Automatic Extraction of Structural Representations of Environments

Roberto Capobianco, Guglielmo Gemignani, Domenico Daniele Bloisi, Daniele Nardi, and Luca Iocchi

Sapienza University of Rome
`<lastname>@dis.uniroma1.it`

Abstract. Robots need a suitable representation of the surrounding world to operate in a structured but dynamic environment. State-of-the-art approaches usually rely on a combination of metric and topological maps and require an expert to provide the knowledge to the robot in a suitable format. Therefore, additional symbolic knowledge cannot be easily added to the representation in an incremental manner. This work deals with the problem of effectively binding together the high-level semantic information with the low-level knowledge represented in the metric map by introducing an intermediate grid based representation. In order to demonstrate its effectiveness, the proposed approach has been experimentally validated on different kinds of environments.

1 Introduction

The requirement that robots operate in structured but dynamic environments, without a priori knowledge, is growing. This is true not only for industrial robots, but also for general purpose or domestic ones. Those, in fact, will soon become available to a wide community of heterogeneous customers having high expectations. When a robot has to intelligently act in an indoor location, maps representing knowledge about the environment are usually provided beforehand. In particular, metric maps and topological maps are used to this purpose. The former is a geometric representation (i.e., an occupancy grid) of the surrounding world, containing spatial information: state-of-the-art techniques rely on such maps for autonomous navigation. The latter, is a graph based representation, where nodes are associated with physical locations (i.e., areas of interest within the environment) and edges are connections between locations. Topological maps are mainly used to accomplish task-oriented goals, as well as to execute complex robot tasks, since they can provide a link between symbols and locations.

While there are tools and methodologies for building metric maps, and, to some extent, the topological map, all the knowledge about the environment must 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 map provide a good representation for adding semantic knowledge. The occupancy grid cannot be directly annotated with symbols, since it has a fine discretization, which is good for a rich description of structural details, but not for the acquisition of high-level information. The topological map 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, it derives that an intermediate abstraction layer, strictly linked to the metric map is needed for both acquiring semantic knowledge and enabling the robot to build the topological map on top of the real 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.

This work focuses on the introduction of a 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. 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 build a 2D grid representation of the environment, since grids are useful to represent areas with equal properties and features with an intermediate level of granularity with respect to metric maps.

The remainder of the paper is structured as follows. Related work is discussed in Section 2. The adopted representation and the algorithms used for its extraction are detailed in Section 3, while the experimental evaluation of the approach is described in Section 4. Conclusions are drawn in Section 5.

2 Related Work

There exists a large literature related to the problem of the symbolic and abstract representation of knowledge about the environment. A first category includes fully automated approaches aiming at constructing environmental maps, by classifying functional areas. To this end, many methods focus on the extraction of room attributes. Fabrizi and Saffiotti [1] and Buschka and Saffiotti [2] present a local technique that uses range information to detect room-like data during navigation, through a virtual sensor. Starting from the idea of dividing occupancy grid into regions separated by local narrowings, as proposed by Thrun [3], they work on an algorithm for partitioning the space in an incremental way. In particular, computing a certain number of parameters for each region, they produce a local-topological map. Similarly, Anguelov *et al.* [4] focus on doorways detection and object-based modeling in order to define a probabilistic model of corridors containing doors and walls. They parametrize each object by its shape, color, and motion model in order to obtain the distribution over possible observations of the robot. Galindo *et al.* [5] represent environmental knowledge by augmenting a topological map with semantic knowledge, which

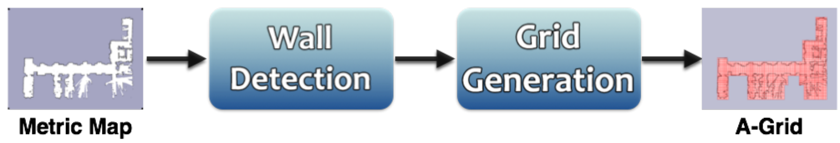

Fig. 1. Block scheme of the processing chain from the Metric Map to the A-Grid.

is extracted with fuzzy morphological operators, while Choi *et al.* [6] produce a discretization of the environment based on quadtrees, which is then used for producing a topological graph and for performing localization on the extracted topological model.

A second category of techniques make use of classification and clustering methods for the automatic segmentation and labeling of metric maps. Nüchter *et al.* [7] extract environmental knowledge by labeling 3D points through the gradient difference between neighboring points, which are then classified as floor-points, object-points or ceiling-points. Mozos *et al.* [8][9], instead, extract simple geometric real-valued features from scans, and classify them through an Adaboost multi-classifier obtained by arranging several weak binary classifiers in a decision list. A similar approach is proposed by Goerke and Braun [10], in which a learning classifier is used to build semantic annotated maps from laser range measurements. Other approaches have been proposed, based, for example, on spectral clustering [11] or Voronoi random fields [12].

The last and more recent part of the research on automated environment representation proposes, instead, techniques for object recognition and place categorization, based on visual features [13], or a combination of visual and range information provided by an RGB-D camera [14]. For example, Pangercic *et al.* [15] investigate the representation and acquisition of Semantic Objects Maps (SOMs) in kitchen environments, by using RGB-D data and active manipulation actions such as opening drawers and doors. However, a suitable structured representation of the acquired knowledge is missing.

As a difference with respect to the above discussed methods, our approach stems from the need for an efficient and general environmental knowledge representation, able to easily and effectively bind together high-level semantic information with low-level knowledge represented in the metric map. We show how, by taking as input a metric map generated through any SLAM method, we can create a general yet effective representation of any environmental knowledge. Moreover, we show how most of the techniques proposed in the literature, either automatic or user-guided, can be easily integrated and exploited to enhance the overall quality of the semantic map.

3 Extracting the Structural Representation

The 2D structure of the environment is captured through an abstract grid, called A-Grid (see Fig. 1). In detail, we first build such a grid, and then we add knowledge on top of it. In order to build the A-Grid, we extract the lines corresponding to the walls of the building (that are the basic elements to form the grid itself).

The A-Grid is a discretization of the metric map having two important properties: 1) It is non-uniform in the interface towards the metric map, since the cell size varies as a function of the detected walls; 2) It provides a uniform interface, being accessible as a simple matrix. Using such a representation structure, we finally divide the building in functional areas and doors, thus providing the information needed for the creation of a topological map, where symbols can be eventually associated to nodes.

3.1 A-Grid Construction

The purpose of the desired representation is to have a tool for creating semantic maps on top of the low level information about the environment and not the pure improvement in terms of computation efficiency. The existing discretization methods already available as KD-Trees or quadtrees, however, would neither produce an easily accessible representation (e.g., a matrix), nor correctly identify the global structure of the environment. The A-Grid represents, instead, a discretization of the environment which allows to associate an area in the real world to a label. The algorithm for constructing such an abstraction takes as input the occupancy grid which is commonly used for robot navigation (metric map), outputting a layer which reflects the structure of the environment. This step can be seen as a rasterization process, since each cell of the grid represents a portion of a physical area of the environment and, from a functional point of view, it is an abstraction of locations that are not distinguishable.

Wall Detection Since the approach is focused on the knowledge acquisition in indoor environments, the production of the grid relies on the extraction of the lines corresponding to the walls. Those lines are highly informative, being commonly used for the separation of different functional areas in offices as well as in domestic environments, and usually describing a top-view, regular structure of the building. Indeed, walls in maps are usually represented as straight horizontal or vertical lines. However, even considering those regular patterns, the lines corresponding to the walls cannot be extrapolated by directly looking at the metric map. When considering the occupancy grid the following aspects may be taken into account: 1) Noise, due to positions occupied either by humans moving in the environment during the map acquisition or by very small objects, such as legs of tables or chairs; 2) Big objects (e.g., a closet), at the border or within the building, which generate occupied locations in the metric map (artificial walls); 3) Incomplete laser-scans, which compromise the regular structure of the building (e.g., caused by closed doors).

In order to deal with noise, a pre-processing step is required at the metric map level. Specifically, we applied the Canny Edge Detector, determining the threshold values through a multi-map tuning procedure and considering a trade-off between overall performance and noise reduction. In detail, the threshold values have been selected to be 1 for the lower one and 30 for the upper one. The result is a reduction of noise which, however, allows to maintain relevant lines from the map (Fig. 2a). Although empirically defined, those thresholds turned out to be suitable in all our experiments, always detecting the edges of the map.

Algorithm 1: WallDetection

Data: $\mathcal{W}$: wall segments set, $\mathcal{D}$: detected segments set

Input: *map*: pre-processed metric map, *max_threshold*, *min_threshold*, *min_line_distance*, *max_angle_distance*

Output: $\mathcal{W}$

1 **Initialize:** $\mathcal{W} \leftarrow \varnothing$; $\mathcal{D} \leftarrow \varnothing$; $threshold \leftarrow max\_threshold$;

2 **while** $threshold > min\_threshold$ **do**

3     $distance\_resolution \leftarrow 2$;

4     $angle\_resolution \leftarrow \pi/2$;

5     $min\_distance \leftarrow \max(\frac{max\_threshold}{threshold}, min\_line\_distance)$ `// compute adaptively a min distance from the already acquired walls`

6     $\mathcal{D} \leftarrow \mathcal{D} \cup \mathrm{HoughTransform}(map,\ distance\_resolution,\ angle\_resolution,\ threshold)$;

7   **for** $d \in \mathcal{D}$ **do**

8         $isFar \leftarrow$ true;

9   **for** $w \in \mathcal{W}$ **do**

10   **if** $|\mathrm{distance}(d) - \mathrm{distance}(w)| < min\_distance$ and $|\mathrm{angle}(d) \pmod{\pi} - \mathrm{angle}(w) \pmod{\pi}| < max\_angle\_distance$ **then**

11                 $isFar \leftarrow$ false;

12   break;

13   **if** $isFar =$ true **then**

14             $\mathcal{W} \leftarrow \mathcal{W} \cup d$;

15     $threshold \leftarrow threshold$ - 1;

Since knowledge about objects cannot be handled within this abstraction layer, we accept artificial walls in the map and postpone the detection of the corresponding objects to a subsequent processing layer. In a first analysis, the Wall Extraction algorithm (see Alg. 1) considers only long and continuous lines. These are detected through the use of the *Hough Transform* (with high accumulator threshold), which returns the Polar coordinates (ρ, θ) of the lines that get many votes. Furthermore, due to the regular structure which usually characterizes our buildings, only horizontal and vertical lines are considered as candidates in this process. In this way the approach can deal with incomplete laser scans, since their image in the metric map is usually highly irregular. This step is shown in the lines 4 and 6 of the pseudo-code. The lines that form the grid are selected by gradually reducing the threshold value and, consequently, the length constraints. The lines are accepted only if their distance from the previously acquired lines is greater than a value (line 10 of the Algorithm), which is determined adaptively as a function of the accumulator threshold (line 5). This allows to discard a large portion of false-positives corresponding to irregularities in the laser scans and to noise (usually very small in length) that are evaluated in the last cycles of execution of the algorithm, when almost all the walls have been already detected. An example of the result obtained from the wall detection algorithm is provided in Fig. 2c, which can be compared to Fig. 2b where the Hough Transform has been used without any additional selection criteria.

Grid Generation Grids are commonly used, both in video games, robotics and artificial intelligence, for representing maps and reasoning on them. The generation of the grid is the main procedure for creating a qualitative representation of the environment, whose importance is linked to robot's high-level behaviors. Indeed,

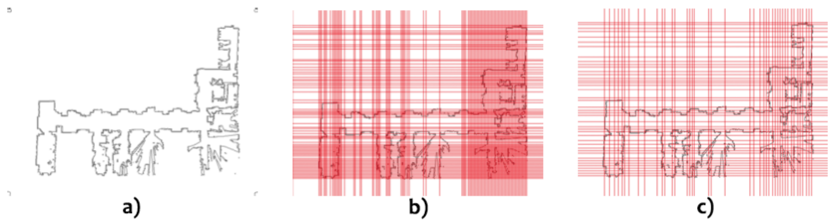

Fig. 2. a) Edges of the metric map. b) Result of the Hough Transform performed on the metric map. c) Result of the wall detection algorithm.

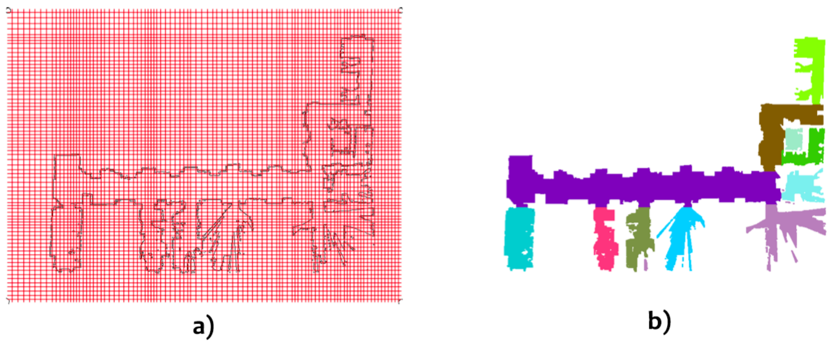

Fig. 3. a) A-Grid example, obtained through the algorithms described in Section 3.1; b) Area segmentation obtained through manual tagging.

an intelligent agent does not necessarily need a uniform grid representation, but a consistent one, in which cells may differ in size in order to represent portion of the environment which will be similar from a functional point of view. In order to achieve this goal, two steps are required. First, extending the segments generated through the wall detection to the whole image (Fig. 2), the minimum horizontal and vertical distances are computed ($x_{\min}, y_{\min}$). Then, each other pair of parallel lines is considered, and a new one is inserted between them only if their distance is at least twice $x_{\min}$ or $y_{\min}$, depending on their angle. This adaptive procedure (Alg. 2), creates a grid whose cells have a size between $x_{\min} \cdot y_{\min}$ and $2x_{\min} \cdot 2y_{\min}$. In particular, from line 2 to line 14 of the reported pseudo-code, the minimum distance between wall segments is computed (line 9). Those wall segments are also used in order to initialize the grid (line 14) and a processing queue (line 13). Then, a pair of parallel lines $< first\_line, second\_line >$ is extracted from the queue itself, until it is not empty (lines 15-16). If their distance is at least twice $x_{\min}$ or $y_{\min}$ (line 22), a new line is inserted between them in the grid, with the same angle, and the two pairs $< first\_line, new\_line >$, $< new\_line, first\_line >$ are added to the queue in order to recursively produce the final grid.

Algorithm 2: GridConstruction

```
Data: 𝒲: wall segments set, 𝒢: grid lines set, Q: processing queue
Input: 𝒲, x_min, y_min, map_height, map_width, max_angle_distance
Output: 𝒢

Initialize: 𝒢 ← ∅; Q ← ∅;
for w_1 ∈ 𝒲 do
    segment_dist ← max(map_height, map_width);
    first_line ← ∅;
    second_line ← ∅;
    for w_2 ∈ 𝒲 do
        if |angle(w_1) (mod π) − angle(w_2) (mod π)| < max_angle_distance then
            first_line ← w_1;
            if distance(w_1) − distance(w_2) > 0 and distance(w_1) − distance(w_2) <
            segment_dist then
                second_line ← w_2;
                segment_dist ← distance(w_1) − distance(w_2);
    if segment_dist ≠ max(map_height, map_width) then
        Q ← Q ∪ < first_line, second_line >;
    𝒢 ← 𝒢 ∪ w_1;
while Q ≠ ∅ do
    < first_line, second_line >← return_and_remove_last_element(Q);
    // for vertical lines x_min, for horizontal ones y_min
    if areVertical(first_line, second_line) then
        max_dist ← x_min;
    else
        max_dist ← y_min;
    value ← ⌊ |distance(first_line) − distance(second_line)| / max_dist ⌋;
    if value ⩾ 2 then
        new_distance ← (distance(first_line) + distance(second_line)) / 2;
        new_line ← line(new_distance, angle(first_line));
        𝒢 ← 𝒢 ∪ new_line;
        Q ← Q ∪ < first_line, new_line >;
        Q ← Q ∪ < new_line, second_line >;
```

The obtained discretization (Fig. 3a) is associated with a matrix data structure, providing a qualitative representation consistent with the metric layer and which allows for a simple integration of symbols and semantics. On the one hand, the grid is built on the basis of the walls and, therefore, it is consistent with the structure of the environment; on the other hand, the granularity is high enough to endow the robot with a structure for embedding symbolic knowledge about the building, and the interface provided by the A-Grid is uniformly accessible as a matrix. Moreover, it is always possible to associate a cell of the matrix to a position in the environment, and viceversa. In fact, each location of the building can be linked to a cell in the A-Grid. On the contrary, if a single location is needed starting from the cell indexes, it is convenient to consider its center as a reference. Specific constraints on the final position may be easily accommodated. In practice, the reference to the center of the cell is acceptable both from a qualitative representation and task execution point of view.

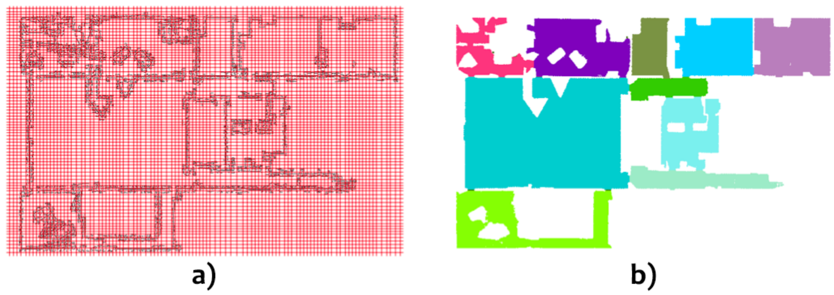

Fig. 4. a) A-Grid computed on a map from the Radish Data Set; b) Area segmentation obtained through manual tagging.

3.2 Area Segmentation

The area segmentation process is used for labeling each location inside the environment, by associating it to a node of the topological map. In particular, a label is assigned to all the cells of the grid within the building, while the elements of the matrix without an associated tag are considered to be out of any functional area. The segmentation of the map in functional areas is performed on the basis of the contours extracted from the metric map, together with a series of tags which can be acquired through several modalities. In detail, two types of tags are available: door-tags or area-tags. Three different approaches are used to perform this activity: manual tagging, human-robot interaction (HRI) based tagging and automatic tagging through simple heuristics. As in previous approaches [1][2], these heuristics find the narrow passages in the metric map. However, we also perform a selection among them based on their regularity in the structure, as well as their proximity and the number of points around them. Starting from those, each room is considered to be separated from the others by a door. Therefore, by evaluating all the acquired tags, a set of segments of appropriate size, corresponding to the doors, is added to the detected contours of the metric map, closing each functional area. If no human collaboration is available, area-tags are obtained from the centroid of each obtained portion of the environment. Then, similarly to Buschka and Saffiotti[2], we use each area-tag as a seed for a Watershed-based region growing algorithm. In particular, the algorithm, which considers the obtained map as a topographic relief, is the OpenCV implementation of the procedure described by Meyer [16]. The opening morphological operator (which is an erosion followed by dilation) is then applied to the obtained segmented map (Fig. 3b). Finally, each area can be extracted by applying a color based segmentation starting from the tag of the desired functional space.

4 Experimental Validation

An experimental validation has been carried out in order to show the effectiveness and the robustness of the proposed method. Several experiments have been

Table 1. Comparison between the pixels of each processed Metric Map and the cells of the corresponding A-Grid.

Map	Pixels	Cells
BelgioiosoCastle	768 792	11 600
dis-B1	1 080 700	10 290
dis-B1-part	501 840	7372
dis-Basement	992 785	13 455
FortAPHill	534 520	7878
Freiburg	335 248	4794
HospitalPart	30 000	285
Intel	336 399	4473
scheggia	92 984	1116
UBremen	831 264	10 962

therefore conducted by using the publicly available Radish Data Set[1]. Specifically, six 2D metric maps[2], obtained from different SLAM methods, have been processed in order to test our A-Grid representation on a number of different occupancy grids (Fig.4). In addition, four maps generated by our robots have been processed, obtaining a total of ten completely different environments processed by the algorithm. To give an intuition of the reduction in terms of the size of the representation, in Table 1 we show the relation between the pixels of the Metric Maps and the cells of the corresponding A-Grid. They differ by roughly two orders of magnitude, depending on the complexity of the map. This underlines that the representation is compact, easily enhanceable with symbolic information and suitable for an optimized access and processing. Using a representation obtained from the application of KD-Trees or quadtrees would probably lead to further improvements in terms of performance. However, as previously stated, KD-Trees or quadtrees, however, would neither produce an easily accessible representation as the one produced by the A-Grid, nor correctly identify the global structure of the environment.

Since only the environments represented in the maps produced by our robots were accessible, only such maps have been tagged through a specific developed human-robot interaction (HRI). In detail, we incrementally augmented these maps by adding semantic knowledge about areas, doors, and objects through the help of the user, according to the approach known as Symbiotic Autonomy [17]. The system allowed to consistently build and update semantic maps of the environments, over a large set of tests, that included many rooms and objects in different kinds of functional spaces. For a detailed overview of such an approach see [18], where a comprehensive description of the full process, with qualitative and quantitative analysis, is provided.

[1] http://radish.sourceforge.net/index.php

[2] The algorithm was applied to the following maps: *albert-b-laser*, *ap_hill_07b*, *ubremen-cartesium*, *intel_lab*, *belgioioso* and a portion of *hospital_floorplan_fort_sam_houston*

Qualitative results obtained by applying our algorithms are shown in Fig. 5 and Fig. 6. In detail, these images depict the integration of the information about areas within the A-Grid structure, from the uniform, matrix-like interface provided to the topological map: each cell in the grid is an element of a matrix, annotated with a set of symbols, and it corresponds to a particular space in the physical world. These representations allow to easily implement high-level reasoning, without loosing the low-level information, being the actual correspondence between cells and real coordinates kept by the system. For example the procedure for area segmentation, when directly performed on the A-Grid, significantly reduces the computation effort and improves the accuracy of the final result. Similarly, using the A-Grid with a simple $A*$ procedure finding the path between two locations in the environment is a fast process, while a search on the full metric map would be expensive, if not unfeasible.

Fig. 6 shows the representation obtained on two HRI-tagged maps directly produced by our robots. As it can be observed from the figure, our representation allows to correctly integrate metric and symbolic information about the environment: each functional area is graphically represented with a different color, doors are shown in blue and objects in red. Note that using HRI tagging enables a higher precision both in performing the area segmentation and in semantically characterizing the environments. Since in this case the representation includes the objects, their spatial positioning on the A-Grid is easily identifiable. Moreover, the A-Grid allows for a suitable definition of a topological graph in order to implement a planning system for the robot on two levels [18].

In general, it can be observed that a very good correspondence can be obtained between the real environment and the structured information within the A-Grid. Moreover, the A-Grid, once produced through the process illustrated in the paper, provides an interface to the environment representation that is easily accessible and independent of the methods chosen to built it. Finally, it is worth noticing that, even if the A-Grid has been developed for ordinary (regular) buildings, it can be successfully applied to environments with irregular edges. Visual evidence of this, as well as of human-driven knowledge acquisition can be found at *www.dis.uniroma1.it/∼gemignani/Articles/ias13.html*, together with all the maps processed by our automatic segmentation algorithm.

5 Conclusions

In this paper we have introduced the A-Grid, an intermediate grid based representation that can easily and effectively bind together high-level semantic information and low-level knowledge represented in the metric map. By taking as input a metric map (e.g., produced by a SLAM algorithm) and a set of semantic tags (acquired automatically or through the interaction with the user), the proposed approach builds such a representation. Several tests have been carried out on multiple maps generated by different robots, as well as on maps from the publicly available Radish Data Set. Finally, the algorithms have been embedded into a complex system [18] that can consistently build and update semantic maps, over a large test set, including a large number of rooms and objects with different functional use.

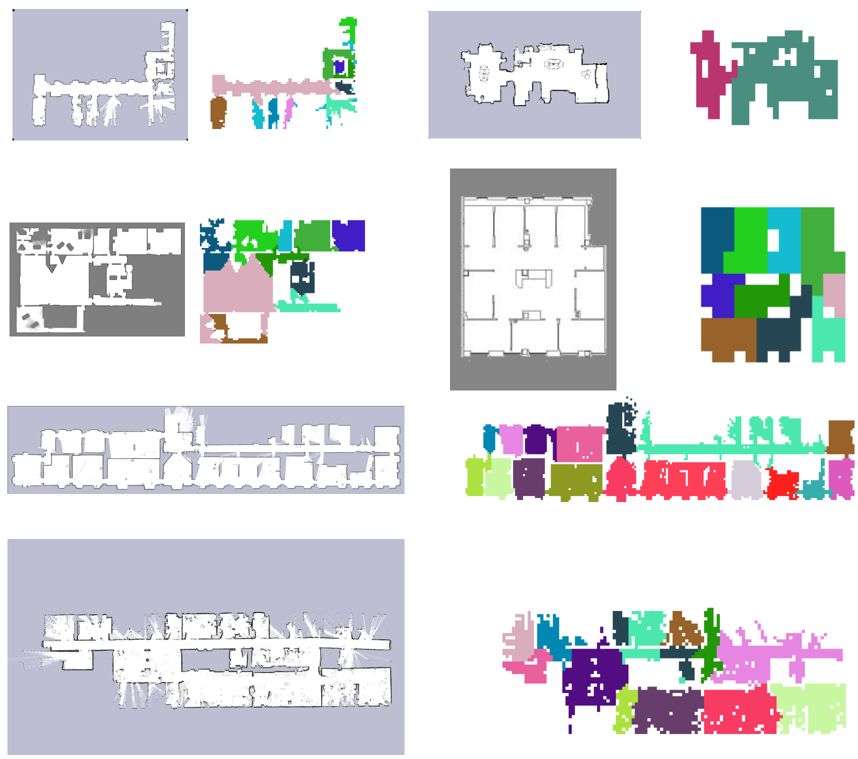

Fig. 5. Structural representation for a set of maps obtained through automatic door detection and area segmentation. The figure shows the integration of the information about areas within the A-Grid structure, from its uniform, matrix-like interface: each cell in the grid is an element of a matrix, with a set of symbols associated, and it corresponds to a particular space in the physical world.

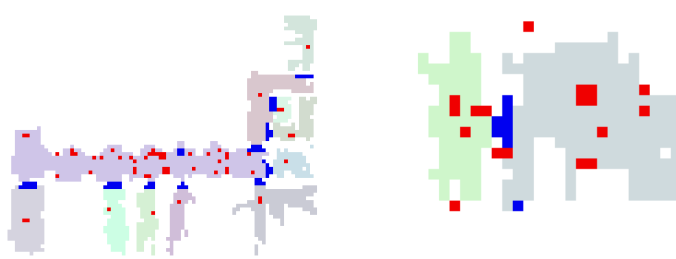

Fig. 6. Integration of the information about areas, doors and objects obtained through HRI within the A-Grid structure, from its uniform, matrix-like interface.

References

1. Fabrizi, E., Saffiotti, A.: Augmenting topology-based maps with geometric information. Robotics and Autonomous Systems **40**(2) (2002) 91–97
2. Buschka, P., Saffiotti, A.: A virtual sensor for room detection. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2002) 637–642
3. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence **99**(1) (1998) 21–71
4. Anguelov, D., Koller, D., Parker, E., Thrun, S.: Detecting and modeling doors with mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (2004) 3777–3784
5. Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernández-Madrigal, J., González, J.: Multi-hierarchical semantic maps for mobile robotics. In: Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (2005) 3492–3497
6. Choi, J., Choi, M., Nam, S.Y., Chung, W.K.: Autonomous topological modeling of a home environment and topological localization using a sonar grid map. Autonomous Robots **30**(4) (2011) 351–368
7. Nüchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., Surmann, H.: 3D Mapping with Semantic Knowledge. In: RoboCup 2005: Robot Soccer World Cup IX. (2005) 335–346
8. Mozos, O.M., Stachniss, C., Burgard, W.: Supervised learning of places from range data using adaboost. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE (2005) 1730–1735
9. Martinez Mozos, O., Triebel, R., Jensfelt, P., Rottmann, A., Burgard, W.: Supervised semantic labeling of places using information extracted from sensor data. Robotics and Autonomous Systems **55**(5) (2007) 391–402
10. Goerke, N., Braun, S.: Building semantic annotated maps by mobile robots. In: Proceedings of the Conference Towards Autonomous Robotic Systems. (2009) 149–156
11. Brunskill, E., Kollar, T., Roy, N.: Topological mapping using spectral clustering and classification. In: Proceedings of IEEE/RSJ Conference on Robots and Systems (IROS). (2007) 3491–3496
12. Friedman, S., Pasula, H., Fox, D.: Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In: Proceedings of 19th International Joint Conference on Artificial Intelligence (IJCAI). (2007) 2109–2114
13. Wu, J., Christenseny, H.I., Rehg, J.M.: Visual place categorization: Problem, dataset, and algorithm. In: Proceedings of IEEE/RSJ Conference on Robots and Systems (IROS). (2009) 4763–4770
14. Mozos, O.M., Mizutani, H., Kurazume, R., Hasegawa, T.: Categorization of indoor places using the kinect sensor. Sensors **12**(5) (2012) 6695–6711
15. Pangercic, D., Pitzer, B., Tenorth, M., Beetz, M.: Semantic object maps for robotic housework-representation, acquisition and use. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE (2012) 4644–4651
16. Meyer, F.: Color image segmentation. In: Image Processing and its Applications, 1992., International Conference on, IET (1992) 303–306
17. Rosenthal, S., Biswas, J., Veloso, M.: An effective personal mobile robot agent through symbiotic human-robot interaction. In: Proc. of 9th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). (2010)
18. Bastianelli, E., Bloisi, D., Capobianco, R., Cossu, F., Gemignani, G., Iocchi, L., Nardi, D.: On-line semantic mapping. In: Proceedings of the 16th International Conference on Advanced Robotics (ICAR). (2013)