Efficient Surface and Feature Estimation in RGBD

Zoltan-Csaba Marton, Dejan Pangercic, Michael Beetz
Intelligent Autonomus Systems Group, Technische Universität München, {marton, pangercic, beetz}@cs.tum.edu

***Abstract*— Extracting useful information from RGBD images at high frame-rates requires fast algorithms that are more than just individual steps in a pipeline. Ideally, they should be doing as much as possible in a single step, and re-use information from previous computations. We will present our first experiences with such a pipeline, consisting of open-source algorithms from the Point Cloud Library (PCL) of ROS, and detail the modifications and parameters used for easy reproduction and extension.**

I. Introduction

We perform object classification based on RGBD data, and present the efficient processing algorithms that are used to achieve it, as well as our freely available object database.

Our approach for efficient object classification is based on a database of object scans using the kinect sensor. We perform smoothing, normal estimation, surface radius estimation and feature extraction using PCL, sped up by the use of voxelization. As an application example, the locally labeled voxels are then used to calculate an RGBD feature for each object hypotheses, and classified using SVM.

We base our processing pipeline on a voxelized representation of the dataset, as this is needed by multiple other applications as well (collision avoidance, keeping track of free and unexplored space, etc.) and it allows for a speed-up proportional to the average number of points in the voxels. While groups of points can be formed in an organized dataset by extracting pixel neighborhoods as well, using voxelization ensures spacial closeness and similar volume.

A. Smoothing and Normal Estimation

In each voxel, the surface defined by the points can be estimated robustly using an MLS approach [1], and the cross product of two tangents of the fitted polynomial in a given point define the normal. Computing the points and normals once per voxel (by sampling multiple points on the surface) avoids nearest neighbor searches for each point and provides enough data to estimate the geometry, as detailed below.

B. Descriptive Local Geometric Features

The radius of the curves with the minimum and maximum curvature can be estimated as in [2], but based on the sampled points for each voxel. As these are metric values, they can be used intuitively to categorize the surfaces in the cells, which can be further used to build object-level features [3].

II. Application: RGBD Object Feature

Having voxels with both RGB and geometric features enables the simultaneous use of two features operating on voxel neighborhoods: GRSD [3] and Color-CHLAC [4]. The former can distinguish different shapes and the latter different color distribution. Their combination, called VOSCH, combines them to capture both geometry and color.

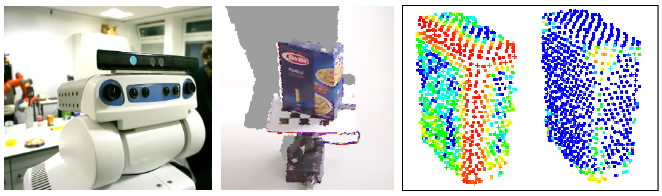

Fig. 1: Left: kinect mounted on PR2. Middle: example training point cloud of a box. Right: voxel centroids of the box colored first by the minimum, then by the maximum radius respectively. Red means 0 and blue 20 cm or higher (values are capped to avoid ∞ at planes).

By adapting the algorithm of the original two features to match well, VOSCH can be computed in a single step. Average computation time is 0.26 s per object (consisting of 4632 points on average). Initial classification results on 63 objects (many of them of the same type or brand) using SVM show 76.4% accuracy, with a classification time of 0.05 s per object. This can be improved further by marking each view as a separate class during training, but considering only the object type when interpreting the results. Initial tests using the less accurate Hokuyo improved the success rate by 12%.

The modular definition of PCL made it easy to use already defined algorithms in our application, and making the necessary extensions was straightforward as the project is open. There are other features in PCL that can be used as well, and some of them can be even adapted to work on voxels to achieve a speed-up and to combine them with the above two.

III. Results

Our results will be presented in a demo and the object database released for free use. The unreleased parts of the code will be published as part of PCL (`pointclouds.org`).

Acknowledgement: This work was supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence at the Technische Universität München and Willow Garage, Menlo Park, CA. We would also like to thank Asako Kanezaki, as well as all PCL developers, contributors and supporters.

References

[1] Z. C. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 12-17*, 2009.

[2] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D Modelling of Novel Objects from a Single View," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.

[3] Z.-C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010.

[4] A. Kanezaki, T. Harada, and Y. Kuniyoshi, "Partial matching of real textured 3D objects using color cubic higher-order local auto-correlation features", in *The Visual Computer*, 26(10):12691281, 2010.