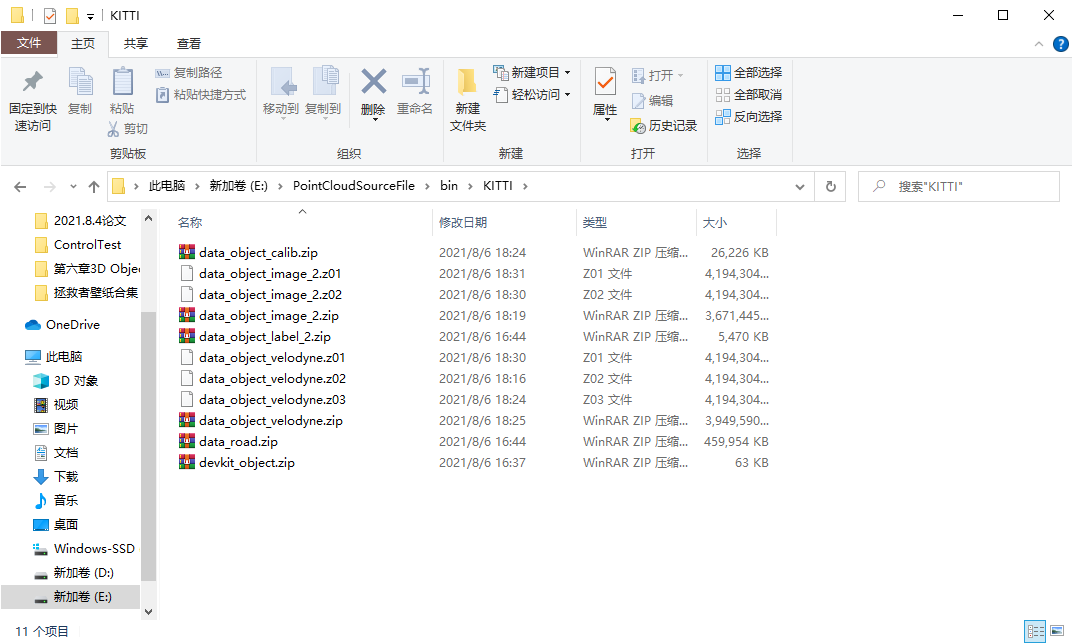
**step1 环境配置**

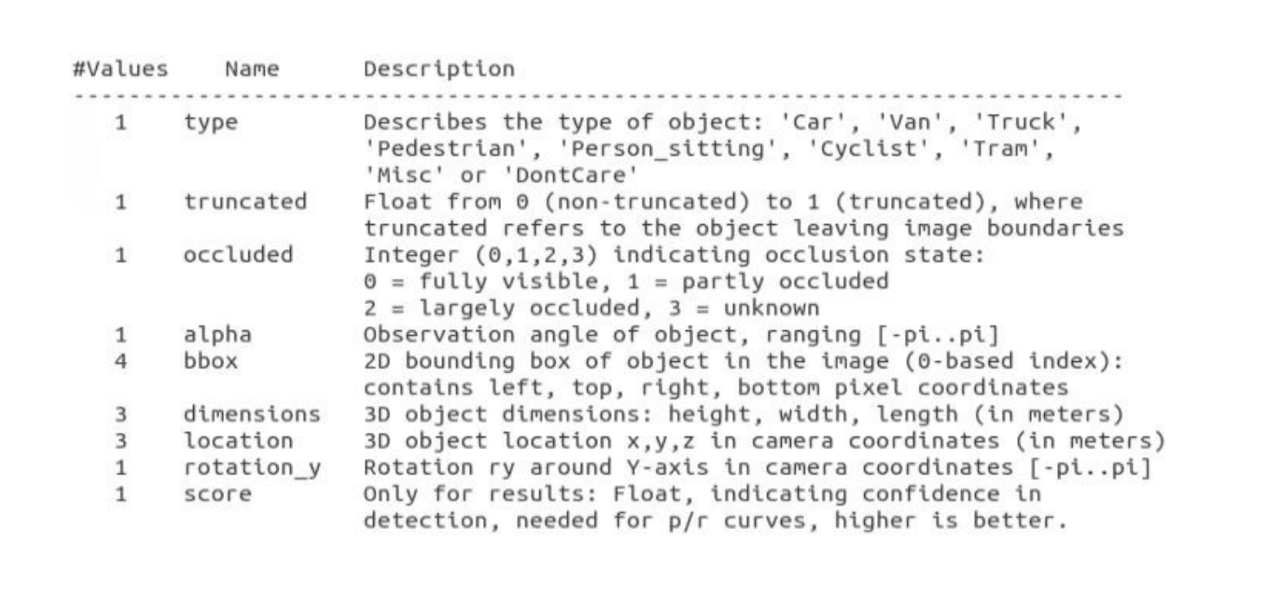
Setup the KITTI object detection evaluation environment

下载数据集



在命令行窗口运行监测git-clone代码git-clone.txt

**step2 阅读 devkit 开发手册，了解result需要的固定数据格式**



其中第1列truck 表示图中出现了卡车（一共有’Car’, ‘Van’, ‘Truck’,‘Pedestrian’, ‘Person\_sitting’, ‘Cyclist’, ‘Tram’,‘Misc’ or 'DontCare’这些类别，Don’t care 是没有3D标注的，原因是雷达扫不了那么远，即使可以视觉检测出来）

第2列0.0表示其是否被截断的程度为0。（如果车在图片边缘，那么就有可能发生部分被截断的情况。用0-1 表示被截断的程度。）

第3列0表示没有被遮挡。（0表示完全可见，1表示部分遮挡，2表示大部分被遮挡，3表示未知。）

第4列 -1.57 表示卡车中心与相机中心构成的矢量与在bird view下的夹角为-1.57，实际上就是说明改开叉在-90，即正前方。这个信息反映目标物体中心在bird view相对方向信息。

第5-8列的599.41 156.40 629.75 189.25是目标的2D bounding box 像素位置，形式为xyxy，前两个值为bounding box左上点的x，y位置，后两个点为右下角的x,y位置。

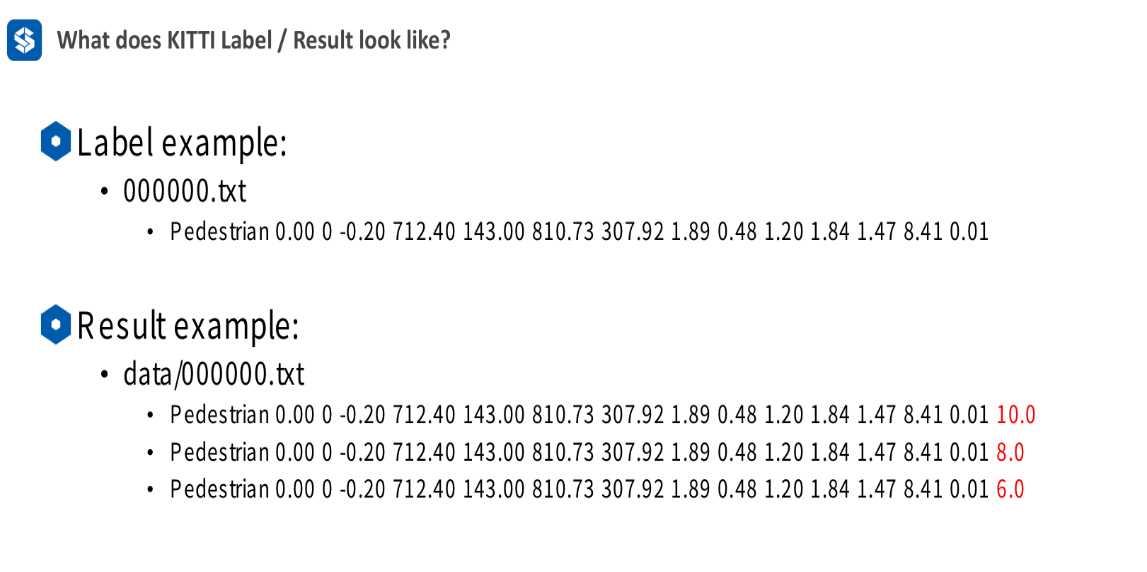
第9-11列 2.85 2.63 12.34 表示该车的高度，宽度，和长度，单位为米。（H,W,L）

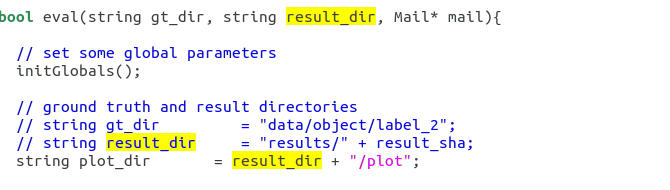
第12-14列 0.47 1.49 69.44 表示该车的3D中心在相机坐标下的xyz坐标。

第15列 -1.56 表示车体朝向，绕相机坐标系y轴的弧度值。注意和第4列区别开来，第四列不在乎车体朝向，而是车体中心与相机中心所构成矢量在与相机坐标系z轴的夹角（其实这里笔者有点疑虑，如果车体中心位置已知，车体朝向是不知道的，但是第4列的alpha是可以算出来的，那么其实第4列的数据是冗余的？）

第十六列，检测的置信度

**step3 生成 results 数据**

使用ground truth的label数据修改，得到result数据，观察检测结果  
输出的resul数据和label数据比较，主要比标签多了第16个值，第16个值为confidence置信度，表示形式为score  
下载的 训练集 label\_2：

1. 创建文件夹  
   根据evaluate\_object\_3d\_offline.cpp中 gtdir result\_dir的路径要求  
     
   创建文件夹  
   label\_2中存放kitti训练集label\_2  
   pred\_2文件夹中创建data文件夹，用于存放result
2. Generate Object Detection Results Using Ground Truth

# create object detection results from ground truth labels:

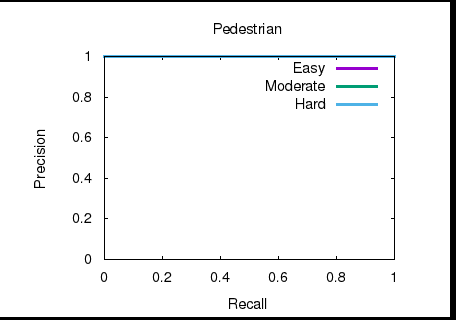
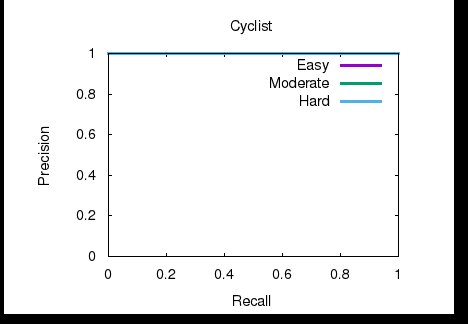
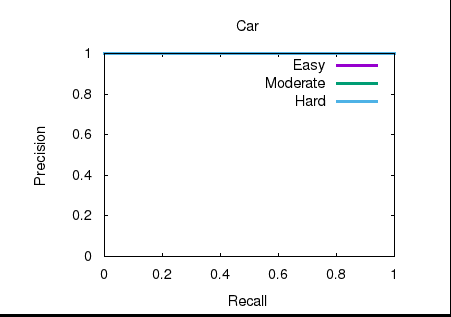
./create\_pred\_from\_ground\_truth.py -i /workspace/data/kitti-3d-object-detection/training/label\_2/ -o /workspace/data/kitti-3d-object-detection/training/pred\_2/

3.使用KITTI进行数据检测，观看标准

运行：

# run:

./kitti\_eval\_node /workspace/data/kitti-3d-object-detection/training/label\_2/ /workspace/data/kitti-3d-object-detection/

输出AP:  


具体代码如create\_pred\_from\_ground\_truth.py所示