Lidar Point Cloud Obstacle Detection & Classification

1. Introduction of datasets:

The dataset of the Lidar point cloud of obstacle detection and classification provides 20,000 frames of 3D point cloud annotation data, including 10,000 frames of training data and 10,000 frames of test data. Among them, the training data can be used for the training of the algorithm model, with a total of about 236,000 obstacles; the test data can be used for the algorithm testing, with a total of about 239,000 obstacles. In addition, the dataset also provides 100 frames of downloadable data that can be used for debugging, testing, and visualization of algorithms.

2. Features of the data annotation:

This dataset covers real road scenes with a rich collection of Lidar data. Professional annotators mark four kinds of obstacles in each frame of the point cloud: pedestrians, vehicles, non-motor vehicles (cyclists) and others (dontCares) (Note 1). The annotations cover all the obstacles within 60m from the scene in a 360 ° view. The total number of obstacles annotated is about 475,000. The annotation results are shown in Figure 1. The green box refers to vehicles, the blue box refers to cyclists, the red box refers to pedestrians, and the yellow box refers to dontCare obstacles. For non-movable objects, such as traffic kiosks, traffic lights and green belts in the middle of roads are not annotated.

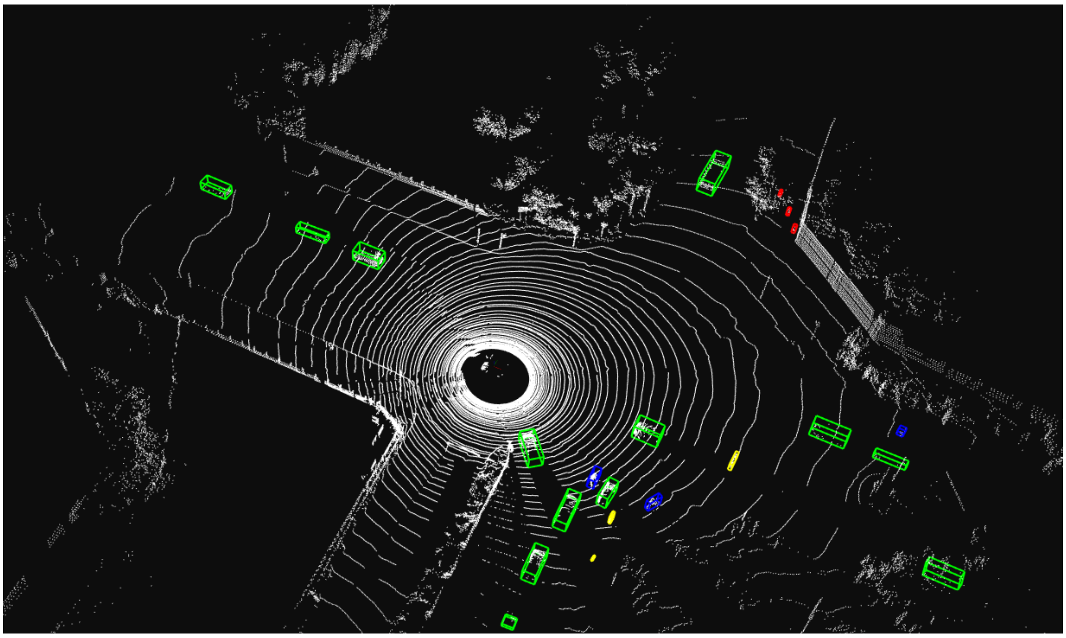

Figure 1. Schematic diagram of annotation results

3. Sensor configuration:

The Laser scanner model used for the data acquisition is Velodyne HDL-64E S3. The mounting position of the sensor is shown in Figure 2. The range of the vertical scanning angle is from + 2.0 ° to -24.9 ° and the scanning angle is 26.9 °. The scanning distance range is 120m. The operating frequency range is from 5Hz to 20Hz. The horizontal angle resolution is from 0.08 ° to 0.35 ° while the vertical angle resolution is 0.42 °. During the data acquisition, the scanner is in its 10Hz and Strongest working mode. The single frame of acquisition point cloud contains about 120,000 points. The point clouds obtained by scanning have had the motion compensation through the high-precision GPS/IMU installed on the vehicle.

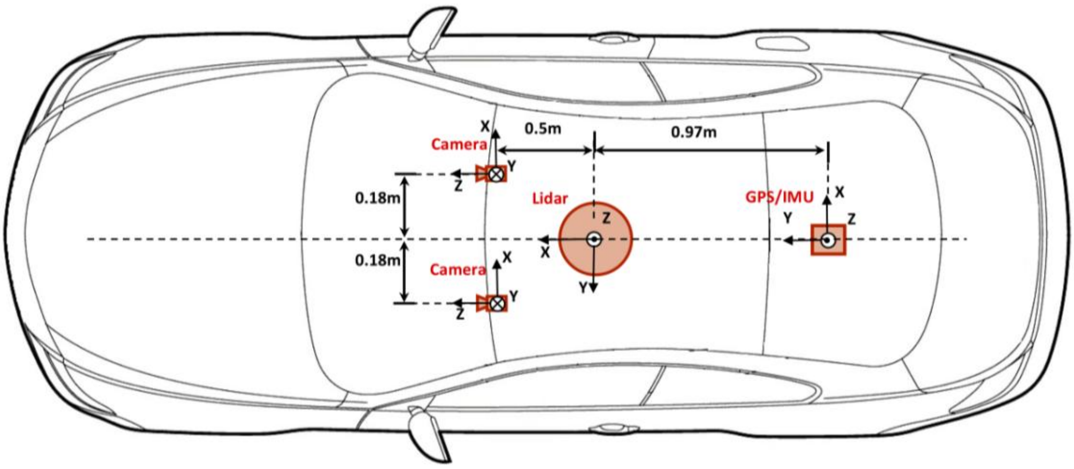

Figure 2. Laser scanner installation diagram

4. Data format:

For each frame, its point cloud data and annotation file are stored separately.with a corresponding name. As shown in the following figure, all the points in the figure are stored in the point cloud file and the annotation box in the figure is stored in the annotation file.

The point cloud data are stored in the format of binary files. The data are arranged in the order of X_1, Y_1, Z_1, I_1, X_2, Y_2, Z_2, I_2… (X_i, Y_i, Z_i refer to the spatial 3D coordinates of each point. I_i represents the reflectance value of this point and the effective value of the reflectance value is from 0 to 255) The data in each dimension are stored as the four-byte float type. The point cloud coordinates are defined under the Velodyne's own coordinate system (as shown in Fig. 2), specifically:

1. With the center of Velodyne installed on the vehicle's roof as the origin, the installation height is about 1.7m from the ground
2. Take the front of the car head as the positive direction of the X-axis
3. The car head faces forwards. The right left perpendicular to the car body is the positive direction of the Y-axis
4. The Z-axis direction meets the right-hand rule, that is, the direction perpendicular to the XY plane is the positive direction of the Z-axis

The annotation files correspond to the point cloud files. Each frame of the annotation file stores the obstacle information in this frame. The stored data contents are: type, center_x, center_y, center_z (coordinates of the center point), length, width, height (of the obstacle), yaw (the obstacle's orientation represented by the angle around the Z-axis).

5. Detailed descriptions of datasets:

5.1. Training set

10,000 frames of the training set data are opened for the model training. There are about 236,000 obstacles in the training set. The ratio of the four kinds of obstacles is about: pedestrians: cyclists: vehicles: dontCares = 15.0%: 7.5%: 73.7%: 3.8%. The point cloud data is named from 000_00000000.bin to 000_00009999.bin. The annotation file corresponds to the point cloud data one by one, which is named from 000_00000000.bin.txt to 000_00009999.bin.txt

The storage directory is as follows:

```
trainsets/
├──── bin_files/
│    ├────000_00000000.bin
│    ├────000_00000001.bin
│    ├────…
│    └────000_00009999.bin
└────label_file/
     ├────000_00000000.bin.txt
     ├────000_00000001.bin.txt
     ├────…
     └────000_00009999.bin.txt
```

5.2. Test set

10,000 frames of the test set data are opened for the algorithm test. There are about 239,000 obstacles in the test set. The ratio of the four kinds of obstacles is about: pedestrians: cyclists: vehicles: dontCares = 14.9%: 7.3%: 74.0%: 3.8%. The point cloud data is named from 001_00000000.bin to 001_00009999.bin. The annotation file corresponds to the point cloud data one by one, which is named from 001_00000000.bin.txt to 001_00009999.bin.txt

The storage directory is as follows:

```
testsets/
├────bin_files/
│    ├────001_00000000.bin
│    ├────001_00000001.bin
│    ├────…
│    └────001_00009999.bin
└────label_file/
     ├────001_00000000.bin.txt
     ├────001_00000001.bin.txt
     ├────…
     └────001_00009999.bin.txt
```

5.3. Sample set

100 frames of the downloadable sample set are opened for debugging, testing, and visualization of algorithms. There are about 3,000 obstacles in the sample set. The ratio of the four kinds of obstacles is about: pedestrians: cyclists: vehicles: others = 11.8%: 7.8%: 77.3%: 3.1%. The point cloud data is named from 002_00000000.bin to 002_00000099.bin. The annotation file corresponds to the point cloud data one by one, which is named from 002_00000000.bin.txt to 002_00000099.bin.txt

The storage directory is as follows:

```
3d-sample/
├───bin_files/
│   ├───002_00000000.bin
│   ├───002_00000001.bin
│   ├───…
│   └───002_00000099.bin
└───label_file/
    ├───002_00000000.bin.txt
    ├───002_00000001.bin.txt
    ├───…
    └───002_00000099.bin.txt
```

6. Data usage:

This dataset can be used for the development and evaluation of obstacle detection algorithms and obstacle classification algorithms. For the machine-based learning algorithm, the training set can be used to train the algorithm model. For the rule-based algorithm, the test set can be directly used to evaluate the effect of the algorithm. The specific data usage may include three steps: model training, prediction and evaluation.

6.1. Model training

The learning-based detection & classification algorithm requires a model training step, which is input and output as follows:

Input	A set of point cloud data and annotation data provided by the platform, such as the training set. The point cloud is named: 001_00000000.bin,…, The annotation file is named: 001_00000000.bin.txt,…The The location of the training data is used

	directly in the script, such as: ./trainsets
Output	Model file, which is named as: Model.iteration (model keyword + '.' + Iterated rounds, for example: the 10th round of the model, storage path for models / model.10)

A typical directory structure is as follows:

```
sample/
├────trainsets/(input)
│    ├────bin_files/
│    │    ├────000_00000000.bin
│    │    ├────000_00000001.bin
│    │    ├────…
│    │    └────000_00009999.bin
│    └────label_file/
│         ├────000_00000000.bin.txt
│         ├────000_00000001.bin.txt
│         ├────…
│         └────000_00009999.bin.txt
└────model/ (output)
     └────model.10
```

6.2. Prediction

When the test set is used to evaluate the performance, the input and output are as follows:

Input	A set of point cloud data provided by the platform, such as the test set named: 001_00000000.bin, ...
Output	The test result format of of each frame of the point cloud is the same as that of the annotation file, specifically: The prediction result of the first frame: file: 001_00000000.bin.txt: line1: type center-x center-y center-z length width height yaw line2: … Each frame corresponds to a result file. Each line in the file corresponds to an obstacle detected by the algorithm.

	In order to help the benchmark evaluate the result, the name of the result file should be consistent with that of the annotation file, such as: 001_00000000.bin.txt, ...

6.3. Evaluation

When Benchmark is used to evaluate the obstacle detection algorithms and classification algorithms, its input and output are as follows:

Input	A set of result files obtained by the algorithm. The format is the same as that of the annotation file, specifically: 001_00000000.bin.txt: line1: type center-x center-y center-z length width height yaw line2: … Each frame corresponds to a result file. Each line in the file corresponds to an obstacle detected by the algorithm. Note: the file name of the evaluation input should be consistent with that of the corresponding annotation file, that is, 001_00000000.bin.txt,…
Output	Results of evaluation indicators. The formats are as follows: ——Obstacle detection: F-measure precision recall —— Obstacle classification: mean_accuracy vehicle_accuracy pedestrian_accuracy cyclist_accuracy

Users can evaluate the effect of the model based on the indicators above.

The dataset provides 100 frames of downloadable data for the debugging, testing and visualization of algorithms.

7. Evaluation Criteria:

7.1. Evaluation Criteria for Obstacle Detection Algorithms

The input of the obstacle detection algorithm is a single-frame 3D point cloud. The detection algorithm segments the 3D points into several point cloud sets (point set) through the segmentation/clustering operation. It also outputs the target obstacles in the point set (pedestrians, vehicles, cyclists, others). The effect of the detection algorithm is measured by two classical indicators [1]: *precision* and *recall*.

The precision is the ratio of the number of obstacles correctly detected by the algorithm N_{object} to the number of point sets of all detected outputs $N_{cluster}$. The precision is defined as the formula (1)

$$precision = \frac{N_{object}}{N_{cluster}} \tag{1}$$

The recall rate is the ratio of the number of obstacles correctly detected by the algorithm N_{object} to the number of real obstacles obtained by manual annotation $N_{groundtruth}$. The recall rate is defined as the formula (2)

$$recall = \frac{N_{object}}{N_{groundtruth}} \tag{2}$$

In order to facilitate the comparison of algorithm detection effects, the evaluation platform uses the single index F-measure [1] to measure the combined effect of the algorithm. F-measure is the weighted harmonic mean of the precision and the recall rate, which is defined as the formula (3):

$$F = \frac{1}{\alpha \frac{1}{precision} + (1-\alpha)\frac{1}{recall}} \tag{3}$$

Among them, the weight value α is used to balance the importance of the precision and recall rate. These two indicators are equally important by default, that is, $\alpha = {}^{1}\!/_{2}$.

In this case, the calculation of the F-measure can be presented by the formula (4)

$$F = 2\frac{precision \times recall}{precision + recall} \tag{4}$$

The calculation of the precision and the recall rate needs to count the number of obstacles correctly detected by the algorithm N_{object}. The standard used by this evaluation platform to determine whether the output point sets of the algorithm are the correct target obstacles is as follows: compare each point set output by the algorithm with the collection of obstacles annotated manually. Calculate the Jaccard Index - JI, that is, the IoU of the point set to the annotation obstacle point set:

$$JI(P_{cluster}, P_{groundtruth}) = \frac{|P_{cluster} \cap P_{groundtruth}|}{|P_{cluster} \cup P_{groundtruth}|} \tag{5}$$

Where, $P_{groundtruth}$ represents the point set of obstacles annotated manually, $P_{cluster}$ stands for the output point set of the corresponding algorithm detection. The evaluation system uses the Jaccard Index calculation based on 3D points instead of the method based on the bounding box mentioned in the literature [2] so as to reduce the impact of manual annotation. For the obstacle set with the same target, different annotators may produce different sizes of bounding boxes to different directions. As shown in Figure 3, green points are the target obstacles. Although two annotators produce different sizes of red and black bounding boxes to different directions, the 3D point sets contained in these two bounding boxes are exactly the same.

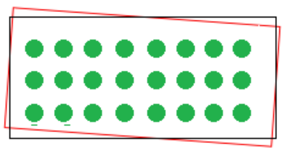

Figure 3

After obtaining the corresponding relation between the point set of the algorithm detection and the point set of the annotation obstacle as well as the Jaccard Index, we use the threshold to filter out obstacles that the system considers to be “correct”, that is, the output point sets whose value of JI is greater than the threshold value T are considered to be true positives. Similar to the literature [2], considering the universal existence of the annotation errors, the platform sets the value of the threshold to be 0.5.

7.2. Evaluation Criteria for Obstacle Classification Algorithms

The obstacle classification algorithm divides the detected obstacles into four categories: pedestrians, motor vehicles, non-motor vehicles and others. The platform uses the category accuracy to measure the effects of different classification algorithms [2]:

$$accuracy = \frac{tp}{tp + fp + fn} \tag{6}$$

Take pedestrians as the example. tp is the number of true positives which are correctly classified as pedestrians, while fp is the number of false positives which are misclassified as pedestrians. Finally, fn defines the number of false negatives which misclassify the pedestrians. Based on the output result of the algorithm, the platform concludes the classification precisions of pedestrians, motor vehicles and non-motor vehicles respectively. Then the platform arranges the algorithm effects in order by calculating the arithmetic means of these three precisions. Note that this evaluation system does not consider the classification effect of "other" obstacles (Note 2).

References:

[1] Christopher Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to information retrieval*, Cambridge University Press. 2008

[2] Mark Everingham,Luc Van Gool, Christopher Williams, John Winn and Andrew Zisserman, *The pascal visual object classes (voc) challenge*. International Journal of Computer Vision, 2010

Note 1: "dontCares" represent obstacles that have impacts on the driving (e.g. vehicle driving road) but cannot be identified as motor vehicles, non-motor vehicles or pedestrians by annotators due to factors such as the long distance or occlusion.

Note 2: During the evaluation process, we believe that the algorithm needs to correctly detect and report "other" obstacles (which has an effect on the precision/recall indicator of the detection task). But the algorithm is not required to correctly classify "other" obstacles (which has no effect on the accuracy indicator of the classified task).