properly interact with people and effectively perform the requested tasks, should have a deep and specific knowledge of the environment they live in. Current capabilities of robotic platforms in understanding the surrounding environment and the assigned tasks are limited, despite the recent progress in robotic perception. Moreover, novel improvements in human-robot interaction support the view that robots should be regarded as intelligent agents that can request the help of the user to improve their knowledge and performance.

In this paper, we present a novel approach to semantic mapping. Instead of requiring our robots to autonomously learn every possible aspect of the environment, we propose a shift in perspective, allowing non-expert users to shape robot knowledge through human-robot interaction. Thus, we present a fully operational prototype system that is able to incrementally and on-line build a rich and specific representation of the environment. Such a novel representation combines the metric information needed for navigation tasks with the symbolic information that conveys meaning to the elements of the environment and the objects therein. Thanks to such a representation, we are able to exploit multiple AI techniques to solve spatial referring expressions and support task execution. The proposed approach has been experimentally validated on different kinds of environments, by several users, and on multiple robotic platforms.

Keywords:
Semantic Mapping, Knowledge Representation, Human Robot Interaction

1. Introduction

Robots are expected to be the next technological frontier, impacting our society in many sectors, including security, industry, agriculture, transportation, and services. In particular, robots are expected to become consumer products, that massively enter our homes to be part of our everyday life. This naturally brings back the view of the robot as an intelligent agent, capable of smoothly interacting with humans and operating in real life environments, whose features are undoubtedly challenging. In addition to the recent developments in robotics, other technological advancements make it more feasible to address the design of robots as intelligent agents. Specifically, new sensors allow for more effective approaches to perception, new devices support multi modal interaction with the user and novel speech recognition technologies enable the realization of actual conversational agents.

This notwithstanding, there is still a gap in terms of user expectations and robot functionality. A key limiting factor is the lack of knowledge and awareness of the robot about the operational environment and the task to be accomplished. In other words, the world models that robots embody do not support even simple forms of common sense knowledge and reasoning. Additionally, in order to support the implementation of simple commands such as to *"go near the fridge"* or *"check whether the TV set is on"*, state of the art systems typically need a significant engineering effort in coding the knowledge about the specific operational environment. Consequently, it is impractical to scale-up, enabling the robot to deal with different environments and different requests from the user.

To this end, in this paper we present an approach that allows a robot to incrementally learn the knowledge about the environment, by relying on a rich multimodal interaction with the user. We specifically address the problem of acquiring the knowledge about the environment (i.e., the semantic map) and maintaining it. Compared with previous work, our approach can be seen as an *incremental online semantic mapping*, in which a rich and detailed representation of the operative scenario is built with the help of the user. The resulting integrated representation enables the robot to perform topological navigation, understanding target locations and the position of objects in the environment.

The approach described in this paper builds on previous work for off-line construction of semantic maps [1], based on two main components: *(i)* a component for Simultaneous Localization And Mapping (SLAM), that provides a metric map of the environment; *(ii)* a multi-modal interface that allows the user to point at the elements of the environment and to assign their semantic role. The novel contri-

butions of this work are the following:

- The representation of the environment, which is automatically extracted from the metric map and is labelled through user interaction. On this representation different form of symbolic AI techniques can be applied.

- The process of building and updating the representation, which is done incrementally during the deployment of a robot through a continuous interactive process;

By exploiting the representation of the environment built through this process, we are able to support several forms of reasoning, task execution and complex interactions.

The above listed features have been embedded into a prototype system that has been extensively used over months to validate the proposed approach, in substantially different environments, and with multiple users. This robotic system is a fully functioning prototype able to enter an unknown environment and incrementally create a semantic map that supports the execution of complex tasks in a partially dynamic environment. A semantic mapping approach similar to the one presented here could be deployed on robots that are currently entering the market, such as telepresence robots [2]. In this specific scenario, users could be allowed to give a tour of their homes or offices to the robot, teaching it about relevant objects or rooms in the environment, possibly in different moments. The knowledge acquired through this process could then be used to simplify the teleoperation of a robot by allowing both the remote and the nearby user to command the robot using natural language. For example, users could just instruct the robot to reach a previously learnt room or object, without the need to tele-operate it.

In this paper we focus on the representation, acquisition and use of the robot knowledge. We refer the reader to previous publications for a more detailed description of the robotic system and for an overall discussion of the natural language interaction [3, 4, 5]. The remainder of the paper is organized as follows. Section 2 describes the background and provides a suitable context for our work with respect to the state of the art. Section 3 outlines in detail the aim and the contributions of our work. The representation of the robot's knowledge is illustrated in Section 4. Section 5 describes the knowledge acquisition and maintenance, while the uses of the acquired knowledge in the behaviors of the robot and in the dialogs with the user are discussed in Section 6. We illustrate the experiments that we have carried out to validate the proposed approach in Section 7. Conclusions are drawn in Section 8.

2. Related Work

Our work mostly relates to the literature on semantic map acquisition and especially on those approaches that rely on human-robot interaction in natural language. After the early-day works in semantic mapping [6], all the recent approaches can be grouped into two main categories, by distinguishing fully automatic methods from approaches involving a human user to help the robot in the semantic mapping process. The first category of *fully automatic* methods, in which human interaction is not considered at all, can be further divided into three different sets. A first set of techniques aim at acquiring features of the environment from laser based metric maps to support labeling and extract high-level information. In [7], for example, environmental knowledge is represented by augmenting a topological map (extracted by means of fuzzy morphological operators) with semantic knowledge using anchoring. A second set of techniques uses classification and clustering for automatic segmentation and labeling of metric maps. For example, the generation of 2D topological maps from metric maps is described in [8] (using AdaBoost), in [9] (using spectral clustering), and in [10] (using Voronoi random fields). Finally, a third set of techniques for object recognition and place categorization uses visual features, such as in [11], or a combination of visual and range information, provided by an RGBD camera, such as in [12]. Although significant progress has been made in fully automated semantic mapping [13], even the most recent approaches still do not scale up in terms of robustness and generality.

In the second category of approaches for *human augmented mapping*, the user role is exploited in grounding symbols into objects that are still autonomously recognized by the robotic platform. In this case, the human-robot interaction is generally uni-modal, and typically achieved through speech. In [14] a system to create conceptual representations of indoor environments is described. A robotic platform owns an *a priori* knowledge about spatial concepts and, through them, builds up an internal representation of the environment acquired through low-level sensors. The user role throughout the acquisition process is to support the robot in place labeling. In [15] the authors present a multi-layered semantic mapping algorithm that combines information about the existence of objects and semantic properties about the space, such as room size, share, and appearance. These properties decouple low-level information from high-level room classification. The user input, whenever provided, is integrated in the system as additional properties about existing objects. In [16] the authors present a mixed initiative strategy for robotic learning by interacting with a user in a joint map acquisition process.

This work however considers only rooms, not enabling the system to embed objects in the semantic map. The approach proposed in [17] alternatively adopts human augmented mapping based on a multivariate probabilistic model to associate a spatial region to a semantic label. A user guide supports a robot in this process, by instructing the robot in selecting the labels. Finally, the method presented in [18] enables robots to efficiently learn human-centric models of their environment from natural language descriptions.

Few approaches aim at a more advanced form of human-robot collaboration, where the user actively cooperates with the robot to build a semantic map, not only for place categorization and labeling, but also for object recognition and positioning. Such an interaction is more complex and requires natural paradigms to avoid a tedious effort for non-expert users. For this reason, multi-modal interaction is preferred to naturally deal with different types of information. For example, the authors in [19] introduce a system to improve the mapping process by clarification dialogs between human and robot, using natural language. A similar construction of the representation is also addressed in [20], where the robot learns the features of the environment through the use of narrated guided tours, and it builds both the metrical and topological representations of the environment during the tour. Spatial and semantic information are then associated to the evolving situations through "events labeling", that occur during a tour, and are later attached to the nodes of a Topological Graph. Finally, the approach in [1] proposes a rich multi-modal interaction, including speech, vision, and the use of a pointing device (similarly to [21]), enabling for a semantic labeling of environment landmarks that makes the knowledge about the environment actually usable. Building upon this work, we propose an incremental on-line semantic mapping able to acquire a richer and more detailed representation of the environment, as explained in the next sections.

3. Aim, Assumptions and Contributions of the Work

Semantic Mapping is the process of gathering information about the environment and creating an abstract representation of it, to support the execution of complex tasks by robots. In literature, semantic maps have been defined as maps containing, in addition to spatial information about the environment, labelings of mapped features to entities of known classes [22]. Formally, Semantic maps can be defined as a triplet [23]

$$SM =< R, G, P > \tag{1}$$

where:

- R is the global reference system in which all the elements of the semantic map are expressed;
- G is a set of geometrical elements obtained as raw sensor data. They are expressed in the reference frame R and describe spatial information in a mathematical form;
- P is a set of predicates that provide for the semantic interpretation of the environment.

As outlined in the previous section, semantic maps can be either created in an automatic process or through the interaction with the user. The process of semantic mapping can be formally defined as:

$$f_{SM} :< M, E >\rightarrow SM \tag{2}$$

where M is a given metric map and $E = \{e_1, e_2, ..., e_n\}$ is a set of events that occur while the robot is exploring the environment. For instance, for automatic semantic mapping approaches, these events might consist of the recognition of a particular object in the environment. Instead, for human aided semantic mapping, such events might consist of different interactions between the robot and the user. These interactions could be realized through different interfaces, such as natural language, tablets or other forms of human-robot interaction.

Independently from the type of approach chosen, the systems proposed in literature are often designed to learn every possible aspect of the environment. Moreover, semantic mapping is seen as a separate and independent process that needs to be carried out before task execution. We note that robots do not need to gather a complete knowledge of the environment to be able to execute tasks. Thus, we propose a shift in perspective, allowing the user to decide what the robot should and should not know in order to carry out the assigned tasks. Moreover, we note that not only semantic mapping does not have to be carried out in a single and independent step, but also that incremental approaches might better suit semantic mapping techniques that rely on the interaction with the user. Hence, to enable the user to better aid the robot, we propose an incremental semantic mapping mechanism based on a multimodal interaction with the user. Formally, incremental semantic mapping can be defined as

$$\phi_{ISM(t)} :< M, e(t), SM(t) >\rightarrow SM' \tag{3}$$

where $e(t)$ is the event occurred at time t, $SM(t)$ is the semantic map built up to time t, and SM' is the new semantic map obtained after applying the function (i.e., after processing $e(t)$).

In order to design a system that supports an incremental semantic mapping, three fundamental questions need to be addressed: i) How to abstractly represent environmental knowledge; ii) How to build this representation; iii) How to use the chosen representation to enable task execution. Solutions to these three questions are present in the literature, but they have been considered as separate components. One contribution of this paper is thus a more comprehensive evaluation of an integrated system that fully implements an incremental semantic mapping approach. In the rest of this section, we describe the motivations and the choices made to develop the proposed solutions to the three issues mentioned above, with respect to previous work in the literature. Instead, in the next sections we provide implementation details and discuss additional implementation choices.

3.1. Environmental Knowledge Representation

As seen in the previous section, multiple representations have been proposed to represent environmental knowledge. Our semantic map mostly relates to the one described by Goerke and Braun [8]. Specifically, we developed a specific four-layered representation of the environment for the setting considered. In fact, while considering a mobile base that needs to operate in a human-populated environment, we require the robot to navigate to certain locations of the environment. We hence build a 2D grid based representation, borrowing the idea from video-games. Indeed, in video-games, the problem of associating logic, symbols and behaviors to areas of the game-scene has been studied since a long time [24, 25]. However, differently from the top-down approach used in video-games, where the logic influences the scene, we propose a bottom-up representation, in which the environment is strictly conditioning the symbolic layer used by the robot. We device this representation to fully support a rich semantic mapping process. In fact, instead of limiting our robot to annotate objects and areas on the metric map, we develop a four-layer representation, introducing an abstract layer that allows for the use of multiple symbolic AI techniques (such as planning, symbol grounding, and qualitative spatial reasoning). This representation is presented in Section 4.

3.2. Knowledge Acquisition

Multiple approaches have been proposed in literature to acquire a specific representation of the environment, ranging from fully automatic techniques[13] to

human-robot interfaces based on natural language [26] or tablets [27]. Our semantic mapping process mostly relates to the one proposed by Diosi *et. al.* [26]. In this work, the user is exploited to label areas visited by the robot through natural language. By contrast, while aiming at deploying our mobile bases in real human-populated environments, we devised a multimodal interface to enable non-expert users to support the robot in the map acquisition process. Specifically, in this process, the user guides the robot through vocal commands, teaching it about objects and rooms in the environment with the aid of a laser pointer. Through this mechanism, any non-expert user is able to support our robots, that are in this way able to enter an unknown environment and incrementally create a specific and rich semantic map. Differently from other works discussed in literature, we leverage human-robot interaction, by allowing the user to decide what aspects of the environment the robot should learn. In this process, the user takes an active role by teaching the robot about objects and areas with the aid of a laser pointer. As outlined in Section 5, this particular interaction has been devised to be intuitively used by any non-expert users, being them elder or young. Moreover, the interaction with the user enables the system to partially handle the dynamic aspects of the environment. In fact, we allow the user to update the semantic map by moving or removing objects in it.

3.3. Knowledge Usage in Common Tasks

In literature, only few works present a fully functioning robotic system able to acquire a semantic map and use it to carry out tasks assigned by users. One of such approaches is presented by Zender *et. al.*[14]. In this work, the authors present a fully functioning system based on natural language interaction. Differently to the approach described in their work, we build a richer four-layered representation that is created incrementally and on-line through human-robot interaction. In our approach, any non-expert user can teach the robot about objects and rooms of the environment through a simple yet effective interaction scheme. The representation obtained through this process is then used to support the execution of complex tasks, issued by the user to the robot through natural language. In fact, we created a fully functional robotic system relying on the semantic map to accomplish tasks assigned by different users. Thanks to our representation, we are able to apply symbolic AI techniques to support the robot during reasoning and planning operations. In particular, by focussing on tasks that require the robot to reach certain positions, we show how the robot can ground such commands into the built semantic map and reason about the objects and rooms described in it. This particular aspect of our prototype is discussed in Section 6.

3.4. Assumptions

The proposed solution aims at being general and domain independent. However, its implementation is based on some assumptions that are discussed in this subsection. These assumptions have been devised for the specific implementation that we have developed, but they do not represent actual limitations of the general approach presented in this work.

We have developed our approach by considering a mobile base that is required to reach certain locations in the environment as specified by the user through natural language. Hence, a 2D semantic map of the environment is considered expressive enough to support task execution. However, the representation could be extended to 3D to support task execution for other types of robots. Moreover, we have assumed that users know how to interact with the robot through the developed interface, having been briefly explained before through demonstrative examples. This assumption has been verified during the system evaluation process described in Section 7. During such experiments, users only required a brief training and a couple of examples to fully learn how to interact with the system. Additionally, we have assumed that the objects pointed by the user through the laser pointer are distinguishable by either depth or texture and color. Again, such an assumption has been verified in the case scenarios analyzed during the evaluation of our prototype. Finally, we have assumed each object in the environment to have an intrinsic front side. Such a front side is used to establish a common reference frame for the qualitative spatial references used by the user in the natural language interactions with the robot, as explained in Section 6. We acknowledge that the assumption that each object has an intrinsic front side is a strong assumption. This assumption has been made observing how users point to objects while interacting with robots. Such an assumption is used in the grounding of the qualitative spatial references found in the command given by the user to the robot. The assumption has been made to establish a common reference frame for the vocal interactions between the human and the robot. A robust object classification and matching with object models would help dropping this assumption, but such an issue was not the focus of this paper.

4. Representation of the Robot's Knowledge

As previously discussed, how to abstractly represent environmental knowledge is the first issue that needs to be addressed while designing an incremental semantic mapping approach. In our robots, environmental knowledge is divided in two layers: *(i)* the *world knowledge*, which encloses the specific knowledge about

the environment acquired by the robot; *(ii)* the *domain knowledge*, consisting in a general knowledge about the domain (Figure 1).

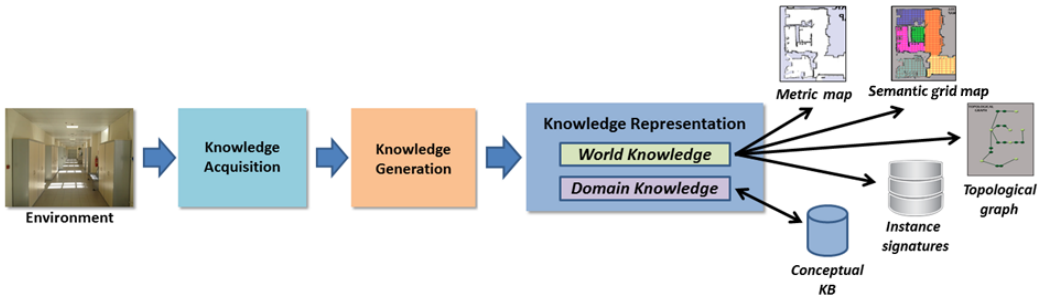

Figure 1: Representation of the robot's knowledge.

It is important to point out that, while the two layers may resemble the extensional (facts) and intentional (general facts, rules and theorems) components of a classical knowledge base [28], here they are independent of each other. In fact, the world knowledge may be inconsistent with the domain knowledge (e.g., the robot may have discovered a fridge in the living room rather than or in addition to the one in the kitchen, while the domain knowledge may state that *"fridges are in the kitchen"*), which is generally used to support the action of the robot, only when specific world knowledge is not available. For example, the robot is only able to reach an object (e.g., the fridge in the living room) if it has been previously discovered; else, if there is no specific knowledge about the position of the fridge, the system will query a general domain knowledge to find out its possible location (e.g., the kitchen). To better explain this point, the fact that a fridge has been found, for example, in the living room and that the robot knows through human dialog that a particular kitchen has no fridges (in contrast with the conceptual knowledge base that states that *"fridges are located in kitchens"*) does not negatively affect any reasoning of the robot when a fridge is discovered elsewhere.

In this paper, we will use the term $\mathcal{C}oncept$ to refer to a set of symbols used in the conceptual KB and to denote general concepts (i.e., abstraction of objects or locations). For example, $Fridge$ and $Kitchen$ are concepts in the general domain knowledge. The term $\mathcal{L}abel$ will be used, instead, to refer to a set of symbols that indicate specific instances of objects or locations. For example, $fridge1$ could be a label denoting a particular fridge, while $kitchen$ could be a label denoting a particular kitchen. In the notation used in this paper, concept names will be capitalized, while labels will have all lower-case letters, typically followed by

a digit. Finally, the associations between labels and concepts will be denoted as $label \mapsto Concept$. For example, $fridge1 \mapsto Fridge$ denotes that the label $fridge1$ is an instance of $Fridge$ (i.e., $fridge1$ is a fridge). In the remaining part of this section, we will describe in detail how the aforementioned knowledge structures, pictured in Figure 1, are subdivided.

4.1. Domain Knowledge

In previous work, domain knowledge has typically been characterized as a conceptual knowledge base, representing a hierarchy of concepts, including their properties and relations, *a priori* asserted as representative of any environment. The conceptual knowledge base usually contains a taxonomy of the concepts involved in the environment tied by an *is-a* relation, as well as their properties and relations [7, 29]. These concepts are used in the world knowledge to characterize the specific instances of the environment, as previously explained. In our representation three top-most classes have been considered: *Areas*, *Structural Elements*, and *Objects*. *Areas* denote places in the environment (corridors, rooms, etc.), *Structural Elements* are entities that form the environment and that topologically connect areas (windows, doors, etc.), while *Objects* are elements in the environment not related to its structure and located within areas (printers, tables, etc.). A snapshot of a portion of the conceptual knowledge base used in the experiments is reported in Figure 2.

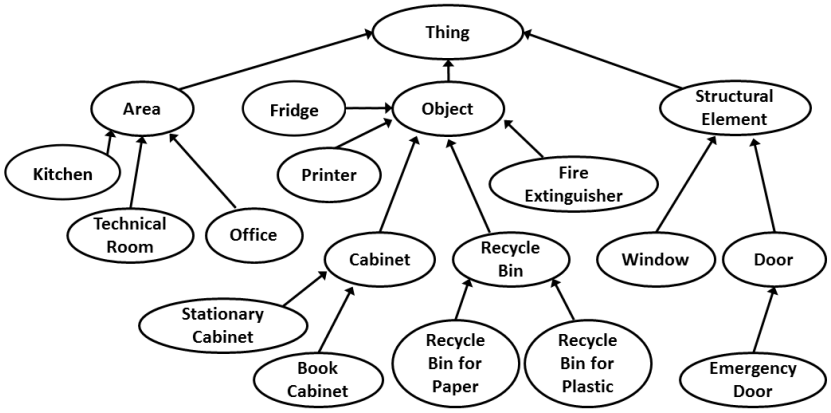

Figure 2: A fragment of the conceptual knowledge base used in the experiments.

Finally, in the conceptual KB we have included synonyms. It is possible, in fact, to refer to the same object with different natural language expressions (e.g.,

referring to a *"plug"* also with the word *"socket"*). Moreover, we use the structure of the taxonomy during the natural language interactions with a user, by looking at a more specific or more generic concept (e.g., it is possible to refer to a *"book cabinet"* with the word *"cabinet"* and vice versa), as shown in Section 6.

4.2. World Knowledge

A semantic map is typically characterized as a low-level representation of the map (i.e., a grid), that is labeled with symbols. Such symbols denote, for example, the area corresponding to a room (e.g., a corridor) or the presence of objects and structural elements of the environment. Our representation builds on a similar structure, supporting however a much more detailed description, based on the interaction and hints provided by the user. Such an environment representation is called World Knowledge and it is composed by the following elements:

Metric map. The *metric map* is represented as an occupancy grid generated by a SLAM method. This map has usually a fine discretization and it is used for low-level robot tasks, such as localization and navigation. In Figure 3 a metric map with a resolution of 5 cm generated is shown in the background of the image, where black pixels represent occupied cells and grey pixels resemble empty spaces.

Semantic Grid Map. The *Semantic Grid Map* is represented as a discretization of the environment in cells of variable size. Each cell represents a portion of a physical area and it is an abstraction of locations that are not distinguishable from the point of view of robotic high-level behaviors. The Semantic Grid Map also includes a connectivity relation $Connect \subseteq Cell \times Cell$, that describes the connectivity between adjacent cells. In Figure 3, the Semantic Grid Map contains multiple cells that are delimited by borders in different colors. Each color corresponds to an area reported in the legend to the right. Connectivity relations between cells are not explicitly shown in Figure 3, but they can be derived by looking at adjacent cells. As we will show in the next sections, this representation is used for high-level forms of reasoning, such as qualitative spatial reasoning.

Topological Graph. The *Topological Graph* is a graph whose nodes are locations associated to cells in the Semantic Grid Map and edges are connections between these locations. In Figure 3, the Topological Graph is depicted as a graph connecting oval nodes. Dark-colored nodes correspond to static locations, while light-colored nodes correspond to variable locations, which are associated to the main areas of the environment. The static locations denote specific positions that

are of interest for the robot tasks (e.g., the position to enter in a room), while the variable locations are used to denote areas, where the instantiation of the position for a navigation behavior is executed at run-time, depending on the current status of the robot and on its goals. Since the Topological Graph is used by the robot for navigation purposes, the edges also contain the specific navigation behavior that is required for the robot to move from one location to another. In this way, the Topological Graph is also used to generate appropriate sequences of behaviors to achieve the robot's navigation goals. For example, by knowing that there is a door that must be traversed to go from a point to another, a particular behavior must be adopted by the robot that will be included in the Topological Graph.

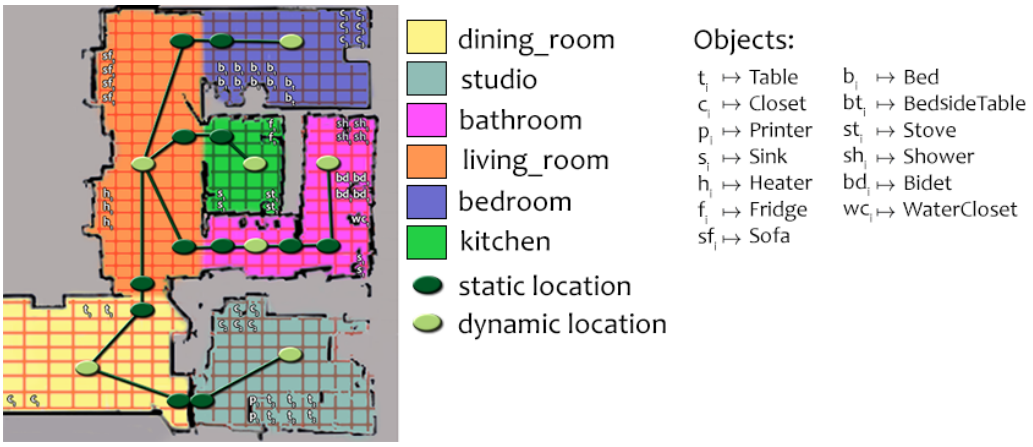

Figure 3: An example of world knowledge.

Finally, our specific representation of the environment is populated with **Instance signatures**. The *instance signatures* are represented as a data base of structured data, where each instance has a unique label ($l \in \mathcal{L}abel$), an associated concept ($C \in \mathcal{C}oncept$) such that $l \mapsto C$, and a set of properties expressed as attribute-value pairs. Additionally, every label is associated to a set of cells ($cells = \{c_1, c_2, ..., c_n\}$) and topological nodes ($nodes = \{n_1, n_2, ..., n_k\}$) associated to them. For example, the green cells labeled with f_1 (top-right corner of the kitchen area) are mapped with the labels $\{f_1, kitchen\}$, that are associated with the concepts $Fridge$ and $Kitchen$, i.e. $f_1 \mapsto Fridge$ and $kitchen \mapsto Kitchen$, respectively. Thus, the corresponding area is characterized as being occupied by a fridge and as belonging to the kitchen. Additionally, f_1 can have the following properties: position = $< x, y, \theta >$, color = $white$, open = $false$. This particular four-layered representation has been devised to be easily used with classical

AI techniques. Specifically, the Topological Graph layer is used to support path-planning. Instead, the Grid Map is used to support qualitative spatial reasoning and natural language command grounding.

5. Acquisition of the Robot's Knowledge

Once we are able to represent environmental knowledge, we need to address the problem of how to acquire such representation. In our robots, the semantic map is built by using two different modalities: a inizialization phase and the on-line incremental extension of the initial knowledge.

The inizialization phase aims at creating an initial representation of the robot's knowledge. Such a process is a standard approach used by the majority of the semantic mapping methods found in literature. During this phase we have allowed the user to specify the objects enclosed in the environment while acquiring the metric map. The inizialization phase can be divided into two steps: the *Metric Map and Instance Signatures Construction* and the *Semantic Grid Map and Topological Graph Generation*. During the first step, a 2D metric map is generated through a graph-based SLAM approach [30, 31] and the initial set of instance signatures is created. In the second step, a grid-based topological representation, called *Semantic Grid Map*, is obtained by using the 2D metric map and a graph, called *Topological Graph*, needed by the robot to perform high level behaviors is built by processing the instance signatures and the Semantic Grid Map.

The on-line modality can be used to enrich the initial knowledge including additional objects and additional properties. It is worth noticing that different users can add knowledge to the system, thus the system has to deal with the possible addition of multiple tags for the same object. The details about the two acquisition modalities are given in the following.

5.1. Metric Map and Instance Signatures Construction

In the first step of the inizialization phase the robot is used to navigate the environment in order to create a 2D metric map and to register the positions of the different objects of interest. As already described, during the guided tour of the environment the user can tag a specific object by using a commercial laser pointer (Figure 4a). While the object is pointed through the laser, the user has to name the object, so that the *Vocal Interface* module can register the semantic label that will be assigned to it [1].

The *Dot Detection* module is responsible for detecting the laser dot by using the RGBD data. By exploiting the odometry data to detect when the robot is not

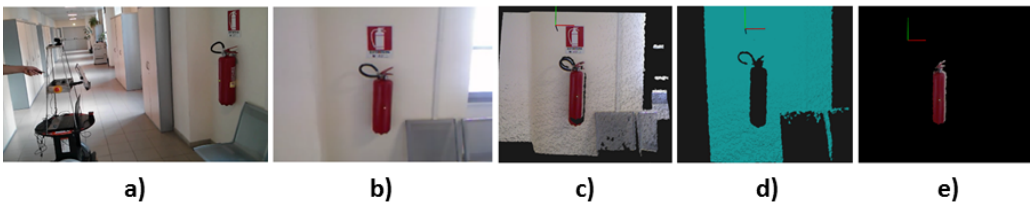

Figure 4: Object tagging. a) The object is tagged with the laser pointer. b) The color image from the RGBD sensor is used to locate the laser dot. c) The 3D point cloud is extracted. d) The planes not containing the laser dot are discarded. e) The dot is used as seed point to segment the tagged object.

moving, a set of image samples is collected over 3 seconds in order to generate a background model of the captured scene by means of a multimodal statistical approach [32]. After 3 seconds, the robot communicates to the user that it is ready for acquiring the tag and a background subtraction process is carried out over the current RGB image (see Figure 4b), thus obtaining a foreground mask. The background subtraction step allows for filtering out possible false positives (e.g., due to illumination sources coming from windows or ceiling lighting). The foreground pixels are then converted into the HSV color space to search for the specific color values of the light dot generated by the laser. To further refine the results, the depth information is used to discard all the points that are above a certain height (2.00 meters in our case) or that are too far from the camera (3.00 meters). The output of the *Dot Detection* module is the image coordinates (i, j) of the laser dot in the RGB image.

Once the dot is found, the *Object Pose Estimation* module is responsible for finding the 2D position x, y and the bearing θ of the tagged element in the metric map [33]. The object pose in global coordinates (x, y, θ) is calculated by taking into account the normal corresponding to the surface of the segmented object in the reference frame of the RGBD sensor and then applying a first transformation to the reference frame of the robot and a second transformation to the reference frame of the map (this is given by a standard self-localization module, i.e., ROS-AMCL).

The coordinates of the laser dot are also used by the *Object Segmentation* module that aims at segmenting the pointed object in order to extract its width (W), height (H), and depth (D) as well as its color properties. The laser dot is then projected onto the 3D point cloud of the scene (Figure 4c). All the planes in the scene are extracted from the 3D point cloud and those that do not contain the dot are discarded (Figure 4d). The remaining points are analyzed to segment

the shape of the object of interest by using the laser dot as a seed point for the expansion and depth discontinuities as stopping criterion (Figure 4e).

The color properties are obtained by analyzing the set of pixels belonging to the segmented object. In Figure 5 an example of color information extraction is reported. The set of points in the laser scan corresponding to the tagged object are detected by extracting the neighbor points around the projection of the laser dot onto the laser scan (Figure 5b). In order to segment the tagged object, the point cloud is rotated of an angle equal to $\frac{\pi}{2} - \theta$ around the axis orthogonal to the floor. In this way, all the objects are treated as being acquired by the same angle of view, thus allowing for a normalization of the point cloud of the tagged object. The final poses of the tagged objects are stored as instance signatures together with the corresponding properties (color and size).

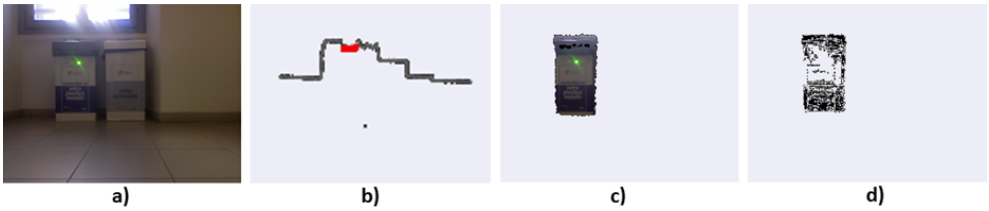

Figure 5: Extracting color information from a tagged object. a) The RGB image. b) The laser scan received from the laser range finder. The detected object is highlighted in red. c) The segmented point cloud. d) The color mask corresponding to the blue color.

5.2. Semantic Grid Map and Topological Graph Generation

While there are tools and methodologies for building metric maps, and, to some extent, topological graphs, all the knowledge about the environment must usually be provided to the robot in a suitable format by an expert. This is not satisfactory if symbolic and semantic knowledge has to be acquired incrementally or if this process has to be performed on top of a metric map. In fact, neither the occupancy grid, nor the topological graph provide a good representation for adding semantic knowledge. The occupancy grid, in fact, since it has a fine discretization, which is good for a rich description of structural details, does not allow to be easily translated into a symbolic representation, needed for a high-level form of spatial reasoning. The topological graph is suitable for representing the connectivity of the environment with respect to a set of symbols associated to it, but its structure can become very nested and complex, leading to difficulties in physically locating the areas of interest. Moreover, updating the topological graph can

become a challenging task, especially if the structure of the environment in the map can change due to the continuous update of the representation.

Given the above considerations, an intermediate abstraction layer, strictly linked to the metric map is indeed useful for both acquiring semantic knowledge and enabling the robot to build a topological graph on top of the actual structure of the building, thus keeping in the high-level description a direct connection with the physical world. Such an abstraction layer can be inserted in a processing chain from the metric map to the symbolic representation of knowledge about the environment, independently both from the mapping methods and from the knowledge acquisition techniques. The Semantic Grid Map and the Topological Graph are generated automatically in the second step of the inizialization phase.

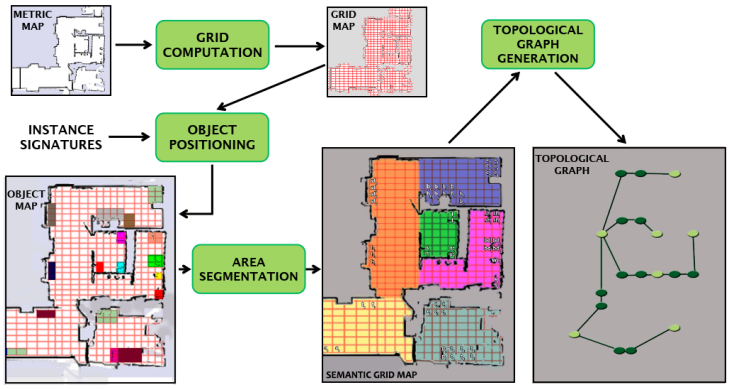

Figure 6: Semantic Grid Map and Topological Graph generation. The metric map is enriched with semantic information to obtain the Semantic Grid Map. The instance signatures and the Semantic Grid Map are used to create a Topological Graph.

The Semantic Grid Map contains a high-level description about the regions, structural elements, and objects contained in the environment. It is generated using both the instance signatures and the 2D metric map. The process to generate the Semantic Grid Map is carried out by the modules reported in Figure 6 and briefly described below (for details see [5]).

The *Grid Computation* module is used to generate a rasterization of the 2D metric map into a grid-based topological representation called *grid map* (not to be confused with the Semantic Grid Map, which is a grid map enriched with semantic information). The operation is accomplished by applying the Hough Transform at different resolution levels to identify the lines passing through the walls. In this way, it is possible to find all the main walls in the map. The cells of the grid have different size with respect to the amount of relevant features in

the considered area (e.g., the cells in the corridor are wider then the ones in the offices). In detail, the cells have been chosen to be variable-sized since we want the grid to be built on the basis of the walls and to be consistent with the structure of the environment, which could not be achieved using a fixed-size grid. The cells have a size between $x_{\min} \cdot y_{\min}$ and $2x_{\min} \cdot 2y_{\min}$, where $x_{\min}$ and $y_{\min}$ are calculated by extending the lines generated through the wall detection to the whole image and computing the minimum horizontal and vertical distances. Of course, such a discretization could affect the object localization for small objects, but in general this error is acceptable. Specifically, we use the center of the grid cell as a reference for the localization of the objects since it is acceptable both from a qualitative representation and task execution point of view.

The *Object Positioning* module is responsible for placing the representations of the objects and structural elements tagged in the environment into the grid map. The objects' pose, shape and size, memorized in the instance signature data base in the previous step, are used to label the cells in the grid map. The output is an *object map*, containing labeled cells representing tagged objects.

The *Area Segmentation* module is applied to further enrich the object map, by automatically labeling the different areas of the environment. The object map is divided into different rooms by using the watershed algorithm [34]. The segmentation algorithm is enhanced by considering the tagged objects and structural elements previously included in the environment. For example, the knowledge about the position of doors is used to separate different areas of the environment. Such knowledge can be gathered both through automatic approaches or through user interaction. Since the automatic approaches that we tested yielded some false positives, during our experiments we relied on a completely human-based knowledge acquisition. The output of the area segmentation module is a *Semantic Grid Map*, containing labeled cells representing tagged objects and areas of the environment.

The *Topological Graph Generation* module completes the inizialization phase by creating a *Topological Graph*. Such a graph embodies the information needed to the robot for navigating and acting in the environment and it is generated by exploiting the knowledge contained both in the Semantic Grid Map and in the instance signatures.

5.3. On-line Knowledge Acquisition

When the inizialization phase is terminated, the robot is ready to incrementally extend its initial knowledge by adding new objects to the Semantic Grid Map. Similarly to the previous phase, the on-line acquisition is also achieved by tagging

objects with the laser pointer (note that we still focus only on static objects). The system operates as for the previous phase, but at the moment of the insertion in the knowledge base, the system updates the symbolic representation accordingly: the Semantic Grid Map and the instance signatures are modified in order to reflect the new situation generated by the addition of a novel object, as discussed in [4].

However, since the system is used incrementally and possibly multiple users can interact with it, some difficulties can arise in the maintenance of the knowledge. For example, a user may try to tag an object that is already present in the knowledge representation or he may want to tag an object that is in a cell already occupied by another object. In the first case, the system can detect the problem, since the Semantic Grid Map contains a cell labeled with the same name used by the current user, and it can avoid to add a duplicate label in the Semantic Grid Map. An example for the second case can occur if we want to tag a plug that is located under an already tagged white-board. In this case, since the names of the objects are different, the system allows to insert more than one label in a single cell, leaving to an interaction with the user the choice of keeping or deleting the existing object.

During the on-line acquisition phase, the system can recognize objects already present in the knowledge representation. Considering again a plug as an example, if a user points to it in position (x_1, y_1) and a different plug in position (x_2, y_2) has been previously memorized, the system tries to recognize the object without the explicit grounding by the user.

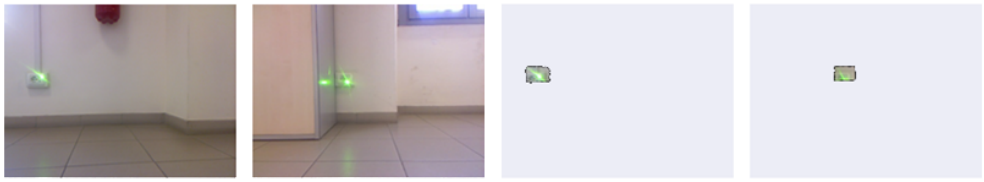

Figure 7: The system tries to recognize the class of an object if another instance of the same class has been already memorized by using the SURF features as descriptors.

In Figure 7 two plugs are tagged. Since the *Object Segmentation* module transforms each segmented object in the same angle of view, the recognition task is simplified. The two objects are therefore matched by using the SURF features as descriptors and by exploiting the FLANN algorithm to find the correct correspondences [35].

6. Robot Behavior using the knowledge base

The semantic map acquired with the previously described method is used for knowledge maintenance and task execution. These two actions, as well as all the other behaviors needed by the robot to act in the environment, have been modeled using the Petri Net Plans (PNP) formalism [36].

6.1. Petri Net Plans

PNPs are particular Petri Nets [37] whose operational semantics is enriched with the use of conditions, which are verified at run-time by querying an external Knowledge Base. No restriction is imposed on the knowledge base, which is supposed to be updated by other modules according to the agent's perceptions.

PNPs are composed by basic actions that can be combined through a set of possible operators in order to express complex execution paradigms. The basic actions of a PNP are *ordinary actions* (Fig. 8a), which represent a durative action, and *sensing actions* (Fig. 8b), which represent procedures whose outcomes depend on one or more conditions to be checked in the knowledge base, similarly to if-statements.

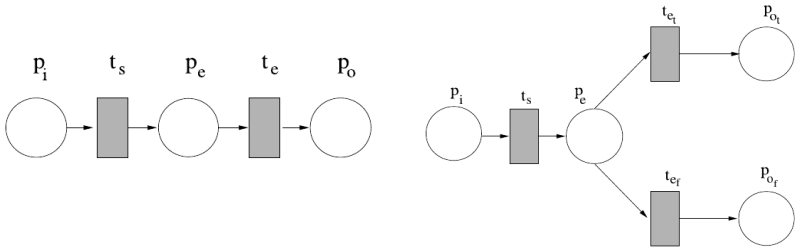

Figure 8: Ordinary and sensing actions in a Petri Net Plan

These basic actions can be seen as a specific concatenation of places and transitions of a Petri Net. Each element of an action represents a specific aspect:

- *Input* places (p_i) model the initial configurations of the network before the action has been executed.
- *Execution* places (p_e) model the configurations during which the action is executed. These places are also used to create hierarchical plans by calling a sub-plan while visiting this place.

- *Output* places (p_o) model the final configurations of the network achieved after the execution of the action.
- *Start* transitions (t_s) model the events that trigger the execution of the action.
- *End* transitions (t_e) model the events that trigger the stop of the action. In a sensing action there is a true (t_{e_t}) and false (t_{e_f}) end transition that fire depending on the outcome of the query made to the knowledge base.

These basic actions can be composed using a set of possible operators for single and multi-agent plans. Here we informally introduce two single-agent operators, referring the reader to the original paper [36] for a formal and complete definition of the available operators:

- *Sequence*. A sequence is obtained by merging two places of different PNPs. It can be applied to any place exception made for the execution ones, as shown in Figure 9.
- *Interrupt*. Interrupts connect the execution place of an action (or sub-plan) to a non-execution place of another network. It causes the temporary termination of the action (or sub-plan) under anomalous conditions, as shown in Figure 10. Interrupts are often used to repeat a portion of a plan that did not realize its post-conditions in order to implement while-loops.

Places in the PNP can be used to denote states of execution of the plan, while transitions represent conditions or events that allow for state changes. PNP supports a hierarchical representation, thus execution tokens in one PNP can refer both to atomic actions and to other PNPs (also called sub-PNPs) that can be executed in parallel. Moreover, the presence of markers in a place is used to define a *context* and these contexts are then used for taking contextual decisions. For example, contexts are used for loading the proper grammar at a specific phase of the plan. Specifically, in our system we have implemented a simple plan that, based on the input given by the speech component, activates one of the eleven possible actions (Turn, LookAt, Follow, Memorize, Forget, Update, Home, GoToPlace, GetCloser, TellObjectKnown, RecognizeObject). When such an action ends or is interrupted, the system waits for another command given by the user. In the rest of this section, we present two examples of such PNP actions in order to

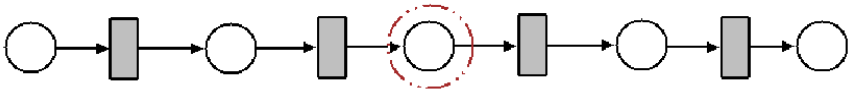

Figure 9: The sequence operator

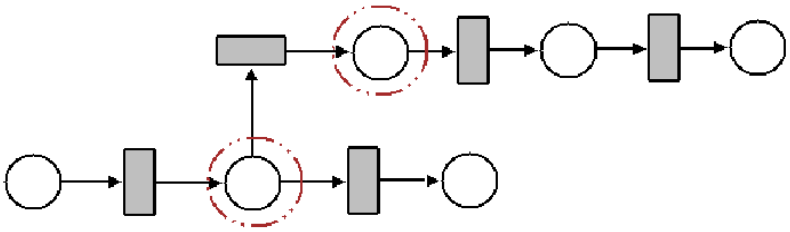

Figure 10: The interrupt operator

show how the knowledge can be acquired and used (see PNP[1] for a more detailed description).

6.2. Knowledge Base Maintenance

First, the knowledge acquired by the robot is used for maintaining the knowledge base. In particular, three operations of memorizing, removing, and better specifing the objects found in the environment have been implemented in the system. Figure 11 shows a simplified PNP used for memorizing the objects located in the environment through the interaction with the user.

In this plan three `Say`, one `AddObject`, one `AskForHelp` and one `KBQuery` actions are used to reach the goal. The `Say` actions represent a communication with the user through the text to speech system. `AddObject` is the action used to store all the information of the object tagged in the semantic map, while `AskForHelp` represents the action responsible of resolving the possible conflicts generated by the addition of a new object to the semantic map. Finally, `KBQuery` is the action that queries the knowledge base of the robot. The conditions used in this plan are the following:

[1] http://pnp.dis.uniroma1.it

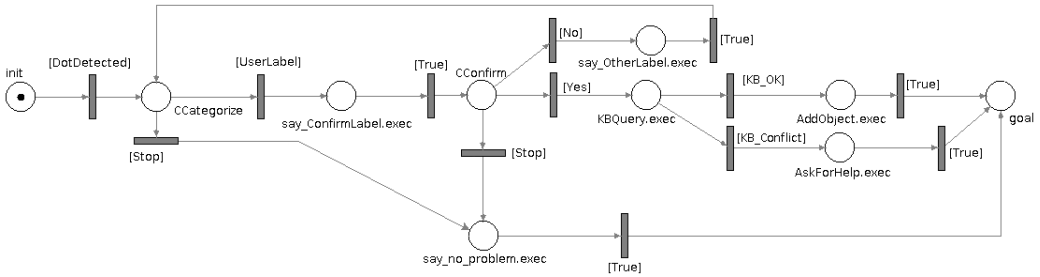

Figure 11: Simplified version of the Memorize PNP used in the experiments.

1. `DotDetected`: the user has pointed to the object that needs to be memorized and the laser pointer has been correctly recognized;
2. `UserLabel`: the user has uttered the label for the object to be memorized;
3. `Yes`, `No`: the user has confirmed or not the label understood by the robot for the object to be memorized;
4. `KB_OK`, `KB_Conflict`: possible answers obtained from the query posed to the knowledge base (`KB_OK`: the place/object tagged by the user does not conflict with anything previously memorized; `KB_Conflict`: the place/object tagged by the user conflicts with something previously memorized and the user help is therefore required).

Finally, there are two context places (`CCategorize`, `CConfirm`), that are used to activate contextual behaviors. Specifically, `CCategorize` and `CConfirm` activate special grammars to parse user sentences, representing the label of the object tagged and the feedback to the robot confirmation request, respectively. When a new object is tagged, the knowledge base is queried for other objects located in the same position of the new element. If one or more objects are found, the system tries to resolve the problem by comparing the labels of the new element with those given to the existing ones. If such a label is found to be a synonym or related by a more generic/specific relation to any of the existing ones, the system automatically resolves the conflict by acknowledging the user of the presence of the existing element and by updating the label attached to the object if the new label is more specific than the known one. An example of execution of this plan is given in the following table.

User	*"Memorize."*
Robot	*"I am ready to memorize an object."*
The user points at the object until an auditory feedback is received.	
User	*"This is the book cabinet."*
Robot	*"I know this is a cabinet. Now I know that this is also a book cabinet."*

The three basic operations (i.e., memorize, remove, and specify) have also been combined in order to create more complex behaviors. For example, when the system cannot resolve the conflict, the user is notified that one or more objects are known to be in the same position of the newly tagged one and she/he is asked whether the new element should be added or substituted to any of the previous ones. An example of such an interaction is given in the following table.

User	*"Memorize."*
Robot	*"I am ready to memorize an object."*
The user points at the object until an auditory feedback is received.	
User	*"This is the fire extinguisher."*
Robot	*"I know that a socket is also here. Should I keep it in my memory?"*
User	*"No."*
Robot	*"I removed the socket and I memorized the fire extinguisher."*

6.3. Task Execution

The second case in which the knowledge acquired by the robot is used is task execution. In this case, the knowledge is used for executing the task of moving in front of a specific object by interacting with a user. A simplified version of the GoToPlace sub-PNP relative to such a behavior is given in Figure 12.

Specifically, the plan uses one `GoTo`, three `KBQuery` and two `Say` actions to reach the goal state. `GoTo` is the action used to move the robot to a particular place, `KBQuery` queries the knowledge base of the robot, while `Say` represents a communication with the user through the text to speech system. The conditions in this plan are:

1. `UserRequest`: The user has asked the robot to go to a place/object (the name of the place is stored in the robot's memory during the execution of this plan);
2. `KB_OK`, `KB_NONE`, `KB_MULTI`: Possible answers obtained from the query posed to the knowledge base (`KB_OK` when the place/object given by the user is unique in the knowledge base and its position in the environment is known; `KB_NONE` when there is no place/object with such name in the

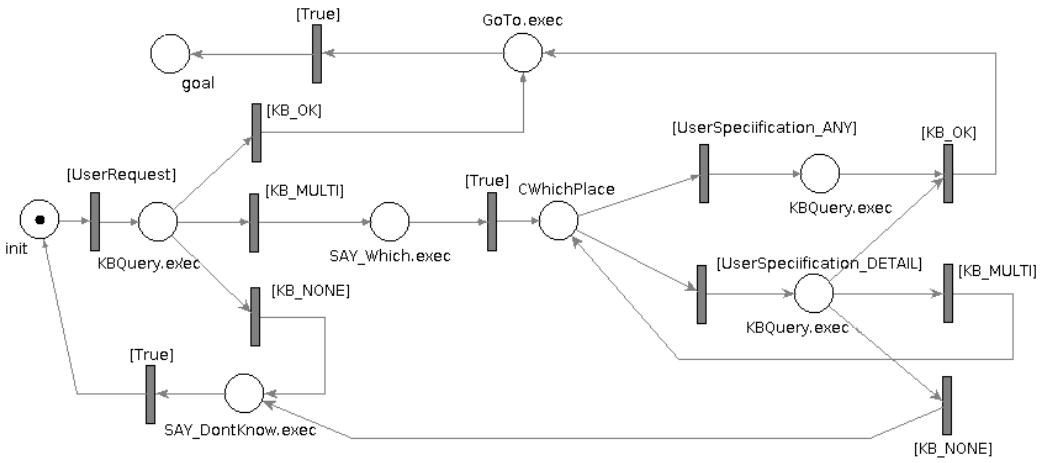

Figure 12: Simplified version of the GoToPlace PNP used in the experiments.

knowledge base; `KB_MULTI` when there are multiple places/objects with the same name in the knowledge base);

3. `UserSpecification_ANY`, `UserSpecification_DETAILS`: The user has specified which place/object by either answering that any place/object is fine or by giving a more detailed description that is the subject of a new query to the knowledge base.

Finally, there is one context place (`CWhichPlace`) that is used to activate contextual behaviors. In particular, such a context place activates a special grammar used to parse user sentences representing a specification of a place or an object.

The specification of a place or an object can be given either by using spatial relations or synonyms and more general/specific concepts. While the latter specifications can be implemented by embedding a vocabulary in the robot knowledge base, for spatial relations a dedicated qualitative spatial reasoner (QSR) has been devised[2]. In particular, three vicinity relations (*near*, *next to*, and *nearest*), their three opposite relations (*far*, *not next to*, and *furthest*) and four orientations (*behind*, *in front*, *on the right*, and *on the left*) have been implemented. By defining C_{Loc} and C_{Ref} the set of cells belonging to the Semantic Grid Map that include

[2]Note that spatial relations could be learnt over time [38]. However, since this topic is not the focus of our paper, we adopted a hand-crafted spatial reasoner to show the usefulness of our semantic map.

a portion of the objects Loc and Ref respectively, we say that Loc has a vicinity relation with Ref if and only if:

$$d(centroid(C_{Loc}), centroid(C_{Ref})) < t$$

where d is the euclidean distance, t is a threshold constant, and $centroid(x)$ is a function that takes a set of cells x in input and returns the coordinates of its centroid in the metric map coordinate system. By specifying a threshold constant for both the relations *near* and *next to*, t_{near} and t_{next} respectively, we therefore define the six distance relations (the nearest attribute is computed by finding the object that minimizes the above defined distance). In order to define the orientation relations, as in [39], we exploited the "intrinsic front side" of the objects, identified with the normal of the surface tagged by the user during the learning phase previously described. Specifically, we have used such a normal to define a forward orientation, later deriving, by rotating clock-wise, respectively the concept of left, backwards, and right regions. By defining the general concept of directions, we adopted the *cone-based* approach [40] to explicate the four directional relations, starting from the centroid of the reference object. By defining A_R^{Ref} the area corresponding to a region in the direction R with respect to the reference object Ref (e.g., $A_{right}^{cabinet}$ is the area on the right of the cabinet), in order for Loc to belong to that particular area, we require that:

$$centroid(C_{Loc}) \in A_R^{Ref}$$

where, again, the $centroid(x)$ is as above defined. Two examples of execution of this plan are given in the following tables.

User	*"Go to the socket."*
Robot	*"There are many sockets in the environment. Which one do you mean?"*
User	*"The one close to the emergency door."*
Robot	*"OK. I am going to the socket."*

User	*"Go to the socket."*
Robot	*"There are many sockets in the environment. Which one do you mean?"*
User	*"Anyone."*
Robot	*"OK. I am going to the nearest socket."*

In these examples, our qualitative spatial reasoner has been used to correctly execute the commands. In particular, in the first example, the robot uses the near

relation to ground the disambiguating command to the socket close the emergency door. Instead, in the second example, the robot goes to the nearest socket. In fact, when the user instructs the robot to reach any socket, the system chooses, as a target, the nearest socket to minimize energy consumption.

When the location to be reached is retrieved from the KB and the user is acknowledged with a feedback sentence similar to the ones described above, the system checks whether the robot is located inside the same room where the target is placed. If so, the robot navigates directly to the target without using the Topological Graph, while if not, the system searches the graph for the shortest path to reach the entrance of the *Area* containing the target cell and, from there, it behaves as in the previous case.

7. Experimental Validation

The main goal of the experiments reported in this section is to show the feasibility, the effectiveness, and the robustness of the proposed approach for on-line acquisition of knowledge through human interaction. In order to validate our approach, we have carried out experiments both on single components and on the whole system. These experiments have been conducted by executing the robot behaviors described in the previous section, across different settings, including different robots, several users, and two very different kinds of environments: a home environment and our department. Multiple videos showing the execution of these behaviors can be found in the dedicated web page[3]. In particular, our approach has been implemented on a mobile base derived from Segway (Figure 13a), on a Videre Design platform (Figure 13b), on a Turtlebot (Figure 13c), and on a MARRtino, a mobile base built by our students (Figure 13d). The standard sensor setting of each mobile base includes a laser range finder for localization and navigation and one (or two) RGBD sensor(s) for the laser dot detection and the object segmentation. The robots carry two additional components, the robot software and the speech component, that are run respectively by a laptop placed on the robot and by a tablet that can be either held by the user or placed on the robotic platform. The robot software, including both core robotic functionalities and the system for acquiring and handling the knowledge about the environment, has been implemented on ROS[4]. The speech processing is based on the imple-

[3] `www.dis.uniroma1.it/~gemignani/Articles/LivingWithRobots.html`
[4] http://www.ros.org

mentation developed through Speaky for Robot[5] [3].

Figure 13: Robots on which our system has been deployed. a) Mobile base derived from Segway. b) Videre Design platform. c) Turtlebot. d) MARRtino, a mobile base built by our students.

7.1. System Component Evaluation

The Semantic Grid Map generation process, the object segmentation module, and the spatial-reasoner are the three key components of our system. Such components have been quantitatively evaluated in order to measure the performance of our implementation.

Semantic Grid Map Generation. In order to evaluate the Semantic Grid Map representation, two experiments have been conducted. In the first experiment the publicly available Radish Data Set[6] has been used with the purpose of evaluating, in terms of compactness of the representation, the effectiveness of the proposed discretization, when dealing both with regular and irregular indoor environments. In particular, six 2D metric maps[7], obtained from different SLAM methods, have been processed to extract the Semantic Grid Map on a number of different occupancy grids. In addition, four maps generated by our robots have been processed, for a total of ten different environments. Qualitative results are shown in Figure 14. A very good correspondence between the real environment and the structured information within the Semantic Grid Map is achieved. Even if the Semantic Grid Map generation process has been developed for dealing with ordinary (regular) buildings, it provides good results also in environments with irregular edges

[5] http://labrococo.dis.uniroma1.it/?q=s4r

[6] `http://radish.sourceforge.net/index.php`

[7] The following maps have been used: *albert-b-laser*, *ap_hill_07b*, *ubremen-cartesium*, *intel_lab*, *belgioioso* and a portion of *hospital_floorplan_fort_sam_houston*

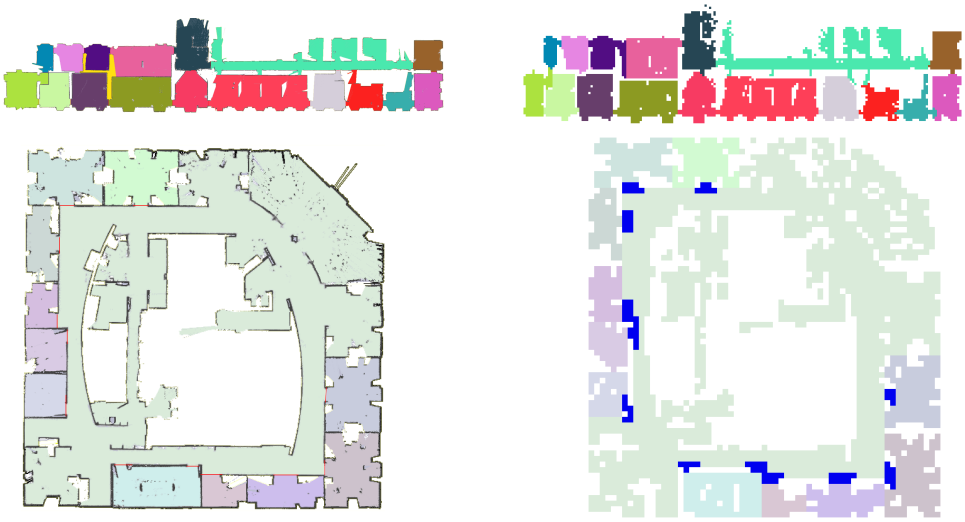

Figure 14: Representation obtained for two different maps from the Radish Data Set. Different rows refer to different maps.

(e.g., see the last row of Figure 14). Quantitative data about the reduction in terms of the size of the representation allowed by our approach are reported in Table 1. The results underline that the Semantic Grid Map representation is, on average, around 98% more compact than the corresponding bitmap (about two orders of magnitude, depending on the complexity of the map).

When objects are located in the Semantic Grid Map, errors in their positions and dimensions are introduced because of the discretization of this map. A second experiment was thus performed in order to evaluate such errors. To this end, we considered 3 different categories and 15 instances for each category of object in our department. To avoid the errors introduced in the perception layer, we measured the ground truth of objects' position and size and we evaluated their error when represented in the Semantic Grid Map. By overlapping the object representations from the Semantic Grid Map (SGM) onto the metric map (i.e., drawing the occupied cells on the metric map), we compared their size in terms of pixels with respect to a ground truth map (GT), thus measuring the error introduced by the use of the Semantic Grid Map. In detail, we measured the percentage of the error, with respect to the ground truth value, for the width (W) and the depth (D) of each object:

$$e_W = \frac{|W_{SGM} - W_{GT}|}{W_{GT}} \quad e_D = \frac{|D_{SGM} - D_{GT}|}{D_{GT}}$$

Table 1: Comparison between the pixels of each processed metric map and the cells of the corresponding Semantic Grid Map.

Map	**Pixels**	**Cells**	**Reduction (%)**
Belgioioso	768 792	11 600	98.5
Dis-B1	1 080 700	10 290	99.0
Dis-B1-part	501 840	7372	98.5
Dis-Basement	992 785	13 455	98.6
FortAPHill	534 520	7878	98.5
Freiburg	335 248	4794	98.5
HospitalPart	30 000	285	99.0
Intel	336 399	4473	98.6
Scheggia	92 984	1116	98.8
UBremen	831 264	10 962	98.6

Table 2: Comparison of the size of the objects in the Semantic Grid Map with respect to ground truth values, for 3 categories and 15 instances of objects for each category. The measures express the mean (μ) and standard deviation (σ) error percentage with respect to ground truth values.

Object	**Num. Samples**	$\mu_{Cells} \pm \sigma$	$\mu_{\mathbf{eW}} \pm \sigma$ **(%)**	$\mu_{\mathbf{eD}} \pm \sigma$ **(%)**
Cabinet	15	3.65 ± 1.29	30 ± 1	23 ± 16
Fire Exting.	15	1.05 ± 0.22	112 ± 25	66 ± 19
Recycle Bin	15	3.80 ± 0.6	64 ± 25	82 ± 21

The results obtained (reported in Table 2) clearly depend on the size of the considered object and on the granularity of the discretization in the portion of the map where the object is located. For example, the representation error for a cabinet, which is big in size and can be easily detected from the metric map, is usually small. The error in the case of a recycle bin is instead higher, since (usually) 4 cells are required to fully represent such an object. Indeed, even if a single cell representation would decrease the error, it would not be suitable for safe navigation. In the case of the fire extinguisher, the error strictly depends on the granularity of the grid, since this object almost always requires one single cell to be represented. In general, even if the error for the object sizes can reach significant values, the loss of precision is still acceptable from the point of view of task execution, since after reaching the desired location on the semantic map, an accurate localization of the objects is performed through perception.

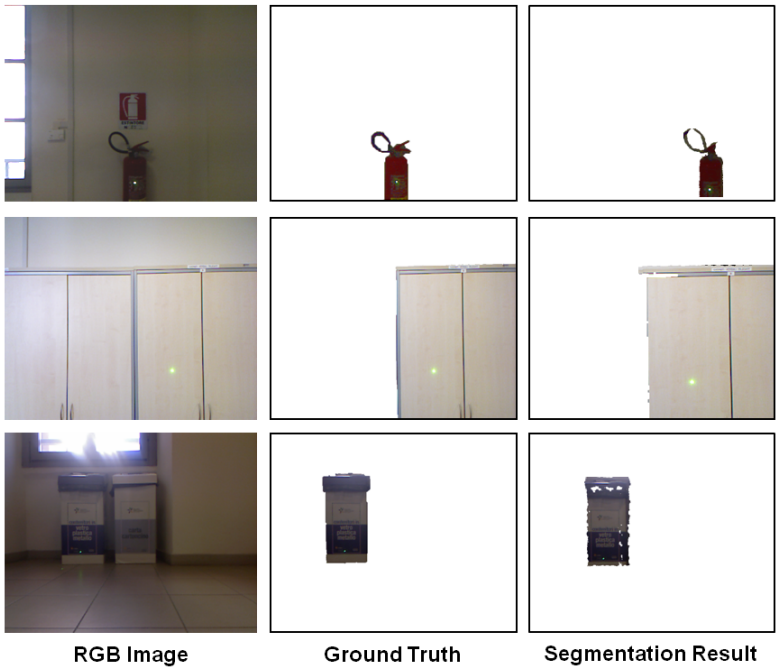

Figure 15: Three classes of objects have been selected for the quantitative evaluation of the object segmentation module: fire extinguisher, cabinet, and recycle bin. In the first column the images of the objects are reported, while the manually obtained ground truth images for the silhouettes of the objects are shown in the second column. The third column contains the results of the segmentation process.

Object Segmentation. A quantitative evaluation for the object segmentation process has been carried out by considering the same objects used for the Semantic Grid Map evaluation. In particular, we have evaluated the accuracy of our approach in segmenting multiple instances of three different classes of objects in our knowledge base (i.e., fire extinguishers, cabinets, and recycle bins) as shown in Figure 15.

Table 3 reports the results of the image segmentation process in terms of Detection Rate (DR) and False Alarm Rate (FAR). DR and FAR are computed as follows:

$$DR = \frac{TP}{TP + FN} \qquad FAR = \frac{FP}{TP + FP}$$

where TP are the true positives, i.e., correctly segmented pixels, FN are the false negatives, i.e., the number of object points detected as background, and FP

Table 3: Mean (μ) and standard deviation (σ) of the error for the Object Segmentation module in terms of Detection Rate (DR) and False Alarm Rate (FAR). We considered, in particular, 3 categories and 15 instances of objects for each category.

Object	$\mu_{DR} \pm \sigma$ **(%)**	$\mu_{FAR} \pm \sigma$ **(%)**
Cabinet	86 ± 8	3 ± 1
Fire Exting.	71 ± 6	18 ± 6
Recycle Bin	83 ± 7	27 ± 16

Table 4: Mean (μ) and standard deviation (σ) of the error in extracting the width (W size) of the tagged objects (15 instances for each category).

Object	$\mu_{error} \pm \sigma$ **(%)**
Cabinet	24 ± 9
Fire Exting.	16 ± 6
Recycle Bin	25 ± 15

are the false positives, i.e., the number of background points detected as object points. Lower values for DR are mainly caused by holes in the depth data, especially along the borders of the objects. Higher values for FAR are mainly caused by a slight misalignment between the RGB image and the depth map provided by the sensor. The highest FAR value are obtained in the case of ***RecycleBins*** since, sometimes, part of a cabinet or a wall alongside the tagged recycle bin is incorrectly segmented as part of it.

Since the final goal of our framework is to acquire knowledge for generating an accurate semantic map, we evaluate also the precision of our segmentation method in extracting the width (W size) of the tagged objects. The results are reported in Table 4. The error e_W is calculated as follows:

$$e_W = \frac{\left| detected_W - GT_W \right|}{GT_W}$$

where $detected_W$ is the width detected by our segmentation algorithm and GT_W is the ground truth width. The analysis of the results suggests that the proposed approach can recover the W size of the tagged objects with an acceptable error e_W. The higher variance of e_W, in the case of ***RecycleBins***, is due to the previously mentioned segmentation error that sometimes occur. It is worth noticing that, when this occurs, the system memorizes the tagged object.

Spatial Reasoning. To demonstrate the improvements that qualitative spatial reasoning can determine in grounding commands, as well as the effectiveness of our approach on a real robot, two different kinds of experiments have been carried out.

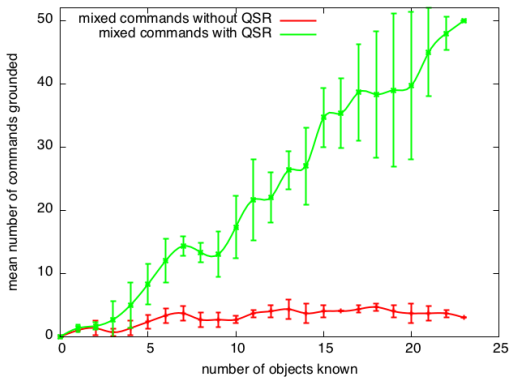

Figure 16: Mean number of grounded commands with respect to the number of objects known in the environment, added in a random order.

Table 5: Number of correctly and wrongly grounded commands with respect to the expectations of the users.

User	**Correct**	**Wrong**
1st	7	3
2nd	8	2
3rd	10	0
4th	6	4
5th	8	2
6th	8	2
7th	10	0
8th	7	3
9th	9	1
10th	8	2
Total	81	19

The purpose of the first experiment was to evaluate the impact of the qualitative spatial reasoner presented in Section 6.3 on an agent whose amount of knowledge continuously grows, as well as the influence of the already available knowledge on such a reasoning. Such an evaluation has been carried out considering the number of unambiguous and ambiguous commands (i.e., commands referring to more than one object with a specific spatial property) grounded by the agent. Indeed, when full knowledge about the environment is available, grounding ambiguous commands would mostly lead to the execution of an action that does not match the user expectation, while all the unambiguous commands are supposed to be correctly grounded. We therefore analyzed first the impact of the presence or absence of the qualitative spatial reasoner (QSR) and then the impact of the amount of knowledge available to the agent. In detail, we first asked to 26 students to provide a set of 3 commands containing spatial relations between objects, by looking at pictures of the test environment. Then, from the 78 acquired commands, we extracted two types of tasks: 28 ambiguous and 50 unambiguous. The commands were grouped following the definition of spatial relations explained in Section 6.3. Specifically, if a command grounded multiple instances

of the same object class, it was considered ambiguous, otherwise unambiguous. By gradually adding knowledge about the objects inside the knowledge base of the agent, we therefore measured how many commands were grounded. In particular, the robot started with an empty knowledge base. At each step, we informed the robot about a new object in the environment. Then, we instructed the robot with all the gathered commands and proceeded to add a new object to its knowledge base. We repeated the experiment for both categories of commands, with or without the qualitative spatial reasoner. Since the curves depend on the order of the objects inserted in the knowledge base, the experiment has been performed five times in order to obtain its average trend (Figure 16). In case the QSR was not present (red curve), only the objects in the environment, whose category has a unique member, were correctly identified. For example, since we had two cabinets in the test environment, there was no way of distinguish them without exploiting spatial relations. By comparing the two curves in the image, it can be noticed that the presence of the QSR does not greatly affect their trend when a little amount of knowledge is available, due to the absence of exploitable spatial relations between objects. On the contrary this is not true when substantial environmental information is accessible. Note that, when complete knowledge about the relevant elements of the environment is known by the robot, the number of grounded commands, as expected, is equal to the number of unambiguous phrases (50 commands) present in the adopted set of commands.

The second experiment aimed at understanding the limitations of the proposed approach. To this end, we measured the agreement between user expectations and the grounding performed by the robot. In particular, we first produced a Semantic Grid Map by driving the robot on a tour of the environment and tagging 23 objects within an office environment, as well as the doors and the functional areas in it. Then, we asked 10 different non-expert users to assign 10 distinct tasks to the robot, additionally asking them to evaluate whether the robot correctly grounded their commands, thus meeting their expectations. The qualitative spatial representation of the robot was briefly explained to the users. Moreover, the commands have been directly acquired through a Graphic User Interface, in order to avoid possible errors due to misunderstandings from the speech recognition system. In detail, the users had the possibility to choose the action to be executed by specifying the located object, the reference object and one of the 10 spatial relations implemented in our reasoner. Table 5 shows that approximately 80% of the given commands have been correctly grounded. The remaining 20% of the wrongly grounded commands were due to two different phenomena: *(i)* the command given was ambiguous, requiring other properties, in addition to direction

and distance, to identify the object. In fact, in the test setting, some commands could be grounded to multiple objects, given the qualitative spatial model of the robot. In such cases, the robot could ground the command to a correct object alternative to the target chosen by the user. *(ii)* the users did not behave coherently during the interaction with the robot, by varying their concept of vicinity or by adopting different reference frames. In fact, in such cases, user expectations disagreed with the qualitative spatial model provided to the robot and explained at the beginning of the experiment.

7.2. Whole-system Evaluation

In this last set of experiments our goal was to evaluate the whole system in a real environment during a typical task executed by the robot. For this reason we deployed our robot in an office environment and we asked both expert and non-expert users to drive the robot around using the vocal interface and to tag the various objects present in the environment. To test the robustness of our system in a noisy environment, we carried out a data collection during a public opening of our department asking 10 visitors, in addition to all of the authors of this paper (for a total of 16 users), to take part in the following experiment. The robot started with no knowledge about the objects enclosed in the environment and each user, after being explained for a minute the commands understood by the robot, had to drive, using the vocal interface, the mobile platform in front of a desired object and teach the robot its position and name. Having memorized different objects, the user had to ask the robot to move in front of them in order to demonstrate that the learning process had been carried out successfully. In this experiment all the users have been able to successfully memorize an object.

After collecting the data, we calculated the distance between the position of the centroid of the learned objects with the one belonging to a ground truth manually created. The result of such a comparison is shown in Table 6. From the table it can be noticed that almost 90% of the objects were placed with an error less than 50 cm. The remaining objects were placed instead at a distance between 50 cm and 1.5 m, due to errors deriving from the object segmentation component, and the Semantic Grid Map Generator. In fact, under particular light conditions (e.g., the camera was facing a bright source of light), the vision component yielded a significant error for the positions of the laser dot. It can also be noticed that the precision does not vary between the expert and non-expert users, thus confirming that this system does not require a specific training to be used. Overall, the evaluation of the performance shows that the system can effectively acquire knowledge about the environment, allowing for the representation in the semantic map of a

Table 6: Result obtained from the test performed on the whole system. The position of the tagged objects is compared with the one obtained from a manually generated ground truth by calculating the distance between the two points.

Thresholds	**Average %**	**Experts %**	**Non-Experts %**
≤ 0.1m	17.5%	20%	16%
≤ 0.2m	43.3%	37%	47%
≤ 0.3m	48.5%	46%	50%
≤ 0.4m	77.0%	72%	80%
≤ 0.5m	86.5%	94%	82%

wide variety of elements. Finally, the results of the final experiment with the users show that the approximations that have been introduced in the representation do not affect the execution of the task, thus providing some evidence of a good balance between abstraction and accuracy reached in our representation.

8. Conclusions

In this paper we have presented a new approach for acquiring and representing the knowledge about arbitrary indoor environments. The knowledge about the environment that is gathered with the help of the user, is turned by the system into a layered representation (i.e. a semantic map), which allows for high level interaction with the user. In the development of our representation, we focused on the acquisition of knowledge about the specific environment, so that the behaviors of the robot over time are supported by grounded facts, rather than by general world knowledge. Additionally, we have shown how human-robot collaboration can play a key role, being the robot instructed by a user through a simple interaction. The resulting semantic map provides a compact and expressive representation, which is used to handle dialogs about the objects and locations in the environment and to ground complex user commands referring to spatial locations. Such maps can be used, for example, to drive the actions of a remote robot through speech, operating in a home or office environment. The implemented system has been tested with four different robots, in different environments and with many users, showing a good performance over extended periods of time. In particular, the system has been experimentally evaluated as a whole, as well as in all its main components. The results of the experiments show that, despite some approximations in the construction of the representation, the knowledge acquisition process is robust and easy to be performed also by non-expert users and on

different robotic platforms.

While the implemented prototype shows the capabilities that can be achieved through the integration of AI techniques (NLP, KR, spatial reasoning, perception, etc.), the implementation can be improved in multiple aspects and several research topics may be addressed. First of all, a better integration with state-of-the-art techniques for object detection and classification, may enable for a more proactive role of the system in building the map, by matching both acquired models of objects and general model of object categories. A better integration of the domain knowledge with external resources, possibly including the web, may also be exploited to support the learning capabilities of the system in grounding the objects of the environment to new concepts and to their linguistic counterparts. Moreover, we plan to improve the whole system to address more cognitive issues related to the construction of the semantic map. Among them, the characterization of changes in the environments by analyzing the evolution of the knowledge base over time and the ability to properly handle multiple instances of objects in the same category. We also plan to allow our robot to learn over time the spatial relations between objects referred by the user to extend its capabilities [38]. Finally, we are considering the extension of the Semantic Grid Map to 3D, which would bring about a new set of spatial relations among the objects. This knowledge acquisition approach could then be extended to support other forms of symbolic representation constructions from low-level data [41].

References

[1] G. Randelli, T. M. Bonanni, L. Iocchi, and D. Nardi, "Knowledge acquisition through humanrobot multimodal interaction," *Intelligent Service Robotics*, 2013.

[2] A. Kristoffersson, S. Coradeschi, and A. Loutfi, "A review of mobile robotic telepresence," *Advances in Human-Computer Interaction*, 2013.

[3] L. C. Aiello, E. Bastianelli, L. Iocchi, D. Nardi, V. Perera, and G. Randelli, "Knowledgeable talking robots," in *Artificial General Intelligence*, 2013.

[4] E. Bastianelli, D. D. Bloisi, R. Capobianco, F. Cossu, G. Gemignani, L. Iocchi, and D. Nardi, "On-line semantic mapping," in *16th International Conference on Advanced Robotics*, 2013.

[5] R. Capobianco, G. Gemignani, D. D. Bloisi, D. Nardi, and L. Iocchi, "Automatic extraction of structural representations of environments," in *Proceedings of the 13th Intelligent Autonomous System conference*, 2014.

[6] B. Kuipers, "The spatial semantic hierarchy," *Artificial Intelligence*, 2000.

[7] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[8] N. Goerke and S. Braun, "Building semantic annotated maps by mobile robots," in *Proceedings of the Conference Towards Autonomous Robotic Systems*, 2009.

[9] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," in *Proceedings of IEEE/RSJ Conference on Robots and Systems*, 2007.

[10] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: extracting the topological structure of indoor environments via place labeling," in *Proceedings of 19th International Joint Conference on Artificial Intelligence*, 2007.

[11] J. Wu, H. I. Christenseny, and J. M. Rehg, "Visual place categorization: problem, dataset, and algorithm," in *Proceedings of IEEE/RSJ Conference on Robots and Systems*, 2009.

[12] O. M. Mozos, H. Mizutani, R. Kurazume, and T. Hasegawa, "Categorization of indoor places using the kinect sensor," *Sensors*, 2012.

[13] M. Gunther, T. Wiemann, S. Albrecht, and J. Hertzberg, "Building semantic object maps from sparse and noisy 3d data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[14] H. Zender, O. Martínez Mozos, P. Jensfelt, G. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, 2008.

[15] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, 2012.

[16] J. Peltason, F. H. K. Siepmann, T. P. Spexard, B. Wrede, M. Hanheide, and E. A. Topp, "Mixed-initiative in human augmented mapping," in *IEEE International Conference on Robotics and Automation*, 2009.

[17] C. Nieto-Granda, J. G. Rogers, A. J. B. Trevor, and H. I. Christensen, "Semantic map partitioning in indoor environments using regional analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[18] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "Learning semantic maps from natural language descriptions," in *Robotics: Science and Systems*, 2013.

[19] G. J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Clarification dialogues in human-augmented mapping," in *Proceedings of the 1st Annual Conference on Human-Robot Interaction*, 2006.

[20] S. Hemachandra, T. Kollar, N. Roy, and S. Teller, "Following and interpreting narrated guided tours," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.

[21] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu, "a point-and-click interface for the real world: laser designation of objects for mobile manipulation," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, 2008.

[22] J. Hertzberg and A. Saffiotti, "Using semantic knowledge in robotics," *Robotics and autonomous systems*, 2008.

[23] R. Capobianco, J. Serafin, J. Dichtl, G. Grisetti, L. Iocchi, and D. Nardi, "A proposal for semantic map representation and evaluation," in *Proceedings of the 7th european conference on mobile robots (ecmr)*, 2015.

[24] J. DeNero and D. Klein, "Teaching introductory artificial intelligence with pac-man," in *Proceedings of the Symposium on Educational Advances in Artificial Intelligence*, 2010.

[25] T. S. Standley, "Finding optimal solutions to cooperative pathfinding problems," in *AAAI*, 2010.

[26] A. Diosi, G. Taylor, and L. Kleeman, "Interactive slam using laser and advanced sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[27] M. Schwarz, J. Stückler, and S. Behnke, "Mobile teleoperation interfaces with adjustable autonomy for personal service robots," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, 2014.

[28] J. A. Leite, *Evolving knowledge bases: specification and semantics*. 2003.

[29] O. Martínez Mozos and W. Burgard, "Supervised learning of topological maps using semantic information extracted from range data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[30] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *Intelligent Transportation Systems Magazine, IEEE*, 2010.

[31] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: a general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.

[32] D. Bloisi and L. Iocchi, "Independent multimodal background subtraction," in *Proceedings of the Third International Conference on Computational Modeling of Objects Presented in Images: Fundamentals, Methods and Applications*, 2012.

[33] T. M. Bonanni, A. Pennisi, D. D. Bloisi, L. Iocchi, and D. Nardi, "Human-robot collaboration for semantic labeling of the environment," in *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration*, 2013.

[34] L. Najman, M. Couprie, and G. Bertrand, "Watersheds, mosaics, and the emergence paradigm," *Discrete Applied Mathematics*, 2005.

[35] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, 2009.

[36] V. Ziparo, L. Iocchi, P. Lima, D. Nardi, and P. Palamara, "Petri net plans - a framework for collaboration and coordination in multi-robot systems," *Autonomous Agents and Multi-Agent Systems*, 2011.

[37] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, 1989.

[38] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *The International Journal of Robotics Research*, 2011.

[39] D. Hernández, "Diagrammtical aspects of qualitative representations of space," *Proceedings AAAI Spring Symposium on Reasoning with Diagrammatic Representations*, 1992.

[40] A. U. Frank, "Qualitative spatial reasoning with cardinal directions," in *Seventh Austrian Conference on Artificial Intelligence*, 1991.

[41] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Constructing symbolic representations for high-level planning," in *Proceedings of the Twenty-Eighth Conference on Artificial Intelligence*, 2014.