RoboSense: Large-scale Dataset and Benchmark for Egocentric Robot Perception and Navigation in Crowded and Unstructured Environments

Supplementary Material

A. Coordinates Transformation

A.1. LiDAR⇔Ego-Vehicle

LiDAR to Ego-Vehicle: (x_v, y_v, z_v) represents a three-dimensional coordinate point in Ego-Vehicle Coordinate System. The transformation from the coordinates (x_v, y_v, z_v) in the Ego-Vehicle Coordinate System to (x_l, y_l, z_l) in the LiDAR Coordinate System is calculated as follows:

$$\begin{pmatrix} x_l \\ y_l \\ z_l \\ 1 \end{pmatrix} = \begin{bmatrix} R_L^{3\times3} & T_L^{3\times1} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x_v \\ y_v \\ z_v \\ 1 \end{pmatrix} \tag{1}$$

where $R_L \in \mathbb{R}^{3\times3}$ and $T_L \in \mathbb{R}^{3\times1}$ represent the rotation and translation from the Ego-Vehicle Coordinate System to the LiDAR Coordinate System, respectively.

Ego-Vehicle to LiDAR: The transformation from Ego-Vehicle Coordinate System to LiDAR Coordinate System is the inverse transformation of Eq.(1).

A.2. LiDAR⇔Camera

LiDAR to Camera: Regardless of whether it is a fisheye or a pinhole camera, the coordinate transformation formula from the LiDAR Coordinate System to the Camera Coordinate System is the same and is given as follows:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{bmatrix} R_C^{3\times3} & T_C^{3\times1} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x_l \\ y_l \\ z_l \\ 1 \end{pmatrix} \tag{2}$$

where (x_c, y_c, z_c) represents a three-dimensional coordinate point in the Camera Coordinate System. $R_C \in \mathbb{R}^{3\times3}$ and $T_C \in \mathbb{R}^{3\times1}$ represent the rotation and translation from the LiDAR Coordinate System to the Camera Coordinate System, respectively.

Camera to LiDAR: The transformation from Camera Coordinate System to LiDAR Coordinate System is the inverse transformation of Eq.(2).

A.3. Camera⇔Pixel

Camera to Pixel: The projection formulas of different types of cameras are different in the RoboSense dataset, the projection formula of a pinhole camera is as follows:

$$z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K^{3\times3} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}, K^{3\times3} = \begin{bmatrix} f_x & -1 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

(u, v) is pixel coordinate, $K \in \mathbb{R}^{3\times1}$ represents the camera intrinsic parameters, (f_x, f_y) represents the focal lengths of the camera, and (u_0, v_0) indicates the displacement of the camera's optical center from the origin of the Pixel Coordinate System. The projection formula from camera coordinate to pixel coordinate of the fisheye camera is very different, the camera projection process refers to the projection formula of Omnidirectional Camera (OCam) in [27].

Pixel to Camera: The transformation from Pixel Coordinate System to Camera Coordinate System in a pinhole camera model requires the inverse of Eq.(3). Since this is a 2D to 3D transformation, it is necessary to first determine the magnitude of z_c. The projection formula from pixel coordinate to camera coordinate of the fisheye camera refers to the projection formula of Omnidirectional Camera (OCam) in [27].

A.4. Ego-Vehicle⇔Global

Ego-Vehicle to Global: $R_G \in \mathbb{R}^{3\times3}$ and $T_G \in \mathbb{R}^{3\times1}$ represent the transformation matrices of the vehicle's orientation and position in the Global Coordinate System, respectively. The transformation formula for converting the coordinates (x_v, y_v, z_v) in the Ego-Vehicle Coordinate System to (x_g, y_g, z_g) in the Global Coordinate System is as follows:

$$\begin{pmatrix} x_g \\ y_g \\ z_g \\ 1 \end{pmatrix} = \begin{bmatrix} R_G^{3\times3} & T_G^{3\times1} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x_v \\ y_v \\ z_v \\ 1 \end{pmatrix} \tag{4}$$

Global to Ego-Vehicle: The transformation from Global Coordinate System to Ego-Vehicle Coordinate System is the inverse transformation of Eq.(4).

B. More Details of RoboSense

B.1. Annotation Statistics

We present more statistics on the annotations of RoboSense as shown in Tab. 8. It can be observed that our RoboSense dataset contains approximately 1.4M annotated

Table 8. The Number and proportion of 3D Boxes from all sensors (Global Scenes) and Livox LiDAR (Local Scenes) per category under different ranges (m) respectively.

Global/Local	Vehicle			Cyclist			Pedestrian			Total
	[0 - 10]	[10 - 30]	[30 -]	[0 - 10]	[10 - 30]	[30 -]	[0 - 10]	[10 - 30]	[30 -]	
Global (Hesai LiDAR)	165K	402K	343K	23K	38K	15K	187K	163K	51K	1.4M
	910K			76K			401K			
	65.00%			5.42%			28.64%			100%
Local (Livox LiDAR)	150K	282K	133K	20K	28K	7K	163K	103K	21K	907K
	565K			55K			287K			
	40.36%			3.93%			20.50%			64.79%

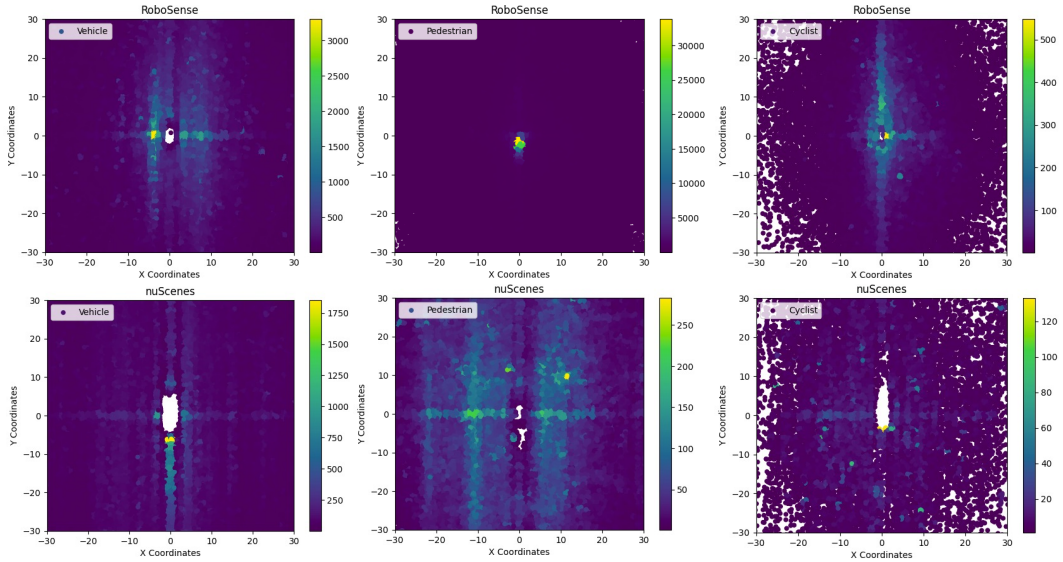


Figure A1. Comparison of annotated object distribution of different classes between RoboSense and nuScenes datasets.

objects, with vehicles and pedestrians comprising the majority, while cyclists are lesser. The distribution of objects is relatively uniform in terms of distance. Additionally, due to the smaller coverage area of Livox pointclouds (Local view) compared to Hesai pointclouds (Global view), the number of annotated objects in the Livox pointclouds is only 64.79% of that in the Hesai pointclouds. In Fig. A1, we further compare the distribution of annotated objects between our Robosense dataset and nuScenes dataset. It is obvious that our Robosense dataset contains significantly more annotated objects of vehicles, pedestrians, and cyclists classes respectively, which tend to be closer to the ego robot.

B.2. 3D Object Label Generation

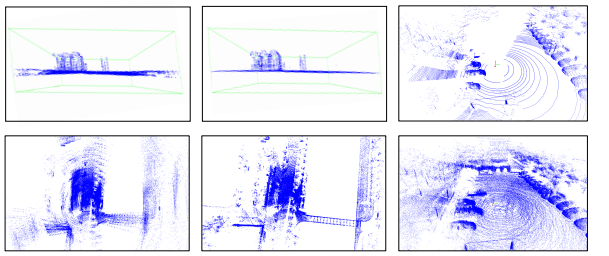
To generate high-quality 3D object annotations, we design a three-stage 3D object generation pipeline for different sensors covering various ranges. First, a pre-trained LiDAR detection model (i.e., [15]) of high precision is adopted to produce 3D objects on the full 360° view using high-quality Pandar64 points as input. Then expert annotators are required to refine the initial 3D boxes continuously throughout the whole sequences in each scene, based on splicing pointclouds which are obtained by aligning 4 vehicle-side LiDARs to the Ego-Vehicle coordinate through affine transformation. Besides, annotators need to supplement surrounding 3D boxes in a near range which are not scanned by the top Hesai LiDAR or fail to be detected owing to high occlusion and truncation. Last but not least, invalid 3D annotations should be excluded for target LiDAR coordinate and Camera coordinate respectively, where the annotated objects are not covered in the corresponding sensor data. Through multiple validation steps, highly accurate annotations can be achieved in both near and far ranges. We also release intermediate Pandar64 points for research usages.

B.3. Occupancy Label Preprocess

Occupancy label generation can be primarily divided into two parts: pointclouds densification and occupancy label determination. Unlike existing counterpart [33] which only utilizes the sparse keyframe LiDAR points, multi-frame aggregation operation is found to be indispensable for dense occupancy generation. For dynamic objects, the extracted

Table 9. 3D Detection results of different modalities on validation sets of RoboSense using *IoU* as matching criteria.

Task	Method	Vehicle@IoU=0.7/0.3			Cyclist@IoU=0.5/0.3			Pedestrian@IoU=0.5/0.3		
		3D AP↑	AOS↑	ASE↓	3D AP↑	AOS↑	ASE↓	3D AP↑	AOS↑	ASE↓
LiDAR 3D Detection	PointPillar [15]	43.7	45.5	13.3	39.5	39.6	69.2	52.6	36.6	34.9
	SECOND [39]	55.8	59.8	17.2	52.3	53.3	65.9	61.7	46.9	37.5
	PVRCNN [29]	53.5	57.9	16.9	53.0	50.7	55.9	58.9	43.4	38.4
	Transfusion-L [1]	**65.8**	**66.3**	17.3	**59.3**	**71.0**	78.5	**67.1**	**56.0**	42.7
Multi-view 3D Detection	BEVDet [12]	32.1	21.8	10.4	19.9	21.2	36.8	25.9	29.7	20.3
	BEVDet4D [11]	33.5	22.8	10.4	20.1	21.1	36.7	26.2	28.3	**17.7**
	BEVDepth [16]	33.4	22.8	**10.2**	22.6	22.2	41.6	27.7	28.1	17.9
	BEVFormer [17]	33.6	23.0	10.3	23.4	22.1	**35.3**	28.0	29.5	17.8



(a) Densified points without ICP (b) After ICP registration (c) Sparse vs. Dense keyframe points

Figure A2. Illustration of ICP and points densified process.

dynamic points of neighboring frames are subsequently concatenating for each object along the corresponding trajectory respectively, thus achieving the pointclouds densification. For static scenes, coordinate transformation is performed from the ego-vehicle coordinate to the global coordinate across time using ego-pose information, and then simply aggregate all static points on the ego-vehicle coordinate of current keyframe through concatenation.

Notably, owing to the complex driving scenarios with uneven ground and rapid pose changes especially when turning directions to avoid obstacles during data collection, pose drifts are observed in the IMU data. Therefore, the temporal aggregation results of pointclouds are inferior with misaligned horizon and ego-motion blur as shown in Fig. A2. To relieve these issues, ICP (Interative-Closed-Point) [28] is conducted additionally for static scene points registeration before multi-frame aggregation. Finally, densified pointclouds for a single frame can be obtained by fusing the static scenes with the dynamic objects.

Given dense points of a specific scene, we label all voxels within a fixed range by a resolution of $0.5m \times 0.5m$, based on the height of majority points inside each voxel. If the height is larger than a threshold σ, the voxel state is set to "occupied", otherwise "free". Moreover, considering the occlusion and truncation situations, some occupied voxels are not scanned by LiDAR beams and camera views actually. Hence we set part of voxels to "unknown" state which are invisible from both the LiDAR and camera views through tracing the casting ray.

Figure A3. Distribution of data collection scenarios in RoboSense dataset in Google Map.

B.4. Metric Comparison

In addition to the evaluation of 3D detection results with the proposed matching criteria (*Center-Point* distance and *Closest Collision-Point* distance), we also provide the corresponding evaluation results using the traditional 3D *IOU* (Intersection-Over-Union) matching criteria for comparison, as shown in Tab. 9. It is obvious that without distance differentiation, the evaluation results of 3D AP for both LiDAR-based and Camera-based methods are all in a low level, which can not reflect the objective performance and fail to satisfy the practical application requirements of the detection model. However, the proposed matching criterion is designed to measure the locating capability of closest collision points of nearby obstacles, which is more challenging and essential for low-speed driving scenarios.

B.5. Scene Distribution

Our RoboSense dataset contains 7.6K sequences, covering 6 main categories (including 22 different locations) of outdoor or semi-closed scenarios (i.e., S1-parks, S2-scenic spots, S3-squares, S4-campuses and S5-sidewalks or S6-streets). Fig. A3 illustrates the scene distributions of our collected data constructed for RoboSense dataset, which are surrounding Dishui Lake in Shanghai, China, with several markers drew in Google Map indicating the main locations performed data collection. Besides, the illustrations for each representative scenario among the collected locations are shown in Fig. A4-A9 respectively.

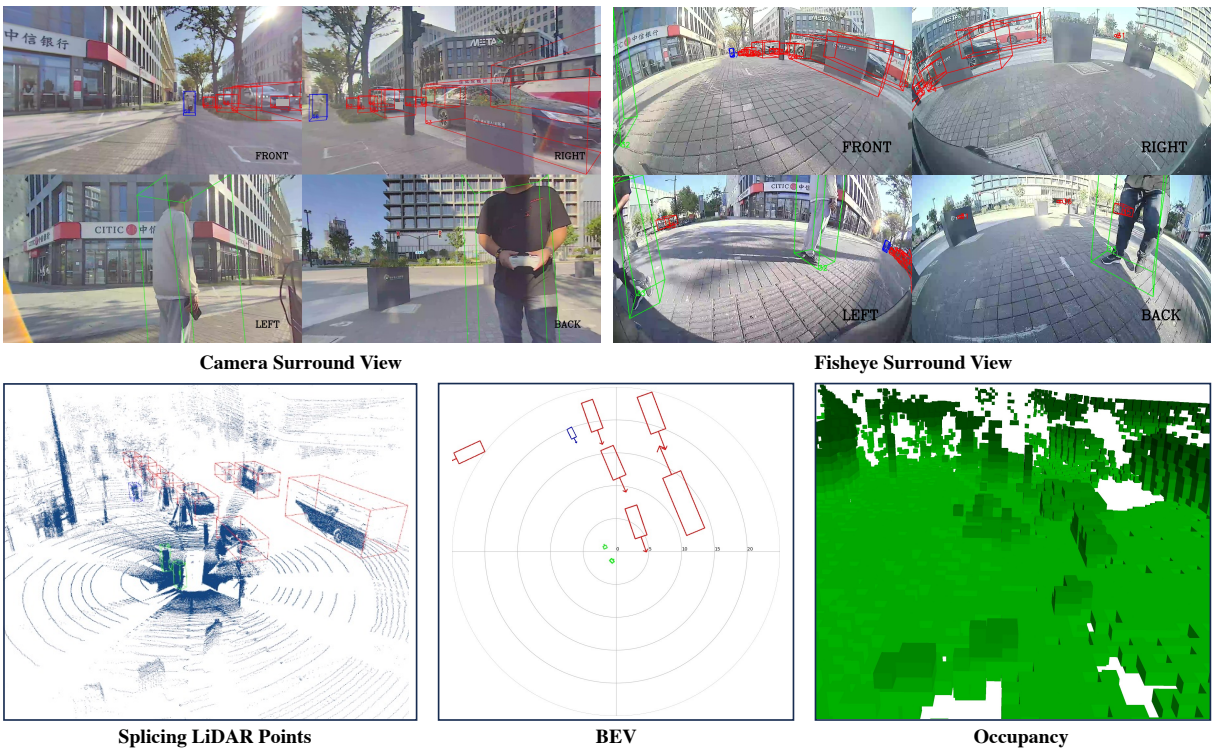


Figure A4. The illustration of S1-parks in Sequence-4906 at the 3-rd frame.

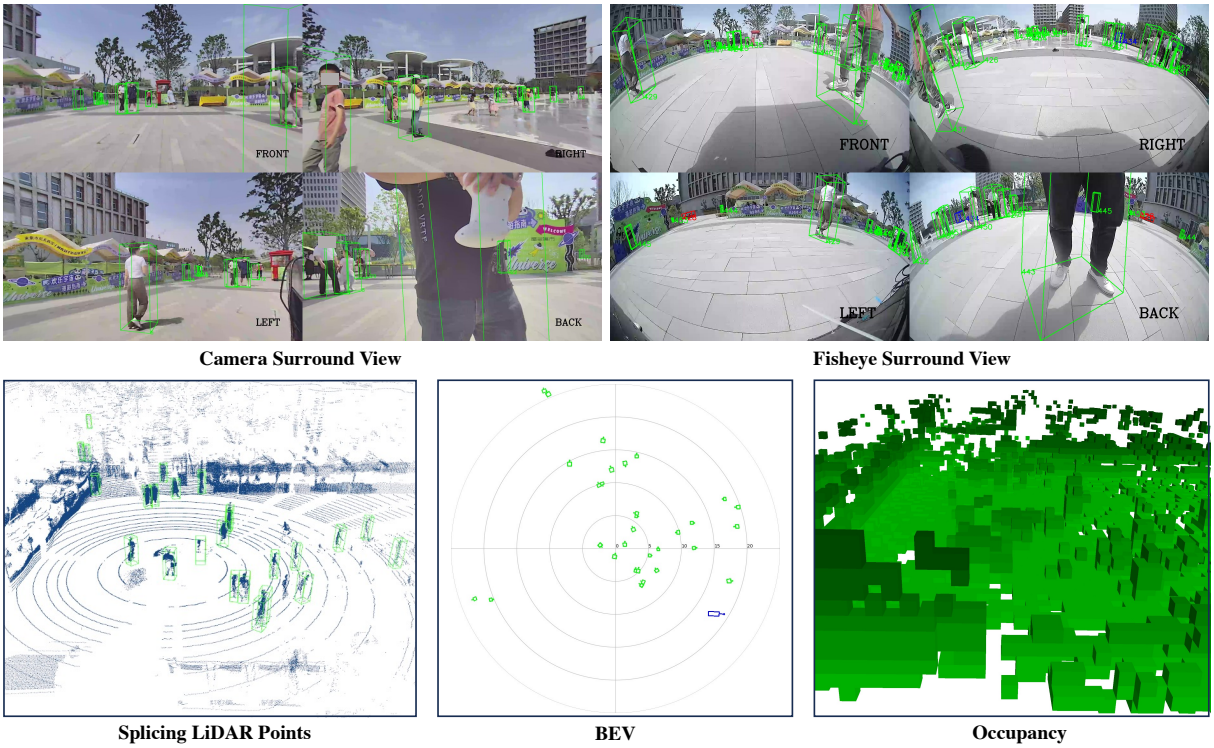


Figure A5. The illustration of S2-scenic spots in Sequence-1491 at the 13-th frame.

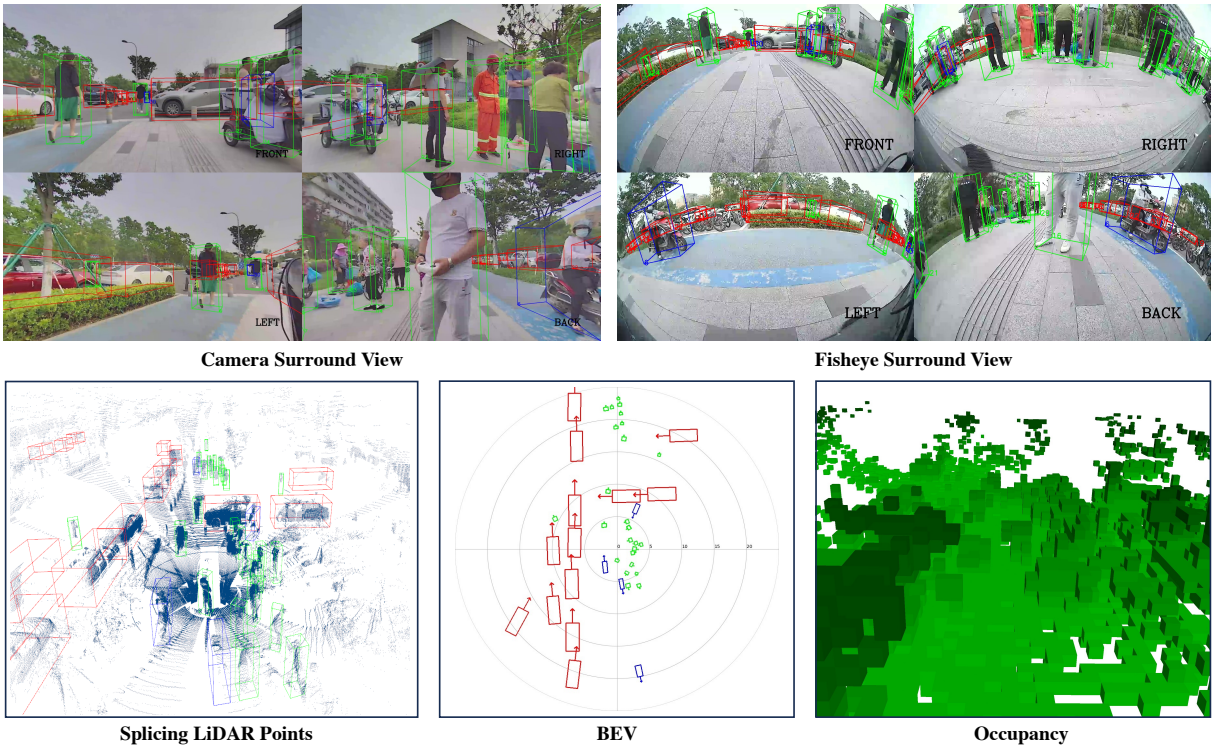


Figure A6. The illustration of S3-squares in Sequence-396 at the 2-nd frame.

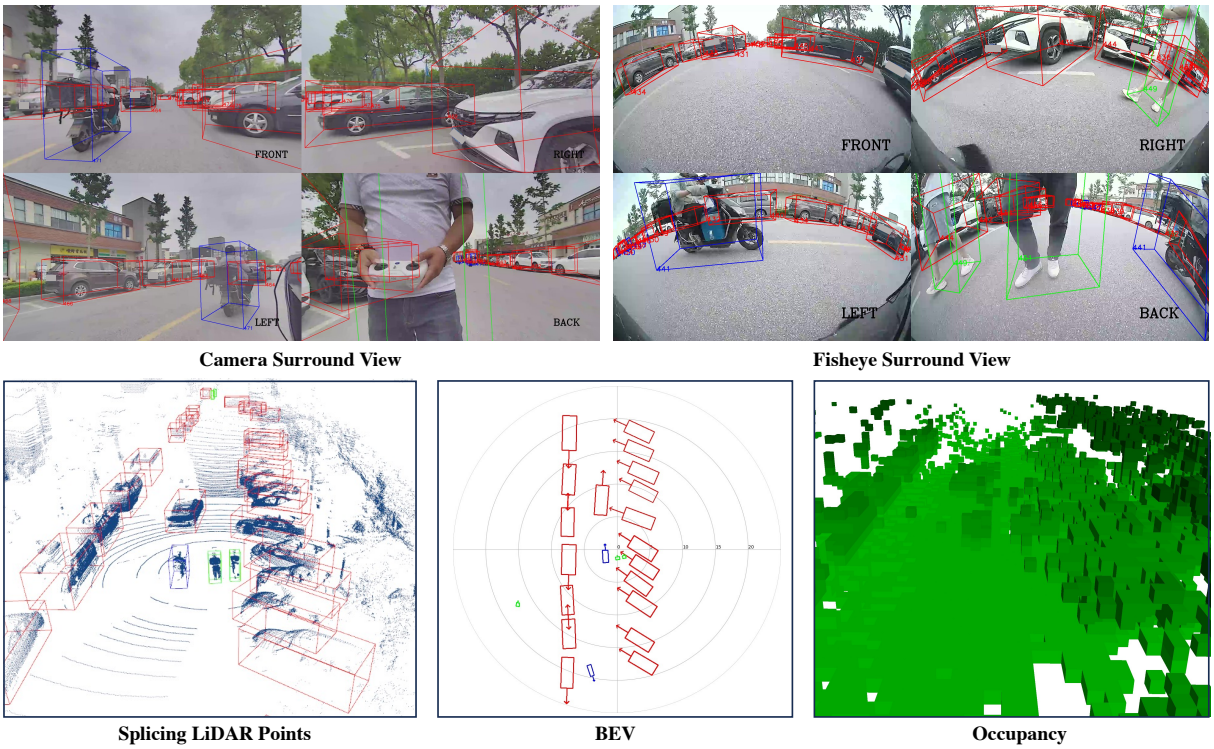


Figure A7. The illustration of S4-campuses in Sequence-2257 at the 16-th frame.

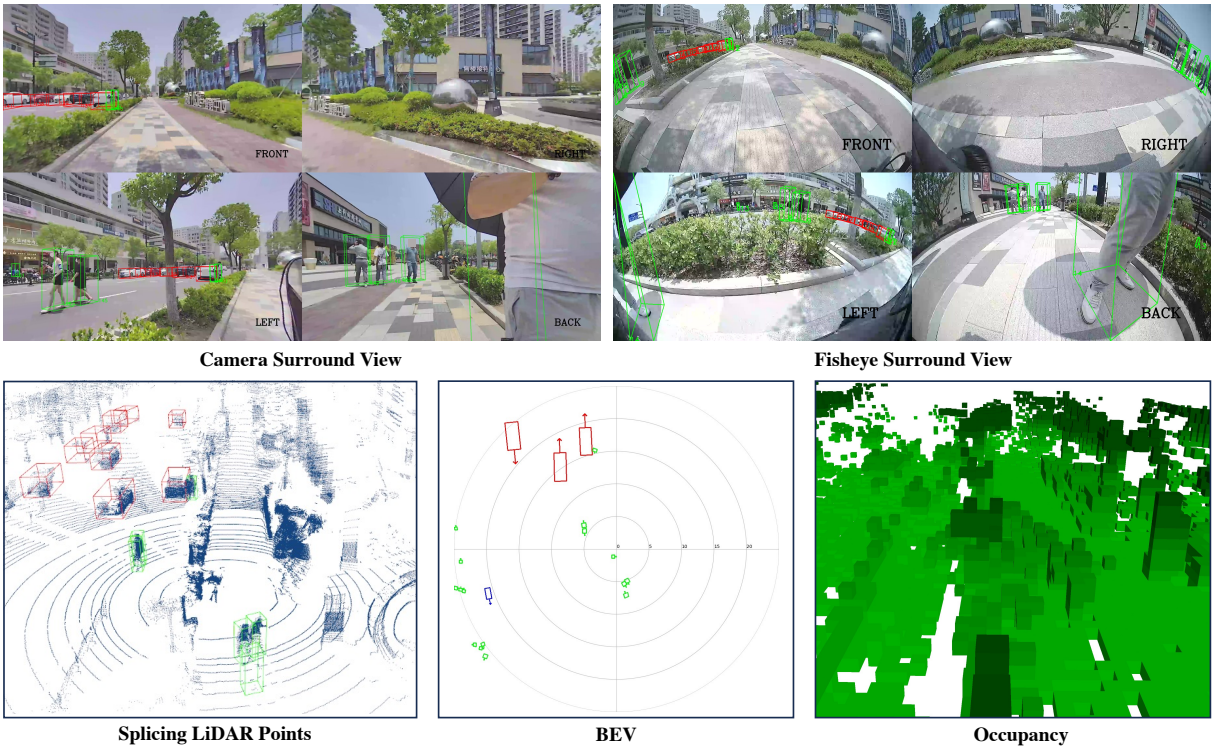


Figure A8. The illustration of S5-sidewalks in Sequence-2990 at the 10-th frame.

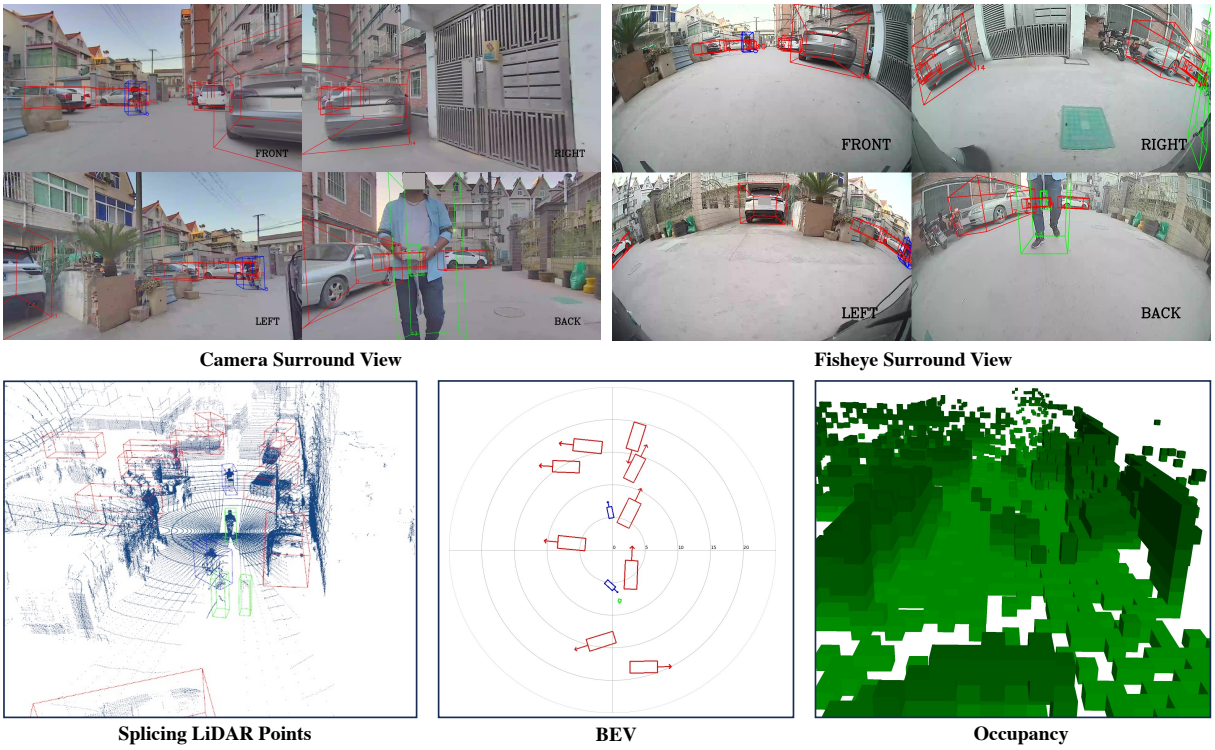


Figure A9. The illustration of S6-streets in Sequence-7018 at the 2-nd frame.