

激光点云障碍物检测分类

1. 数据集简介：

本激光点云障碍物检测分类数据集提供 20,000 帧三维点云标注数据，包括 10,000 帧训练数据和 10,000 帧测试数据。其中训练数据可用于算法模型的训练，含障碍物共约 236,000 个；测试数据可用于算法的测试，含障碍物共约 239,000 个。此外，数据集还提供 100 帧可下载数据，可用于算法的调试、测试及可视化。

2. 数据标注特色：

本数据集全部为真实路面场景，数据采集场景丰富，在每帧点云中通过专业标注人员标注出四类障碍物：行人(pedestrian)、机动车(vehicle)、非机动车(cyclist)以及其他(dontCare) (注 1)。标注覆盖了场景 360°内 0-60 m 的所有障碍物，标注障碍物总量约为 475,000 个。标注结果如图 1 所示，图中绿色框表示机动车，蓝色框表示非机动车，红色框表示行人，黄色框表示其它障碍物。对于不可移动物体，如路中间的交通亭、红绿灯、绿化带等不进行标注。

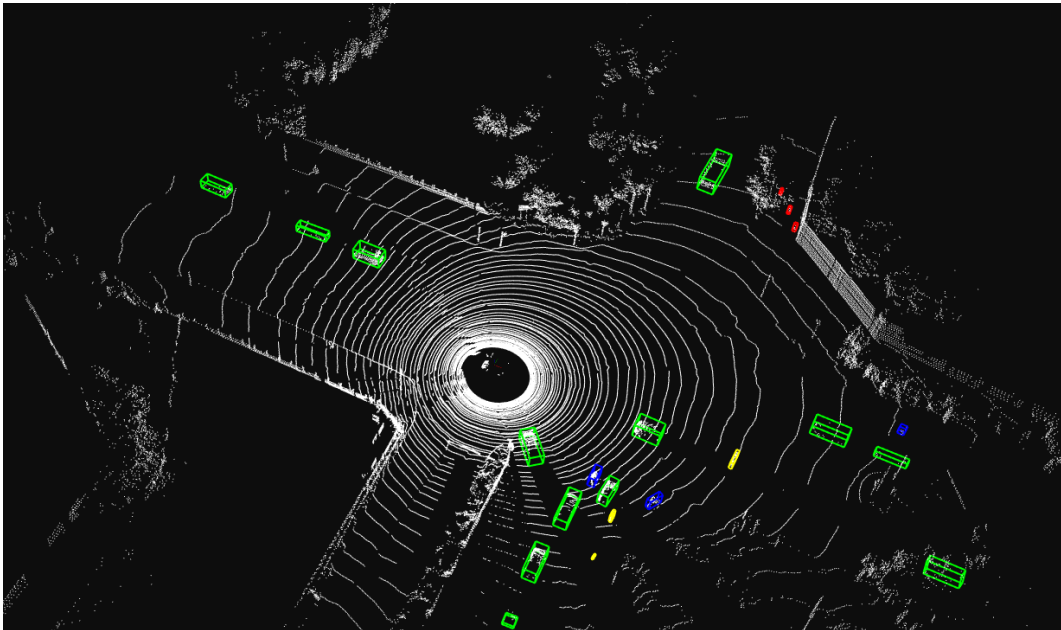


图 1. 标注结果示意图

3. 传感器配置：

数据采集使用的激光雷达型号为 Velodyne HDL-64E S3。传感器安装位置如图 2 所示。竖直扫描角度范围为 $+2.0^{\circ} \sim -24.9^{\circ}$ ，扫描角度 26.9° 。扫描距离范

围为 120m。工作频率区间为 5Hz~20Hz，水平角度分辨率为 $0.08^{\circ} \sim 0.35^{\circ}$ ，竖直角度分辨率为 0.42° 。数据采集过程使用 10Hz、Strongest 的雷达工作模式，单帧采集点云包含约 120,000 个点。扫描得到的点云已经通过采集车上安装的高精度 GPS/IMU 进行了运动补偿。

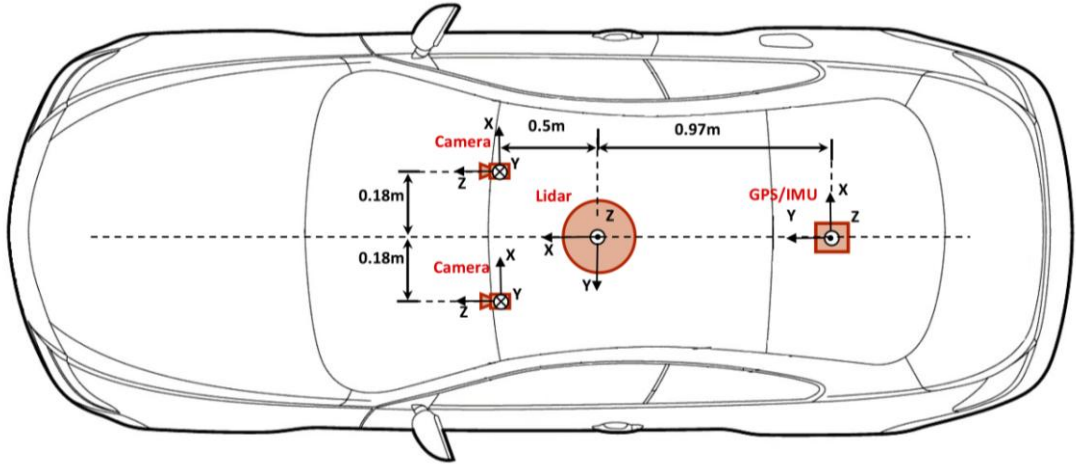


图 2. 激光雷达安装示意图

4. 数据格式:

点云数据和标注文件分别存储，每一帧单独存为一个文件，名称一一对应。如下图示例，图中所有点存储于点云文件，图中标注框存储于标注文件。

点云数据格式为二进制文件存储，数据依次以 $X_1, Y_1, Z_1, I_1, X_2, Y_2, Z_2, I_2, \dots$ 排列 (X_i, Y_i, Z_i 表示每个点的空间三维坐标, I_i 表示此点的反射值, 反射值的有效范围为 0—255)，每一维度的数据分别使用四字节 float 类型存储，点云坐标定义在 Velodyne 自身坐标系下（如图 2 所示），具体为：

1. 以车顶安装的 Velodyne 中心为原点，安装高度大约离地 1.7m
2. 以车头正前方为 X 轴正方向
3. 车头朝前，垂直车身正左方为 Y 轴正方向
4. Z 轴方向满足右手法则，即垂直 XY 平面向上为 Z 轴正方向

标注文件与点云文件一一对应，每一帧标注文件中存储该帧中的障碍物信息。存储的数据内容依次为：type (类别)，center_x, center_y, center_z(中心点坐标), length, width, height(障碍物长宽高), yaw (偏航角，即障碍物绕 z 轴的旋转角)。

5. 数据集展开说明:

5.1. 训练集

本次开放训练集数据 10000 帧，可用于模型的训练。训练集中含障碍物共约

236,000 个。四类障碍物占比约为：行人:非机动车:机动车:其他 = 15.0% : 7.5% : 73.7%: 3.8%。点云数据命名为：000_00000000.bin~000_00009999.bin.标注文件与点云数据一一对应，命名为 000_00000000.bin.txt~000_00009999.bin.txt

存储目录如下：

```
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. 测试集

本次开放测试集数据 10000 帧，可用于算法的测试。测试集中含障碍物共约 239,000 个。四类障碍物占比约为：行人:非机动车:机动车:其他 = 14.9% : 7.3% : 74.0%: 3.8%。点云数据命名为：001_00000000.bin~001_00009999.bin.标注文件与点云数据一一对应，命名为 001_00000000.bin.txt~001_00009999.bin.txt

存储目录如下：

```
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. 样例集

本次开放可下载样例集 100 帧，可用于算法的训练、测试和调试。样例集中含障碍物共约 3,000 个，四类障碍物占比约为：行人:非机动车:机动车:其他 = 11.8% ： 7.8% ： 77.3% :3.1% 。点云数据命名为：002_000000000.bin ~ 002_000000999.bin,标注文件与点云数据一一对应，命名为 002_000000000.bin.txt~ 002_000000999.bin.txt

存储目录如下：

```
3d-sample/
├── bin_files/
│   ├── 002_000000000.bin
│   ├── 002_000000001.bin
│   ├── ...
│   └── 002_000000999.bin
└── label_file/
    ├── 002_000000000.bin.txt
    ├── 002_000000001.bin.txt
    ├── ...
    └── 002_000000999.bin.txt
```

6. 数据用途：

本数据集可用于障碍物检测算法和障碍物分类算法的研发和评测。对于基于机器学习的算法，可使用训练集对算法模型进行训练。对于基于规则的算法，可直接使用测试集对算法的效果进行评测。具体数据用途可包括模型训练、预测和评测三个步骤。具体接口详情参见“入门指南->平台接口介绍”。

6.1. 模型训练

基于学习的检测分类算法需要模型训练步骤，该步骤输入输出如下，

输入	平台提供的一组点云数据与标注数据，如训练集，点云命名为：001_000000000.bin, ... , 标注文件命名为：001_000000000.bin.txt, 。训练数据的位置在脚本中直接使用，如： ./trainsets
输出	模型文件，命名定义为:

	model.iteration(model 关键字+ ‘.’ +迭代轮数,例如:表示第 10 轮的模型，存放路径为 models/model.10)
--	--

一个典型的目录结构为：

```
sample/
├── trainsets/(input)
│   ├── bin_files/
│   │   ├── 000_000000000.bin
│   │   ├── 000_000000001.bin
│   │   ├── ...
│   │   └── 000_000099999.bin
│   └── label_file/
│       ├── 000_000000000.bin.txt
│       ├── 000_000000001.bin.txt
│       ├── ...
│       └── 000_000099999.bin.txt
└── model/ (output)
    └── model.10
```

6.2. 预测

在使用测试集进行效果评测时，其输入输出如下：

输入	平台提供的一组点云数据，如测试集，命名为： 001_000000000.bin, ……
输出	每帧点云的检测结果，其格式同标注文件，具体为： 第一帧预测结果：file: 001_000000000.bin.txt: line1: type center-x center-y center-z length width height yaw line2: ... 每帧对应一个结果文件，文件中每行对应一个算法检测到的障碍物。 为便于 benchmark 对结果进行测评，结果文件命名应与标注文件保持一致，如：001_000000000.bin.txt, ……

6.3. 评测

benchmark 评估障碍物检测算法和分类算法时，其输入输出如下：

输入	<p>算法得到的一组结果文件，其格式同标注文件，具体为：</p> <p>001_000000000.bin.txt:</p> <p>line1: type center-x center-y center-z length width height yaw</p> <p>line2: ...</p> <p>每帧对应一个结果文件，文件中每行对应一个检测到的障碍物。</p> <p>注意：评测输入的文件名需与对应的标注文件名称保持一致，即 001_000000000.bin.txt, ……</p>
输出	<p>评测指标结果，格式如下：</p> <p>——障碍物检测：</p> <p>F-measure</p> <p>precision</p> <p>recall</p> <p>——障碍物分类：</p> <p>mean_accuracy</p> <p>vehicle_accuracy</p> <p>pedestrian_accuracy</p> <p>cyclist_accuracy</p>

用户可根据以上指标评判模型的效果。

数据集中提供 100 帧可下载数据，可用于算法的调试、测试及可视化。

7. 评测标准：

7.1.障碍物检测算法评测标准

障碍物检测算法的输入为单帧三维点云，检测算法通过分割/聚类（segmentation/clustering）操作将三维点分割成若干个点云集合（点集）并输出点集中的目标障碍物（行人、机动车、非机动车、其他）。检测算法的效果使用两个经典指标来衡量[1]：精确率（*precision*）和召回率（*recall*）。

精确率是指被算法正确检测到的障碍物个数 N_{object} 与所有检测输出的点集个数 $N_{cluster}$ 的比率，衡量的是检测算法的查准率，精确率定义如公式（1）所示：

$$precision = \frac{N_{object}}{N_{cluster}} \quad (1)$$

召回率是指被算法正确检测到的障碍物个数 N_{object} 和经人工标注验证得到的

真实障碍物个数 $N_{groundtruth}$ 的比率，衡量的是检测算法的查全率，召回率定义如公式 (2) 所示：

$$recall = \frac{N_{object}}{N_{groundtruth}} \quad (2)$$

为便于比较算法检测效果，本评测平台采用单指标 F-measure[1] 来度量算法综合效果。F-measure 是精确率和召回率的加权调和平均值，其定义如公式 (3) 所示：

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

其中，权重值 α 用于平衡精确率和召回率的重要性。这两个指标默认同等重要，即 $\alpha = 1/2$ ，此时 F-measure 的计算方式可用公式 (4) 表示为：

$$F = 2 \frac{precision \times recall}{precision + recall} \quad (4)$$

计算精确率和召回率需要统计被算法正确检测到的障碍物个数 N_{object} 。本评测平台判断算法输出点集是否为正确目标障碍物的标准如下：将算法输出的每一点集与人工标注障碍物集合做比对，计算杰卡德系数 (Jaccard Index - JI)，即该点集与标注障碍物点集的交并比：

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

其中 $P_{groundtruth}$ 表示人工标注障碍物的点集， $P_{cluster}$ 表示与其对应的算法检测输出点集。本评测系统采用基于三维点的杰卡德系数计算方式，而非文献[2]中提到的基于包围盒 (bounding box) 的方式，旨在降低人工标注的影响。对于同一目标障碍物点集，不同标注人员有可能标出不同朝向和尺寸的包围盒。如图 3 所示，绿点为目标障碍物点，虽两位标注人员分别标出朝向大小不完全相同的红色和黑色包围盒，但这两个包围盒中包含的三维点集完全相同。

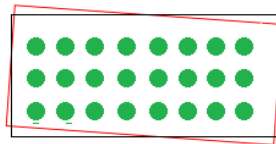


图 3

计算得到算法检测点集和标注障碍物点集的一一对应关系以及它们间的杰卡德系数后，我们用阈值筛选出系统认为检测“正确”的障碍物，即 JI 值大于阈值 T 的输出点集被认为正确(true positives)。同文献[2]，考虑标注误差的普遍存在，本平台将阈值 T 设为 0.5。

7.2. 障碍物分类算法评测标准

障碍物分类算法将检测到的障碍物分为四个类别：行人、机动车、非机动车和其他。本平台用类别准确率（*accuracy*）度量不同分类算法的效果[2]：

$$accuracy = \frac{tp}{tp + fp + fn} \quad (6)$$

以行人类为例， tp 为被正确分类为行人的障碍物（true positives）个数， fp 为被错误分类成行人的障碍物（false positives）个数，最后 fn 定义为行人类障碍物被错分成别类（false negatives）的个数。本平台根据算法输出结果分别统计行人、机动车和非机动车三种类别的分类准确率，再通过计算该三种类别准确率的算术平均值对算法效果进行排序。注意本评测系统不考虑“其他”类障碍物的分类效果^{（注2）}。

参考文献：

- [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

注 1：“其他类”代表对行驶有影响（如位于车辆行驶道路上），但由于距离远或遮挡等因素无法被标注人员确定为机动车、非机动车或行人的一类障碍物。

注 2：在评测过程中，我们认为算法需要正确检测并汇报出“其他”类障碍物（对检测任务的 precision/recall 指标有影响）但不要求算法对其进行正确分类（对分类任务的 accuracy 指标没有影响）。