Journal Title
XX(X):1–9

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Towards 3D LiDAR-based Semantic Scene Understanding of 3D Point Cloud Sequences – The SemanticKITTI Dataset

Jens Behley[1], Martin Garbade[2], Andres Milioto[1], Jan Quenzel[3], Sven Behnke[3], Jürgen Gall[2], and Cyrill Stachniss[1]

Abstract

A holistic semantic scene understanding exploiting all available sensor modalities is a core capability to master self-driving in complex everyday traffic. To this end, we present the *SemanticKITTI* dataset that provides point-wise semantic annotations of Velodyne HDL-64E point clouds of the KITTI Odometry Benchmark. Together with the data, we also published three benchmark tasks for semantic scene understanding covering different aspects of semantic scene understanding: (1) semantic segmentation for point-wise classification using single or multiple point clouds as input, (2) semantic scene completion for predictive reasoning on the semantics and occluded regions, and (3) panoptic segmentation combining point-wise classification and assigning individual instance identities to separate objects of the same class. In this article, we provide details on our dataset showing an unprecedented number of fully annotated point cloud sequences, more information on our labeling process to efficiently annotate such a vast amount of point clouds, and lessons learned in this process. The dataset and resources are available at http://www.semantic-kitti.org.

Keywords

Dataset, LiDAR, point clouds, semantic segmentation, panoptic segmentation, semantic scene completion

1 Introduction

Since autonomous vehicles successfully completed the driving tasks at the DARPA Urban Challenge (Urmson et al. 2008; Montemerlo et al. 2008), the prospect of fully autonomous cars led to founding of many startups pursuing the endeavor of creating always attentive robotic cars that prevent and avoid traffic fatalities occurring today due to human error. After the Urban Challenge, many expected and predicted that cars capable of driving fully autonomously in dense urban traffic would be already reality by today. However, solving the driving task in real-world environments with many non-compliant traffic participants that might violate rules and with the complexity of sometimes contradictory signage is a stupendous endeavor.

Perception is a centerpiece of every intelligent robotic system to handle the complexities of operating in semi-structured and natural environments. In particular, self-driving cars rely on robust and accurate perception systems that allow them to perform safe and efficient driving maneuvers. For a holistic semantic scene understanding, they need to perceive obstacles, identify other traffic participants, but also reason about functional street surface types, e.g., parking areas, lanes, and sidewalks.

Recent progress in perception using images, but also LiDAR sensors, is driven by advances in deep learning enabling end-to-end trainable perception systems without the need to hand-craft features (LeCun et al. 2015). Training deep neural networks with millions of parameters was mainly enabled by two essential developments: (1) the possibility to repurpose graphic processing units (GPUs) from producing pixels on a computer screen to compute matrix products in a highly parallel fashion, and (2) the availability of large-scale labeled datasets, such as ImageNet (Deng et al. 2009), that enabled training networks without over-fitting to the training data.

Real-world datasets play an important role in the aforementioned endeavor to attain fully autonomous cars, since they provide realistic data on the one hand, but also allow us to measure progress towards our goal on the other hand. Datasets, like Cityscapes (Cordts et al. 2016) and Mapillary Vistas (Neuhold et al. 2017), enable investigating fine-grained perception tasks, such as semantic segmentation (Everingham et al. 2010) providing classes for each pixel, but also panoptic segmentation (Kirillov et al. 2019), which additionally distinguishes between individual instances of the same class. More specifically, semantic segmentation distinguishes between classes but assigns different objects the same label, e.g., different cars cannot be distinguished. Panoptic segmentation differentiates additionally between objects leading to clear object boundaries or instances. However, panoptic segmentation requires instance ids only for so-called thing classes, which have clear boundaries, such as cars, pedestrians, and bicyclists. The remaining classes are called stuff classes and do not get an instance id assigned, such as vegetation, road, or sidewalks.

[1]Photogrammetry & Robotics Lab, University of Bonn, Germany.
[2]Computer Vision Group, University of Bonn, Germany.
[3]Autonomous Intelligent Systems, University of Bonn, Germany.

Corresponding author:
Jens Behley, Photogrammetry & Robotics Lab, University of Bonn, Nussallee 15, 53155 Bonn, Germany.
Email: jens.behley@igg.uni-bonn.de

Table 1. Overview of other point cloud datasets with bounding box (top) and semantic annotations (bottom).

Name	#Scans[1]	#Boxes	#Classes[2]	Data[3]	FoV[4]	Sequential	Reference
KITTI (Detection)	7k/7k	1k	3(3)	B	F	✗	Geiger et al. (2012)
Argoverse	22k	993k	17	B	C	✓	Chang et al. (2019)
Lyft	46k	1.3M	9	B	C	✓	Kesten et al. (2019)
CADC	7k	305k	10	B	C	✓	Pitropov et al. (2020)
nuScenes	44k	1.4M	10 (23)	B	C	✓	Caesar et al. (2020)
Waymo	200k	12M	4	B	C	✓	Sun et al. (2020)
A2D2	12k	12k	14	B	F	✗	Geyer et al. (2020)
H3D	27k	1.1M	8	B	F	✗	Patil et al. (2019)
PandaSet	16k	1.4M	12	B	C	✓	PandaSet (2020)
SemanticKITTI	23k/20k	682k	8	P	C	✓	-

Name	#Scans[1]	#Points	#Classes[2]	Data[3]	FoV[4]	Sequential	Reference
Oakland3d	17	1.6M	5 (44)	P	C	✗	Munoz et al. (2009)
Freiburg	77	1.1M	4 (11)	P	C	✗	Behley et al. (2012)
Wachtberg	5	400k	5 (5)	P	C	✗	Behley et al. (2012)
Semantic3d	15/15	4009M	8 (8)	P	C	✗	Hackel et al. (2017)
Paris-Lille-3D	3	143M	9 (50)	P	C	✗	Roynard et al. (2018)
Zhang et al.	140/112	32M	10 (10)	P	F	✗	Zhang et al. (2015)
SemanticPOSS	2k	216M	14	P	C	✗	Pan et al. (2020)
A2D2	31k	930k	38	P†	F	✗	Geyer et al. (2020)
PandaSet	16k	1388M	42	P	C	✓	PandaSet (2020)
nuScenes-lidarseg	34k/6k	1.4B	16 (32)	P	C	✓+	Caesar et al. (2020)
SemanticKITTI	23k/20k	4549M	25 (28)	P	C	✓	-

[1] Number of scans for train and test set, [2] Number of classes used for evaluation and number of classes annotated in brackets, [3] type of annotations, where B and P correspond to bounding boxes (B) and point-wise (P), [4] field-of-view (FoV) of LiDAR sensor with annotations, where F denotes frontal and C denotes complete 360°. † point-wise annotations via projection to annotated image and using corresponding image label. + frames labeled at 2 Hz

We present a dataset based on the KITTI Vision Benchmark (Geiger et al. 2012, 2013) that enables to investigate semantic segmentation and panoptic segmentation using point clouds from an automotive LiDAR sensor. To this end, we annotated all 22 sequences of odometry evaluation of the KITTI Vision Benchmark (Geiger et al. 2012, 2013) consisting of over 43,000 scans using 28 classes. We labeled each point of the point cloud such that corresponding instances of object classes get temporally consistent instance annotations. Additionally, we use the annotated sequential data and accurate poses to generate a real-world dataset for semantic scene completion, where an algorithm needs to provide class labels for voxels, but also predict the completed scene which is not visible in the given input voxelized scene. As commonly done with other datasets (Neuhold et al. 2017; Cordts et al. 2016; Lin et al. 2014), the test set labels are not published to ensure an unbiased and fair evaluation. We use CodaLab Competitions (see https://competitions.codalab.org/ for more information) that provides a platform to upload results for a hidden test set, which are then evaluated using a custom evaluation script on cloud-based evaluation servers without revealing the testset annotations.

This article complements our other papers (Behley et al. 2019, 2020), since it focuses on the dataset itself. We provide more details on the annotation process and statistics about the automatic instance extraction process. We, furthermore, discuss lessons learned in the process of annotating a large-scale dataset and maintaining an online evaluation with a hidden test set. Thus, the article provides additional information, compared to the more task-oriented papers, and we refer to these for more details on the tasks, baselines, and evaluation metrics for semantic segmentation and scene completion (Behley et al. 2019) and panoptic segmentation (Behley et al. 2020).

2 Related Work

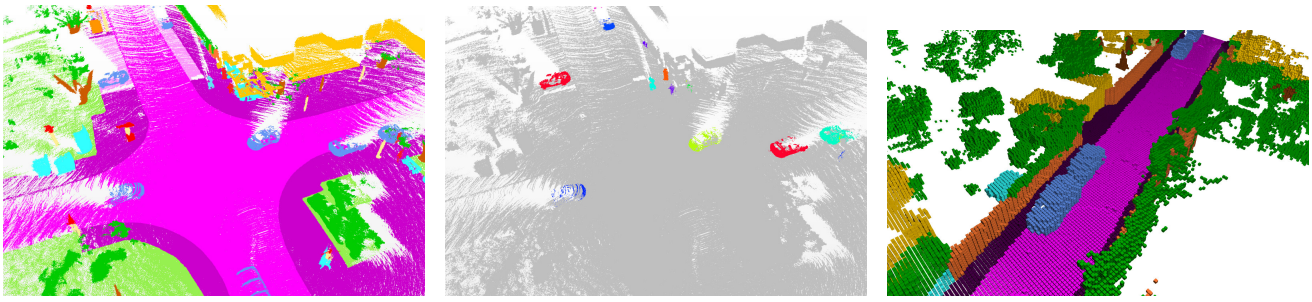
In the following, we focus on datasets providing LiDAR point clouds and in particular on datasets that provide annotations for perception tasks, such as object detection, semantic segmentation, or panoptic segmentation.

The seminal KITTI Vision Benchmark (Geiger et al. 2012) aimed at providing a collection of different benchmark tasks to evaluate perception algorithms for autonomous driving. It made a vast collection of data (Geiger et al. 2013) available that was recorded with a sensor suite commonly used by self-driving cars including a stereo camera, a Velodyne HDL-64E LiDAR, and an inertial navigation system to generate ground truth data for pose information. The provided benchmarks propelled research in the areas of motion estimation and traffic scene perception and enabled reproducible experiments with standardized metrics.

Since then, only a few LiDAR datasets were recently published that are recorded either with a terrestrial laser scanner (TLS), like the Semantic3d dataset (Hackel et al. 2017), or using automotive LiDARs, like the Paris-Lille-3D dataset (Roynard et al. 2018).

Recently, several major self-driving car ventures released datasets providing besides camera images also LiDAR point clouds, including Waymo (Sun et al. 2020), Lyft (Kesten et al. 2019), Audi (Geyer et al. 2020), Argo (Chang et al. 2019), Honda (Patil et al. 2019) and Motional (Caesar et al. 2020). While all these datasets provide instance annotations using bounding boxes, only a few datasets provide point-wise semantic annotation (Geyer et al. 2020; PandaSet 2020).

The A2D2 dataset (Geyer et al. 2020) provides annotations for semantic segmentation of images that can be used to obtain point-wise labels by projecting LiDAR points into the images and using the associated semantic class from



(a) (b) (c)

Figure 1. Qualitative examples of provided annotations from sequence 08. In (a), we show the semantic annotation and the middle (b) shows the corresponding instance annotation of 50 aggregated scans. We also provide input (not shown) and target voxel grids aggregated from multiple scans for the semantic scene completion task as shown in the right image (c).

the pixel-level annotation. However, this projection will never cover all LiDAR points due to the sensor placement and the resulting different view point. Very recently, the PandaSet (PandaSet 2020) provides point-wise annotations of LiDAR point clouds with 42 classes focusing on objects on the road, such as traffic participants, barriers, and cones, and more fine-grained distinction between different vehicle types compared to our annotation. SemanticPOSS (Pan et al. 2020) provides also semantic annotation of point clouds with focus on scenes with pedestrians captured in a campus environment. The classes are compatible with our classes and the authors ensured to provide labels in the same format as our annotation data. Pan et al. (2020) used our annotation tool presented in Sec. 3.1, but used tracking information to extract instances. NuScenes (Caesar et al. 2020) also added recently annotations for LiDAR point clouds with more diverse categories for different traffic participants. Together with the bounding box annotations, this dataset can also be used for panoptic segmentation. Due to the large number of different scenes, it provides a highly diverse set of situations.

Table 1 provides an overview of the aforementioned datasets and their characteristics. Other automotive datasets might provide more diversity in terms of cities or number of different scenes. However, our dataset is *the only dataset that combines point-wise semantic annotations directly made in sequences of three-dimensional point clouds with temporally consistent instance annotations for both non-moving and moving traffic participants*.

3 Dataset

Our dataset provides point-wise semantic annotations for the odometry sequences of the KITTI Vision Benchmark Suite (Geiger et al. 2013), which was the first large-scale dataset providing data recorded with a platform equipped with sensors commonly used on self-driving cars since the DARPA Urban Challenge (Montemerlo et al. 2008). The recording vehicle was equipped with a stereo camera covering the frontal field-of-view and a rotating 3D LiDAR sensor, the Velodyne HDL-64E S2, covering the full 360° field-of-view. Both modalities are synchronized such that the cameras are triggered when the spinning LiDAR sensor faces in forward direction (Geiger et al. 2013). The vehicle is additionally equipped with an inertial navigation system

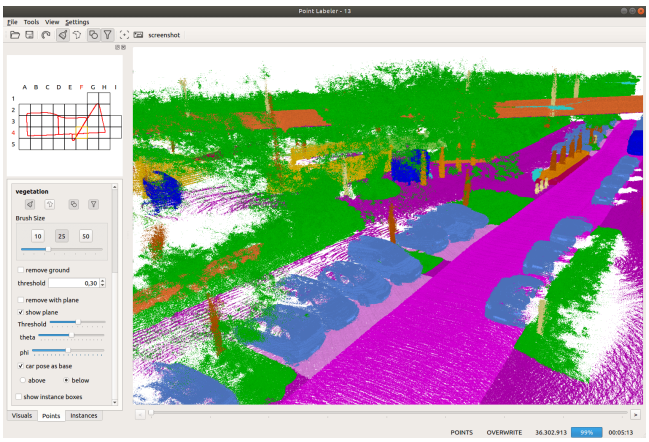


Figure 2. Our point cloud labeling application for sequential point clouds that we provide together with the dataset.

(INS) integrating an automotive-grade inertial measurement unit (IMU) with GPS providing position measurements.

In case of the odometry benchmark*, we use the data provided that uses point clouds of a single turn of the LiDAR sensor which are compensated for sensor motion, i.e., individual points in the point cloud are transformed to account for the movement of the sensor during a single turn of the rotating LiDAR sensor.

The odometry dataset comprises 22 sequences: 11 training sequences (sequences 00-10) with ground truth poses and 11 testing sequences (sequences 11-21) without pose information for which an odometry approach should estimate the poses. The pose error is only locally evaluated and therefore the provided poses are not loop closed or optimized to give globally consistent poses, such that already visited areas would be consistently mapped and old and new point cloud observations of the same place would be aligned properly.

3.1 Point Cloud Annotation

Our primary objective is to generate a consistent, accurate labeling of the sequential point clouds. It is essential to have consistent and loop-closed poses to facilitate the

*See http://www.cvlibs.net/datasets/kitti/eval_odometry.php for more information and download of the data.

Table 2. Relabeling and instance correction statistics for training (top) and test data (bottom)

		00	01	02	03	04	05	06	07	08	09	10
all	num. scans	4,541	1,101	4,661	801	271	2761	1,101	1,101	4,071	1,591	1,201
	relabeled [%]	0.2	13.1	7.0	15.1	4.5	4.2	8.8	2.5	5.4	0.5	2.6
non-moving	num. instances	670	0	296	41	15	210	144	205	523	178	86
	num. bboxes	83,138	0	21,056	3,344	450	21,270	20,012	23,071	55,079	11,933	5,409
	over-segmented [%]	20.3	0.0	37.2	46.3	46.7	27.6	25.7	31.2	24.5	25.8	34.9
	under-segmented [%]	16.3	0.0	31.1	7.3	6.7	9.5	17.4	13.7	14.3	13.5	10.5
moving	num. instances	32	317	70	3	23	24	14	20	114	28	15
	num. bboxes	2222	2509	1273	384	1075	1860	491	1986	6726	1570	796
	id switches [%]	50.0	19.2	11.4	66.7	60.9	50.0	50.0	60.0	31.6	42.9	20.0
		11	12	13	14	15	16	17	18	19	20	21
all	num. scans	921	1,061	3,281	631	1,901	1,731	491	1,801	4,981	831	2,721
	relabeled [%]	8.3	14.2	0.1	1.7	5.7	5.5	2.0	5.1	5.4	4.7	7.1
non-moving	num. instances	80	7	1589	0	189	276	5	371	919	6	3
	num. bboxes	4,636	540	175,147	0	17,911	24,810	130	35,697	86,101	441	96
	over-segmented [%]	31.2	42.9	15.9	0.0	23.8	22.5	0.0	10.8	28.0	0.0	0.0
	under-segmented [%]	17.5	14.3	37.5	0.0	10.6	12.7	40.0	32.6	15.9	0.0	0.0
moving	num. instances	7	84	102	0	13	31	26	174	142	456	1,853
	num. bboxes	197	1,909	5,208	0	571	416	537	8,133	5,630	15,945	32,332
	id switches [%]	14.3	45.2	41.2	0.0	30.8	6.5	46.2	23.6	22.5	30.5	32.9

consistent annotation by accumulating point clouds from multiple scans. Fig. 1 shows some qualitative examples of the provided annotations and the achieved fidelity of the point cloud annotations.

Pre-processing. To estimate globally consistent poses, we employed our surfel-based SLAM approach (Behley and Stachniss 2018). Our mapping approach finds heuristically loop closures using a map-based criterion and performs then pose graph optimization using loop closure constraints to obtain globally consistent poses. For sequences 02 and 07, where the automatic loop closure detection missed loop closures, we manually inserted loop closure constraints to ensure consistent mapping results.

Point cloud annotation. Using the estimated poses, we split the complete trajectory of a sequence into tiles of a given size (we used 100×100 m), where we always show an overlap of 15 m with neighboring tiles to ensure consistent labeling at tile boundaries. We collect all points overlapping with a tile and it's boundary, which allows us to consistently label points. The tiling is needed to enable consistent labeling of the overlapping point clouds and still allow to visualize all points inside a tile.

Our point cloud labeling graphical user interface provides different filtering methods to facilitate labeling, such as hiding points from a specific class or above/below an adjustable plane. The filtering allows us to accurately label points even in complex situations with overhanging vegetation that cover the view onto parts of the point cloud. Fig. 2 shows our labeling application with a challenging situation, where we have a bridge over a street and foliage from trees overhanging.

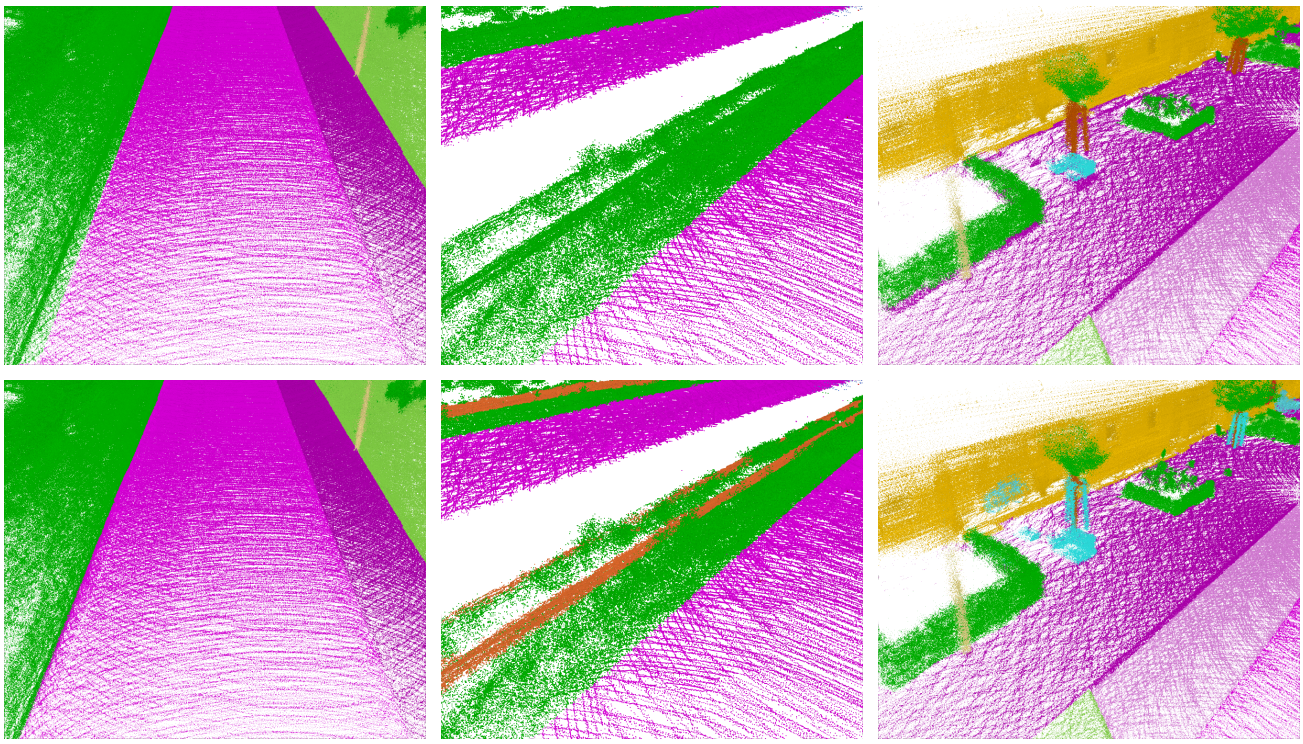
We provide two tools for annotating the point cloud: (1) a brush, which labels all points in a circular region around the current mouse position and (2) a polygon, which labels all points inside the polygon area specified by setting the polygon corners. Both operations happen on the projected points, where we check the projected three-dimensional point in the image plane for inclusion in the brushed or selected area. Thus, labeling of the point cloud is viewpoint dependent and requires to find a viewpoint that does not affect points that lie in the line of sight. Here, the aforementioned filtering tools allow to filter such points or simply all already annotated points.

Class annotation. Following best practices for dataset labeling, we compiled a labeling instruction based on Mapillary's instructions (Neuhold et al. 2017) and provided instruction videos on how to label certain objects such as cars and bicycles standing near a wall to our annotators.

Compared to image-based annotation, the annotation process with point clouds is more involved due to the aforementioned view dependency. The annotator often needs to change the viewpoint to find a perspective where an annotation is possible without annotating unwanted or already labeled points. An annotator needs on average 4.5 h per 100×100 m tile, when labeling residential areas—the most complex encountered scenes. Highway scenes are easier to label and an annotator needs on average 1.5 h to label a complete highway tile.

We provided feedback to the annotators to improve the quality and accuracy of labels. Furthermore, we also checked the labels in a second pass and inspected already labeled point clouds. During this check, inconsistencies were corrected and missing labels were added.

To ensure consistent annotations, a single annotator performed the verification of all sequences. Tab. 2 shows the fraction of points for each sequence, which we relabeled in the verification process. Thus, relabeling percentages are low for sequences mainly handled by this person, i.e., sequence 00, 09, 13. The number of relabeled points with the remaining sequences show the range of agreement between our annotators ranging from 2% up to 15%. The rather large spread compared to other image datasets (Gupta et al. 2019; Cordts et al. 2016; Lin et al. 2014) can be attributed to having annotators from different backgrounds, which never



(a) (b) (c)

Figure 3. Examples of correction during the verification process (top shows before validation and bottom after the validation). Here, we show (a) refined road boundaries, (b) added details (guard rail in the vegetation), (c) refined boundaries between classes.

annotated point clouds before. Thus, we and the annotators had to learn and refine the process while creating the annotations.

Fig. 3 shows examples of corrected labels during the verification process. Usually, we refined the boundaries of the road/sidewalk (e.g., curbs are always labeled as sidewalk), added details that where missed (e.g., tree trunks, guard rails, etc.), or refined the boundaries between different classes (e.g., building and sidewalk, car and road, etc.).

Instance annotation. For non-moving objects, we first cluster all points for each class using a fast grid-based segmentation approach (Behley et al. 2013). We use the aforementioned tiles to build a two-dimensional grid with cell size 0.1×0.1 m, where we insert all points using their x and y-coordinates into the corresponding grid cells. Finally, only grid cells with points exceeding a height threshold of 0.5 m are considered and grouped into segments using a simple flood fill algorithm.

For moving objects, we cluster each scan individually using a distance-based clustering as this provided more reliable results and it can be used to associate instances between consecutive scans using the same principle. First, we search for each point it's radius neighbors within 0.5 m and group points that share neighbors. To find associations with the previous four scans, we use a slightly larger radius of 1.0 m to find neighbors in the previous scans. If we find enough neighbors (i.e., we used at least 10 points) with the previous segments at different timestamps, we assign the same instance ID.

The described clustering leads inevitably to over- and under-segmentation, but also to wrong or missing associations between consecutive timestamps. We correct these issues manually using our point labeling tool, which allows us to create, join, and split instances.

Tab. 2 shows the fraction of over- and under-segmented non-moving objects that where manually corrected. For determining over- and under-segmented segments, we inspect all segments after the correction and compared the instance ids before and after the correction of the corresponding segments. If we find multiple instance ids in the segment before the correction, we record an over-segmentation. If we find that the segment before the correction is larger then the segment after the correction, we record this segment as an under-segmentation.

Note that we can exploit the pose information of the LiDAR sensor for non-moving objects, since we can cluster points based on their global coordinates. Only this allows us to inspect all 682k bounding boxes, since a large part of instances originate from non-moving objects.

However, Tab. 2 also reveals that 30%-50% of all segments were affected by over- and under-segmentation, which had to be manually corrected. Our axis-aligned grid often under-segmented nearby parked cars in non-axis aligned directions and over-segmented cars that were farther away due to the point distance.

For moving objects, we report in Tab. 2 the fraction of object trajectories that have at least once a different instance id in the automatically extracted instances. These data association errors were mainly caused by occlusions or by cars moving in the same direction with a small gap. Using here a tracking-based data association, which accounts for the object motion, would have resolved many of these issues.

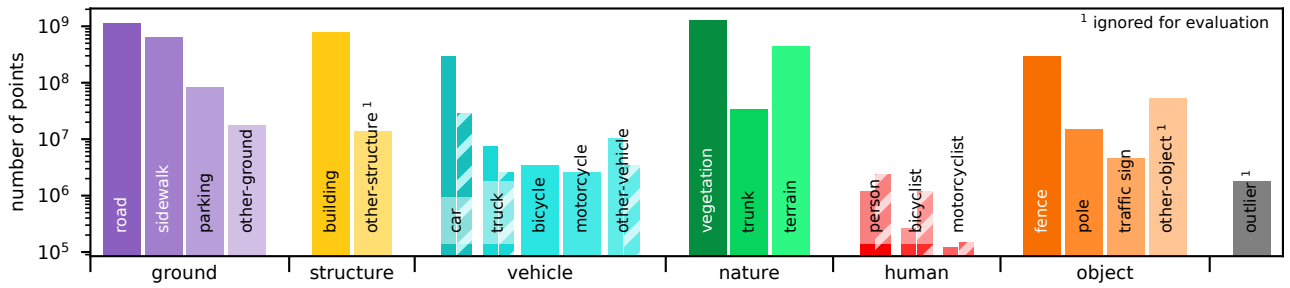


Figure 4. Class-wise distribution over the complete dataset. For the potentially moving classes car, truck, other-vehicle, person, bicyclist, motorcyclist, we furthermore distinguish between non-moving (solid) and moving instances (dashed bars).

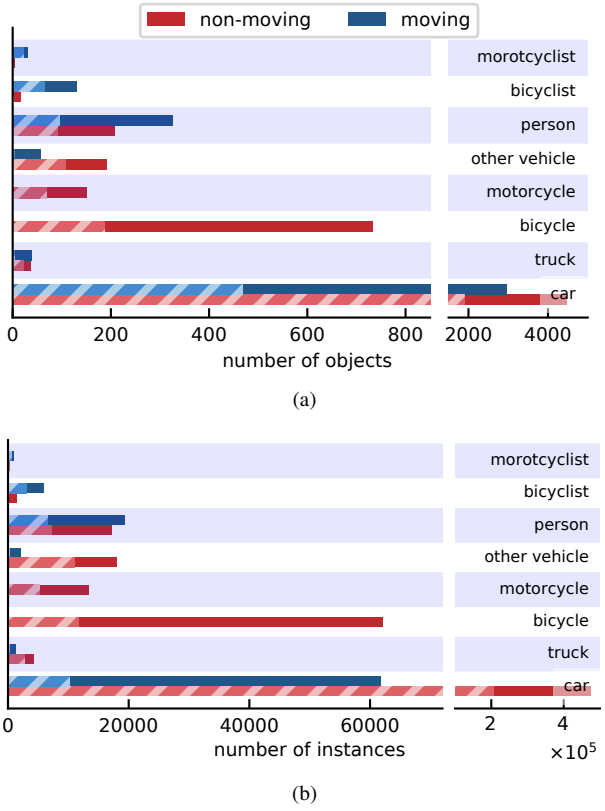


Figure 5. Instance-wise statistics over individual (a) object occurrences and (b) total count of bounding boxes or instances. Dashed bars show the proportion of objects in the training and validation data (sequence 00 – 10).

3.2 Class and Instance Distribution

Overall, we annotated over 500 tiles resulting in the class distribution shown in Fig. 4. We included besides traffic participants, such as car, truck, bicyclist, motorcyclist, also more functional ground classes relevant for autonomous driving, like sidewalk and parking areas. Note that we also differentiate for potentially moving classes between non-moving, i.e., the object did not move while observing it, and moving objects, i.e., the object moved at least for a single frame while the sensor observed it. In total, we distinguish between 28 classes including the moving and non-moving classes.

Note that we label points belonging to a bicycle or a motorcycle also as bicyclist or motorcyclist, when the vehicle has a rider. This simplification was needed, since persons on a bicycle or motorcycle cannot be separated reliably and accurately from the vehicle. Due to the sparser resolution of the point clouds compared to images, we preferred to label only the complete objects.

Naturally, road, vegetation and building points are by far the most often occurring classes. Motorcyclist is a class with only few examples present in the data.

The class car is by far the most often occurring traffic participant, which can be also seen in the object count depicted in Fig. 5. We show in the upper part of Fig. 5 the sequence-wise counts of instance annotations, i.e., we count each object only once even if it is seen multiple times by the sensor. The lower part of the figure shows the accumulated scan-wise counts of instances, where we count the instances without considering the temporally consistent instance ID.

3.3 Dataset Organization and Format

We tried to keep the data organization as close as possible to the original KITTI Vision Benchmark. Thus, we use the same format for our pose files and a similar binary format corresponding to the point cloud format of KITTI. Pose information is given in the coordinate frame of the camera, therefore the calibration data of each sequence provided by the KITTI odometry benchmark is needed to correctly transform the point clouds.

We added the folder `labels` containing for each point cloud a binary label file. Corresponding to each point, this file contains a 32-bit unsigned integer, where the upper 16 bit contain the instance id and the lower 16 bit contain the class id. Here, a value of zero corresponds to no instance assigned to the corresponding point and a zero label corresponds to an unlabeled point.

The folder `voxels` contains the voxelized point clouds, the voxel-wise labels, the invalid voxels, and the occluded voxels. Here, we opted for reducing the size of the data by using only bit flags to encode 8 voxels inside a single byte. Labels are represented by a 16 bit integer and we do not distinguish between different instances.

3.4 Benchmark Competitions and Development Kit

Together with the data, we also provide competitions on CodaLab Competitions that allow evaluating approaches on a hidden test set, i.e., part of the data for which we do not provide annotations. We use the original test sequences 11-21 of the odometry benchmark, since we do not want to

interfere with the original benchmark which could exploit the provided labels to get better results on the original odometry benchmark. The large test set might furthermore incentivize the development of fast approaches to process the large number of scans.

We currently provide competitions for semantic segmentation, semantic scene completion, and panoptic segmentation. More information regarding the baselines and metrics are provided the corresponding papers (Behley et al. 2019, 2020).

Together with the dataset, we also published a development kit[†] implemented in Python that provides methods to read the data and contains all evaluation scripts used in the aforementioned competitions. We furthermore provide tools to visualize the point clouds and the voxel grids.

4 Lessons Learned

In the process of labeling around 43,000 point clouds with a team of nine annotators, we learned quite a bit about managing the annotation process, but also encountered pitfalls. From these experiences, we want to share some insights that might help others to organize annotating data at such scale and avoid some failures.

We relied on university students from computing related, but also unrelated fields. Accurate and still efficient annotators are hard to find and we had a selection process or casting, where we let the prospective annotators use our point cloud annotation application and could directly see how they performed in the session. During this process, we recognized that point cloud annotation is quite challenging and that even annotators that are comfortable with labeling images had big problems to navigate and transfer this knowledge to the annotation of point clouds. Thus, annotator screening turned out to be a mandatory first step to achieve good results.

However, we learned most about the problems and challenges by labeling ourselves. This also helps to rectify the expectations and helps in the selection of the targeted classes. We found early on that labeling lane markings would have increased the time per tile to much, such that we decided not to include these in our labeling effort.

Next, we had to invest additional time in training and correcting the annotators continuously to ensure sufficient quality of the provided annotations. Only this ensured a high quality of the resulting annotations and a speed up in the process. Interestingly, all annotators improved during this process substantially and could adopt our suggestions to label more efficiently. Since we also used our own tools, we could develop and refine our annotation application over time and introduce new ways of filtering points, which also decreased the time needed to label a point cloud. Initially, we relied mostly on a height-based ground removal, which was not sufficient to consistently remove ground points. Adding a plane-based removal, where the annotator could set the angles and the threshold along the z-axis of the associated plane, proved to be much more robust.

Nevertheless, some of our early attempts to speed up the process by using pre-trained and fine-tuned image segmentation methods did not succeed. First, we aimed at having consistent labels for the aggregated point cloud and therefore the projection of single image labels to the point cloud needed to be aggregated using a point-wise majority vote. Second, the view point differences between the LiDAR sensor and the camera led to occlusions that could not always be resolved by aggregating multiple projected image labels, since one had only a frontal view available. Third, the misalignment between camera and LiDAR caused by pixel shifts or simply the rotation of the LiDAR resulted in "bleeding" of semantic labels into background regions or inconsistent labels on objects where parts of nearby classes got projected onto. Surprisingly, correction of these boundary issues took nearly as much time as labeling the whole tile from scratch. Additionally, our impression was also that the transferred labels biased the annotations in being "good enough", but being actually inferior to a labeling from scratch, since error or small problems with the projection of annotations are hardly visible.

5 Conclusion

SemanticKITTI provides point-wise semantic and instance annotations for all sequences of the KITTI odometry benchmark. At the time of writing, it is still the largest dataset that provides such annotations for point cloud sequences. Based on these annotation, we provide benchmarks for semantic segmentation and semantic scene completion (Behley et al. 2019). We added recently panoptic segmentation joining semantic and instance segmentation to the set of benchmarks (Behley et al. 2020). Furthermore, we plan to extend the set of benchmarks with additional tasks.

Besides these benchmarks, we also see already adoption of the data for other domains benefiting from semantics, like semantic SLAM (Chen et al. 2019; Gan et al. 2020), LiDAR-based localization (Yan et al. 2019), and loop closure detection (Chen et al. 2020). Moreover, domain adaptation (Jaritz et al. 2020; Langer et al. 2020) was also quite recently investigated to exploit our annotations for other sensors with different sensor geometries. We are curious in which ways the annotations and our tools will be used in future.

Acknowledgements

We thank all students that helped with annotating the data. The work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under FOR 1505 Mapping on Demand, BE 5996/1-1, GA 1927/5-2 (FOR 2535), and under Germanys Excellence Strategy, EXC-2070 – 390732324 (PhenoRob).

References

Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Stachniss C and Gall J (2019) SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*.

Behley J, Milioto A and Stachniss C (2020) A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI. *arXiv preprint* .

[†] See https://github.com/PRBonn/semantic-kitti-api

Behley J and Stachniss C (2018) Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In: *Proc. of Robotics: Science and Systems (RSS)*.

Behley J, Steinhage V and Cremers A (2013) Laser-based Segment Classification Using a Mixture of Bag-of-Words. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Behley J, Steinhage V and Cremers AB (2012) Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.

Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G and Beijbom O (2020) nuScenes: A Multimodal Dataset for Autonomous Driving. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Wang D, Carr P, Lucey S, Ramanan D and Hays J (2019) Argoverse: 3D Tracking and Forecasting with Rich Maps. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Chen X, Läbe T, Milioto A, Röhling T, Vysotska O, Haag A, Behley J and Stachniss C (2020) OverlapNet: Loop Closing for LiDAR-based SLAM. In: *Proc. of Robotics: Science and Systems (RSS)*.

Chen X, Milioto A, Palazzolo E, Gigure P, Behley J and Stachniss C (2019) SuMa++: Efficient LiDAR-based Semantic SLAM. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S and Schiele B (2016) The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Deng J, Dong W, Socher R, Li L, Li K and Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. DOI:10.1109/CVPR.2009.5206848.

Everingham M, Van Gool L, Williams C, Winn J and Zisserman A (2010) The Pascal Visual Object Classes (VOC) Challenge. *Intl. Journal of Computer Vision (IJCV)* 88(2): 303–338.

Gan L, Zhang R, Grizzle J, Eustice R and Ghaffri M (2020) Bayesian Spatial Kernel Smoothing for Scalable Dense Semantic Mapping. *IEEE Robotics and Automation Letters (RA-L)* 5(2).

Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets Robotics: The KITTI Dataset. *Intl. Journal of Robotics Research (IJRR)* .

Geiger A, Lenz P and Urtasun R (2012) Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. pp. 3354–3361.

Geyer J, Kassahun Y, Mahmudi M, Ricou X, Durgesh R, Chung AS, Hauswald L, Pham VH, Mühlegg M, Dorn S, Fernandez T, Jänicke M, Mirashi S, Savani C, Sturm M, Vorobiov O, Oelker M, Garreis S and Schuberth P (2020) A2D2: Audi Autonomous Driving Dataset. *arXiv preprint* :2004.06320.

Gupta A, Dollar P and Girshick R (2019) LVIS: A Dataset for Large Vocabulary Instance Segmentation. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K and Pollefeys M (2017) SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1. pp. 91–98.

Jaritz M, Vu TH, Charette R, Wirbel E and Perez P (2020) xMUDA: Cross-Modal Unsupervised Domain Adaptation for 3D Semantic Segmentation. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Kesten R, Usman M, Houston J, Pandya T, Nadhamuni K, Ferreira A, Yuan M, Low B, Jain A, Ondruska P, Omari S, Shah S, Kulkarni A, Kazakova A, Tao C, Platinsky L, Jiang W and Shet V (2019) Lyft Level 5 AV Dataset 2019. url1https://level5.lyft.com/dataset/.

Kirillov A, He K, Girshick R, Rother C and Dollár P (2019) Panoptic Segmentation. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Langer F, Milioto A, Haag A, Behley J and Stachniss C (2020) Domain Transfer for Semantic Segmentation of LiDAR Data using Deep Neural Networks. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

LeCun Y, Bengio Y and Hinton G (2015) Deep Learning. *Nature* 521: 436–444.

Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL and Dollsár P (2014) Microsoft COCO: Common Objects in Context. In: *Proc. of the Europ. Conf. on Computer Vision (ECCV)*.

Montemerlo M, Becker J, Bhat S, Dahlkamp H, Dolgov D, Ettinger S, Haehnel D, Hilden T, Hoffmann G, Huhnke B, Johnston D, Klumpp S, Langer D, Levandowski A, Levinson J, Marcil J, Orenstein D, Paefgen J, Penny I, Petrovskaya A, Pflueger M, Stanek G, Stavens D, Vogt A and Thrun S (2008) Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics* 25(9): 569–597. DOI:10.1002/rob.20258.

Munoz D, Bagnell JA, Vandapel N and Hebert M (2009) Contextual Classification with Functional Max-Margin Markov Networks. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Neuhold G, Ollmann T, Bulo SR and Kontschieder P (2017) The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In: *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*.

Pan Y, Gao B, Mei J, Geng S, Li C and Zhao H (2020) SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances. *arXiv preprint* (2002.09147).

PandaSet (2020) PandaSet Dataset (Available at https://scale.com/open-datasets/pandaset).

Patil A, Malla S, Gang H and Chen YT (2019) The H3D dataset forfull-surround 3D multi-object detection and tracking in crowded urban scenes. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.

Pitropov M, Danson G, Rebello J, Smart M, Wang C, Czarnecki K and Waslander S (2020) Canadian Adverse Driving Conditions Dataset. *arXiv preprint* 2001.10117.

Roynard X, Deschaud JE and Goulette F (2018) Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Intl. Journal of Robotics Research (IJRR)* 37(6): 545–557.

Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, Vasudevan V, Han W, Ngiam J,

Zhao H, Timofeev A, Ettinger S, Krivokon M, Gao A, Joshi A, Zhang Y, Shlens J, Chen Z and Anguelov D (2020) Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark MN, Dolan J, Duggins D, Galatali T, Geyer C, Gittleman M, Harbaugh S, Hebert M, Howard TM, Kolski S, Kelly A, Likhachev M, McNaughton M, Miller N, Peterson K, Pilnick B, Rajkumar R, Rybski P, Salesky B, Seo YW, Singh S, Snider J, Stentz A, Whittaker W, Wolkowicki Z, Ziglar J, Bae H, Brown T, Demitrish D, Litkouhi B, Nickolaou J, Sadekar V, Zhang W, Struble J, Taylor M, Darms M and Ferguson D (2008) Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics (JFR)* 25(8): 425–466. DOI:10.1002/rob.20255.

Yan F, Vysotska O and Stachniss C (2019) Global Localization on OpenStreetMap using 4-bit Semantic Descriptors. In: *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*.

Zhang R, Candra SA, Vetter K and Zakhor A (2015) Sensor Fusion for Semantic Segmentation of Urban Scenes. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.