

# Towards Extraction of Topological Maps from 2D and 3D Occupancy Grids

Filipe Neves Santos, António Paulo Moreira, and Paulo Cerqueira Costa

INESC TEC (formerly INESC Porto) and Faculty of Engineering, University of Porto,  
Campus da FEUP, Rua Dr. Roberto Frias, 378, Porto, Portugal  
{fbnsantos, amoreira, paco}@fe.up.pt

**Abstract.** Cooperation with humans is a requirement for the next generation of robots so it is necessary to model how robots can sense, know, share and acquire knowledge from human interaction. Instead of traditional SLAM (Simultaneous Localization and Mapping) methods, which do not interpret sensor information other than at the geometric level, these capabilities require an environment map representation similar to the human representation. Topological maps are one option to translate these geometric maps into a more abstract representation of the world and to make the robot knowledge closer to the human perception. In this paper is presented a novel approach to translate 3D grid map into a topological map. This approach was optimized to obtain similar results to those obtained when the task is performed by a human. Also, a novel feature of this approach is the augmentation of topological map with features such as walls and doors.

**Keywords:** SLAM, Semantic, topological maps.

## 1 Introduction

The well known Simultaneous Localization And Mapping (SLAM) problem for mobile robots was widely explored by the robotics community in the last two decades. Several approaches were proposed and explored. Nowadays there are valuable solutions based on these approaches that makes possible to the robots operate in crowded places and in buildings without special requirements. Rhino by [1], Robox by [2], Minerva by [3] and RoboVigil by [4] are some of those robots that can create a map and use it to localize and navigate through the environment. These robots rely on 2D or 3D accurate metric representations of the environment, derived from SLAM techniques.

The concept behind majority of SLAM approaches is collect natural or artificial features in the environment through the observations of the robot sensors. These features are tagged with a Cartesian position and its uncertainty. Recurring to special robot trajectories, Bayes theorem and Bayesian derivative techniques is possible to relate the position of each feature to the position of others features and reduce the uncertainty in that relative relative position. These features are useful to correct the estimated robot localization and also to build

the grid map map. In [5], [6], [7] and [8] is possible to find different approaches to SLAM using the same concept. Correct feature extraction and association is the most hard issue.

Using these SLAM approaches is possible for the robot gets autonomously an accurate 2D or 3D grid map of the environment. However considering true that robots are moving from high tech factories to our homes, offices, public spaces and small factories, this will require that some of these robots should work and cooperate alongside us. In this scenario the information about cell occupation in the grid map is not enough to associate a human word to a place name or even to an object. Also this knowledge, in the form of a grid map, is not easily communicable. So taking this in mind, we have proposed an extension to SLAM called HySeLAM - HYbrid SEmantic Localization And Mapping system, described in [9]. This extension creates two new layers over this grid map, the topological layer and the semantic layer. The topological layer describes the places and places connectivity, detailed in section 3. The semantic layer relates Object/place in the 3D space. In the semantic layer, each object is an instance of a generic object described in Object dictionary.

The focus of this paper is to show an approach that translates a grid map obtained from SLAM into a topological map. Section 2 presents a global overview to the problem and different approaches to translate a grid map into a topological map. Section 3 presents the formal definition for our augmented topological map. Section 4 presents our approach to translate a gridmap into a topological map. Section 5 presents the obtained results. Section 6 presents the paper conclusions.

## 2 Build a Topological Map from a Grid Map

A grid map is a metric map ([10],[11]) that makes discretization of the environment into 2D or 3D cells. The grid map consists of empty cells,  $m(x, y) = 0$ , which represent free space and occupied cells,  $m(x, y) = 1$ , where obstacle exists. 2D grid map are the most common but 3D grid map is growing in popularity due low cost of 3D acquisition system, as low cost RGB-D cameras and 3D laser range finder scan solutions. These 3D grid map are extremely expensive in terms of memory size so often these grid map are store in a form of octomaps.

A topological map ([12], [13]) describes the world using vertices and edges. The vertices represent the significant spaces in the environment, and the edges represent the connection between the different spaces. In this work, the edge should represent real or virtual door and the vertices should represent a room or a distinctive place.

Several researchers have tried to extract topological models from grid maps to solve global path planning and to help perform navigation and localization in local areas, [14] [15] [16] [17].

One way to get a topological map is extracting the Voronoi diagram from the grid map. A Voronoi diagram is a way of dividing space into a number of regions which are delimited by the Voronoi segments. The segments of the Voronoi diagram are all the points in the plane that are equidistant to the two

nearest sites. The Voronoi vertices (vertices) are the points equidistant to three (or more) sites. In [18] we found an approach to extract the Voronoi diagram from a dynamic grid map, this approach was been optimized for online Voronoi diagram extraction.

Thrun in [14] extracts from the grid map the Voronoi diagram and critical points. Critical points are used to delimit regions and the Voronoi edges are used to relate the regions connectivity. These regions and edges are used to build the topological map.

In [19] and [20], graph partitioning methods are used to divide a grid map into several vertices. Buschka and Saffiotti in [21], Room-like spaces are extracted in grid maps by using fuzzy morphological opening and watershed segmentation. Even though those methods show successful topology extraction from grid maps, they are not easy to apply directly in home environments because they are suitable for corridor environments or considers only narrow passages to extract topological model.

In [15], Virtual door is defined as the candidates of real door, and the virtual doors are detected as edges of the topological map by extracting corner features from the occupancy grid-map; using this method, an initial topological map is generated, which consists of vertices and edges. The final topological map is generated using a genetic algorithm to merge the vertices and reduce the edges. As a result, the generated topological map consists of vertices divided by virtual doors and edges located in real doors. The proposed methods provide a topological map for the user interaction and the cleaning robot, and the topological map can be used to plan more efficient motion including room-to-room path planning and covering each room.

In [22] build the topological modeling using only low-cost sonar sensors. The proposed method constructs a topological model using sonar grid map by extracting subregions incrementally. A confidence for each occupied grid is evaluated to obtain reliable regions in a local grid map, and a convexity measure is used to extract subregions automatically.

### 3 Topological Map in HySeLAM

The augmented topological map, a layer of HySeLAM [9], defines a place by its delimitations (real or virtual walls), location, visual signature and by human words. The edges store the connectivity between places and are labeled with virtual or real doors. These doors are defined by their size, location and human words. This map will be part of the HySeLAM, managed by the topological engine. This is a map layer between grid map and semantic map. The semantic engine will relate each place from this map to a semantic label (human words). The topological engine will be able to buildup the map using the grid map and the descriptions received from other entities (other robots/people).

Therefore, the topological map  $\mathcal{M}_t$  is defined by an attributed graph :

$$\mathcal{M}_t = (\mathcal{P}, \mathcal{C}) \quad (1)$$

where:  $\mathcal{P}$  is the set of vertices (places) and  $\mathcal{C}$  the edges ( $\mathcal{C} \subseteq \mathcal{P} \times \mathcal{P}$ ). The places are augmented with five attributes: semantic words, geometric description, visual appearances, area and central position. A place is defined as:

$$p_i = \{\mathcal{SP}, \mathcal{W}, \mathcal{V}, A_r, X_c\} \quad (2)$$

Where:  $\mathcal{SP}$  is a semantic set of words labeling the place,  $\mathcal{W}$  defines the real and/or virtual delimitation of the place with a set of wall's ( $\mathcal{W} = \{w_0, w_1, \dots, w_{nw}\}$ );  $\mathcal{V}$  is the visual appearance described by a set of local panoramic views ( $\mathcal{V} = \{v_0, v_1, \dots, v_{nv}\}$ );  $A_r$  is a real number which defines the place area; and  $X_c$  is the centric position ( $X = [x, y, z]$ ) of the place.

The parameter wall  $w_i$  is defined by:

$$w_i = (X, \vec{V}, L, S, e) \quad (3)$$

Where:  $X = ([x, y, z], \Sigma_X)$  defines the position of the center of mass of the wall and the uncertainty associated to this observed position, this position is related to the origin of SLAM referential frame;  $\vec{V} = ([vx, vy, vz], \Sigma_V)$  contains the normal vector which defines the wall direction and the uncertainty associated to that vector;  $L = ([vl, vh], \Sigma_V)$  defines the length (height and width) of the wall and the uncertainty associated to the length;  $S = [sl, sh]$  defines the shape curvature of the wall;  $e$  defines the existence of the plan, (0 when the wall is virtual, and 1 when the wall is a brick wall).

The visual appearance parameter  $v_i$  is defined by:

$$v_i = (J, X, La, t) \quad (4)$$

Where:  $J$  is a matrix containing the image,  $X = ([x, y, z], \Sigma_X)$  is the center of image acquisition and the associated uncertainty described by a covariance matrix ( $\Sigma_X$ );  $La$  is the angular aperture ( $L = [Lx, Ly]$ ); and  $t$  is time of acquisition.

The edges ( $\mathcal{C} \subseteq \mathcal{P} \times \mathcal{P}$ ) are labeled with two attributes: semantic words and doorway set, as follow:

$$\{\mathcal{SD}, \mathcal{D}\} \quad (5)$$

Where:  $\mathcal{SD}$  is a semantic set of words labeling the edge, and  $\mathcal{D}$  is the doorway definition.

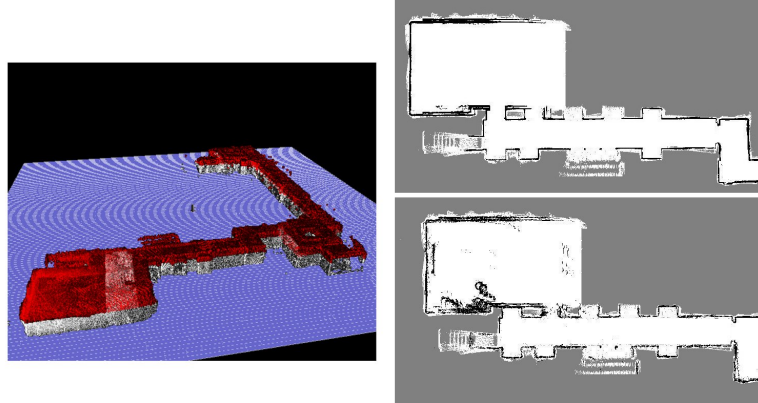
The parameter doorway  $\mathcal{D}$  is defined by:

$$\mathcal{D} = (X, Ld, e, o, I) \quad (6)$$

Where:  $X = ([x, y, z], \Sigma_X)$  defines the position of the center of mass of the door and the uncertainty associated to this observed position, this position is related to the origin of SLAM referential frame ( $\Sigma_X$ );  $Ld$  defines the length of the plan ( $L = [vl, vh]$ );  $e$  defines the existence of the door (0 when the door is virtual, and 1 when the door is real);  $o$  is the last door state observed (open or closed);  $I$  stores an visual appearance of the door.

#### 4 Gr2To - 3D Grid Map to Topological Map Conversion

Gr2To Algorithm defines our approach to convert the grid map into the augmented topological map and it has been optimized to work in indoor robots. Gr2To is divided into five stages.



**Fig. 1.** At left, the 3D grid map obtained by the RobVigil [4]. The red points are occupied cells that are higher than 1.80 meters, the white are occupied cells bellow. This map was obtained in the first floor of the building I of Engineering Faculty From Porto University. At right top, the 2D grid map obtained using the 3D grid map and Gr2To compression stage. At right bottom, the 2D grid map obtained directly from a 2D SLAM approach with the laser range finder at 1.20 meter from the floor. Using a 3D grid map and Gr2To compression stage was possible to remove the furniture present in the environment and in the gridmap of 2D SLAM approach.

*First stage) Compression from 3D grid map to 2Dgrid map and Filtering.* This stage is considered in order to take the advantages when 3D grid map is available. One of those advantages is to make possible to filter the furniture and other objects/things from the grid-map. Furnitures or other kind of objects present in the environment are considered noise to our algorithm because they do not define the boundaries of a room/place. When the input is a 3D map, this map is compressed into a 2D map, figure 1. With this compression is possible to remove from the final map the furniture/noise present in the environment.

The compression algorithm compress all cells in Z-axis into a single cell, as follow:

$$map_{2D}(x, y) = \begin{cases} 0, & \text{if } NO > NO_{min} \text{ and } NF < NF_{max} \\ 1, & \text{if } NF > NF_{max} \\ 0.5, & \text{otherwise} \end{cases} \quad (7)$$

Where,  $NO$  is function that returns the number of occupied cells from  $map_{3D}(x, y, z_{min})$  to  $map_{3D}(x, y, z_{max})$ ,  $NF$  is function that returns the number

of empty cells from  $map_{3D}(x, y, z_{min})$  to  $map_{3D}(x, y, z_{max})$ . The compression algorithm requires six parameters,  $(z_{min}, z_{max}, NF_{max}, NO_{min}, P_{Max}, P_{Min})$ . Where,  $z_{min}$  and  $z_{max}$  defines the interval of 3D map to be compressed in a 2D map.  $NF_{max}$  and  $NO_{min}$  defines the maximum number of free cell and minimum number of occupied cells to consider a vertical occupied cell,  $P_{Max}$  defines the maximum probability for a cell to be consider empty cell and  $P_{Min}$  defines the minimum probability for a cell to be consider occupied cell.



**Fig. 2.** In the left gridmap, the Gr2To has marked with red circles locations with higher probability for door existence. Blue lines represent the door in a closed state. In the right gridmap, the Gr2To draws the obtained Voronoi graph diagram over the gridmap. The gray squares represents the nodes.

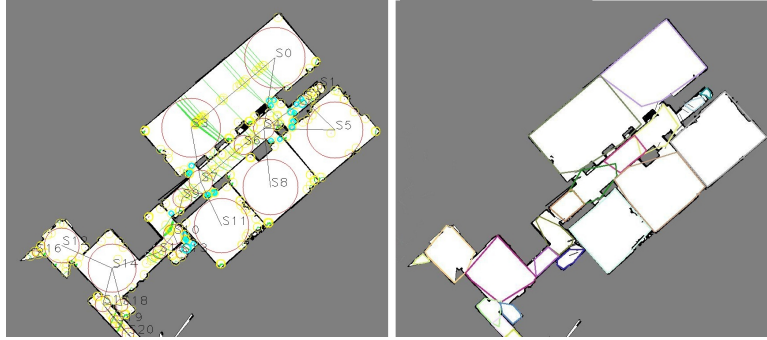
*Second stage) Door detection* In this stage is created a map of distances,  $map_{dist}$ , from the filtered map. This map of distances contains in each cell the euclidean distance to the nearest occupied cell. After, door detection algorithm search for locations with probability for door existence, figure 2. This search have two parameters  $Door_{min}$  and  $Door_{max}$ , which defines the minimum and maximum door size, in meters. These values are converted to pixel units,  $Door_{minPixel} = \frac{Door_{min}}{2 \times Gridmap_{resolution}}$  and  $Door_{maxPixel} = \frac{Door_{max}}{2 \times Gridmap_{resolution}}$ . The search finds in the distance map for cells that satisfy the condition  $Door_{minPixel} < map_{dist}(x, y) < Door_{maxPixel}$ . For the cells that satisfy this condition, the algorithm takes eight samples from distance map, using this formula:

$$v(i) = map_{dis} \left( x + d \times \cos \left( \frac{2\pi i}{8} \right), y + d \times \sin \left( \frac{2\pi i}{8} \right) \right) \quad (8)$$

Where,  $d$  is the distance parameter, in our tests this parameter takes the value of  $Door_{minPixel}$ ,  $i$  is an integer number between 0 and 7. If exist two samples that satisfy the condition  $v(i) > map_{dist}(x, y) \wedge v(j) > map_{dist}(x, y)$  with  $2 < |i - j| < 6$ , this place is consider with higher probability for door existence.

The closer cells with higher probability for door existence are used to estimate a central location. This central location is stored in vector of door locations. Closer cells are those with a distance under  $Door_{minPixel}$  pixels.

*Third stage) Voronoi Graph Diagram extraction* In this stage is constructed the Voronoi Graph diagram from the distance map, left map of figure 2. The development of this algorithm was based on previous works, as [18].



**Fig. 3.** In the left map, the Gr2To has marked with: red circles the main places (vertices), cyan circles the locations of the doors, and with black lines to show the connections between the main places. The yellow circles are the critical points. The green lines are the boundaries defined by the critical points. In the right map, the Gr2To has marked with random colors the delimitations of segmented places.

*Fourth stage) Topological map construction and door validation* To construct the topological map we have considered the definition of critical point from previous works of Thrun [14]. In this stage, the critical point is used to validate the doors found in the second stage. After this validation, the algorithm travels by the cells that belongs to the Voronoi diagram. In each one of this cell, the algorithm gets three parameters to define a circle, the circle location  $\vec{r}_c = (x_c, y_c)$  which is the same as cell location and the circle radius  $r_c = map_{dist}(x, y)$ . With this circle definition the algorithm search in all stored circles if the condition  $r_c(i) + r_c < \sqrt{(x_c - x_c(i))^2 + (y_c - y_c(i))^2}$  is satisfied. If this condition is not satisfied this circle is stored, if this condition is satisfied, the algorithm select the circle with bigger radius for the stored circles and drops the circle with smaller radius. After, the algorithms verify the next conditions for all circles:

$$\begin{cases} \sqrt{2}r_c(i) + \sqrt{2}r_c(j) < \sqrt{(x_c(j) - x_c(i))^2 + (y_c(j) - y_c(i))^2} \\ \min(r_c(i), r_c(j)) > map_{dist}\left(\frac{x_c(j) - x_c(i)}{2}, \frac{y_c(j) - y_c(i)}{2}\right) \end{cases} \quad (9)$$

if these two conditions are satisfied, the algorithm drops the circle with smaller radius. After, the algorithm find the circles connections using the the Voronoi Diagram Graph, the final result is in the figure 3.

*Fifth stage) Optimization of space delimitation and topological map augmentation with real and virtual walls.* In this stage the algorithm draws a closed door in all places that have a confirmed door. Then, it is associated one polygon of eight vertices to which stored circle. The eight vertices are placed over the circle edge and equally spaced. Each vertex of the polygon is updated to the farthest occupied cell. This update occurs inside of one beam, the beam origin is the circle center and it have an aperture of  $\frac{2\pi}{8}$  radians and is aligned to the vertex. This update is limited to a maximum distance from the center circle, defined by:

$$dist = \frac{r_c(i)}{r_c(i) + r_c(j)} \sqrt{(x_c(j) - x_c(i))^2 + (y_c(j) - y_c(i))^2} \quad (10)$$

Where,  $i$  is the index of the actual circle and  $j$  is the index of the closest circle that is inside of actual beam. Then, the algorithm test each polygon edge and if it lies over 80% of occupied cells is updated the place definition in the topological map with a real wall defined by the polygon edge. If the condition is not verified the place definition is updated with a virtual wall.

At this stage, the Gr2To approach over segments the most common corridors. This happens because the most common corridors are long rectangular spaces. To reduce this over segmentation, this algorithm merges three or more collinear places, that are directly connected by a Voronoy edge and which are defined by the circles with same radius.

At the final step, the Gr2To approach stores this augmented topological map into the robot memory and into a XML (eXtensible Markup Language) file. The vertices and door properties are related to the map referential defined by the SLAM approach.

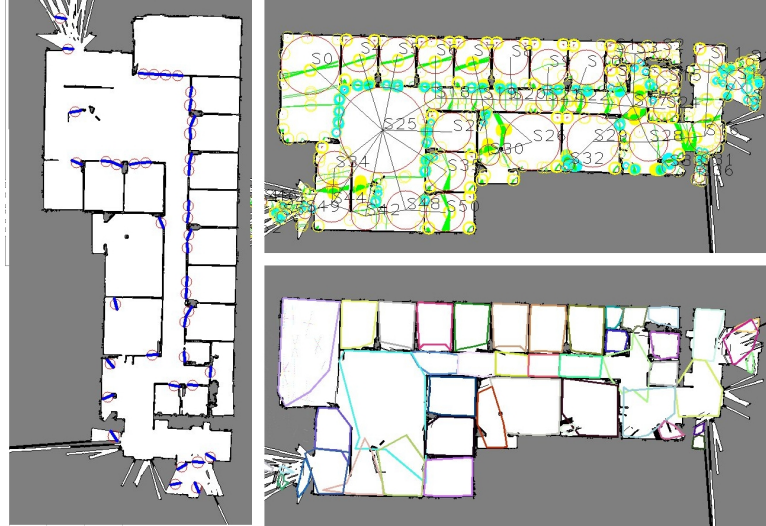
## 5 Experimental Results

This approach Gr2To was tested using three grid-maps obtained in two real scenarios and in one virtual scenario. The output results were compared to a human segmentation. The three grid-maps, input of Gr2To, were printed and given to eleven persons. It was asked, to each person individually, to place a mark in each door, room and corridor of the map. The number of each item identified and the time to complete the task is shown in the table 1. Also, these three maps were processed by Gr2To in a computer with a processor Intel Pentium Dual Core T4300 at 2,16 GHz and with 2GB of memory.

The first grid-map was obtained from the robot RoboVigil using a 3D SLAM approach [4], in the ground floor of build I of Engineering Faculty of Porto University (FEUP). In figure 2 and 3 it is possible to see intermediate steps and final place segmentation done by Gr2To.

The second grid-map was obtained in TRAC Labs facility, which is available in [23]. The third grid-map was obtained using the gazebo simulator with a turtlebot robot, Hector SLAM, and a virtual building. In figures 4 and 5 it is possible to see intermediate steps and final place segmentation done by Gr2To.



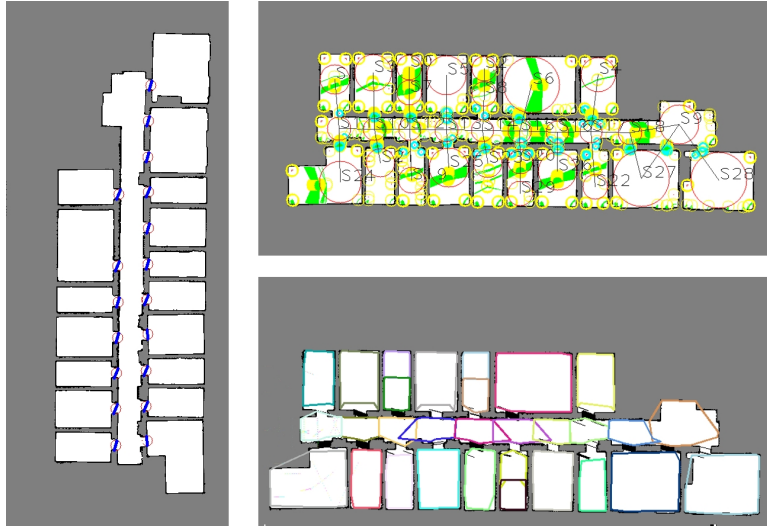


**Fig. 4.** Using the grid map of TRAC Labs facility, available in [23]. In the left map, the Gr2To has marked with red circles locations with higher probability for door existence. In the right top map, the Gr2To draws red circles in the identified places, cyan circles in critical points with high probability for door existence and black lines the connectivity between vertex. In the right bottom map, the Gr2To draws places delimitation with random colors.

**Table 1.** Human segmentation Versus Gr2To approach segmentation

Grid Map	Segmentation by Human				Segmentation by Gr2To			
	Rooms	Corridors	Doors	t(s)	Rooms	Corridors	Doors	t(s)
FEUP	7 [5,8]	3 [2,4]	9 [5,10]	34 [26,47]	11	2 (7)	9 (2,2)	8.7
TRAC Labs	17 [16,21]	2 [1,3]	18 [15,30]	43 [37,54]	29	1 (5)	39 (3,7)	8.4
Virtual	18 [17,21]	1 [1,2]	18 [18,20]	41 [33,53]	21	1 (9)	18 (0,0)	8.9

Table 1 summarizes the number of rooms, corridors and doors counted/identified by eleven persons and by Gr2To. Also, it shows the time taken to complete this task by humans and Gr2TO. In the side of human results, the three values in each cell are: the value of mean of the samples rounded to the nearest integer, and inside of brackets the minimum and maximum value of the samples. In the corridors row, of Gr2To results, there are two values, the first number is the number of corridors detected and inside of the brackets the number of vertices merged. In the doors column, of Gr2To results, there are three values: the first number is the number of doors detected, the second number is the number of missed doors, and the third number is the number of wrong doors detection.



**Fig. 5.** The intermediate steps and final place segmentation done by Gr2To with the grid map obtained from the virtual scenario

From table 1 is possible to see that in the virtual scenario the algorithm have detected the same number of doors, rooms and corridor as humans, and it took less time to process. In the real scenarios, the Gr2To have failed to detect the doors without standard width. To make these doors detectable by Gr2To it was required to increase the value of  $Door_{max}$  parameter, however it have also increased the number of outliers in the door detection. Although, these outliers can be removed using visual doors detectors. The number of doors detected by Gr2To in the TRAC Labs is more higher when compared to the human average, this happens because humans have counted several pairs of doors as one door.

In the two real scenarios, the number of rooms detected by Gr2To is more higher then the number of rooms detected by the average human. One approach to reduce the number of segmented places and make it closer to the human is merge all vertices between a door and a terminal vertex or door. However, from RoboVigil experience we found that the over segmentation appears in long rectangular places. These long rectangular places are sometimes divided by humans in two distinctive places. So, if the Gr2To can fail to detect a door and these rectangular places are sometimes divided by humans, this vertices merge should happen only when the robot gets more information about the place. The augmented topological map in the HySeLAM framework is a dynamic map and it is updated during the human robot interaction (voice), so the robot should only infer that a set of vertices are the same place when the robot gets the same human word for all vertices.

## 6 Conclusion

The contribute made in this work was the development of a novel approach to translate a grid map into a topological map. This approach is able to compress a 3D grid map into a 2D grid map and filter objects present in the environment from the map. Another contribute of this work, was the optimization of Gr2TO in order to obtain similar results to those obtained when the task is performed by a human.

This work shows that is possible to translate the robot knowledge, stored in the form of occupation grid map, to closer the human perception. Indeed, the RoboVigil with Gr2To was able to segment the obtained gridmap and ask in each place for the human word that tags the place. Gr2To simplifies the human-robot interaction and it makes possible to the robot understand simple missions, as “robot go to Sam office and then go to robot charger place”.

**Acknowledgments.** This work is financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP - 01-0124-FEDER-022701.

## References

1. Burgard, W., Cremers, A.B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: Experiences with an interactive museum tour-guide robot. *Artificial Intelligence* 114(1-2), 3–55 (1999)
2. Siegwart, R., Arras, K.O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Pignet, R., Ramel, G., Terrien, G., Tomatis, N.: Robox at Expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems* 42(3-4), 203–222 (2003)
3. Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D.: Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *Journal of Robotics Research*, 972–999 (2000)
4. Pinto, M., Moreira, A.P., Matos, A., Santos, F.: Fast 3D Matching Localisation Algorithm. *Journal of Automation and Control Engineering* 1(2), 110–115 (2013) ISSN:2301–3702
5. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine* 13(3), 108–117 (2006)
6. Thorpe, C.: Simultaneous localization and mapping with detection and tracking of moving objects. In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pp. 2918–2924. IEEE (2002)
7. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In: *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE (2011)
8. Montemerlo, M., Thrun, S.: *The FastSLAM Algorithm for Simultaneous Localization and Mapping*. Springer Tracts in Advanced Robotics (2007)

9. Santos, F.: HySeLAM - Hybrid Semantic Localization and Mapping (2012), <http://hyselam.fbnsantos.com> (accessed: May 20, 2013)
10. Elfes, A.: Occupancy grids: a probabilistic framework for robot perception and navigation. Phd, Carnegie Mellon University (1989)
11. Chatila, R., Laumond, J.: Position referencing and consistent world modeling for mobile robots. In: Proceedings of the 1985 IEEE International Conference on Robotics and Automation, vol. 2, pp. 138–145. Institute of Electrical and Electronics Engineers (1985)
12. Mataric, M.J.: A Distributed Model for Mobile Robot Environment-Learning and Navigation, Msc. MIT (1990)
13. Kuipers, B., Byun, Y.T.: A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal of Robotics and Autonomous Systems* 8, 47–63 (1991)
14. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 99(1), 21–71 (1998)
15. Joo, K., Lee, T.K., Baek, S., Oh, S.Y.: Generating topological map from occupancy grid-map using virtual door detection. In: IEEE Congress on Evolutionary Computation, pp. 1–6. IEEE (July 2010)
16. Fabrizi, E., Saffiotti, A.: Extracting topology-based maps from gridmaps. In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA. Millennium Conference Symposia Proceedings (Cat. No.00CH37065), vol. 3, pp. 2972–2978. IEEE (2000)
17. Myung, H., Jeon, H.-M., Jeong, W.-Y., Bang, S.-W.: Virtual door-based coverage path planning for mobile robot. In: Kim, J.-H., et al. (eds.) FIRA 2009. LNCS, vol. 5744, pp. 197–207. Springer, Heidelberg (2009)
18. Lau, B., Sprunk, C., Burgard, W.: Improved Updating of Euclidean Distance Maps and Voronoi Diagrams. In: IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan (2010)
19. Brunskill, E., Kollar, T., Roy, N.: Topological Mapping Using Spectral Clustering and Classification. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3491–3496 (2007)
20. Zivkovic, Z., Bakker, B., Krose, B.: Hierarchical Map Building and Planning based on Graph Partitioning. In: Proc. of IEEE International Conference on Robotics and Automation, pp. 803–809 (2006)
21. Buschka, P., Saffiotti, A.: A Virtual Sensor for Room Detection. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 637–642 (2002)
22. Choi, J., Choi, M., Chung, W.K.: Incremental topological modeling using sonar gridmap in home environment. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3582–3587. IEEE (October 2009)
23. Kortenkamp, D.: TRACLabs in new facility (2012), <http://traclabs.com/2011/03/traclabs-in-new-facility/> (accessed: May 20, 2013)