A Semantic Knowledge Base for Cognitive Robotics Manipulation

Elisa Tosello*, Zhengjie Fan† and Enrico Pagello‡

*‡Intelligent Autonomous Systems Lab (IAS-Lab)

Department of Information Engineering (DEI), University of Padova

Via Gradenigo 6/B, 35131 Padova, Italy

Email: *toselloe@dei.unipd.it, ‡epv@dei.unipd.it

†Big Data Team

Department of Big Data and IT Technology, China Mobile Research Institute (CMRI)

Innovation Building, No. 32, Xuanwumen West Avenue, Xicheng District, Beijing 100053, P.R. China

Email: fanzhengjie@chinamobile.com

***Abstract*—This paper presents a Semantic Knowledge Base that can be adopted as an information resource for autonomous cognitive robots performing manipulation tasks. Robots can use this database to save and share descriptions of learned tasks, detected objects, and explored environments. The Semantic Knowledge Base becomes: I) a data store for efficient and reliable execution of repeated tasks; II) a web-based repository for information exchange among robots.**

Compared to existing work the database does not only store the 3D CAD model of each object, it stores also its function and 3D geometric shape. A set of suitable manipulation poses is generated and stored according to the shape and the type of gripper. In this way, manipulation can be performed also on objects seen for the first time or for which only an incomplete 3D CAD model exists.

In this paper, we describe the ontology and the language adopted to define the knowledge base and to query it, together with the novel approach employed to solve the data sets interlinking problem related to the rescue and recovery of duplicate data. A sense-model-act framework is implemented to test the manipulation of an object which shape is in the database.

Keywords: Semantic Web, Cognitive Robotics, Manipulation, ROS

I. Introduction

One of today's robotics challenges is to improve robots' versatility: robots should be able to perceive and independently, safely and timely adapt to the constantly changing surroundings. The other challenge is to enhance efficiency: in order to intelligently act, robots should be able to learn from past experiences, errors, and actions performed by other agents (either humans or robots). These capabilities cannot arise from precompiled software routines. Robots must be able to reason like humans. For this purpose, a significant research program exists and is known under the name of *cognitive robotics*.

Over the years, cognitive robotics has been well investigated with regard to social robots. An example is the humanoid robot platform iCub [1]. With the advent of Industrie 4.0 [2], the field begins to be investigated by the industrial community. Smart Factories should be populated by manipulator robots able to flexibly adapt to constantly changing line configurations. A safe and productive human-robot interaction should be adopted in presence of ambiguous situations and tight workspaces. Robots can collaborate with humans or they can learn from humans, e.g., through a Learning From Demonstration framework similar to the one implemented in [3]. Moreover, multi-robots cooperative systems should speed up and improve the way to operate. A team of robots, in fact, can subdivide tasks according to their abilities, as suggested in the ontology proposed in [4].

This paper presents a Semantic Knowledge Base that can be adopted as an information resource for autonomous cognitive robots performing manipulation tasks. Robots that have learnt how to manipulate an object (e.g., exploring the workspace, observing a human demonstration or the actions performed by another robot), can save the acquired information in this base and share it with other robots. The purpose is not to formulate a *cognitivist* precoded symbolic representation of the workspace, but to create a knowledge base able to improve the *co-determination* of *emergent systems*.

Traditional cognitive modeling approaches involve symbolic coding schemes as a means for depicting the world. This symbolic representation originates a designer-dependent action domain [5] that is successful if the system acts under the conditions specified by descriptors; otherwise, a *semantic gap* [6] between perception and possible interpretation follows and must be bridged implementing ad-hoc frameworks [7]. Codetermination is the solution. It means that the agent constructs its reality (its world) as a result of its operations in the world: intelligently acting is functionally-dependent on the interaction between perception and cognition.

Focusing on manipulation, the robot should be able to perceive and explore the surrounding workspace, e.g. combining vision sensors with navigation capabilities, in order to detect objects to be manipulated, obstacles to be avoided, or actions performed by human teachers; it has to compute or learn the path that allows the manipulation of objects without collision; it has to learn how to approach the object (e.g., the grasp pose and grasp force), e.g. combining a trial-and-error reinforcement learning with a matching phase that compares the achieved results with the desired ones. It should be able to store descriptions of learned tasks, detected objects, and explored environments creating a knowledge base that facilitates the replication of actions by the same robot or by different robots with the same assignment. Creating a Knowledge Base that can be adopted as information resource become fundamental to create intelligent cognitive robots.

The rest of the paper is organized as follows. Section II gives an overview of existing work on using a Semantic Knowledge Base, and our general approach to its design. Section III describes in details our implementation with regards to its instances, the language used to define and query it, and the interlink algorithm adopted to avoid replicas. Section IV describes the experiment set up to prove the good functioning of the system. Section V contains conclusions and future work. Authors discuss about how the adoption of this system can face robotics challenges such as knowledge acquisition, learning from previous tasks, and dissemination of knowledge to other robots.

II. Related Work

Many existing works aim to define and populate a robotics knowledge base. The Semantic Object Maps (SOMs) of Rusu at al. [8], its extended version (SOM$^+$s) presented by Pangercic at al. [9], the Columbia Grasp dataset [10], the MIT KIT object dataset [11], and the Willow Garage Household Objects Database [12] are available on line. KnowRob [13], the knowledge base of RoboEarth [14], is the most widespread. The Semantic Object Maps only stores information about objects in the environment, including 3D CAD models of objects, their position and orientation, their appearance, and function. The others also give information about grasp poses and can be used to evaluate different aspects of grasping algorithms, including grasp stability [15], robust grasping [16] and scene understanding [17]. The Household object database is a *simple* SQL database. All other approaches aim to make robots autonomous and able to store and share data without human intervention. Hence, as stated in [18] for the RoboEarth language, information is represented in a machine-understandable format, the same format required by the Semantic Web [19], in which computers exchange information between each other. The meaning of the content needs to be represented explicitly in terms of separating logical axioms that a computer can understand. These logical axioms need to be well-defined, for example in an ontology.

Similar to RoboEarth, our Semantic Knowledge Base is defined by an ontology and provides an interface to the open-source Robot Operating System (ROS) [20] that guarantees its reusability. The Base defines a semantic representation language for actions, objects, and environment. It contains 3D models of objects and their functions, but these models are not essential to manipulate the queried objects. Existing databases stores objects as triangular meshes. Stored items are of high quality, but object creation required either a lot of manual work or expensive scanning equipment. In order to save time and money, RoboEarth models objects as 3D colored point clouds. [21]. Anyway, each object model still consists of several recordings from different points of view. In a real world scenario I) it is difficult to reconstruct the 3D model of an object seen for the first time II) manipulation can be performed independet of the 3D model of the object. For these reasons, without loss of information, the proposed database models objects as a set of basic 3D geometric shapes, such as Sphere, Cylinder, Cube, Cone, and etc. Manipulation configurations are generated according to these shapes. In this way, it is possible to manipulate known objects (objects which 3D model is saved in the Cloud), objects according to their functions (objects which functon is saved in the Cloud), novel objects (objects which shape is saved in the Cloud).

The knowledge base can be used either by service robots or industrial manipulators, either by autonomous robots exploring their surrounding or robots learning from demonstrations of other agents (either human or robotic).

Instances of the ontological database are interlinked through a novel interlinking algorithm [22]. As stated in [23], large datasets collected from distributed sources are often dirty with erroneous, duplicated, or corrupted data. The adopted interlinking algorithm finds the interlink pattern of two data sets applying two machine learning methods, the K-medoids [24] and the Version Space [25]. This algorithm largely reduces the computation of comparing instances with respect to the commonly used manually interlinking.

III. The Semantic Knowledge Base

The implemented Semantic Knowledge Base contains descriptions of manipulation actions (grasp, push, place), objects (3D models and shapes), and environments (trajectories already performed). A single database encodes the information and collects the data provided by different robots with different sensing and gripping capabilities. The robot can query the knowledge base to retrieve known information or it can save new information and share it with other robots. To improve re-usability, the knowledge base is fully integrated into ROS. The database and the ontology are accessible via the Web and can easily be extended by users (see Figure 1).

A. Classes and instances

Let a robot, equipped with a gripper and a vision system, stand in the scene (real or simulated environment). The database encodes the scene as follows.

- **Automaton**: stores automata models. Every automaton is composed of a tuple $<$**Robot**, **Gripper**, **Sensor**$>$. The system has a self-model consisting of an XML file describing its kinematic structure, and a semantic model describing the meaning of the system's parts (e.g., robot, gripper, vision sensor). The Universal Robot Description File (URDF) format is used to represent the kinematic model, and the Semantic Robot Description Language (SRDL) [26] is used to describe every component and its capabilities, matching them against the requirements specified for the action space;
- **Robot**: models some robots (robot joints and limits);

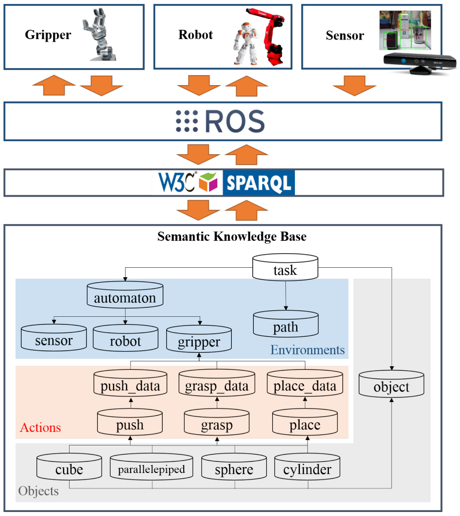

Fig. 1. The Semantic Knowledge Base

- **Gripper**: models some robotics end-effectors. It contains information about gripper joints and their limits. The authors distinguish the **Robot** class from the **Gripper** one because a user can attached a single gripper to different types of robotic arms. The same is true in reverse. The combination of an arm (instance of **Robot**) and a gripper (instance of **Gripper**) forms an instance of **Automation**. If a robot incorporates a gripper, e.g., the humanoid Aldebaran NAO[1], then the instance *NAO* of the class **Robot** will not be linked to any instance of **Gripper**. In this case, **Automation** will consist only of **Robot**;
- **Sensor**: models some robotics sensors. E.g., the Kinect vision sensor [27].

Every automaton in the scene can map its workspace and construct a 2D or 3D map of its surrounding. The map includes the trajectories performed.

- **Path**: contains Cartesian pose points sampled by a motion planning algorithm to lead a robot, or its end-effector, from a start to an end configuration.

Also the description of encountered objects is available.

- **Object**: describes 3D object surface models. It contains the list of object's meshes. Every object has a function (e.g., drink) and a pose [x, y, z]. It is connected to its shape. Every object is associated with a shape, but a shape could be correlated with no item. This feature guarantees the chance to model unknown objects;
- **Sphere**: represents spheres. It contains elements such as the $radius$ of the sphere;
- **Cylinder**: represents cylinders ($radius$, $height$);
- **Cube**: represents cubes ($side$);
- **Parallelepiped**: represents parallelepipeds (*height*, *width*, *depth*).

Currently, detected objects are mapped only into the four above-mentioned 3D basic shapes (Sphere, Cylinder, Cube, Parallelepiped). If a computed shape does not perfectly match with its corresponding object, the manipulation task will still end well thanks to the application of an *Effect Evaluator* which will refine the poses of the end-effector (see Section IV). Please note that the database is available on-line and can be extended by users.

This information allows robots to efficiently manipulate already manipulated objects or navigate already navigated environments. In detail, a task is assigned. It asks the automaton for the fulfillment of a manipulation action on an object. The knowledge base provides support to execute commands like *'Grasp a cup!'*, *'Grasp something suitable for drinking!'*, *'Grasp the object at pose [x, y, z]!'*, that means to manipulate a known object, an object for which the function is known, a novel object. The first query needs a 3D description of the object and its recognition, the second one requires an ontology linking objects with their function, the last one involves the detection of new objects and a way to represent the retrieved information into the dataset. The best way we found is associating the object with its shape and generating the manipulation action according to this shape.

- **Task**: Contains the list of tasks. Every instance models the action to be performed and the object to be manipulated. Information about the initial and final pose of the object are included, together with the time limit under which the action must be completed. For every task, its outcome is reported as a string (*completed, error, running, to_run*) describing the final state of the manipulation.

Studies have demonstrated that: I) placing the arm at a certain distance in front of the object before acting improves actions; II) humans typically simplify the manipulation task by selecting only one among different prehensile postures based on the object geometry [28]. Following this approach, the MoveIt! Simple Grasps tool[2] implemented by Dave T. Coleman provides the following instances:

- **Grasp**: contains the safety distance to be kept during the pre-grasp phase and the grasp poses [x, y, z, $roll$, $pitch$, yaw] of the gripper;
- **Grasp_data**: contains the joints configuration that the gripper has to maintain during the pre-grasp and grasp phases.

We extended the MoveIt! Simple Grasps tool and our implementation provides the **Grasp** and **Grasp_data** instances, and besides, the following istances:

- **Push**: as **Grasp**;
- **Push_data**: as **Grasp_data** but with different joints configurations;
- **Place**: contains the place poses [x, y, z, $roll$, $pitch$, yaw] of the gripper and the distance that the gripper has to mantain after the place;
- **Place_data**: contains the gripper's joints configurations. The grasp configuration becomes the place one and the pre-grasp configuration becomes the post-place one.

[1] Aldebaran NAO robot: https://www.aldebaran.com/en/humanoid-robot/nao-robot

[2] MoveIt! Simple Grasps tool: https://github.com/davetcoleman/moveit_simple_-grasps

The actual organization of the knowledge base allows the definition of an actions' hierarchy that is able to generate complex actions by composing simple actions. Examples follow:

- **Put**: **Grasp** $\cap$ **Place**.

B. The ontology

As in [4], we first have to choose the appropriate Semantic Web language to describe the information provided by the database. We compared Extensible Markup Language (XML)[3], Resource Description Format (RDF)[4] and Web Ontology Language (OWL)[5]. XML lets the definition of a format, it is verbose and data exchange requires a set of basic rules to allow different systems to communicate and understand each other. RDF is used to define a model and it does not need to be redefined when new knowledge should be stated: its schema stays the same. If we want to define an ontology, we do not have to define a message format. We have to define a knowledge representation, naming and defining the types, properties, and interrelationships of the entities of a particular domain [29]. For this reason, we selected the union of RDF and OWL, namely OWL Full. RDF is used to define the structure of the data, OWL adds semantics to the schema and allows the user to specify relationships among the data. OWL Full allows an ontology to augment the meaning of the pre-defined RDF vocabulary guaranteeing the maximum expressiveness of OWL and the syntactic freedom of RDF. Indeed, OWL is adopted by the Word Wide Web Consortium (W3C)[6] and it is the representation language used by the IEEE Robotics and Automation Society (RAS)'s Ontologies for Robotics and Automation (ORA) Working Group [30], [31], [32].

The ontology describes the relationship between the defined classes and their attributes. For example, for each entity of the class **Object** it defines the properties *shape* and *function*. It associates the shapes to the suitable manipulation actions, the type of gripper and the type of robot.

C. Queries

Queries allow robots to investigate the knowledge base and retrieve existing data. A robot able to query the database has the capability to efficiently and intelligently perform tasks. In our case, a Python interface lets ROS users query the Semantic Knowledge Base using SPARQL[7].

D. The interlinking algorithm

In order to populate the knowledge base as much as possible, we interlinked instances of the proposed knowledge base with the ones of the Household Objects Database [12] provided by Willow Garage to the ROS community. Willow Garage created this database to provide, for each 3D surface model of the objects in the database, a large number of grasp points that are specific to a PR2 robot and its gripper.

In the considered sets, there can be instances that describe the same resource in the world. By interlinking these two instances, the two sets can be treated as one.

Interlinking can be done manually, if there are not many instances being created. Otherwise, an algorithm should be applied to automate the interlinking process. In [22] the interlink pattern of two data sets is found out applying two machine learning methods, the K-medoids and the Version Space. Although interlinking algorithms require interactions with users for the sake of the interlinking precision, computations of comparing instances are largely reduced than manually interlinking.

Algorithm 1 aims to interlink instances across two data sets *D* and *D'*. The algorithm first computes property/relation correspondences across two data sets (line 5). Then, instances property values are compared by referring to the correspondences (line 10). A similarity value v is generated upon all similarities of property values (line 11). If such a similarity is equal to or larger than a predefined threshold T, the two compared instances can be used to build a link with the relation *owl:sameAs* (line 12-14).

Algorithm 1 Interlinking Instances across Data Sets

Input: Two Data Sets
Output: Links accross Data Sets
1: The data set *D*, *D'*; /*two data sets to be interlinked*/
2: Similarity threshold *T*
3: **for** Each property/relation in the data set *D* **do**
4:     **for** Each property/relation in the data set *D'* **do**
5:         Match properties/relations that are corresponding to each other and store as the alignment *A*
6:     **end for**
7: **end for**
8: **for** Each instance in the data set *D* **do**
9:     **for** Each instance in the data set *D'* **do**
10:         Compare instances' property values according to the correspondences of the alignment *A*;
11:         Aggregate all similarities between property values as a similarity value v
12:         **if** $v >= T$ **then**
13:             The two compared instances are interlinked with *owl:sameAs*.
14:         **end if**
15:     **end for**
16: **end for**

IV. Experiments and Results

In order to prove the improvement introduced by the Semantic Knowledge Base, we implemented a sense-model-act framework.

A. Sense

A vision sensor acquires world data. We selected a Microsoft Kinect. RGBD images of the environment are converted into 3D point clouds and segmented into individual graspable objects using the ROS Tabletop segmentation tool[8] developed by Marius Muja.

1) Shape Detector: Each segmented object is represented by a point cloud. We use raw clouds and the input coordinates to extract the object and compute its shape.

[3]Extensible Markup Language (XML): http://www.w3.org/XML/
[4]Resource Description Format (RDF): http://www.w3.org/RDF
[5]Web Ontology Language (OWL): http://www.w3.org/2001/sw/wiki/OWL
[6]Word Wide Web Consortium (W3C): http://www.w3c.com/
[7]Simple Protocol and RDF Query Language (SPARQL): http://www.w3.org/TR/sparql11-query
[8]Tabletop Object Detector: http://www.ros.org/wiki/tabletop_object_detector

2) Object Recognizer: The tool allows robots to recognize objects. The tool is based on the ROS Tabletop segmentation tool. From the point cloud of an object, the tool extracts its meshes. Matching new objects meshes with existing ones, objects are recognized. The tool adds knowledge to the data set but it is not essential to solve manipulation tasks.

B. Model

1) Semantic Knowledge Base: After the environment mapping, the robot accesses the Semantic Knowledge Base to find a match between the segmented objects and the objects saved in the database. If the match exceeds a certain threshold, then the object is assumed to be recognized (if its 3D model exists) or detected (if only the shape is known). If (at least) the shape, the assigned action, and the gripper joints configuration are retrieved from the base, then a plan is generated containing the kinematics information required for the system to pass from the initial to the goal configuration.

2) Action Generator: If no information about the object, action, and joints configurations is stored in the database, then new data are generated. The framework extends the MoveIt! Simple Grasps tool to generate all possible grasp/push/place poses. Given the desired safety distance, the generator aligns the hand with the object principal axes and tries to manipulate the object around its center of mass starting from either the top or from the side of the object. It generates a list of possible poses of a gripper relative to the object, stores data in the Semantic Knowledge Base, and shares them among the robotics community.

3) Motion Planner: The planner adopts MoveIt! and the Kinodynamic Planning by Interior-Exterior Cell Exploration (KPIECE) [33] planner from the Open Motion Planning Library (OMPL) [34] library to compute a collision free path for manipulating the object. It chooses, from the set of generated actions, the first behavior that is kinematically feasible and collision-free and generates the plan. The collision check is performed on the 3D collision map created from the Kinect point cloud and takes into account collisions with the environment, joint limits and self collisions of the robot's arm. Any object manipulated by the robot is included in the robot model, in order to avoid collisions between the object and the rest of the environment during transport.

C. Act

1) Robot Controller: It activates the simulated/real engines that drive the robot.

2) Effect Evaluator: If we reason about arbitrary shapes, collisions or detachments can be induced by pre-selecting the manipulation configuration. To overcome the problem, the *Action Generator* generates the gripper's joints configuration required to perform the task. The *Motion Planner* plans movements. The *Robot Controller* actives robot motors and moves the robotic system along the planned path. During the execution of the task, failures may occur. The *Effect Evaluator* uses the information acquired by sensors to compare the system final state with the expected one. In case of mismatch, a learning module starts a trial and error routine that corrects the joints configuration and generates a new configuration. The configuration that allows the task achievement overwrites the existing one in the Semantic Knowledge Base.

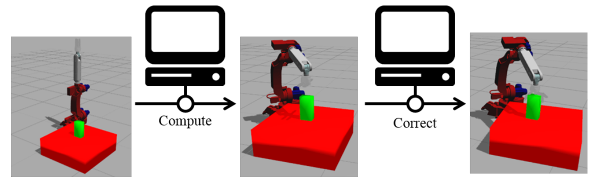

Fig. 2. The first experiment: the system compute the first pose and correct it until the achievement of the result

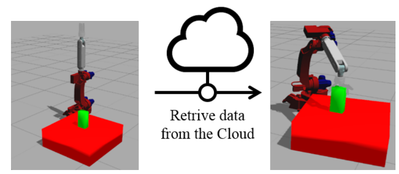

Fig. 3. The second experiment: the system retrive the information from the Cloud

D. Results

We validated the framework by performing experiments on a Comau Smart5Six robot equipped with a Schunk SDH gripper. Gazebo [35] was used as a simulated environment. In simulation, the robot has to grasp a parallelepiped.

During the first attempt, the system has no prior knowledge about the object. Manipulation data must be computed. The robot approaches the object and fails when trying to attempt the action. The trial-and-error approach allows the robot to manipulate the object (see Figure 2).

During the second attempt, the object is known and the Semantic Knowledge Base stores its manipulation data. The robot is able to manipulate the object on the first try (see Figure 3).

V. Conclusion and Future Work

As stated in [36], "Knowledge processing is an essential resource for autonomous robots that perform complex tasks in dynamic environments. Robots need advanced reasoning capabilities to infer the control decisions required for competently performing complex tasks like manipulation. Their knowledge processing system has to provide them with common-sense knowledge, with the ability to reason about observation of the environment, and with methods for learning and adapting over time". In this paper, from the study of humans actions when handling objects, an abstraction of the manipulation domain was formulated and encoded into an OWL ontology. A Semantic Knowledge Base was created to collect data representing the domain. We proved our approach by building a ROS framework able to associate manipulation actions to objects' shapes. Linking actions to shapes instead of objects' 3D CAD models increases the framework's reusability and guarantees its functioning when dealing with unknown objects. Tests were performed in simulation and required the manipulation of I) a novel object II) the same object located in the Cloud.

As future work, the authors aim to extend the type of encoded actions storing not only translational pushes but also rotational ones. This implies the possibility to accomplish complex movements such as opening or closing doors. In fact, if a robot has to open a door, it will grasp the handle and perform a rotation. Moreover, while the *model* and *act* modules are well understood, we are actively working to fully define the *sense* module and to prove its well-functioning. We aim to improve also the *Effect evaluator* and to provide proofs of its well-functioning using other robots and grippers. The authors are performing tests in simple manipulation tasks on a real Comau Smart5Six robot.

REFERENCES

[1] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, pp. 1125–1134, 2010.

[2] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," Technische Universitt Dortmund (Fakultt Maschinenbau), Audi Stiftungslehrstuhl Supply Net Order Management, Tech. Rep., 2015.

[3] E. Tosello, S. Michieletto, A. Bisson, and E. Pagello, "A Learning from Demonstration Framework for Manipulation Tasks," in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of.*, 2014, pp. 1–7.

[4] Z. Fan, E. Tosello, M. Palmia, and E. Pagello, "Applying Semantic Web Technologies to Multi-Robot Coordination," in *Workshop NRF-IAS-2014, Proceedings Of.*, Venice, Italy, 2014.

[5] G. Metta, G. Sandini, D. Vernon, D. Caldwell, N. Tsagarakis, R. Beira, J. Santos-Victor, A. Ijspeert, L. Righetti, G. Cappiello, G. Stellin, and F. Becchi, "The RobotCub project - an open framework for research in embodied cognition," *Humanoids Workshop, IEEE RAS International Conference on Humanoid Robots*, 2005.

[6] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 1349–1380, 2000.

[7] J. Pauli and G. Sommer, "Perceptual organization with image formation compatibilities," *Pattern Recognition Letters*, pp. 803–817, 2002.

[8] 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 *Procceding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.

[9] D. Pangercic, M. Tenorth, B. Pitzer, and M. Beetz, "Semantic Objects Maps for Robotic Housework - Representation, Acquisition and Use," in *Procceding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[10] C. Goldfeder, M. Ciocarlie, and P. Allen, "The Columbia grasp dataset," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Kobe, Japan, 2009, pp. 1710–1716.

[11] A. Kasper, Z. Xue, and R. Dillman, "The KIT object models database: an object model database for object recognition, localization and manipulation in service robotics," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 8, pp. 927–934, 2012.

[12] M. Ciocarlie, K. Hsiao, E. Jones, S. Chitta, R. Rusu, and I. Sucan, "Towards reliable grasping and manipulation in household environements," in *in Proc. Int. Symp. Experimental Robotics*, Delhi, India, 2010, pp. 1–12.

[13] M. Tenorth and M. Beetz, "KnowRob: A Knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, vol. 23, no. 5, pp. 566–590, 2013.

[14] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft, "RoboEarth - A World Wide Web for Robots," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 2, pp. 69–82, 2011.

[15] H. Dang and P. K. Allen, "Learning grasp stability," in *in Proc. Int. Conf. Robot. Autom. (ICRA)*, Saint Paul, MN, 2012, pp. 2392–2397.

[16] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, St. Paul, MN, USA, 2012, pp. 557–562.

[17] M. Popovic, G. Koostra, J. A. Jorgensen, D. Kragic, and N. Kruger, "Grasping unknown objects using an early cognitive vision system for general scene understanding," in *Proc. Int. Conf. Intell. Robot. Syst. (IROS)*, San Francisco, California, USA, 2011, pp. 987–994.

[18] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz, "The RoboEarth language: Representation and Exchanging Knwoledge about Actions, Objects, and Environments," in *23rd International Joint Conference on Artificial Intelligence (IJCAI) Special track on Best Papers in Sister Conferences (invited paper)*, 2013.

[19] T. Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[21] D. Di Marco, A. Koch, O. Zweigle, K. Haussermann, B. Schiessle, P. Levi, D. Galvez-Lopez, L. Riazuelo, J. Civera, J. M. M. Montiel, M. Tenorth, A. Perzylo, M. Waibel, and R. Van de Molengraft, "Creating and using RoboEarth object models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3549–3550.

[22] Z. Fan, "Concise Pattern Learning for RDF Data Sets Interlinking," Ph.D. dissertation, University of Grenoble, 2014.

[23] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[24] L. Kaufman and P. Rousseeuw, *Statistical Data Analysis Based on the L1-Norm and Related Methods*, North-Holland, 1987, ch. Clustering by means of Medoids, pp. 405–416.

[25] V. Dubois and M. Quafafou, "Concept learning with approximation: Rough version spaces," in *Rough Sets and Current Trends in Computing: Proceedings of the Third International Conference (RSCTC)*, Malvern, PA, USA, 2002, pp. 239–246.

[26] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shangai, China, 2011, pp. 9–13.

[27] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318 – 1334, 2013.

[28] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," in *Robotics and Automation, IEEE Transactions on*, vol. 5(3), 1989, p. 269279.

[29] T. R. Gruber, "A translation approach to portable ontology specifications," in *Knowledge Acquisition*, vol. 5(2), 1993, p. 199220.

[30] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez, "An IEEE standard Ontology for Robotics and Automation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Vilamoura, Algarve, Portugal, 2012, pp. 1337 – 1342.

[31] C. I. Schlenoff, "An Overview of the IEEE Ontology for Robotics and Automation (ORA) Standardization Effort," in *Standardized Knowledge Representation and Ontologies for Robotics and Automation, Workshop on the 18th*, Chicago, Illinois, USA, 2014, pp. 1–2.

[32] E. Prestes, S. R. Fiorini, and J. Carbonera, "Core Ontology for Robotics and Automation," in *Standardized Knowledge Representation and Ontologies for Robotics and Automation, Workshop on the 18th*, Chicago, Illinois, USA, 2014, pp. 7–9.

[33] I. A. Sucan and L. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Workshop on the Algorithmic Foundations of Robotics*, Guanajuato, Mexico, 2008.

[34] I. A. Sucan, M. Moll, and L. Kavraki, "OMPL Library," 2010. [Online]. Available: http://www.ros.org/wiki/ompl

[35] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Sendai, Japan, 2004, pp. 2149–2154.

[36] M. Tenorth, D. Jain, and M. Beetz, "Knowledge processing for cognitive robots," *KI - Knstliche Intelligenz*, vol. 24, no. 3, pp. 233–240, 2010.