Semantic Object Maps for Robotic Housework - Representation, Acquisition and Use

Dejan Pangercic, Benjamin Pitzer
Robert Bosch LLC
Palo Alto, CA
dejan.pangercic@us.bosch.com
benjamin.pitzer@us.bosch.com

Moritz Tenorth
Intelligent Robotics and Communications Lab
ATR, Kyoto, Japan
tenorth@atr.jp

Michael Beetz
Intelligent Autonomous Systems/Artificial Intelligence
Department of Computer Science and
Centre for Computing Technologies (TZI)
University of Bremen, Germany
beetz@informatik.uni-bremen.de

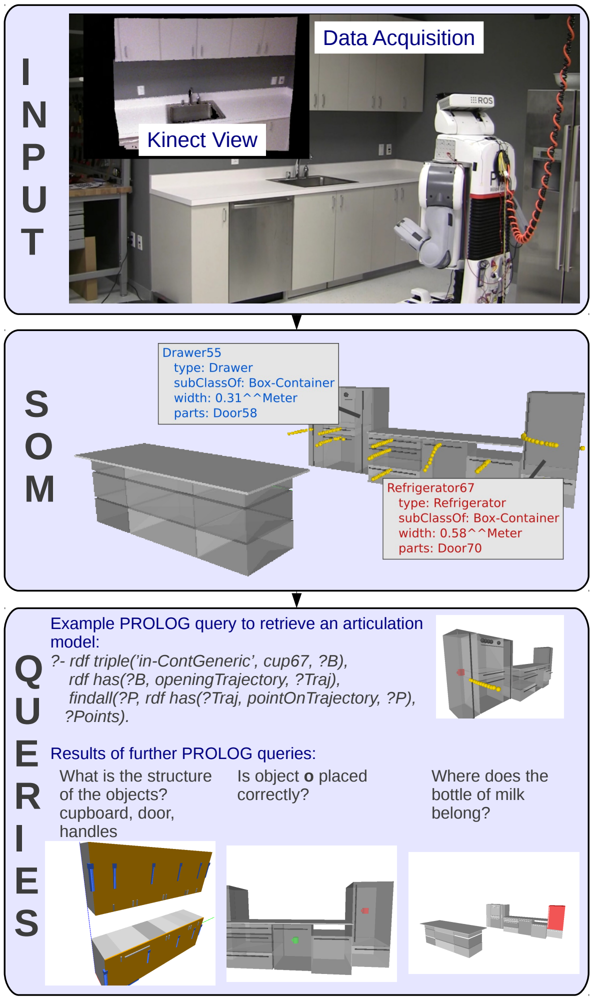
***Abstract*— In this article we investigate the representation and acquisition of Semantic Objects Maps (SOMs) that can serve as information resources for autonomous service robots performing everyday manipulation tasks in kitchen environments. These maps provide the robot with information about its operation environment that enable it to perform fetch and place tasks more efficiently and reliably. To this end, the semantic object maps can answer queries such as the following ones: “What do parts of the kitchen look like?”, “How can a container be opened and closed?”, “Where do objects of daily use belong?”, “What is inside of cupboards/drawers?”, etc.**

The semantic object maps presented in this article, which we call SOM$^+$, extend the first generation of SOMs presented by Rusu et al. [1] in that the representation of SOM$^+$ is designed more thoroughly and that SOM$^+$ also include knowledge about the appearance and articulation of furniture objects. Also, the acquisition methods for SOM$^+$ substantially advance those developed in [1] in that SOM$^+$ are acquired autonomously and with low-cost (Kinect) instead of very accurate (laser-based) 3D sensors. In addition, perception methods are more general and are demonstrated to work in different kitchen environments.

I. Introduction

Robots that do not know where objects are have to search for them. Robots that do not know how objects look have to guess whether they have fetched the right one. Robots that do not know the articulation models of drawers and cupboards have to open them very carefully in order to not damage them. Thus, robots should store and maintain knowledge about their environment that enables them to perform their tasks more reliably and efficiently. We call the collection of this knowledge the robot’s *maps* and consider maps to be models of the robot’s operation environment that serve as information resources for better task performance.

Robots build environment maps for many purposes. Most robot maps so far have been proposed for navigation. Robot maps for navigation enable robots to estimate their position in the environment, to check the reachability of the destination and to compute navigation plans. Depending on their purpose maps have to store different kinds of information in different forms. Maps might represent the occupancy of environment of 2D or 3D grid cells, they might contain landmarks or represent the topological structure of the environment. The maps might model objects of daily use, indoor, outdoor, underwater, extraterrestrial surfaces, and aerial environments.


INPUT
Data Acquisition
Kinect View
SOM
Drawer55
type: Drawer
subClassOf: Box-Container
width: 0.31^^Meter
parts: Door58
Refrigerator67
type: Refrigerator
subClassOf: Box-Container
width: 0.58^^Meter
parts: Door70
QUERIES
Example PROLOG query to retrieve an articulation model:
Results of further PROLOG queries:
What is the structure of the objects? cupboard, door, handles
Is object **o** placed correctly?
Where does the bottle of milk belong?

```
?- rdf triple('in-ContGeneric', cup67, ?B),
   rdf has(?B, openingTrajectory, ?Traj),
   findall(?P, rdf has(?Traj, pointOnTrajectory, ?P),
  ?Points).
```

Fig. 1. Building of a SOM$^+$ map in a kitchen environment (top), SOM$^+$ map representation (middle) and a set of robot queries made possible due to such powerful representation (bottom).

A research area that has received surprisingly little attention is the automatic acquisition of environment maps that enable robots to perform human-scale manipulation tasks, such as setting a table, preparing a meal, and cleaning up.

In our research we investigate *semantic object maps* (SOM$^+$s), which are a subcategory of maps that store infor-

[0] Both leading authors contributed equally to this work.

mation about the task-relevant objects in the environment, possibly including geometric 3D models of the objects, their position and orientation, their appearance, and object category. We focus here on semantic object maps that represent all the furniture entities of kitchen environments including cupboards, electrical devices, tables, counters, positions, appearances, and articulation models.

Overview of our system for the generation of SOM$^+$ maps is depicted in Fig. 1 where a PR2 robot acquires the data using an RGBD sensor in a kitchen environment (top), the resulting representation of an environment as a SOM$^+$ map is in the middle and a set of example queries that SOM$^+$ map provides to the robot is shown in the bottom. We can see that the SOM$^+$ map is an abstract representation of the environment that represents the environment as a hierarchically structured object where the parts themselves are objects that have a geometric 3D model, an appearance, and a 3D position and orientation. In addition, objects might have associated articulation models that tell the robot how they can be opened and closed, which is visualized by the yellow trajectories in the bottom part of the figure.

In this article we investigate and describe how SOM$^+$ maps can be represented and how the representations can be acquired autonomously.

In this context the key contributions of this article are:

- a functional end-to-end system that covers all steps required to automatically reconstruct textured SOM$^+$ models of kitchens, annotates them with the functional and semantic information and articulation models for opening and closing drawers and doors;
- methods for acquiring accurate environment maps with low-cost RGBD sensors by using vision and active manipulation actions such opening drawers and doors;
- a formal language based on the symbolic knowledge bases for the representation of SOM$^+$ maps;
- an application level with a rich set of task queries that the system can answer and thus enable the personal robot to carry out every day manipulation tasks.

We validate the concept of SOM$^+$ maps and the robot system for their acquisition in extensive experimental studies, in which the robots operate autonomously to acquire SOM$^+$ maps in 5 kitchens.

The remainder of the article is organized as follows. In Sec. II we introduce our representation language for SOM$^+$ maps and explain how the maps are organized and how different types of information are stored and handled. Sec. III will present the system integration by i) giving an overview of the SOM$^+$ map acquisition step and ii) discussing the data interpretation step. Sec. IV presents the empirical evaluation and explain example queries and put them to use. In the final two sections we will conclude and give an overview of the related work respectively.

II. Representation of SOM$^+$ maps

We represent SOM$^+$ maps as symbolic knowledge bases that contain facts about objects in the environment and that link objects to data structures such as appearance models or SIFT features which can be directly used by the robot's perception system to recognize the respective objects. Encoding maps into symbolic knowledge bases rather than lower-level data structures has two main advantages: First, it allows to have a uniform interface for querying for information, combining low-level information like the dimensions and poses of objects with semantic information about their properties. Second, this approach facilitates the integration of background knowledge from other sources like the WWW [2] or common-sense knowledge bases [3]. This enables the robot to apply this knowledge to reason about objects in the map, for example to infer problems that can occur when operating the dishwasher.

More formally, we consider a SOM$^+$ map to be a pair SOM$^+$ = $\langle \mathcal{SOM}^+\mathcal{KB}, \mathcal{C} \rangle$, where $\mathcal{SOM}^+\mathcal{KB}$ is the knowledge base representing the environment and $\mathcal{C}$ is a set of inference methods that can infer knowledge that is implied by the knowledge base but not directly stored. For example, $\mathcal{C}$ includes a method to infer whether two positions p_1 and p_2 satisfy the qualitative spatial relationship “on top of”.

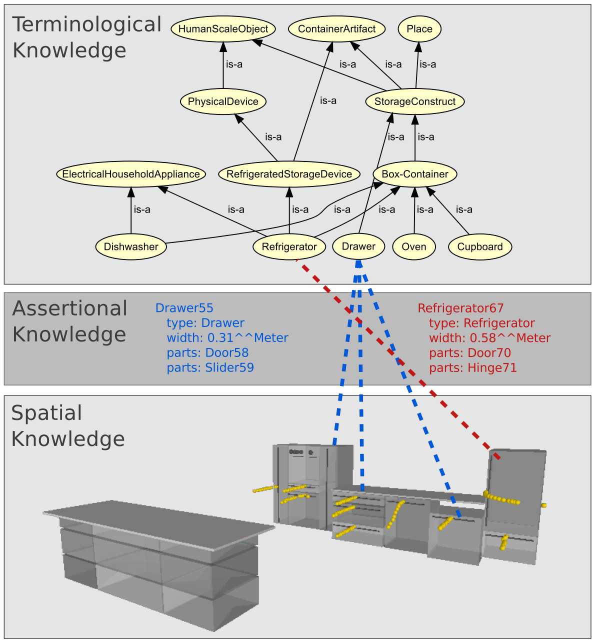


Fig. 2. Part of the ontology of household appliances and entities of furniture. Super-classes of e.g. *HumanScaleObject* have been omitted for better readability.

The knowledge base $\mathcal{SOM}^+\mathcal{KB}$ itself is formalized as a triple $\langle \mathcal{T}, \mathcal{A}, \mathcal{S} \rangle$ where $\mathcal{T}$ is a terminological knowledge base that specifies the categories of objects that are used to represent the environment. $\mathcal{A}$ denotes assertional knowledge, for example that *Refrigerator67* is a physical object in the environment and an instance of concept *Refrigerator*. Finally, $\mathcal{S}$ is spatial knowledge that asserts the pose of *Refrigerator67* in the environment. The different components of a SOM$^+$ knowledge base are depicted in Fig. 2.

The encyclopedic knowledge stored in $\mathcal{T}$ provides definitions of classes of objects and their properties, similar to those that can be found in a dictionary. It is very useful as a common vocabulary to describe the robot's knowledge. The different classes are arranged in a hierarchy, and are inter-connected by roles, a structure called an "ontology". A small part of this ontology, describing entities of furniture and household appliances, is shown in the upper part of Fig. 2. The major part of $\mathcal{T}$ is prior knowledge that is already available before the map has been built.

The objects in the semantic map are represented as instances of the semantic classes in $\mathcal{T}$ and form the assertional knowledge base $\mathcal{A}$. It contains information about their composition from parts, e.g. that *Refrigerator* consists of a box-shaped frame, a door, a hinge, and a handle, that the door is rotationally connected to the frame by the hinge, and that the handle is attached to the door. Each of these components is described as an instance of the respective semantic class with all of its properties, e.g. the information that a refrigerator is used as storage place for perishable items, or that an oven can be used for heating food. The elements of $\mathcal{A}$ are generated from the perception system and can be passed as arguments to the robot executive. They are thus grounded in both the perception and in the action execution system.

The spatial knowledge $\mathcal{S}$ includes both metric and qualitative spatial information about the poses of objects in the environment. Metric object poses are determined by the mapping procedure (Sec. III-B) and are stored in the knowledge base. Additional qualitative descriptions, like "on the table" can be computed as a different view on the data. These more abstract descriptions are not directly stored in the knowledge base, but computed at query. This approach helps to avoid inconsistencies due to duplicate data storage [4]. The computational methods are part of the set of inference procedures $\mathcal{C}$, which further includes methods to e.g. transparently convert units of measure (Sec. II-B).

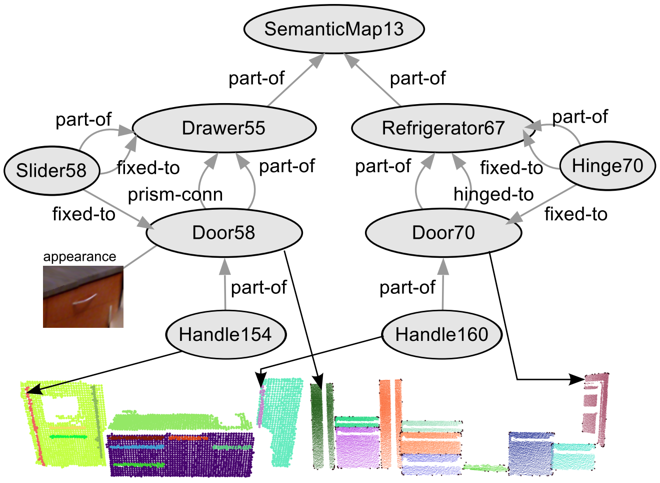


Fig. 3. Hierarchy of *part-of* relations between the different object components in the semantic map and a grounding example for doors and handles.

The SOM$^+$ map provides a *tell-ask*-interface to other components in the system. The *tell*-interface allows to add knowledge to the knowledge base and is mainly used by the mapping component: Whenever new objects are detected in the environment, they are added to the knowledge base. The *ask*-interface provides reasoning services to the robot's executive and to other components that require map information.

A. Object Representation in SOM$^+$ Maps

Most of the objects found in semantic maps of household environments are furniture entities and household appliances – which are complex, composed objects consisting of several parts (Fig. 3). Complementary to this part-of hierarchy, the connections between parts in terms of links and joints describe a kinematic chain. In the example, hinges are described as parts of the door, which is linked to the refrigerator's body with the *hingedTo* relation:

```
Individual: semanticmap14
  Types:
      SemanticEnvironmentMap

Individual: Refrigerator67
  Types:
      Refrigerator
  Facts:
      describedInMap semanticmap14
      width "0.51"^^Meter
      depth "0.59"^^Meter
      height "0.78"^^Meter
      properPhysicalParts Door70
      properPhysicalParts Hinge70

Individual: Door70
  Types:
      Door
  Facts:
      width "0.51"^^Meter
      depth "0.01"^^Meter
      height "0.78"^^Meter
      hingedTo Refrigerator67
      properPhysicalParts Handle160
```

An explicit description of the units of measure is important for the representation of spatial information in order to correctly interpret coordinate values. In the proposed representation, all numeric values can be annotated with the unit of measure that is being used. The units are described in the extensive QUDT ontology[1] including conversion factors. Compatible units, such as lengths, can be transparently converted into each other. For example, if the map contains dimensions and positions in meters, the user can query for information in feet and will automatically receive the converted values.

B. Spatio-temporal Object Pose Representation

The hierarchical representation introduced in the previous section qualitatively describes the composition of the environment out of objects and their parts, but does not specify their poses. We represent the pose information separately to account for object poses that change over time. Such a spatio-temporal representation is especially important for objects that are regularly moved, but it can also describe static objects as well as movable parts of static objects, such as the furniture doors.

We realized the spatio-temporal aspect by reifying the event that created some belief about an object pose, e.g. the detection of an object at some position. Instead of storing the information that an object "is at location A", the system thus describes that it "has been detected at location A at time T". This allows to store multiple detections of the object at different poses. In a naive implementation, attaching multiple poses to one object would lead to inconsistencies. The following code describes the detection of an object that is modeled in the knowledge base: An instance of a

[1] http://qudt.org/

SemanticMapPerception is created for each detection (*perception24*), and is annotated with the time at which the perception has been made (*timepoint24*), the pose at which the object was estimated to be (*pose24*), and the object instance that has been perceived (*Refrigerator67*).

```
Individual: perception24
  Types:
      SemanticMapPerception
  Facts:
      eventOccursAt  pose24
      objectActedOn  Refrigerator67
      startTime      timepoint24

Individual: pose24
  Types:
      Pose3D
  Facts:
      m00  "1.00"^^float
      m01  "0.00"^^float
```

By default, system determines the pose of an object based on the most recent detection, but if needed, it can also go back in time and reconstruct previous world states.

C. SOM$^+$ Inference Methods

Using the inference methods $\mathcal{C}$, the system can infer novel statements from the information in the map. Let us consider the computation of the "inside" relation as an example. If this relation holds can be calculated based on the poses and dimensions of two objects. Based on the spatio-temporal representation of object poses described in the previous section, such qualitative relations can be evaluated both for the current and for previous world states.

We use the *holds(rel(?A, ?B), ?T)* predicate to express that a relation *rel* between *?A* and *?B* is true at time *?T*. The following Prolog code computes the "inside" relation in a simplified way (not taking the rotation of the objects into account) by comparing the axis-aligned bounding boxes of the inner and outer object to check whether one contains the other. First, the latest perception of each object before time *?T* is determined using the *object_detection* predicate. The poses where objects have been perceived are read using the *eventOccursAt* relation. Then, the system reads the objects' positions and dimensions, and compares the bounding boxes.

```
holds(in_ContGeneric(?InnerObj, ?OuterObj), ?T) :-
    object_detection(?InnerObj, ?T, ?VPI),
    object_detection(?OuterObj, ?T, ?VPO),
    rdf_triple(eventOccursAt, ?VPI, ?InnerObjPose),
    rdf_triple(eventOccursAt, ?VPO, ?OuterObjPose),
    % read the center coordinates of the inner entity
    rdf_triple(poseX, ?InnerObjPose, ?IX), [...]
    % read the center coordinates of the outer entity
    rdf_triple(poseX, ?OuterObjPose, ?OX), [...]
    % read the dimensions of the outer entity
    rdf_has(?OuterObj, widthOfObject, ?OW), [...]
    % read the dimensions of the inner entity
    rdf_has(?InnerObj, widthOfObject, ?IW), [...]
    % compare bounding boxes
    >=((?IX - 0.5*?ID), (?OX - 0.5*?OD)),
    =<((?IX + 0.5*?ID), (?OX + 0.5*?OD)),
    [...]
    ?InnerObj \= ?OuterObj.
```

III. Acquisition of SOM$^+$ Maps

In our research we investigate domain specific map acquisition. This means that we make assumptions/assertions about the environments such as the existence of horizontal planar surfaces at table height, or the existence of front faces of furniture pieces that contain doors and drawers and let the interpretation algorithms use this prior knowledge to infer much richer environment models that contain all the furniture objects and structures introduced in Sec. III-B.

The version of our mapping system described in this paper makes a set of assumptions reasonable for typical kitchens, which include the following ones. (1) Kitchens have vertical planar walls (outmost boundaries) and kitchens have planar floors (the ground plane) and ceilings. (2) Front faces of furniture are vertical planes often in front of walls. Front faces of furniture's are typically rectangular and contain doors and drawers. Front faces of drawers and doors are parts of containers (typically used for placing objects inside). (3) Doors typically have handles and hinges, drawers have handles. Both can be opened (by pulling the handles). Some cupboards have tabletops that are planar surfaces in table height. Tables are tabletops standing on legs. (4) Some cupboards have special purposes: fridge, oven, microwave, dishwasher, etc. They are specializations of boxed containers. (5) Task-relevant objects are liftable and stand on table tops and shelves. (6) All others "object" structures are obstacles. Our system is currently specialized for the kitchenette parts of household environments, inclusion of other locations, semi-static (e.g. chairs) and dynamic objects we address in the separate papers [5] and will integrate them in the future work.

We will come back to the issue that these assumptions are of heuristic nature in Sec. V and outline how the next generation of the system is supposed to generalize from this overspecialization.

The SOM$^+$ mapping algorithms exploit these assumptions to better and faster process the raw sensor data through registration, plane fitting, etc and to generate and validate object hypotheses and infer better models of them.

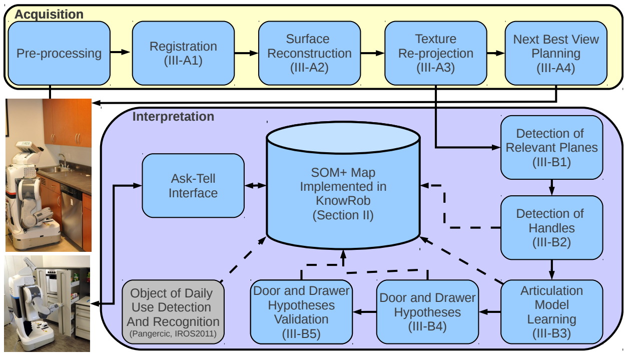

Acquisition
Pre-processing
Registration (III-A1)
Surface Reconstruction (III-A2)
Texture Re-projection (III-A3)
Next Best View Planning (III-A4)
Interpretation
Ask-Tell Interface
SOM+ Map Implemented in KnowRob (Section II)
Detection of Relevant Planes (III-B1)
Detection of Handles (III-B2)
Articulation Model Learning (III-B3)
Door and Drawer Hypotheses (III-B4)
Door and Drawer Hypotheses Validation (III-B5)
Object of Daily Use Detection And Recognition
(Pangercic, IROS2011)

Fig. 4. System integration described in Sec. III. Module for objects of daily use detection and recognition is part of the system but not discussed in this paper due to space constraints.

In order to acquire a SOM$^+$ map the robot has to explore and solve a number of perceptual tasks in order to obtain the necessary information pieces. The overall structure of the map acquisition process is illustrated in Fig. 4. The first phase in doing so is to obtain an accurate, smoothed and textured triangular 3D mesh of the environment where holes in the mesh are eliminated as much as possible (upper block in Fig. 4). The result of this phase is a mesh representation that on the one hand is much more compact than a point cloud representation and on the other hand forms the basis for the detection, categorization and recognition of furniture objects (lower block in Fig. 4). These two blocks will be further detailed in the following two subsections.

The SOM$^+$ mapping system is designed for autonomous manipulation platforms (but not limited to) that are equipped with a low-cost RGBD sensor on a pan-tilt basis (we use a Personal Robot 2 (PR2) with a head-mounted Kinect sensor).

A. Acquisition of the Basic Mesh Representation

The robot acquires an accumulated RGBD point cloud by exploring the environment and panning and tilting its head in order to cover the desired view frustum. The raw data are processed using a statistical noise removal kernel and then run through a Moving Least Squares module. These pre-processing steps enable a robust alignment of the point clouds and facilitate mesh reconstruction and texture reprojection.

1) Registration: To create a consistent and accurate 3D mesh model, the individual point clouds views are transformed into one common coordinate system and merged with each other. The merging step is performed through the geometric alignment of three-dimensional views using the estimated robot position as an initial guess using a variant of the *Iterative Closest Point (ICP)* algorithm [6]. Here we employ the more robust *point-to-plane* variant of ICP that uses a LevenbergMarquardt[2] algorithm to minimize distances between points in one point cloud to respective corresponding tangent planes in the other point cloud. To avoid the accumulation of registration errors over many scans, which could cause inconsistencies in the map, we globally optimize the registration in a second step using a graph optimization technique [8]. In the recent version of the system we also deployed a joint optimization method [9] in order to combine dense point cloud and visual feature data in one optimization step to calculate the transformations. This, so called RGBD-based registration, on the one hand enables us to perform mapping with the handheld camera (without a robot) and on the other hand generates maps with an accuracy under $1cm$ which is enough for the robot to reliably perform useful tasks such as opening the drawers (see the bottom row of Fig. 5.

2) Surface Reconstruction: To obtain a compact and fast-loading 3D model of the environment we use triangle meshes as our geometric and visual representation for SOM$^+$ maps. We apply a volumetric approach for reconstructing triangle meshes from the point clouds generated by the registration module. The first step of this approach calculates a 3D indicator function with positive values for points inside the model, and negative values for points outside. Kazhdan et al. [10] proposed an efficient way of calculating this indicator function on a regular grid constructed of smoothly overlapping volumetric field functions using a system of Poisson equations. The second step extracts the iso-surface of this indicator function by creating mesh vertices at zero-crossings along edges of grid cells [11]. The middle column in Fig. 5 shows the reconstructed triangle meshes of five kitchens. Each mesh consists of roughly 50K triangles while the raw point cloud is made up of more than 18M points.

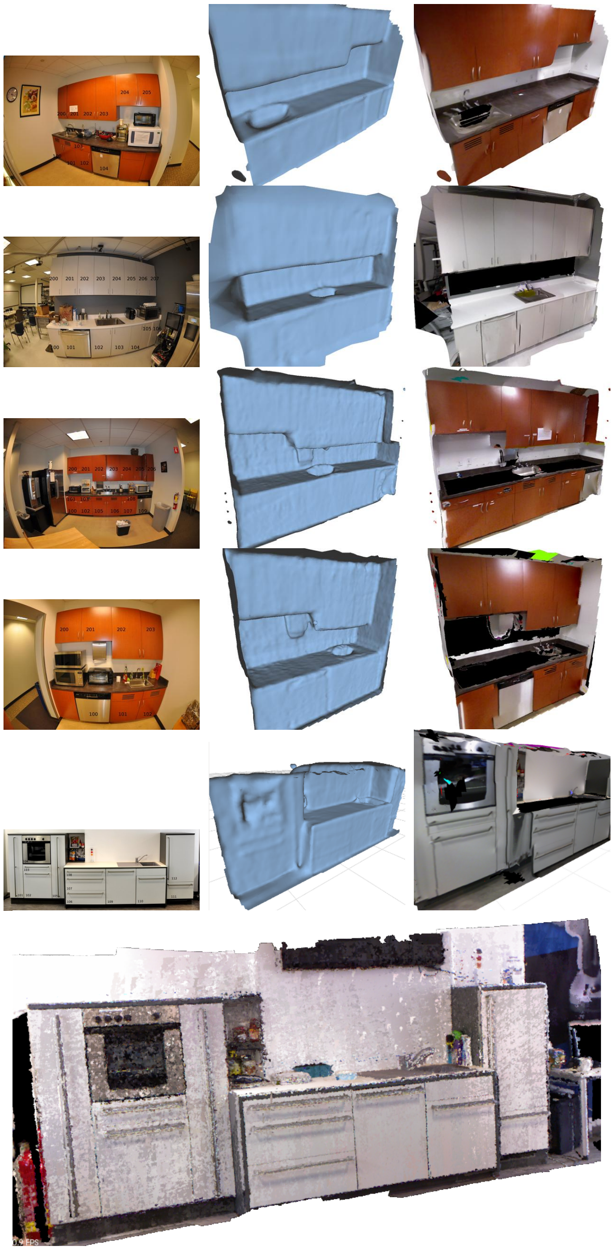


Fig. 5. **Left-column:** Testbed kitchens at TUM and Bosch RTC. **Middle-column:** Poisson-based surface reconstruction. **Right-column:** Blending-based texture re-projection on the left surface mesh. **Bottom-row:** Data of the TUM kitchen obtained with the handheld Kinect sensor and registered using the RGBD-based registration.

3) Texture Reconstruction: In general, the environments are made out of a variety of different materials which influence their appearance. Realistic reconstruction and reproduction of the surface appearance greatly enhances the visual impression by adding more realism and can thus be used for segmentation of surfaces, environment change detection, scene analysis or for object of daily use recognition. To achieve the texture reconstruction we capture color images together with point clouds. We use those images to reconstruct texture maps that are mapped onto the 3D mesh.

[2] Note that for the point-to-plane case, no closed-form solution is available which rules out the use of e.g. singular value decomposition method [7].

The first step of texture reconstruction computes a mapping for each mesh 3D vertex position into the 2D texture domain. In our system we use a least-squares method for finding the *conformal mapping* that minimizes distortions introduced by the 3D-2D mapping. When stitching multiple images into a texture, discontinuities on boundaries between images may become visible. For a consistent texturing we want to minimize the visibility of those undesired artifacts. Here we employ the blending technique proposed in [12] to globally adjust the color of all pixels simultaneously. The result is a texture composite without visual boundary artifacts. The right column in Fig. 5 presents the final texture mapped meshes.

4) Next Best View Planning: In this paper we focus on the SOM$^+$ maps of the kitchenette parts of indoor environments. Whole room data acquisition that requires next best view planning was presented in our earlier work [13] and is based on the information gain approach in which we use costmaps to find those poses that guarantee enough coverage of the unknown space as well as sufficient overlap with the already containing data for successful registration.

B. Interpretation of SOM$^+$Maps

1) Detection of Relevant Planes: Given the mesh generated by the texture reprojection module, our system first extracts relevant planes from it, categorizes them as walls, floor, ceiling, tables or other horizontal structures and doors or drawers. The latter is achieved by first locating the relevant planar structures, testing for the existence of handles and segmenting the doors and drawers first passively, and then actively through an interaction of the robot with the environment. As an exhaustive search for all planar structures is computationally intractable, we only search for those that are aligned with the walls of the room. The alignment of the latter is determined using a box fitting approach as proposed in [14]. 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.

2) Detection of Handles: We identified two types of handle appearances[3] that have different characteristics with respect to sensor data: handles that have specular reflection and the ones that do not. To tackle these two distinct cases we propose a two-fold approach that first tries to recognize and localize a handle in a 3D model of the given environment. Shall the latter fail we resolve to finding the handle in the parts of the 3D model that lacks range measurements due to the reflection of the sensor's projected infrared light pattern on specular surfaces [15]. We assert the handle's pose and dimension as SOM$^+$'s assertional knowledge according to Fig. 2. The example result of this handle detection is depicted in the bottom of Fig. 3. Erroneously detected handles could stem from specular flat surfaces or elongated objects (e.g. metal bars). In the first case the gripper opening distance after the grasp must be non zero and in the second case we abort opening motion if the robot exceeds an empirically determined force threshold.

3) Articulation Model Learning: To open the cabinets we use a controller developed by Sturm et al. [16]. The controller assumes that the robot has already successfully grasped the handle of an articulated object and that a suitable initial pulling direction is known. The robot then pulls in this direction using an equilibrium point control (EPC) and observes the resulting motion of its end effector. From this partial trajectory, it continuously (re-)estimates the kinematic model of the articulated object. The robot uses the kinematic model to predict the continuation of the trajectory. To deal with the workspace limits of the manipulator we make use of a secondary controller that moves the omni-directional base of the robot so that the reachable volume of the manipulator is maximized. After the motion of the end effector has come to a rest, the range of valid configurations of the articulated object is estimated. In sum, this gives us the full kinematic model of the articulated object. Finally, we sample the so-generated trajectory and store the poses of the sampled points on the trajectory as SOM$^+$'s spatial knowledge according to Fig. 2. An example of the model learning step is visualized in Fig. 6.

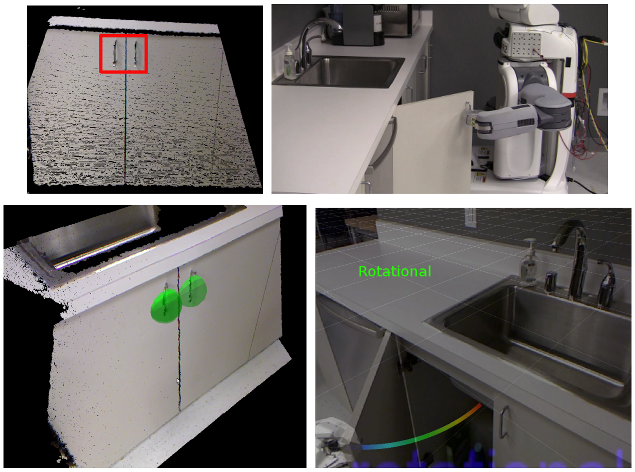


Fig. 6. The PR2 robot operates the cabinet in the Bosch RTC kitchen and learns the kinematic model (See also a video). Left column depicts a pair of doors with the handle with specularity (top) and a successful handle detection (bottom). Right column shows two snapshots from the opening sequence.

4) Generation of Door and Drawer Hypotheses: This module, initially proposed in [13], uses mesh vertices as seed points around footprint of handles to estimate an initial model of the color distribution of the door. The model consists of the intensity values' median $\tilde{i}$ and median average distance (MAD). The seed regions are expanded by adding neighboring vertices whose colors match the estimated color model, using a basic region growing algorithm based on the assumption that vertices on the door border are surrounded by vertices with different color. The color model for a region is updated after all possible vertices 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

[3] Please note that we only considered handles that correspond to the Americans with Disabilities Act http://www.ada.gov/pubs/ada.htm.

are added to the SOM$^+$ map, along with the rectangular approximations to the found planar segments.

5) Active Door and Drawer Hypotheses Validation through Interaction: Concurrently with the learning of the articulation models we also make use of the movement of the respective front of a cabinet and accept and reject the hypothesis generated in the previous subsection. To achieve this we use a temporal difference registration of two *point clouds* (of a closed and an open cabinet), using a search radius parameter of 0.5 cm, which is above the noise level of the sensor data[4] for the distances up to $1.5m$. We project the points that only appear in the second point cloud (corresponding to the door or the drawer $plane_{SEG}$) by applying the inverse of the transform between the first and the last pose of the stored opening trajectory. We then obtain the convex hull around such projected $plane_{PROJ}$, and assuming an environment based on rectangular furniture, we extract the width and the height of the cabinet front. For prismatic joints such as in case of drawers, we compute the distance between the two planes, which gives us the maximum opening distance and the depth of a drawer. For rotational joints, we assume that the depth of the cabinet is the same as the depth of the horizontal surface above it. We store poses and dimensions of cabinets as SOM$^+$'s assertional knowledge according to Fig. 2. Result of the final segmentation is shown in the bottom of Fig. 3.

IV. Experiments and Results

We evaluated the proposed integrated approach in five kitchens (see Fig. 5) by measuring the quality of the generated SOM$^+$ map in terms of the handle re-detection and the re-opening of the doors using learned and stored articulation models, and by measuring the average run times needed to generate one instance of SOM$^+$ map. In the accompanying video we also present a range of possible queries that our system can answer but are hard to evaluate quantitatively.

kitchen	#cabinets	#trials	#handle detection success	#opening success (w/o model)	#opening success (w model)
1	3	9	9 (100%)	8 (89%)	9 (100%)
2	5	15	15 (100%)	15 (100%)	15 (100%)
3	7	21	**18 (86%)**	19 (90%)	18 (100%)
4	1	3	3 (100%)	0 (0%)	3 (100%)
5	6	18	18 (100%)	14 (78%)	18 (100%)
Total:	22	66	63 (95%)	**56 (85%)**	**66(100%)**

TABLE I

Results of detecting the handles and opening the cabinets based on the information derived from the SOM$^+$ map.

A. Door Opening

In this experiment, we had the PR2 robot detect handles and three times open each of the 22 cabinets within five different kitchens (see Fig. 5). Due to the PR2's limited arm reach, we omitted the cabinets with handles located above $1.2m$ and the cabinets positioned in constrained spaces such as the ones adjacent to walls. The objective of the experiment was to asses the detection rate of handles given their apriori poses stored in the SOM$^+$ map, and to evaluate the robustness of a cabinet opening given their apriori learned and stored articulation models. The results of the experiment are presented in Table I. In column four we notice that the detection of the handle only failed three times. All failures occurred on a cabinet located next to the metal dishwasher that generated the invalid measurements which our handle detection algorithm took as a handle hypothesis. Column five presents the success rate of opening the cabinets without a priori learned model and column six with the a priori learned model. Playing back the stored trajectories turned out to be be 100% successful.

B. Performance Profiling

In Table II we broke down our processing pipeline into a set of independent components and profiled their performance on Intel Xeon W3520 desktop computer with 2.67GHz processor and 24GB of memory. Total time amounting to building of one SOM$^+$ map of one kitchenette from the scratch is $1.2h$ and the peak memory consumption of around 12GB incurred during the registration step. The latter can however be avoided through caching of point clouds to the disk. Querying times for the information stored in SOM$^+$ map are around $1s/query$.

$Component$	$Runtime$
Data acquisition and pre-processing	$0.1h$
Registration	$0.4h$
Surface reconstruction	$0.3h$
Texture re-projection	$0.3h$
Door opening and segmentation	$0.1h$
Generation of SOM$^+$map	$1s$
Total	$1.2h$

TABLE II

Execution times for all components in the processing pipeline (Fig. 4).

C. SOM$^+$ Example Queries

The bottom part of Fig. 1 and an accompanying video show different queries that can be answered by the SOM$^+$ map representation. Let us consider the following query as an example:

```
?- rdf_triple(knowrob:'in-ContGeneric',knowrob:'Cup67',B),
   rdf_has(B,knowrob:openingTrajectory,Traj),
   findall(P,rdf_has(Traj,knowrob:pointOnTrajectory,P),
           Points).
```

It reads the trajectory for opening the container where cups are stored in by first computing the 'in-ContGeneric' relation based on the poses and dimensions of the objects. For the resulting containers, it is checked whether there is an opening trajectory attached, and if that is the case, all points on this trajectory are returned. This query shows how the semantic map representation can translate qualitative abstract queries into information that can be used to parameterize the robot's actions such as the trajectory. In prior work, we showed

[4] http://www.ros.org/wiki/openni_kinect/kinect_accuracy

how different kinds of knowledge can be integrated with semantic maps, such as statistical relational information [4] or observations of human activities [17]. Please also consult the video `http://youtu.be/B7kMviETh50` for the whole range of queries and other system details.

V. Discussion

We presented an integrated systems paper for semantic mapping which enables the robot to build Semantic Object Maps with rich and powerful queries. We are aware that some of our perceptual heuristics do not fit (to e.g. old fashioned doors or doors without handles) and will in the future look into the ensemble of experts-based methods to alleviate that. Furthermore, we will also integrate algorithms for the recognition of beds, chairs, etc to scale towards mapping of whole apartments. Another avenue worth exploring to overcome the heuristic nature of the perceptual routines is to learn probabilistic models for the appearance of furniture entities. This however requires huge training data bases [5] and the scaling of probabilistic learning and reasoning.

VI. Related Work

Conceptually the closest to our work is the work done in project CoSy [18], where they adopted a multi-layered conceptual spatial model that consists of four maps: metric, navigation, topological and conceptual one. Their system, like ours, uses a SLAM algorithm to generate a metric map. Navigation and topological maps work hand in hand to classify the places into rooms of various types - a feature that our system could benefit from. Their conceptual map is endowed with a fully handcrafted commonsense OWL ontology of an indoor environment and is thus inferior to our ontology in KnowRob [17] which is one of the largest ontologies for service robots in the world. In addition their conceptual map currently does not allow for the storage of articulation models. Galindo et al. [19] present a multi-hierarchical approach where they connect the spatial and the conceptual hierarchies via anchoring. Similar to [18] and in contrast to our approach their map does not represent intra-room objects. Nüchter et al. [20] propose a semantic map system which up to the classification into floor, ceiling and walls is similar to ours. While they integrated object detection algorithms, on the other hand their representation language consists of Prolog facts only which is more limiting than our first-order knowledge representation based on description logics.

Acknowledgments

This work is supported in part by the EU FP7 Projects *RoboEarth* (grant number 248942) and *RoboHow* (grant number 288533).

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] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz, “Web-enabled Robots – Robots that Use the Web as an Information Resource,” *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 58–68, 2011.

[3] L. Kunze, M. Tenorth, and M. Beetz, “Putting People’s Common Sense into Knowledge Bases of Household Robots,” in *33rd Annual German Conference on Artificial Intelligence (KI 2010)*. Karlsruhe, Germany: Springer, September 21-24 2010, pp. 151–159.

[4] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, “KNOWROB-MAP – Knowledge-Linked Semantic Object Maps,” in *10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010, pp. 430–435.

[5] O. M. Mozos, Z. C. Marton, and M. Beetz, “Furniture Models Learned from the WWW – Using Web Catalogs to Locate and Categorize Unknown Furniture Pieces in 3D Laser Scans,” *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 22–32, June 2011.

[6] P. J. Besl and N. D. McKay, “A method for registration of 3D shapes.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 239–256, 1992.

[7] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[8] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, “Efficient estimation of accurate maximum likelihood maps in 3D,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3472–3478, 29 2007-Nov. 2 2007.

[9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments,” in *ISER 2010*, December 2010.

[10] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *4th Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, 2006, pp. 61–70.

[11] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1987, pp. 163–169.

[12] B. Pitzer, S. Kammel, C. DuHadway, and J. Becker, “Automatic reconstruction of textured 3d models,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3486–3493.

[13] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz, “Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September, 25–30 2011.

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

[15] T. Rühr, J. Sturm, D. Pangercic, D. Cremers, and M. Beetz, “A generalized framework for opening doors and drawers in kitchen environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.

[16] J. Sturm, C. Stachniss, and W. Burgard, “Learning kinematic models for articulated objects,” *Journal on Artificial Intelligence Research (JAIR)*, 2011.

[17] M. Tenorth and M. Beetz, “KnowRob – Knowledge Processing for Autonomous Personal Robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4261–4266.

[18] A. Pronobis, P. Jensfelt, K. Sjöö, H. Zender, G.-J. M. Kruijff, O. M. Mozos, and W. Burgard, “Semantic modelling of space,” in *Cognitive Systems*, ser. Cognitive Systems Monographs, H. I. Christensen, G.-J. M. Kruijff, and J. L. Wyatt, Eds. Springer Berlin Heidelberg, 2010, vol. 8, pp. 165–221. [Online]. Available: http://www.pronobis.pro/publications/pronobis2010cogsys

[19] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González, “Multi-hierarchical semantic maps for mobile robotics,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, CA, 2005, pp. 3492–3497.

[20] A. Nüchter and J. Hertzberg, “Towards semantic maps for mobile robots,” *Robot. Auton. Syst.*, vol. 56, no. 11, Nov. 2008.