Supplementary Material for
Scribble-Supervised LiDAR Semantic Segmentation

Ozan Unal[1] Dengxin Dai[1,2] Luc Van Gool[1,3]
[1]ETH Zurich, [2]MPI for Informatics, [3]KU Leuven
{ozan.unal, dai, vangool}@vision.ee.ethz.ch

1. The ScribbleKITTI Dataset

The goal of generating scribble-annotations is to to be fast and efficient while retaining as much information as possible to allow relatively high performance when compared to fully-supervised training. To this end, we formulate a set of guidelines for our annotators that also allows us to remain consistent across the dataset.

Process: We modify the point labeler [1] to include line annotations. An example of the labeler GUI can be seen in Fig. 1. As seen, the annotator draws lines on the LiDAR scene by determining its start and end points. The tool also allows multi-segment lines (when providing more than two points) to allow easier labeling of curved surfaces. As LiDAR point clouds are inherently sparse, we add a thickness to the drawn line. All points, who's projections fall onto the thickened line, are labeled. At $25m$ height we set the line thickness to 4 pixels. We adjust the thickness proportionally to the zoom settings to remain consistent throughout the labeling.

Guidelines: During labeling, each object in a scene (e.g. vehicle, person, sign, trunk) is marked with a single line. To ease the process and eliminate any spillage to the ground points, the annotators can use a threshold based filter for the z-axis (which was already implemented in the point labeler [1]) to hide ground points. An example can be see in Fig. 1 bottom-right. However, unlike the dense annotated case, annotators do not need to later remove the filter in order to determine difficult border points between objects and ground classes.

For classes that cover large distances, e.g ground classes (e.g. road, sidewalk, parking) and structure façades (e.g. building, fence), we try to annotate each segment using the least amount of scribbles. For example, given a north-south facing road segment that later turns right, the annotator draws two line-scribbles: 1) a north-south facing scribble that extends from the tile edge to junction, and 2) a west-east facing scribble that extends from the junction to the corresponding tile edge. If object interfere with the line-scribble (e.g. a car is in the middle of the road) the annotator can chose to scribble on either side of the object. For vegetation, each patch of greenery is annotated once. When periodically placed trees or bushes have similar heights, the threshold based filter can be used to isolate them, allowing a single annotation line to cover multiple individual trees. This also holds for sparse vegetation clusters in empty space (see main text Fig. 3 - bottom right). As 2D lines are projected onto the 3D surface to generate annotations, such scribbles may become indistinguishable once the viewing angle changes.

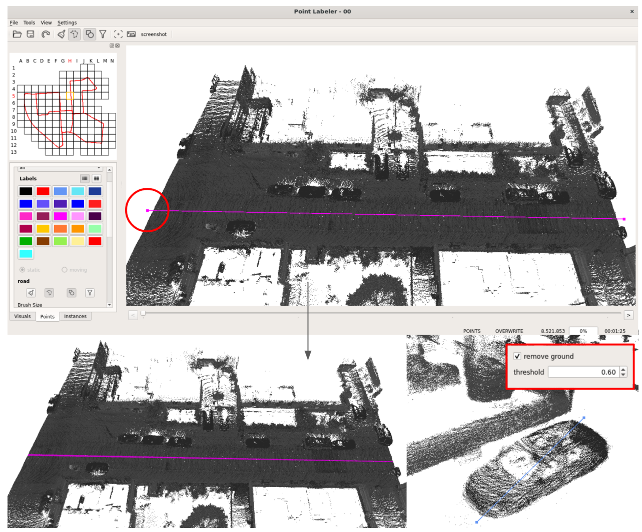

Figure 1. Screenshot of the labeling GUI and illustration of the process. As seen, the labeling tool [1] has been modified to be able to generate line annotations. The annotator needs to only select the starting and ending positions of the line.

2. Ablation Studies

Semi-supervised dataset: Our line-scribbles label roughly 8% of the total point count and take 10% of the time to acquire compared to their fully labeled counterpart (based on the reported times of SemanticKITTI [1]). Under a fixed labeling budget, we show that scribble-annotating all frames

β	mIoU	SS/FS
30%	60.9	94.7
50%	61.3	95.3
70%	60.8	94.6

Table 1. Investigating the effect of β for CRB-ST.

enables better representation capabilities compared to fully labeling partial frames (see main text Sec. 5.2). For these experiments, when simulating the semi-labeled setting, we follow the data generation process of Semi-sup [2] with 10% labeling.

Labeling Percentage for CRB-ST: We further investigate the effect of the labeling percentage β for CRB-ST. In Tab. 1 we compare results for three $beta$ values at 30%, 50%, 70%. As seen, the mIoU performance does depend on the percentage of predictions selected as pseudo-labels. $\beta = 50\%$ outperforms 30% and 70% by 0.4% and 0.5% respectively, achieving a better balance between the introduction of more supervision through pseudo-labeling, and the reduction of errors propagating from pseudo-labeling to distillation.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 1

[2] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6423–6432, October 2021. 2