KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D

Yiyi Liao, Jun Xie, and Andreas Geiger

Abstract—For the last few decades, several major subfields of artificial intelligence including computer vision, graphics, and robotics have progressed largely independently from each other. Recently, however, the community has realized that progress towards robust intelligent systems such as self-driving cars requires a concerted effort across the different fields. This motivated us to develop KITTI-360, successor of the popular KITTI dataset. KITTI-360 is a suburban driving dataset which comprises richer input modalities, comprehensive semantic instance annotations and accurate localization to facilitate research at the intersection of vision, graphics and robotics. For efficient annotation, we created a tool to label 3D scenes with bounding primitives and developed a model that transfers this information into the 2D image domain, resulting in over 150k images and 1B 3D points with coherent semantic instance annotations across 2D and 3D. Moreover, we established benchmarks and baselines for several tasks relevant to mobile perception, encompassing problems from computer vision, graphics, and robotics on the same dataset, e.g., semantic scene understanding, novel view synthesis and semantic SLAM. KITTI-360 will enable progress at the intersection of these research areas and thus contribute towards solving one of today's grand challenges: the development of fully autonomous self-driving systems.

Index Terms—Point cloud labeling, semantic label transfer, scene understanding, self-driving, datasets, performance evaluation

1 Introduction

One of the pioneering works in *computer vision* can be traced back to Larry Roberts' "Blocks World" in the 1960s [85], which aimed at identifying individual objects and inferring the 3D structure of simple shapes from 2D images. With the goal of understanding a scene from visual cues, computer vision was viewed as a comparably easy first step towards solving higher-level reasoning tasks in *robotics* at that time (e.g., the MIT copy demo [104]). Albeit being seemingly easy for humans, robustly perceiving geometry and semantics from images proved hard for machines due to the high complexity of real-world environments. Thus, in the 1980s, computer vision and robotics evolved into their own, largely independent research fields. Only recently, the communities have realized that it is impossible to solve one without the other, e.g., in the context of self-driving [47]. Similarly, computer vision's interaction with *computer graphics* emerged in the 1990s [92] and has gained traction over the last decade, in particular in areas such as neural and image-based rendering [64]. These advances can in turn benefit robotics as simulation will be crucial for training and validating the next generation of robotic systems.

The converging trend of vision, graphics, and robotics motivates us to create a new dataset, KITTI-360, that addresses tasks at the intersection of these fields with a focus on autonomous driving. While the KITTI dataset [34] has pushed the state-of-the-art in computer vision algorithms forward, it does not contain dense and complete semantic labels. Thus, many interesting interdisciplinary tasks, e.g., synthesizing novel view images jointly with semantics or reconstructing large-scale semantic maps, cannot be evaluated on KITTI. Moreover, the captured perspective front images provide only a partial view of the scene and the 3D information provided by the LiDAR sensor is very sparse. The GPS localization of KITTI is reliable but does not reach sub-pixel accuracy when fusing multiple frames. With KITTI-360 we address these shortcomings by providing a new dataset with more comprehensive semantic/instance labels in 2D and 3D, richer 360° sensory information (fisheye images and pushbroom laser scans), very accurate and geo-localized vehicle and camera poses, and a series of new challenging benchmarks, see Fig. 1 for an overview.

A key challenge towards building such a dataset is to obtain coherent dense and comprehensive semantics in 2D and 3D. Many existing datasets are annotated in the 2D image domain where pixel-wise labeling requires up to 60 minutes per image for a human annotator[6]. Other datasets [8], [96], [119] are annotated in 3D space while ignoring information in the 2D image domain. A few datasets [18] offer labels in both 2D and 3D. However, annotation is conducted independently, thus duplicating the labeling effort.

- *Yiyi Liao is with Autonomous Vision Group, University of Tübingen and Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany, and also with Zhejiang University, Hangzhou 310027, China. E-mail: yiyi.liao@tue.mpg.de.*
- *Jun Xie is with Google Research, Mountain View, CA 94043 USA. E-mail: junx@google.com.*
- *Andreas Geiger is with Autonomous Vision Group, University of Tübingen and Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany. E-mail: a.geiger@uni-tuebingen.de.*

Manuscript received 20 Sept. 2021; revised 11 Feb. 2022; accepted 20 May 2022. Date of publication 1 June 2022; date of current version 3 Feb. 2023.
The work of Andreas Geiger was supported in part by the ERC Starting under Grant LEGO-3D 850533 and the DFG EXC number 2064/1 - project number 390727645. The work of Yiyi Liao was supported in part by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A.
(Corresponding author: Yiyi Liao.)
Recommended for acceptance by K. Barnard.
Digital Object Identifier no. 10.1109/TPAMI.2022.3179507

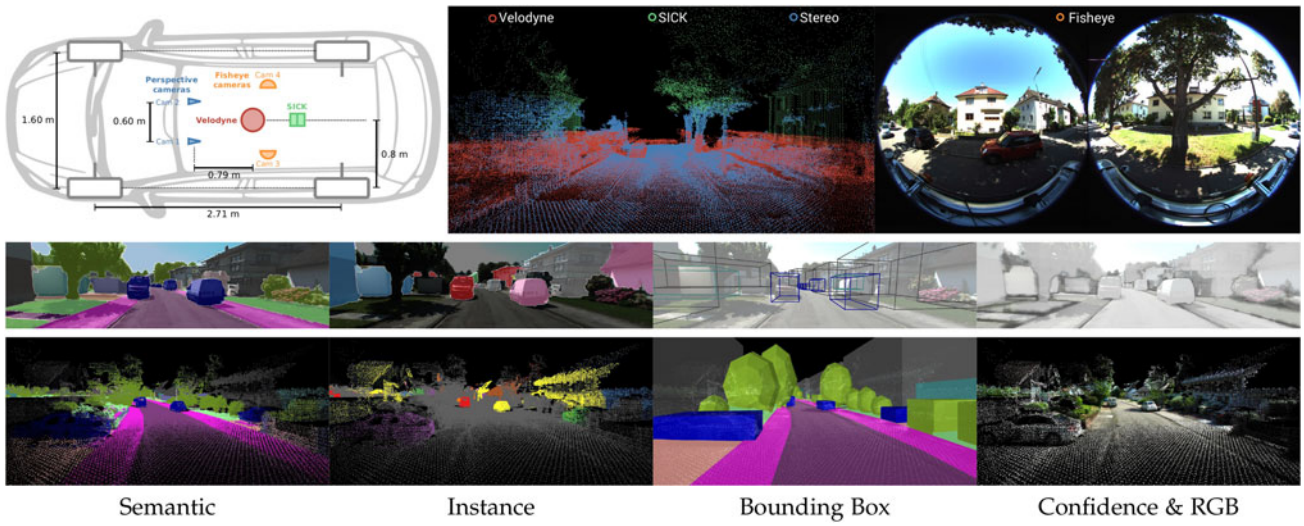


Fig. 1. *KITTI-360.* Our dataset contains rich sensor modalities, including a perspective stereo camera, a pair of fisheye cameras, a Velodyne and a SICK laser scanning unit which together enable 360° scene perception. We release comprehensive annotations including consistent semantic and instance labels for every 2D image pixel and 3D point.

In this paper, we propose an alternative approach that leverages coarse 3D annotations to significantly simplify the dense annotation task and establish coherent labels in both 2D and 3D space. Moreover, this yields a unique *instance* index for each object in the scene across all 2D video frames. Specifically, we build a WebGL-based annotation tool that allows for annotating both *static* and *dynamic* scene elements directly in 3D using simple primitives. This approach has several advantages over labeling in 2D: First, objects often project into several video frames, thus lowering annotation efforts considerably. Further, the obtained 2D instance annotations are temporally coherent as they are associated with a single physical 3D object. Finally, our 3D annotations covering the full 3D scene are useful on their own, e.g., for reasoning in 3D [35], [114] or to enrich 2D annotations with approximate 3D geometry.

However, obtaining dense and accurate pixel-wise 2D labels and point-wise 3D labels from sparse, noisy point clouds and coarse 3D annotations is a challenging task. Towards solving this problem, we propose a non-local multi-field CRF model which reasons jointly about semantic and instance labels of all 3D points and 2D pixels. Our approach also leverages learning-based methods to provide dense semantic and instance priors in the 2D image domain. As evidenced by our experiments, our method outperforms label propagation methods operating purely in 2D as well as pure learning-based approaches. Furthermore, the probabilistic nature of our model allows for estimating label uncertainties which can be used to increase label accuracy when only a subset of the pixels require a label.

From the annotated dataset, we derive several benchmarks and baselines with novel and challenging tasks at the intersection of vision, graphics and robotics which we believe are crucial for making progress towards the grand challenge of fully autonomous driving. Our *semantic scene understanding* benchmark includes tasks for 2D/3D recognition and semantic scene completion. The former requires predicting a semantic/instance label for the visible part of the scene, while the latter aims for joint geometric completion and semantic perception that can benefit higher-level reasoning, e.g., control and planning. In our *novel view synthesis* benchmark, we establish a challenging task that requires synthesis of both RGB appearance and semantic labels at a given novel viewpoint, aiming to foster research on building fully labeled simulation environments from real-world images. Lastly, our *semantic SLAM* benchmark evaluates vehicle localization as well as geometric and semantic 3D reconstruction over long sequences.

We summarize the contributions of this paper as follows:

- We present a novel georegistered dataset of suburban scenes recorded by a moving platform. The dataset comprises over 300k images and 80k laser scans.
- We create and release a WebGL-based annotation tool that allows for labeling street scenes in 3D space. Exploiting our annotation tool, we obtain 3D annotations for all static and dynamic scene elements.
- We propose a method which transfers these labels from 3D into 2D, yielding pixel-wise semantic instance annotations. We validate our approach in ablation studies and demonstrate its potential with respect to several 2D and 3D baselines.
- Enabled by our dense and coherent semantic instance annotations in both 2D and 3D as well as accurate vehicle and camera poses, we establish an online benchmark with novel and challenging tasks at the intersection of computer vision, graphics and robotics. We believe that our dataset and benchmarks will complement existing datasets and foster novel research towards solving the grand goal of full autonomy.

This journal paper is an extension of a conference paper published at CVPR 2016 [107]. In comparison to [107], we 1) extend our annotation tool and update our inference algorithm to support the annotation of dynamic objects; 2) provide a detailed description of the annotation tool and process; 3) establish new online benchmarks with held-out test data on a set of challenging tasks; 4) propose and evaluate several baselines to bootstrap the leaderboards and

TABLE 1
Overview of Publicly Available Datasets

Dataset	2D Annotations			3D Annotations					Coherency		Test
	#Smt. Img.	#Ins. Img.	Dense	#Smt. Pts.	#Ins. Pts.	FoV Azm.	FoV Plr.	#3D Bbox	Temporal	3D-2D	Server
CamVid [13]	631	–	✓	–	–	–	–	–	✓	–	–
DUS [1]	1k	–	✓	–	–	–	–	–	✓	–	–
CityScape (fine) [25]	5k	5k	✓	–	–	–	–	–	–	–	✓
CityScape (coarse) [25]	20k	20k	–	–	–	–	–	–	–	–	✓
Mapillary Vistas [70]	25k	25k	✓	–	–	–	–	–	–	–	–
CityScape-VPS [52]	3k	3k	✓	–	–	–	–	–	✓	–	–
KITTI-STEP [103]	19k	19k	✓	–	–	–	–	–	✓	–	✓
WoodScape [111]	10k	10k	✓	–	–	–	–	–	✓	–	–
Toronto-3D [96]	–	–	–	78.3M	–	**360°**	40°	–	–	–	–
Paris-Lille-3D [88]	–	–	–	143.1M	–	**360°**	40°	–	–	–	✓
DublinCity [119]	–	–	–	260M	–	-	-	–	–	–	–
Semantic3D.net [39]	–	–	–	4.0B	–	**360°**	**180°**	–	–	–	✓
SemanticKITTI [8]	–	–	–	**4.5B**	–	**360°**	26.8°	–	–	–	✓
Argoverse [21]	–	–	–	–	–	–	–	993k	–	–	–
Lyft [50]	–	–	–	–	–	–	–	**1.3M**	–	–	–
Waymo [94]	–	–	–	–	–	–	–	12M	–	–	✓
A*3D [75]	–	–	–	–	–	–	–	230k	–	–	–
KITTI [34]	200	200	✓	–	–	–	–	200k	–	–	✓
ApolloScape [45]	144k	90k	✓	–	–	–	–	70k	–	–	✓
nuScenes [18]	93k	93k	–	1.2B	78.9M	**360°**	40°	1.2M	–	–	✓
A2D2 [36]	41k	41k	✓	387.1M	23.8M	60°	30°	43k	–	✓	–
SemKITTI-DVPS [80]	23k	23k	–	**4.5B**	**400M**	**360°**	26.8°	–	✓	✓	✓
KITTI-360 (Ours)	**2× 78k**	**2× 78k**	✓	1.0B	172.4M	**360°**	120°	68k	✓	✓	✓

For pixel-level 2D annotations, we compare the number of semantic maps (#Smt. Img.), the number of instance maps (#Ins. Img.) and the density of the 2D semantic maps (Density). Note that the proposed KITTI-360 dataset includes images from both left and right view of the stereo camera. For 3D annotation, we show the number of semantically labeled points (#Smt. Pts.), the number of points with instance labels (#Ins. Pts.), the field of view with 3D semantic annotations at both azimuthal and polar directions (FoV Azm. and FoV Plr.), and the number of 3D bounding boxes (#3D Bbox). We further compare temporal consistency and 3D-to-2D consistency of the instance labels. The last row indicates whether the dataset hosts an online evaluation benchmark with held-out ground truth.

assess the difficulties of the tasks. We make our dataset,[1] utility scripts[2] and annotation tool[3] publicly available.

2 Related Work

In this section, we first discuss existing datasets in the context of autonomous driving, followed by a review of current methods for efficient (semi-automatic) label annotation.

2.1 Datasets

Indoor Video Datasets. Several datasets provide annotations for video sequences captured in indoor scenes [20], [26], [93], [105]. The SUN RGB-D dataset [93] provides labeled 2D polygons as well as 3D cuboids for 10k indoor RGB-D images. In a closely related work, ScanNet [26] is annotated in 3D with its 2D labels directly obtained from 3D-to-2D projection based on the dense depth from RGB-D sensors. In this work, we focus on outdoor street scenes where 3D observations are much more sparse, posing a challenging task for 3D-to-2D label transfer.

Outdoor Datasets. A number of outdoor datasets of driving scenes have been released in the literature [8], [9], [13], [18], [25], [34], [36], [45], [52], [56], [62], [65], [70], [80], [84], [96], [99], [103], [111], [119]. We summarize the most related ones in Table 1, categorized by whether they offer labels in the 2D image domain or in 3D space.

For datasets focusing on 2D labels, CamVid [13] is the first dataset for semantic segmentation in the context of self-driving. However, CamVid does not provide instance labels and only a very limited number of frames. Both Cityscapes [25] and Mapillary Vistas [70] release thousands of manually annotated 2D images. However, they do not offer temporally coherent semantic instance annotations. Recently, Cityscape-VPS [52] extends Cityscapes by providing semantic instance labels for every 5 frames. Furthermore, KITTI-STEP [103] offers spatially and temporally dense semantic instance annotations for the KITTI tracking dataset [34]. While aforementioned works focus on perspective images, WoodScape [111] releases semantic instance annotations of fisheye images. Our dataset differs from the above in that we provide not only temporally coherent semantic instance annotations for perspective images, but also omnidirectional imagery, 3D laser scans, and 3D annotations which are useful for 3D reasoning. While [25] focuses on inner-city scenes, our dataset comprises mainly suburban areas, thus both datasets complement each other (we use the same label definition to facilitate research).

Another line of works provides labels in 3D space. Toronto-3D [96], Paris-Lille-3D [88] and DublinCity [119] offer annotated point clouds collected from urban environments. Semantic3D.net [39] presents a large-scale dataset with 4 billion points, labeled with 8 semantic categories. SemanticKITTI [8] provides semantic labels for raw laser scans in KITTI, resulting in 4.5 billion labeled 3D points in 28 classes. Instead of focusing on point cloud semantic classification, Argoverse [21], Lyft [50], Waymo [94], and A*3D [75] offer

1. http://www.cvlibs.net/datasets/kitti-360
2. https://github.com/autonomousvision/kitti360scripts
3. https://github.com/autonomousvision/kitti360labeltool

2D/3D bounding boxes and establish benchmarks for 2D/3D detection and tracking.[4] In contrast to KITTI-360, the aforementioned datasets either lack dense annotations in images or they do not have per-point 3D annotations of stuff classes.

Our dataset provides labels for both 2D images and corresponding 3D points. Within this category, KITTI [34] provides dense semantic information on 200 images and 200k 3D bounding boxes. However, KITTI does not provide dense (per-point) 3D labels on the point cloud. Closely related to our work, ApolloScape [45] annotates static scene elements in the 3D space[5] and projects them to the 2D image space, followed by manual annotation of dynamic objects in images. In this work, we annotate both static and dynamic objects in 3D, providing coherent annotations for dynamic objects both in 2D and 3D. More recently, nuScenes [18] and A2D2 [36] released labels in both 2D and 3D. However, the labels of nuScenes are manually and independently annotated in 2D and 3D, and not every pixel is labeled in 2D. In contrast, we propose to leverage labels in the 3D space to infer dense labels in the image domain, thus providing consistent labels across 2D and 3D space. A2D2 [36] labels 2D images and maps 2D labels to 3D to obtain per-point 3D labels. Thus, the 3D labels are limited to a small FoV of the cameras in the azimuthal direction. While A2D2 also provides 3D bounding boxes, all of them are within the FoV of the forward-facing camera. We instead offer per-point 3D labels and 3D bounding boxes within an azimuthal FoV of 360°. A concurrent work, SemKITTI-DVPS [80] provides labels in both 2D and 3D by projecting the 3D labels of SemanticKITTI to images. Compared to the projected sparse 2D labels of SemKITTI-DVPS, KITTI-360 offers dense pixel-wise labels and additionally provides 3D bounding boxes.

There also exist several synthetic urban datasets [17], [32], [82], [86]. However, there still exists a significant perceptual gap between the virtual and real domains [43], [97], making synthetic-to-real generalization difficult.

Benchmarks. Recently, evaluation benchmarks have been widely recognized by the community. Some of the previously mentioned datasets also provide online evaluation benchmarks and held-out test data for different tasks. For instance, Cityscapes [25] offers a benchmark suite for pixel and instance-level semantic segmentation as well as 3D vehicle detection. SemanticKITTI [4], [8] hosts lidar segmentation challenges to predict the category of every point. For datasets including both 2D and 3D annotations, KITTI [33], nuScenes [18], and ApolloScape [45] provide benchmarks on a set of vision tasks including detection, stereo, localization, multi-object tracking, and segmentation in both 2D and 3D, etc. Moving beyond the established tasks, KITTI-360 provides novel benchmarks and will hold new challenges, e.g., on novel view semantic synthesis and semantic SLAM, to foster new progress towards full autonomy.

2.2 Methods

Efficient Annotation. Many works have attempted to reduce the per pixel annotation time of individual images, including classical methods [38], [60] and learning based methods [2], [3], [19], [59]. While all of these methods focus on annotating images individually, we are interested in annotating 2D video sequences as well as 3D scenes. There is also a growing interest in autolabeling 3D shapes or 3D bounding boxes [79], [112]. These methods are only applicable to a specific class, e.g., vehicles. We instead annotate the full 3D scene and aim to obtain coherent per-pixel 2D annotations and per-point 3D annotations.

2D Label Propagation. Compared to annotating individual images, video sequences offer the advantage of temporal coherence between adjacent frames. Label propagation techniques exploit this fact by transferring labels from a sparse set of annotated keyframes to all unlabeled frames based on color and motion information. While in some works a single foreground object is assumed [46], here we focus on methods that can handle multiple object categories. Towards this goal, [6] and [15] propose a coupled Bayesian network based on video epitomes and semantic regions to propagate label information between two annotated keyframes. [118] proposes a joint propagation strategy with synthesized training samples. To better account for errors in label propagation, [68] proposes a hierarchy of local classifiers for this task and [5] leverages a mixture-of-tree model for temporal association. The work of [16] leverages label propagation as a data augmentation scheme and demonstrate improved performance on semantic segmentation. Optical flow is also commonly used for semantic video label transfer. [31] uses optical flow of adjacent frames to warp network representations across time and thus propagates labels from previous frames to the current one. [117] proposes to run a convolutional sub-network only on sparse keyframes and propagate the deep feature maps to other frames via flow fields. In the indoor scenario where dense geometry is available, [81] proposes a method on RGB-D video propagating labels on super-pixel.

In contrast to the aforementioned methods which propagate labels in 2D, in this paper we propose to annotate both semantic and instance labels directly in 3D and then project these annotations into the 2D domain. While this approach requires a source of 3D information (e.g., SfM, stereo, laser), it is able to produce more accurate semantic and temporally consistent instance annotations for tracking purposes. Further, our experiments indicate that annotation in 3D is more time-efficient than labeling in 2D as scene elements can be separated more easily and often project into many images of the input video sequence while being only annotated once.

3D-to-2D Label Propagation. There are a few existing works on 3D-to-2D label transfer. Chen *et al.* [22] leverage annotations from KITTI [33] as well as 3D car models to infer separate figure-ground segmentation for all vehicles in the image. In comparison, our approach reasons jointly about all objects in the scene and also handles categories for which CAD models or 3D point measurements are unavailable (e.g., "Tree", "Sky"). Huang *et al.*[45] also applies 3D to 2D label transfer for generating the ApolloScape dataset. In this work, labels are transferred from 3D point clouds to images with simple splatting and projection. However, 3D points are too sparse compared with image pixels, thus, setting the splatting range is not trivial. Similarly, in [26],

4. We refer to 2D annotations as pixel-level annotations in Table 1. 2D bounding boxes are not included.

5. The 3D annotation has not been released yet.

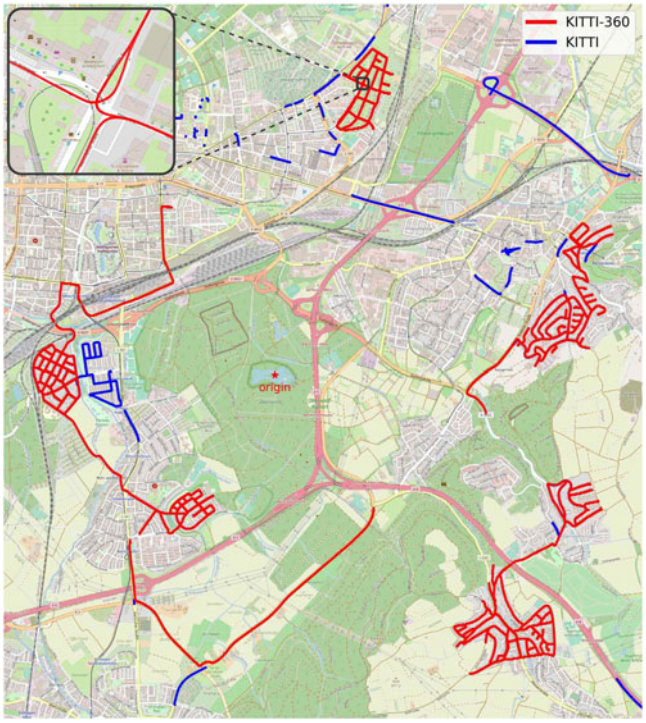


Fig. 2. Georegistered poses overlaid on OpenStreetMap.

semantic labels annotated in the reconstructed scene are projected into each frame but not all 2D pixels are covered due to missing geometry. In addition, the two aforementioned works are limited to static scenes.

In the context of street view image segmentation, [14], [63], [65], [67], [69], [106] exploit the interaction between image pixels and 3D points to improve classification performance or efficiency. In comparison, our goal is to transfer ambiguous 3D primitive labels to every pixel in the image.

3 Annotation

In this section, we describe our data collection efforts, data preprocessing, the annotation tool, and annotation details.

3.1 Data Collection

For data collection, we equipped a station wagon with one 180° fisheye camera to each side and a 90° perspective stereo camera (baseline 60 cm) to the front. Furthermore, we mounted a Velodyne HDL-64E and a SICK LMS 200 laser scanning unit in pushbroom configuration on top of the roof. This setup is similar to the one used in KITTI [33], [34], except that we gain a wider field of view with the additional fisheye cameras and the pushbroom laser scanner while KITTI only provides perspective images and Velodyne laser scans with a 26.8° vertical field of view. Compared to omnidirectional camera systems [90], [91], our setup benefits from increased resolution of the 3D reconstruction. Localization is provided by IMU and GPS which we fuse with visual features. Fig. 1 (top left) illustrates our setup.

Using this setup, we recorded several suburbs of a midsize city corresponding to over 300k images and 80k laser scans, covering a driving distance of 73.7km. We estimate all vehicle and camera poses using structure-from-motion [41]. More specifically, we minimize 3D reprojection errors

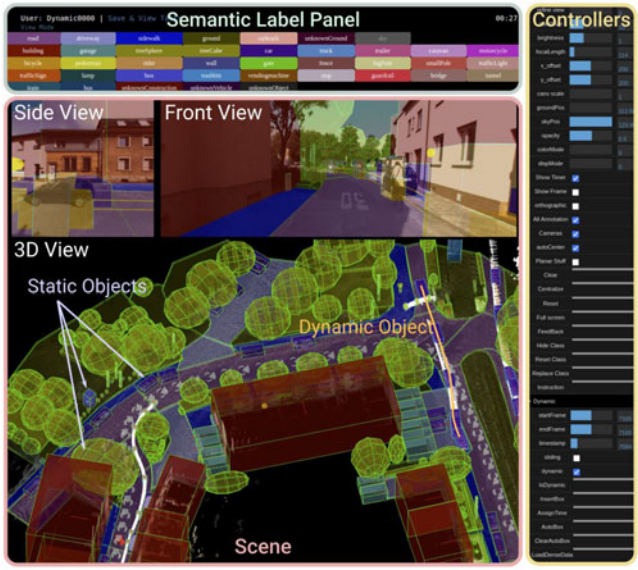


Fig. 3. *Annotation interface.* Our interface consists of three main components: scene view (perspective views and 3D view), semantic label panel, and controllers.

based on all feature matches while regularizing against the GPS location. We further add loop-closures detected from LiDAR scans as regularization to complement image feature matching (which might fail on opposite-facing frames). This results in accurate georegistered camera poses. Fig. 2 illustrates the camera poses overlaid on OpenStreetMap.[6] We also plot the camera poses of the KITTI dataset [33], [34] for reference. KITTI-360 follows KITTI's forward facing camera configuration, but has minimal overlap with KITTI in terms of trajectories. This allows us to split training and test data without conflicting with the KITTI dataset, e.g., avoiding the situation where a region is used for training in KITTI but testing in KITTI-360. Following KITTI, we use Mercator projection [73] to convert geographic coordinates to a local euclidean coordinate frame in order to facilitate usage of the dataset. The origin of the coordinate frame is chosen as the center of the map as illustrated in Fig. 2.

3.2 Annotation Interface

To facilitate 3D annotation, we developed an online annotation tool based on WebGL. We release our annotation tool (see Fig. 3) as part of this project. It consists of three main components: a scene viewer (including 2D images and 3D scene), a semantic label selection panel, and controllers. Annotators are asked to insert 3D primitives with adjustable shapes and semantic labels into the 3D scene.

3.2.1 Scene

To annotate the data while limiting transfer bandwidth, we split the collected data into batches according to the accumulated driving distances. Specifically, a single batch contains observations within a driving distance of about 200 meters (240 frames on average) and there is an overlap of 10 meters between two consecutive batches. Within one batch,

6. http://www.openstreetmap.org/

we accumulate 3D points observed from the Velodyne and SICK laser scanning unit as well as the stereo camera.

During annotation, the accumulated point clouds are downsampled to reduce data loading traffic and memory. However, downsampling makes it hard to precisely perceive dynamic objects whose 3D observations are distributed along a moving trajectory. To allow for accurate labeling of dynamic objects, we apply a simple heuristic to detect dynamic objects, see Appendix A.2, which can be found on the Computer Society Digital Library at[7]. We then load all detected dynamic points at each frame into the annotation tool without down-sampling. To help the annotators efficiently identifying dynamic objects, we highlight dynamic objects using white color as illustrated in Fig. 3.

As auxiliary visualization to the 3D point clouds, we provide fisheye and perspective images (see "Side View" and "Front View" in Fig. 3) in order to allow annotators to select and perceive the scene from different camera views. We also visualize the pose of each camera, enabling annotators to quickly select informative viewpoints.

3.2.2 Semantic Label Panel and Controllers

We show semantic labels with different colors in the label panel for users to choose from. To better assist annotators in placing the primitives accurately, we also offer easy-to-use controllers to interact with the 3D scene, including zoom, pan, rotation of the point cloud, switching data sources or camera views, and toggling annotations. We provide more details about the annotation interface in Appendix B, available in the online supplemental material.

3.3 Annotation Details

We ask the annotators to annotate the 3D point clouds in the form of bounding primitives, i.e., place cuboids and ellipsoids to enclose objects in 3D and assign a semantic label to each of them. The 3D scene is annotated with 37 label classes, including 24 "instance" classes and 13 "stuff" classes. Labels are defined in accordance with the Cityscapes dataset [25] label definition. More details about the label definition can be found in Appendix C, available in the online supplemental material. The annotations are categorized into static and dynamic objects, which are treated differently by our annotation tool.

3.3.1 Static Objects Annotation

Static labels can be further classified into two categories: "stuff" and "instance". For *instance* classes, each object is constrained to be associated with only one cuboid primitive, representing both semantic and instance labels of this object. We ask the annotators to tightly enclose the point clouds with the bounding primitives. For *stuff* classes, which usually have irregular shapes, annotators are allowed to use multiple cuboids or ellipsoids to roughly enclose the 3D points of the target objects.

We also provide a "planar" annotation option for stuff categories on the ground such as "Road" and "Sidewalk". Using this option, we allow annotators to draw a 2D polygon representing the ground object's boundary in bird's eye view. The interface then automatically estimates the height of the polygon based on the surrounding 3D geometry and extrudes the 2D polygon into 3D along the vertical direction to enclose corresponding 3D ground points. We provide more details in Appendix B.3, available in the online supplemental material.

3.3.2 Dynamic Objects Annotation

Dynamic objects mainly comprise moving vehicle and pedestrian instances. In contrast to ApolloScape [45] which annotates static objects in 3D and dynamic objects in 2D respectively, we annotate both static and dynamic objects in 3D space. However, compared to static objects, annotating dynamic objects in 3D outdoor scenes is more challenging as individual dynamic objects in the 3D reconstruction are hard to perceive and distinguish. Moreover, we need to label not only where the moving instance is, but also "when" the instance appears, requiring the annotation of moving 3D bounding boxes over time. A naïve solution is to place a 3D bounding box in every frame where the dynamic object is present. However, such an annotation process would be intractably slow. Thus, we instead implement a semi-automatic annotation scheme to reduce label time. Specifically, we minimize the effort required by annotators by making two assumptions: the size of the dynamic object is fixed over time and its trajectory is smooth. Under these assumptions, the required annotation is reduced to the size of a single 3D primitive and the pose of this primitive at several keyframes. Our annotation tool then automatically places the remaining primitives along the trajectory, see Appendix B.4, available in the online supplemental material, for more details.

3.4 Annotation Procedure

We annotated 379 batches in total, assigning one batch to one annotator. To control the annotation quality, we train and evaluate the annotators based on multiple pilot tasks until they have proven qualified for the full task. We also regularly verify their annotation quality and ask them for correction if necessary. We further identify a few annotators who consistently produced high-quality labels and ask them to cross-check other annotators' quality. Our annotation interface simplifies the detection and correction of annotation errors compared to annotating image sequences, which requires corrections across multiple frames. Fig. 3 shows parts of an annotated batch via our web interface.

3.5 Annotation Time

On average, annotating one full batch ($\sim$ 240 frames) in 3D required about 3 hours. Thus, our annotators spend only 3 $\times$ 60 / 240 = 0.75 minutes for "annotating" one image. In comparison, 7 minutes are required for coarse annotation of semantic instance labels in the image domain, and 1.5 hours for pixel-accurate annotations as discussed by the creators of the Cityscapes dataset [25].

4 Label Transfer Method

In this section, we first provide an overview of our method for transferring the 3D annotation to semantic instance

7. http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3179507.

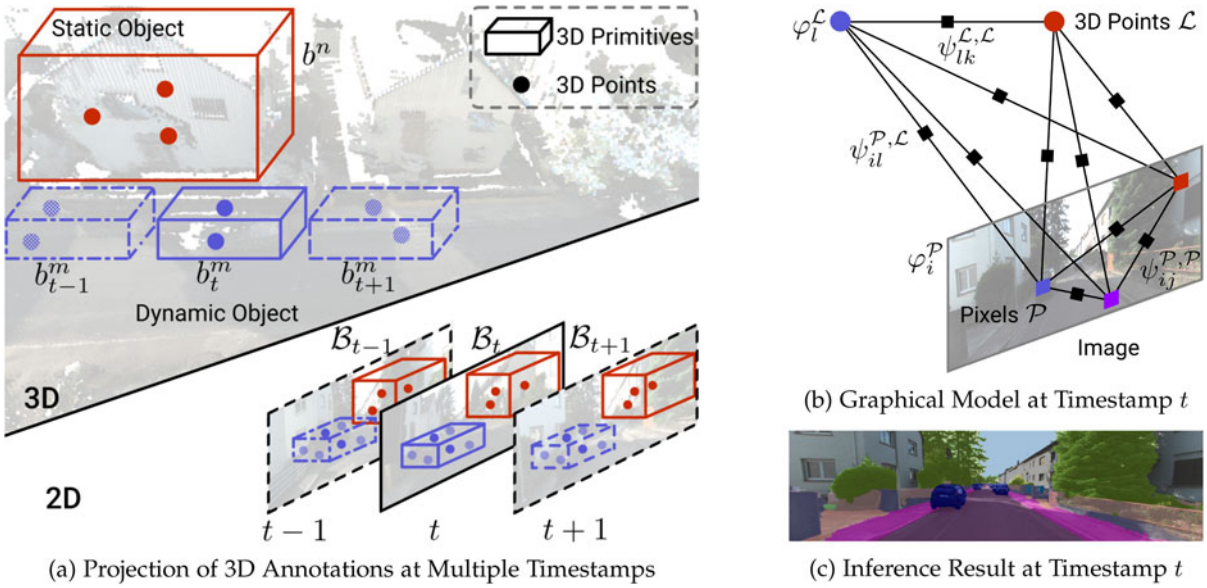


Fig. 4. *3D-to-2D label transfer.* (a) We illustrate the 3D-to-2D projection of static and dynamic object annotations. A static 3D primitive is projected to multiple frames while a dynamic 3D object is projected only into the corresponding frame. (b) Factor graph representation of our model. Note that the CRF model is defined over all pixels and visible 3D points at a single timestamp. (c) We show the semantic inference result at timestamp t.

annotations in 2D. Next, we formally introduce the model and discuss parameter learning and inference.

4.1 Overview

Given 3D annotations, we are interested in generating dense semantic instance annotations for all images and all 3D points. To incorporate inductive biases about image formation and label smoothness, we explore a Conditional Random Field (CRF) model which reasons jointly about the labels of the *3D points* and *all pixels* in the image. In practice, we apply the CRF at every timestamp independently to keep inference tractable. Despite independent inference, we are able to obtain consistent results over multiple frames thanks to the shared 3D annotations. We also experimented with inference over multiple adjacent frames but did not observe measurable improvements.

Let $\mathcal{B}_t = \{\{b^n\}, \{b^m_t\}\}$ denote all 3D annotations available at timestamp t. Here, b^n and b^m_t correspond to 3D bounding primitives of static and dynamic objects respectively, with n and m indexing each primitive. Note that a static primitive b^n is used at all timestamps (if visible) whereas a dynamic primitive b^m_t is only included in $\mathcal{B}_t$ when it is labeled to appear at timestamp t. Fig. 4a illustrates static and dynamic bounding primitives as well as their projection into the 2D image domain. With this design, our framework allows for annotating the same object using a unique instance ID across the entire sequence as well as across 2D and 3D.

Let $\mathcal{P}_t$ denote the set of image pixels at timestamp t and $\mathcal{L}_t$ denote the visible 3D points at the same timestamp. The CRF model is defined over all elements in $\mathcal{P}_t$ and $\mathcal{L}_t$. To obtain more complete 3D information, $\mathcal{L}_t$ fuses stereo and laser scans over multiple frames. We first fuse points covering static parts of the scene, and then accumulate points of each dynamic object according to its bounding primitives and insert them into the static scene depending on the location of b^m_t. We provide more details regarding the accumulation of static and dynamic 3D points in Appendix D.1, available in the online supplemental material.

4.2 Model

We now formalize the CRF model applied at every frame as illustrated in Fig. 4b. Note that our 3D annotations are sparse and noisy, i.e., 3D points can carry none, one or multiple labels due to overlapping bounding primitives in 3D. The algorithm described in this section is designed to resolve these situations and infers marginal estimates for all 3D points and pixels in the image.

As the CRF model is applied at every frame independently, we drop the dependency on timestamp t of $\mathcal{B}, \mathcal{L}$ and $\mathcal{P}$ for simplicity. For each pixel $i \in \mathcal{P}$ and each 3D point $l \in \mathcal{L}$, we specify random variables s_i and s_l taking values from the set of semantic (or instance) labels $\{1, \ldots, S\}$, where S denotes the number of classes. For instance inference, we assign a unique ID to each object which projects into the image. Thus, semantic and instance inference can be treated equally under our model and we will refer to both as "semantic labels" in the following. Note that there is no need to distinguish static or dynamic objects in the single frame-based CRF model. Still, we are able to retrieve whether a pixel or a 3D point belongs to a dynamic object or not according to its instance ID.

Let $\mathbf{s} = \{s_i | i \in \mathcal{P}\} \cup \{s_l | l \in \mathcal{L}\}$ denote the set of semantic labels. Dropping all dependencies on the image and point cloud for clarity we specify our CRF in terms of the following Gibbs energy function:

$$E(\mathbf{s}) = \sum_{i \in \mathcal{P}} \varphi^{\mathcal{P}}_i(s_i) + \sum_{l \in \mathcal{L}} \varphi^{\mathcal{L}}_l(s_l) + \sum_{i,j \in \mathcal{P}} \psi^{\mathcal{P},\mathcal{P}}_{ij}(s_i, s_j) + \sum_{l,k \in \mathcal{L}} \psi^{\mathcal{L},\mathcal{L}}_{lk}(s_l, s_k) + \sum_{i \in \mathcal{P}, l \in \mathcal{L}} \psi^{\mathcal{P},\mathcal{L}}_{il}(s_i, s_l), \quad (1)$$

with unary potentials $\varphi(\cdot)$ and pairwise potentials $\psi(\cdot)$. For notational clarity, we omit all conditional dependencies on the input images, 3D points and 3D annotations.

Pixel Unary Potentials. The pixel unary potentials $\varphi_i^{\mathcal{P}}(s_i)$ encode the likelihood of pixel i taking label s_i

$$\varphi_i^{\mathcal{P}}(s_i) = w_1^{\mathcal{P}}(s_i)\,\xi_i^{\mathcal{P}}(s_i) - w_2^{\mathcal{P}}(s_i)\log p_i^{\mathcal{P}}(s_i), \tag{2}$$

where $w_1^{\mathcal{P}}$ and $w_2^{\mathcal{P}}$ denote learned feature weights. Our first constraint $\xi_i^{\mathcal{P}}(s_i)$ determines the set of admissible labels and is obtained by projecting all 3D bounding primitives $\mathcal{B}$ (which are an upper bound on the objects' extent) into the image. We formulate the constraint via a binary feature $\xi_i^{\mathcal{P}}(s_i) \in \{0, 1\}$ which takes 0 for pixel i if its ray passes through a primitive of class s_i, and 1 otherwise.

In addition, we exploit a data-driven approach in order to obtain a per-pixel probability distribution over semantic labels $p_i^{\mathcal{P}}(s_i)$. Specifically, we project all non-occluded and uniquely labeled sparse 3D points into the image plane, and use these sparse projections as supervision to train a semantic segmentation network (PSPNet [116]) on the entire dataset. The output of the network's last layer is taken as the probability distribution. We also augment the training dataset using Cityscape images and labels [25] to enable the model to learn accurate object boundaries which is difficult to learn based on the projection of sparse and noisy LiDAR point clouds. As the semantic segmentation model does not distinguish instances, we further adopt a state-of-the-art instance segmentation method [108] to obtain instance hypotheses for "car", "truck", and "pedestrian". Thus, we effectively exploit the inductive biases of modern neural network architectures and co-training on related labeled datasets. As demonstrated in Appendix D.4, available in the online supplemental material, this leads to a significant improvement at object boundaries.

3D Point Unary Potentials. The 3D point unary potentials $\varphi_l^{\mathcal{L}}(s_l)$ encode the likelihood of 3D point l taking label s_l

$$\varphi_l^{\mathcal{L}}(s_l) = -w^{\mathcal{L}}(s_l)\,\xi_l^{\mathcal{L}}(s_l), \tag{3}$$

where $\xi_l^{\mathcal{L}}(s_l)$ denotes a feature which takes 0 if the 3D point l lies within a 3D primitive of class s_l within $\mathcal{B}$, and 1 otherwise. As the "sky" class can't be modeled with primitives, we set $\xi_l^{\mathcal{L}}(s_l)$ to 0 if s_l takes the label "sky". Additionally, we create "virtual sky points" at infinity for all pixels whose ray doesn't intersect any 3D primitive. Note that these pixels must correspond to sky regions as we assume that the scene is densely annotated, hence each object is contained in one or several bounding 3D primitive(s).

Pixel Pairwise Potentials. Our dense pairwise term encourages semantic label coherence and connects all pixels in the image via Gaussian edge kernels following [55]

$$\begin{aligned}\psi_{ij}^{\mathcal{P},\mathcal{P}}(s_i, s_j) \;=\; & w_1^{\mathcal{P},\mathcal{P}}(s_i, s_j)\exp\left\{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\,\theta_1^{\mathcal{P},\mathcal{P}}}\right\} \\ + \; & w_2^{\mathcal{P},\mathcal{P}}(s_i, s_j)\exp\left\{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\,\theta_2^{\mathcal{P},\mathcal{P}}} - \frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\,\theta_3^{\mathcal{P},\mathcal{P}}}\right\},\end{aligned} \tag{4}$$

where $\mathbf{p}_i$ is the 2D location of pixel i and $\mathbf{c}_i$ denotes its color value. Further, $w_1^{\mathcal{P},\mathcal{P}}$ and $w_2^{\mathcal{P},\mathcal{P}}$ are learned pairwise feature weights and $\theta^{\mathcal{P},\mathcal{P}}$ parameterizes the kernel width.

3D Pairwise Potentials. Similarly, we apply a Gaussian edge kernel to encourage label consistency between 3D points based on their 3D location and surface normals

$$\begin{aligned}\psi_{lk}^{\mathcal{L},\mathcal{L}}(s_l, s_k) = \; & w^{\mathcal{L},\mathcal{L}}(s_l, s_k) \\ & \times \exp\left\{-\frac{\|\mathbf{p}_l^{3d} - \mathbf{p}_k^{3d}\|^2}{2\,\theta_1^{\mathcal{L},\mathcal{L}}} - \frac{(n_l - n_k)^2}{2\,\theta_2^{\mathcal{L},\mathcal{L}}}\right\},\end{aligned} \tag{5}$$

where $\mathbf{p}_l^{3d}$ is the 3D location of point l and n_l denotes the vertical (up) component of its normal. We use the normal's z-component as it is the most discriminative cue for label changes between horizontal (e.g., road, sidewalk) and vertical (e.g., side of car, wall) surfaces. We estimate the respective normals using principal component analysis in a local neighborhood around each 3D point.

2D/3D Pairwise Potentials. Finally, we encourage coherence between all 3D points and the image pixels

$$\psi_{il}^{\mathcal{P},\mathcal{L}}(s_i, s_l) = w^{\mathcal{P},\mathcal{L}}(s_i, s_l)\exp\left\{-\frac{\|\mathbf{p}_i - \boldsymbol{\pi}_l\|^2}{2\,\theta^{\mathcal{P},\mathcal{L}}}\right\}, \tag{6}$$

where $\boldsymbol{\pi}_l$ denotes the projection of the 3D laser or stereo point l onto the image plane. Importantly, we project only points into the image which are likely to be visible. We determine these points by meshing the 3D point cloud using the ball-pivoting method of Bernardini *et al.*[10], and considering only 3D points in front of the mesh. We also experimented with multi-view reconstruction approaches [48] for mesh generation, but obtained better results using this simpler approach. As applying the meshing algorithm independently for every frame is time-consuming, we generate meshes on entire batches, processing the static part and dynamic objects independently. This allows us to reuse the mesh of the static part for all frames of a batch.

4.3 Learning and Inference

This section describes inference and parameter estimation in our label transfer model.

Inference. At test time, we are interested in estimating the marginal distribution of each semantic or instance label in $\mathbf{s}$ under our model, specified by the Gibbs distribution defined in Eq. (1). A likely configuration can then be estimated by variable-wise maximization of these marginals. As our graphical model is loopy, exact inference in polynomial time is intractable. We thus resort to variational inference and approximate the probability distribution on $\mathbf{s}$ by replacing it with a factorized mean field distribution $Q(\mathbf{s}) = \prod_{i \in \mathcal{P} \cup \mathcal{L}} Q_i(s_i)$. This mean field approximation can be computed efficiently using bilateral filtering [55]. As our model comprises three sets of densely connected variables (namely $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{P} \leftrightarrow \mathcal{L}$), we exploit the algorithm of [51], [101] which generalizes [55] to multiple fields. Fig. 4c illustrates the inference result for a single frame, overlaid on the corresponding input image. Moreover, we obtain an uncertainty estimate for each pixel/3D point by computing the entropy over the respective marginal distribution. We will use this estimate in Section 6 to weigh the evaluation metrics according to the confidence of our label estimates.

Learning. We employ empirical risk minimization in order to learn the parameters in our model, considering the univariate logistic loss, defined as $\Delta(s) = -\log(P(s))$ where $P(\cdot)$ denotes the marginal distribution at the respective site. Let us subsume all model parameters into $\boldsymbol{\Theta} = \{w_1^{\mathcal{P}}, w_2^{\mathcal{P}}, w^{\mathcal{L}}, w_1^{\mathcal{P},\mathcal{P}}, w_2^{\mathcal{P},\mathcal{P}}, w^{\mathcal{P},\mathcal{L}}, w^{\mathcal{L},\mathcal{L}}\}$. We define our minimization

objective $f(\Theta)$ as the regularized univariate logistic loss

$$f(\Theta) = \sum_{n=1}^{N} \sum_{i \in \mathcal{P}} -\log\left(Q_{n,i}(s^*_{n,i})\right) + \lambda C(\Theta). \quad (7)$$

Here, N is the number of training images, $s^*_{n,i}$ denotes the ground truth semantic label and $Q_{n,i}(\cdot)$ the approximate marginal at pixel i in image n, calculated via mean field approximation. $C(\Theta)$ is a quadratic regularizer on the parameter vector Θ. We whiten all features and use a single value λ which we select via cross-validation on the training set. For learning the instance segmentation parameters we exploit the same loss $f(\Theta)$ as for semantic segmentation, but assign unique labels to each individual object, e.g., different cars will be assigned different labels even if they occlude each other. In order to associate 2D ground truth instances with 3D instances we project all visible 3D points into the image and find a consensus via the majority vote which gave good results in practice. As the number of instances per semantic class varies between images, we learn intra- and inter-class pairwise potentials using parameter tying. We optimize the objective function $f(\Theta)$ using stochastic gradient descent and obtain $\partial Q / \partial \Theta$ using auto differentiation. We make use of the ADADELTA algorithm [113] with decay parameter 0.95 and $\epsilon = 10^{-8}$, and randomly sample a batch of 16 training images at each iteration for which all gradients can be computed in parallel.

5 Label Transfer Evaluation

In this section, we first introduce the datasets we use for training and evaluating our label transfer method. Next, we evaluate our method in ablation studies and compare it against several label transfer baselines. Finally, we also show qualitative results of our method.

5.1 Training and Evaluation Data

We manually annotate a set of images with pixel-wise ground truth to train and evaluate our label transfer method. The *training* set contains 125 images selected from diverse scenarios such that a substantial amount of pixels are labeled within each class. These training images are different from those used in our conference version [107]. We create this new training set following the label definition of CityScapes [25] as [107] considers fewer classes. To enable comparison to 2D label transfer methods which require images with large overlapping regions, we additionally annotate 240 adjacent frames from 13 different suburbs in equidistant steps of 5 frames in the 2D image domain for *evaluation*. The evaluation set has no spatial overlapping with the training set, allowing us to assess the generalization ability of our method. We evaluate our label transfer method on static and dynamic objects separately. Following [107], the performance of static objects is evaluated on 120 densely labeled frames from 5 suburbs containing the most frequently occurring 14 classes. The remaining 120 frames are sampled from 8 different suburbs which contain dynamic objects. For these frames we label the dynamic objects while leaving the static region unannotated. We consider 7 common dynamic objects, see Appendix E.1, available in the online supplemental material, for details.

TABLE 2
Comparison to Label Transfer Baselines on Semantic Segmentation Transfer

Method	Label source	Static		Dynamic	
		mIoU	Acc	mIoU	Acc
Label Prop.[100]	2D	49.0	81.0	37.2	59.1
Sparse Track. + GC [95]	2D	51.2	79.1	8.2	12.5
3D Prop. + GC	2D	72.1	87.4	14.5	21.7
Fully Conn. CRF [55]	2D	63.6	88.7	–	–
PSPNet [116]	CS + 3D	67.2	90.4	–	–
3D Primitives + GC	3D	49.4	73.4	–	–
3D Mesh + GC	3D	66.8	85.7	–	–
3D Points + GC	3D	72.6	87.8	–	–
Proposed Method	3D	**81.2**	**93.1**	**63.5**	**94.1**

We compare our method to 2D label transfer baselines (top) and to 3D to 2D label transfer baselines (bottom). Label source: "2D": Labeled neighboring image frames, "CS": Cityscapes training images, "3D": 3D bounding primitives.

5.2 Quantitative Evaluation

This section presents our quantitative evaluation on semantic and instance segmentation. We compare our method with several label transfer baselines and conduct ablation studies.

5.2.1 Semantic Segmentation Transfer

For evaluating semantic segmentation transfer performance, we measure overall performance by the mean intersection over union (mIoU) and the average pixel accuracy (Acc). While [107] evaluates the *weighted* mean IoU which is biased by object occurrences, we follow Cityscapes [25] and measure the mean IoU without weighting. For all experiments, we provide results for individual classes in Appendix E.1, available in the online supplemental material.

Baselines. We compare our method to several 2D to 2D label transfer methods on both static and dynamic objects in Table 2. Here, the task is to predict the center frame from two annotated images (± 5 frames corresponding to 0.5 seconds of driving or ~ 5 meters travel distance). Our first baseline ("Label Prop.") is the label transfer approach presented in [100]. To ensure that all baselines have access to the same information, we do not select frames actively but use equidistantly spaced frames for all methods. We construct a second baseline ("Sparse Track. + GC") using the feature tracking approach of [95] to propagate semantic labels from the two closest labeled frames to the target frame. To densify the label map, we apply graph cuts (GC) with contrast sensitive edge potentials [11]. In order to evaluate the value of 3D information, we implemented a third baseline ("3D Prop. + GC") which works similar to the previous one, but replaces the sparse tracking part with correspondences obtained by transferring pixels of the two closest labeled frames to the target image via the visible vertices of our 3D mesh followed by graph cuts propagation.

While all aforementioned baselines require labeled adjacent frames as input at inference time, we consider two more methods that generalize to arbitrary frames. First, we train the segmentation model of Krähenbühl *et al.*[55] ("Fully Conn. CRF") which was also used in [107] and

TABLE 3
Semantic Instance Segmentation Transfer Ablation
Evaluated on Static Objects

Method	Semantic		Instance	
	mIoU	Acc	mIoU	Acc
LA	67.6	88.7	72.3	86.7
LA+3D	70.4	89.7	72.9	88.7
LA+PW	66.6	87.9	72.9	85.8
LA+PW+CO	76.8	91.8	81.6	91.0
LA+PW+CO+3D	78.2	92.4	83.6	91.7
Full Model	**81.2**	**93.1**	**83.7**	**91.8**
Full Model (90%)	88.3	96.0	89.0	94.9
Full Model (80%)	92.5	97.6	91.3	96.6
Full Model (70%)	**94.3**	**98.4**	**92.7**	**97.4**

The components are abbreviated as follows: LA = local appearance ($p^{\mathcal{P}}$), PW = 2D pairwise constraints ($\psi^{\mathcal{P},\mathcal{P}}$), CO = 3D primitive constraints ($\xi^{\mathcal{P}}$), 3D = 3D points ($\varphi^{\mathcal{L}}, \psi^{\mathcal{P},\mathcal{L}}$), Full Model = all potentials including 3D pairwise constraints ($\psi^{\mathcal{L},\mathcal{L}}$). Percentages denote fractions of estimated pixels.

which uses a similar inference algorithm as our label transfer method on all annotated adjacent frames of the test sequence. Finally, we evaluate the deep semantic segmentation network [116] ("PSPNet") that also provides dense unary information for our method. As discussed in Section 4.2, this model is trained on non-occluded sparse 3D projections combined with the CityScapes training set [25]. Note that neither PSPNet nor our method has access to adjacent annotated frames for training or inference.

We further consider several 3D to 2D label transfer baselines that exploit our 3D annotations without requiring equidistantly labeled 2D annotations. Specifically, we project 3D primitives, meshes or visible 3D points into the 2D image domain, followed by graph cut inference ("3D Primitives + GC"; "3D Mesh + GC"; "3D Points + GC").

Static Objects. Table 3 (left) shows the comparison on 120 consecutive images of static objects.[8] From the 2D label transfer baselines shown at the top, the mesh transfer method which uses projected 3D information performs best in terms of mIoU. Furthermore, and maybe surprisingly, the sequence-specific fully connected CRF model performs on par or even better than special purpose label transfer methods. This is caused by the fact that optical flow (as used in [95], [100]) often fails for street scenes like ours due to large displacements, perspective distortions, textureless regions and challenging lighting conditions. Interestingly, PSPNet achieves the best accuracy while performing worse on mIoU. Despite obtaining superior results on large objects (e.g., "Building"), it struggles with less-occurring classes such as "Trailer" and "Gate".

The bottom half of Table 2 (left) compares the proposed method with respect to the 3D to 2D label transfer baselines. As evidenced by our results, simply projecting 3D primitives or meshes into the image and smoothing via GC does not perform well due to the crude approximation of the geometry. Better results are obtained when projecting the visible 3D points followed by spatial propagation. Finally, we observe that all baselines are outperformed by the proposed method (last row). Note that we also map the 37 semantic labels of our 3D annotations to the most common 14 categories considered in the static evaluation images (see Appendix C, available in the online supplemental material) for all 3D to 2D label transfer methods.

Dynamic Objects. We evaluate our method on dynamic objects against 2D label transfer baselines. Here, we consider all static regions as a single background class during evaluation. Note that we neglect "Fully Conn. CRF" and "PSPNet" as both methods address semantic segmentation and thus cannot distinguish static and dynamic objects within the same class. Table 2 (right) shows that our method also outperforms all 2D label transfer baselines on dynamic objects. While the mIoU is calculated over a different set of classes, the average performance of our method on dynamic objects is slightly degraded compared to our result on static objects. Labeling of dynamic objects is more challenging in our annotation pipeline for two reasons: Since we accumulate 3D points of dynamic points according to the annotated bounding primitives over multiple frames, slight misalignments of the primitives may lead to inaccurate accumulation and thus erroneous 3D cues. Furthermore, the accumulation of deformable objects ("Rider", "Person") leads to noisy 3D point clouds. Despite these challenges, our method achieves satisfying performance on all dynamic objects.

Annotation Time Comparison. While all 2D methods require every 10th frame to be labeled, our method (as well as the other 3D baselines) requires 3D annotations in the form of 3D primitives. Assuming 60 minutes annotation time per image, this amounts to 20 hours of annotation time per batch of 200 frames when labeling one 2D image every 10th frame, while the respective 3D annotations for this scene can be obtained in about 3 hours. This gain multiplies with the frame rate and the number of cameras (our setup has four).

Ablation Study. We validate the importance of the individual components of our model on semantic segmentation in Table 3 (upper left), evaluated on the densely labeled images of static objects. Starting with the appearance classifier $p^{\mathcal{P}}$ trained on the projected sparse 3D points ("LA"), we incrementally add the terms $\varphi^{\mathcal{L}}, \psi^{\mathcal{P},\mathcal{L}}$ related to the 3D points ("3D"), the semantic pairwise term $\psi^{\mathcal{P},\mathcal{P}}$ between pixels ("PW"), the 3D primitive constraints $\xi^{\mathcal{P}}$ ("CO") and finally the 3D pairwise constraints $\psi^{\mathcal{L},\mathcal{L}}$ as specified in Eq. (1). We note that each component is able to increase performance. We obtain the largest improvement by reasoning about the relationship between points in 3D and pixels in the image.

Label Uncertainty. Here, we leverage our model's awareness of label uncertainty to demonstrate that higher accuracy can be achieved in confident regions. To quantify uncertainty, we measure the entropy of the label marginal distribution at every pixel, see Fig. 5 (last row). Sorting all pixels according to their entropy allows us to predict the most certain regions in the image. Table 3 (bottom) shows our results on static objects when predicting only those parts of the image. Note how this helps to boost our performance to 94.3% mIoU and 98.4% accuracy when predicting at 70% pixel density, demonstrating that our uncertainty

8. The results differ slightly from those presented in [107] as 1) we updated the ground truth labels to be consistent with the extended label definition and 2) we measure mIoU following Cityscapes [25] while [107] reports weighted mIoU.

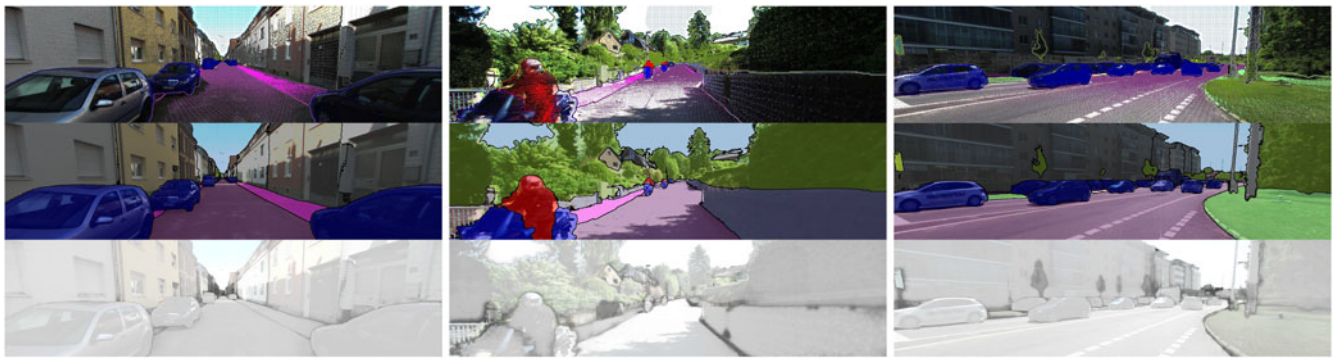


Fig. 5. *Qualitative results on semantic instance segmentation transfer.* Each subfigure shows from top-to-bottom: the input image with the projected 3D points and inferred semantic segmentation boundaries, the inferred semantic instance segmentation, as well as the confidence map of the inferred label with bright and dark colors indicating high and low confidence, respectively. See supplementary material and text for details. The first scene (1st column) contains only static objects while the others (2nd and 3rd columns) also contain dynamic objects.

estimates are well calibrated. In contrast, uncertainty is not directly accessible in most baseline models as they are deterministic or rely on MAP estimates. In the benchmarks introduced in Section 6 where our inferred labels are considered as pseudo-ground truth, we adopt confidence weighted evaluation metrics leveraging the uncertainty to take into account the ambiguity in our automatically generated annotations.

5.2.2 Instance Segmentation Transfer

As time consistent 2D instance ground truth is hard to obtain, most existing 2D label transfer methods focus on the semantic segmentation problem. Therefore, we chose to evaluate instance segmentation performance in an ablation study. We annotated the classes "Building", "Car", "Trailer", "Caravan" and "Box" with instances in our 2D ground truth.[9] For evaluation, we exploit the mIoU metric defined on instances following [107]. Specifically, we first match the ground truth instances to the predicted instances. A pixel is then classified as true positive only when its predicted instance index matches the ground truth. Table 3 (right) shows our results. Note how the instance segmentation results are on par with the semantic segmentation, demonstrating our model's intra-class separation ability. Moreover, we also observe higher instance segmentation accuracy when filtering uncertain predictions.

5.3 Qualitative Evaluation

Fig. 5 illustrates our dense inference results qualitatively for 3 different scenes in terms of semantic instance segmentation on both static and dynamic objects. The first two rows illustrate both semantic and instance labels, where semantic information is color-coded and instances are separated by boundaries. The last row shows the confidence maps. While the proposed method is able to delineate most object boundaries satisfyingly, some challenges remain. Errors occur in regions where 3D points are absent due to far distance (1st & 3rd scene: far building). Another source of errors is inherent label ambiguities that occur for porous objects such as fences or trees (3rd scene: tree boundary) where even 2D ground truth annotation is a hard and ambiguous task. Finally, 3D points of dynamic objects are accumulated over multiple frames (2nd & 3rd scene), providing dense but less accurate 3D cues to the CRF model. However, note that our probabilistic inference algorithm is able to successfully identify those uncertain regions as demonstrated in the last row, where far buildings and object boundaries are predicted as less certain compared to other image regions.

6 Dataset & Benchmarks

We apply the proposed label transfer method to all frames captured by perspective cameras, resulting in 2×78k 2D semantic/instance segmentation maps, 1.0B 3D semantic points and 172.4M 3D instance points. We provide a statistical analysis of the 2D & 3D labels in Appendix F.1, available in the online supplemental material. We further deploy an online evaluation server and establish benchmarks on a set of challenging tasks relevant to autonomous driving. For all tasks, we split the data at the batch-level into disjoint training, validation and held-out test sets as specified in Appendix F.2, available in the online supplemental material. Specifically, we leverage KITTI-360 to address tasks at the intersection of vision, graphics and robotics which are commonly viewed as relevant towards achieving full autonomy, including tasks within the scope of semantic scene understanding, novel view synthesis and semantic SLAM. We now describe each task and the corresponding evaluation protocol in detail. Furthermore, we introduce initial baselines for each task.

6.1 Semantic Scene Understanding

In this section, we establish scene perception benchmarks in both 2D image space and 3D domain. We first implement benchmarks for the traditional tasks of 2D semantic segmentation and 2D instance segmentation on perspective images, using the inferred semantic/instance segmentation maps as pseudo ground-truth. While not the main focus of this work, we establish these standard 2D benchmarks to investigate whether there is a performance gap between methods operating in 2D and 3D. Furthermore, as our label definition is compatible with Cityscapes, this benchmark

9. While [107] uses two sets of parameters for semantic and instance segmentation, we train a single model for instance segmentation and read semantic labels directly from the instance maps. Therefore, our predictions on classes without instance labels are the same in both semantic and instance segmentation maps.

TABLE 4
Quantitative Results for 2D & 3D Scene Understanding on Various Different Tasks

Method	$\text{mIoU}_{\text{class}}$	$\text{mIoU}_{\text{category}}$
FCN [61]	54.0	77.6
PSPNet [116]	**64.9**	**82.2**

(a) 2D Semantic Segmentation

Method	Backbone	AP	AP_{50}
Mask R-CNN [40]	Res. 50	19.5	36.3
	Res. 101	**20.9**	**40.1**

(b) 2D Instance Segmentation

Method	AP_{50}	AP_{25}
BoxNet [76]	**4.1**	23.6
VoteNet [76]	3.4	**30.6**

(c) 3D Bounding Box Detection

Method	$\text{mIoU}_{\text{class}}$	$\text{mIoU}_{\text{category}}$
PointNet [77]	13.1	30.4
PointNet++ [78]	**35.7**	**58.3**

(d) 3D Semantic Segmentation

Method	AP	AP_{50}
PointNet++ [78]+ [30]	23.7	40.1
PointGroup [49]	**34.8**	**53.6**

(e) 3D Instance Segmentation

Method	Acc / Cmp / F_1	$\text{mIoU}_{\text{class}}$
Raw Input	**98.2** / 19.1 / 32.4	–
Enc-Dec	41.4 / **41.2** / **41.3**	9.1

(f) Semantic Scene Completion

opens up the possibility for studying domain adaption across datasets in future work. Next, we establish benchmarks in the 3D domain, including bounding box detection and semantic/instance segmentation. Moreover, we consider a semantic scene completion task where the goal is to simultaneously complete the scene and infer corresponding semantic labels given limited observations. This task allows autonomous vehicles to hallucinate future possibilities and thus can benefit downstream tasks, e.g., predictive control.

2D Semantic Segmentation. We train and evaluate 2D segmentation baselines on the densely labeled images in KITTI-360. We consider two well-known methods, Fully Convolutional Neural Network (FCN) [61] and Pyramid Scene Parsing Network (PSPNet) [116], as a reference. Following Cityscapes [25], we adopt mean intersection over union (mIoU) at two semantic granularities, i.e., classes and categories, where 19 classes are grouped into 7 coarse-grained categories. To account for label uncertainty, the mIoU is weighted by the confidence of our pseudo-ground truth labels. A formal definition of our metrics and a detailed definition of the classes and categories can be found in Appendix G.1, available in the online supplemental material. Table 4a shows that, unsurprisingly, PSPNet outperforms the naïve FCN on the test set.

2D Instance Segmentation. We use the established Mask R-CNN framework [40] with different backbones as our baselines, see Table 4b. We measure the Average Precision (AP) weighted by the label confidence over 10 thresholds, ranging from 0.5 to 0.95 with a step size of 0.05. The mean AP is then calculated over 7 classes that contain instance labels. We also compare mean AP_{50} given a threshold of 0.5. Both Table 4b and Fig. 6 suggest that Mask R-CNN with a deeper backbone leads to better performance. Note that we provide instance segmentation labels of "Buildings" which are not available for other outdoor datasets [18], [25], [36], [45]. This information allows future works to explore scene compositionality [57], [72], [74] in real-world street scenes.

3D Bounding Box Detection. In this benchmark we measure the mean AP over two classes, "Building" and "Car", since it is particularly challenging for learning-based algorithms to generalize well to other classes with fewer training samples. Following [76], the mean AP is calculated at a threshold of 0.25 and 0.5, respectively. We consider VoteNet [76] and its simplified version BoxNet [76] as baseline methods. Both methods require 3D point locations as input and output 3D bounding boxes and their semantic labels. Table 4c suggests that VoteNet can make reasonable predictions for both building and cars while it fails to predict 3D bounding boxes with high IoU values, see Fig. 7f.

3D Semantic Segmentation. We establish a 3D semantic segmentation benchmark on the accumulated point clouds, where PointNet [77] and PointNet++ [78] are trained and evaluated as baselines. Both methods take as input point

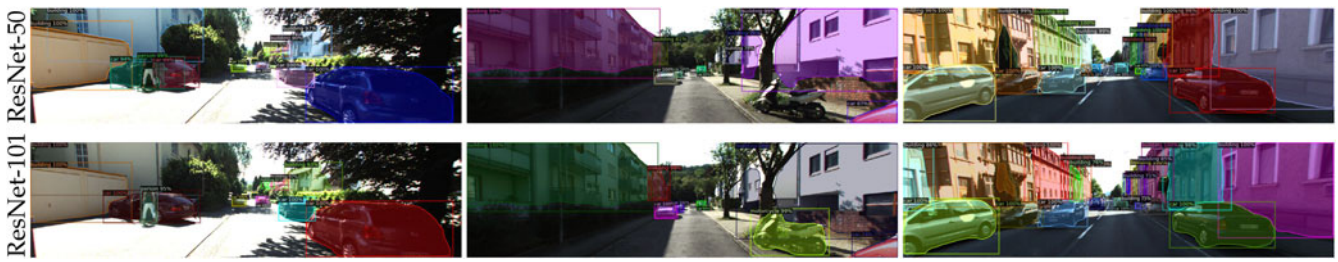


Fig. 6. *Qualitative results for 2D instance segmentation.* The first row shows inference results of Mask-RCNN with a ResNet-50 backbone while the second row uses a ResNet-101 backbone. The ResNet-101 backbone leads to better results, e.g., with the ResNet-50 backbone, the car occluded by the person is split into two instances (left) and the motorcycle is not detected (middle). Note that both variants are able to predict "Building" instances after being trained on KITTI-360.

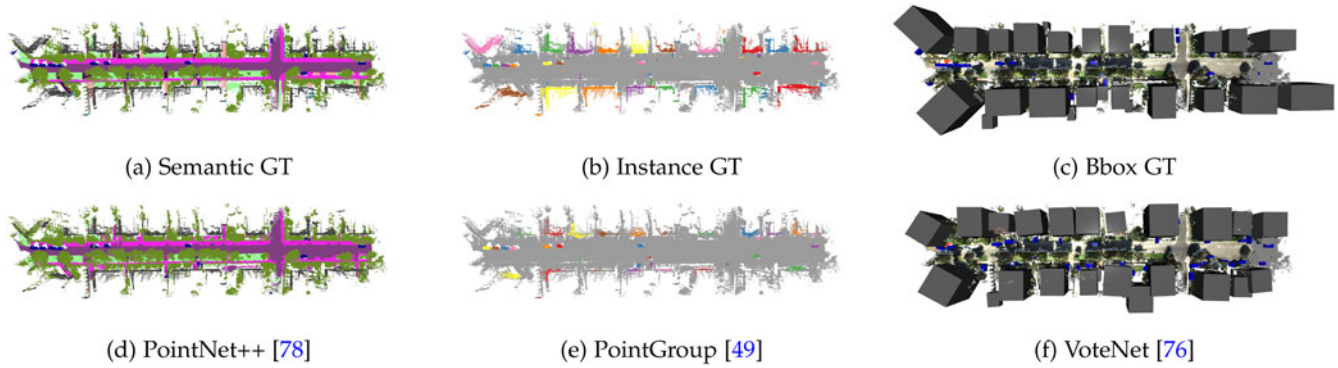


Fig. 7. *Qualitative results for 3D scene perception.* We establish benchmarks for 3D semantic/instance segmentation and 3D bounding box detection. This figure shows the ground truth and the prediction of a baseline method for each sub-task.

locations and colors to predict a semantic label for each 3D point. Following the 2D semantic segmentation task, we measure mIoU weighted by label confidence over classes and categories, respectively. Table 4d shows the quantitative comparison and Fig. 7d illustrates the performance of PointNet++. Interestingly, comparing Tables 4a and 4d shows that the 3D semantic segmentation baselines' overall performances are inferior compared to the 2D semantic segmentation methods, suggesting that parsing the semantic meaning of irregularly structured 3D point clouds remains more challenging and requires further work.

3D Instance Segmentation. We evaluate 3D instance segmentation results for "Building" and "Car". Specifically, we measure the mean AP over a set of thresholds ranging from 0.5 to 0.95 with a step size of 0.05 and AP at a threshold of 0.25 and 0.5. As a first simple baseline, we naïvely cluster semantically labeled points into instances. We use PointNet ++ [78] for semantic segmentation and DBSCAN [30] for clustering. We further evaluate PointGroup [49] as a state-of-the-art method for 3D instance segmentation which takes as input point locations and colors. Table 4e demonstrates that PointGroup outperforms the naïve clustering-based method. The qualitative result of PointGroup is shown in Fig. 7e. While the 2D and 3D results in Tables 4b and 4e are defined over different sets of classes, we provide detailed results on each class in Appendix G.5, available in the online supplemental material. Interestingly, 3D instance segmentation methods achieve better performance on "Car" than 2D methods while performing worse on "Building". We hypothesize that unlike in 2D where occlusions strongly impact the results (e.g., Fig. 6 left, pedestrian standing in front of a car), cars can be more easily separated in 3D. As for buildings, many instances are spatially connected (e.g., Fig. 6 right), making the instance segmentation task harder in 3D where boundaries are harder to detect on the sparse point cloud.

Semantic Scene Completion. While standard scene perception tasks aim to predict a semantic label for each observed scene point, the semantic scene completion task additionally requires predicting geometry and semantics in unobserved regions. Given a single LiDAR scan as input, this task requires semantic scene completion within a corridor of 30m around the vehicle poses of a 100m trajectory. For evaluation, we measure reconstruction quality and semantic prediction accuracy. The former measures geometric accuracy independent of semantics, using *completeness* and *accuracy* over a range of distance thresholds following common practice [89]. We consider a threshold of 20cm as the main metric. As our ground truth reconstruction may not be complete, we evaluate accuracy only in observed regions. We further measure the F_1 score as the harmonic mean of the completeness and the accuracy. The semantic prediction quality is conditioned on the geometric reconstruction. Specifically, we measure the confidence weighted *mIoU* over the same set of thresholds where a true positive prediction is made when 1) a ground truth point is classified as complete at the given threshold and 2) its closest reconstructed point has the correct label. See Appendix G.6, available in the online supplemental material, for more details of the ground truth construction and evaluation metrics.

We consider two baselines for this task, both taking a single raw LiDAR frame as input. For calibration, we implement a naïve baseline which returns the input as output. The second baseline is a learning-based approach where we use an encoder-decoder architecture to predict the complete scene structure from the raw LiDAR scan. More details about this baseline can be found in Appendix G.6, available in the online supplemental material. Table 4f and Fig. 8 illustrate the results. As expected, the raw LiDAR scans are accurate but incomplete. The learning-based approach instead achieves higher completeness but the predictions are less accurate. For the learning-based approach we also predict a semantic label at each 3D point. Fig. 8 shows that the model is able to correctly predict semantic labels at a coarse level but struggles to predict smaller objects like cars.

Discussion. Our results show that 3D semantic segmentation is harder than 2D semantic segmentation. In contrast, our conclusions for instance segmentation vary for different classes. Some classes, e.g., cars, are easier to segment in 3D, suggesting further works can explore 3D information to enhance 2D instance segmentation. 3D bounding box detection remains challenging, especially when a high IoU is desired. Lastly, while inferring dense geometry and semantics from raw sparse observations can benefit autonomous driving, completing the scene and predicting semantics jointly is a difficult task that requires further research.

6.2 Novel View Synthesis

Simulation is an essential tool for training and evaluating autonomous vehicles. While existing methods trained in simulated scenes struggle to generalize to real scenes,

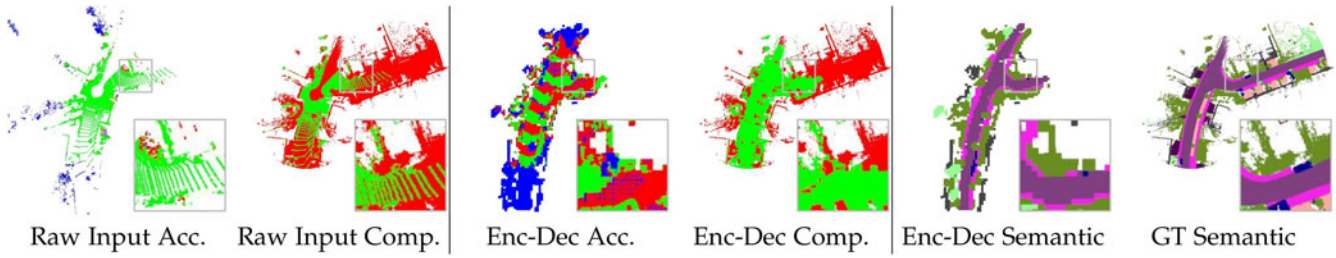


Fig. 8. *Qualitative results for semantic scene completion* evaluated at a distance threshold of 20cm. Green denotes complete/accurate, red denotes incomplete/inaccurate and blue denotes points in unobserved region.

TABLE 5
Quantitative Results for Novel View Appearance & Novel View Semantic Synthesis Using a 50% Drop Rate

Method	Input	PSNR	SSIM	LPIPS	$\text{mIoU}_{\text{class}}$	$\text{mIoU}_{\text{category}}$
GT Image	–	–	–	–	**72.0**	**83.6**
PCL	PC	12.81	0.576	0.549	39.4	46.8
NeRF [64]	PI	21.18	0.779	0.343	53.0	73.9
mip-NeRF [7]	PI	**21.54**	0.778	0.365	51.2	72.0
DS-NeRF [27]	PI	21.28	0.777	0.347	54.8	75.5
FVS [83]	PI	20.00	0.790	**0.193**	67.1	78.5
PBNR [54]	PI	19.91	**0.811**	**0.191**	65.1	77.8

Input: "PC" denotes the accumulated point cloud and "PI" means perspective images.

creating a simulation environment based on real-world images is a promising direction to close the gap between real-world scenarios and synthetic environments [24], [110]. We thus establish challenging benchmarks towards this goal, including novel view appearance synthesis and novel view semantic synthesis.

Novel View Appearance Synthesis. In this benchmark, we are interested in novel view RGB image synthesis for driving scenarios. While we evaluate on a set of held-out perspective images, the benchmark participant can choose from a set of input modalities,[10] including posed perspective/fisheye images or accumulated point clouds. For perspective and fisheye images, we release approximately 50% of the frames for training and use the remaining 50% for testing. In addition, the evaluation server also provides a harder setting with a 90% drop rate. See appendix, available in the online supplemental material, for details. The point cloud is accumulated over all frames where each point fuses colors from different viewpoints. We adopt three standard evaluation metrics for this benchmark: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and perceptual metric (LPIPS) [115].

We evaluate two sets of baselines on two different input modalities. We first consider a naïve baseline using the accumulated point cloud (PCL) as input. Specifically, we project non-occluded colored points to test viewpoints, followed by nearest neighbor interpolation to fill in the missing values. As there are no 3D points in the sky, we heuristically assign a mean blue color to the sky region. We further consider several state-of-the-art baselines for image-based novel view synthesis, including methods based on Neural Radiance Fields [7], [27], [64] or per-view depth maps [54], [83]. Results in Table 5 (left) and Fig. 9 (1st row) reveal the challenges of this benchmark. We observe that NeRF [64] shows promising results but struggles to synthesize fine structure. FVS performs better in rendering fine details (e.g., license plate) but exhibits noticeable artifacts due to the inaccurate underlying geometry (e.g., left car). Interestingly, FVS/PBNR performs better in LPIPS but has a lower PSNR compared to NeRF-based methods, suggesting that LPIPS is more sensitive to fine detail than larger regional errors.

Novel View Semantic Synthesis. An important property of simulation environments like CARLA [29] is that they provide not only RGB images but also auxiliary information like semantic label maps. Towards a real-world simulator with the same capability, we therefore consider a novel benchmark that requires joint novel view and semantic synthesis. The input data for this task is the same as for the novel view synthesis task, while the methods are tasked to predict both an RGB image and a semantic segmentation map at a given target camera pose. Therefore, the evaluation metric of this task additionally comprises mIoU for semantic segmentation as shown in Table 5 (right). As no prior work has addressed this problem yet, we consider a naïve two-stage solution as baseline to bootstrap this benchmark, i.e., we apply an existing semantic segmentation model (PSPNet [116]) on the synthesized images. For comparison, we also evaluate the semantic segmentation performance on the original ground truth images (GT Image). Note that the artifacts in the synthesized images lead to a significant performance drop for semantic segmentation. As illustrated in Fig. 9, the fence is misclassified as building when the synthesized images are taken as input to PSPNet, despite that the fence is still visible in these images. It is also interesting to note that semantic segmentation performance is aligned with the LPIPS metric, as both apply pre-trained networks on synthesized images.

Discussion. Our baselines reveal the different challenges in novel view appearance synthesis with different input modalities. While point clouds provide a good representation of 3D

10. The used input modalities will be indicated on the leaderboard.

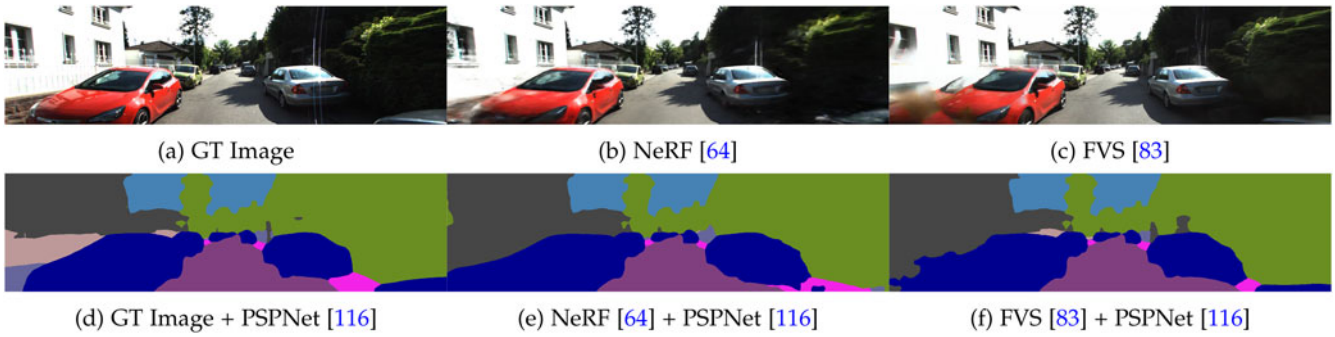


Fig. 9. *Qualitative results for novel view appearance & semantic synthesis.* The first row shows the GT image and novel view appearance synthesis results. The second row shows the corresponding semantic segmentation using PSPNet [116].

geometry, it is not easy to model view dependency. When instead taking a sparse set of multi-view images as input, the task is similarly difficult despite little variation in the camera orientation. We believe that future works should explore the combination of different input modalities to improve image fidelity further. Moreover, given the low performance of our simple baselines on the novel view semantic synthesis task, there is a large potential for future improvements, i.e., by learning view and semantic synthesis jointly.

6.3 Semantic SLAM

We further establish a semantic SLAM benchmark at the intersection of robotics and computer vision. Here, the goal is to simultaneously estimate poses and reconstruct a *semantic* map from monocular/stereo images and/or LiDAR scans. While there is a growing interest in evaluating indoor semantic reconstructions of SLAM algorithms at room-level [87], [102], existing works on outdoor semantic SLAM typically evaluate only pose estimation while ignoring the quality of the semantic reconstruction [12], [23]. Considering that the semantic reconstruction is valuable on its own for down-stream tasks, e.g., planning [28], we thus additionally evaluate geometric and semantic reconstructions where the latter is enabled by the dense semantic annotations of KITTI-360. For this benchmark, the test sequences are separated from those used for 3D scene perception such that the accumulated point cloud is held out from the public.

Localization. Given an estimated trajectory, we adopt the standard Absolute Pose Error (APE) and Relative Pose Error (RPE) [37] as metrics for evaluating pose estimation. We consider four test sequences for this task and report the evaluation results on each test sequence without averaging.

We evaluate two baseline methods, ORB-SLAM2 [66] and SUMA++ [23], where the former takes stereo images as input and the latter is applied on LiDAR scans. Table 6a compares the localization results of ORB-SLAM2 and SUMA++. For both methods, the APE exceeds 2 meters and the RPE is around 2% in general. ORB-SLAM2 achieves better overall performance compared to SUMA++, suggesting that the stereo images of our dataset contain rich features for the purpose of localization. One possibility for improving localization accuracy is to exploit the 3D bounding boxes and instance labels available in our dataset [71], [109].

Geometric and Semantic Mapping. We measure the quality of the geometric and semantic mapping using the same metrics considered in the semantic scene completion benchmark. As richer input observations are available in this task, we adopt a smaller distance threshold of 10cm as the main metric. Specifically, we first measure geometry accuracy using completeness and accuracy, and then evaluate semantics on the completed ground truth points via the confidence weighted mIoU. As the mapping accuracy is highly correlated with the APE, we compare ground truth and estimated reconstruction in local windows to minimize the impact of pose drifts. Each local window consists of 50 consecutive frames and is aligned to the ground truth based on the trajectory, see Appendix I.2, available in the online supplemental material, for more details.

We use the same baselines considered in the localization benchmark. As ORB-SLAM2 does not provide dense reconstruction nor semantic information, we obtain dense semantic reconstruction by unprojecting 2D semantic segmentations (PSPNet [116]) using depth maps from semi-global matching (SGM) [42]. SUMA++ aims for semantic SLAM and estimates poses and a semantic surfel map from LiDAR scans. We experimentally observe that it is sufficient to take the center of the surfels as the reconstructed points.

Table 6b and Fig. 10 show the reconstruction and semantic prediction results. We observed that both baselines produce good reconstructions on the ground region. For regions above the ground, SUMA++ is less complete as it

TABLE 6
Quantitative Results for Semantic SLAM

Test Seq.	ORB-SLAM2 APE (m)	ORB-SLAM2 RPE (%)	SUMA++ APE (m)	SUMA++ RPE (%)
0	**1.53** $\pm$ 0.74	**2.42** $\pm$ 1.34	2.27 $\pm$ 1.38	2.66 $\pm$ 2.12
1	**2.22** $\pm$ 0.78	2.46 $\pm$ 1.37	2.87 $\pm$ 1.50	**2.43** $\pm$ 1.80
2	**2.12** $\pm$ 0.94	**1.50** $\pm$ 1.01	4.62 $\pm$ 4.24	2.90 $\pm$ 2.90
3	**1.79** $\pm$ 0.96	**1.72** $\pm$ 1.22	2.77 $\pm$ 1.44	2.88 $\pm$ 2.42

(a) Localization. RPE evaluated with a delta unit of 1 meter.

Test Seq.	ORB-SLAM2 + PSPNet Acc./ Comp. / F_1	ORB-SLAM2 + PSPNet mIoU	SUMA++ Acc./ Comp. / F_1	SUMA++ mIoU
0	78.5 / **72.8** / **75.5**	**35.3**	**90.8** / 63.1 / 74.5	19.9
1	81.8 / **76.8** / **79.2**	**31.6**	**89.3** / 62.9 / 73.8	17.3
2	82.5 / **70.8** / **76.2**	**30.4**	**89.6** / 64.5 / 75.0	17.6
3	84.3 / **79.1** / **81.6**	**32.7**	**94.2** / 66.3 / 77.8	22.8

(b) Semantic mapping evaluated at a threshold of 10cm.

Evaluated on four test sequences.

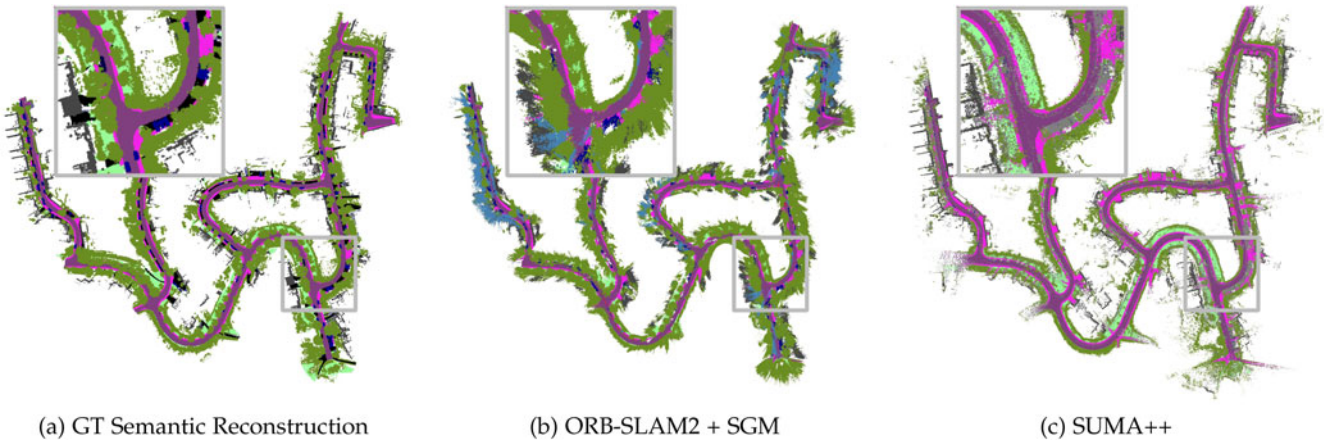


Fig. 10. *Qualitative results for semantic mapping* on test sequence 3 colored based on semantic class labels.

only uses LiDAR scans and thus the maximum height is limited. ORB-SLAM2 + SGM results in higher completeness but worse accuracy. In terms of semantic predictions, SUMA++ produces reasonable results on the LiDAR scans but struggles to achieve good overall performance due to its low completeness. In contrast, ORB-SLAM2 + PSPNet contains more flying points due to the outliers of stereo matching (e.g., sky points colored blue). Exploring semantic information to remove sky points may further improve the performance of this baseline.

Discussion. We evaluate localization accuracy of existing SLAM methods and suggest exploring 3D instance-level information in further works. Further, reconstructing accurate geometry and semantics remains a challenging task. Our benchmark allows to investigate important questions towards solving this challenging task, e.g., which input modality is better suited for this task, whether semantic prediction and geometric reconstruction can benefit each other and if joint optimization is desirable.

7 Conclusion

We present KITTI-360, a large scale 3D video dataset comprising 300k images and laser point clouds with consistent semantics in both 2D and 3D. We create a WebGL-based annotation tool and annotate both static and dynamic objects in 3D. We propose a method to obtain dense semantic instance labels from annotated 3D primitives. In the presence of 3D data, our method yields better results compared to several 2D label transfer baselines while lowering the annotation time.

Furthermore, we establish novel online benchmarks for several challenging tasks at the intersection of computer vision, graphics and robotics. We evaluate several baselines for each benchmark. Our results show that existing methods achieve satisfactory results on well-established benchmarks, e.g., 2D/3D segmentation, where inference is directly performed on given observations. However, it is much harder to solve tasks that require jointly recovering the geometry, appearance and estimating the semantics as in the newly introduced tasks for semantic scene completion, novel view appearance/semantic synthesis and semantic SLAM. We hope that our dataset, online benchmarks and annotation tools will fertilize new research across communities, fostering progress towards the grand goal of full autonomy.

Acknowledgments

The authors would like to thank Siyuan Peng, Bernhard Jaeger, Shrisha Bharadwaj, Apratim Bhattacharyya, Paul Henderson, and Zehao Yu for their help in implementing the baselines, Kashyap Chitta, Katja Schwarz, and Yue Wang for proofreading, and SurfingTech for annotating parts of the dataset.

References

[1] Daimler urban segmentation dataset. [Online]. Available: http://www.6d-vision.com/scene-labeling
[2] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with Polygon-RNN++," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 859–868.
[3] M. Andriluka, J. Uijlings, and V. Ferrari, "Fluid annotation: A human-machine collaboration interface for full image annotation," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1957–1966.
[4] M. Aygun *et al.*, "4D panoptic LiDAR segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5523–5533.
[5] V. Badrinarayanan, I. Budvytis, and R. Cipolla, "Mixture of trees probabilistic graphical model for video segmentation," *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 14–29, 2014.
[6] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3265–3272.
[7] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5835–5844.
[8] J. Behley *et al.*, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9296–9306.
[9] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 4391–4398.
[10] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. Comput. Graphics*, vol. 5, no. 4, pp. 349–359, Fourth Quarter 1999.
[11] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[12] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular SLAM for highly dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 393–400.

[13] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, Jan. 2009.

[14] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1863–1870.

[15] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Label propagation in complex video sequences using semi-supervised learning," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 2258–2259.

[16] I. Budvytis, P. Sauer, T. Roddick, K. Breen, and R. Cipolla, "Large scale labelled video data augmentation for semantic segmentation in driving scenarios," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 230–237.

[17] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," 2020, *arXiv: 2001.10773*.

[18] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 618–11 628.

[19] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4485–4493.

[20] A. Chang *et al.*, "Matterport3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 667–676.

[21] M. Chang *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8740–8749.

[22] L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun, "Beat the MTurkers: Automatic image labeling from weak 3D supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3198–3205.

[23] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, "Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1326–1334.

[24] Y. Chen *et al.*, "GeoSim: Realistic video simulation via geometry-aware composition for self-driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7226–7236.

[25] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

[26] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2432–2443.

[27] K. Deng, A. Liu, J. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," 2021, *arXiv:2107.02791*.

[28] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Trans. Robot. Autom.*, vol. 4, no. 3, pp. 2997–3004, Jul. 2019.

[29] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[30] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.

[31] R. Gadde, V. Jampani, and P. V. Gehler, "Semantic video CNNs through representation warping," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4463–4472.

[32] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4340–4349.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[35] A. Geiger and C. Wang, "Joint 3D object and layout inference from a single RGB-D image," in *Proc. German Conf. Pattern Recognit.*, 2015, pp. 183–195.

[36] J. Geyer *et al.*, "A2D2: Audi autonomous driving dataset," 2020, *arXiv: 2004.06320*.

[37] M. Grupp, "EVO: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

[38] M. Guillaumin, D. Küttel, and V. Ferrari, "ImageNet auto-annotation with segmentation propagation," *Int. J. Comput. Vis.*, vol. 110, no. 3, pp. 328–348, 2014.

[39] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. IV-1-W1, pp. 91–98, 2017.

[40] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[41] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1793–1800.

[42] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[43] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the wild: Pixel-level adversarial and constraint-based adaptation," 2016, *arXiv:1612.02649*.

[44] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," in *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[45] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[46] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 656–671.

[47] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, pp. 1–308, 2020.

[48] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3121–3128.

[49] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4866–4875.

[50] R. Kesten *et al.*, "Level 5 perception dataset 2020," 2019. [Online]. Available: https://level-5.global/level5/data/

[51] M. Kiefel and P. Gehler, "Human pose estimation with fields of parts," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 331–346.

[52] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9856–9865.

[53] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 78.

[54] G. Kopanas, J. Philip, T. Leimkühler, and G. Drettakis, "Point-based neural rendering with per-view optimization," *Comput. Graph. Forum*, vol. 40, no. 4, pp. 29–43, 2021.

[55] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

[56] M. Larsson, E. Stenborg, L. Hammarstrand, M. Pollefeys, T. Sattler, and F. Kahl, "A cross-season correspondence dataset for robust semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9524–9534.

[57] Y. Liao, K. Schwarz, L. Mescheder, and A. Geiger, "Towards unsupervised learning of generative models for 3D controllable image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5870–5879.

[58] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[59] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-GCN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5252–5261.

[60] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[61] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[62] V. Madhavan and T. Darrell, "The BDD-Nexar collective: A large-scale, crowsourced, dataset of driving scenes," Master's thesis, 2017.

[63] A. Martinović, J. Knopp, H. Riemenschneider, and L. Van Gool, "3D all the way: Semantic segmentation of urban scenes from start to end in 3D," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4456–4465.

[64] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[65] D. Munoz, J. A. Bagnell, and M. Hebert, "Co-inference machines for multi-modal scene analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 668–681.

[66] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[67] A. Mustafa and A. Hilton, "Semantically coherent co-segmentation and reconstruction of dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5583–5592.

[68] N. S. Nagaraja, P. Ochs, K. Liu, and T. Brox, "Hierarchy of localized random forests for video annotation," in *Proc. Joint 34th DAGM 36th OAGM Symp. Pattern Recognit.*, 2012, pp. 21–30.

[69] S. T. Namin, M. Najafi, M. Salzmann, and L. Petersson, "A multi-modal graphical model for scene analysis," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2015, pp. 1006–1013.

[70] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5000–5009.

[71] L. Nicholson, M. Milford, and N. Sünderhauf, "QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM," *IEEE Trans. Robot. Autom.*, vol. 4, no. 1, pp. 1–8, Jan. 2019.

[72] M. Niemeyer and A. Geiger, "GIRAFFE: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 448–11 459.

[73] P. Osborne, "The mercator projections," 2008. [Online]. Available: http://mercator.myzen.co.uk/mercator.pdf

[74] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2855–2864.

[75] Q.-H. Pham *et al.*, "A*3D dataset: Towards autonomous driving in challenging environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2267–2273.

[76] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9276–9285.

[77] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[78] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[79] C. R. Qi *et al.*, "Offboard 3D object detection from point cloud sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6130–6140.

[80] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3996–4007.

[81] M. A. Reza, H. Zheng, G. Georgakis, and J. Kosecka, "Label propagation in RGB-D video," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4917–4922.

[82] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 102–118.

[83] G. Riegler and V. Koltun, "Free view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 623–640.

[84] H. Riemenschneider, A. Bódis-Szomorú, J. Weissenberg, and L. V. Gool, "Learning where to classify in multi-view semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 516–532.

[85] L. G. Roberts, "Machine perception of three-dimensional solids," PhD dissertation, Massachusetts Institute of Technology, 1963.

[86] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3234–3243.

[87] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1689–1696.

[88] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A point cloud dataset for urban scene segmentation and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 2108–21083.

[89] T. Schöps *et al.*, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2538–2547.

[90] M. Schönbein and A. Geiger, "Omnidirectional 3D reconstruction in augmented manhattan worlds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 716–723.

[91] M. Schönbein, T. Strauß, and A. Geiger, "Calibrating and centering quasi-central catadioptric cameras," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 4443–4450.

[92] S. M. Seitz and R. Szeliski, "Applications of computer vision to computer graphics," *Comput. Graph.*, vol. 33, no. 4, pp. 35–37, 1999.

[93] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[94] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2443–2451.

[95] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 438–451.

[96] W. Tan *et al.*, "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 797–806.

[97] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7472–7481.

[98] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.

[99] J. P. C. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. S. Torr, "Mesh based semantic modelling for indoor and outdoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2067–2074.

[100] S. Vijayanarasimhan and K. Grauman, "Active frame selection for label propagation in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 496–509.

[101] V. Vineet, G. Sheasby, J. Warrell, and P. H. S. Torr, "PoseField: An efficient mean-field based method for joint estimation of human pose, segmentation, and depth," in *Proc. Int. Workshop Energy Minimization Methods Comput. Vis. Pattern Recognit.*, 2013, pp. 180–194.

[102] J. Wald, K. Tateno, J. Sturm, N. Navab, and F. Tombari, "Real-time fully incremental scene understanding on mobile platforms," *IEEE Trans. Robot. Autom.*, vol. 3, no. 4, pp. 3402–3409, Oct. 2018.

[103] M. Weber *et al.*, "STEP: Segmenting and tracking every pixel," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, 2021.

[104] P. H. Winston, "Heterarchy in the M.I.T. robot," MIT Artificial Intelligence Laboratory, 1971.

[105] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1625–1632.

[106] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 686–693.

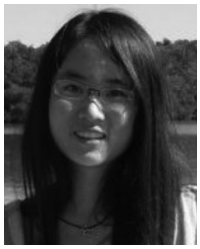
[107] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3D to 2D label transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3688–3697.

[108] Y. Xiong *et al.*, "UPSNet: A unified panoptic segmentation network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8810–8818.

[109] S. Yang and S. A. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[110] Z. Yang *et al.*, "SurfelGAN: Synthesizing realistic sensor data for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 115–11 124.
[111] S. Yogamani *et al.*, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9307–9317.
[112] S. Zakharov, W. Kehl, A. Bhargava, and A. Gaidon, "Autolabeling 3D objects with differentiable rendering of SDF shape priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 221–12 230.
[113] M. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.
[114] H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3056–3063.
[115] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[116] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.
[117] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4141–4150.
[118] Y. Zhu *et al.*, "Improving semantic segmentation via video propagation and label relaxation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8848–8857.
[119] S. M. I. Zolanvari *et al.*, "DublinCity: Annotated LiDAR point cloud and its applications," in *Proc. Brit. Mach. Vis. Conf.*, 2019, Art. no. 44.

Yiyi Liao received the PhD degree from the Department of Control Science and Engineering, Zhejiang University, China, in 2018. She is currently a postdoctoral researcher with the Autonomous Vision Group, University of Tübingen and Max Planck Institute for Intelligent Systems, Germany. Her research interests include 3D vision and scene understanding.



Jun Xie received the PhD degree from the Electrical Engineering Department, University of Washington, in 2016. She is currently a research engineer with Google. Her research interests include image segmentation and 3D vision.

Andreas Geiger received the diploma degree in computer science and the PhD degree from the Karlsruhe Institute of Technology, in 2008 and 2013. Currently, he is leading the Autonomous Vision Group, University of Tübingen and the Max Planck institute for Intelligent Systems in Tübingen. His research interests include computer vision, machine learning, and scene understanding with a focus on self-driving vehicles.