Fuzzy-DL Perception for Multi-Robot Systems

Markus Eich, Ronny Hartanto

DFKI
Robotics Innovation Center
Bremen, Germany
markus.eich@dfki.de
ronny.hartanto@dfki.de

Abstract

For future planetary robot missions, multi-robot-systems can be considered as a suitable platform to perform space mission faster and more reliable. In heterogeneous robot teams, each robot can have different abilities and sensor equipment. In this paper we describe a lunar demonstration scenario where a team of mobile robots explores an unknown area and identifies a set of objects belonging to a lunar infrastructure. Our robot team consists of two exploring scout robots and a mobile manipulator. The mission goal is to locate the objects within a certain area, to identify the objects, and to transport the objects to a base station. The robots have a different sensor setup and different capabilities. In order to classify parts of the lunar infrastructure, the robots have to share the knowledge about the objects. Based on the different sensing capabilities, several information modalities have to be shared and combined by the robots. In this work we propose an approach using spatial features and a fuzzy logic based reasoning for distributed object classification.

1 Introduction

In this paper we present our latest results of the project IMPERA (Integrated mission planning using heterogeneous robots). The main goal of the project is the development of a planning and a plan execution architecture using a team of robots within a lunar scenario. For future lunar and other planetary missions, system autonomy becomes more and more mandatory. Current NASA missions (e.g. Mars Exploration Rover[1] and Mars Science Lab[2]) are dealing mainly with the exploration of the Mars surface and the analysis of the surface. The systems Spirit, Opportunity, and Curiosity work thereby as individual systems. Looking into the future, it is a likely scenario to install infrastructure and other scientific components on Mars or on the Moon. This infrastructure can consist of small stations measuring environmental conditions, units used for providing drill cores for subsurface analysis, or modules for communication and energy supply. In order to perform a mission cooperatively, the robots have to share a common knowledge about the environment, such as the type of objects and modules the robots have to identify, or the location of these objects.

Figure 1. : Left: The mobile manipulator robot AMPARO. Right: The scout robot used for visual detection of infrastructure modules.

In our scenario, the robot team consists of three robots having different sensing abilities. The robot AMPARO (cf. Figure 1, left) uses a tilting 3D laser range finder to generate 3D point clouds of the environment. These data are used to detect three dimensional objects during a fetch and transport phase of the mission. Two scout robots, based on the Pioneer AT robot (cf. Figure 1, right), are used for cooperative map building and object candidate detection using monocular color images during the exploration phase of mission. The objects used during our demonstration scenario are shown in Figure 2. The mission consists of the sequentially executed tasks of explor-

[1]www.nasa.gov/mission_pages/mer/index.html
[2]www.nasa.gov/mission_pages/msl/index.html

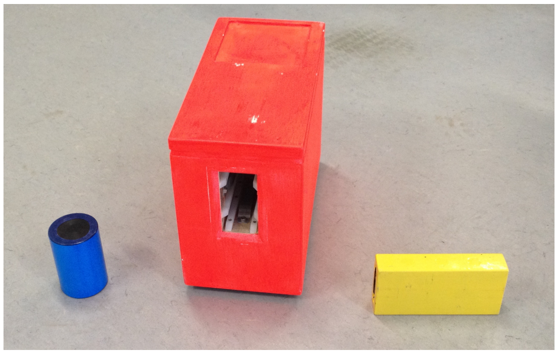


Figure 2. : The three objects which need to be detected by the robot team during the mission: battery pack (yellow), soil sample containers (blue), base station (red).

ing an unknown environment while simultaneously locating three distinct objects. The objects consists of a battery pack, a soil sample container and a base station to which the other two objects have to be transported. The described scenario is an extension to the scenario described in [3] where only cylindrical sample containers are regarded. In this work, we extended the concept to other regular shaped objects, such as battery packs and the base station. In our approach we use spatio-semantic knowledge about the environment in order to classify the objects [2]. Spatio-semantic description is based on the concepts that objects can be described how they "look like" in terms of shape features (e.g. planar, cubical, cylindrical), spatial features (e.g. size, extension, orientation), spatial relations, and color. Spatio-semantic definitions can be extracted directly from the sensor data (e.g. using point cloud processing in combination with segmentation and cluster analysis) and are describable in a spatio-semantic ontology, such as "The soil sample container is a blue cylinder *and* is perpendicular to the ground" or "The battery pack is a yellow box *and* the battery pack has an edge length of 20 cm".

Extracted spatial features cannot be matched against a discrete ontology (due to sensor noise, occlusions, etc.), therefore an imprecise (fuzzy) knowledge base has to be modeled to estimate the best model match. A constraint network is one way for object classification [2]. Most of the basic perception approaches extract and analyze clusters within the point cloud prior to object classification. Some 2D and 3D shape extraction algorithms are given in [11] for 2D shapes and in [10] for 3D shapes. Semantic perception methods, which also take semantics into account, are presented in [5]. The authors describe in their work the bridging between the spatial domain and the semantic domain which they call S-Box (spatial box) and T-Box (taxonomy box). The semantic interpretation of physical objects is done by optical marker identification but not directly on spatial interpretation of point cloud data. In approach described in [7], a constraint network is used to identify spatial entities such as walls, floors or doors.

2 System Overview

The system architecture presented in this work consists of several independent modules which are running on distinct robots. On each robot we use the Robot Operating System (ROS) as a communication backbone between the modules [9]. The overall architecture and its components are depicted in Figure 3. In order to establish the communication

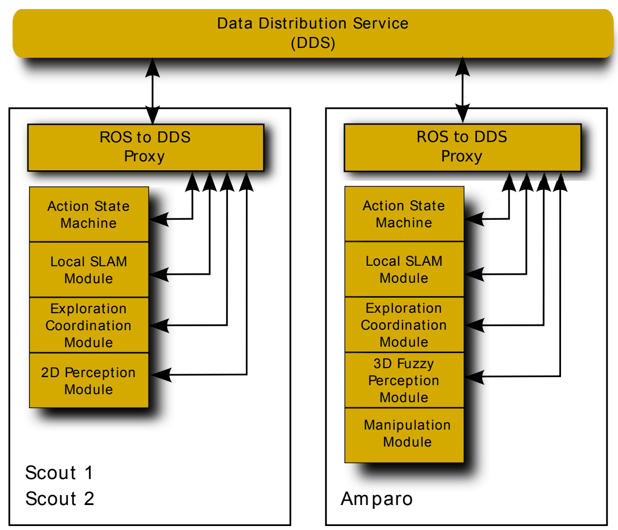


Figure 3. : System architecture of the robot components.

between the systems, we make use of a reliable, cloud based communication interface based on a commercial version of the *Data Distribution Service (DDS)* [8]. DDS is based on the loosely coupled publish subscribe paradigm and provides several QoS options like automatic re-connection and data buffering in case of communication loss. More details about the inter-robot communication framework is given in Section 3. The DDS cloud ensures that each robot has the same knowledge about the internal status of each individual robot. Some information, which is distributed via DDS has to be stored persistent in order to enable the robots to perform actions, based on the common knowledge base about the mission and the environment. This information is taken directly from the DDS cloud and stored as a local copy on each individual robot. The DDS cloud itself stores only information needed by other robots for coordination, especially during the exploration and mapping phase. Some information has to be stored persistently, such as the poses and the IDs of detected sample containers.

3 Cloud-Based Multi-Robot Communication

All participating robots in the team have to exchange data constantly in order to achieving a common goal. One challenge in the multi-robot communication is the limited resource and the availability of the network. In a large area, a robot can lose the connection with the other robots due to the distance. However, all messages from one robot to the other systems have to be maintained and have to be guaranteed to be delivered. Having a limited resource or bandwidth, it is not possible to have a communication protocol which allows request of certain messages, e.g., based on the sequence IDs or time interval. This kind of communication can introduce huge network traffic, depending on the number of involved robots. In [3, 6], a transparent communication network based on publish/subscribe paradigm using Data Distribution Service [8] (DDS) is described in detail. The advantages of this approach are transparency and robustness. All messages will be delivered successfully from the publisher to the subscribers without any explicit commands. Thus, this approach is resource efficient as each message will be delivered only once from the source to the destinations. The messages are guaranteed to be delivered to all subscribers, even on a temporarily network outage.

Another important aspect of the cloud-based communication is the modeling of the messages format. A well defined message format can minimize the required bandwidth for communicating them among the agents and reduce further post-processing tasks. Two message formats are used for sharing objects information between robots. The first message, "candidate-object", is used by the scouts for publishing the detected object candidates with respect to the map. The second message, "identified-object", is used by the AMPARO for publishing the identified object with their properties.

The scout robots detect the object candidates using a monocular color camera, thus the position of the object cannot be extracted from the camera data due to the missing depth information. However, the position while detecting the candidate can be used by the AMPARO later on for the identification of the objects. Depending on the size and position of the color blob on the camera data, a probability function can be calculated based by other agents from the shared information. The candidate-object (CO) message is modeled as a 5-tuple as follows:

$$CO = (R_{ID}, O_{prop}, P_{R_{ID}}, O_{R_{ID}}, C)$$

where: R_{ID} is the ID of the robot who detects the object candidate, e.g. scout1. O_{prop} is is the color property of the object, e.g. yellow, blue, or red. $P_{R_{ID}}$ and $O_{R_{ID}}$ are the position in cartesian coordinate (p_x, p_y, p_z) and orientation in quaternion (o_x, o_y, o_z, o_w) of the robot while detecting the object candidate. C is the two-dimensional covariance (c_x, c_y) defining the certainty of the object candidate, depending on the size and position of the color blob on the camera data.

Once the candidate object message is published to the cloud, all participating robots will receive the shared information. The candidate object message enables the agents to calculate the two-dimensional probability density function on their map. The function is defined as follows:

$$\begin{aligned}
f(map, \mu, \Sigma) &= \frac{1}{\sqrt{|\Sigma|(2\pi)^2}} e^{-\frac{1}{2}(map-\mu)\Sigma^{-1}(map-\mu)'} \\
\Sigma &= R * \begin{bmatrix} c_x & 0 \\ 0 & c_y \end{bmatrix} * R' \\
\mu &= \begin{bmatrix} p_x \\ p_y \end{bmatrix}
\end{aligned}$$

where: *map* is two-dimensional map of the environment and R is the rotation matrix extracted from the yaw angle of the $O_{R_{ID}}$.

AMPARO can use the probability density function (PDF) from the candidate object message for calculating a pose where it can start identifying the object. Sometimes, an object can be found by different robots, thus multiple candidate object messages referring to the same object are broadcasted through the cloud. Processing these PDFs could infer the position of the object itself instead of the location where the scouts detected it. As a result, AMPARO could also compute the ideal pose for identifying the object from the inferred object pose.

Once the object is successfully identified, the information about the identified object is shared to all participating robots through the cloud. The identified object message is modeled as such for enabling further tasks by other agent, e.g. object manipulation. The identified object (IO) message is defined as 8-tuple, as follows:

$$
\begin{aligned}
ID \;\; = \;\; & (O_{ID}, O_{type}, O_{color}, O_{bbox}, O_{likelihood}, \\
& P_{O_{ID}}, O_{O_{ID}}, C_{O_{ID}})
\end{aligned}
$$

where: O_{ID} is the ID of the identified object that uniquely generated for the overall mission. O_{type} is the object type from the enumerated value, e.g. battery, soil-sampled, etc. O_{color} is the color of the identified object. O_{bbox} is the dimension of the identified object represented in bounding box, which later on be used for calculating the grasping pose. $O_{likelihood}$ is the likelihood of the perception module that telling how good the object is being identified as a type O_{type}. $P_{O_{ID}}$ and $O_{O_{ID}}$ are the position and orientation of the identified object in cartesian coordinate and quaternion. $C_{O_{ID}}$ is the covariance of the identified object on the map.

4 Semantic Perception

4.1 Spatial Feature Description and Extraction

In our demonstration scenario, the team of robots has to classify the objects shown in Figure 2. The objects are first roughly located as candidates using the faster scout robots, using only monocular sensing capabilities. The robots can detect the approximate position within the map but no information about the accurate 3D pose of the objects. Once the candidates are identified, the robot AMPARO moves to the estimated location of the objects and uses the tilting 3D laser range finder for object localization and classification. For the detection process, we introduce a perception pipeline, depicted in Figure 4. To extract the spatial features from the 3D point cloud, we use a RanSaC-based segmentation approach described in [10]. The estimation of the spatial relationship between the objects and the ground plane is fundamental in our approach, therefore the ground plane is segmented and classified first.

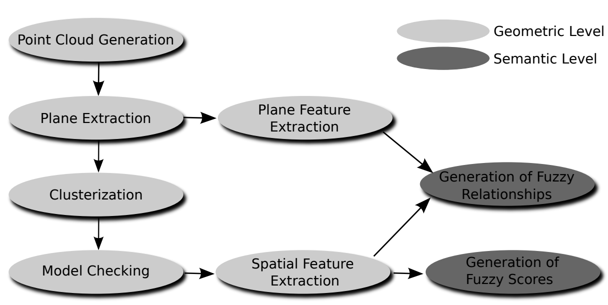


Figure 4. : Perception Pipeline

This is achieved using a region growing algorithm, described in [2]. The remaining points, not belonging to the ground plane, are clusterized. In the following step, the spatial features are extracted from these remaining clusters. The objects (cf. Figure 2) are basically cylindrical shapes (the soil sample container) and cuboids (base station and battery pack). The base station is a cuboid which has a vertical slot on one side and a switch on the other opposide side. The detection of the soil sample containers using spatial features is described in detail in [3]. The spatial features we use for the detection are summarized below:

$$
\begin{aligned}
\Phi_1 &= CylinderModelFit & &|\Phi_1 \in \mathbb{R}, 0 \le \Phi_1 \le 1 \\
\Phi_2 &= Radius & &|\Phi_2 \in \mathbb{R} \\
\Phi_3 &= Height & &|\Phi_3 \in \mathbb{R} \\
\Phi_4 &= Orientation & &|\Phi_4 \in \mathbb{R}^3 \\
\Phi_5 &= Position & &|\Phi_5 \in \mathbb{R}^3 \\
\Phi_6 &= HeighTo & &|\Phi_6 \in \mathbb{R} \\
\Phi_7 &= OrthoTo & &|\Phi_7 \in \mathbb{R}, 0 \le \Phi_7 \le 1 \\
\Phi_8 &= ParallelTo & &|\Phi_8 \in \mathbb{R}, 0 \le \Phi_8 \le 1
\end{aligned}
$$

For the cuboids used in our scenario (i.e. the base station and the battery pack), additional features have to be extracted from the 3D point clusters. Instead of searching directly for cuboids in the point cloud, a more general approach has been selected, based on a divide and conquer strategy. In this approach, only planar (2D based) planes have to be extracted. The planar features used are have been selected based on the following observation of the objects. These observations can be described using an informal description logic representation: **a)** The base station is a cuboid *and* the base station has a vertical, rectangular slot on the rear side. **b)** The battery is a cuboid. **c)** The box has three planes which are orthogonal to each other. **d)** In a cuboid model, three orthogonal planes share a common edge. **e)** The base station has planar components named "panels". **f)** All side panels are rectangular.

Given the observations above, the list of features $\Phi_1 - \Phi_8$ is extended in order to check the extracted features with a reasoner.

$$
\begin{aligned}
\Phi_9 &= PlanarModelFit & &|\Phi_9 \in \mathbb{R}, 0 \le \Phi_9 \le 1 \\
\Phi_{10} &= Rectangular & &|\Phi_{10} \in \mathbb{R}, 0 \le \Phi_{10} \le 1 \\
\Phi_{11} &= HasCommonEdge & &|\Phi_{11} \in \mathbb{R}, 0 \le \Phi_{11} \le 1 \\
\Phi_{12} &= Area & &|\Phi_{12} \in \mathbb{R} \\
\Phi_{13} &= maxExtension & &|\Phi_{13} \in \mathbb{R}
\end{aligned}
$$

The features described above are directly extractable from the planar clusters of the point cloud. For details, see [2, 4]. Note that the features $\Phi_9 - \Phi_{11}$ are imprecise definitions and not binary true/false assumptions in a logical sense. The idea behind this is that by using fuzzy sets in a logic expression, sensor noise, measurement errors, and occlusions can be covered. In other words, the specific features are defined by a membership function calculating the likelihood of how extracted features match the features given in a knowledge base. With the definition of the spatial features, a spatio-semantic ontology can be defined which is described in the next section.

4.2 Spatio-Semantic Object Ontology

In this section, we introduce our method on how perceivable features of objects and the environment can be matched with a knowledge based system. Semantic object annotation is in this case accomplished by ontology queries. In our multi robot scenario, only the robot AMPARO has currently the ability to perceive the environment using a 3D point cloud. During the mission the scout robots estimate the positions of the objects based on color segmentation and based on the robots position as described already in Section 3. The robot AMPARO moves to the vicinity of the estimated object poses and initiated the perception pipeline.

The knowledge about the relevant objects and the spatial relation are defined in the T-Box (describing the terminology of the ontology). The A-Box describes the assertion part of the ontology, i.e. the individuals. The ontology of the domain is manually generated using the spatial knowledge about the sample containers in scope. The geometric features are described as concepts of the T-Box. The individuals of the knowledge base are automatically generated by the perception layer (cf. Figure 5). The A-Box is updated after each 3D scan and the FuzzyDL reasoner is triggered to classify the soil sample container, the base station and the battery pack within the point cloud, based on the spatio-semantic description of the T-Box. In this paper, we extend the fuzzy description logic based approach described in [3]. The key idea is to describe the knowledge base for object classification using an ontology which can deal with vagueness and membership values. FuzzyDL as a knowledge base and the corresponding reasoner was firstly introduced in [1]. The language defined by FuzzyDL is given by

$$C, D := \top | \bot | A | C \sqcup D | C \sqcap D | \neg C | \forall R.C | \exists R.C$$

where C and D are concepts, A defines the atomic concept, and R.C the role of a concept. For our spatial reasoning approach we make also use of the Gödel t-norm and Gödel t-conorm to express FuzzyDL

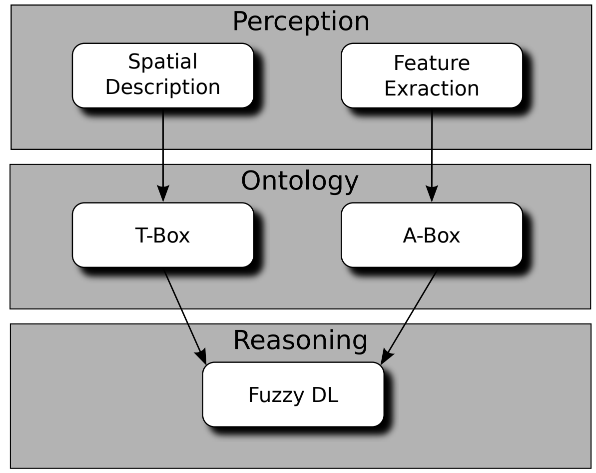


Figure 5. : The reasoning architecture using Description Logic. The A-Box is provided by the feature extractor from the perception side. The T-Box is modeled using FuzzyDL syntax.

union and intersection (i.e. $C \sqcap_G D := min(C, D)$ and $C \sqcup_G D := max(C, D)$ respectively. For a detailed description of the FuzzyDL semantic, please refer to [1]. In order to model the ontology of the objects, the following concrete names are assigned to the different DL concepts and roles (concepts start with capital letters while roles and functionals start with a small letters):

Listing 1: The T-Box of the Knowledge Base

```
GroundPlane linear(0,1,0.7,0.5))
CylinderType linear(0,1,0.7,0.5))
Plane linear(0,1,0.7,0.5))
Rectangular linear(0,1,0.7,0.5))
Slot linear(0,1,0.7,0.5))
Red linear(0,1,0.5,0.5))
Blue linear(0,1,0.5,0.5))
Yellow linear(0,1,0.5,0.5))

SampleContainer (and
  (some isCylinderType CylinderType)
  (some isOrthogonal GroundPlane)
  (>= hasHeight 0.1)(<= hasHeight 0.2)
  (<= hasRadius 0.1)(some isBlue Blue))

FrontPanel (and
  (some isPlane Plane)
  (some isRectangular Rectangular)
  (not Slot)
  (>= hasShortEdgeLength 0.15)
  (<= hasShortEdgeLength 0.25)
  (>= hasLongEdgeLength 0.25)
  (<= hasLongEdgeLength 0.35))
```

```
RearPanel (and
  (some isPlane Plane)
  (some isRectangular Rectangular)
  (some hasSlot Slot)
  (>= hasShortEdgeLength 0.15)
  (<= hasShortEdgeLength 0.25)
  (>= hasLongEdgeLength 0.25)
  (<= hasLongEdgeLength 0.35))

TopPanel (or
  (and
  (some isPlane Plane)
  (some isRectangular Rectangular)
  (not Slot)
  (>= hasShortEdgeLength 0.15)
  (<= hasShortEdgeLength 0.25)
  (>= hasLongEdgeLength 0.35)
  (<= hasLongEdgeLength 0.45))
  (some isParallel GroundPlane))

SidePanel (and
  (some isPlane Plane)
  (some isRectangular Rectangular)
  (not Slot)
  (>= hasShortEdgeLength 0.25)
  (<= hasShortEdgeLength 0.35)
  (>= hasLongEdgeLength 0.35)
  (<= hasLongEdgeLength 0.45)
  (some isOrthogonal TopPlane))

BaseStation (and
  (or
  (some hasFrontPanel FrontPanel)
  (some hasRearPanel RearPanel)
  (some hasSidePanel SidePanel)
  (some hasTopPanel TopPanel))
  (some isRed Red))

BatteryPack (and
  (some isSmallBoxType SmallBoxType)
  (not Slot)
  (<= hasLongEdgeLength 0.2)
  (some isYellow Yellow))
```

The ontology depicted in Listing 1 shows the knowledge base of the objects in our scenario. The features described in the T-Box of the knowledge base are matched with the object candidates. The color of the candidates is identified by the scouts using the monocular camera. The other spatial features are extracted from the 3D point cloud, provided by AMPARO. For instance, the knowledge base for the base station is interpreted as follows: The base station has at least one visible panel and the panel is red. The panels may be one of the type "front panel", "rear panel", "top panel", "side panel". The side panel is an entity which has no slot, is planar, rectangular, and the longest edge of the panel is between 35 cm and 45 cm (the actual model has exactly 40 cm). The output of the reasoner in this case is the likelihood of a perceived object belonging to a defined object concept. Fuzzy concepts are described using spatio-semantic features (cf. Figure 6). In the T-Box we use

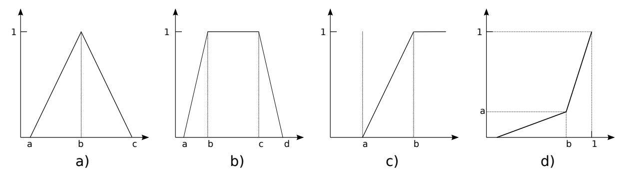


Figure 6. : The used fuzzy concepts for the spatial reasoning approach. The features are represented using the fuzzy concepts of triangular_set (a), crisp_set (b), right_shoulder (c) and the linear_modifier (d).

for our experiments(cf. Listing 1), only linear concepts are used. To query the knowledge base, after the individuals are updated in the A-Box of the ontology, a min-instance query is executed in order to check the satisfiability of the models for The three objects base station, sample container and battery pack. This is achieved by $\inf\{n|\mathcal{K} \models$ (instance $object_i$ BaseStation n)$\}$, $\inf\{n|\mathcal{K} \models$ (instance $object_i$ SampleContainer n)$\}$, and $\inf\{n|\mathcal{K} \models$ (instance $object_i$ BatteryPack n)$\}$ respectively.

5 Experiments and Results

Figure 7 shows the map with the calculated PDFs from the candidate object messages on the cloud. Additionally, the positions of the objects are overlaid on the map for the illustration purpose. Due to the coordinated exploration, the exploration tasks were distributed among the scout robots by minimizing overlapping areas. Due to the start location and position of the object, both scouts were detecting the red object from different viewpoints. The figure also shows that scout1 had higher probability, smaller but higher PDF, than scout2. This is due to the size and position of the color blob on the camera. Table 1 shows some properties of the candidate object messages. This information was used for calculating the PDFs in Figure 7. The second part of the experiment is dealing with the perception of the defined

Table 1. : Object candidate messages

Robot ID	Color	Pos (x,y)	Covar. (x,y)
scout1	red	(3.84, 4.65)	(0.1, 0.5)
scout1	blue	(18.17, 1.68)	(0.3, 1.25)
scout2	yellow	(-2.83, 7.30)	(0.25, 1)
scout2	red	(3.30, 6.52)	(0.25, 2.5)

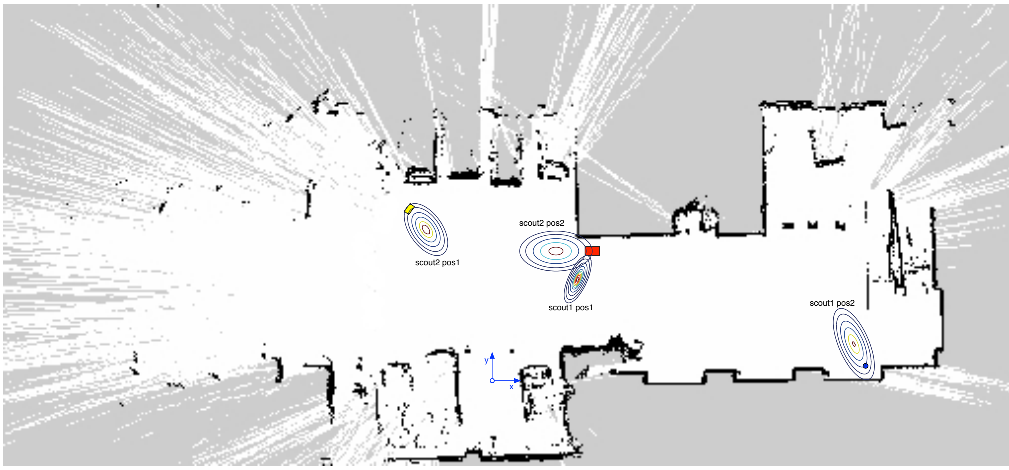
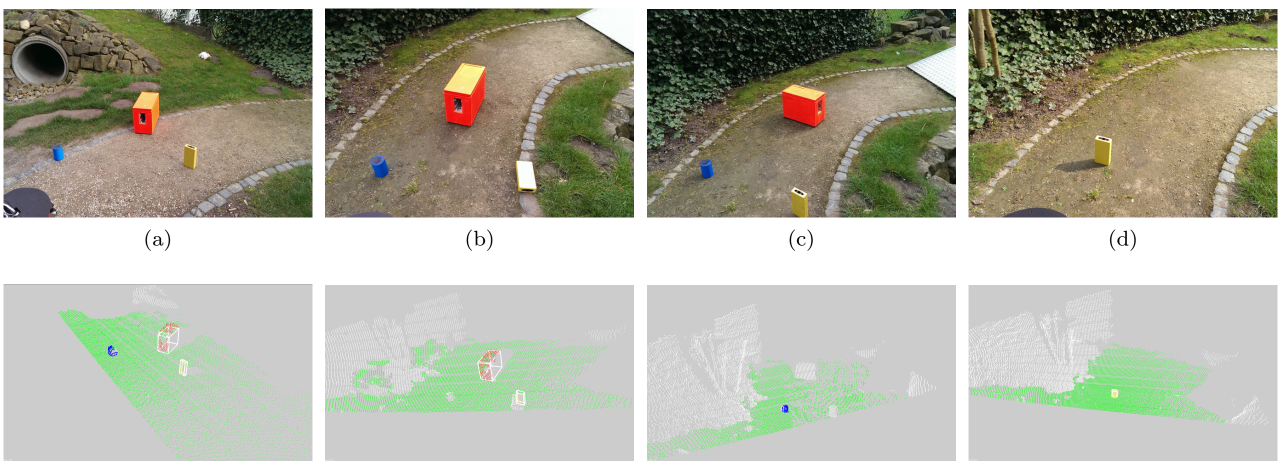


Figure 7. : Experiment map with object candidates' messages.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8. : Experimental setup for the 3D FuzzyDL perception method described in this paper. The objects were arranged in different locations, distanced and orientations with respect to the perceiving AMPARO robot. The corresponding detection results are given in e)-h). The detected ground plane is visualized in green and the objects in their corresponding color including the detected bounding box.

objects using the FuzzyDL-based semantic perception approach described in this paper. Figure 8 show the experimental setup. To verify the approach, several 3D point cloud scans have been recorded and the object features have been extracted using the approach described in Section 4. The A-Box of the ontology has been filled with the extracted spatial entities, consisting of planar clusters, cylindrical clusters, and also undefined shapes which do not match any object. Table 2 gives the perception and the FuzzyDL reasoning results. Table 2 gives the number of potential entities extracted from the 3D point cloud. The overall likelihoods are given after the reasoning is completed for the objects "base station" (BS), "sample container" (SC) and "battery" (BAT). For the base station, the number of detected components is given (e.g. slot, number of matched panels). The BS and BAT components describe how many spatial entities do match with the object components (i.e. sides or slot). The maximum likelihood is given after the reasoner matched the shape and spatial relationship with the object ontology. Therefore the number of potential components can be large, before the reasoner filters mismatched features.

6 Conclusions

In this paper we described our approach of using a multi-robot system to identify objects within a lunar environment. As an example, three objects

Table 2. : Perception results using spatial feature extraction and spatial reasoning based on Fuzzy Description Logic. The scenes are corresponding to Figure 8

Scene	Entities	BS components	BS score	BAT components	BAT score	SC score
(a)	3	2	0.77	1	0.76	0.92
(b)	21	7	0.94	25	0.97	0.0
(c)	24	5	0.0	55	0.0	0.83
(d)	25	3	0.0	12	0.84	0.0

were provided. The spatial features of the objects are given in a Fuzzy Description logic based ontology. A team of mobile robots explores the area and identifies the objects based on different sensor modalities. The interaction between the systems and the exchange of the knowledge base about the objects is provided by a reliable DDS (Data Distribution Service) interface which implicitly prevents data loss. The semantic classification of three objects has be verified during experiments. In a next step we will extend the semantic perception approach to more basic shapes as well as free-form shapes. The reasoner currently relies on the fact, that the objects are compound objects consisting of planes, rectangular shapes and cylindrical objects. An upcoming research topic is the classification of free form shapes, such as NURBS (Non-Uniform Rational B-Splines) or free form shapes consisting of basic shapes and NURBS.

Acknowledgment

This work is partially supported by the IMPERA project and the INCASS project (MOVE-FP7-605200-INCASS). The project IMPERA is funded by the German Space Agency (DLR, Grant number: 50RA1111) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament.

References

[1] F. Bobillo and U. Straccia. fuzzyDL: An expressive fuzzy description logic reasoner. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008.*, pages 923–930. Ieee, June 2008.

[2] M. Eich, M. Dabrowska, and F. Kirchner. Semantic Labeling: Classification of 3D Entities Based on Spatial Feature Descriptors. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, Alaska, USA, 2010. best of robotics.

[3] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, and J. Wollenberg. Towards Coordinated Multirobot Missions for Lunar Sample Collection in an Unknown Environment. *Journal of Field Robotics*, 31(1):35–74, 2014.

[4] M. Eich and F. Kirchner. Reasoning About Geometry: An Approach Using Spatial-Descriptive Ontologies. In K. Schill, B. Scholz-Reiter, and L. Frommberger, editors, *In Workshop Artificial Intelligence and Logistics (AILog), (ECAI-10)*, Lisbon, Portugal, 2010.

[5] C. Galindo, J. Fernández-Madrigal, J. González, and A. Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[6] R. Hartanto and M. Eich. Reliable, Cloud-based Communication for Multi-Robot Systems. In *Technologies for Practical Robot Applications (TePRA), 2014 IEEE International Conference on*, Apr. 2014.

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

[8] O. M. G. OMG. Data Distribution Service for Real-time Systems. Technical Report OMG Available Specification formal/07-01-01, OMG, 2006.

[9] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, Kobe, Japan, 2009. IEEE.

[10] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[11] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga. Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*, pages 1–6. IEEE Press, 2007.