Model-Based Object Recognition from 3D Laser Data

Martin Günther, Thomas Wiemann, Sven Albrecht, and Joachim Hertzberg

University of Osnabrück, 49069 Osnabrück, Germany

Abstract. This paper presents a method for recognizing objects in 3D point clouds. Based on a structural model of these objects, we generate hypotheses for the location and 6DoF pose of these models and verify them by matching a CAD model of the object into the point cloud. Our method only needs a CAD model of each object class; no previous training is required.

Keywords: object recognition, 3D point clouds, OWL-DL ontology, CAD model matching, semantic mapping

1 Introduction

Learning maps of a previously unknown environment is a classical task for mobile robots. The large body of literature about the topic would address it mostly as Robotic Mapping or as Simultaneous Localization and Mapping (SLAM). Traditionally, robot maps contain landmarks and/or environment structures and the according geometry information in 2D or 3D only, as they are mostly used for planning collision-free paths and for localization. In more recent work, semantic categories in maps are becoming more and more important: These may be, for example, classes of objects contained in the map, classes of rooms, or topological regions. Semantic information in maps promises to be of help for both the robot control itself and as a basis for human-robot or robot-robot interaction.

A technical challenge for semantic mapping approaches is to extract the semantic categories from the raw sensor data, and to do so in real time on board the robot. Most previous object recognition approaches are appearance-based: Based on a 2D image or 3D point cloud of the scene, features are extracted and passed into a classifier that has previously been trained using labeled training examples.

However, in recent years, CAD models of many types of objects have become widely available. From the CAD model of an object, a declarative, structural model of the object in terms of its primitive geometric constituents (planar patches, cylinders, spheres) and their spatial interrelations can be obtained. This offers the possibility of a complementary, model-based approach to object recognition: Instead of classifying objects by their appearance, we extract the geometric primitives found in the raw sensor data and match them to the structural models of our objects.

Our approach extends and goes beyond previous work [10], where coarse structures like walls, doors, ceiling and floor were labeled. Here, we take this idea one step further and extend it to the recognition of medium-scale objects. We describe a system that recognizes different types of furniture in 3D laser scans and present first empirical results.

2 Related Work

Previous work on model-based object recognition in 3D point clouds has been carried out by Rusu et al. [15, 16]. They present a complete semantic mapping framework based on 3D laser data; in contrast to our work, the model fitting part uses cuboids of variable dimensions instead of CAD models.

Most other work in object-recognition is appearance-based (in the sense that no structural model of the object is used, although some use CAD models). Lai and Fox [6] use sampled CAD models from Google 3D Warehouse to train an object detection system used to label objects in urban 3D laser scans. Mian et al. [9] use a geometric hashing scheme to recognize objects by matching CAD models into a 3D point cloud.

Several methods [18, 14, 20] for extracting interest points and creating stable descriptors based on 3D shape have been proposed which can be used to recognize objects in 3D point clouds. Stiene et al. [19] detect objects in range images using an Eigen-CSS method on the contours of objects, combined with a supervised learning algorithm. In a similar setting, Nüchter et al. [11] use Haar-like features together with AdaBoost for object detection.

An idea similar in spirit to our approach is proposed in [5] where matching of CAD models is utilized to allow for manipulation tasks such as grasping in a household environment. Contrary to our approach, the matching itself is performed within 2D image data, instead of the 3D environment representation.

A robot manipulation system that integrates knowledge processing mechanisms with semantic perception routines, including CAD model matching, is presented in [12].

Part of our method requires the use of spatial relational reasoning, which is often done using constraint calculi [13]. Our task, however, differs from that, because we consider the spatial relations between geometric primitives as part of the definition of an aggregated object. For this reason, we use ontological reasoning for this task. Another advantage of this approach is that numerical ranges can be used via SWRL rules.

3 Model-Based Object Recognition

In recent years, CAD models of many kinds of objects have become widely available. One resource of CAD models is Google's 3D Warehouse, which allows querying and retrieving CAD models of virtually any kind of object via the web. In the domain of furniture recognition, CAD models are often available directly from the manufacturer or from companies specialized in creating CAD models for interior designers. We use a database of CAD models supplied by our university's furniture manufacturer.

In this paper, we focus on the domain of furniture recognition for several reasons: First, due to the widespread use of CAD models in interior design, the availability of CAD models in this domain is especially strong. Second, most kinds of furniture feature a set of planar surfaces which can be robustly recognized in 3D laser scans. Third, due to the rigidness of furniture, these planar surfaces are in a clearly defined relation to each other.

Figure 1 shows the embedding of our system in a general semantic mapping framework. We see our model-based object recognition method as complementary to appearance-based methods based on 2D image features or 3D shape features.

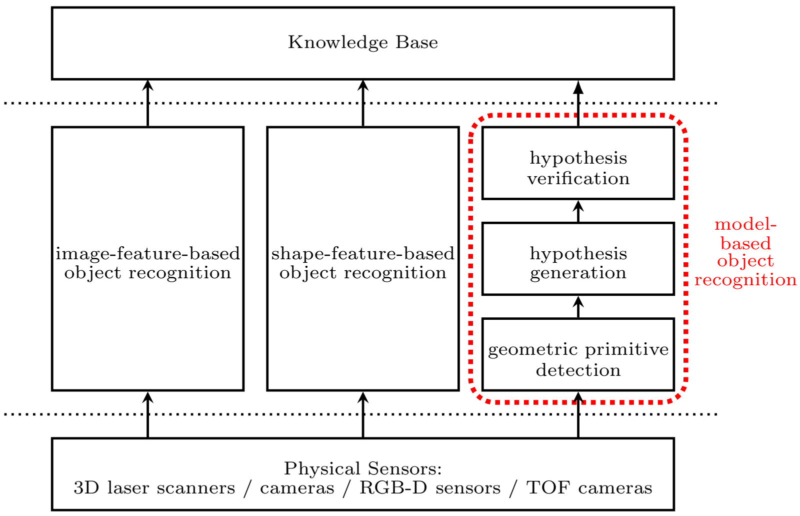


Fig. 1: System overview. While the present paper is focused on model-based object recognition, we consider this method as yielding complementary information to standard recognition methods. So in a more general system architecture, they may well co-exist. We will not deepen this issue here.

Using the information contained in CAD models for object recognition has several advantages. Instead of having one classifier for each kind of object, only the geometric primitives have to be detected. Based on these, objects are reconstructed. Also, no classificator training and no labeled training sets are required; to add a new object class, only the CAD model is required. In the future, it would even be conceivable that such a CAD model could be retrieved on-line from the web. Another advantage is that once an object is recognized, the corresponding part of the sensor data can be replaced by the CAD model, thus filling up occlusions in the sensor data.

On the other hand, appearance-based methods have an advantage where the to-be-recognized object is non-rigid, does not consist of clearly identifiable geometric primitives of a certain minimum size or where labeled training data, but no CAD model is available.

Our model-based object recognition method consists of three parts, which will be detailed in the remainder of this section: geometric primitive detection, which extracts the geometric primitives from the sensor data; object hypothesis generation, which classifies the geometric primitives to find potential object locations; and object hypothesis verification, where these potential matches are accepted or rejected.

3.1 Geometric Primitive Detection

The first step of the object recognition procedure is to extract geometric primitives from the 3D point cloud. In our approach we rely on planar patches, since they are most relevant for detecting furniture. In the remainder of this paper we will refer to these patches simply as "planes". However, our approach could be extended to other kinds of geometric primitives, such as cylinders or spheres.

The idea of our plane extraction algorithm is the following: First, a triangle mesh of the scanned scene using an optimized marching cubes [7] implementation is generated. Within this mesh, connected planar regions are extracted using a region growing approach. A detailed description of our procedure can be found in [21]. Figure 2 shows two exemplary results of this procedure: The left picture shows a reconstruction of a closable office shelf with one side open. The right picture shows a filing cabinet. Both reconstructions are based on 3D laser scans taken with a tilting SICK LMS 200 laser scanner (about 156.000 data points per scan). The extracted planes are shown in different colors. Non-planar regions in the mesh that were not considered by or algorithm are rendered in green. As one can see, the relevant planes for our purposes can clearly be distinguished. For each extracted plane, we save the characteristic information that is needed for the OWL-DL reasoner to generate object hypotheses. The relevant key figures for our process are size of the plane, centroid and surface normal.

3.2 Object Hypothesis Generation

Semantic knowledge about identified objects is stored using an OWL-DL ontology in combination with SWRL rules, which will be used to generate hypotheses of possible object locations and initial pose estimation, based on the planes extracted in the previous section.

OWL, the Web Ontology Language, is the standard proposed by the W3C consortium as the knowledge representation formalism for the Semantic Web. It consists of three sublanguages (OWL-Full, OWL-DL and OWL-Lite). The sublanguage OWL-DL [3] corresponds to a Description Logic, a subset of First-Order Logic, allowing to use many expressive features while guaranteeing decidability

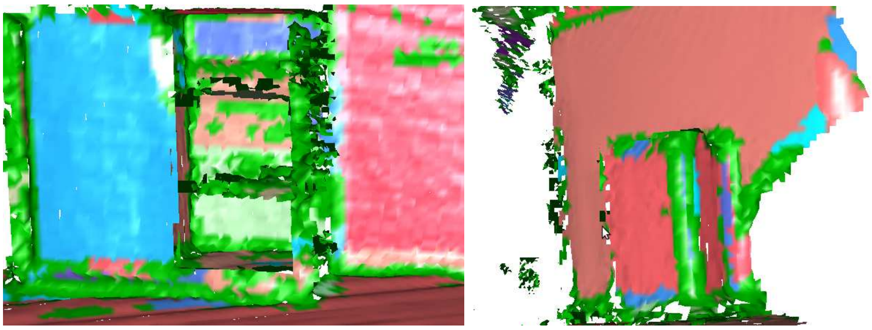


Fig. 2: Two examples for plane extraction. Detected planes are colored individually; non-planar surfaces are colored in green. Left: closable shelf (sliding doors) in front view. Right: a filing cabinet. The pictures show that our plane extraction algorithm is able to extract the relevant geometric information for our model based object recognition procedure.

(in polynomial time). It has been extended by SWRL, the Semantic Web Rule Language [4], which allows to use Horn-like in combinations with an OWL-DL knowledge base and includes so-called built-ins for arithmetic comparisons and calculations. OWL-DL was chosen as the knowledge representation format for this paper for several reasons: OWL-DL ontologies can be easily re-used and linked with other sources of domain knowledge from the Semantic Web, they easily scale to arbitrarily large knowledge bases, and fast reasoning support is available. In our implementation, we use the open-source OWL-DL reasoner Pellet [17], which provides full support for OWL-DL ontologies using SWRL rules.

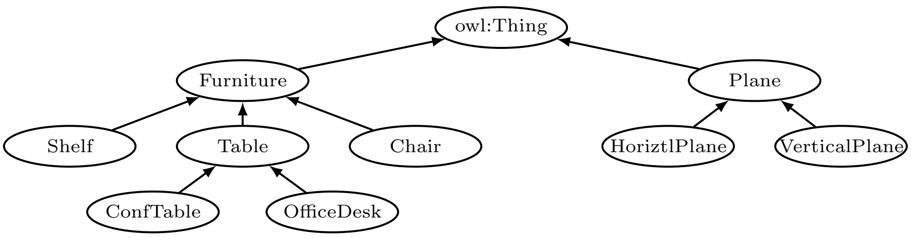


Fig. 3: The parts of the ontology's class hierarchy relevant for the examples in this paper. For our approach, we distinguish between horizontal and vertical planes (right branch). The relations between different kinds of furniture are modeled in the left branch.

Figure 3 shows part of the ontology used in this paper. The right part models the geometric primitives (here: horizontal and vertical planes). Each plane extracted in the previous section is added as an individual to the ontology, along with its orientation (horizontal/vertical, based on the normal), its height above ground (based on the centroid), its bounding box and its area. Additionally, we add two different spatial relations between planes, based on the centroid and surface normal which have been extracted in the previous subsection, as OWL properties. The property *isAbove* is added between two horizontal planes if the distance of their centroids, projected onto the ground plane, is below a certain threshold. Likewise, the property *isPerpendicular* is added between a horizontal and a vertical plane if their centroid distance is below the threshold.

The definitions of furniture classes in the ontology (the left part of Fig. 3) contain a set of conditions that are used to classify the planes into possible furniture instances. For example, most standard desks have a height of approximately 70 cm. So all horizontal planes that have about this height and have a certain minimal size are valid candidates for table tops. This can be expressed by adding the following SWRL rule to the ontology:

$$\begin{aligned} Table(?p) \leftarrow\ & HorizontalPlane(?p) \wedge hasSize(?p, ?s) \\ & \wedge\, swrlb : greaterThan(?s, 1.0) \wedge hasPosY(?p, ?h) \\ & \wedge\, swrlb : greaterThan(?h, 0.65) \wedge swrlb : lessThan(?h, 0.85) \end{aligned}$$

Similar considerations apply to office chairs: A chair has a ground parallel plane to sit on (at a height of around 40 cm) and another perpendicular plane near it (the backrest). Figure 4 presents, as an example, the ontology representation of a shelf. The ranges in the properties reflect possible reconstruction errors or modifications of the actual object that were not represented in the original CAD model. At the moment, these structural object models are encoded into OWL-DL by hand; in the future, these could be extracted automatically from the CAD model or generalized from a set of CAD models.

For each object instance returned by the OWL-DL reasoner, we calculate axis-parallel bounding boxes and center points of the constituting planes. The center point of one predefined plane (e. g., the table top) is used to anchor the position. Information about the orientation depends on the geometry of the expected models. The intrinsic orientation has to be identified and encoded according to the model class. For some objects this orientation is easy to identify, e. g., chairs where the normal of the back rest defines the orientation of the whole object. For other objects like tables, we apply a Principal Component Analysis (PCA) to the points that lie within the plane that defines the intrinsic orientation. This method delivers two new orthonormal basis vectors that approximate the orientation within the global coordinate system. For successful matching, all used models have to be pre-processed to be in a center-point-based local coordinate system that reflects the assumptions described above.

3.3 Object Hypothesis Verification

In order to verify the generated hypotheses of Section 3.2 we match an appropriate CAD model for each hypothesis with the point cloud data. Creating a 3D point surface sampling for a given CAD model yields a point cloud retaining

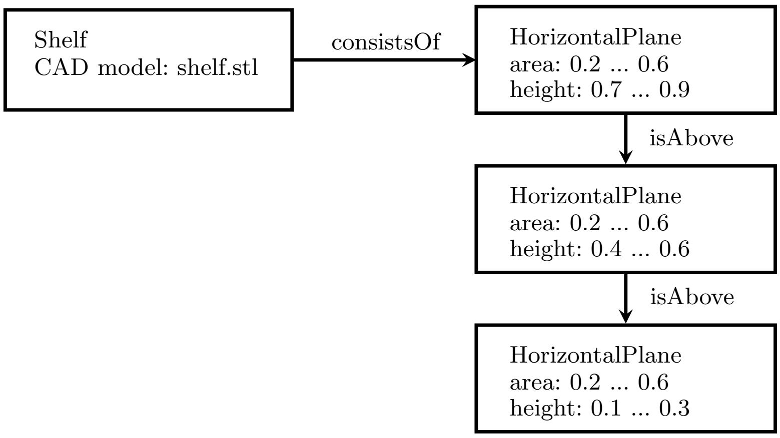


Fig. 4: A fragment of the ontology representing a shelf. The range intervals for the different properties take account of user modifications (someone could have lifted a shelf board) and reconstruction errors.

the geometric properties of the model. This point cloud in combination with the initial pose obtained in Section 3.2 allows for a straight forward application of the ICP [2] algorithm. For more details concerning the surface sampling please refer to [1]. Since ICP converges in a local minimum, the average correspondence error between sampled model and scene point cloud can already give a rough estimate about the quality of the match, i. e., if a hypothesis could be verified or not.

Although this approach might correctly reject some false hypotheses, many scenarios are plausible in which the average correspondence error between sampled CAD model and data point cloud is small but still the object is not present in the scene. To have a better estimate to determine if an object is present in the data points we therefore propose another heuristic measure. For this heuristic we change the perspective compared to the ICP algorithm used to obtain the final pose: Instead of fitting the sampled model to the point cloud data we now check how closely the point cloud data resembles the sampled model data in the model's final pose. To this end we discretize the model data into voxels, thus creating a more coarse representation of the model. Now we check how many data points of the scan data are contained in each of these voxels in the final pose determined by ICP. If the number of data points in such a voxel is larger than a given threshold, we assume that this part of the model was present in the scan data, otherwise the point cloud locally does not fit. Once this process is done for each voxel, we compare the ratio of voxels resembled in the scan to voxels not present in the scan data. If this ratio is above a given threshold we assume that the model was present in the scan data.

However a more sophisticated measure to compare the CAD model in its final pose with its surroundings is topic of ongoing research.

4 Experimental Results

For our test scenario, we use a database of CAD models that is directly available from our university's furniture manufacturer. We present an example for automatically recognizing furniture using our model-based object recognition method: Finding two office table CAD models in a 3D point cloud of an office. The first part displays the instantiation of the object hypotheses from our hypothesis generation method. The second part shows pose refinement for these instances derived from the ICP-based hypothesis verification step.

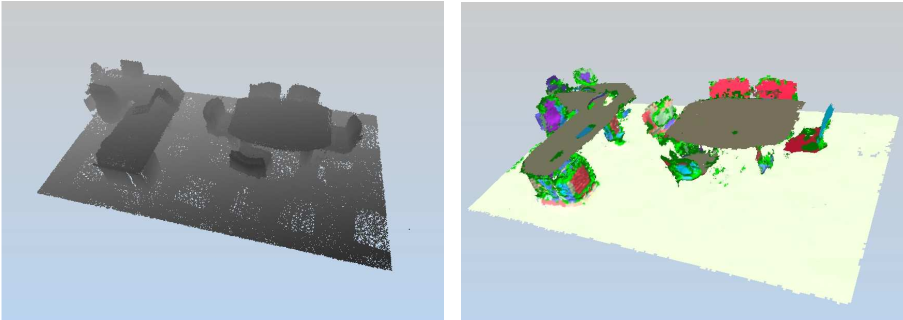
4.1 Hypothesis Generation

The input to our reasoner is a set of planes that were extracted from the input data. Figure 5 shows the input point set for our experiments together with a surface reconstruction. The data was obtained using a SICK LMS-200 laser scanner mounted on a rotational unit. Several scans from different positions were registered via ICP into a single point cloud. The planar structures found in the scene are rendered in a red to blue gradient, all other surfaces are green. The basic characteristics of these patches (centroid, normal, bounding box, area) are used by the reasoner to identify possibly present models. The current implementation of our plane extraction procedure is highly optimized for parallel processing and scales well with the number of CPU cores. The objects in the presented data set are extracted in less than 4 seconds on a Intel Quad Core processor, including normal estimation for the data points, mesh generation and plane extraction. This time is in the order of magnitude that it takes to capture a single 3D laser scan with our equipment.

As one can see, the large connected surfaces on the floor are recognized, as well as smaller structures like the tabletops (gray) or the backrests of the chairs around the conference table (red, blue, light green). After feeding the extracted planes into the reasoner, two possible present objects were detected: The conference table on the right and the desk on the left. The main remaining problem is to determine the model's orientation. To solve this, we use a PCA implementation by Martagh [8] on the vertices of the table top reconstruction. To analyze the stability of this approach, we rotated a reference model of the conference table to different predefined angles to get ground truth and compared the original rotation angles with the PCA estimation. The results are shown in Table 1. The time for PCA computation for the considered planes is negligible for our application (some 100 milliseconds). Although the estimated poses derived from the bounding box show several degrees difference from ground truth, we were able to correct these deviations automatically via ICP and confirm the detected objects in the scanned scene as shown in the following section.

4.2 Hypothesis Verification and Model Replacement

After the object hypotheses and pose estimations are generated, we subsample the corresponding CAD models. This synthetic point cloud is then used to refine



(a) (b)

Fig. 5: The used input point cloud (a) and the automatically extracted planes (b). The detected planes are colored in a red to blue gradient based on a running number. All surfaces that were not classified as belonging to a plane are rendered in green. Walls and ceiling in the original data set were manually removed to create a suitable perspective.

Table 1: Estimated orientations for two table models via PCA compared to ground truth.

Ground Truth	12.0°	25.0°	55.0°	90.0°	125.0°	160.0°
Conference Table	8.2°	22.1°	51.4°	86.0°	121.0°	157.0°
Office Desk	4.0°	28.1°	46.7°	82.0°	118.0°	153.0°

the initial pose estimation. Figure 6 shows the results of the matching process for the tables that were detected in the given scene. The pictures on the left clearly show that the ICP process significantly improves the estimated poses. For the conference table we get an almost perfect fit. The fit for the office desk is not as good as the one for of the conference table. Here we have an offset to the right of about two centimeters. This is due to registration errors in the used point cloud and differences between the CAD model and the real world object. The real object shows clearances that are not considered in the model.

These two examples show that our ICP based object hypotheses verification procedure is able to instantiate the presumed objects from the OWL-DL reasoner. This instantiation provides additional semantic knowledge about the scanned environment, namely that there are an office table and a conference table present. Furthermore, the replacement of the original point cloud data with the appropriate CAD models of the recognized objects can be used to enhance the initial sensor data, e. g. by filling in missing data points from laser shadows by sampling the surfaces of the CAD model.

This fact shows another advantage of model based object recognition over appearance based methods. These methods usually do not encode the whole geometric information of the trained objects, only the abstract characteristic

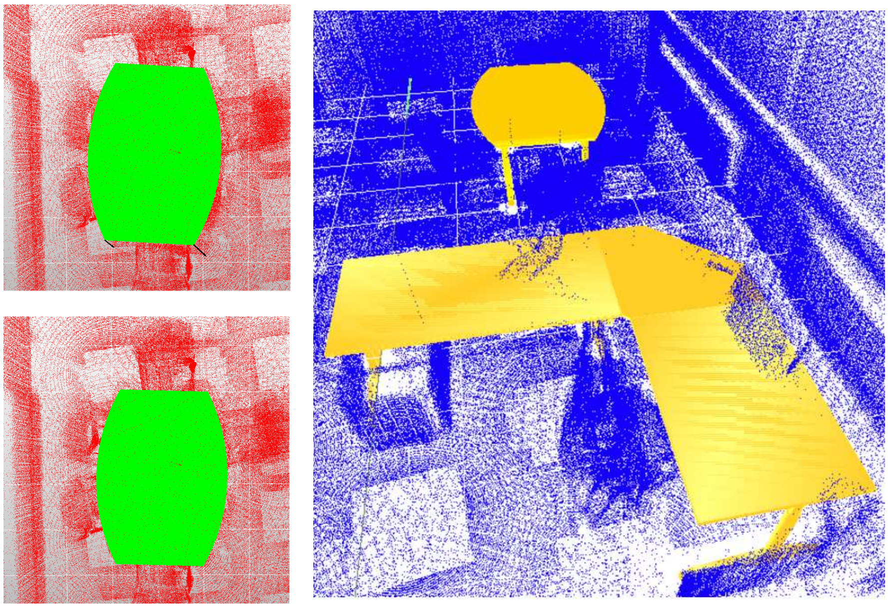


Fig. 6: Results of the ICP model matching process for the two table models. The left column shows the pose of the conference table before and after matching. The offset of the initial pose estimation from the final pose is indicated by the black arrows at the lower edge of the table. The picture on the right shows the CAD models of both detected tables rendered in the original point cloud.

feature descriptors for the used machine learning algorithm. Although a link between these features and more detailed object descriptions is feasible, the handling of these relations requires additional efforts. With our model based approach we have the geometric properties already encoded in the CAD models themselves.

5 Summary and Future Work

We have presented initial results on a model-based method for recognizing furniture objects in 3D point clouds, based on CAD models, and demonstrated the viability of this approach on a real-world example.

In the future, we plan to extend this approach in several directions: First, we plan to explore alternative representation formalisms for the object hypothesis generation step. In particular, Statistical Relational Models offer themselves here due to their potential robustness to occlusions or false positives in the sensor data. Second, more work needs to be done on the error function of the hypothesis verification step. Third, we intend to expand the approach to articulated furniture (such as a cabinet with sliding or hinged doors, chairs and tables with adjustable height) and variability (such as a bookshelf with a variable number of shelves). One way to do this would be to use parametric CAD models. Fourth and last, we plan to automate the extraction of OWL-DL structural models for hypothesis generation from the CAD models.

References

1. Albrecht, S., Wiemann, T., Günther, M., Hertzberg, J.: Matching CAD object models in semantic mapping. In: Proc. ICRA 2011 workshop: Semantic Perception, Mapping and Exploration, SPME '11. Shanghai, China (2011)
2. Besl, P., McKay, N.: A method for registration of 3–D shapes. IEEE T. Pattern Anal. Mach. Intell. 14(2), 239–256 (February 1992)
3. Dean, M., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL web ontology language reference. W3C recommendation, W3C (February 2004), http://www.w3.org/TR/owl-ref/
4. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3C member submission, World Wide Web Consortium (2004), http://www.w3.org/Submission/SWRL
5. Klank, U., Pangercic, D., Rusu, R.B., Beetz, M.: Real-time cad model matching for mobile manipulation and grasping. In: 9th IEEE-RAS Intl. Conf. on Humanoid Robots. Paris, France (December 7-10 2009)
6. Lai, K., Fox, D.: Object recognition in 3D point clouds using web data and domain adaptation. Int. J. Robot. Res. 29(8), 1019–1037 (July 2010)
7. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: Proc. ACM SIGGRAPH (1987)
8. Martagh, F., Heck, A.: Multivariate Data Analysis. Kluwer Academic (1987)
9. Mian, A.S., Bennamoun, M., Owens, R.A.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE T. Pattern Anal. Mach. Intell. 28(10), 1584–1601 (2006)
10. Nüchter, A., Hertzberg, J.: Towards semantic maps for mobile robots. Robot. Auton. Syst., Special Issue on Semantic Knowledge in Robotics 56(11), 915–926 (2008)
11. Nüchter, A., Surmann, H., Hertzberg, J.: Automatic classification of objects in 3D laser range scans. In: Proc. 8th Conf. on Intelligent Autonomous Systems (IAS 2004). pp. 963–970. Amsterdam, The Netherlands (March 2004)
12. Pangercic, D., Tenorth, M., Jain, D., Beetz, M.: Combining perception and knowledge processing for everyday manipulation. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems. Taipei, Taiwan (October 18-22 2010)
13. Renz, J., Nebel, B.: Qualitative spatial reasoning using constraint calculi. In: Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.) Handbook of Spatial Logics, pp. 161–215. Springer (2007)
14. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Intl. Conf. on Robotics and Automation (ICRA). pp. 3212–3217. IEEE, Kobe, Japan (2009)
15. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M.E., Beetz, M.: Towards 3D point cloud based object maps for household environments. Robot. Auton. Syst., Special Issue on Semantic Knowledge in Robotics 56(11), 927–941 (2008)
16. Rusu, R.B., Marton, Z.C., Blodow, N., Holzbach, A., Beetz, M.: Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments.

In: 2009 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, October 11-15, 2009. pp. 3601–3608. IEEE, St. Louis, USA (2009)
17. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. 5(2), 51–53 (2007)
18. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Point feature extraction on 3D range scans taking into account object boundaries. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). Shanghai, China (May 2011)
19. Stiene, S., Lingemann, K., Nüchter, A., Hertzberg, J.: Contour-based object detection in range images. In: Proc. 3rd Intl. Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT '06. Chapel Hill, NC, USA (June 2006)
20. Unnikrishnan, R.: Statistical Approaches to Multi-Scale Point Cloud Processing. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (May 2008)
21. Wiemann, T., Nüchter, A., Lingemann, K., Stiene, S., Hertzberg, J.: Automatic construction of polygonal maps from point cloud data. In: Proc. 8th IEEE Intl. Workshop on Safety, Security, and Rescue Robotics (SSRR-2010). Bremen, Germany (July 2010)