Room Segmentation: Survey, Implementation, and Analysis

Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hägele

***Abstract*— The division of floor plans or navigation maps into single rooms or similarly meaningful semantic units is central to numerous tasks in robotics such as topological mapping, semantic mapping, place categorization, human-robot-interaction, or automatized professional cleaning. Although many map partitioning algorithms have been proposed for various applications there is a lack of comparative studies on these different algorithms. This paper surveys the literature on room segmentation and provides four publicly available implementations of popular methods, which target the semantic mapping domain and are tuned to yield segmentations into complete rooms. In an attempt to provide new users of such technologies guidance in the choice of map segmentation algorithm, those methods are compared qualitatively and quantitatively using several criteria. The evaluation is based on a novel compilation of 20 challenging floor plans.**

I. Introduction

The segmentation of grid maps into semantically meaningful regions is an important task for many applications with mobile robots. For instance, the computation of appropriate topological maps yields significant savings in computational efforts for obtaining navigation trajectories [1]. The proper division of floor plans into individual room units can be a valuable part of semantic mapping [2] or the categorization of places [3]. Likewise, human-robot communication greatly benefits from a common notion about the extent of rooms [4]–[7]. Besides the aforementioned application cases the work in this paper was also motivated by our previous work on professional office cleaning robots [8]. Such a cleaning robot shall autonomously clean the floor in each office and clear the trash. It is desirable if the cleaning robot only needs a floor plan to conduct its work. The plan must be segmented into single rooms or work units autonomously by the robot to generate an efficient navigation sequence throughout the building. Fig. 1 provides an example for this application case.

According to the variety of applications numerous approaches to obtain an appropriate floor map partition have been proposed. Literature mainly distinguishes between fully automatic map segmentation, e.g. [1], [2], [9], and interactive segmentation incorporating the commands of a human, e.g. [4]–[7]. Also, there is a difference between methods operating on previously recorded maps [1], [10]–[13], and such computing a space partition online during the construction of a map [7], [14], [15]. Finally, map segmentation algorithms may be distinguished by their underlying approach, e.g. Voronoi graph based segmentation [1], [11], [15], graph partitioning [14], [16], feature-based segmentation [2], [12], [13], [17], [18], morphological segmentation [9], [10], distance transform-based segmentation [4], or the interpretation of architectural floor plans [19]–[21].

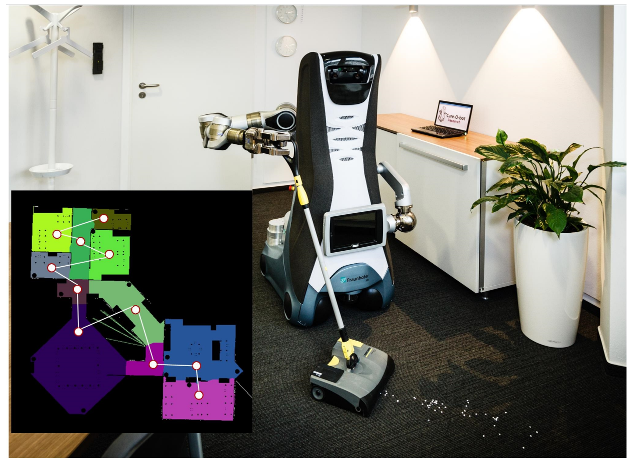

Fig. 1. An office cleaning robot utilizes a map segmentation to visit all rooms in an optimal order.

Although, autonomous and semi-autonomous map partitioning have quite some importance for several robotics technologies and applications, there is a limited body of comparative literature assessing the different approaches. Most papers just evaluate their own proposed method for a very specific purpose. Some papers solely evaluate by showing success on a couple of exemplary maps, while others also compute some numerical performance measures, however, these often apply a very task-specific measure of success. Comparisons to alternative map segmentation algorithms are scarce, a notable exception is e.g. [17]. Likewise, publicly available implementations are hard to find. This paper provides an overview over different types of room partitioning methods, describes our publicly available open source implementation of four popular map segmentation algorithms, and yields a thorough comparison of those methods based on various numerical and qualitative criteria. The evaluation utilizes a diverse compilation of publicly available maps and further maps contributed by the authors.

Summarized, the main contributions of this paper are:

1) a survey on the various kinds of room segmentation approaches,
2) a publicly available open source implementation of different room segmentation algorithms,
3) a thorough evaluation and comparison between those approaches, and
4) a set of 20 floor maps and gazebo simulation environments of diverse kinds of office environments.

[1] The authors are with the Robot and Assistive Systems Department, Fraunhofer IPA, 70569 Stuttgart, Germany. `<first name>.<last name>@ipa.fraunhofer.de`

The remainder of this paper is structured as follows. Sec. II briefly discusses the variety of approaches to floor plan segmentation, followed by a description of the implemented algorithms in Sec. III. Subsequently, Sec. IV is dedicated to the experimental evaluation of the considered methods before we conclude in Sec. V.

II. Survey: Room Segmentation Approaches

The most popular approaches to segment floor plans base upon generalized *Voronoi graphs*. Voronoi graphs are spatial partitions of a map whose nodes and edges have a maximal distance to at least two points of a finite set of obstacle points in the map (see Fig. 4, upper left image). Methods for the computation of (generalized) Voronoi graphs on grid maps are described in [22]–[27]. Thrun and Bücken [1], [28] utilize Voronoi graphs in order to find critical points on them which lie closer to two obstacle points than all other neighboring points of the Voronoi graph. Critical points hence represent narrow passages such as doors. The open space is intersected at each critical point yielding many small segments that finally become merged with their neighboring segments according to several heuristics. That algorithm creates a topological map with map cells that represent simple rooms or parts of rooms. It is intended to speed up navigation planning. In a similar way, this algorithm is applied to incremental, multi-robot mapping and map segmentation in [15] with some optimizations for selecting the critical points only at real doors. A significant extension of the basic Voronoi segmentation is introduced by Beeson et al. [11] who employ an extended Voronoi graph, which remains closer to walls in large areas, for the use with robots that have limited sensing range. They also define an ample set of heuristics for the detection of place changes through gateways.

Zivkovic et al. [16] apply *graph partitioning* to retrieve a hierarchy of topological maps that can be used with a hierarchical path planning method. They consider a basic navigation graph of approachable areas and connections between them and simplify this graph with a graph cut criterion or with spectral clustering [29]. The resulting map partitions are similar to those of Voronoi segmentation. Brunskill et al. [14] propose a system for incremental online generation of topological maps and compute the graph partitioning with spectral clustering. Additionally, AdaBoost classifiers on simple laser scanner features, similar to [2], are trained for learning the appearance of submaps for loop-closure detection.

The latter idea leads to the group of *feature-based room segmentation* algorithms that usually rely on learning grid cell labels from local appearance and harmonizing neighboring labels afterwards. Martinez Mozos et al. [2], [30] propose to provide an AdaBoost classifier with a large set of simple features computed directly on laser scanner measurements to learn the respective place label, e.g. room, hallway, or door. The place labels can then be smoothed along a robot trajectory with a hidden Markov model or within the whole map using associative Markov networks. Friedman et al. [17] connect the feature-based approach of Mozos et al. [2] with Voronoi segmentation by setting up a conditional random field on top of the Voronoi nodes to represent different types of rooms, e.g. room, hallway, door, crossing. Place types are learned with AdaBoost applied to features from both the map and the Voronoi topology, e.g. distance to obstacles or number of neighbors in the Voronoi graph. The resulting map partitions are less noisy than those in [2]. The map partitioning system of Ekvall et al. [12] attempts to separate rooms by learning the appearance of doors or narrow passages by means of line, point, and SIFT features. More complex geometric shapes, e.g. sets of (rotated) rectangles, are used to simplify a given map within the approach of Oberländer et al. [18] and capture regular room structures like offices and hallways. The fitted shapes are linked amongst each other to yield a topological map. In addition, the room type (office or hallway) can be derived from laser scanner features. Sjöö [13] describes an algorithm that assigns functional labels to a map with already determined smaller regions. Labels such as room, corridor, kitchen, office, or entrance are optimized for these regions via energy maximization. This procedure may be applied to improve an existing partition with higher level semantic information.

Morphological operators have been applied to room partitioning by Fabrizi, Buschka, and Saffiotti [9], [10]. They set up a fuzzy grid map that represents the certainty of cell occupancy by obstacles. The fuzzy-morphological opening operator is then applied to the fuzzy grid map to split areas apart that are only connected through narrow passages like doors. Starting at a seed point inside each area, the watershed algorithm determines which pixels belong to the same room.

Diosi et al. [4] describe a semi-autonomous map segmentation based on the *distance transform*. The robot is supposed to follow the user who utters place labels at different locations. After completion of the mapping the distance transform is used to group map cells to rooms and connect them with the closest label. The distance transform generates local maxima and all pixels that move to a common local maximum during gradient ascent are said to belong to one room. Another application of an *interactive room segmentation* method can be found in Spexard et al. [5], who developed a personal care robot that is able to learn the location and extent of functional places or rooms in homes. Topp and Christensen [6] describe a system that models indoor areas based on feature representations and which can incorporate place transitions without explicit door or passage indicators. A semantic place labeling application using a mixture of Gaussians model is proposed by Nieto-Granda et al. [7]. Besides assigning soft labels to locations it is capable of asking the user deliberately for the label of novel places.

Finally, there is a group of methods that interpret architectural floor plans. Ahmed et al. [19] and Heras et al. [20] discuss a system that accepts architectural floor plans with all specific symbols and textual annotations. The method for dividing the floor map into individual rooms is based on line detection to find the walls. Gaps that originate from windows or doors are interpolated. The walls are then used to construct a grid map. The final room partitioning follows straightforward from the detection of doors in the architectural plan. The system is also capable of adding semantic room names from the plan. Capobianco et al. [21] discuss a similar system that uses Canny edge detection and Hough transform to find the walls and divide the floor plan.

This overview on room segmentation methods underlines that the variety of applications produced a variety of specialized algorithms. For a practical comparison of different approaches, this paper analyzes the properties of a Voronoi graph-based method [1], a morphological segmentation, a distance transform-based algorithm, as well as a feature-based room partitioning approach [2] in the following.

III. Implemented Room Segmentation Algorithms

The following implementations of room segmentation algorithms assume that a complete grid map of the environment is available from the beginning, i.e. these methods are not optimized for online generation of map partitions. Nevertheless, the iterative application of those algorithms to the respective current state of the recorded map is always possible but may not be computationally the most efficient solution.

A. Morphological Segmentation

The morphological segmentation is identical with the procedure described in our previous work [8]. The algorithm shares some ideas with [10]. Major highlights of the morphological segmentation are the algorithmic simplicity and high computation speed.

The algorithm works on a grid map M_1 whose pixels are initially labeled accessible at mapped accessible areas and inaccessible at walls and outer areas. Such a map is depicted in Fig. 2 in the upper left corner. White areas shall represent accessibility. The walls of M_1 are grown iteratively by one pixel with the morphological erosion operator. Then a connected component analysis verifies whether previously connected accessible areas have become separated by the erosion (see Fig. 2 upper right image). If a separated region has a certain size between a lower and higher threshold, which represent the desired segment size dependent on the number of erosion steps, all its cells become labeled as an individual room r_i in a second map M_2 (see Fig. 2 lower left image), which is a copy of the original map M_1, and are set inaccessible in map M_1. This procedure repeats with the erosion until all accessible cells have been removed from map M_1. Following, the labeled areas in M_2 become extended into the accessible unlabeled space with wavefront propagation until all accessible cells have been labeled. The resulting segmentation is visualized by the lower right image in Fig. 2.

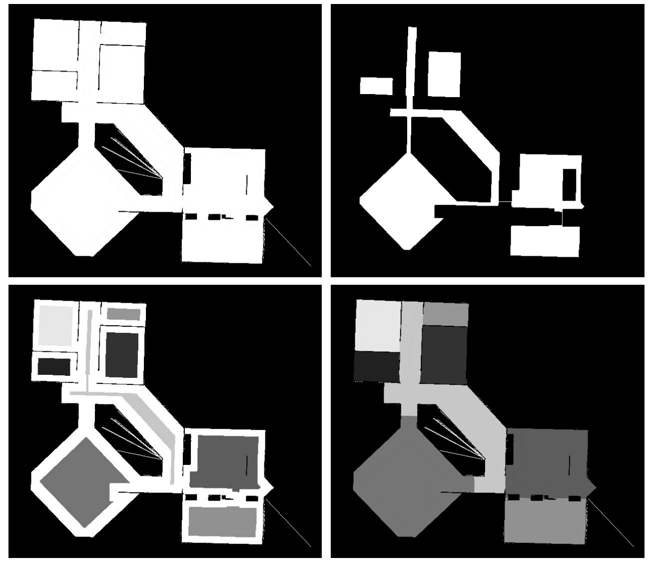

Fig. 2. Stages of the morphological segmentation algorithm: (i) initial floor map, (ii) iteratively eroded map, (iii) initial labeling of separated rooms, and (iv) segmentation after wavefront propagation.

B. Distance Transform-based Segmentation

The distance transform-based segmentation initially requires a distance transform of the map. A distance transform represents the distance of each accessible (white) pixel to the closest border pixel (black). Fig. 3 provides an example for a distance transform (upper left image) on the map of Fig. 2. The local maxima of the distance transform always lie in the center of a room. At narrow corridors or doors, the local maxima are smaller than those inside a large room. Hence, if the distance transform is thresholded appropriately one can obtain the centers of rooms (e.g. Fig. 3, upper right image). Looping through all such possible thresholds t in descending order delivers a set of room centers $\mathcal{C}$, which first increases in size until the threshold equals the local maxima at doors. From that point the number of room centers decreases again because neighboring room centers become merged through the door connections. The segmentation algorithm chooses threshold t^* in such a way that the number of retrieved room centers $|\mathcal{C}|$ is maximal (see Fig. 3, lower left image). Eventually, the found room centers are labeled uniquely and extended into remaining unlabeled space via wavefront propagation.

Effectively, the distance transform-based segmentation shares some similarities with the morphological segmentation and hence the segmentation results are quite similar for some maps. The computational complexity is comparable to the morphological segmentation.

C. Voronoi Graph-based Segmentation

The Voronoi graph-based segmentation majorly follows the original formulation of [1] but replaces the final merging step by a couple of more intricate heuristics that bias the procedure towards segmenting complete rooms. First, a generalized Voronoi graph is computed on the grid map and pruned to the major skeleton by collapsing leave edges into the node points of their origin (Fig. 4, upper left image). Any point on the resulting Voronoi graph, that has exactly two closest obstacle pixels, is a candidate for becoming a

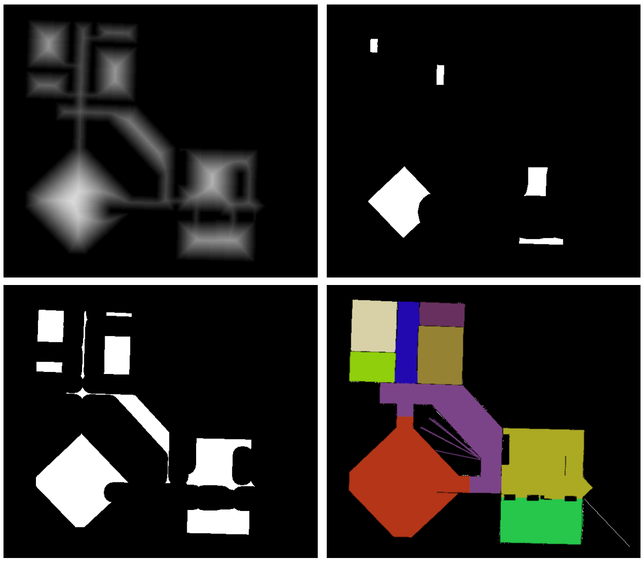

Fig. 3. Stages of the distance transform-based segmentation algorithm: (i) distance transform of the floor plan, (ii) set of room centers at a non-optimal threshold, (iii) room centers at optimal threshold t^*, and (iv) segmentation after wavefront propagation.

critical point that might delineate two room segments in the later stages. From the set of possible critical points those are stored in set $\mathcal{P}$ for further consideration which are closest to their obstacles within a certain neighborhood (Fig. 4, upper right image). Next, critical lines are drawn into the map at all critical points of set $\mathcal{P}$. These critical lines connect the critical points with their two closest obstacle points. The angle between both line segments is important if there are too many critical lines within an area. Then those with the smallest angles are removed since these often lie at corners of a wall. Critical lines with a large angle occur frequently at doors. The obtained map division still has too many small Voronoi cells that need to be merged into larger structures (Fig. 4, lower left image). Various approaches to the merge step can be found in literature (see Sec. II). In this work, we propose the following heuristics for merging Voronoi cells into room-like structures:

1) Areas smaller than a threshold (12.5m^2) with exactly one neighbor and less than 75% of the border touching walls (i.e. not a closed room area) become merged with that neighbor.
2) Small regions below a threshold size (2m^2) are merged with a surrounding area that is in touch with at least 20% of the small region's border pixels.
3) Merge regions with (i) exactly one neighbor that has maximal 2 neighbors and with (ii) at least 50% of the perimeter touching walls (this connects two parts inside the same room).
4) Merge regions that share a significant part of their borders, i.e. at least 20% for the smaller room and 10% for the larger room.
5) Merge regions with more than 40% of their perimeter touching another segment (this often happens near/below tables or other ragged obstacles).

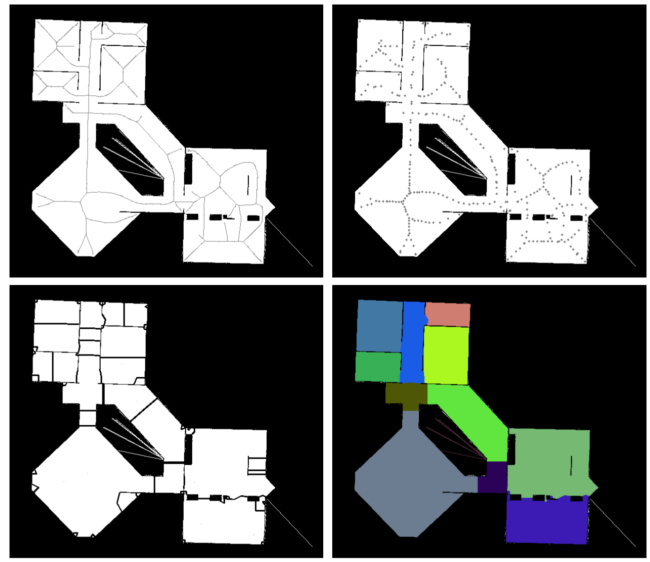

Fig. 4. Stages of the Voronoi graph-based segmentation algorithm: (i) computation of the generalized Voronoi graph, (ii) set of extracted critical points, (iii) critical lines, and (iv) segmentation after merging Voronoi cells.

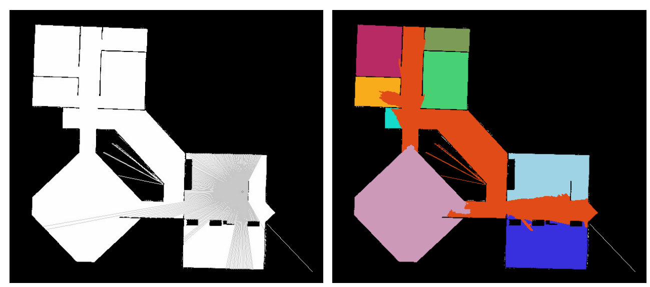

Fig. 5. Stages of the feature-based segmentation algorithm: (i) simulated laser scanner measurement within the map, and (ii) segmentation after AdaBoost classification.

After applying all merging rules the segmentation illustrated by the lower right image in Fig. 4 is obtained.

D. Feature-based Segmentation

The feature-based segmentation is implemented as described in [2]. We will provide a brief summary of the method here and refer the interested reader to the original publication. The basic data used for feature computation are the simulated rays of a 360° laser scanner that is placed at every accessible pixel of the map (see Fig. 5, left image). From this laser scan, a set of 33 simple geometric features such as difference in ray length or average ray length is computed. Following, the feature vector is classified by an AdaBoost classifier into room labels like office or corridor. Finally, all neighboring points with the same label are merged. Additionally, some smoothing can be applied by conducting a merge step with an associative Markov network. In order to obtain good results from the pixel-wise AdaBoost classification, the AdaBoost classifier needs to be trained first with sufficient amounts of representative data. This can be a drawback if new environments shall be segmented which do not resemble the training data.

IV. EVALUATION

This section applies three measures to compare the described map partitioning algorithms in different ways. First, a set of general and objective numeric properties is determined for each method, second, the segmentation quality is assessed against a human labeled segmentation, and third, the partition is evaluated on the performance of a cleaning robot that has to visit all rooms sequentially. The third case is a special application inspired by our previous work on professional office cleaning robots [8].

A. Dataset

In order to challenge the segmentation algorithms we have compiled a diverse set of 20 different office floor plans with ample individual characteristics. These floor plans are partly taken from or inspired by data publicly available from the Radish Repository [31] or from the data set of Oscar Martinez Mozos[1]. The remainder has been contributed by the authors based on real and fictional floor plans. All 20 floor plans have been generated with and without furniture in order to assess the influence of disturbances to the original architecture. The resolution of all maps was set to 0.05 m per grid cell. The ground area of the maps ranges from around 100 m^2 up to over 1000 m^2. Most of the scenes have also been modeled for the gazebo simulation environment. All maps, simulation environments, and the implementations of the segmentation software can be downloaded from our website [32].

B. General Numerical Properties

The first comparison of the map segmentation procedures is based on the following general and objective properties of the resulting segmentation:

- *Algorithm runtime:* the runtime of the segmentation algorithm in seconds (running at one core of a mobile Intel i7 4800MQ CPU with 2.7 GHz and 16 GB RAM),
- *Number of segments:* the total number of segments created by the partition,
- *Segment area:* the average area A_i of the obtained segments in m^2,
- *Segment perimeter:* the average perimeter u_i of the obtained segments in m,
- *A-Compactness:* the average area/perimeter compactness A_i/u_i^2 of the obtained segments,
- *B-Compactness:* the average area/bounding box compactness $A_i/A_{bb,i}$ of the obtained segments,
- *Shape:* average quotient of eigenvalues $e_{i,1}/e_{i,2}$ of the obtained segments.

Segments with a large perimeter and small area are complex structures with many small facets. This kind of segments is rather disadvantageous for most applications. The A-Compactness measure yields a dimensionless measure that puts segment area and perimeter into relation. The larger this measure the higher the compactness of the segment. The B-Compactness criterion measures the similarity of the obtained segment to a rectangular area. For the B-Compactness measure we compute the minimal rotated bounding box around each segment and divide the segment area A_i by the bounding box area $A_{bb,i}$. Again, the room shape resembles a rectangle the more the higher the B-Compactness. The Shape property measures the extension of the area by computing the principal components and their corresponding eigenvalues $e_{i,1}, e_{i,2}$ on the set of room cells via PCA. Dividing the larger by the smaller eigenvalue yields a measure that is close to 1 for quadratic or round shaped rooms and getting larger for elongated regions.

For comparison, those general properties are reported in Tab. I for the different segmentation methods averaged over the whole dataset. It shows that the morphological and distance transform-based approaches can be computed very fast in 1-2 s under all circumstances whereas the Voronoi segmentation takes about 13 s. The feature-based method consumes over 4 minutes of computation time majorly caused by the computationally intense sampling of laser scanner data all over the map. All four algorithms are more or less speeding up on the furnished maps since the number of pixels that needs to be processed reduces with clutter.

Morphological and distance transform-based segmentation behave very similar with respect to number of rooms, room area, perimeter, shape, and compactness measures. If furniture is added their compactness measures reduce significantly while the number of segments increases and the room area decreases significantly. Obtaining more segments with smaller area and lower compactness indicates the instability of these approaches under clutter, which can be a significant drawback for certain applications. In contrast, Voronoi segmentation yields more but smaller rooms with shorter perimeter and higher compactness. Most measures change less drastically for Voronoi segmentation when furniture is added. The feature-based approach produces an intermediate number of segments that are larger than those from the Voronoi segmentation but smaller than those from morphological and distance transform-based segmentation. The compactness measures are the lowest because place labels can change abruptly at the rugged border with this method. However, there is only little difference between empty and furnished maps in the number and area of found rooms, which renders the feature-based segmentation the most stable method. Please note that compactness always decreases for the furnished maps because the added clutter leads to longer segment perimeters and less occupied areas.

C. Quality of Room Segmentation

This part of the evaluation compares the segmentations of the proposed algorithms with a subjective ground truth segmentation. A student of our lab was told to label all maps with a reasonable ground truth segmentation which basically separates all rooms from each other and from the corridor as it would be desirable e.g. for semantic mapping. However, because every application has its own demands, the reader is encouraged to re-produce the following statistics

[1] http://webpages.lincoln.ac.uk/omozos/place_data_sets.html

TABLE I

AVERAGED GENERAL PROPERTIES ($\pm$ STANDARD DEVIATION) OF THE SEGMENTATION METHODS OVER 20 MAPS WITHOUT AND WITH FURNITURE.

	morphological	distance transform	Voronoi	feature-based	
runtime [s]	1.6 $\pm$ 2.6	1.8 $\pm$ 2.7	13.0 $\pm$ 15.3	269.3 $\pm$ 196.7	no furniture
	1.1 $\pm$ 1.2	1.3 $\pm$ 1.4	12.0 $\pm$ 14.2	245.0 $\pm$ 171.1	furnished
number of segments	22.8 $\pm$ 12.3	24.7 $\pm$ 11.7	37.9 $\pm$ 20.3	32.6 $\pm$ 21.1	no furniture
	29.5 $\pm$ 16.3	38.2 $\pm$ 19.9	43.1 $\pm$ 24.2	30.6 $\pm$ 17.2	furnished
segment area [m^2]	47.9 $\pm$ 54.0	43.7 $\pm$ 31.8	29.0 $\pm$ 24.1	36.6 $\pm$ 52.9	no furniture
	35.6 $\pm$ 45.9	27.4 $\pm$ 27.8	23.6 $\pm$ 21.1	35.8 $\pm$ 59.1	furnished
segment perimeter [m]	36.2 $\pm$ 36.6	34.2 $\pm$ 21.6	22.7 $\pm$ 9.5	36.8 $\pm$ 57.4	no furniture
	46.8 $\pm$ 43.0	39.1 $\pm$ 27.4	30.9 $\pm$ 20.6	48.8 $\pm$ 71.2	furnished
A-Compactness	0.043 $\pm$ 0.013	0.043 $\pm$ 0.013	0.048 $\pm$ 0.010	0.037 $\pm$ 0.016	no furniture
	0.018 $\pm$ 0.008	0.019 $\pm$ 0.007	0.029 $\pm$ 0.014	0.016 $\pm$ 0.009	furnished
B-Compactness	0.85 $\pm$ 0.14	0.84 $\pm$ 0.14	0.90 $\pm$ 0.11	0.80 $\pm$ 0.21	no furniture
	0.76 $\pm$ 0.14	0.74 $\pm$ 0.13	0.85 $\pm$ 0.10	0.72 $\pm$ 0.20	furnished
Shape	4.39 $\pm$ 8.14	4.35 $\pm$ 7.78	5.08 $\pm$ 7.72	8.62 $\pm$ 20.8	no furniture
	4.40 $\pm$ 8.00	4.01 $\pm$ 6.26	4.44 $\pm$ 6.38	8.71 $\pm$ 25.3	furnished

TABLE II

AVERAGED RECALL AND PRECISION ($\pm$ STANDARD DEVIATION) OF THE SEGMENTATION METHODS OVER 20 MAPS WITHOUT AND WITH FURNITURE.

		morph	distance	Voronoi	feature-based
no furniture	recall	$\mathbf{98.1}\% \pm \mathbf{2.4}\%$	$96.9\% \pm 2.8\%$	$95.0\% \pm 2.3\%$	$89.2\% \pm 11.8\%$
	precision	$88.5\% \pm 9.2\%$	$88.4\% \pm 9.3\%$	$\mathbf{94.8}\% \pm \mathbf{5.0}\%$	$90.4\% \pm 8.0\%$
furnished	recall	$84.6\% \pm 7.2\%$	$76.1\% \pm 12.3\%$	$\mathbf{86.6}\% \pm \mathbf{5.2}\%$	$85.1\% \pm 7.2\%$
	precision	$90.5\% \pm 8.1\%$	$88.4\% \pm 8.5\%$	$\mathbf{94.5}\% \pm \mathbf{5.1}\%$	$87.1\% \pm 14.5\%$

given an own ground truth segmentation with the software provided by the authors [32]. Fig. 6 provides a selection of exemplary room partitioning results and the ground truth labels. A complete listing of segmentation results for each method and each map can be reviewed online [32] as well.

It becomes evident that the morphological and distance transform-based segmentations tend to grow into the corridor under certain circumstances and then become connected with a larger group of rooms. In other maps, however, these methods yield perfectly separated rooms and corridors. Furthermore, the Voronoi graph-based segmentation shows to be quite accurate with identifying single room segments but often oversegments corridors into smaller regions. Besides the expected problems with exceptional building structures that have not been part of the training set, the feature-based segmentation exposes another shortcoming at rooms which are not separated by a corridor but touch directly: these are always merged because they carry the same semantic label.

A quantitative comparison of the algorithms with respect to the given ground truth segmentation is summarized in Tab. II by means of recall and precision. Recall is defined as the maximum pixel overlap of a ground truth room with a segmented room divided by the area of the ground truth room. Recall is high if the ground truth rooms are fully contained in the found segments. Precision is similarly defined as the maximum overlapping area of a segmented room with a ground truth room divided by the area of the segmented room. Precision is high if the segmented rooms are fully contained in the ground truth rooms. Only if both measures are high, the segmentation fits the ground truth well. On undersegmentation, recall can be high at low precision, on oversegmentation precision can be high on low recall. For the more realistic cases with furniture, the Voronoi segmentation achieves the highest recall and precision with least deviation throughout the tested maps. I.e. the Voronoi segmentation approximates the desired ground truth segmentation best from the tested methods. On maps without furniture, the highest recall can be obtained with the morphological segmentation, however, at lower precision.

D. Performance of a Cleaning Robot

As a third evaluation criterion we analyze the influence of segmentation on the performance of a professional office cleaning machine. Here we are considering the two-stage navigation procedure that is described in [8]: having obtained a room or working unit partition, the robot needs to plan its optimal traveling sequence to these rooms for cleaning the floor and for clearing trash bins. The robot also carries along a tool trolley with a waste collection bag for the trash. It is necessary to determine several central locations to place the tool trolley, which is needed in the vicinity of the rooms the robot will work at next. Moving the tool trolley, however, consumes additional handling time (90 s on average) and moving with the trolley is one third slower than the normal robot speed, so that the trolley movements should be limited. Having computed a set of trolley positions, the robot visits the neighboring rooms in an optimal order. The whole cleaning process has been implemented in a simulation which is used to simulate different parameters and optimization procedures.

Specifically, a minimal set of trolley placement locations are determined by solving a set cover problem with a greedy algorithm as described in [8]. The visiting order of the trolley positions as well as the sequence through the associated rooms around is solved as a traveling salesman problem (TSP) each time. We compared a nearest neighbor heuristic against a genetic solver and the popular Concorde solver

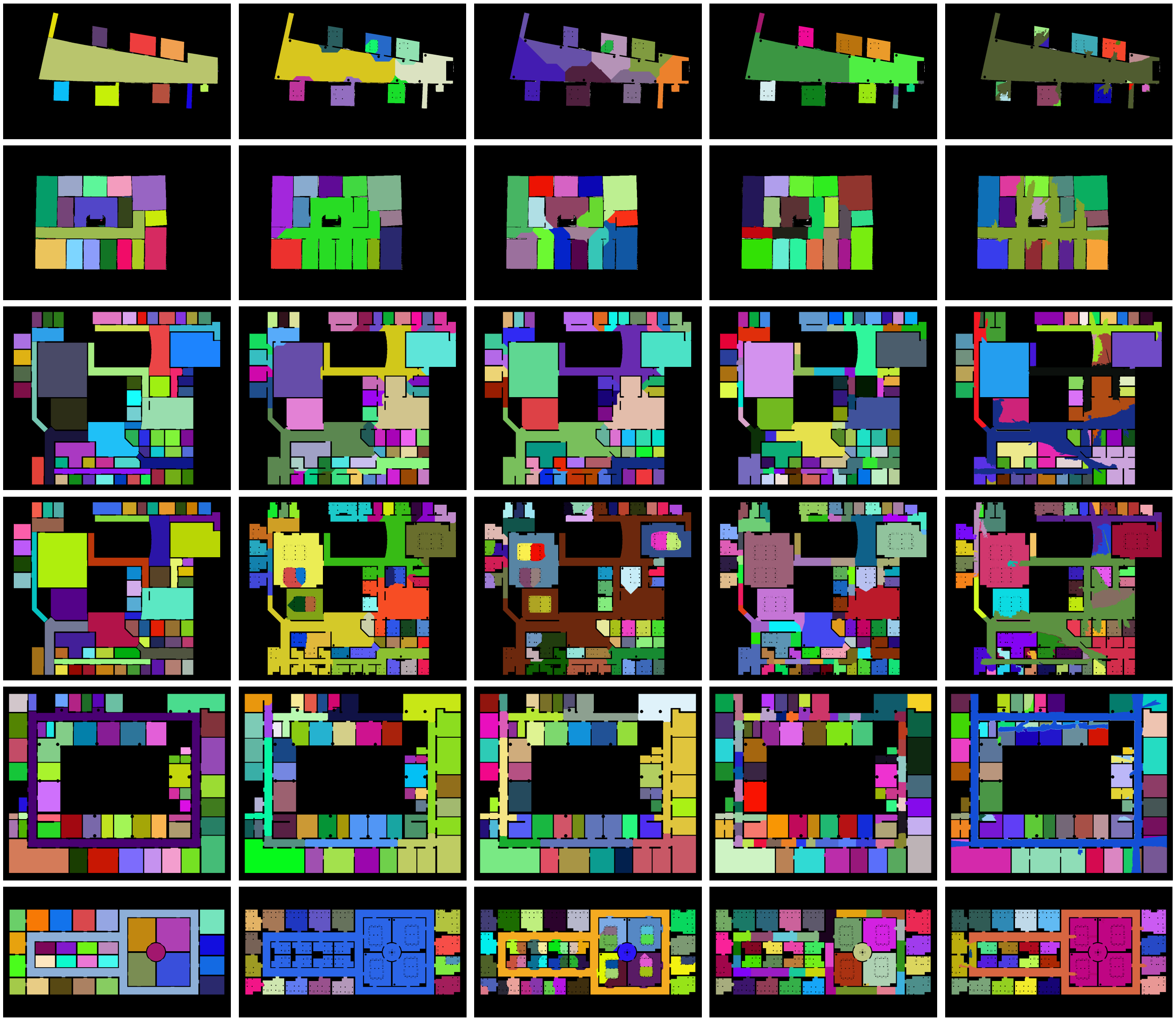

Fig. 6. Exemplary segmentation results: the first column depicts the ground truth room segmentation from human labeling, the second column shows morphological segmentation, the third column yields the distance-based segmentation, column 4 is the Voronoi graph-based segmentation, and column 5 shows the feature-based room segmentation. The segmentation results over the complete data set can be found online [32].

[33]. The maximally allowed driving distance, determined by an A* path planner, between tool trolley and associated rooms was varied in 2 m steps from 6 m to 20 m. This length has large impact on the number of trolley positions and covered rooms by each trolley location. Eventually, we also experimented with a one-stage sequence planner that computes a single optimal sequence through all rooms of the building and pulls the tool trolley to the next room whenever the maximal driving distance between room and tool trolley would become exceeded.

Together with the four segmentation algorithms, this setup yields 192 different configurations of algorithms and parameters. In the respective best configuration, the whole cleaning procedure takes the following times (averaged over all 20 maps) with the according segmentation methods:

1) morphological: 8173 s
2) distance transform: 8167 s
3) Voronoi graph: 7996 s
4) feature-based: 9026 s

The robot can clean an office environment the fastest when applying the Voronoi graph-based segmentation, probably because of its compact clusters. Next comes the morphological and distance transform-based algorithms which need almost 3 minutes longer on average. The feature-based segmentation generates the longest driving time with over 16 minutes more on average.

V. Conclusion

This paper has reviewed the literature on room segmentation. Biased by the goal to segment complete rooms instead of smaller regions, we have selected four algorithms that have been implemented as a publicly available ROS package [32].

The comparison of these algorithms with respect to several criteria has shown the advantages and disadvantages of the single algorithms, and will hopefully provide some useful guidance when the optimal method for a given application has to be selected.

As future work the authors will integrate the room segmentation method of Friedman et al. [17] into the ROS package, which is supposed to combine the advantages of Voronoi-based segmentation (room-like segments, fast computation) with feature-based partitioning (stability, automatic learning). We expect that tuning Voronoi-cell merging heuristics, as done in this work, can be avoided by having the algorithm learn from a set of desired segmentation results.

References

[1] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, Feb. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370297000787

[2] O. M. Mozos, A. Rottmann, R. Triebel, P. Jensfelt, and W. Burgard, "Semantic labeling of places using information extracted from laser and vision sensor data," in *Proceedings of the IEEE/RSJ IROS Workshop: From sensors to human spatial concepts*, Beijing, China, 2006. [Online]. Available: http://www.informatik.uni-freiburg.de/~omartine/publications/mozos2006iros_w.pdf

[3] A. Pronobis and P. Jensfelt, "Hierarchical multi-modal place categorization," in *Proceedings of the 5th European Conference on Mobile Robots (ECMR'11)*, Örebro, Sweden, Sept. 2011. [Online]. Available: http://www.pronobis.pro/publications/pronobis2011ecmr

[4] A. Diosi, G. Taylor, and L. Kleeman, "Interactive SLAM using Laser and Advanced Sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2005, pp. 1103–1108.

[5] T. Spexard, S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. Kröse, "BIRON, where are you? Enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 934–940.

[6] E. Topp and H. Christensen, "Detecting Region Transitions for Human-Augmented Mapping," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 715–720, Aug. 2010.

[7] C. Nieto-Granda, J. Rogers, A. Trevor, and H. Christensen, "Semantic map partitioning in indoor environments using regional analysis," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 1451–1456.

[8] R. Bormann, J. Hampp, and M. Hägele, "New Brooms Sweep Clean - An Autonomous Robotic Cleaning Assistant for Professional Office Cleaning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[9] E. Fabrizi and A. Saffiotti, "Augmenting topology-based maps with geometric information," *Robotics and Autonomous Systems*, vol. 40, no. 2, pp. 91–97, 2002.

[10] P. Buschka and A. Saffiotti, "A virtual sensor for room detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2002, pp. 637–642 vol.1.

[11] P. Beeson, N. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[12] S. Ekvall, D. Kragic, and P. Jensfelt, "Object Detection and Mapping for Service Robot Tasks," *Robotica*, vol. 25, no. 2, pp. 175–187, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1017/S0263574706003237

[13] K. Sjöö, "Semantic map segmentation using function-based energy maximization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 4066–4073.

[14] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 3491–3496.

[15] K. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2008, pp. 1160–1165.

[16] Z. Zivkovic, B. Bakker, and B. Kröse, "Hierarchical map building and planning based on graph partitioning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2006, pp. 803–809.

[17] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting topological structure of indoor environments via place labeling," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 7, 2007, pp. 2109–2114.

[18] J. Oberländer, K. Uhl, J. Zöllner, and R. Dillmann, "A region-based SLAM algorithm capturing metric, topological, and semantic properties," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 1886–1891.

[19] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Automatic Room Detection and Room Labeling from Architectural Floor Plans," in *2012 10th IAPR International Workshop on Document Analysis Systems (DAS)*, Mar. 2012, pp. 339–343.

[20] L.-P. d. l. Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sanchez, "Statistical segmentation and structural recognition for floor plan interpretation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 3, pp. 221–237, Dec. 2013. [Online]. Available: http://link.springer.com/article/10.1007/s10032-013-0215-2

[21] R. Capobianco, G. Gemignani, D. D. Bloisi, D. Nardi, and L. Iocchi, "Automatic extraction of structural representations of environments," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems*, 2014.

[22] S. Fortune, "A sweepline algorithm for voronoi diagrams," *Algorithmica*, vol. 2, no. 1-4, pp. 153–174, 1987.

[23] H. Choset, "Incremental construction of the generalized voronoi diagram, the generalized voronoi graph, and the hierarchical generalized voronoi graph," in *Proceedings of the First CGC Workshop on Computational Geometry*, 1997.

[24] S. O'Sullivan, "An empirical evaluation of map building methodologies in mobile robotics using the feature prediction sonar noise filter and metric grid map benchmarking suite," Ph.D. dissertation, University of Limerick, 2003.

[25] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.

[26] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of Euclidean distance maps and Voronoi diagrams," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 281–286.

[27] F. Karimipour and M. Ghandehari, "A Stable Voronoi-based Algorithm for Medial Axis Extraction through Labeling Sample Points," in *2012 Ninth International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, June 2012, pp. 109–114.

[28] S. Thrun and A. Bücken, "Integrating Grid-Based and Topological Maps for Mobile Robot Navigation," in *Proceedings of the National Conference on Artificial Intelligence*, 1996, pp. 944–951.

[29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 849–856. [Online]. Available: http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf

[30] O. M. Mozos, *Semantic Labeling of Places with Mobile Robots*, ser. Springer Tracts in Advanced Robotics (STAR). Germany: Springer Berlin Heidelberg, 2010, vol. 61.

[31] A. Howard and N. Roy, "The robotics data set repository (radish)," http://radish.sourceforge.net/, 2003.

[32] "Maps, simulation environments, room segmentation software," http://wiki.ros.org/ipa_room_segmentation, 2016.

[33] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Concorde tsp solver," http://www.math.uwaterloo.ca/tsp/concorde.html, 2006.