International Conference on Computer Vision (ICCV), Seoul, Korea, October 2019.

SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences

Jens Behley* Martin Garbade* Andres Milioto Jan Quenzel
Sven Behnke Cyrill Stachniss Juergen Gall
University of Bonn, Germany

www.semantic-kitti.org

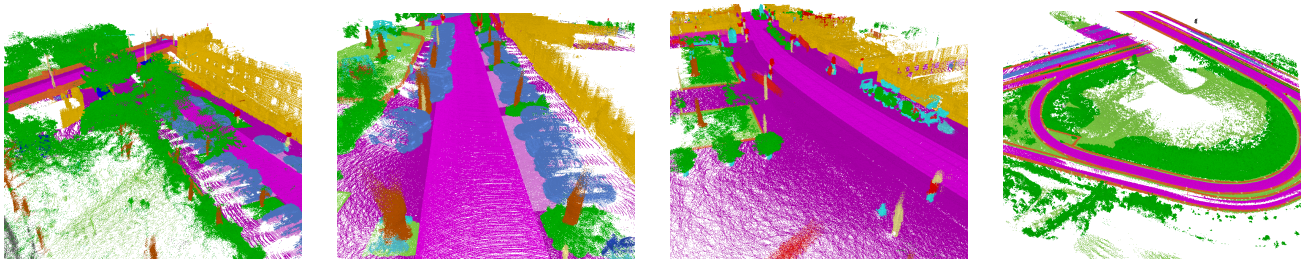

Figure 1: Our dataset provides dense annotations for each scan of all sequences from the KITTI Odometry Benchmark [19]. Here, we show multiple scans aggregated using pose information estimated by a SLAM approach.

Abstract

Semantic scene understanding is important for various applications. In particular, self-driving cars need a fine-grained understanding of the surfaces and objects in their vicinity. Light detection and ranging (LiDAR) provides precise geometric information about the environment and is thus a part of the sensor suites of almost all self-driving cars. Despite the relevance of semantic scene understanding for this application, there is a lack of a large dataset for this task which is based on an automotive LiDAR.

In this paper, we introduce a large dataset to propel research on laser-based semantic segmentation. We annotated all sequences of the KITTI Vision Odometry Benchmark and provide dense point-wise annotations for the complete 360^o field-of-view of the employed automotive LiDAR. We propose three benchmark tasks based on this dataset: (i) semantic segmentation of point clouds using a single scan, (ii) semantic segmentation using multiple past scans, and (iii) semantic scene completion, which requires to anticipate the semantic scene in the future. We provide baseline experiments and show that there is a need for more sophisticated models to efficiently tackle these tasks. Our dataset opens the door for the development of more advanced methods, but also provides plentiful data to investigate new research directions.

1. Introduction

Semantic scene understanding is essential for many applications and an integral part of self-driving cars. Particularly, fine-grained understanding provided by semantic segmentation is necessary to distinguish drivable and non-drivable surfaces and to reason about functional properties, like parking areas and sidewalks. Currently, such understanding, represented in so-called high definition maps, is mainly generated in advance using surveying vehicles. However, self-driving cars should also be able to drive in unmapped areas and adapt their behavior if there are changes in the environment.

Most self-driving cars currently use multiple different sensors to perceive the environment. Complementary sensor modalities enable to cope with deficits or failures of particular sensors. Besides cameras, light detection and ranging (LiDAR) sensors are often used as they provide precise distance measurements that are not affected by lighting.

Publicly available datasets and benchmarks are crucial for empirical evaluation of research. They mainly fulfill three purposes: (i) they provide a basis to measure progress, since they allow to provide results that are reproducible and comparable, (ii) they uncover shortcomings of the current state of the art and therefore pave the way for novel approaches and research directions, and (iii) they make it possible to develop approaches without the need to first painstakingly collect and label data. While multiple

* indicates equal contribution

	#scans[1]	#points[2]	#classes[3]	sensor	annotation	sequential
SemanticKITTI (Ours)	**23201/20351**	**4549**	**25** (28)	Velodyne HDL-64E	point-wise	✓
Oakland3d [36]	17	1.6	5 (44)	SICK LMS	point-wise	✗
Freiburg [50, 6]	77	1.1	4 (11)	SICK LMS	point-wise	✗
Wachtberg [6]	5	0.4	5 (5)	Velodyne HDL-64E	point-wise	✗
Semantic3d [23]	15/15	4009	8 (8)	Terrestrial Laser Scanner	point-wise	✗
Paris-Lille-3D [47]	3	143	9 (50)	Velodyne HDL-32E	point-wise	✗
Zhang et al. [65]	140/112	32	10 (10)	Velodyne HDL-64E	point-wise	✗
KITTI [19]	7481/7518	1799	3	Velodyne HDL-64E	bounding box	✗

Table 1: Overview of other point cloud datasets with semantic annotations. Ours is by far the largest dataset with sequential information. [1]Number of scans for train and test set, [2]Number of points is given in millions, [3]Number of classes used for evaluation and number of classes annotated in brackets.

large datasets for *image-based* semantic segmentation exist [10, 39], publicly available datasets with point-wise annotation of three-dimensional point clouds are still comparably small, as shown in Table 1.

To close this gap we propose *SemanticKITTI*, a large dataset showing unprecedented detail in point-wise annotation with 28 classes, which is suited for various tasks. In this paper, we mainly focus on *laser-based* semantic segmentation, but also semantic scene completion. The dataset is distinct from other laser datasets as we provide accurate scan-wise annotations of sequences. Overall, we annotated all 22 sequences of the odometry benchmark of the *KITTI Vision Benchmark* [19] consisting of over 43 000 scans. Moreover, we labeled the complete horizontal 360° field-of-view of the rotating laser sensor. Figure 1 shows example scenes from the provided dataset. In summary, our main contributions are:

- We present a point-wise annotated dataset of point cloud sequences with an unprecedented number of classes and unseen level-of-detail for each scan.
- We furthermore provide an evaluation of state-of-the-art methods for semantic segmentation of point clouds.
- We investigate the usage of sequence information for semantic segmentation using multiple scans.
- Based on the annotation of sequences of a moving car, we furthermore introduce a real-world dataset for semantic scene completion and provide baseline results.
- Together with a benchmark website, the point cloud labeling tool is also publicly available, enabling other researchers to generate other labeled datasets in future.

This large dataset will stimulate the development of novel algorithms, make it possible to investigate new research directions, and puts evaluation and comparison of these novel algorithms on a more solid ground.

2. Related Work

The progress of computer vision has always been driven by benchmarks and datasets [55], but the availability of especially large-scale datasets, such as *ImageNet* [13], was even a crucial prerequisite for the advent of deep learning.

More task-specific datasets geared towards self-driving cars were also proposed. Notable is here the *KITTI Vision Benchmark* [19] since it showed that off-the-shelf solutions are not always suitable for autonomous driving. The *Cityscapes* dataset [10] is the first dataset for self-driving car applications that provides a considerable amount of pixel-wise labeled images suitable for deep learning. The *Mapillary Vistas* dataset [39] surpasses the amount and diversity of labeled data compared to *Cityscapes*.

Also in point cloud-based interpretation, *e.g.*, semantic segmentation, RGB-D based datasets enabled tremendous progress. *ShapeNet* [8] is especially noteworthy for point clouds showing a single object, but such data is not directly transferable to other domains. Specifically, LiDAR sensors usually do not cover objects as densely as an RGB-D sensor due to their lower angular resolution, in particular in vertical direction.

For indoor environments, there are several datasets [48, 46, 24, 3, 11, 35, 32, 12] available, which are mainly recorded using RGB-D cameras or synthetically generated. However, such data shows very different characteristics compared to outdoor environments, which is also caused by the size of the environment, since point clouds captured indoors tend to be much denser due to the range at which objects are scanned. Furthermore, the sensors have different properties regarding sparsity and accuracy. While laser sensors are more precise than RGB-D sensors, they usually only capture a sparse point cloud compared to the latter.

For outdoor environments, datasets were recently proposed that are recorded with a terrestrial laser scanner (TLS), like the *Semantic3d* dataset [23], or using automotive LiDARs, like the *Paris-Lille-3D* dataset [47]. However, the *Paris-Lille-3D* provides only the aggregated scans with

point-wise annotations for 50 classes from which 9 are selected for evaluation. Another recently used large dataset for autonomous driving [57], but with fewer classes, is not publicly available.

The *Virtual KITTI* dataset [17] provides synthetically generated sequential images with depth information and dense pixel-wise annotation. The depth information can also be used to generate point clouds. However, these point clouds do not show the same characteristics as a real rotating LiDAR, including defects like reflections and outliers.

In contrast to these datasets, our dataset combines a large amount of labeled points, a large variety of classes, and sequential scans generated by a commonly employed sensor used in autonomous driving, which is distinct from all publicly available datasets, also shown in Table 1.

3. The SemanticKITTI Dataset

Our dataset is based on the odometry dataset of the KITTI Vision Benchmark [19] showing inner city traffic, residential areas, but also highway scenes and countryside roads around Karlsruhe, Germany. The original odometry dataset consists of 22 sequences, splitting sequences 00 to 10 as training set, and 11 to 21 as test set. For consistency with the original benchmark, we adopt the same division for our training and test set. Moreover, we do not interfere with the original odometry benchmark by providing labels only for the training data. Overall, we provide 23 201 full 3D scans for training and 20 351 for testing, which makes it by a wide margin the largest dataset publicly available.

We decided to use the KITTI dataset as a basis for our labeling effort, since it allowed us to exploit one of the largest available collections of raw point cloud data captured with a car. We furthermore expect that there are also potential synergies between our annotations and the existing benchmarks and this will enable the investigation and evaluation of additional research directions, such as the usage of semantics for laser-based odometry estimation.

Compared to other datasets (cf. Table 1), we provide labels for sequential point clouds generated with a commonly used automotive LiDAR, *i.e.*, the Velodyne HDL-64E. Other publicly available datasets, like *Paris-Lille-3D* [47] or *Wachtberg* [6], also use such sensors, but only provide the aggregated point cloud of the whole acquired sequence or some individual scans of the whole sequence, respectively. Since we provide the individual scans of the whole sequence, one can also investigate how aggregating multiple consecutive scans influences the performance of the semantic segmentation and use the information to recognize moving objects.

We annotated 28 classes, where we ensured a large overlap of classes with the *Mapillary Vistas* dataset [39] and *Cityscapes* dataset [10] and made modifications where necessary to account for the sparsity and vertical field-of-view. More specifically, we do not distinguish between persons riding a vehicle and the vehicle, but label the vehicle and the person as either *bicyclist* or *motorcyclist*.

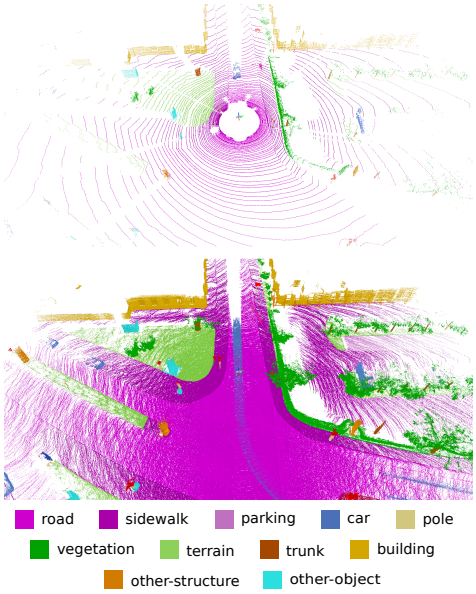

Figure 2: Single scan (top) and multiple superimposed scans with labels (bottom). Also shown is a moving car in the center of the image resulting in a trace of points.

We furthermore distinguished between moving and non-moving vehicles and humans, *i.e.*, vehicles or humans gets the corresponding moving class if they moved in some scan while observing them, as shown in the lower part of Figure 2. All annotated classes are listed in Figure 3 and a more detailed discussion and definition of the different classes can be found in the supplementary material. In summary, we have 28 classes, where 6 classes are assigned the attribute moving or non-moving, and one *outlier* class is included for erroneous laser measurements caused by reflections or other effects.

The dataset is publicly available through a benchmark website and we provide only the training set with ground truth labels and perform the test set evaluation online. We furthermore will also limit the number of possible test set evaluations to prevent overfitting to the test set [55].

3.1. Labeling Process

To make the labeling of point cloud sequences practical, we superimpose multiple scans above each other, which conversely allows us to label multiple scans consistently. To this end, we first register and loop close the sequences using an off-the-shelf laser-based SLAM system [5]. This step is needed as the provided information of the inertial navigation system (INS) often results in map inconsistencies, *i.e.*, streets that are revisited after some time have differ-

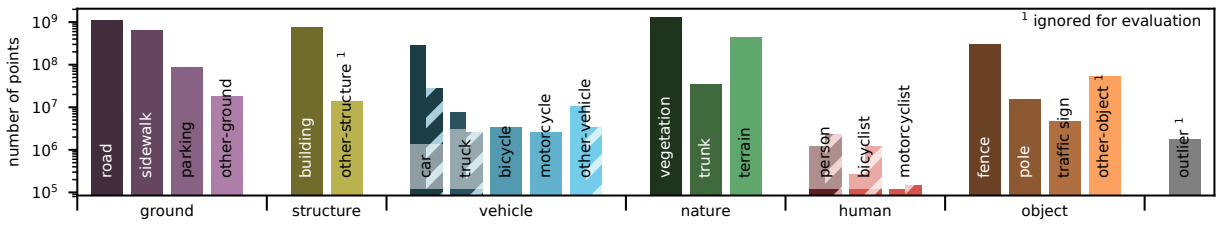

Figure 3: Label distribution. The number of labeled points per class and the root categories for the classes are shown. For movable classes, we also show the number of points on non-moving (solid bars) and moving objects (hatched bars).

ent height. For three sequences, we had to manually add loop closure constraints to get correctly loop closed trajectories, since this is essential to get consistent point clouds for annotation. The loop closed poses allow us to load all overlapping point clouds for specific locations and visualize them together, as depicted in Figure 2.

We subdivide the sequence of point clouds into tiles of 100 m by 100 m. For each tile, we only load scans overlapping with the tile. This enables us to label all scans consistently even when we encounter temporally distant loop closures. To ensure consistency for scans overlapping with more than one tile, we show all points inside each tile and a small boundary overlapping with neighboring tiles. Thus, it is possible to continue labels from a neighboring tile.

Following best practices, we compiled a labeling instruction and provided instructional videos on how to label certain objects, such as cars and bicycles standing near a wall. Compared to image-based annotation, the annotation process with point clouds is more complex, since the annotator often needs to change the viewpoint. An annotator needs on average 4.5 hours per tile, when labeling residential areas corresponding to the most complex encountered scenery, and needs on average 1.5 hours for labeling a highway tile.

We explicitly did not use bounding boxes or other available annotations for the KITTI dataset, since we want to ensure that the labeling is consistent and the point-wise labels should only contain the object itself.

We provided regular feedback to the annotators to improve the quality and accuracy of labels. Nevertheless, a single annotator also verified the labels in a second pass, *i.e.*, corrected inconsistencies and added missing labels. In summary, the whole dataset comprises 518 tiles and over 1 400 hours of labeling effort have been invested with additional $10 - 60$ minutes verification and correction per tile, resulting in a total of over 1 700 hours.

3.2. Dataset Statistics

Figure 3 shows the distribution of the different classes, where we also included the root categories as labels on the x-axis. The ground classes, *road*, *sidewalk*, *building*, *vegetation*, and *terrain* are the most frequent classes. The class *motorcyclist* only occurs rarely, but still more than 100 000 points are annotated.

The unbalanced count of classes is common for datasets captured in natural environments and some classes will be always under-represented, since they do not occur that often. Thus, an unbalanced class distribution is part of the problem that an approach has to master. Overall, the distribution and relative differences between the classes is quite similar in other datasets, *e.g.* *Cityscapes* [10].

4. Evaluation of Semantic Segmentation

In this section, we provide the evaluation of several state-of-the-art methods for semantic segmentation of a single scan. We also provide experiments exploiting information provided by sequences of multiple scans.

4.1. Single Scan Experiments

Task and Metrics. In semantic segmentation of point clouds, we want to infer the label of each three-dimensional point. Therefore, the input to all evaluated methods is a list of coordinates of the three-dimensional points along with their remission, *i.e.*, the strength of the reflected laser beam which depends on the properties of the surface that was hit. Each method should then output a label for each point of a scan, *i.e.*, one full turn of the rotating LiDAR sensor.

To assess the labeling performance, we rely on the commonly applied mean Jaccard Index or mean intersection-over-union (mIoU) metric [15] over all classes, given by

$$\frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c}, \tag{1}$$

where TP_c, FP_c, and FN_c correspond to the number of true positive, false positive, and false negative predictions for class c, and C is the number of classes.

As the classes *other-structure* and *other-object* have either only a few points and are otherwise too diverse with a high intra-class variation, we decided to not include these classes in the evaluation. Thus, we use 25 instead of 28 classes, ignoring *outlier*, *other-structure*, and *other-object* during training and inference.

Furthermore, we cannot expect to distinguish moving from non-moving objects with a single scan, since this Velodyne LiDAR cannot measure velocities like radars exploiting the Doppler effect. We therefore combine the moving classes with the corresponding non-moving class resulting in a total number of 19 classes for training and evaluation.

State of the Art. Semantic segmentation or point-wise classification of point clouds is a long-standing topic [2], which was traditionally solved using a feature extractor, such as Spin Images [29], in combination with a traditional classifier, like support vector machines [1] or even semantic hashing [4]. Many approaches used Conditional Random Fields (CRF) to enforce label consistency of neighboring points [56, 37, 36, 38, 62].

With the advent of deep learning approaches in image-based classification, the whole pipeline of feature extraction and classification has been replaced by end-to-end deep neural networks. Voxel-based methods transforming the point cloud into a voxel-grid and then applying convolutional neural networks (CNN) with 3D convolutions for object classification [34] and semantic segmentation [26] were among the first investigated models, since they allowed to exploit architectures and insights known for images.

To overcome the limitations of the voxel-based representation, such as the exploding memory consumption when the resolution of the voxel grid increases, more recent approaches either upsample voxel-predictions [53] using a CRF or use different representations, like more efficient spatial subdivisions [30, 44, 63, 59, 21], rendered 2D image views [7], graphs [31, 54], splats [51], or even directly the points [41, 40, 25, 22, 43, 28, 14].

Baseline approaches. We provide the results of six state-of-the-art architectures for the semantic segmentation of point clouds in our dataset: PointNet [40], PointNet++ [41], Tangent Convolutions [52], SPLATNet [51], Superpoint Graph [31], and SqueezeSeg (V1 and V2) [60, 61]. Furthermore, we investigate two extensions of SqueezeSeg: DarkNet21Seg and DarkNet53Seg.

PointNet [40] and PointNet++ [41] use the raw unordered point cloud data as input. Core of these approaches is max pooling to get an order-invariant operator that works surprisingly well for semantic segmentation of shapes and several other benchmarks. Due to this nature, however, PointNet fails to capture the spatial relationships between the features. To alleviate this, PointNet++ [41] applies individual PointNets to local neighborhoods and uses a hierarchical approach to combine their outputs. This enables it to build complex hierarchical features that capture both local fine-grained and global contextual information.

Tangent Convolutions [52] also handles unstructured point clouds by applying convolutional neural networks directly on surfaces. This is achieved by assuming that the data is sampled from smooth surfaces and defining a tangent convolution as a convolution applied to the projection of the local surface at each point into the tangent plane.

SPLATNet [51] takes an approach that is similar to the aforementioned voxelization methods and represents the point clouds in a high-dimensional sparse lattice. As with voxel-based methods, this scales poorly both in computation and in memory cost and therefore they exploit the sparsity of this representation by using bilateral convolutions [27], which only operates on occupied lattice parts.

Similarly to PointNet, Superpoint Graph [31], captures the local relationships by summarizing geometrically homogeneous groups of points into superpoints, which are later embedded by local PointNets. The result is a superpoint graph representation that is more compact and rich than the original point cloud exploiting contextual relationships between the superpoints.

SqueezeSeg [60, 61] also discretizes the point cloud in a way that makes it possible to apply 2D convolutions to the point cloud data exploiting the sensor geometry of a rotating LiDAR. In the case of a rotating LiDAR, all points of a single turn can be projected to an image by using a spherical projection. A fully convolutional neural network is applied and then finally filtered with a CRF to smooth the results. Due to the promising results of SqueezeSeg and the fast training, we investigated how the labeling performance is affected by the number of model parameters. To this end, we used a different backbone based on the Darknet architecture [42] with 21 and 53 layers, and 25 and 50 million parameters respectively. We furthermore eliminated the vertical downsampling used in the architecture.

We modified the available implementations such that the methods could be trained and evaluated on our large-scale dataset. Note that most of these approaches have so far only been evaluated on shape [8] or RGB-D indoor datasets [48]. However, some of the approaches [40, 41] were only possible to run with considerable downsampling to 50 000 points due to memory limitations.

Results and Discussion. Table 2 shows the results of our baseline experiments for various approaches using either directly the point cloud information [40, 41, 51, 52, 31] or a projection of the point cloud [60]. The results show that the current state of the art for point cloud semantic segmentation falls short for the size and complexity of our dataset.

We believe that this is mainly caused by the limited capacity of the used architectures (see Table 3), because the number of parameters of these approaches is much lower than the number of parameters used in leading image-based semantic segmentation networks. As mentioned above, we added DarkNet21Seg and DarkNet53Seg to test this hypothesis and the results show that this simple modification improves the accuracy from 29.5 % for SqueezeSeg to 47.4 % for DarkNet21Seg and to 49.9 % for DarkNet53Seg.

Approach	mIoU	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign
PointNet [40]	14.6	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPGraph [31]	17.4	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [51]	18.4	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [41]	20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
SqueezeSeg [60]	29.5	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5
SqueezeSegV2 [61]	39.7	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
TangentConv [52]	40.9	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5
DarkNet21Seg	47.4	91.4	74.0	57.0	26.4	81.9	85.4	18.6	26.2	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0
DarkNet53Seg	49.9	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2

Table 2: Single scan results (19 classes) for all baselines on sequences 11 to 21 (test set). All methods were trained on sequences 00 to 10, except for sequence 08 which is used as validation set.

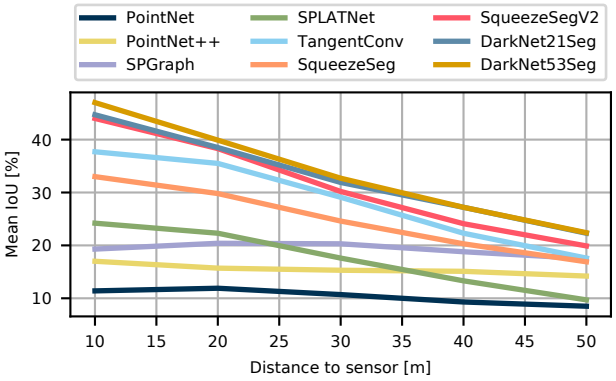

Figure 4: IoU vs. distance to the sensor.

Approach	num. parameters (million)	train time $\left(\frac{\text{GPU hours}}{\text{epoch}}\right)$	inference time $\left(\frac{\text{seconds}}{\text{point cloud}}\right)$
PointNet	3	4	0.5
PointNet++	6	16	5.9
SPGraph	0.25	6	5.2
TangentConv	0.4	6	3.0
SPLATNet	0.8	8	1.0
SqueezeSeg	1	0.5	0.015
SqueezeSegV2	1	0.6	0.02
DarkNet21Seg	25	2	0.055
DarkNet53Seg	50	3	0.1

Table 3: Approach statistics.

Another reason is that the point clouds generated by LiDAR are relatively sparse, especially as the distance to the sensor increases. This is partially solved in SqueezeSeg, which exploits the way the rotating scanner captures the data to generate a dense range image, where each pixel corresponds roughly to a point in the scan.

These effects are further analyzed in Figure 4, where the mIoU is plotted w.r.t. the distance to the sensor. It shows that results of all approaches get worse with increasing distance. This further confirms our hypothesis that the sparsity is the main reason for worse results at large distances. However, the results also show that some methods, like SPGraph, are less affected by the distance-dependent sparsity and this might be a promising direction for future research to combine the strength of both paradigms.

Especially classes with few examples, like motorcyclists and trucks, seem to be more difficult for all approaches. But also classes with only a small number of points in a single point cloud, like bicycles and poles, are hard classes.

Finally, the best performing approach (DarkNet53Seg) with 49.9% mIoU is still far from achieving results that are on par with image-based approaches, *e.g.*, 80% on the *Cityscapes* benchmark [10].

4.2. Multiple Scan Experiments

Task and Metrics. In this task, we allow methods to exploit information from a sequence of multiple past scans to improve the segmentation of the current scan. We furthermore want the methods to distinguish moving and non-moving classes, *i.e.*, all 25 classes must be predicted, since this information should be visible in the temporal information of multiple past scans. The evaluation metric for this task is still the same as in the single scan case, *i.e.*, we evaluate the mean IoU of the current scan no matter how many past scans were used to compute the results.

Baselines. We exploit the sequential information by combining 5 scans into a single, large point cloud, *i.e.*, the current scan at timestamp t and the 4 scans before at timestamps $t-1, \ldots, t-4$. We evaluate DarkNet53Seg and TangentConv, since these approaches can deal with a larger number of points without downsampling of the point clouds and could still be trained in a reasonable amount of time.

Results and Discussion. Table 4 shows the per-class results for the movable classes and the mean IoU (mIoU) over all classes. For each method, we show in the upper part of the row the IoU for non-moving (unshaded) and in the lower part of the row the IoU for moving objects (shaded). The

Approach	car	truck	other-vehicle	person	bicyclist	motorcyclist	mIoU
TangentConv [52]	84.9	21.1	18.5	1.6	0.0	0.0	34.1
	40.3	42.2	30.1	6.4	1.1	1.9	
DarkNet53Seg	84.1	20.0	20.7	7.5	0.0	0.0	41.6
	61.5	37.8	28.9	15.2	14.1	0.2	

Table 4: IoU results using a sequence of multiple past scans (in %). Shaded cells correspond to the IoU of the moving classes, while unshaded entries are the non-moving classes.

performance of the remaining static classes is similar to the single scan results and we refer to the supplement for a table containing all classes.

The general trend that the projective methods perform better than the point-based methods is still apparent, which can be also attributed to the larger amount of parameters as in the single scan case. Both approaches show difficulties in separating moving and non-moving objects, which might be caused by our design decision to aggregate multiple scans into a single large point cloud. The results show that especially bicyclist and motorcyclist never get correctly assigned the non-moving class, which is most likely a consequence from the generally sparser object point clouds.

We expect that new approaches could explicitly exploit the sequential information by using multiple input streams to the architecture or even recurrent neural networks to account for the temporal information, which again might open a new line of research.

5. Evaluation of Semantic Scene Completion

After leveraging a sequence of past scans for semantic point cloud segmentation, we now show a scenario that makes use of future scans. Due to its sequential nature, our dataset provides the unique opportunity to be extended for the task of 3D semantic scene completion. Note that this is the first real world outdoor benchmark for this task. Existing point cloud datasets cannot be used to address this task, as they do not allow for aggregating labeled point clouds that are sufficiently dense in both space and time.

In semantic scene completion, one fundamental problem is to obtain ground truth labels for real world datasets. In case of NYUv2 [48], CAD models were fit into the scene [45] using an RGB-D image captured by a Kinect sensor. New approaches often resort to prove their effectiveness on the larger, but synthetic SUNCG dataset [49]. However, a dataset combining the scale of a synthetic dataset and usage of real-world data is still missing.

In the case of our proposed dataset, the car carrying the LiDAR moves past 3D objects in the scene and thereby records their backsides, which are hidden in the initial scan due to self-occlusion. This is exactly the information needed for semantic scene completion as it contains the full 3D geometry of all objects while their semantics are provided by our dense annotations.

Dataset Generation. By superimposing an exhaustive number of future laser scans in a predefined region in front of the car, we can generate pairs of inputs and targets that correspond to the task of semantic scene completion. As proposed by Song *et al*. [49], our dataset for the scene completion task is a voxelized representation of the 3D scene.

We select a volume of $51.2\,\text{m}$ ahead of the car, $25.6\,\text{m}$ to every side and $6.4\,\text{m}$ in height with a voxel resolution of $0.2\,\text{m}$, which results in a volume of $256 \times 256 \times 32$ voxels to predict. We assign a single label to every voxel based on the majority vote over all labeled points inside a voxel. Voxels that do not contain any points are labeled as *empty*.

To compute which voxels belong to the occluded space, we check for every pose of the car which voxels are visible to the sensor by tracing a ray. Some of the voxels, *e.g*. those inside objects or behind walls are never visible, so we ignore them during training and evaluation.

Overall, we extracted $19\,130$ pairs of input and target voxel grids for training, 815 for validation and $3\,992$ for testing. For the test set, we only provide the unlabeled input voxel grid and withhold the target voxel grids. Figure 5 shows an example of an input and target pair.

Task and Metrics. In semantic scene completion, we are interested in predicting the complete scene inside a certain volume from a single initial scan. More specifically, we use as input a voxel grid, where each voxel is marked as empty or occupied, depending on whether or not it contains a laser measurement. For semantic scene completion, one needs to predict whether a voxel is occupied and its semantic label in the completed scene.

For evaluation, we follow the evaluation protocol of Song *et al*. [49] and compute the IoU for the task of scene completion, which only classifies a voxel as being occupied or empty, *i.e*., ignoring the semantic label, as well as mIoU (1) for the task of semantic scene completion over the same 19 classes that were used for the single scan semantic segmentation task (see Section 4).

State of the Art. Early approaches addressed the task of scene completion either without predicting semantics [16], thereby not providing a holistic understanding of the scene, or by trying to fit a fixed number of mesh models to the scene geometry [20], which limits the expressiveness of the approach.

Song *et al*. [49] were the first to address the task of semantic scene completion in an end-to-end fashion. Their work spawned a lot of interest in the field yielding models that combine the usage of color and depth informa-

Figure 5: Left: Visualization of the incomplete input for the semantic scene completion benchmark. Note that we show the labels only for better visualization, but the real input is a single raw voxel grid without any labels. Right: Corresponding target output representing the completed and fully labeled 3D scene.

tion [33, 18] or address the problem of sparse 3D feature maps by introducing submanifold convolutions [64] or increase the output resolution by deploying a multi-stage coarse to fine training scheme [12]. Other works experimented with new encoder-decoder CNN architectures as well as improving the loss term by adding adversarial loss components [58].

Baseline Approaches. We report the results of four semantic scene completion approaches. In the first approach, we apply SSCNet [49] without the flipped TSDF as input feature. This has minimal impact on the performance, but significantly speeds up the training time due to faster preprocessing [18]. Then we use the Two Stream (TS3D) approach [18], which makes use of the additional information from the RGB image corresponding to the input laser scan. Therefore the RGB image is first processed by a 2D semantic segmentation network, using the approach DeepLab v2 (ResNet-101) [9] trained on Cityscapes to generate a semantic segmentation. The depth information from the single laser scan and the labels inferred from the RGB image are combined in an early fusion. Furthermore, we modify the TS3D approach in two steps: First, by directly using labels from the best LiDAR-based semantic segmentation approach (DarkNet53Seg) and secondly, by exchanging the 3D-CNN backbone by SATNet [33].

Results and Discussion. Table 5 shows the results of each of the baselines, whereas results for individual classes are reported in the supplement. The TS3D network, incorporating 2D semantic segmentation of the RGB image, performs similar to SSCNet which only uses depth information. However, the usage of the best semantic segmentation directly working on the point cloud slightly outperforms SSCNet on semantic scene completion (TS3D + DarkNet53Seg). Note that the first three approaches are based on SSCNet's 3D-CNN architecture, which performs a 4 fold downsampling in a forward pass and thus renders them incapable of dealing with details of the scene. In our final approach, we exchange the SSCNet-backbone of TS3D + DarkNet53Seg with SATNet [33], which is capable of dealing with the desired output resolution. Due to memory limitations, we use random cropping during training. During inference, we divide each volume into six equal parts, perform scene completion on them individually and subsequently fuse them. This approach performs much better than the SSCNet based approaches.

	Completion (IoU)	Semantic Scene Completion (mIoU)
SSCNet [49]	29.83	9.53
TS3D [18]	29.81	9.54
TS3D [18] + DarkNet53Seg	24.99	10.19
TS3D [18] + DarkNet53Seg + SATNet	50.60	17.70

Table 5: Semantic scene completion baselines.

Apart from dealing with the target resolution, a challenge for current models is the sparsity of the laser input signal in the far field as can be seen from Figure 5. To obtain a higher resolution input signal in the far field, approaches would have to exploit more efficiently information from high resolution RGB images provided along with each laser scan.

6. Conclusion and Outlook

In this work, we have presented a large-scale dataset showing unprecedented scale in point-wise annotation of point cloud sequences. We provide a range of different baseline experiments for three tasks: (i) semantic segmentation using a single scan, (ii) semantic segmentation using multiple scans, and (iii) semantic scene completion.

In future work, we plan to provide also instance-level annotation over the whole sequence, *i.e.*, we want to distinguish different objects in a scan, but also identify the same object over time. This will enable to investigate temporal instance segmentation over sequences. However, we also see potential for other new tasks based on our labeling effort, such as the evaluation of semantic SLAM.

Acknowledgments 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 1927/2-2, and under Germanys Excellence Strategy, EXC-2070 – 390732324 (PhenoRob).

References

[1] Anuraag Agrawal, Atsushi Nakazawa, and Haruo Takemura. MMM-classification of 3D Range Data. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009. 5

[2] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 169–176, 2005. 5

[3] Iro Armeni, Alexander Sax, Amir R. Zamir, and Silvio Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv preprint*, 2017. 2

[4] Jens Behley, Kristian Kersting, Dirk Schulz, Volker Steinhage, and Armin B. Cremers. Learning to Hash Logistic Regression for Fast 3D Scan Point Classification. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5960–5965, 2010. 5

[5] Jens Behley and Cyrill Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018. 3

[6] Jens Behley, Volker Steinhage, and Armin B. Cremers. 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)*, 2012. 2, 3

[7] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 2017. 5

[8] Angel X. Chang, Thomas Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University and Princeton University and Toyota Technological Institute at Chicago, 2015. 2, 5

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation withDeep Convolutional Nets, Atrous Convolution,and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):834–848, 2018. 8

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 4, 6

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[12] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[14] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. *arXiv preprint*, 2018. 5

[15] Mark Everingham, S.M. Ali Eslami, Luc van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge a Retrospective. *International Journal on Computer Vision (IJCV)*, 111(1):98–136, 2015. 4

[16] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow. Structured Prediction of Unobserved Voxels From a Single Depth Image. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5431–5440, 2016. 7

[17] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[18] Martin Garbade, Yueh-Tung Chen, J. Sawatzky, and Juergen Gall. Two Stream 3D Semantic Scene Completion. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 8

[19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 1, 2, 3

[20] Andres Geiger and Chaohui Wang. Joint 3d Object and Layout Inference from a single RGB-D Image. In *Proc. of the German Conf. on Pattern Recognition (GCPR)*, pages 183–195, 2015. 7

[21] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[22] Fabian Groh, Patrick Wieschollek, and Hendrik Lensch. Flex-Convolution (Million-Scale Pointcloud Learning Beyond Grid-Worlds). In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, Dezember 2018. 5

[23] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys. 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, pages 91–98, 2017. 2

[24] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A Scene Meshes Dataset with aNNotations. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2016. 2

[25] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise Convolutional Neural Networks. In *Proc. of the IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR), 2018. 5

[26] Jing Huang and Suya You. Point Cloud Labeling using 3D Convolutional Neural Network. In *Proc. of the Intl. Conf. on Pattern Recognition (ICPR)*, 2016. 5

[27] Varun Jampani, Martin Kiefel, and Peter V. Gehler. Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[28] Mingyang Jiang, Yiran Wu, and Cewu Lu. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv preprint*, 2018. 5

[29] Andrew E. Johnson and Martial Hebert. Using spin images for effcient object recognition in cluttered 3D scenes. *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(5):433–449, 1999. 5

[30] Roman Klukov and Victor Lempitsky. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017. 5

[31] Loic Landrieu and Martin Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 6

[32] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. InteriorNet: Mega-scale Multi-sensor Photo-realistic Indoor Scenes Dataset. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018. 2

[33] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yainhe Han, and Xiaowei Li. See and Think: Disentangling Semantic Scene Completion. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, pages 261–272, 2018. 8

[34] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015. 5

[35] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017. 2

[36] Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 5

[37] Daniel Munoz, Nicholas Vandapel, and Marial Hebert. Directional Associative Markov Network for 3-D Point Cloud Classification. In *Proc. of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 63–70, 2008. 5

[38] Daniel Munoz, Nicholas Vandapel, and Martial Hebert. Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009. 5

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017. 2, 3

[40] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6

[41] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2017. 5, 6

[42] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint*, 2018. 5

[43] Dario Rethage, Johanna Wald, Jürgen Sturm, Nassir Navab, and Frederico Tombari. Fully-Convolutional Point Networks for Large-Scale Point Clouds. *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 5

[44] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning Deep 3D Representations at High Resolutions. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[45] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3D Object Shape from One Depth Image. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[46] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[47] Xavier Roynard, Jean-Emmanuel Deschaud, and Francois Goulette. 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, 2018. 2, 3

[48] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012. 2, 5, 7

[49] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7, 8

[50] Bastian Steder, Giorgio Grisetti, and Wolfram Burgard. Robust Place Recognition for 3D Range Data based on Point Features. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2010. 2

[51] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 6

[52] Maxim Tatarchenko, Jaesik Park, Vladen Koltun, and Qian-Yi Zhou. Tangent Convolutions for Dense Prediction in 3D.

In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 6, 7

[53] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, Jun Young Gwak, and Silvio Savarese. SEGCloud: Semantic Segmentation of 3D Point Clouds. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2017. 5

[54] Gusi Te, Wei Hu, Zongming Guo, and Amin Zheng. RGCNN: Regularized Graph CNN for Point Cloud Segmentation. *arXiv preprint*, 2018. 5

[55] A. Torralba and A. Efros. Unbiased Look at Dataset Bias. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2, 3

[56] Rudolph Triebel, Krisitian Kersting, and Wolfram Burgard. Robust 3D Scan Point Classification using Associative Markov Networks. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2603–2608, 2006. 5

[57] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep Parametric Continuous Convolutional Neural Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[58] Yida Wang, Davod Tan Joseph, Nassir Navab, and Frederico Tombari. Adversarial Semantic Scene Completion from a Single Depth Image. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, pages 426–434, 2018. 8

[59] Zongji Wang and Feng Lu. VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes. *arXiv preprint*, 2018. 5

[60] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 5, 6

[61] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019. 5, 6

[62] Xuehan Xiong, Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. 3-D Scene Analysis via Sequenced Predictions over Points and Regions. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2609–2616, 2011. 5

[63] Wei Zeng and Theo Gevers. 3DContextNet: K-d Tree Guided Hierarchical Learning of Point Clouds Using Local and Global Contextual Cues. *arXiv preprint*, 2017. 5

[64] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient Semantic Scene Completion Network with Spatial Group Convolution. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 733–749, 2018. 8

[65] Richard Zhang, Stefan A. Candra, Kai Vetter, and Avideh Zakhor. Sensor Fusion for Semantic Segmentation of Urban Scenes. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015. 2