



HAL
open science

Autonomous Apartment Exploration, Modelling and Segmentation for Service Robotics

Guido Manfredi, Sandra Devin, Michel Devy, Daniel Sidobre

► **To cite this version:**

Guido Manfredi, Sandra Devin, Michel Devy, Daniel Sidobre. Autonomous Apartment Exploration, Modelling and Segmentation for Service Robotics. IAV 2016 - 9th IFAC Symposium on Intelligent Autonomous Vehicles, Jun 2016, Leipzig, Germany. pp.120-125, 10.1016/j.ifacol.2016.07.719 . hal-01355112

HAL Id: hal-01355112

<https://hal.science/hal-01355112>

Submitted on 22 Aug 2016

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

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

Autonomous Apartment Exploration, Modelling and Segmentation for Service Robotics

Guido Manfredi* Sandra Devin* Michel Devy*
Daniel Sidobre**

* LAAS-CNRS, 7 avenue du colonel Roche, F-31400 Toulouse, France
(e-mail: michel@laas.fr).

** LAAS, UPS, Universite de Toulouse, F-31400 Toulouse, France.

Abstract: This work proposes a full pipeline for a robot to explore, model and segment an apartment from a 2-D map. Viewpoints are found offline and then visited by the robot to create a 3-D model of the environment. This model is segmented in order to find the various rooms and how they are linked (windows, doors, walls) yielding a topological map. Moreover areas of interest are also segmented, in this case furniture’s planar surfaces. The method is validated on a realistic three rooms apartment. Results show that, despite occlusion, autonomous exploration and modeling covers 95% of the apartment. For the segmentation part, 1 link out of 14 is wrongly classified while all the existing areas of interest are found.

Keywords: Autonomous, Exploration, Modeling, Segmentation, Service Robotics.



Fig. 1. The ADREAM apartment is used to illustrate the different algorithms and to perform the experiments described later. Note that the furniture visible in this view is slightly different from the real setup.

1. INTRODUCTION

Let us consider a robot designed to help humans at home. It faces an unknown environment, regularly evolving and always unpredictable. In order to operate in such an environment, the robot has to first explore and model it. This model can be a two dimensional map, e.g. for navigation, or a three dimensional map, e.g. for full body motion planning or scene understanding. When a model of the environment is available, given by the user or created by the robot, it must be segmented to extract places and areas of interest. Indeed, typical requests from the user will be “fetch OBJECT from PLACE in/on AREA”, where PLACE and AREA are room and furniture names. This implies segmenting the environment model in various parts and associating spacial volumes with semantic information. On top of that, humans may lack precision, so a user could just ask “fetch OBJECT from AREA”, omitting the place. In order to handle the lack of information, the environment model segmentation has to provide areas-places links.

The present work focuses on robots using directional sensors in indoor scenarios. The next section shows that in this case most existing works propose online exploration and modeling methods. To the best of our knowledge, there are no offline methods. In section 3, we propose an offline method to find the best 3-D viewpoints to explore a site. Then, a segmentation method based on local and global cues is presented to build a topological map of the site with the different places and how they interconnect. Areas of interest are also extracted and associated to the different places. Finally, section 4 describes an experiment in a real environment to demonstrate the possibility of autonomously discovering areas, places and how they are linked in an initially unknown environment. The results are discussed and error sources identified. The algorithms described hereafter are illustrated on the ADREAM apartment (Figure 1).

2. EXISTING WORK

As analysed in Aouina et al. (2014), an environment can be modeled at various levels: the geometric level, based on features; the topological level, based on views; the semantic level, based on objects and places. Though different, all these problems can be expressed using the SLAM formalism. In Rusu et al. (2009) the authors show good localization performances by using a SLAM approach where landmarks are known fixed objects and triangulated surfaces dynamically acquired. The authors of Nüchter and Hertzberg (2008) recommend the use of a semantic map to enable the robot knowledge to be reviewable and communicable. They use a 3-D laser scanner to acquire a 3-D map through 6-D SLAM. The SLAM considers coarse features, like walls, and finer features, like objects. The work of Aouina et al. (2014) shows how to decouple the construction of a localization model and of a dense 3-

D map. The localization is performed with a 2-D laser range finder while the 3-D modeling is done with a tilting laser. In the present work, two problems are tackled: site exploration and modeling, i.e. automated construction of a volumetric map, and segmentation of the site model into meaningful parts, i.e. construction of a semantic map. We provide a quick review of the state of the art for these problems.

Though this task can appear similar to the modelling of an unknown object by a mobile sensor, like in Amigoni and Caglioti (2010), it is different. In this problem the site may be cluttered by objects preventing the robot from accessing some viewpoints. This problem is also different from the art gallery problem (Blaer and Allen (2009)). In the art gallery problem, guards have infinite view range and field of view; this is not the case for a robot. It is also different from a dense reconstruction (Newcombe et al. (2011); Arbeiter et al. (2012)) as we try to limit the robot motion by minimizing the number of viewpoints needed. According to Amigoni and Caglioti (2010) exploration strategies are divided among three types: fixed trajectories, random movements and observation positions. The first approach uses precomputed trajectories to explore any kind of site (Taylor and Kriegman (1993)). Though easy to implement, these methods do not adapt to the site’s specificities and can fail for some geometrical configurations. In the random movements approach of Freda and Oriolo (2005), random points or trajectories are chosen and the robot explores them. This kind of approach has already been rather successfully applied to vacuum cleaner robots, see Tribelhorn and Dodds (2007). However, it suffers from the trade off between number of random draw, i.e. time spent, and quality of the coverage. Finally, the last type of methods determine the best viewpoints to visit depending on some constraints. Because they are adaptive and robust to the geometry of the site, we focus on these approaches. In Reed and Allen (2000), the authors start by acquiring manually a rough estimate of the environment. Then, the model is completed automatically. This method relies on a geometrical approach with volumes representing visibility, mobility and occlusion constraints. In Blaer and Allen (2009), the three dimensional exploration is initialized with the information of a two dimensional map. A voxel grid is filled with *empty*, *occupied* and *unknown* cells. Then, a greedy algorithm is used to select from a set of random viewpoint the one seeing the maximum number of *unknown* voxels. Ray casting is used to test visibility of every *unknown* voxel. A similar approach has been demonstrated at LAAS in Albalade et al. (2002) in the framework of the European project CAMERA. A voxel grid is progressively filled with voxels being classified as *unknown*, *empty*, *occupied*, *occluded*, *occpplane* (occluded but adjacent to an empty voxel) and *border* (on the border of the line of sight). The next best view is selected based on the number of visible *unknown* voxels and on an estimation of the number of *occluded*, *occpplane* and *border* voxels discovered. Finally, in Amigoni and Caglioti (2010), the authors develop a probabilistic framework based on information theory to choose the next best view according to given constraints like travel distance and expected information. In these four works, the aim is to find the best view, acquire it and search for the next best view, and so on.

In the context of this work, the robot has already many critical processes running (motion planning, human perception, actuators control, task planning, etc.) and limited computational power. Site exploration is not a critical activity, so instead of taking processing power, we advocate in favor of doing this offline, e.g. when the robot is idle. Thus the process finds all the good viewpoints at once and they can be explored when the apartment is quiet and not being modified, for example at night. Such method is proposed in the first part of section 3.

Once a 3-D map of the environment is available, the robot should segment it into meaningful parts. This segmentation depends on various criteria. In Wurm et al. (2008), the authors propose a segmentation algorithm for multi-agent exploration. They segment a 2-D map geometrically with a Voronoi Graph (VG) (Choset and Burdick (2000)). The graph is then partitioned by separating clusters at junction nodes which are: local minima, at least of degree 2 (two edges), with at least a neighbor of degree 3 and that lead from unknown to known areas. The work from Holz et al. (2010) elaborates on this method by changing the conditions to choose a critical node. The node must be: close to a Voronoi site, of degree 2, adjacent to a junction node or adjacent to a node adjacent to a junction node (2^{nd} degree adjacency). These modifications provide a better representation of locations such as doorways. In both cases, the segmentation is based on a geometrical criterion. A more comprehensive survey on the subject is available in Bormann et al. (2016)

However, the present work aims at a human representation of the site. In particular, we want to discover the location of rooms and how they are connected by doors and windows. This allows building a graph representation of the site where rooms are vertices and windows and doors are represented by edges. The segmentation method is shown in the second part of Section 3. The resulting graph with the rooms and their connectivity is of crucial importance for tasks like object search (see Rogers and Christensen (2013)) or learning areas-place context.

3. EXPLORATION, MODELLING AND SEGMENTATION

To deal with an unknown environment, the first step is to explore it and represent it as a model. In this work, the entities of interest are rooms and areas, so the second step is finding the different rooms and areas from the model. These two steps are described hereafter.

3.1 Exploration and Modelling

The exploration step searches for a set of observation points from which the environment can be modeled. Then, the modeling process aggregates the point clouds found at each observation point and handles occlusions. In the following, it is assumed that the robot has a 2-D map of the environment for localization purposes and is able to localize itself in the world. It also assumed that the environment is not heavily cluttered so that the robot can move around and observe most of the room.

For the exploration step, the world is considered flat, so there is no line of sight occlusion. Starting from the 2-

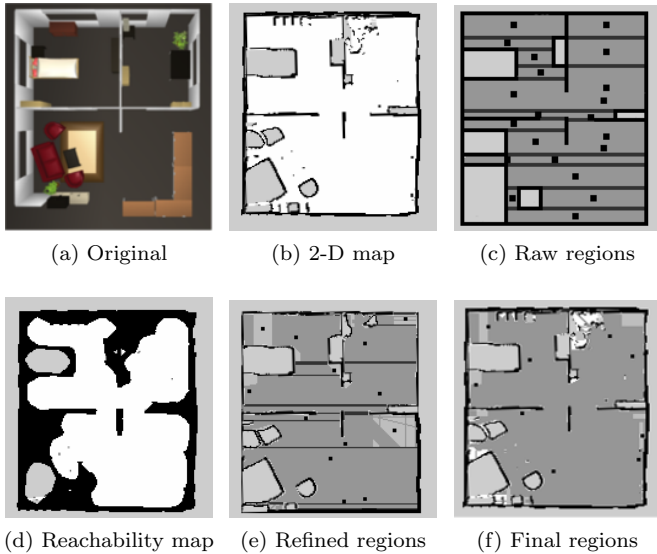


Fig. 2. View point selection process. The boundaries of the 2-D map are extended, horizontally in this case. Though, a RANSAC scheme could be used to extend lines with arbitrary orientation. Small and unreachable regions are removed. Regions with similar view points are merged.

D map, a RANSAC scheme is used to find the obstacle boundary lines and extend them until they intersect with another obstacle (Figure 2c). The new boundaries define regions, the center of these regions form a raw set of observation points. This set is refined by removing regions too small to be of interest. Some parts of the site may not be accessible by the robot. Based on the size of the robot base, a reachability map is created (Figure 2d). The points out of reach are also excluded. Finally, regions fully visible from a single observation point are merged (Figure 2e). Visibility is tested by ray casting. To find the shortest route along the observation points, the robot solves a salesman problem with a nearest neighbor approach.

Now, the modeling part assumes the world is 3-D. It starts from the observation points obtained in the previous step. The robot goes from point to point and observes its surroundings. At each point, a ray casting algorithm classify each voxel as occupied, empty or unknown, if occluded. After this first observation, if large regions of voxels remain unknown, they are represented by the projection of their barycentre on the floor plane. Taking circles centered on the robot sensor with various radii allows searching for a point of view from which the unknown region centre is visible. The robot moves to this point and explores the unknown region. This is repeated until there are no more large unknown regions around the observation point. Then the robot moves on to the next observation point. The 3-D point clouds acquired at each observation point are registered based on the robot localization and fed into a probabilistic voxel map representation, in this case an Octomap Hornung et al. (2013).

The resulting model is a voxel map containing occupied and empty voxels. The voxel map is built with its Z axis pointing upwards and its X and Y axis in the ground plane.

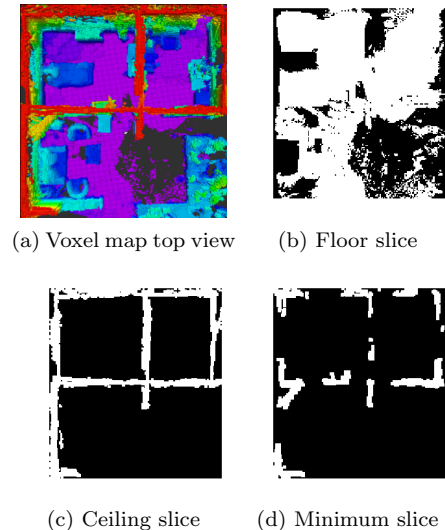


Fig. 3. The original voxel map (3a). The floor slice (3b) is removed in further processing to avoid detecting plans at the floor level. In the ceiling slice (3c), the walls are clearly visible and the rooms well segmented. The slice where the number of point is minimum (3d), shows holes at the door and windows position, plus there is little traces of furniture.

3.2 Rooms and Areas Segmentation

With a voxel map of the environment, the goal is now to extract rooms and areas. A room is defined as an empty space enclosed by walls, while an area is defined as a horizontal surface, corresponding to the planar parts of furniture.

To segment both rooms and areas, we take advantage of the fact that the model is aligned with the vertical axis. It means that some $Z = cst$ planes, hereafter called slices, bear a particular signification. There are three slices of interest. The lower Z slice corresponds to the floor (Figure 3b). The higher Z slice corresponds to the walls (Figure 3c). For the third slice we make the assumption that the windows and doors represent a considerable amount of empty surfaces with respect to the walls. Then, the slice with the minimum number of occupied points corresponds to a slice going through the door and windows holes and where the other obstacles are as little visible as possible (Figure 3d). To obtain the minimum slice, the histogram along Z of the voxel map is computed. The minimum slice corresponds to the smallest bin of the histogram. In order to reduce noise, each slice is processed with a morphological opening step. These three slices are central to segment rooms and areas.

For rooms segmentation, empty connected regions are extracted on the walls slice with a flood-fill algorithm, this yields the different rooms. Note that we define rooms as empty spaces limited by walls. The contour of each room is represented as the bounding box of the empty pixels belonging to the room (Figure 4a). The next step is to find out how the rooms are linked. This means determining which rooms are linked by how many windows or doors. Bridges are created between the rooms by connecting each point of a room's bounding box to the closest point

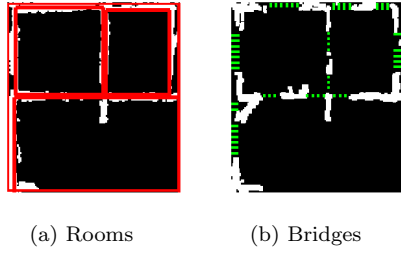
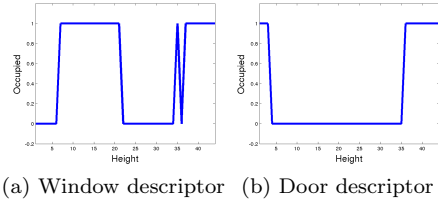
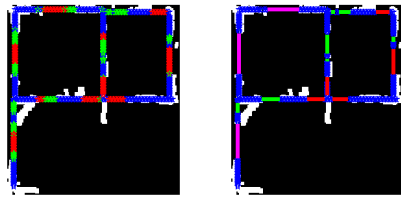


Fig. 4. The rooms are segmented and their bounding box extracted (red boxes). The exterior of the map forms an additional room. Bridges (green strips) link a room’s bounding box to the closest room bounding box, without going through an occupied pixel of the minimum slice. For clarity, only one in five bridges is drawn.



(a) Window descriptor (b) Door descriptor

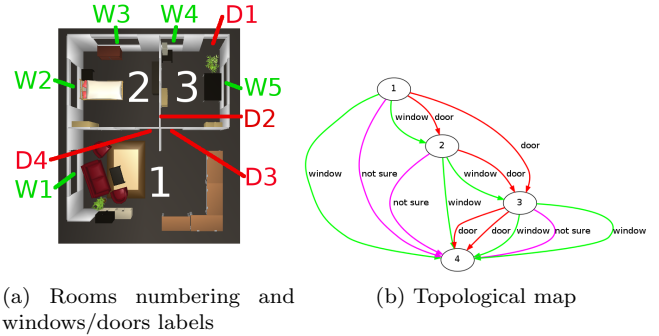


(c) Initial classification (d) Final classification

Fig. 5. Typical descriptors for a bridge going respectively through a window (5a) and a door (5b). Initial classification (5c) with the walls in blue, windows in green and doors in red. In the final classification (5d), hard cases are labeled as unknown and colored in purple.

on any other room bounding box. Raycasting between these pair of points is done with the Bresenham algorithm and the points along the ray form a bridge. A bridge is a set of points linking two points from two different rooms. If a bridge comes across an occupied pixel from the minimum slice, it is to say if a bridge goes across a wall, it is classified as a wall. The remaining bridges connect rooms through doors and windows (Figure 4b). Note that the world beyond the borders of the map is considered as an additional room. To classify a bridge as going through a window or a door, we use the Z column of each point of the bridge. The columns are combined with a *OR* operation to obtain a binary descriptor for the whole bridge. The descriptors are then fed to a hierarchical clustering method to categorize each bridge. The clustering is done by computing the euclidean distance with respect to the clusters medians. The descriptors have specific shapes depending on whether they go across a door (Figure 5b) or a window (Figure 5a). This results in an initial classification (Figure 5c).

With all the bridges classified, the adjacent bridges are grouped into segments. The segments represent whole windows and doors. To categorize a segment, each composing bridge votes for its category (door or window). This allows classifying correctly partially filled windows or doors, for example with a potted plant. If the difference in number of door and window bridges is lower than 75%, it is classified as unknown (Figure 5d). Small segments are removed as they are likely to originate from noise. Finally, in the same way as for the category, the bridges from a segment vote to determine the rooms connected by the segment. Knowing the number of rooms and the segments that join them, a topological map of the site is created (Figure 6b).



(a) Rooms numbering and windows/doors labels

(b) Topological map

Fig. 6. Rooms are numbered from left to right, top to bottom: 2,3,1. Room 4 corresponds to the map’s exterior.

For area segmentation, the first step is to remove the floor slice from the voxel map so the floor does not get extracted as an area. The columns corresponding to occupied pixels in the walls slice are also removed so the walls are no longer present. Then, for each part of the voxel map corresponding to a room, a histogram along Z is computed and the local maxima of the histogram are extracted (Figure 7). The slices at these heights are the ones with the most occupied points, they correspond to the slices where planar surfaces are present. Slices closer than a certain threshold, 20cm in this work, are merged as it is likely only the upper one is visible. For each of these slices, the connected regions are extracted to retrieve a set of areas (Figure 8). Using such method to retrieve the areas offers higher resilience to noise compared with a RANSAC based plane extraction.

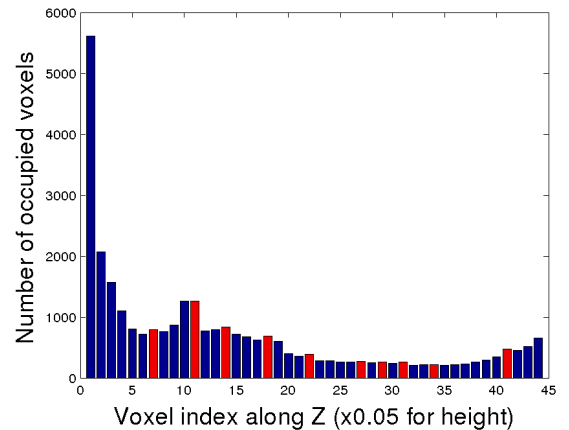


Fig. 7. Histogram of room 2 with local maxima in red.



Fig. 8. Each red box represents an area. Boxes can be superimposed meaning that there are areas at various height levels.

The result of this rooms and areas segmentation stage is a set of places and a topological map with their connections, plus corresponding areas for each room. The areas are represented by bounding boxes with their center and dimensions.

4. EXPERIMENTAL DETAILS AND RESULTS

4.1 Experiments

For validation, the methods described above are illustrated on a three room apartment staged at the LAAS-CNRS experimental ADREAM building (Figure 3a). The apartment is composed of three rooms furnished with IKEA furniture and as similar to a real apartment as possible. The robot is a PR2 running ROS and equipped with a base 2-D laser range finder, a head mounted Kinect. A map of the apartment is built with the laser range finder by teleoperating the robot while running the gmapping package. The localization is done through the AMCL package which merges the laser SLAM, odometry and map data to estimate the robot pose. The localization and laser data allow updating a 2-D collision map. Finally, the navigation and collision avoidance is done with the PR2 navigation stack.

For the exploration and modelling, the robot autonomously computes the observation points and models the site. When moving, the robot is localised thanks to AMCL, this allows registering the point clouds from the Kinect and aggregating them in an Octomap (see Hornung et al. (2013)).

The segmentation step is done offline and a number of rooms and interest areas are found. To estimate the segmentation quality, the number of areas and their height are compared with the ground truth. Area dimensions are not accounted for as they are considered small enough to be handled by an appropriate scanning strategy. The percent of false positive and false negative are used as error metric.

4.2 Results

Contrary to the preferred full-teleoperation method when modeling a site, in this work the robot is only teleoperated when creating the 2-D localization map. The environment's 3-D map is created in a fully autonomous fashion.

The exploration results (Figure 9) show that as few as 5% of the voxels remain unknown, these represent the parts

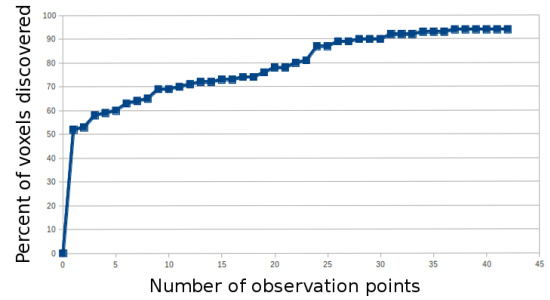


Fig. 9. The number of unknown voxels goes down as the number of observation points increase.

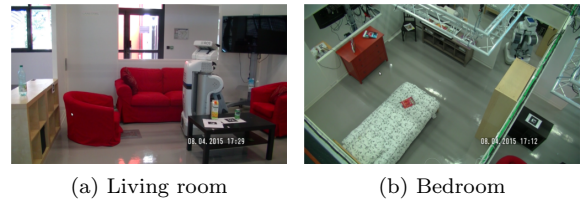


Fig. 10. Three situations where the segmentation is not sure about the segment category. Each time a piece of furniture is blocking the view: the couch (10a), the bed and the chest of drawers (10b).

of the site where the robot could not go. The resulting map is noisy for three main reasons. First comes the intrinsic precision of the Kinect sensor. In this work, the viewing range is limited to two meters but significant error is already present at this range. Improving on this, for example using a laser sensor, would dramatically reduce the scanning speed of the robot. A second factor is the robot controllers. The head motion when scanning its surroundings is jerky due to the motion controllers. Using soft controllers like in Zhao et al. (2014) could solve this problem, however to the best of our knowledge, there is no such package available for the PR2 robot. Third, the registration process is dependent on the localization precision which add to the noise. Using visual registrations methods could help reduce this noise. Though the model is noisy, it is currently hard to improve on this aspect without human intervention.

Despite the noise, the topological map created by the segmentation model (Figure 6b) has a single error, the window W5 (Figure 6a) is mistaken as a door. This is due to the fact that a table in front of the window prevents the model from seeing the wall part under the window. Moreover, when computing the bridges descriptors, the table is barely visible, it corresponds to a peak at a specific height which is not differentiable from noise. In the three other cases where there is a piece of furniture below a window (windows W1, W2 and W3), preventing points from being acquired there during the modeling step, our method finds the ambiguity and classify the segment as unknown (Figure 10).

For the interest areas segmentation, the main goal is to obtain as few false negative as possible, at the cost of false positives. It is preferable to find useless areas than to miss interest areas. Results on Figure 8 show that

there are no false negative, i.e. the 38 interest areas in the apartment are found, though there are 46% false positives. Roughly half of them are due to noise on partially reflective surfaces, screens in this case, introducing enough noise to make them appear like a narrow planar area. These could be removed with close inspection when the robot goes through the environment. It is not done here as the goal of this work is to have a purely offline segmentation.

5. CONCLUSION

Through this work, we have proposed a full pipeline which would allow a robot to autonomously explore, model and segment a site in a realistic situation. There are no hypothesis on the rooms except that a bounding box should be a good approximation of the rooms and areas shapes. More importantly, exploration and segmentation do not require rooms to be aligned or in any particular arrangement. The results showed that despite a noise inherent to modeling with a robot using a low cost sensor, the segmentation and classification stages can be performed correctly, yielding a topological map of the site. Future works involve adding online corrections to the process, so places and areas extracted during the offline segmentation are refined online.

REFERENCES

- Albalate, M.T.L., Devy, M., and Martí, J.M.S. (2002). Perception planning for an exploration task of a 3d environment. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02) Volume 3-Volume 3*, 30704. IEEE Computer Society.
- Amigoni, F. and Caglioti, V. (2010). An information-based exploration strategy for environment mapping with mobile robots. *Robot. Auton. Syst.*, 58(5), 684–699. doi:10.1016/j.robot.2009.11.005. URL <http://dx.doi.org/10.1016/j.robot.2009.11.005>.
- Aouina, A., Devy, M., and Hernandez, A.M. (2014). 3d modeling with a moving tilting laser sensor for indoor environments. In *World Congress*, volume 19, 7604–7609.
- Arbeiter, G., Bormann, R., Fischer, J., Hägele, M., and Verl, A. (2012). Towards geometric mapping for semi-autonomous mobile robots. In *Spatial Cognition VIII*, 114–127. Springer.
- Blaer, P.S. and Allen, P.K. (2009). View planning and automated data acquisition for three-dimensional modeling of complex sites. *Journal of Field Robotics*, 26(11-12), 865–891. doi:10.1002/rob.20318. URL <http://dx.doi.org/10.1002/rob.20318>.
- Bormann, R., Jordan, F., Li, W., Hampp, J., and Hgele, M. (2016). Room segmentation: Survey, implementation, and analysis. URL <http://wiki.ros.org/ipa.room.segmentation>.
- Choset, H. and Burdick, J. (2000). Sensor-based exploration: The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19(2), 96–125.
- Freda, L. and Oriolo, G. (2005). Frontier-based probabilistic strategies for sensor-based exploration. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 3881–3887. doi:10.1109/ROBOT.2005.1570713.
- Holz, D., Basilico, N., Amigoni, F., and Behnke, S. (2010). Evaluating the efficiency of frontier-based exploration strategies. *ISR/ROBOTIK 2010*.
- Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. doi:10.1007/s10514-012-9321-0. URL <http://octomap.github.com>. Software available at <http://octomap.github.com>.
- Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, 127–136. IEEE.
- Nüchter, A. and Hertzberg, J. (2008). Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11), 915–926.
- Reed, M.K. and Allen, P.K. (2000). Constraint-based sensor planning for scene modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12), 1460–1467. doi:10.1109/34.895979. URL <http://dx.doi.org/10.1109/34.895979>.
- Rogers, J. and Christensen, H.I. (2013). Robot planning with a semantic map. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2239–2244. IEEE.
- Rusu, R.B., Marton, Z.C., Blodow, N., Holzbach, A., and Beetz, M. (2009). Model-based and learned semantic object labeling in 3d point cloud maps of kitchen environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 3601–3608. IEEE.
- Taylor, C. and Kriegman, D. (1993). Exploration strategies for mobile robots. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, 248–253 vol.2. doi:10.1109/ROBOT.1993.292154.
- Tribelhorn, B. and Dodds, Z. (2007). Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education. In *Robotics and Automation, 2007 IEEE International Conference on*, 1393–1399. IEEE.
- Wurm, K.M., Stachniss, C., and Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 1160–1165. IEEE.
- Zhao, R., Sidobre, D., and He, W. (2014). Online via-points trajectory generation for reactive manipulations. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, 1243–1248. IEEE.