3D Segmentation Evaluation Instructions

David Held
Computer Science Department, Stanford University
davheld@cs.stanford.edu

I. Introduction

This document describes how to reproduce the evaluation for 3D segmentation that was given in the paper: Held, David, et al. "A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues."

II. Evaluation

Dataset: We evaluate our segmentation method on the KITTI tracking dataset [1, 2, 3]. We use sequences 0001 and 0013 to train our method and select parameters and the remaining 19 sequences for testing and evaluation.

Although the KITTI tracking dataset has been made publicly available, the dataset has typically been used to evaluate only tracking and object detection rather than evaluating segmentation directly. However, segmentation is an important step of a 3D perception pipeline, and errors in segmentation can cause subsequent problems for other components of the system. Because the KITTI dataset is publicly available, we encourage other researchers to evaluate their 3D segmentation methods on this dataset using the procedure that we describe here.

Pre-processing As a pre-processing step, we remove the points that belong to the ground using the method of Montemerlo et al. [6]. Results may vary with different ground detection methods, but unfortunately, we are unable to release the code for this ground detection method at this time.

Evaluation Metric: The output of our method is a partitioning of the points in each frame into disjoint subsets (“segments"), where each segment is intended to represent a single object instance. The KITTI dataset has labeled a subset of objects with a ground-truth bounding box, indicating the correct segmentation. We wish to evaluate how well our segmentation matches the ground-truth for the labeled objects.

To evaluate our segmentation, we assign a best-matching segment to each ground-truth bounding box. For each ground-truth bounding box gt, we find the set of non-ground points within this box, C_{gt}. For each segment s, let C_s be the set of points that belong to this segment. We then find the best-matching segment to this ground-truth bounding box by computing

$$s = \arg\max_{s'} |C_{s'} \cap C_{gt}| \tag{1}$$

The best-matching segment is then assigned to this ground-truth bounding box for the evaluation metrics described below.

We describe on the project website how the intersection-over-union metric on 3D points [11] is non-ideal for autonomous driving because this score penalizes undersegmentation errors more than oversegmentation errors. Instead, we propose to count the number of oversegmentation and undersegmentation errors directly. Roughly speaking, an undersegmentation error occurs when an object is segmented together with a nearby object, and an oversegmentation error occurs when a single object is segmented into multiple pieces. More formally, we count the fraction of undersegmentation errors as

$$U = \frac{1}{N}\sum_{gt} \mathbb{1}\Big(\frac{|C_s \cap C_{gt}|}{|C_s|} < \tau_u\Big) \tag{2}$$

where $\mathbb{1}$ is an indicator function that is equal to 1 if the input is true and 0 otherwise and where τ_u is a constant threshold. We count the fraction of oversegmentation errors as

$$O = \frac{1}{N}\sum_{gt} \mathbb{1}\Big(\frac{|C_s \cap C_{gt}|}{|C_{gt}|} < \tau_o\Big), \tag{3}$$

where τ_o is a constant threshold. In our experiments, we choose $\tau_u = 0.5$ to allow for minor undersegmentation errors as well as errors in the ground-truth labeling. We use $\tau_o = 1$, since even a minor oversegmentation error causes a new (false) object to be created. We do not evaluate oversegmentations or undersegmentations when two ground-truth bounding boxes overlap. In such cases, it is difficult to tell whether the segmentation result is correct without more accurate ground-truth segmentation annotations (i.e. point-wise labeling instead of bounding boxes). Examples of undersegmentation and oversegmentation errors are shown in Figure 1.

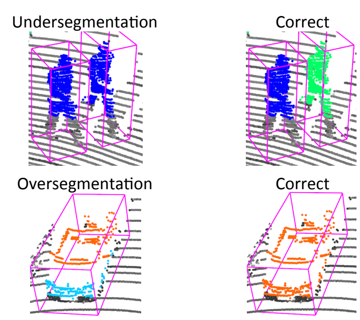

Fig. 1. Examples of an undersegmentation error (top) and an oversegmentation error (bottom). Each color denotes a single segment, and the ground-truth annotations are shown with a purple box, where each box represents a single object instance. (Best viewed in color)

We then compute an overall error rate based on the total number of undersegmentation and oversegmentation errors, as

$$E = U + \lambda_c O \tag{4}$$

where λ_c is a class-specific weighting parameter that penalizes oversegmentation errors relative to undersegmentation errors. For our experiments we simply choose $\lambda_c = 1$ for all classes, but λ_c can also be chosen for each application based on the effect of oversegmentation and undersegmetation errors for each class on the final performance.

Segmentation output: The output of our segmentation was saved in a set of 21 files, one for each sequence in the KITTI tracking dataset. Note that two of these sequences (0001 and 0013) were used for training and are not used as part of the evaluation. Each file has one line for each ground-truth segment. Each file also has a number of columns, as follows:

- frame: The KITTI frame number. Due to an error in our processing, our segmentation begins on frame 2.
- type: The type of object, based on the KITTI label: Car, Van, Truck, Pedestrian, Misc, Person sitting, Cyclist, or Tram
- seg_score: Defunct
- pos_points: $|C_s \cap C_{gt}|$
- blob_points: $|C_s|$
- gt_points: Number of points in the ground-truth bounding box, before ground-detection.
- other_pos_points: $|C_{gt}| - |C_s \cap C_{gt}|$
- label_id: Kitti label number
- track_id: Track ID assigned by our tracker
- distance: Euclidean distance between the center of the ground-truth bounding box and the Velodyne
- n_matched_tracks:$\sum_{s'} \mathbb{1}\left(|C_{s'} \cap C_{gt}| > 0\right)$
- under_segmentation: Defunct
- id_switch: An indicator of whether the track ID associated with this bounding box has changed to a different kitti label at this frame.
- attempted_correction: Defunct
- class_idx: The index of the class with the highest confidence (0: bicyclists, 1: cars, 2: pedestrians)
- class_confidence: The probability of the class with the highest confidence
- occluded: Whether the object is occluded in the image
- has_overlap: Whether this ground-truth bounding box overlaps with another ground-truth bounding box

where $\mathbb{1}$ is an indicator function that is equal to 1 if the input is true and 0 otherwise.

To process these files, run the MATLAB file segmentation_score.m with the directory of these files as input (see the default arguments in segmentation_score.m as an example). To compare multiple segmentations, run the MATLAB file compare_segmentations.m (see the default arguments in compare_segmentations.m as an example).

References

[1] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[4] David Held, Jesse Levinson, Sebastian Thrun, and Silvio Savarese. Combining 3d shape, color, and motion for robust anytime tracking. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[5] Yani Ioannou, Babak Taati, Robin Harrap, and Michael Greenspan. Difference of normals as a multi-scale operator in unorganized point clouds. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 501–508. IEEE, 2012.

[6] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

[7] Frank Moosmann and Christoph Stiller. Joint self-localization and tracking of generic objects in 3d range data. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1146–1152. IEEE, 2013.

[8] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[9] Alex Teichman and Sebastian Thrun. Group induction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2757–2763. IEEE, 2013.

[10] Alex Teichman, Jesse Levinson, and Sebastian Thrun. Towards 3d object recognition via classification of arbitrary object tracks. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4034–4041. IEEE, 2011.

[11] Dominic Zeng Wang, Ingmar Posner, and Paul Newman. What could move? finding cars, pedestrians and bicyclists in 3d laser data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4038–4044. IEEE, 2012.