Fast segmentation of RGB-D images for semantic scene understanding

Dirk Holz, Alexander J. B. Trevor, Michael Dixon, Suat Gedikli and Radu B. Rusu

I. Introduction

As robots move away from pre-programmed action sequences in controlled laboratory setups towards complex tasks in real-world scenarios, both the perception capabilities of these systems and their abilities to acquire and model semantic information must become more powerful. In this context, fast means for pre-processing acquired sensory information and segmenting task-relevant regions are an enabling technology and a prerequisite for avoiding longer delays in sense-plan-act cycles.

We present (on a poster and in a demo) two fast segmentation methods for RGB-D images.

II. Approximate Meshes for Pre-processing

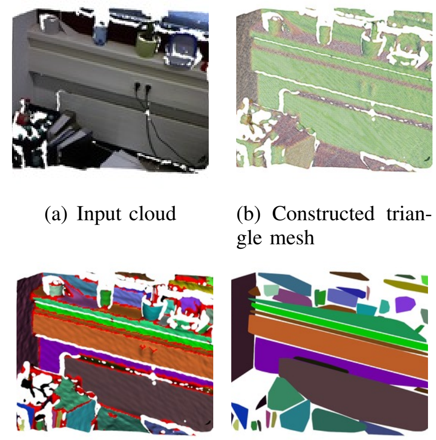
In the first method, we exploit the organized structure of RGB-D images and apply an approximate surface reconstruction [1] by simply connecting adjacent image pixels. The resulting mesh efficiently caches local neighborhoods for further processing. Furthermore, we compute approximate surface normals directly on the mesh and apply a multi-lateral filtering step to considerably smooth the data. Using an efficient region growing implementation and different region models, we can efficiently compute plane segmentations and full polygonalizations (Fig. 1), or segment locally smooth regions and detect geometric shape primitives (Fig. 2).

III. Direct Image Segmentation

The second method also exploits the image structure by using a connected component based technique. Each pixel is compared to neighboring pixels (in a 4-connected sense) using a comparison function (similar to the region models above). Points are considered part of the same segment if the comparison function returns true. Different comparison functions can be used for different segmentation tasks, such as a plane equation comparison (the dot product between normals and range must match), euclidean distance, color, or combinations of these. These can be run in a sequence and with an optional mask, for example: tabletop objects can be detected by first detecting planar regions, then using these regions as a mask and segmenting with a euclidean distance comparison (Fig. 3). See the Point Cloud Library [2] at www.pointclouds.org for further details, documentation, and an open source implementation.

D. Holz is with the University of Bonn, Bonn, Germany. A. J. B. Trevor is with the Georgia Institute of Technology, Atlanta, Georgia, USA. M. Dixon, S. Gedikli and R. B. Rusu are with Willow Garage, Inc., California, USA.
Contact: holz@ais.uni-bonn.de, atrevor@cc.gatech.edu

A demonstration video of the approach is available at http://youtube.com/watch?v=LZ8l4w3qw3E.



(a) Input cloud (b) Constructed triangle mesh (c) Segmentation (d) Polygonalization

Fig. 1: Approx. surface reconstruction and segmentation.

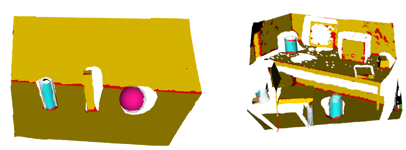


Fig. 2: Detecting geometric shape primitives: planes (yellow), cylinders (cyan) and spheres (magenta).

Fig. 3: Planes and connected components.

References

[1] D. Holz and S. Behnke, “Fast Range Image Segmentation and Smoothing using Approximate Surface Reconstruction and Region Growing,” in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, Jeju Island, Korea, 2012.

[2] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 1–4.