Autonomous Semantic Mapping for Robots Performing Everyday Manipulation Tasks in Kitchen Environments

Nico Blodow, Lucian Cosmin Goron, Zoltan-Csaba Marton, Dejan Pangercic,
Thomas Rühr, Moritz Tenorth, and Michael Beetz **
Intelligent Autonomous Systems, Technische Universität München
{blodow, goron, marton, pangercic, ruehr, tenorth, beetz}@cs.tum.edu

Abstract— **In this work we report about our efforts to equip service robots with the capability to acquire 3D semantic maps. The robot autonomously explores indoor environments through the calculation of next best view poses, from which it assembles point clouds containing spatial and registered visual information. We apply various segmentation methods in order to generate initial hypotheses for furniture drawers and doors. The acquisition of the final semantic map makes use of the robot's proprioceptive capabilities and is carried out through the robot's interaction with the environment. We evaluated the proposed integrated approach in the real kitchen in our laboratory by measuring the quality of the generated map in terms of the map's applicability for the task at hand (e.g. resolving counter candidates by our knowledge processing system).**

I. Introduction

Consider a robot that is to act as a household assistant in an unknown kitchen environment. This robot has to acquire and use knowledge about where the task-relevant objects, such as the dishwasher and the oven are and how the robot can act on them – open, operate, and close them. This knowledge about the environment that can be used to perform the robot's tasks more efficiently is typically called the robot's environment model or map.

Creating such models or maps requires the robot to acquire 3D object models that categorize objects, include geometric information about them, articulation models and a hierarchical part structure, which include functional components like the handle for opening the cupboard. We call environment maps that provide these kinds of information *semantic object maps*.

From a pragmatic point of view we consider maps to be *semantic object maps* if 1) we can generate a functional model in a physical simulator in which drawers can be opened and cupboards have doors that can be opened and objects can be put inside of them; and 2) objects in the maps are linked to symbolic representations which enables us to infer e.g. that a particular 3D object model in the map is an oven if it has a container-like shape and is used for heating meals.

In this paper we investigate how these semantic object maps can be autonomously acquired by a mobile robot. The robot explores its surroundings in order to acquire a three-dimensional representation of the environment and interprets it to detect, categorize, and reconstruct the relevant objects, in particular the furniture pieces and essential devices. Sensor data is acquired using a tilting laser scanner and color video cameras.

This paper presents the next generation of the semantic mapping process described in [1]. Advancements over the previous version aim at the autonomy of map acquisition, the multi-modality of sensing technology and the use of interaction with the environment to resolve ambiguous segmentation results. We obtain the following contributions with respect to the current state of the art.

- autonomous exploration including the selection of the next best view pose in order to actively explore unknown space based using a visibility kernel and associated costmaps;
- registration, segmentation, and interpretation of color point cloud data from low-cost devices;
- segmentation of differences of point clouds through interaction using the robot manipulator.

In the remainder of the paper we proceed as follows. After the overview of the related work we describe the operation of the whole system in Section III. We provide details of the primary submodules, namely the acquisition of sensor data in Section IV, point cloud data interpretation in Section V and semantic map generation is presented in Section VI. In the end we give evaluation details and conclude with future work.

II. Related Work

Semantic mapping for robot manipulation has garnered much interest recently, and considerable work has been done.

Nüchter et al. [2] propose a 6D SLAM approach and continue processing the resulting point clouds into basic elements like walls, floor, and doors, followed by an object detection step. The robot seems to be controlled remotely, however in previous work [3], the authors propose an autonomous exploration strategy around the notion of *seen* and *unseen* lines in a two dimensional projection of their environment. A similar approach has been presented by Yamauchi et al. [4], where the authors use a grid cell approach and label each cell as *open*, *unknown* or *occupied* and use basic image processing techniques to find frontiers between unknown and open space. Navigation planning simply tries to go to the closest frontier.

** Graduate student authors in alphabetic order.

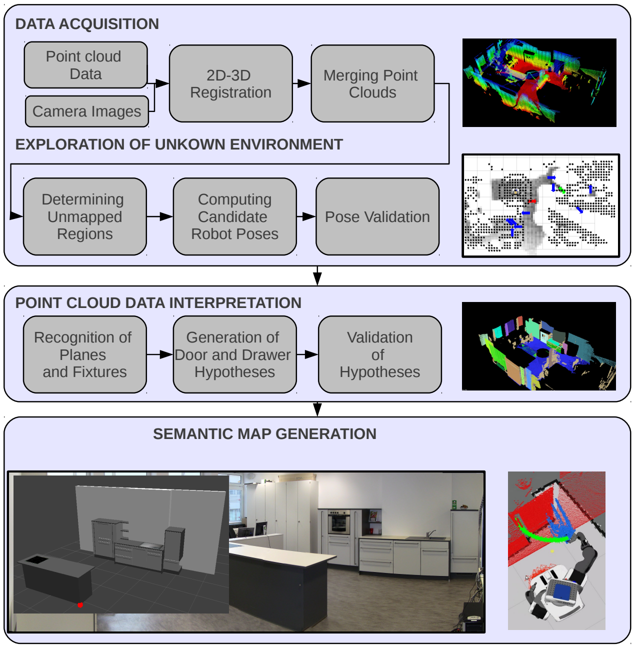
DATA ACQUISITION
Point cloud Data
Camera Images
2D-3D Registration
Merging Point Clouds
EXPLORATION OF UNKOWN ENVIRONMENT
Determining Unmapped Regions
Computing Candidate Robot Poses
Pose Validation
POINT CLOUD DATA INTERPRETATION
Recognition of Planes and Fixtures
Generation of Door and Drawer Hypotheses
Validation of Hypotheses
SEMANTIC MAP GENERATION

Fig. 1. System pipeline overview which consists of modules for Data Acquisitions (Section IV), Next Best View Planning (Section IV-B), Point Cloud Data Interpretation (Section V) and Semantic Map generation (Section VI). Images in the right most column depict the resulting outcome of every module.

In the field of autonomous 3D exploration, various approaches have been developed. The next-best-view planning problem received significant attention for a long time, especially for the purpose of object modeling, as reviewed in [5]. Our solution is a combination of the work-space exploration approach from [6] and the object modeling method from [7], firstly because a labeled voxelized representation is needed by the robot for mapping/collision avoidance [8] and model verification [9], and the flexible approach presented in [7] is readily adaptable to our problem where the 3D sensor has only 3 degrees of freedom (i.e. robot pose and orientation within the ground plane).

Point cloud data from different view points need to be merged in order to create a single representation of the environment for further analysis. A well-known technique is the Iterative Closest Point (ICP) algorithm [10], and several variants and improvements of it have been proposed. In [11], Pathak et al. exploit the planar structure of their environment, and propose a consensus based algorithm that extracts rotation and translation between two scans using fitted planes and their uncertainties. If we assume a level ground plane, we can treat the robot pose estimate to be free of errors in roll, pitch and height, and thus perform a far simpler ICP step that reduces the complexity from six degrees of freedom to three.

A recent example of 3D mapping using inexpensive RGB-D sensors was presented in [12]. They perform spatial alignment of scans, detection of loop closures and bundle adjustment to achieve a globally consistent alignment. The latter two fall outside of the scope of our paper, since we are dealing with smaller indoor environments and need to navigate considerable less in order to capture a room.

Eich et al. [13] present the transfer from the spatial to the semantic domain which is known as the gap problem in AI. They, too, extract spatial entities from unorganized point cloud data generated by a tilting laser scanner and then proceed on to shape recovery using alpha shapes, and classification of entities using the projection of rectangular structures and Hough space classifier. While their approach seems to scale well for windows, desks and entrance doors it remains unclear how they would segment drawers and furniture doors. Authors conclude with the promise to build a descriptive ontology that would allow for spatial reasoning in the semantic space – an issue that we already tackled in [14] and now successfully coupled with mapping part reported in this paper.

Our work presented in this paper is based on previous work presented in [1], where multiple point clouds of a kitchen environment are registered and scene interpretation segments the environment into vertical and horizontal planar regions. Curvature based region growing and fixture segmentation aid in separating a coherent row of cabinet doors. We extend the ideas and methods presented in [1] by augmenting laser data using a high resolution color camera and making the whole mapping process autonomous.

III. System Operation

An overview of the system presented herein is depicted in Figure 1. The robot starts at an (arbitrary) initial position in the environment and acquires a 360 degree point cloud of the room. The robot then performs a Next Best View planning step to determine positions from which the robot should acquire additional scans to fill the holes in the point cloud that were caused by occlusions.

After point cloud acquisition, the data is interpreted to extract the structure of the environment. In order to do so, the mapping system detects planes in the point cloud and categorizes them into wall, front faces of furniture, horizontal, table-like horizontal planes, etc. In the next phase, the planes that are classified as front faces of furniture pieces are further processed to locate rectangular regions and to find fixtures on them that are candidates for handles and knobs. The third step is the verification of these candidates and learning the respective articulation models that can be used by the robot to open and close the respective containers more competently.

The result of the map acquisition process is then transformed into a 3D semantic object model of the environment [14] which can be used by the robot to answer queries such as: *Where are all doors and their handles?* or *Are there doors that have two handles (hinting at undersegmentation)?*. Some formal description logic queries of this type are presented in Section VII and serve to equip the robots with the elementary capabilities that enable them to perform their tasks more reliably and efficiently.

IV. Acquisition of Sensor Data

The input to the autonomous mapping system is a set of color point clouds which were generated with a PR2 robot [15] equipped with a tilting laser scanner (Hokuyo UTM-30LX) and a registered color camera (Prosilica GC2450C, resolution 2448×2050px). Point clouds with embedded color information enable us to perform segmentation using visual as well as geometric cues, e.g. detecting furniture door handles based on their prominence from the door and segmenting the door based on its appearance. In order to find the association between the 2D and the 3D data we make use of the PR2 stock calibration [16]. Registration of incoming point clouds is done pairwise using a modified variant of the Point-to-Point Linear Iterative Closest Point (ICP) algorithm, as described in Section IV-A. Since we designed the whole mapping system to have the robot acquiring point clouds until all holes in the environment are filled, the objective of the acquisition sub-module is thus to a) generate registered 360 degree color point clouds which are further processed by the Next Best View sub-module (Section IV-B), and to b) generate a final color point cloud of the whole room which is needed by the Point Cloud Data Interpretation sub-module discussed in Section V.

Since the view frustrum of the laser scanner (180°) is wider than that of the color camera (70°), we have to pan and tilt the robot's head in order to obtain color information for all laser points. We then reproject points from the point cloud into the respective images and thus obtain the color information for every point. In cases where the projection of one point falls onto the overlapping area of two or more images, we obtain the final RGB value by simple averaging. This requires accurate calibration between the camera and the laser scanner. We perform this calibration within the whole-robot calibration procedure as described by Pradeep et al [16], which produces re-projection errors within 10 pixels for the range readings we are interested in (closer than 5 m).

A. Merging Point Cloud Views

Prior to registration, the input point clouds are processed for statistical sparse outlier removal and estimation of surface normals and curvature for each point as described in [17]. The overlapping regions between the source and the target point clouds are estimated, which facilitates faster convergence and minimizes the ICP alignment error. Using a fixed-radius nearest neighbor search within the target cloud for each point in the source cloud, we can identify which points are overlapping. Note that this relies on a sufficiently low localization error between the two scan poses, and the search radius can be modeled after the expectation of this displacement ($r = 10$ cm in our experiments).

We introduced some optimizations to adapt ICP to the problem at hand that exploit the fact that the displacement originates from the localization error of the robot, which is in the ground plane. We assume roll and pitch to be zero, and reduce the problem to a 2 dimensional one. For correspondence selection, we enforce a height criterion that requires that both points are within a certain ϵ_z band from each other. We also only select point pairs for which their curvature estimates is within a certain distance ϵ_c.

By applying the above two filters, the number of false correspondences is reduced and the transformation matrix is calculated based on the obtained correspondences. The final transformation is then applied to the original point cloud. This leads to substantially better and quicker registration (*360* seconds for the registration of *13* point clouds with an average number of *120000* points as opposed to *512* seconds). The final and the most relevant test of registration accuracy was carried out in the validation sub-module (Section V-C) where we successfully found and grasped 17 out of 18 handles in our kitchen laboratory.

B. Next Best View

In order to act and explore autonomously, the robot needs to be able to determine a good position to acquire the next scan based on the previously acquired data. The basic idea is to find robot poses from which as much new (unknown) geometry as possible is visible, while still containing enough overlap to the existing data to allow for successful registration.

The proposed solution performs the following steps: i) find a set of interesting points that promise information gain, ii) using a two-dimensional projection, compute a set of poses with an attached measure of quality from which these points can be seen, and iii) a verification step in which a simulated sensor, modeled after the actual sensor, considers the three-dimensional problem in order to assess the validity of the most promising of these poses.

1) Determining unmapped regions: For selecting interesting points that promise acquisition of unmapped areas, we employ an octree containing the accumulated points, where voxels are marked *occupied* if they contain points, *free* if the laser ray passed through them, or *unknown* otherwise. We search for voxels that are labeled free but have neighbors in unknown space. We call these voxels *fringe* voxels, and the rationale is that they represent "windows" into unexplored space. Note that these voxels can be filtered to eliminate uninteresting areas, such as ceilings.

2) Computing candidate robot poses: In the next step, we strive to estimate poses from which as many fringe voxels as possible can be scanned by the robot. For performance reasons, we do this in a two-dimensional projection of said voxels onto the ground plane. We define a *visibility kernel* $K(\phi, d_{min}, d_{max})$, which encodes the set of poses (relative to a point) from which that point is visible. The kernel is a function of the sensor's horizontal opening angle ϕ, and its minimum and maximum range d_{min} and d_{max} and represents a volume in the robot's 3D planar pose space $\langle x, y, \vartheta \rangle$. In Figure 2, we show one slice of the visibility kernel for a given robot orientation (view direction) ϑ. The shaded areas represent poses from which the voxel v can be seen given a sensor opening angle of ϕ (light gray) or ϕ_2 (dark gray).

Note that in our case, we set d_{min} to the distance of the robot base center to the closest point that can be scanned on the ground (~ 1.0 m), and d_{max} to 4.0 m, since we consider the point density of areas which are farther away too low.

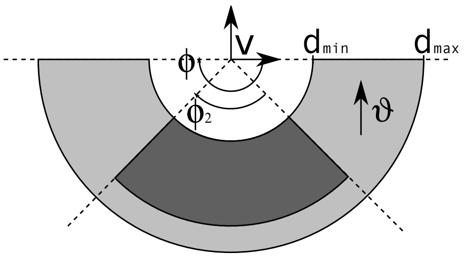

Fig. 2. Visibility kernel for a robot orientation ϑ representing all positions from where voxel $\mathbf{v}$ is visible given the sensor model.

We define a discretized representation of the robot's pose space within our environment using a 3D voxel grid. The dimensions of this grid in x and y are taken from the maximal extents of the accumulated point cloud, dilated by d_{max}, and we discretize the robot's rotation into n bins. For our experiments, we chose a coarse discretization into $n = 8$ bins because of the large opening angle of our sensor. We set the spatial resolution to 10 cm in x and y. For every of these n 2D costmaps, we loop over the fringe voxels and apply the visibility kernel as an additive stencil on the respective costmap, yielding a stack of costmaps in which cells with a high value correspond to poses from where many fringe points can be seen.

However, since the resulting data points from the next view will need to be registered to the existing points, we not only want to maximize the information gain, but also need to achieve about 50% overlap. To this extent, we compute a new stack of costmaps for all occupied voxels in the octree, and combine the two using a minimum intersection approach: Let $C_F = \{f_{x,y,\vartheta}\}$ be the fringe and $C_O = \{o_{x,y,\vartheta}\}$ the occupied costmap. The resulting costmap is thus defined as follows: $C = \{c_{x,y,\vartheta}\}$ with $c_{x,y,\vartheta} = \min(c_F, c_O)$ for all x, y, ϑ. Maxima in C represent poses from which many fringe and occupied voxels are visible according to the 2D model.

Note that we intersect C with a dilated version of the occupancy grid used for navigation to eliminate impossible poses such as within walls, and we set all cells $c_{x,y,\vartheta}$ to zero if the octree does not have an occupied voxel on the floor at $\langle x, y \rangle$. Other penalty or reward functions could easily be added, such as the distance that the robot would need to travel to the next scan pose.

3) Pose validation: In order to generate poses that achieve our 50% overlap goal mentioned earlier, we sample poses from C with a probability proportional to the pose quality and perform a raycasting step for each of the sampled poses. Given the sensor parameters such as angular resolutions and opening angles, we cast rays in the octree and count the number of occupied n_o and fringe n_f voxels. We compute the entropy as:

$$H = -\sum_{i=1}^{2} p_i log(p_i) \;\; where \;\; \begin{array}{l} p_1 = n_o/(n_o + n_f) \\ p_2 = n_f/(n_o + n_f) \end{array} . \qquad (1)$$

H is maximal if the numbers of visible fringe and occupied voxels are the same.

For every pose sample, we multiply the reward from the costmap with H and sort the list of poses by this score. The robot traverses this list and selects the first for which the navigation planner can generate a path.

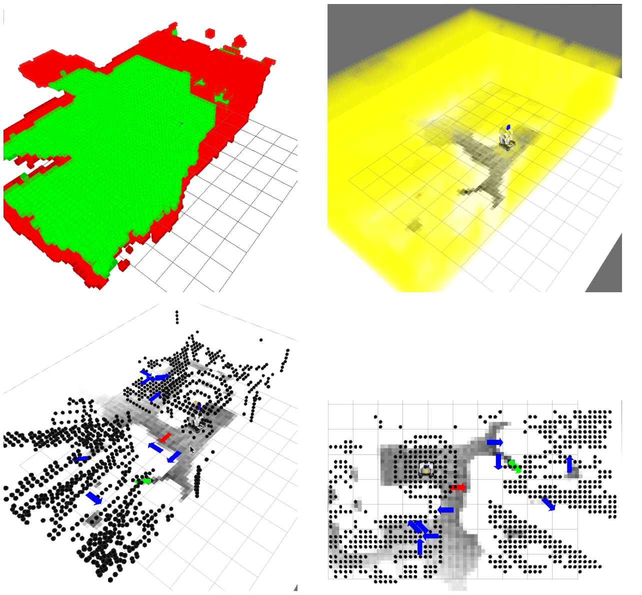

Fig. 3. Computation of next best view for the pose 1. **Top left:** free (green) and occupied (red) voxels, **Top right:** unknown voxels in yellow, **Bottom left:** orbital view of fringe voxels (black dots), next best poses (blue arrows) as well as scan-most-fringe-points pose (arrow) and best-registration pose (arrow). **Bottom right:** situation from the bottom left figure in topographical view.

V. Point Cloud Data Interpretation

The system extracts relevant planes from the registered point cloud, categorizes them as doors or drawers, walls, floor, ceiling, and tables or other horizontal structures. This is achieved by first locating the relevant planar structures, testing for the existence of fixtures, and segmenting the different doors.

A. Recognition of Planes and Fixtures

As an exhaustive search for all planes is computationally intractable, we are only searching for those that are aligned with the walls of the room. The orientations of the main walls are determined using a RANSAC based approach on the normal sphere, as in [18]. Since in many indoor environments, most of the surface normals estimated at every point coincide with one of the three main axes of the room, these directions can be used to limit the plane extraction.

After extracting the primary planes, they are classified into floor and ceiling based on (horizontal orientation and) height, and the walls based on the observation that they are adjacent to the ceiling. The remaining planar connected components – if they exceed a minimum size (set to empirically deduced value of 500 point inliers in herein presented experiments) – constitute candidates for tables or furniture faces. In order

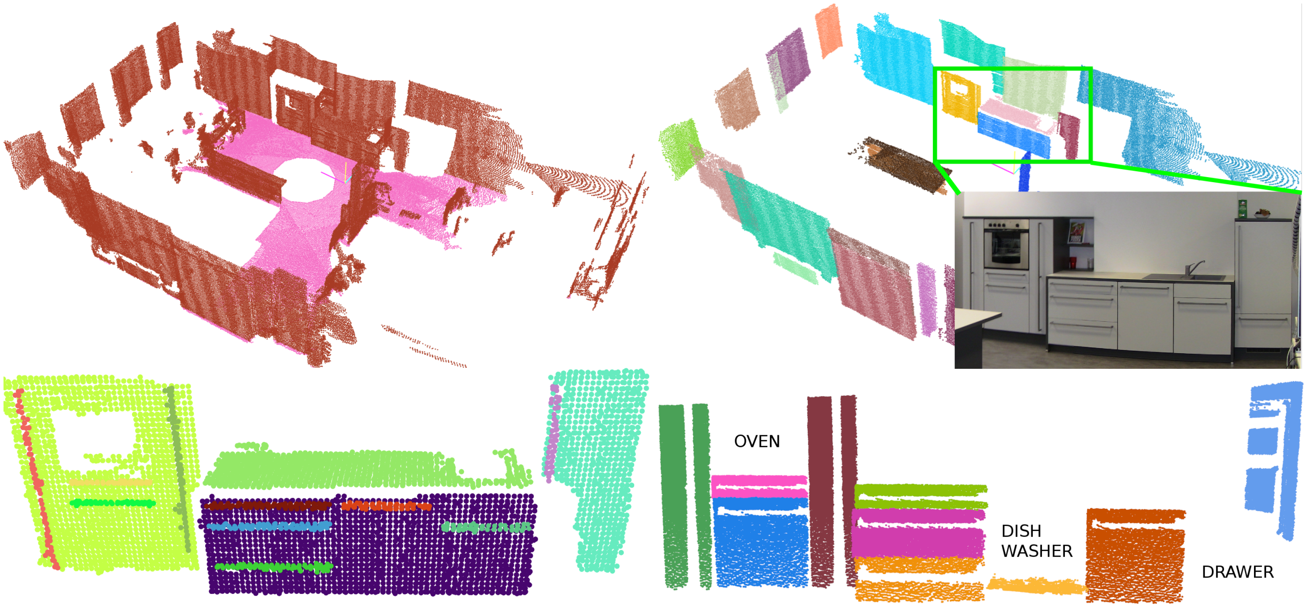

Fig. 4. Segmentation results visualised without ceiling. **Top left:** point cloud with highlighted floor, **top right:** segmentation of horizontal planes, **bottom left:** segmentation of handles shown in the photo above, **bottom right:** furniture faces after segmentation through interaction. Please note that the scans were taken after opening the drawers, dishwasher and fridge. The segmentation of the oven and the drawer under the fridge failed due to the lack of points on the handle fixture, whereas the dishwasher's inner surface is largely reflective. Figure 8 shows the final and manually augmented result.

to detect fixtures we first find point clusters that are withing the polygonal prism of the furniture faces using euclidean distance measure and then fit RANSAC lines or circles to those clusters and thereby differentiate between handles and knobs. We use a down-sampled (voxel size 3.5 cm) version of the point cloud for speed considerations and to simplify the computation of patch areas.

Kitchen appliances, doors and drawers typically have fixtures that allow interaction with them. The existence of fixtures is a good indication to the presence of these objects, so the algorithm searches for clusters of points in the vicinity of detected vertical planar structures. Since the ultimate goal is the manipulation of the handles by our robot, we discard clusters that are too big in diameter (or linear and too big in diameter) relative to the gripper aperture. These filters are simple enough to be performed for all possible clusters and explain all of the fixtures in typical kitchen environments. The result of this process can be seen in Figure 4.

B. Generation of Door and Drawer Hypotheses

In previous work [1], we found gaps between adjacent cabinet doors using curvature from laser data alone. Since the accuracy of our current laser sensor is considerably lower than the one employed there, we cannot purely rely on geometry, but perform the segmentation of furniture faces using camera images registered onto our laser data (see Figure 5).

The algorithm uses seed points around the footprint of fixtures to estimate an initial model of the color distribution of the door, consisting of the intensity values' median $\tilde{i}$ and median average distance (MAD). The seed regions are expanded by adding neighboring points whose colors match the estimated color model, using a basic region growing algorithm based on the assumption that points on the door border are surrounded by points with different color. The color model for a region is updated after all possible points are added, and the process is repeated until the values of $\tilde{i}$ and MAD stabilize. After this step, fixtures that produce overlapping segments are marked for further examination, while the rest are added to the map, along with rectangular approximations to the found planar segments.

The algorithm is parameterized on the maximum search radius for fixture footprint points, the equivalent threshold for the region growing phase, and a color threshold α, which defines how much the color of a point is allowed to deviate from the door color model. In our experiments, we found a value of $\alpha = 2 \cdot MAD$ to yield stable results. The method deals well with the shadows of handles and doors of different appearance (metallic for the oven and light gray for the rest of our kitchen). Stronger shadows did prevent small parts of some doors to be segmented correctly, but the rectangular approximation still included them.

Since the robot can interact with the environment, doors can also be segmented by opening them, and evaluating the temporal differences. However, this process is relatively slow, so one idea would be to only do this for ambiguous segmentations. However, when opening them, we can also determine the type of joint (rotational or prismatic, i.e. translational). This means that for mapping whole kitchens, including estimating articulation models, we do perform this for every handle found, but if necessary, we can fall back to geometric and visual segmentation in cases where time is critical or where the handles can not be operated by the robot gripper.

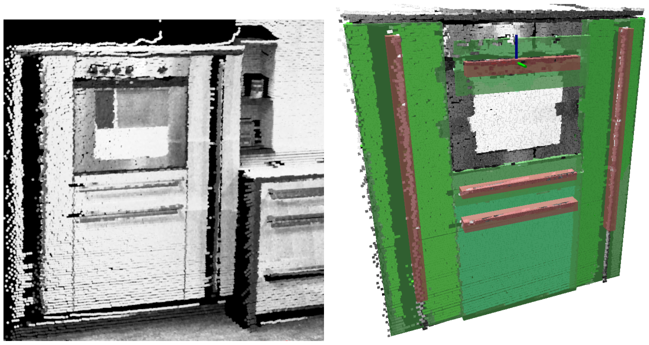

Fig. 5. Region growing based generation of drawer and door hypotheses. *Left:* point cloud data overlaid with reprojected intensity information. *Right:* the same kitchen part overlayed with segmentation results for doors and handles.

C. Active Door and Drawer Hypotheses Validation through Interaction

The idea of the robot interacting with the environment in order to overcome problems with uncertainties [19] or to verify grasp models for objects [20] has been present for a while. In this work, we describe a method to open detected drawers and cabinet doors without a priori knowledge about the validity of the handle assumption or the underlying articulation model. In fact, we can estimate articulation models for every detected handle, which can be used in subsequent manipulation tasks directly with a more optimized strategy. Furthermore, we compute the regions of the point cloud which have changed after manipulating the handle and segment the differences, which gives us the correct segmentation of all furniture parts which are rigidly connected to the handle.

Algorithm 1: Controller for opening containers with unknown articulation model. Note: poses are stored as transformation matrices (translation vector and rotation).

```
Initialize $p_0$ = point on the handle candidate;
$p_1 = p_0 + n_{furnitureplane}; t = 0$
while gripper_not_slipped_off AND cartesian_error $< threshold$ do
    if d($p_{t+1}$,projection of robot footprint) $< .1\ m$ then
        move_base(artif. workspace constr. for $p_{t+1}$)
    move_tool($p_{t+1}$)
    stabilize_grasp() (see Figure 6)
    $Rel = p_0^{-1} * p_{curr}$ with current tool pose $p_{curr}$
    Extrapolate: $Rel_s$ = scale $(Rel, (|Rel| + .05)/|Rel|)$
    $p_{t+2} = p_0 * Rel_s$
    t = t + 1
Return: Set of poses $P\{p_0 ... p_n\}$ representing the
opening trajectory.
```

1) Opening of Drawers and Doors with Unknown Articulation Models: We developed a general controller (see Algorithm 1) that makes use of the compliance of the PR2 robot's arms and the force sensitive finger tip sensors to open different types of containers without a priory knowledge of the articulation model. Since the arms lack force sensors, the algorithm uses the Cartesian error of the end effector (commanded vs. actual position) to determine when the maximum opening is reached. The algorithm relies on the grippers maintaining a strong grasp while the arms are compliant. This way the mechanism that is to be opened steers the arm in its trajectory even when there is a considerable difference between the pulling and the opening direction. The robot also adjusts its base position if the door mechanism requires this. The controller records a set of poses with the stable (aligned) grasps and returns those as an articulation model P. The controller works reliably as long as the force required to open the container is lower than the limit the friction of the gripper tips imposes.

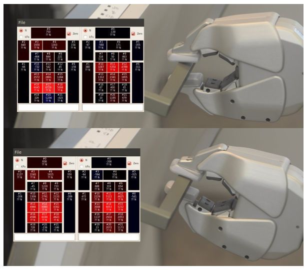

Fig. 6. The fingertip sensors are used to adjust the tool frame rotation to the rotated handle. Left part of the figure displays arrays of sensor cells on the PR2 robot's fingers. Asymmetry in the top-left part gives the measure of misalignment between the gripper and the handle.

A particular problem when opening unknown containers is the possibility of collisions of a container with the robot. This could occur e.g. when a low drawer is being opened and pulled into the robot's base. Since the articulation model is not known, an a priori motion planning step is not possible. We thus propose the following heuristic: we exclude tool poses whose projections of the gripper to the floor fall close to or within the projection of the robot's footprint from the allowed workspace limit L of the gripper. This way, the robot tries to move backwards and prevents the collision.

2) Segmentation of Point Cloud Differences: To segment out the front furniture faces we use temporal difference registration as put forth in [17], using a search radius parameter of 1 cm. We project the points that only appear in the second scan into the plane orthogonal to the last opening direction p_n. We obtain the convex hull in this plane, and assuming an environment based on rectangular furniture, we extract the width and height of the furniture front. For prismatic joints such as drawers, we can compute the distance between the two planes, which gives us a maximum opening distance and the depth of a drawer. For rotational joints, a similar value for the maximum opening angle can be found from the angles between the two planes. Depth of the container is in this case computed from the second cluster corresponding to the measurements of its inner side. Results of this step for three furniture pieces are depicted in Figure 7.

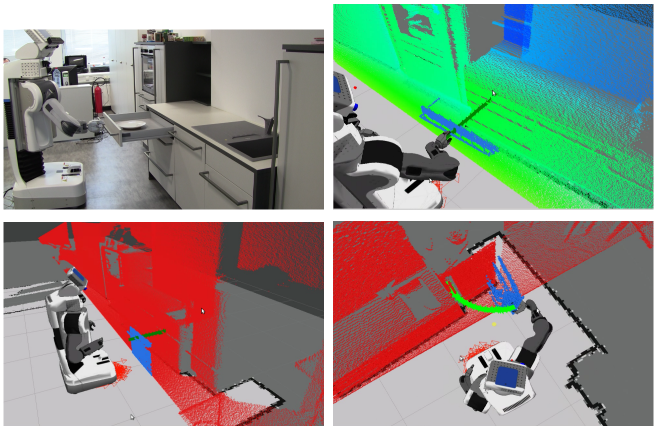

Fig. 7. Examples of interactive segmentation.

VI. Semantic Map Generation

The creation of the final semantic map from this process, by snapping of object extents to match, is an extension of the work presented in [17]. The map is represented using OWL-DL and can be shared between different modules or even different robots. For every detected furniture object, we record a dataset that contains an ID, the type of the container/articulation model found, geometric extents in depth, width and height, and the position and orientation in space. We also record hierarchical information such as a kitchenette consisting of several pieces of furniture or which handles were found on which furniture pieces. Please see Figure 8.

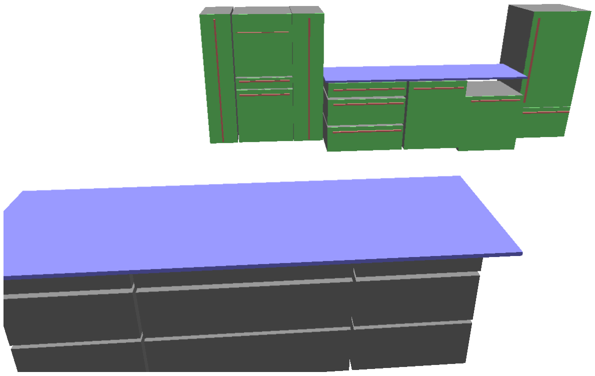

Fig. 8. Final Semantic Map as a result of the system presented herein.

The map is represented using OWL-DL and thus can be shared between different modules or even different robots. It is being used in the knowledge processing system KnowRob [14] and can be exported to URDF (Unified Robot Description Format), which we use in order to get the collision and the articulation models of environments. Please also refer to our video submission for the demonstration of various use cases.

For every detected furniture object, the map contains a dataset that holds an ID, the type of the container/articulation model found, geometric extents in depth, width and height, and the position and orientation in space. We also record hierarchical information such as a kitchenette consisting of several pieces of furniture or which handles were found on which furniture pieces.

VII. Evaluation and Results

In order to evaluate the approach proposed herein, we performed experiments in our kitchen laboratory depicted in the bottom part of Figure 1. In total, we acquired 26 point clouds, where the first scanning pose was selected randomly. The point clouds were first registered locally for the respective pose, then fed to the next best view module and finally registered globally together. Navigation planning was performed by modules provided with the robot. A thorough evaluation of the exploration behaviours generated by the next best view planning falls outside the scope of this paper and will be addressed in a separate topical publication.

A. Logical Queries

One measure of success of the mapping process is the range of queries that can be answered based on the extracted information. We linked the result of the methods described in this paper to the KnowRob-MAP system [14], which represents the environment information in terms of description logics. Objects are represented as instances of classes such as *Container* or *Handle*. These classes are arranged in a hierarchical structure, which allows to generalize: A query for *StorageConstruct*s returns all instances of sub-classes like *Cupboard* or *Drawer*. The classes and their properties can be used to formulate e.g. the following queries in order to check for mapping problems (query 1), to classify objects based on their parts (query 2) or the relation to other objects (query 3).

1) Are there doors that have two handles (hinting at under-segmentation)?

$$Door \sqcap (> \ 2\, physicalParts.Handle)$$

2) Which containers have handles as well as knobs (e.g. to search for microwave ovens, dish washers or similar appliances)?

$$\begin{aligned} Container \sqcap & \exists physicalParts.Handle \\ \sqcap & \exists physicalParts.Knob \end{aligned}$$

3) Which horizontal surfaces are likely to be counters (those that are above some drawers, cupboards, household appliances – briefly, StorageConstructs)? Which are tables (those with no StorageConstruct below)?

$$\begin{aligned} CounterTop \sqsubseteq & \ ObjectSupportingFurniture \\ & \sqcap \exists aboveOf.StorageConstruct \\ KitchenTable \sqsubseteq & \ ObjectSupportingFurniture \\ & \sqcap \neg\exists aboveOf.StorageConstruct \end{aligned}$$

VIII. Conclusions and Future Work

We presented an integrated systems paper for autonomous exploration and semantic mapping which enables the robot to autonomously explore the environment and concurrently build its semantic map. The final map contains functional models and is also linked to symbolic representations which renders our system as being useful for various applications such as e.g. classification of places and scenes. Our system is running on the PR2 robot and fully integrated into ROS (`www.ros.org`) and the resulting map has been used regularly in the demos in our laboratory (`http://www.youtube.com/user/iasTUMUNICH#p/u/7/4usoE981e7I`).

Despite system's general robustness there are still limitations in that it can not deal with translucent surfaces and surfaces that generate specular reflections. Since the 3D sensors are at the limit with these kind of problems we plan to investigate vision-driven approaches to handle them, as the one presented in [21]. Furthermore, we plan to improve or exchange our registration module in favor of one that requires less overlap [12], [11], thereby exploiting the full potential of the entropy function in the proposed next best view generation module.

Lastly, we plan to perform a large quantity of tests in various kitchens in order to increase system's generality. The use of the PR2 and ROS in various labs across the World is a big advantage in this undertaking. We intend to use such collected data to train specific classifiers for identification of more complex furniture pieces (e.g. stoves, sinks, chairs).

Acknowledgments

We would like to thank our colleagues Hozefa Indorewala, Felix Ruess and Gheorghe Rus for their support with the experiments. This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, by Willow Garage, Menlo Park, CA, and by the project "Doctoral studies in engineering sciences for developing the knowledge based society – SIDOC" contract no. POSDRU/88/1.5/S/60078.

References

[1] 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, October 11-15 2009.

[2] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Journal of Robotics and Autonomous Systems (JRAS), Special Issue on Semantic Knowledge in Robotics*, vol. 56, no. 11, pp. 915–926, 2008.

[3] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 181–198, 2003.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, July 1997, pp. 146 –151.

[5] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.

[6] P. Renton, M. Greenspan, H. A. Elmaraghy, and H. Zghal, "Plan-n-scan: A robotic system for collision-free autonomous exploration and workspace mapping," *Journal of Intelligent and Robotic Systems*, vol. 24, no. 3, pp. 207–234, 1999.

[7] G. Impoco, P. Cignoni, and R. Scopigno, "Closing gaps by clustering unseen directions," in *SMI '04: Proceedings of the Shape Modeling International 2004*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 307–316.

[8] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010, software available at http://octomap.sf.net/. [Online]. Available: http://octomap.sf.net/

[9] N. Blodow, R. B. Rusu, Z. C. Marton, and M. Beetz, "Partial View Modeling and Validation in 3D Laser Scans for Grasping," in *9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Paris, France, December 7-10 2009.

[10] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.

[11] K. Pathak, A. Birk, N. Vaškevičius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-d mapping," *Trans. Rob.*, vol. 26, pp. 424–441, June 2010. [Online]. Available: http://dx.doi.org/10.1109/TRO.2010.2042989

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Using depth cameras for dense 3d modeling of indoor environments," in *International Symposium on Experimental Robotics (ISER)*, New Delhi, India, 12/2010 2010.

[13] M. Eich and M. Dabrowska, "Semantic labeling: Classification of 3d entities based on spatial feature descriptors," in *Best Practice Algorithms in 3D Perception and Modeling for Mobile Manipualtion. IEEE International Conference on Robotics and Automation (ICRA-10), May 3, Anchorage, United States*, 5 2010.

[14] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, "KNOWROB-MAP – Knowledge-Linked Semantic Object Maps," in *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010.

[15] K. Wyrobek, E. Berger, H. V. der Loos, and K. Salisbury, "Towards a Personal Robotics Development Platform: Rationale and Design of an Intrinsically Safe Personal Robot," in *Proc. International Conference on Robotics and Automation (ICRA)*, 2008.

[16] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *International Symposium on Experimental Robotics (ISER)*, New Delhi, India, 12/2010 2010.

[17] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point Cloud Based Object Maps for Household Environments," *Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge)*, 2008.

[18] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D Modelling of Novel Objects from a Single View," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.

[19] M. Dogar and S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, October 2010.

[20] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan, "Towards reliable grasping and manipulation in household environments," in *Proceedings of RSS 2010 Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments*, 2010.

[21] M. Fritz, T. Darrell, M. Black, G. Bradski, and S. Karayev, "An additive latent feature model for transparent object recognition," in *NIPS*, 12/2009 2009.