Tables, Counters, and Shelves: Semantic Mapping of Surfaces in 3D

Alexander J. B. Trevor, John G. Rogers III, Carlos Nieto-Granda, Henrik I. Christensen

Abstract— **Semantic mapping aims to create maps that include meaningful features, both to robots and humans. We present an extension to our feature based mapping technique that includes information about the locations of horizontal surfaces such as tables, shelves, or counters in the map. The surfaces are detected in 3D point clouds, the locations of which are optimized by our SLAM algorithm. The resulting scans of surfaces are then analyzed to segment them into distinct surfaces, which may include measurements of a single surface across multiple scans. Preliminary results are presented in the form of a feature based map augmented with a set of 3D point clouds in a consistent global map frame that represent all detected surfaces within the mapped area.**

I. INTRODUCTION

The goal of semantic mapping is to create maps that include meaning, both to robots and humans. Maps that include semantic information make it easier for robots and humans to communicate and reason about goals.

Service robots and mobile manipulators operating in indoor environments can benefit from maps that include 3D surfaces. For example, one of the most commonly discussed tasks for mobile manipulators is the object retrieval task. For this type of task, the user makes a request of the robot similar to "Get the coffee mug from the kitchen". The robot might know which part of its map corresponds to "kitchen", or at least a single point that is in the area called "kitchen", but this still doesn't restrict the search space as much as we might like. Household objects tend to be found on horizontal surfaces such as tables, counters, and shelves. A map that includes the location of all such surfaces could facilitate searches for household objects. In this paper, we present an extension to our SLAM system that allows the positions and extent of such surfaces to be included in maps.

We employ a feature-based mapping technique in which the robot builds and uses maps based on semantically meaningful features in the environment. In previous work, we have demonstrated the use of wall features (linear features detected in 2D laser scans), as well as door signs detected in images using a learned classifier. In contrast to popular grid-based mapping techniques, we believe that a feature based approach is better suited for semantic mapping because the landmarks used represent actual physical objects (or parts of objects).

The remainder of the paper is structured as follows: we provide an overview of related works in Section II. In Section III, we describe our feature based mapping system, and our extension to the mapper that allows it to map surfaces such as tables and shelves in Section IV. Preliminary mapping results are presented in Section V. Finally, conclusions and future work are described in Section VI.

II. RELATED WORK

In [22], Rusu *et.al.* performed plane segmentation on close range 3D point clouds to find horizontal surfaces in the scene, segmented out objects on these surfaces, and built models of them. Multiple planes were be extracted, each of which could support several tabletop objects. Rusu *et.al.* also investigated semantic labeling of planar surfaces in indoor environments in [23]. The plane extraction approach we use is based upon Rusu's work. This paper demonstrated that point clouds can be used very effectively to find planar surfaces as well as objects in the context of close range scenes. In our approach, we investigate this type of approach for larger scale scenes that include multiple point clouds taken from different poses by coupling it with our SLAM system.

Our work is also related to other papers concerning the topic of semantic mapping. One key result in this area is Mozos *et.al.*'s work [20] on determining semantic labels in grid based maps. This approach was successful at providing labels such as room, corridor, and doorway for cells in grid based maps. In contrast, our approach to semantic mapping builds a feature based map that includes semantically meaningful landmarks such as walls and tables instead of providing a set of labels to discrete grid cells.

Semantic labeling of points in 3D point cloud based maps has also been investigated by Nüchter *et.al.* in [19] and [18]. These papers used an Iterative Closest Point (ICP)[2] based approach to SLAM to build point cloud based maps. The resulting maps were then semantically interpreted by labeling either individual points or extracted planes with labels such as floor, wall, ceiling, or door. This approach was successfully applied to both indoor an outdoor environments.

The idea of using horizontal surfaces as landmarks has been investigated previously by Donsung and Nevatia [16]. This work presented a method for detecting the relative pose of horizontal surfaces such as tables or desks by using an edge-based computer vision approach. Surfaces could be recognized in images and localized with respect to the robot's current pose, but were not integrated into large scale maps.

The Simultaneous Localization and Mapping (SLAM) problem has seen a lot of development over the past 25 years. Smith and Cheeseman proposed the first consistent solution to the SLAM problem in [24] by expanding the Extended Kalman Filter (EKF) to include the landmark positions in the state vector and covariance matrix. A complete review of the early developments on the SLAM problem can be found in [7] and a summary of modern developments can be found in [1].

Many modern SLAM implementations now maintain the

entire robot trajectory to keep landmark poses uncorrelated since they are conditionally independent given the robot's pose. Folkesson and Christensen developed GraphSLAM [9] which used a nonlinear optimization engine to solve for robot trajectories and landmark positions. GraphSLAM techniques such as Square Root Smoothing and Mapping (SAM) developed by Dellaert [5], use sparse linear algebra and optimization techniques to improve efficiency. This technique solves a measurement matrix through sparse QR factorization. It was improved to enable incremental online operation in [14], [15]. We use the GTSAM library which Dellaert has developed as an implementation of these techniques.

III. Mapping

An overview of our SLAM system is given in this section, including the features used by our mapper. A more complete description of the mapper and its capabilities is given in our previous work [25]. A system diagram of our mapping system and its components is shown in Figure 1. Our system makes use of Willow Garage's Robot Operating System (ROS)[21] for interprocess communication, modularity, and its many useful open source tools and components.

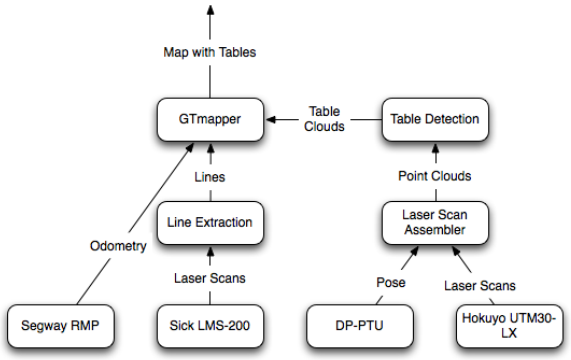

Fig. 1. A diagram representing our mapping system.

A. Features

Our system makes use of several types of features for localization and mapping, including walls extracted from 2D laser scans, as well as images of door signs extracted from camera images by a learned classifier. We will briefly describe these feature types here as they are both of interest for semantic mapping. These features are used by our SLAM system to produce a map and localize the robot.

1) Laser-line-extractor: The wall / line features used by our mapper are extracted from 2D laser scans. Examples of such line features can be seen as the thick red lines shown in Figure 7. These features are extracted using a technique based on RANdom SAmple Consensus (RANSAC) [8], as described in Nguyen *et.al.*'s paper comparing methods for detecting lines in 2D range scans [17]. Pairs of points are uniformly selected from the laser scan, and all collinear points from the scan are fitted to this line. If there are a sufficient number of inliers, this is considered a valid wall / line measurement. The process is performed iteratively, and all lines with sufficient support from points are used by our SLAM system. This type of feature is particularly useful because walls tend to stay in the same place over time, unlike furniture or other objects that might be observed in the robot's environment.

2) Door-sign-detector: Another type of feature that our mapper can make use of is measurements of door signs detected in camera images. Door signs are useful to humans who are navigating indoor environments, and can serve as landmarks for robots as well. While door signs were not available in the data set used for the mapping results presented in this paper, we provide this description due to their relevance for creating maps that include semantically meaningful features.

Our door sign detection approach uses the Histogram of Oriented Gradients (HOG) features of Dalal [3] for recognition. Hand-labeled example images were selected and scaled to a uniform size. HOG features were then extracted from this region and presented to a Support Vector Machine (SVM) classifier for training. To detect these in images, a spectral residual saliency approach described in [13] is used to detect candidate sign regions. These are then scaled to a uniform size, and presented to the previously trained SVM for classification. If classified as a door sign, it is used as a measurement by the mapper. The actual measurement used consists of a pixel location in the image corresponding to the sign region's centroid, along with the image patch corresponding to the detected region and the text string read from the sign using optical character recognition.

B. Mapping

Our SLAM system uses the GTSAM library developed by Dellaert [6]. GTSAM approaches the graph SLAM problem by using a factor graph that relates landmarks to robot poses through factors. The factors are nonlinear measurements produced by measuring various features of the types described above. The graph is optimized by converting this factor graph into a chordal Bayes Net, for which we use the elimination algorithm. To do this efficiently, it is crucial to select a good elimination ordering, which is done using the COLAMD approximate minimum degree heuristic [4]. The variables (pose and landmark features) are iteratively expressed in terms of the other variables related to them via factors, making use of the elimination order.

GTSAM makes it easy to add new factor types for new feature representations. The only components needed for a new factor type in the GTSAM framework are a measurement function and its derivatives. The measurement function gives the difference between the sensor measurement and the expected value computed from the relative position of the landmark from the robot. We have built an interface to the GTSAM library which uses the Measurement space (M-space) feature representation from Folkesson *et. al.* [11],[10], and [12]. The M-space representation allows for complex landmark types such as walls with shared endpoints.

The measurement function for M-space walls consists of terms for error in distance and angle, $\eta = (\phi, \rho)$. In addition to this measurement function, we have specified its derivatives in terms of the free variables. The M-space feature representation uses the Chain rule to simplify the expression of these Jacobians into smaller building blocks which can be shared between multiple measurement functions. A detailed explanation of this implementation can be found in [25].

IV. Surface Mapping

In this section, we describe our approach for segmenting surfaces from 3D point clouds, as well as our approach for including these in a map and determining their positions in a global map coordinate frame.

A. Plane Segmentation Approach

Our technique involves taking 3D scans of the area to be mapped, which yield 3D point clouds. We then process these point clouds in order to segment out any horizontal surfaces such as tables that are present within each point cloud. To do this, we use the well known RANdom SAmple Consensus (RANSAC) method for model fitting [8]. In our case, we are fitting planes to the full point cloud to determine the largest plane present in each cloud.

We use an iterative RANSAC to find planes in the scene, returning the plane with the most inliers from the point cloud. For the purposes of this work, only planes that are roughly horizontal (as determined by their surface normal) are considered. If the plane we have found is not horizontal, we remove all inliers for this plane from our point cloud, and then perform RANSAC again to find the next largest plane. Once we have found a horizontal plane, we perform clustering on the inliers to find contiguous regions of points within our plane, discarding clusters that are too small. This clustering step serves two purposes: to remove individual points or small clusters of points that fit to the plane but aren't part of a large contiguous surface, and to separate multiple surfaces that are coplanar but are in different locations, such as two tabletops at the same height. Each cluster with a sufficient number of points (a threshold of 500 was used for this work) is saved and will be used for mapping purposes. Finally, the inliers to the plane are removed, and we iterate again. The process terminates when no plane with a sufficient number of points can be found. The resulting set of detected surface point clouds is then sent to the mapper.

For much of our point cloud processing, we use the Point Cloud Library (PCL) developed by Rusu and others at Willow Garage, which includes a variety of tools for working with 3D point cloud data including RANSAC plane fitting, outlier removal, and euclidean clustering methods. PCL is an open source library with ROS integration, and is freely available from the ROS website.

B. Surface Mapping Approach

Our SLAM system, as described in Section III, adds measurements of features as we move through the environment. In addition to collecting odometry and 2D laser scans while driving the robot through an environment to collect data for mapping, we periodically stop the robot's movement and take 3D scans of the environment using our tilting laser scanner. Our extension in this work is to make maps that include tables detected from these point clouds by using the RANSAC based technique described in the previous subsection. To do this, our mapper needs to add a pose to the SLAM graph when a table point cloud has been generated. Measurements of lines detected in the 2D laser scan are used to build a map of the surrounding walls (and any other linear structures that may be nearby) while keeping the robot localized throughout the process. When the mapper receives a horizontal surface point cloud from our table extraction, it will add a pose to the graph, and store the point cloud attached to this pose. As we continue to navigate through the environment, these poses and the surface point clouds attached to them will be continually updated as we receive more data.

Because we are solving the *full-SLAM* problem, the robot's whole trajectory is optimized, not just the most recent pose as in a filtering (EKF or particle filter) based approach. Note that using a full-SLAM approach as opposed to a filtering based approach provides significant benefit to this technique. We rely on the ability to optimize past poses to correct the locations of the robot where point clouds were collected throughout our trajectory. In contrast, if we were simply to put our point clouds into the map frame using our best estimate at the time as a filter approach would, there would be significant error if we were ever poorly localized, with no means to correct this as we receive new information.

To build a map of the table locations based on point clouds, we begin by moving all point clouds into the map frame. Each point cloud was stored along with a pose in the graph, and this pose has been optimized as the robot continued to move around and collected more measurements. By the end of the trajectory, the pose has been optimized using landmark measurements along the whole trajectory. We then move the points in the point cloud from this optimized pose into the map frame so that we can visualize and compare them to the other surfaces in a consistent global coordinate frame.

Next, we check for overlap between detected surfaces in order to handle surfaces that were detected in point clouds from multiple locations. All point clouds have been moved to the map frame, so we just check if there are any surfaces that include points within a given distance of each other. An appropriate value for this threshold depends on the particular environment and sensor used, but for the maps shown here, a threshold of 5cm was used. If two surface measurements are within this distance, their point clouds are merged, and we consider it to be just one surface that was measured multiple times. Examples where this has occurred include the kitchen table, shown in Figure 7 in cyan, as well as the long counter-like table in the kitchen area, shown in Figure 2 in green.

Once the surfaces are in the common map frame, we have our end result: a set of point clouds representing all surfaces detected within the common coordinate system used by our localization system. The map could then be used to

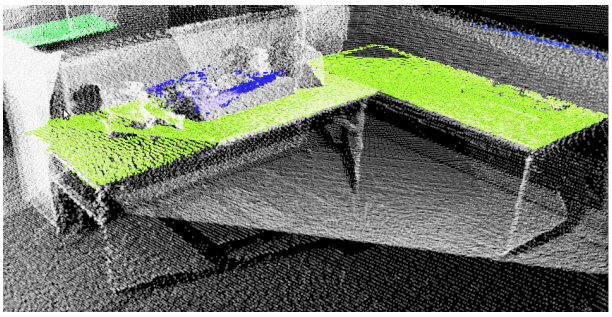

Fig. 2. Surfaces segmented in the kitchen area of the Robocup@Home 2010 arena. The L-shaped counter-like table is shown in green, and its cloud was merged from point clouds taken in multiple locations. The "stovetop" is shown in purple, which was an a planar surface several centimeters above the table. A photo of this area can be seen in Figure 4.

determine in which areas the robot should look for objects within its map. While usage of this type of map for mobile manipulation tasks is beyond the scope of this paper, we do intend to explore this in future work.

V. Results

In this section, we provide an overview of our robot platform, describe our data collection process and the area we mapped, and give preliminary mapping results.

A. Robot Platform

The robot platform used in this work consists of a Segway RMP-200 mobile base, which has been modified to be statically stable. It is equipped with a SICK LMS-291 for localization, mapping, and obstacle avoidance. 3D point cloud information is collected using a tilting laser scanner, which consists of a Directed Perception D-46-70 Pan-Tilt Unit and a Hokuyo UTM-30LX scanning laser range finder. Computation is performed on an onboard Mac Mini computer.

While not used in this work, the robot is also equipped with a Schunk PG-70 gripper with custom fingers attached to a vertically moving linear actuator, which allows the robot to grasp objects on tabletop surfaces.

B. Data Collection

Our algorithm was tested on data collected in the Robocup@Home 2010 venue. The robot was manually driven through the arena while logging all odometry and sensor data. We periodically stopped the robot and triggered a 3D laser scan, which uses the tilting laser scanner to collect 3D point clouds which were also logged for our offline mapping process.

The 3D scans were taken from many locations throughout the venue, but not all areas were scanned with the same amount of detail. The poses were not distributed evenly throughout the area, so due to the fact that the distance between neighboring points increases with the distance from the scanner, some areas have quite dense coverage while other areas have very sparse coverage.

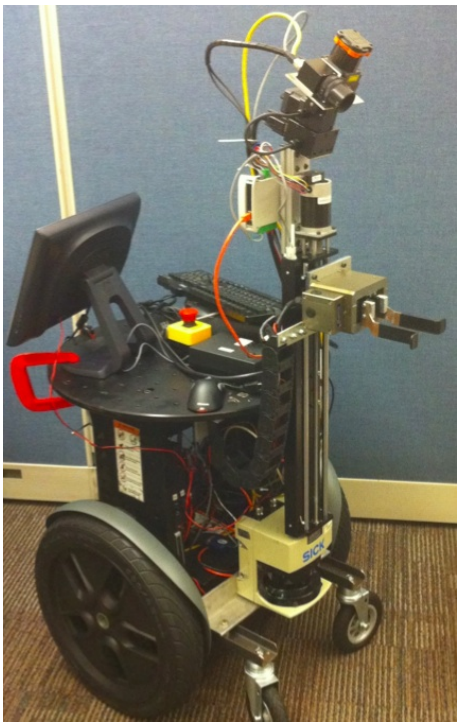

Fig. 3. A photo of the robot platform used in this work.

Fig. 4. A photo of the arena at Robocup@Home 2010, where our dataset for this work was collected. The kitchen table, "stove" area, and general layout are visible here.

C. Results

To test our algorithm, we ran our mapper offline on the logged data from the Robocup@Home 2010 arena, as seen in Figure 4 and Figure 5. We then performed a qualitative analysis of the resulting map. Many surfaces were successfully detected and mapped throughout the environment, as can be seen in Figure 7. The surfaces mapped include several tables, a set of shelves, the flat surface of a sofa's seat, the flat surface of a stove, and a small ledge. A diagram showing the room's layout is given in Figure 6, for comparison with the map resulting from our algorithm. A perspective view of the resulting map is also provided, shown in Figure 8.

Fig. 5. Another photo of the arena at Robocup@Home 2010. This shows the TV area.

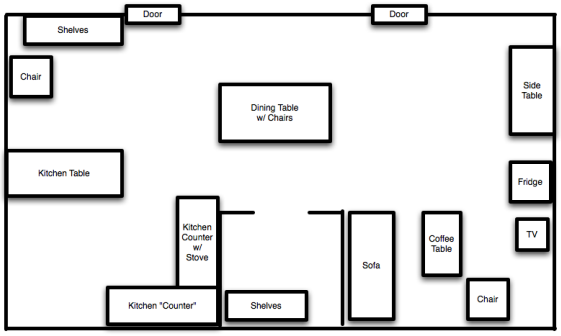

Fig. 6. A diagram showing the floor plan of the mapped area.

Several planar surfaces present in the space were not detected by our algorithm, including one set of shelves, as well as the table labeled ”sideboard” in the floor plan shown in Figure 6. In order to avoid false positives, we required a minimum of 500 points on the surface, and that each point was within no more than 10cm from the next nearest coplanar point. While these relatively strict constraints did avoid false positives, they also mean that surfaces with relatively few points such as surfaces that were only scanned from far away will not be detected. Additionally, surfaces with many objects on them have few planar inliers, because the objects occlude the surface of the plane. It should also be noted that we do not search for surfaces lower than the minimum height our robot is capable of manipulating at, so surfaces very near the ground such as the shelves that are only a few centimeters above the floor will not be detected.

It is also evident in the resulting map that there are some misalignments between scans of some surfaces that were detected in multiple point clouds, indicating a relative error between the poses in the graph from which the point clouds were taken. Specifically, the yellow-green kitchen counter surface has a small misalignment between the two scans it was detected in, as can be seen in Figure 7. The dining table, shown in cyan in Figure 7 also has a similar misalignment. As future work, we intend to investigate the use of ICP on these scans to inform our mapping and localization results with the hope that this will reduce some of this error.

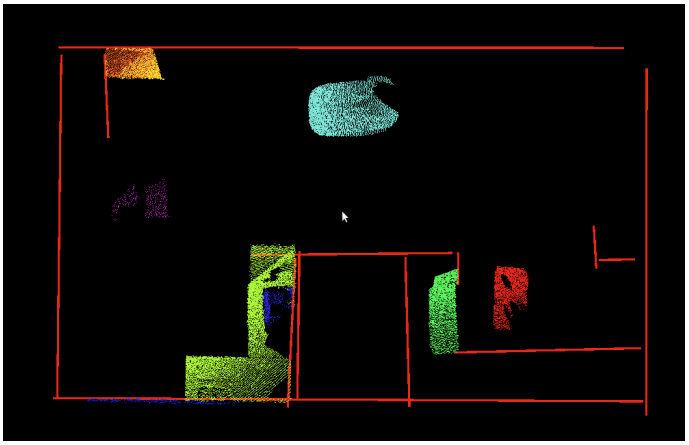

Fig. 7. An orthographic projection of the final table map. Surfaces are shown as distinctly colored point clouds. 2D linear structures used by our SLAM system are also shown as thick red lines.

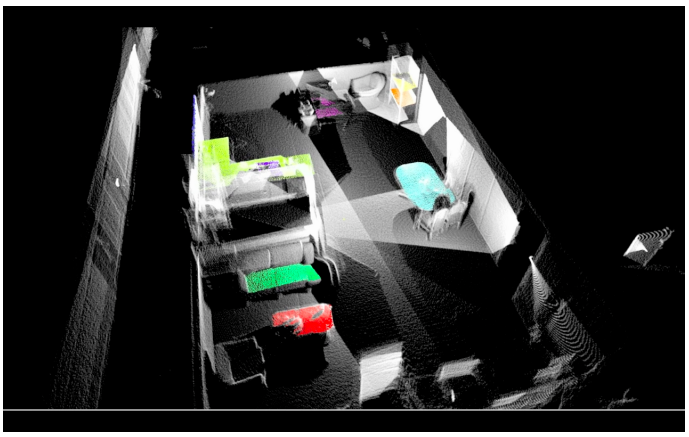

Fig. 8. A 3d perspective view of the resulting map.

VI. Conclusions & Future Work

We have presented a method for segmentation of planar surfaces from point clouds and in particular considered horizontal surfaces for integration into mapping. In domestic settings, the presence of large horizontal surfaces typically represents tables, shelves, counters, and floors. The labeled surfaces are added to the localization and mapping system to enable generation of globally consistent semantic maps. Through tracking of surfaces over the robot trajectory it is possible to fuse multiple views of surfaces into consistent representations. Mapping and scene segmentation were reported using the RoboCup@Home setup from Singapore 2010.

The current system does not leverage the segmented regions as high-level features in map estimation. There is a clear need to utilize this information as part of future work. The fusion of multi-view data to generate increased accuracy

can also be improved. Through use of ICP, it is expected that current differences across views can be reconciled into surfaces with higher accuracy. Finally there is an interest to consider how the segmented surfaces can be leveraged for planning and mobile manipulation tasks.

VII. ACKNOWLEDGMENTS

This work was made possible through the ARL MAST CTA project 104953, the Boeing corporation, and the KORUS project. We would also like to thank the reviewers for their helpful comments.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II state of the art. *Robotics and Automation Magazine*, September 2006.

[2] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, pages 239–256, 1992.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 866, 2005.

[4] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. Algorithm 836: COLAMD, a column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 2004.

[5] F. Dellaert. Square root SAM: Simultaneous localization and mapping via square root information smoothing. In *Robotics: Science and Systems*, pages 24–31, Cambridge, MA, June 2005.

[6] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1204, 2006.

[7] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, June 2006.

[8] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24:381–395, 1981.

[9] J. Folkesson and H. Christensen. Graphical SLAM - a self-correcting map. *International Conference on Robotics and Automation*, pages 1894–1900, April 2004.

[10] J. Folkesson, P. Jensfelt, and H. Christensen. Vision SLAM in the measurement subspace. *International Conference on Robotics and Automation*, pages 30–35, 2005.

[11] J. Folkesson, P. Jensfelt, and H.I. Christensen. Graphical SLAM using vision and the measurement subspace. In *Intl Conf. on Intelligent Robotics and Systems (IROS)*, pages 3383–3390, Edmundton, Canada, August 2005.

[12] J. Folkesson, P. Jensfelt, and H.I. Christensen. The M-space feature representation for SLAM. *IEEE Transactions on Robotics*, 23(5):106–115, 2007.

[13] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. *CVPR*, 2007.

[14] M. Kaess, A. Ranganathan, and F. Dellaert. Fast incremental square root information smoothing. In *Internation Joint Conference on Artificial Intelligence*, 2007.

[15] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 2008.

[16] Dongsung Kim, , and Ramakant Nevatia. A method for recognition and localization of generic objects for indoor navigation. *Image and Vision Computing*, 16:729–743, 1994.

[17] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. *International Conference on Intelligent Robots and Systems*, 2005.

[18] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[19] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3d mapping with semantic knowledge. *RoboCup 2005: Robot Soccer World Cup IX*, pages 335–346, 2006.

[20] Óscar Martínez Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robot. Auton. Syst.*, 55(5):391–402, 2007.

[21] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. *International Conference on Robotics and Automation*, 2009.

[22] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz. Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA*, 2009.

[23] R.B. Rusu, Z.C. Marton, N. Blodow, A. Holzbach, and M. Beetz. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA*, 2009.

[24] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Winter 1987.

[25] A. J. B. Trevor, J. G. Rogers III, C. Nieto-Granda, and H.I. Christensen. Applying domain knowledge to SLAM using virtual measurements. *International Conference on Robotics and Automation*, 2010.