Graph-Based Visual Semantic Perception for Humanoid Robots

Markus Grotz, Peter Kaiser, Eren Erdal Aksoy, Fabian Paus and Tamim Asfour

***Abstract*— Semantic understanding of unstructured environments plays an essential role in the autonomous planning and execution of whole-body humanoid locomotion and manipulation tasks. We introduce a new graph-based and data-driven method for semantic representation of unknown environments based on visual sensor data streams. The proposed method extends our previous work, in which loco-manipulation scene affordances are detected in a fully unsupervised manner. We build a geometric primitive-based model of the perceived scene and assign interaction possibilities, i.e. affordances, to the individual primitives. The major contribution of this paper is the enrichment of the extracted scene representation with semantic object information through spatio-temporal fusion of primitives during the perception. To this end, we combine the primitive-based scene representation with object detection methods to identify higher semantic structures in the scene. The qualitative and quantitative evaluation of the proposed method in various experiments in simulation and on the humanoid robot ARMAR-III demonstrates the effectiveness of the approach.**

I. Introduction

Autonomous robots that perform whole-body locomotion and manipulation tasks in partially or fully unknown environments require an accurate and detailed scene model for a successful operation. Examples for such loco-manipulation actions include grasping handrails, leaning against walls or using other structures for support. In cluttered or partially destructed environments, where whole-body loco-manipulation is particularly important, the actual scene can largely differ from the expected situation. Hence, the autonomous perception and representation of the robot's environment is a crucial prerequisite for reliable and robust execution of loco-manipulation in unknown environments. To tackle this issue, we propose a novel method for improved semantic scene representation by iteratively fusing scene representations obtained from consecutive RGB-D images.

In our previous work [1], we proposed a perceptual pipeline for the detection of loco-manipulation affordances in unknown environments. In this pipeline, mid-level segmentation and primitive extraction steps construct a simplified environment model in terms of geometric primitives, which is then used for reasoning about possible affordances. The pipeline proposed in [1] has been successfully implemented on real humanoid robot platforms and proofed to provide reasonable information for the detection of sophisticated loco-manipulation affordances.

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no 611832 (WALK-MAN).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. markus.grotz@kit.edu

Fig. 1: The humanoid robot ARMAR-III performing a bimanual whole-body grasp to lift a box. The scene representation is iteratively fused from consecutive RGB-D images. Furthermore, our reasoning process allows for identification of higher semantic structures, i.e. structures composed of several geometric primitives. This allows to automatically extract and associate more complex actions, e.g. a bimanual lift, with the geometric primitives.

In this work, we extend the existing perceptual pipeline with geometric and semantic reasoning steps to further improve the quality of the pipeline's results. The goal is to reduce the number of time-consuming capturing processes required for obtaining an accurate and rich environment representation. This is of direct use for subsequent steps in the pipeline, such as the detection of loco-manipulation affordances. Since the geometric primitive extraction is computationally expensive for large scenes, our method performs iterative primitive extraction and subsequent fusion with primitives obtained from consecutive frames. This increases the data availability since partial results can be processed immediately. The proposed pipeline further employs a strategy to model spatio-temporal relationships between extracted geometric primitives: While the spatial relationship is particularly useful if a scene is only partially visible to the robot and essential information is obstructed, the temporal relationship can be incorporated for tracking changes over time, e.g. in order to perceive the results of manipulation actions. Finally, the proposed pipeline contains methods for the identification of *higher semantic structures* in the arrangements of geometric primitives. This problem is approached by combining a purely data-driven approach with an object detection system based on the open-source library

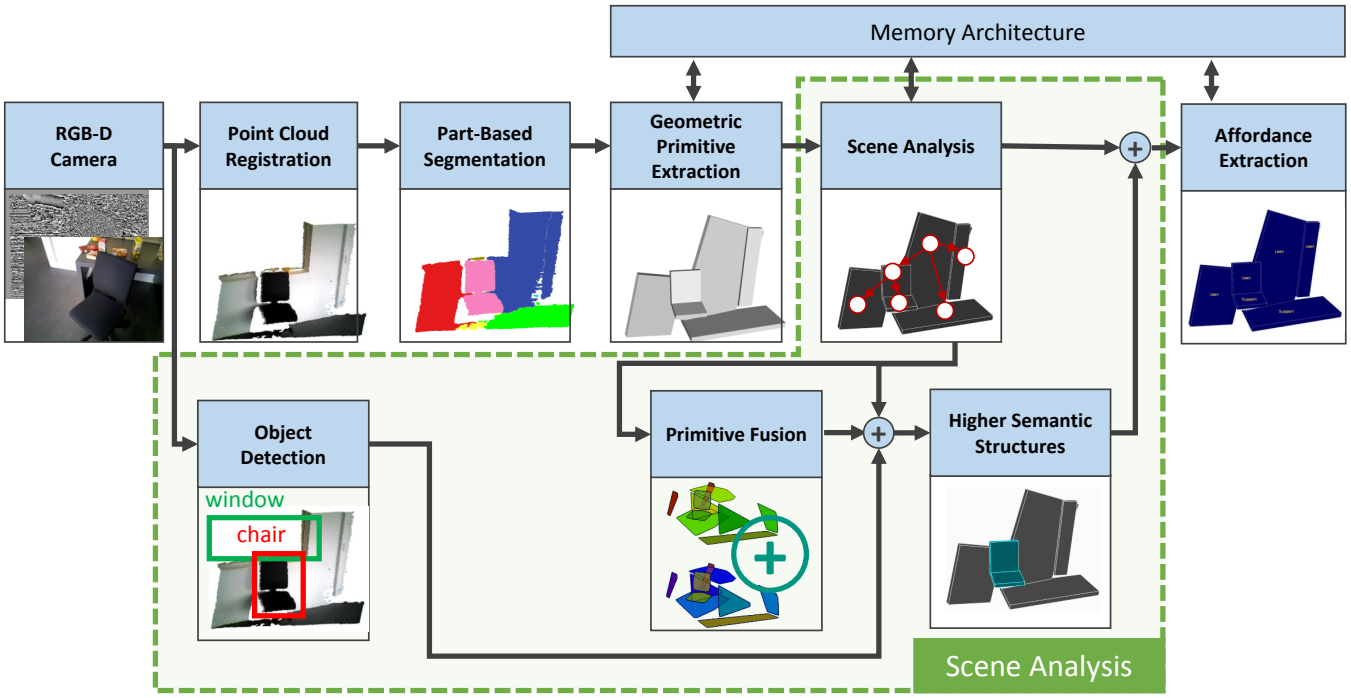

Fig. 2: The perceptual pipeline and their different components. The pipeline creates a model of the current environment using geometric primitives based on a 3D point cloud input. Additional components include, for example, the extraction of affordance hypotheses or the estimation of grasp points. In this work, the pipeline has been extended by a scene analysis component and a stage to identify structures composed of multiple geometric primitives. New components are highlighted with a dashed box.

You only look once (YOLO) [2].

The remainder of this work is structured as follows. After discussing related approaches to semantic scene understanding in section II, the proposed perceptual pipeline is introduced and explained in great detail in section III. Section IV evaluates the proposed methods in multiple experimental setups in simulation and on the humanoid robot platform ARMAR-III. Finally, section V concludes the paper and discusses future work.

II. Related Work

A variety of approaches to autonomous scene perception have been proposed in the literature. One way to model the environment is to use occupancy maps [3]. This allows for easily modeling the free space and is often used for collision avoidance. In this context, however, we aim at leveraging the environment for whole-body locomotion and manipulation tasks as demonstrated in [4] and [1]. Thus, we are interested in creating detailed and rich geometric models of the environment.

Schnabel et al. [5] extend their previous work [6] by decomposing a scene into geometric primitives. Their method extracts planar or cylindrical primitives from registered LIDAR point clouds. Neighborhood relations between the geometric primitives are represented within a graph structure. In this graph, the set of geometric primitives matched to the point cloud represents the vertices. Edges are added to the scene graph for adjacent primitives. Based on geometric neighborhood relations, this graph structure can then be used to identify higher semantic structures, e.g. roof tops or stairs. The authors also present an algorithm for efficiently querying the scene graph. In contrast to [6], the methods proposed in this work go beyond a rule-based model and incorporate semantic and temporal relationships between the geometric primitives.

Rusu et al. present a way to model object maps for household environments based on 3D laser scans [7]. Their solution features several filtering steps, in which the input point cloud is smoothed and cuboid structures are extracted using a geometric and functional reasoning process. Detected cuboids are hierarchically structured and stored in a multi-dimensional tuple representation.

Gupta et al. [8] propose a method for physical outdoor scene representation based on 2D image data. A 3D block-based interpretation of the scene is derived from the input images. These blocks are further enriched with simple relationships and based on estimated depth relationships, candidates for splitting and fusing are proposed. To join primitives that are separated by occluding obstacles, the authors use volumetric constraints as hints. In contrast to Gupta et al., we directly obtain 3D information by employing depth sensors instead of relying on depth cues. Finally, our method is not limited to blocks, but also supports other geometric primitives such as spheres and cylinders as shown in our previous work. This allows us to obtain a more accurate description of the environment.

Another research field focuses on indoor environments and recovering the building structure. In this case, it is possible to exploit the Manhattan structure to extract planar surfaces from single images (e.g. [9], [10]). A coherent floor plan is obtained by refining extracted planes and optimizing the structure using dynamic programming. This idea was later extended by Taylor et al. in [11] to RGB-D images. Recently, Pham et al. presented in [12] an unsupervised segmentation algorithm for plane extraction. The segmentation is refined using a scene graph to exploit an indoor metric. While their proposed method achieves slightly better results in their evaluation than the LCCP segmentation, which is presented in [13] and is used in this work, their approach possess higher computational costs. In order to deal with this issue, the authors suggest that the results can be propagated incrementally to speed up the segmentation process. In this work, we will describe a method for fusing the results from consecutive frames. However, we are not dealing with segmentation but rather with a scene model and affordance extraction. Finally, we will not limit ourself to planes and also include other geometric shapes, such as cylinders and spheres.

III. The Perceptual Pipeline

In the following, we provide a complete overview over the proposed perceptual pipeline, highlighting differences to our previous work [1]. An overview of the system architecture is illustrated in Fig. 2, where new components, introduced in this paper, are highlighted in green.

The principles of the proposed pipeline can be summarized as follows. The robot's current view is captured with a RGB-D sensor and registered, resulting in a 3D point cloud representation. This point cloud and the corresponding input images are now processed in two parallel processes: *geometric primitive extraction* using our previously proposed pipeline [1] (section III-A) and *object detection* using the open source library YOLO [2]. Extracted primitives are further processed in a *scene analysis* step, which models the spatial relationship between the extracted primitives in a graph structure (section III-B). We then exploit the spatio-temporal relationship between the primitives to fuse existing primitives with those coming from previous frames, already stored in the memory (section III-C). By combining the fused geometric primitives with the output of an object detection system and an alternative, purely data-driven approach, we identify *higher semantic structures*, i.e. semantically coherent structures composed of multiple geometric primitives (section III-D). In the following, the individual components of the proposed pipeline outlined in Fig. 2 are discussed in further detail.

A. Geometric Primitive Extraction

In the geometric primitive extraction stage, the RGB-D image is first registered and the associated point cloud is segmented into plausible parts using the LCCP segmentation algorithm [13]. Other segmentation algorithms, such as region growing or euclidean clustering, are also applicable in this stage. The *part-based segmentation* component decomposes the point cloud of the current view P^t into disjoint segments $P^t_{s_i}$, such that:

$$P^t = \bigcup_i P^t_{s_i}, \tag{1}$$

where t is the temporal parameter, indicating the current frame. In the next step, geometric primitives are iteratively fitted to each segment in the point cloud using a customized approach based on the implementation provided by the widely used *point cloud library* (PCL) [14]. The set of considered geometric shapes mainly includes planes, but also comprises cylinders and spheres. Formally, this step allows to decompose the current view into a set of geometric primitives

$$\Psi^t = \{\psi^t_1, \ldots, \psi^t_{m_t}\}. \tag{2}$$

For brevity, we will omit the temporal parameter t if it can be derived from the context. Each geometric primitive ψ_i is linked to an inlier point cloud $P_{\psi_i} \subset P_{s_i}$ of the corresponding segment s_i. As pointed out by [6] it is important to distinguish between the segment and the inlier point cloud. Further properties of the geometric primitives ψ_i, such as the oriented bounding box $\mathrm{OBB}(\psi_i)$ are computed in this stage. An exemplary scene together with the corresponding geometric primitives is illustrated in Fig. 3.

Fig. 3: *Left*: An exemplary point cloud of a cluttered tabletop scene. *Right*: Extracted geometric primitives. Higher semantic structures, e.g. boxes, are visualized with a gray volume.

B. Scene Graph Representation

In the *scene analysis* stage of the perceptual pipeline, we model the primitive relationship in a graph structure similar to Schnabel et al. [5] and the following work of Berner et al. [15]. In the following, we assume that only planes exists in the scene. However, the methods can be easily extended to other geometric shapes which can be extracted by the pipeline, such as cylinders or spheres.

In order to construct the graph, a distinguished geometric primitive needs to be selected as root, which in our approach is defined to be the floor plane. We identify the floor plane as the plane with the lowest height w.r.t. the robot:

$$\rho = \arg\min_{p \in S} \mathrm{height}(p). \tag{3}$$

By iteratively checking for primitive intersections, we obtain a graph-based scene representation, in which the nodes n_i represent primitives $\psi_i \in \Psi$. Edges between pairs of

nodes n_i and n_j, or conceptually between their associated geometric primitives ψ_j and ψ_i, are added to the graph iff.

$$\text{OBB}(\psi_i) \cap \text{OBB}(\psi_j) \neq \emptyset. \tag{4}$$

In practical applications, the bounding boxes $\text{OBB}(\psi_i)$ and $\text{OBB}(\psi_j)$ are slightly extended by $\varepsilon > 0$ in order to account for perceptual inaccuracies.

To avoid exhaustive intersection tests, we employ frustum culling and thus consider only primitives in the field of view for comparison with primitives Ψ^t obtained from the robot's current view. An exemplary set of primitives with a visualization of the obtained scene graph is depicted in Fig. 4.

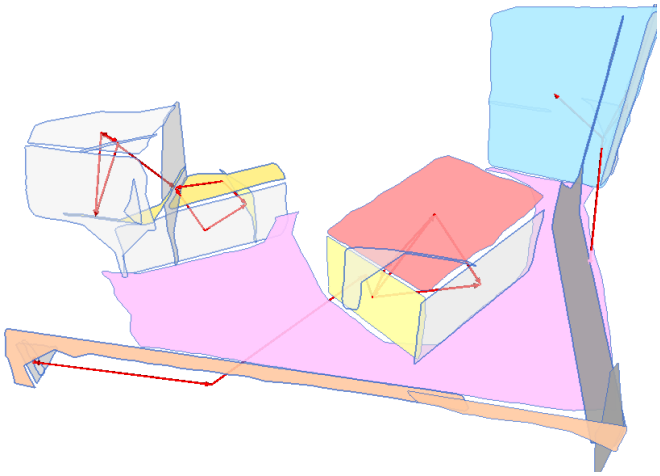

Fig. 4: Exemplary set of geometric primitives and the visualization of an associated spatial scene graph (red).

C. Spatio-temporal Fusion of Geometric Primitives

Primitives Ψ^{t_1} and Ψ^{t_2} resulting from multiple individual pipeline runs at times t_1 and t_2 cannot be regarded as entirely independent. For example, consistent primitives which are too large for the robot's field of view will result in a segmented arrangement of multiple smaller primitives detected at different points in time. A fusion step of geometric primitives is required in order to match multiple views.

Given the scene graph for a set of existing primitives Ψ^t, we propose to fuse Ψ^t with primitives Ψ^{t+1} from the current view based on an inlier ratio $\text{Inlier}(\psi_i, \psi_j)$, defined as:

$$\text{Inlier}(\psi_i, \psi_j) = \frac{1}{|P^{t+1}_{\psi_i}|} |P^{t+1}_{\psi_i} \cap \text{OBB}(\psi_j)|, \tag{5}$$

where $|\cdot|$ denotes the set-cardinality. In practical applications, we slightly extent the oriented bounding boxes by ε to take also inaccuracies into account[1].

Algorithm 1 outlines our approach to the spatio-temporal fusion of geometric primitives. Once a new set of primitives Ψ^{t+1} arrives, the present, previously fused, primitives Ψ are retrieved from the robot's memory and culled according to the robot's current field of view. For each pair of primitives

[1]In our experiments: $\varepsilon = 5\,\text{cm}$.

Algorithm 1: Spatio-temporal fusion of geometric primitives

Data: New Primitives Ψ^{t+1}
Result: Set of fused primitives

$\Psi \leftarrow$ ReadPrimitivesFromMemory()
$\Psi \leftarrow$ FrustrumCulling(Ψ)
foreach $\psi \in \Psi$ **do**
  foreach $\varphi \in \Psi^{t+1}$ **do**
    if $OBB(\varphi) \cap OBB(\psi) = \emptyset$ **then**
      continue
    if *not compareModelParameters(φ, ψ)* **then**
      continue
    if $\varphi \subset \Psi$ **then**
      remove primitive φ from Ψ^{t+1}
    else if $Inlier(\varphi, \psi) > \lambda_o \wedge\ Inlier(\psi, \varphi) < \lambda_p$ **then**
      remove φ
    else if $Inlier(\psi, \varphi) > \lambda_o \wedge\ Inlier(\varphi, \psi) < \lambda_p$ **then**
      remove ψ
foreach $\varphi \in \Psi^{t+1}$ **do**
  addToMemory(φ)

$\psi \in \Psi$ and $\varphi \in \Psi^{t+1}$, we first cheaply test if the primitives are overlapping at all by testing if the oriented bounding boxes $\text{OBB}(\psi)$ and $\text{OBB}(\varphi)$ are intersecting.

If two primitives ψ and φ are intersecting we evaluate the inlier ratio from Equation 5. This ratio expresses the degree of coverage between two primitives. Once the degree of coverage between ψ and φ exceeds a threshold λ_o, the covered primitive is removed from the robot's memory. To prevent the removal of partially overlapping primitives we additionally check the number of inliers between φ and ψ. The threshold is denoted by the lower bound λ_p. This step is necessary since the oriented bounding box can be larger than the actual primitive. In this context, we set $\lambda_o = 0.7$ and $\lambda_p = 0.3$. We further check the model parameters (e.g. plane normals, position) to test if both primitives belong to the same real world segment.

D. Higher Semantic Structures

Ultimately, we aim at increasing the robot's autonomy by automatically extracting whole-body locomotion and manipulation actions. Therefore, we need to identify complex structures composed of multiple geometric primitives in order to reason about possible, complex ways of interaction. We rely on two different approaches to identify higher semantic structures in the scene: A purely data-driven approach based on the scene graph representation outlined in section III-B and a semantic approach based on an object detection system.

1) Data-Driven Approach: Our first approach follows the work of Schnabel et al. [5]. By exploiting the spatial relationship between the primitives higher semantic structures can be detected using a purely data-driven approach. For

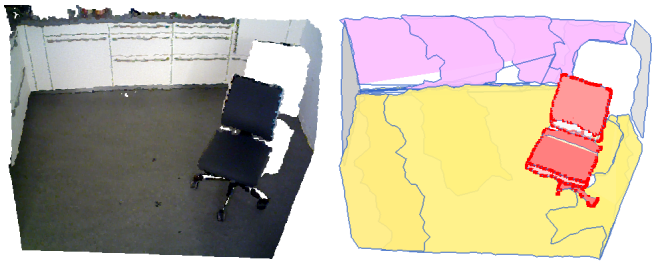

Fig. 5: Higher semantic structures extracted from geometric primitives. *Left*: The input point cloud. *Right*: Extracted geometric primitives. Primitives corresponding to the detected chair are highlighted in red.

example, to detect boxes in the environment, we search for three perpendicular planes. Albeit being practical, this process requires a rule for every shape composed of several primitives.

2) Semantic Approach: In parallel to the extraction of geometric primitives, the input RGB image and the associated point cloud are passing an object detection stage. This stage is implemented as a wrapper for the open source library YOLO [2], which provides a state-of-the-art object detection system with real-time capabilities. Given a RGB image, YOLO provides geometric and semantic information for detected objects in terms of 2D bounding boxes and semantic labels describing object classes. To this end, we utilize the geometric information provided by YOLO as hints for higher semantic structures in the scene.

3) Fusion of the Results: To fuse the output of these two detection methods, we first convert the output of the object detection library to a labeled point cloud by projecting the 2D object class bounding boxes to the original 3D point cloud. Due to the fact that the sets of geometric primitives are extracted from the same RGB-D images, we can cluster the primitives by testing if they intersect with the object class bounding box, according to the inlier ratio defined in Equation 5. This additional step allows us to attribute higher semantic labels to primitive structures detected by YOLO, in a way that is consistent with the data-driven approach discussed above.

Fig. 5 illustrates an example of higher semantic structures detected in an exemplary point cloud. In this scene, the chair has been detected as a higher semantic structure via the object detection approach, while large and consistent primitives for the floor and the cupboards are the result of spatio-temporal primitive fusion as described in section III-C. An example for the data-driven detection of higher semantic structures can be seen in Fig. 9, where box structures were successfully detected in an experiment with the humanoid robot ARMAR-III [16] .

IV. Evaluation

To evaluate the proposed pipeline for semantic scene perception, we will first investigate the accuracy of the spatio-temporal fusion of geometric primitives by comparing primitives fused from real RGB-D camera data to manually labeled ground truth segmentations. The ground truth has been generated by extracting a manually labeled point cloud $G = \{G_1 \dots G_m\}$ from a simulated model of the environment, with mutually disjoint segments G_i:

$$G_i \cap G_j = \emptyset \;\; \forall i \neq j. \tag{6}$$

For each bounding box $\mathrm{OBB}(G_i)$ of a ground truth segment G_i, we select the primitive $\psi_j \in \Psi$ that maximizes the number of inliers as specified by the inlier ratio defined in Equation 5:

$$\psi(G_i) = \arg\max_j \mathrm{Inlier}(G_i, \Psi_j). \tag{7}$$

To quantify the quality of extracted geometric primitives Ψ compared to a ground truth segmentation G, the *inlier index* is defined as the average ground truth coverage:

$$\mathrm{InlierIndex}(G, \Psi) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Inlier}(G_i, \psi(G_i)). \tag{8}$$

Since this metric only captures over-segmentation, we further use the 2D overlapping criteria used by [17]. However, since we aggregate the results from multiple views the data cannot be re-projected into a 2D image. Therefore, we adapt the overlapping criteria to work with 3D labeled point clouds:

$$\mathrm{OverlappingIndex}(G, \Psi) = \frac{1}{m} \sum_{i=1}^{m} \max_j \frac{|G_i \dot{\cap} \Psi_j|}{|G_i \cup \Psi_j|}. \tag{9}$$

In order to account for perceptual variances in point cloud data, the set-intersection operator $X \dot{\cap} Y$ from Equation 9, used to intersect two point clouds X and Y, is defined as[2]:

$$X \dot{\cap} Y = M_\varepsilon(X, Y) \cup M_\varepsilon(Y, X), \tag{10}$$

with

$$M_\varepsilon(X, Y) = \{x \in X | \exists\, y \in Y : \|x - y\| \leq \varepsilon\}. \tag{11}$$

A. Experiment 1: Spatio-Temporal Primitive Fusion

An outline of the experimental setup is shown in Fig. 8. A total of 58 point clouds resembling a kitchen environment were captured from different angles using the sensory equipment[3] of the humanoid robot ARMAR-III. The captured point clouds were sequentially processed by the perceptual pipeline proposed in this work. The corresponding values of *InlierIndex* and *OverlappingIndex* for the geometric primitives aggregated from consecutive frames over time are plotted in Fig. 6. Since the robot constantly moves, and hence new parts of the scene become visible over time, the number of inliers is monotonically increasing until the scene is fully covered. Over time, both indices converge towards a value of 1, indicating that the result of the spatio-temporal fusion of iteratively detected geometric primitives eventually accurately resembles the ground truth primitives. Fig. 7 displays the number of removed and modified primitives over time in the same experimental setup. It can be seen that the number of primitive modifications drops after the scene is

[2] In our experiments: $\varepsilon = 5\,\mathrm{cm}$.

[3] In this case the ASUS Xtion Pro.

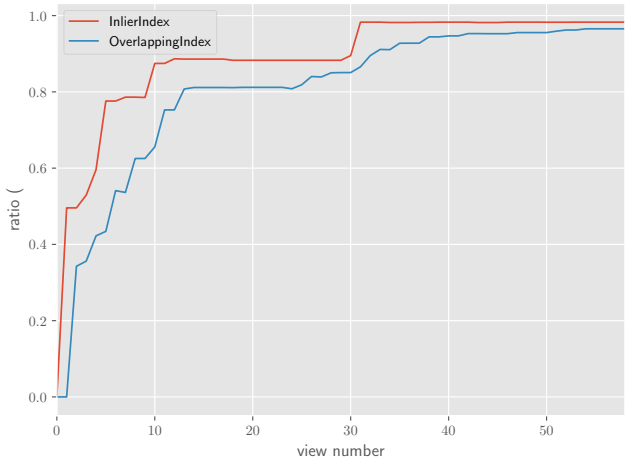

Fig. 6: The plot visualizes the inlier ratio (Equation 8) and the overlapping index (Equation 9) over time. After 30 views most of the scene is mapped with geometric primitives. In the following views the scene model is refined as one can see from the increasing *OverlappingIndex* value.

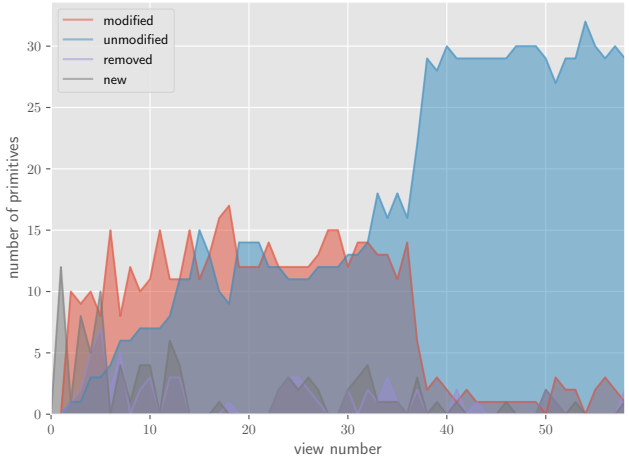

Fig. 7: Number of primitives modified, removed and added. After 40 views the scene is almost fully covered. Thus, only a few geometric primitives are added. Re-detected primitives are updated.

reasonably well covered by the aggregated primitive model, indicating that the method for spatio-temporal primitive fusion eventually reaches an equilibrium state, in which additional views of the scene have only little influence on the aggregated primitive model.

B. Experiment 2: Higher Semantic Structures

In a final experiment, we demonstrate the applicability of the proposed pipeline for semantic scene perception in real world applications using the humanoid robot ARMAR-III. In the experimental setup, ARMAR-III is located in a kitchen environment, standing in front of a cluttered tabletop scenario with multiple boxes. A pilot interface for shared autonomous control of ARMAR-III based on detected geometric primitives and affordances [18] is employed for visualizing the robot's perceptual information and for initiating actions. Once the environment representation is available, the pilot is presented a 3D visualization of the aggregated geometric primitives, including detected box-shaped primitives as higher semantic structures (see Fig. 9, top row). In this case, the data-driven approach for the detection of box-shaped primitives (see section III-D) was employed. Once, the pilot interactively selects a box primitive and its associated bimanual *liftability* affordance, the robot autonomously executes the associated skill (see Fig. 9, bottom row).

V. Conclusion and Future Work

We presented a novel approach to semantic scene perception based on the detection of geometric primitives using RGB-D sensors in unknown environments. The resulting extended perceptual pipeline comprises the construction of a graph-based scene representation based on neighborhood relations among primitives, the spatio-temporal fusion of primitives iteratively detected in point cloud sequences, and the detection of higher semantic structures based on geometric primitives and the object detection system YOLO. The evaluation shows that the perceptual pipeline is able to construct accurate and stable geometric primitive representations from real sensor data. We further demonstrated the applicability of the proposed approach within the perceptual pipeline in the context of humanoid robot tasks, showing the shared autonomous control of ARMAR-III and the detection and utilization of bimanual *liftability* affordances based on the geometric primitives obtained by the pipeline proposed in this work.

In future work, we will further investigate the detection of higher semantic structures, particularly with respect to their associated affordances in the context of autonomous and shared autonomous operation of humanoid robots. Another important direction will also be the integration of active vision methods, which allow a robot to selectively plan next-best-views for improving the internal primitive model.

References

[1] P. Kaiser, M. Grotz, E. E. Aksoy, M. Do, N. Vahrenkamp, and T. Asfour, "Validation of whole-body loco-manipulation affordances for pushability and liftability," in *International Conference on Humanoid Robots (Humanoids)*, pp. 920–927, 2015.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, IEEE, 2016.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[4] D. I. Kim and G. S. Sukhatme, "Semantic labeling of 3d point clouds with object affordance for robot manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5578–5584, IEEE, 2014.

[5] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, "Shape recognition in 3d point-clouds," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, UNION Agency-Science Press, 2008.

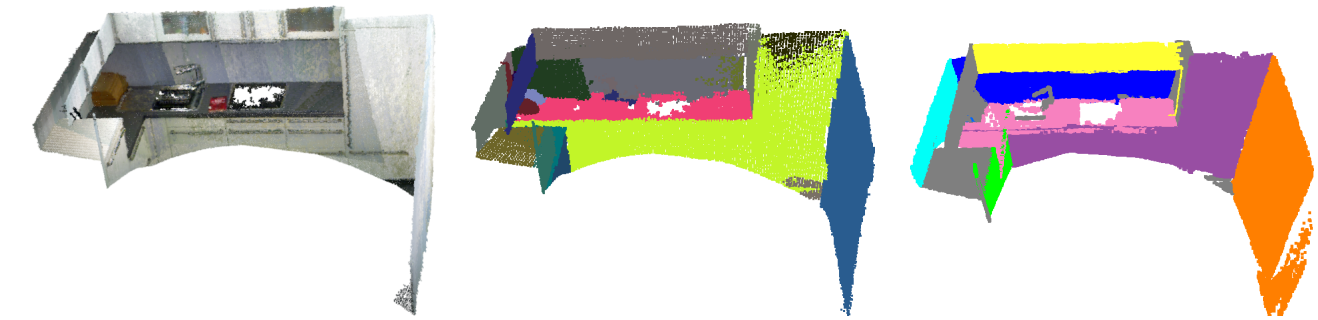

Fig. 8: Kitchen scenario as described in section IV-A. *Left*: The image shows the registered result of the source point cloud. *Center*: Multiple views are first registered and then geometric primitives are extracted. *Right*: Primitives are extracted iteratively from consecutive frames.

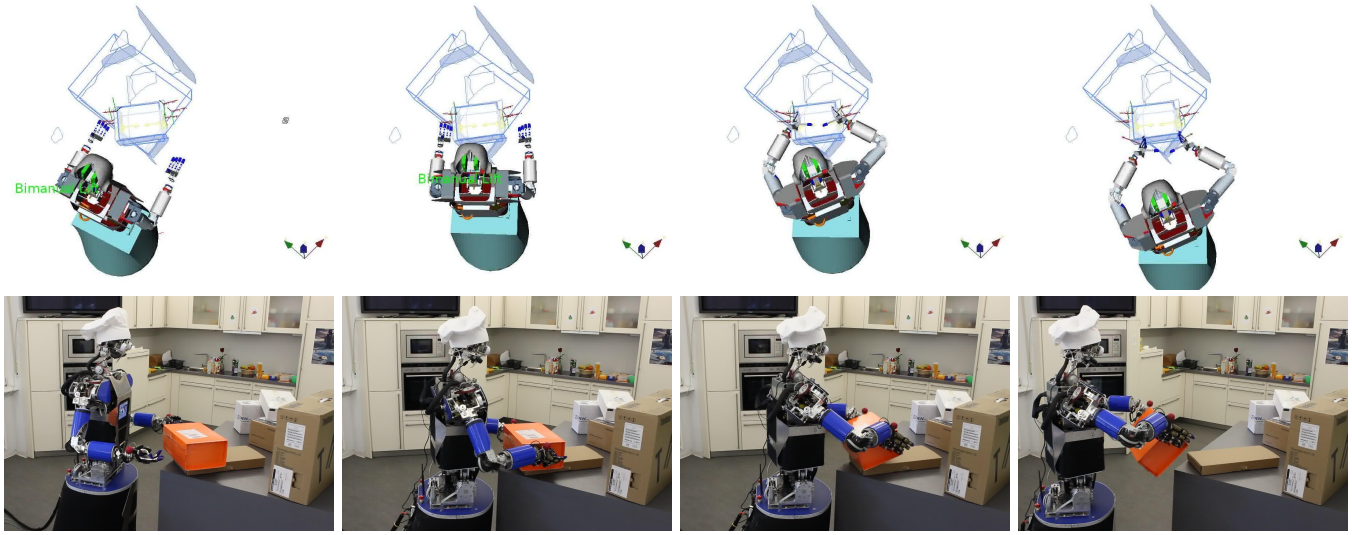

Fig. 9: ARMAR-III bimanually lifting a box. Geometric primitives are extracted from the environment. Using spatial and semantic reasoning the geometric primitives are aggregated into higher semantic structures. The top row shows the scene representation at four different timestamps as presented to the pilot. The bottom row shows the actual environment.

[6] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, pp. 214–226, June 2007.

[7] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[8] A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *Computer Vision–ECCV*, pp. 482–496, Springer Berlin Heidelberg, 2010.

[9] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2136–2143, IEEE, 2009.

[10] A. Flint, C. Mei, D. Murray, and I. Reid, "A dynamic programming approach to reconstructing building interiors," in *Computer Vision–ECCV*, pp. 394–407, Springer Berlin Heidelberg, 2010.

[11] C. Taylor and A. Cowley, "Parsing indoor scenes using rgb-d imagery," in *Robotics: Science and Systems*, July 09-13, 2012.

[12] T. T. Pham, M. Eich, I. Reid, and G. Wyeth, "Geometrically consistent plane extraction for dense indoor 3d maps segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4199–4204, IEEE, 2016.

[13] S. C. Stein, F. Worgotter, M. Schoeler, J. Papon, and T. Kulvicius, "Convexity based object partitioning for robot applications," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3213–3220, 2014.

[14] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4, 2011.

[15] A. Berner, J. Li, D. Holz, J. Stuckler, S. Behnke, and R. Klein, "Combining contour and shape primitives for object detection and pose estimation of prefabricated parts," in *IEEE International Conference on Image Processing (ICIP)*, pp. 3326–3330, 2013.

[16] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-iii: An integrated humanoid platform for sensory-motor control," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 169–175, 2006.

[17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV*, pp. 746–760, Springer Berlin Heidelberg, 2012.

[18] P. Kaiser, D. Kanoulas, M. Grotz, L. Muratore, A. Rocchi, E. M. Hoffman, N. G. Tsagarakis, and T. Asfour, "An affordance-based pilot interface for high-level control of humanoid robots in supervised autonomy," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 621–628, IEEE, 2016.