Spotting the Unexpected (STU): A 3D LiDAR Dataset for Anomaly Segmentation in Autonomous Driving

Supplementary Material

6. Hardware Setup

The sensors and hardware included in the data collection platform are as follows:

- 5 SF3325 automotive GMSL cameras (ONSEMI CMOS image sensor AR0231), SEKONIX ultra high-resolution lens with 60 horizontal and 38 vertical FOV, images captured at a resolution of 1928 $\times$ 1208 (2.3M pixel) at 30 Hz.
- 3 SF3324 automotive GMSL cameras, (ONSEMI CMOS image sensor AR0231), SEKONIX ultra high-resolution lens with 120 horizontal and 73 vertical FOV, images captured at a resolution of 1928 $\times$ 1208 (2.3M pixel) at 30 Hz.
- 1 OS1-128 Ouster Lidar, with a vertical resolution of 128 beams within a 45 FOV and range of 200 meters, point cloud captured at 10 Hz.
- 1 NVIDIA DRIVE Pegasus, with two NVIDIA Xavier™ SoCs.

The placement and reference frames of the sensors on the vehicle are shown in Figure 8. The arrangement of the cameras and LiDAR sensors allows the vehicle to achieve a full 360° field of view (FOV) of its surroundings.

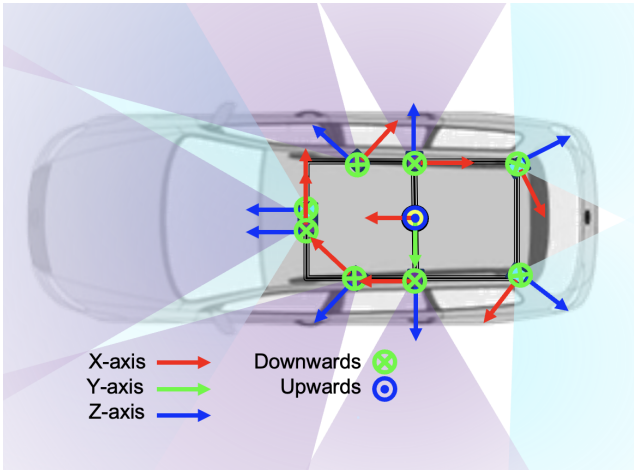


Figure 8. Sensor setup of the data collection vehicle. The field of view for the 60-degree and 120-degree cameras is represented in purple and blue, respectively.

6.1. Extrinsic Calibration

The camera positions on the vehicle were determined through a LiDAR-camera calibration process in which we computed the homogeneous transformation matrices from the LiDAR to each camera. Without ground truth for these transformations, their accuracy is typically validated visually by examining the correspondence between objects in the camera images and the LiDAR point cloud. In this case, Figure 9 shows the alignment between the LiDAR point cloud and an image captured by the front camera, illustrating the accuracy of the calibration process.

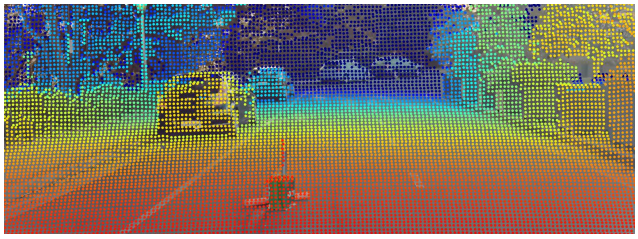


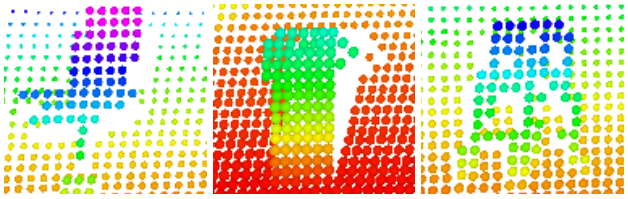
Figure 9. Lidar-Camera projection. The point cloud is colored by distance to the camera.

6.2. Software

A Robotic Operation System (ROS) framework manages the LiDAR point-cloud acquisition pipeline. Camera data is captured using the NVIDIA DRIVE Pegasus video capture and image compression pipeline. All images are encoded and stored as H.264 video, with associated metadata stored in a custom ROS message.

7. Data Collection

For staged data collection, we used a diverse collection of objects, including buckets, indoor garbage bins, brooms, chairs, pots, stuffed animals, balloons, balls, backpacks, bags, pillows, shoes, umbrellas, hats, yoga mats, helmets, swimming noodles, tissue boxes, ladders, car seats, sleeping bags, and bottles. The point cloud captures the three-dimensional structures of objects at varying distances. Fig-



(a) Chair. **(b)** Garbage bin. **(c)** Ladder.

Figure 10. Point cloud of some objects on the road colored by height.

ure 10 shows the point clouds of a chair, an indoor garbage can and a ladder, each color-coded according to height to highlight their spatial dimensions.

7.1. Postprocessing

After data collection, the raw information is post-processed to estimate vehicle poses and to anonymize the image data. Point cloud registration was performed using KISS ICP [57], a lidar odometry pipeline. It includes point cloud motion compensation, subsampling, adaptive thresholding to determine correspondences, and lidar pose estimation. The calculated LiDAR pose was then exported in SemanticKITTI format as required for the labeling tool [1].

Ethical considerations for this dataset include anonymization of camera images to protect individual privacy. Identifiable information, such as faces and license plates, is processed using DashcamCleaner [52] and DeepPrivacy2 [29]. DashcamCleaner uses a license plate detector to locate and blur license plates, while DeepPrivacy2 identifies facial features and generates new unidentifiable photorealistic faces to replace the original ones. Figure 11 illustrates the anonymization process using a publicly available image from the internet.

(a) Original image (source: https://tinyurl.com/3s66za36)

(b) Anonymized image.

Figure 11. Anonymization of camera images.

7.2. Ground Plane Segmentation

One of the popular approaches for anomaly detection in the point-cloud domain involves applying ground-plane removal algorithms to reduce the search space. We used Patchwork++ [35] to remove the ground plane from the point-cloud data. While ground plane removal is effective at short ranges, its performance is reduced at longer distances and on roads with varying geometries. Under these conditions, Patchwork++ often results in many false positives or incorrectly segments objects as part of the ground plane. In addition, manually fine-tuning the parameters of such methods to adapt to different topographies is a very challenging task.

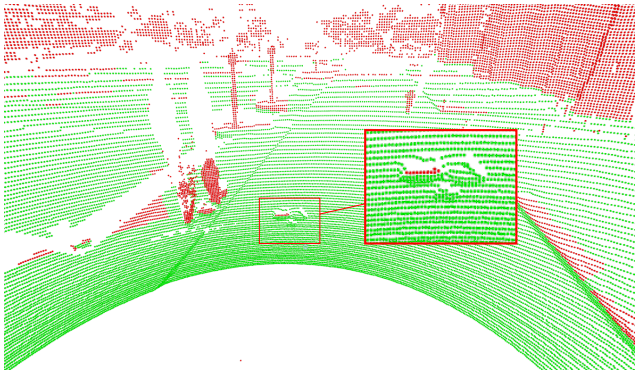


Figure 12. Patchwork++ performance in a wide environment.

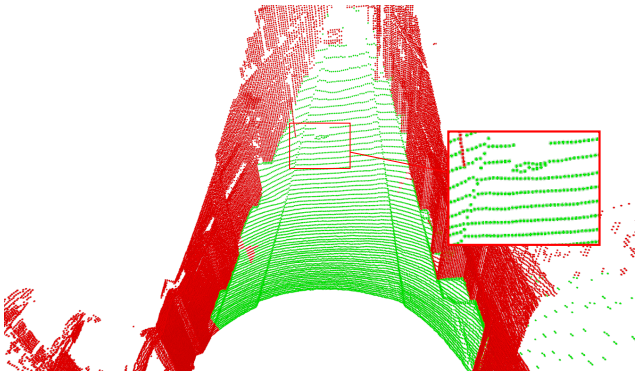


Figure 13. Patchwork++ performance in a narrow urban street.

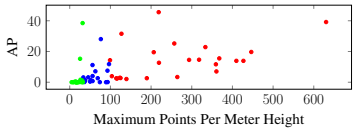
8. Low Performance of the 3D Models

8.1. Relation of Performance to Distance and Size

We calculated the AP metric for different distance thresholds, as shown in Table 4. In the lower ranges, from 0 to 10, and from 10 to 20 range models perform better, then at other distances. Note, that methods evaluate on points within this range, treating points outside of the range as unlabeled, *i.e.* evaluation on 10–20 meters means that points at least as far as 10 meters are considered for evaluation. Expectedly, we see a decrease in performance as the distance to the anomaly increases. In addition, we looked at

Size	Ens AP
● Large	14.35
● Medium	4.30
● Small	2.50

(a) Mean AP for the Category.



(b) Mean AP Per Sequence.

Figure 14. Deep Ensembles AP for differently sized objects over validation and test datasets.

Method	0–10m	10–20m	20–30m	30–40m	40–50m
Deep Ensemble [34]	7.63	8.49	3.42	0.38	0.03
MC Dropout [50]	0.16	0.53	0.06	0.04	0.01
Max Logit [24]	2.25	1.53	1.20	0.27	0.01
Void Classifier [4]	2.95	1.78	1.98	0.28	0.03
RbA [41]	1.85	1.28	0.73	0.15	0.01

Table 4. Anomaly segmentation performance per distance measured by AP.

Method	Aux Data	AUROC↑	FPR@95 ↓	AP↑
DenseHybrid [21]	✗	87.09	76.36	26.63
RbA [41]	✗	89.58	75.21	37.12
UNO [16]	✓	89.52	62.29	37.10
Mask2Anomaly [47]	✓	90.54	78.09	36.38

Table 5. Evaluation of 2D methods on our the validation set using only a front-view camera.

the relation between the size of the object and anomaly segmentation performance in a sequence. We observe a drop in performance for small objects in Figure 14b.

8.2. Number of Foreground Points

The class imbalance remains a challenge for Point-Level evaluation, as it is more pronounced in terms of the occupied space and number of points. If we compare to a SegmentMeIfYouCan setup (see Table 1 from [10]), our dataset has 0.03% of anomaly and 36.9% inlier points; that is twice as few anomaly points. In addition, we evaluate objects with at least 5 anomaly points (instead of 50 or more [10]). However, for some sequences, we observe performance similar to 2D methods, especially for large objects. On Figure 14a we show performance of the Ensemble method on a combined validation and test dataset for better illustration. Here, we split our data into sequences base on the effective size of an object. We divide the maximum number of points for an instance by the maximum height of an instance in a sequence, and separate sequences into three categories: sequence with an object that has $0-33$ points per meter, $33-99$ and $999+$ points per meter. We observe that deep ensembles perform better on sequences with larger objects.

9. Results on Validation Datasets

We show results for the SemanticKITTI [1] validation set in Table 6 and our dataset in Table 7. For the OOD validation set, we evaluate in three sequences and provide scores in Table 8.

9.1. Evaluation of 2D Methods

We focus on automotive applications, where it is common practice to evaluate multimodal methods on 3D LiDAR annotations, since 3D distances to objects are important for driving. As a control experiment, we applied 2D-only methods to the frontal camera and evaluated only within the corresponding frustum of LiDAR points (see Table 5). Methods that use only RGB images have a higher false-positive rate in this setup compared to 2D benchmarks. We attribute this to a domain gap of the Cityscapes and SemanticKitti, partially because of label conventions, *i.e.*, parking lots or backs of traffic signs are "unlabeled" in Cityscapes but are "inlier" in SemanticKitti and these regions contribute to higher FPR. As well as to the LiDAR-Camera points misalignment, *i.e.* at image boundaries. However, in our dataset, anomalies appear in images from cameras with other perspectives, and evaluating a full 360-degree view would be more difficult.

10. Annotation and Qualitative Examples

We visualize the annotation interface with an example of a correctly annotated scene in the figure 15. Anomaly points are cyan, unlabeled regions are black, and inliers are purple. We provide further visualizations of the dataset and the predictions shown in the Figure 16.

11. SemanticKITTI Other-object Examples

Several examples of the other-object class in the SemanticKITTI dataset can be seen in Figure 17, Figure 18, and Figure 19. The other-object class consists of many miscellaneous items, including trash bins, advertisement posts, and small pots. We superimposed the class label on the front-facing camera, with light blue denoting the other object class.

12. Note on Training

Initially, jointly training with both SemanticKITTI and Panoptic-CUDAL led to diverging losses for Mask4Former-3D. This also occurred during training runs solely on Panoptic-CUDAL. Lowering the preset learning rate from 0.0004 to 0.0002 was enough to mitigate the loss divergence in both cases.

Table 6. Class-wise PQ scores on SemanticKITTI validation set.

Method	void	car	truck	bicycle	motorcycle	other vehicle	person	bicyclist	motorcyclist	road	sidewalk	parking	other ground	building	vegetation	trunk	terrain	fence	pole	traffic sign	PQ
Mask-PLS* [40]	–	94.12	83.19	44.55	61.44	59.36	77.08	91.64	0.0	93.94	78.67	37.19	0.0	86.99	84.85	53.28	57.06	21.35	59.85	53.62	59.9
Mask-PLS [40]	–	91.90	77.93	16.96	51.28	45.66	65.85	83.36	0.0	93.86	77.69	31.39	0.0	86.97	87.61	50.40	59.10	22.79	60.77	53.34	55.62
Mask4Former-3D* [60]	–	93.81	70.95	62.92	68.97	56.79	81.98	87.35	24.06	93.94	78.05	27.33	0.0	88.39	88.65	50.93	60.82	25.38	57.89	58.59	61.94
Mask4Former-3D	–	93.53	59.39	62.55	64.82	54.36	79.61	89.16	25.01	93.24	77.90	28.79	0.0	87.27	87.28	51.08	59.92	24.85	56.76	58.14	60.72
Mask4Former-3D-void	6.08	74.36	47.00	32.19	43.34	33.30	42.90	68.75	00.33	93.35	77.07	19.01	0.0	82.77	81.34	47.56	56.94	19.98	54.48	36.82	47.97

Table 7. Class-wise PQ scores on STU-inlier Validation Set.

Method	void	car	truck	bicycle	person	road	sidewalk	parking	building	vegetation	trunk	terrain	fence	pole	traffic sign	PQ
Mask-PLS* [40]	–	78.88	1.89	0.0	70.92	56.34	26.18	0.0	54.96	74.69	1.36	55.20	48.00	41.51	44.51	39.60
Mask-PLS	–	78.66	22.16	0.0	75.33	81.77	41.46	0.0	78.79	89.16	49.92	25.53	46.53	56.79	66.94	50.93
Mask4Former-3D* [60]	–	78.45	11.29	10.59	69.44	59.07	41.99	0.27	84.70	88.46	0.0	0.0	36.51	25.92	57.50	42.80
Mask4Former-3D	–	80.99	37.28	47.65	80.99	71.46	17.74	0.0	84.08	89.73	29.34	30.79	47.6	59.62	60.96	52.73
Mask4Former-3D-void	0.07	23.88	20.78	1.01	43.30	38.24	20.03	11.11	48.45	43.09	20.20	17.31	30.80	27.26	33.16	26.96

Table 8. Anomaly Segmentation Performance on the Validation Set with Anomalies

Method	Point-Level OOD			Object-Level OOD				
	AUROC ↑	FPR@95 ↓	AP ↑	RecallQ	SQ	RQ	UQ	PQ
Deep Ensemble [34]	90.93	37.34	6.94	17.70	79.96	9.10	14.15	7.27
MC Dropout [50]	65.76	79.82	0.17	3.54	74.36	3.48	2.63	2.59
Max Logit [24]	87.27	68.76	2.02	26.64	79.26	2.06	21.12	1.63
Void Classifier [4]	89.77	79.50	2.62	17.35	81.27	8.98	14.10	7.30
RbA [41]	73.00	100.0	1.64	21.84	78.58	2.75	17.16	2.16

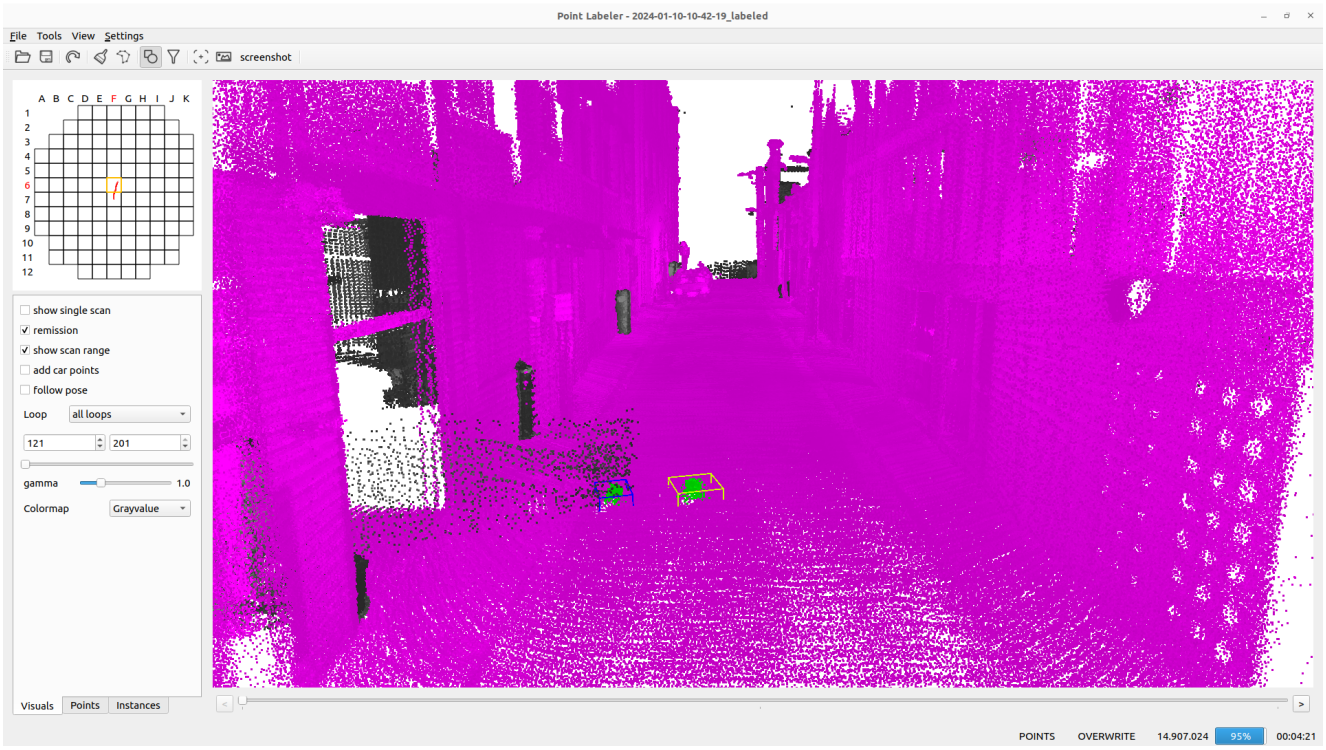
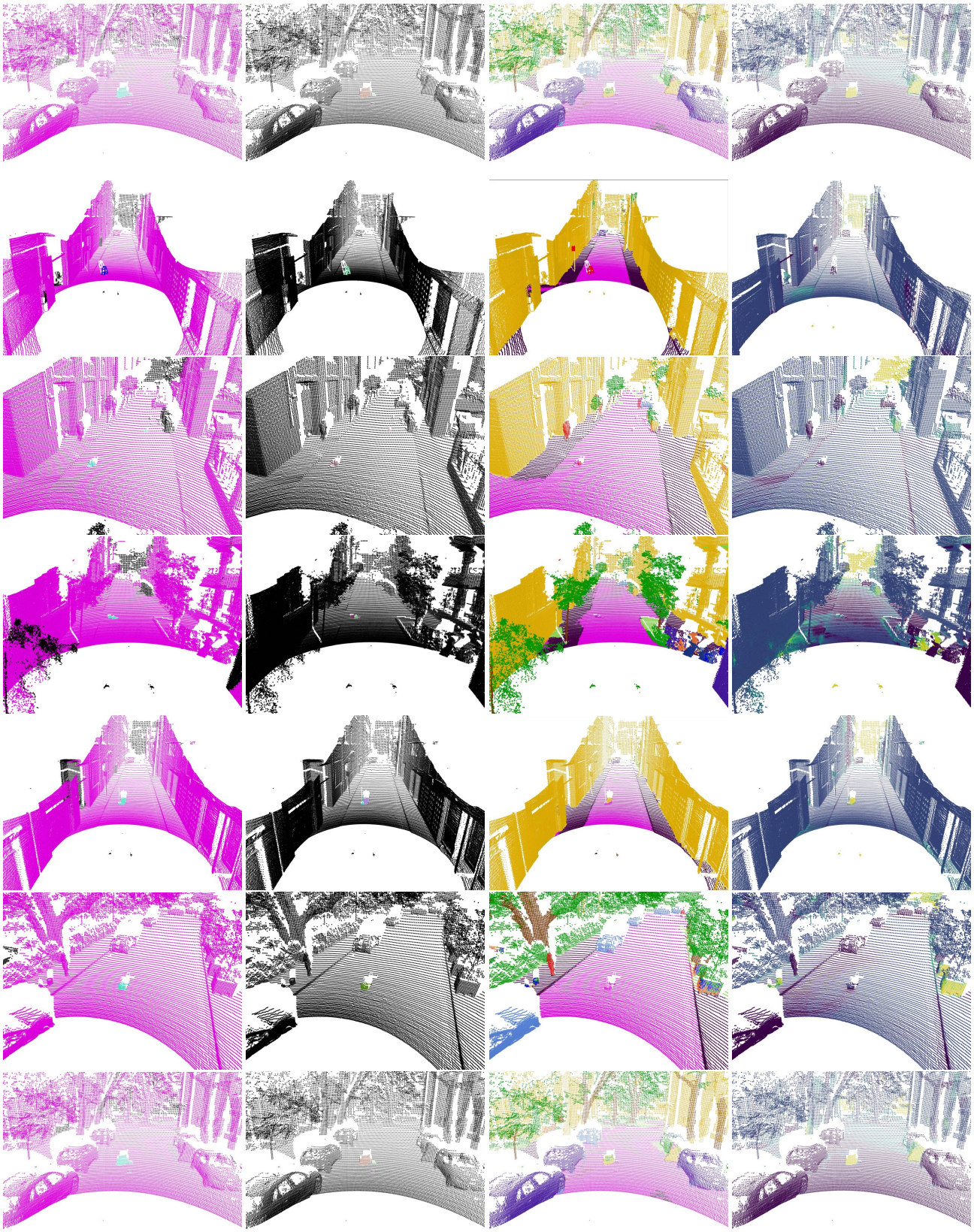


Figure 15. Data Annotation Example: Each color represents a specific label — Purple for inlier, Green for anomaly, and Black for void. Boxes represent instance boundaries.



(a) Anomaly Label. **(b)** Instance Label. **(c)** Inlier Prediction. **(d)** RBA.

Figure 16. Visualization of the proposed dataset with anomaly labels, instance labels, inlier class predictions, and anomaly scores of the selected anomaly methods.

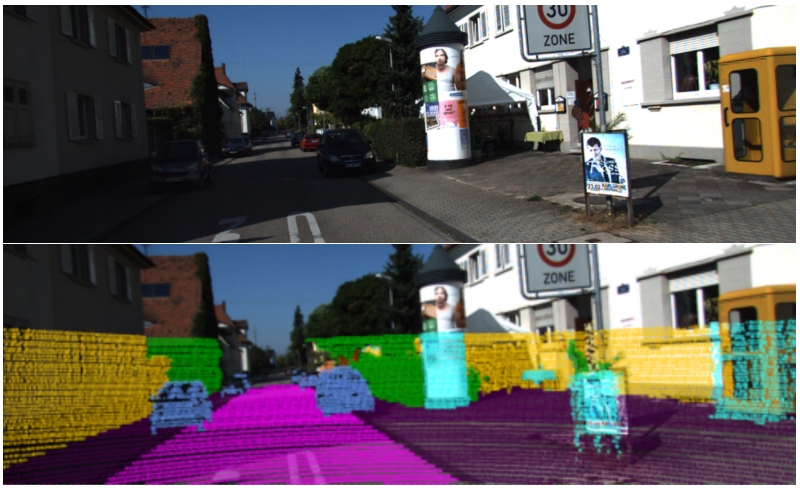


Figure 17. Example of the other-object class: a billboard, a smaller billboard, a phone booth, and a small table, all of which belong to the other-object class.

Figure 18. Example of the other-object class: a car dealership sign and two garbage cans, all belonging to the other-object class.

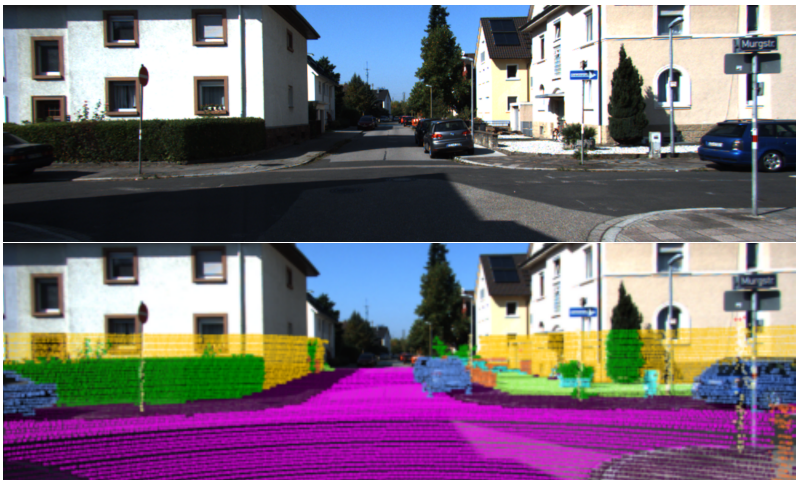


Figure 19. Example of the other-object class: a potted plant and a power adapter, all of which belong to the other-object class.