**文件：**

lanenet-lane-detection-master：修改后的Lanenet模型

最终保存的模型在lanenet-lane-detection-master\model\pb\_model2\frozen\_model.pb

mydata： 训练数据

pcl\_test：对点云数据进行处理的代码，

random\_sample\_consensus---RANSAC提取平面

1554884725.127955000.pcd---原始点云

lane2.pcd---三维重建得到的车道线和道路边沿

combine.pcd---原始点云+车道线道路边沿点云

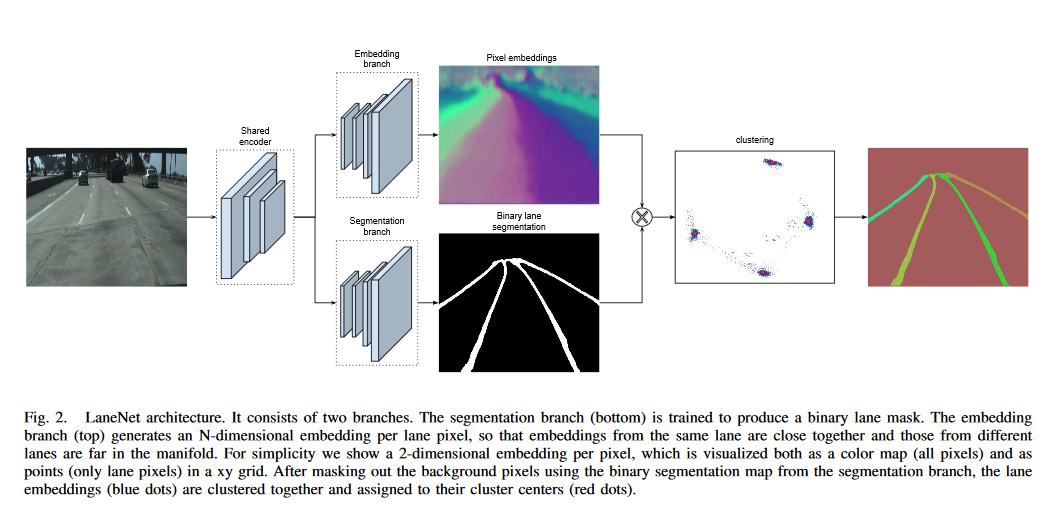
lanenet-c：c++实现Lanenet对车道线和道路边沿的检测

**I车道线和路沿检测**

**1. 使用 Lanenet模型**

https://github.com/MaybeShewill-CV/lanenet-lane-detection

该模型由编码解码、二值语义分割(cross-entropy loss)、实例语义分割(discriminative loss)三部分组成



网络分支

binary segmentation branch

input: raw image

output: binary image (255-lane;0-background)

embedding branch

input: raw image

output: feature map

instance segmentation

input: feature map \* binary image

output: cluster result

试验：使用清扫车采集的图像、原论文提供的训练结果

raw image



half image



denoised image



denoised output



**2. 修改网络以适用于车道线和道路边缘的同时检测**

将二值分割修改为3值 binary→3-value lane, background & roadside

修改的代码:

in lanenet-lane-detection-master

./data\_provider/lanenet\_data\_processor.py: #2019.4.4

./encoder\_decoder\_model/fcn\_decoder.py: #2019.4.4

./lanenet\_model/lanenet\_binary\_segmentation.py: #2019.4.4

./lanenet\_model/lanenet\_cluster.py: #2019.4.4

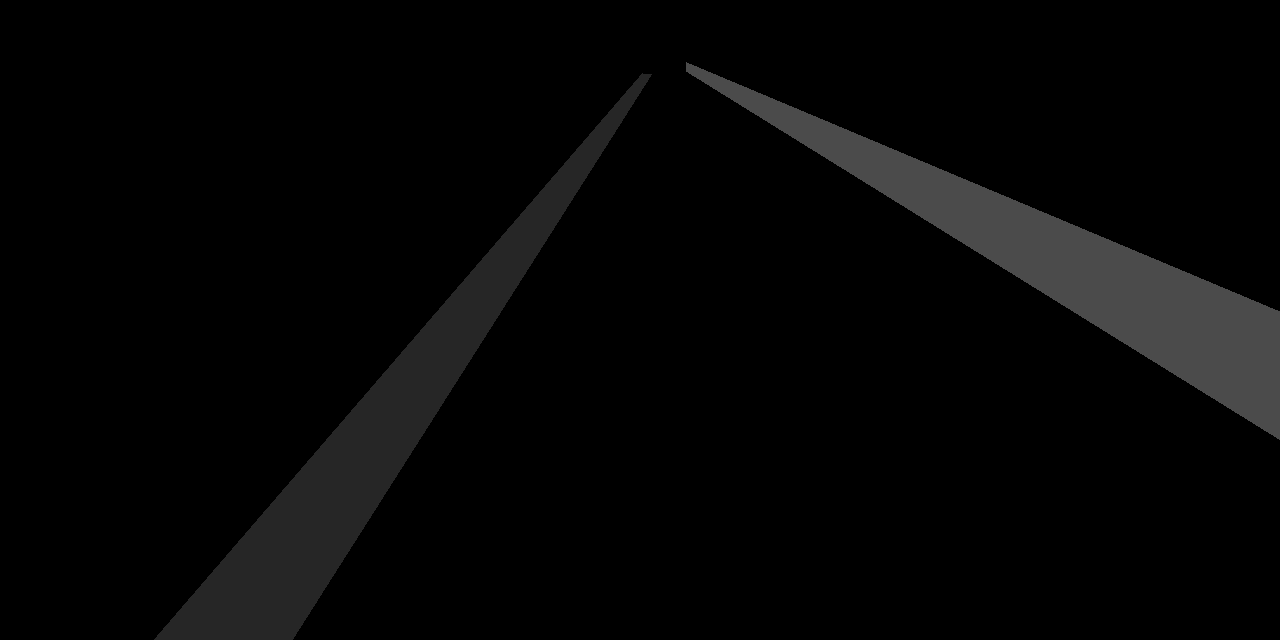
./lanenet\_model/lanenet\_cluster.py: #2019.4.4

./tools/test\_lanenet.py: #2019.4.4

训练数据格式:

gt\_image\_binary、gt\_image\_instance、image三个文件夹

gt\_image\_binary 车道线/道路边沿/背景 三值图



gt\_image\_instance 实例分割结果图(如果有 n 种车道线和m种道路边沿，实例语义分割图像为 m+n+1值图像)

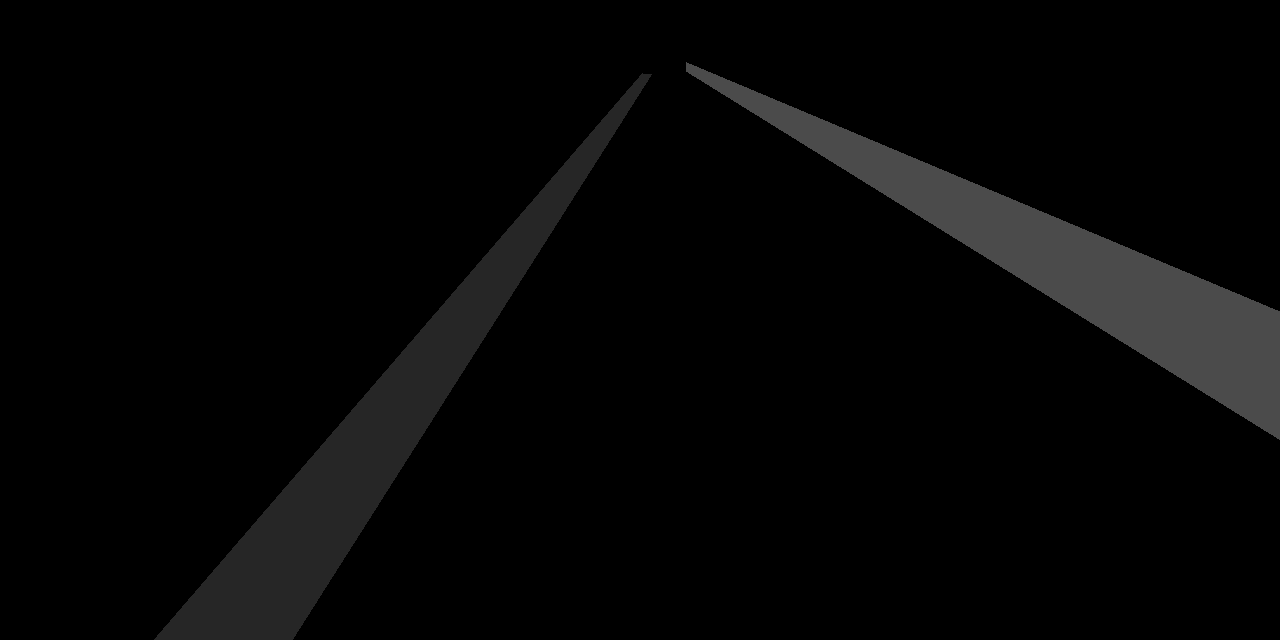
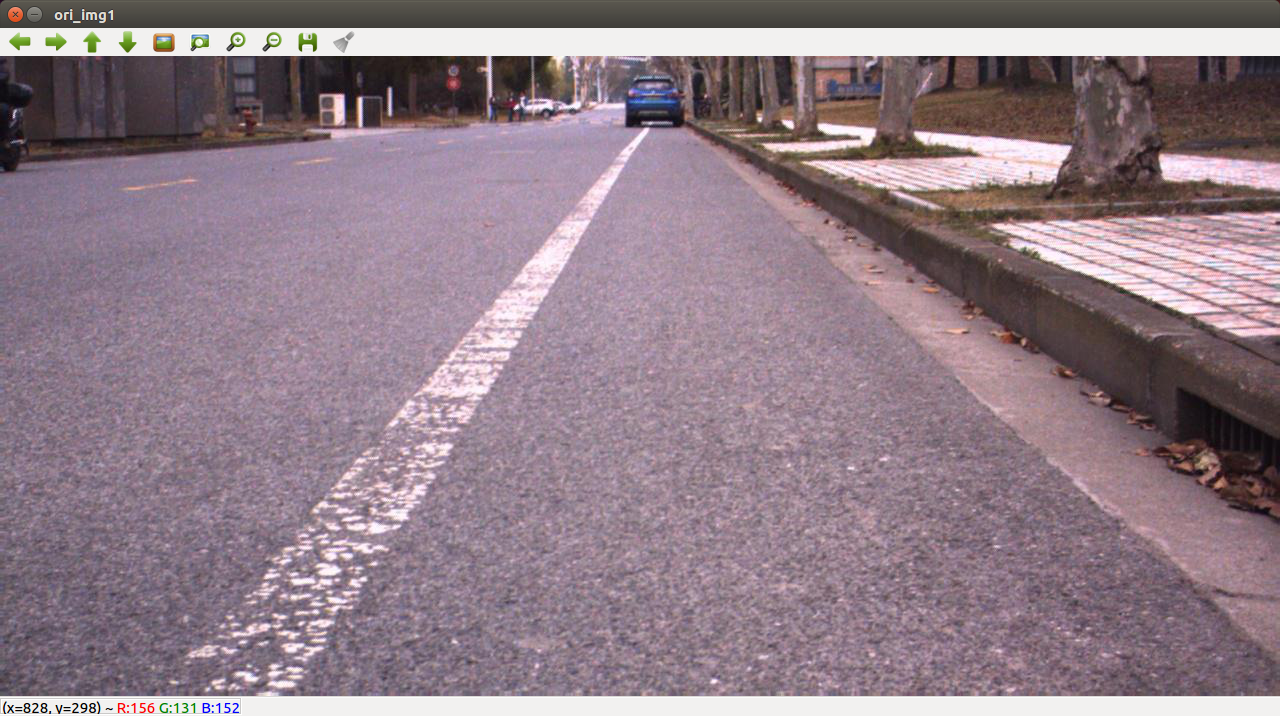


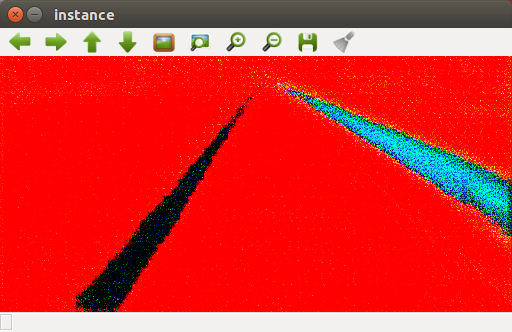
image: raw image(原图，将上部天空剪裁掉)

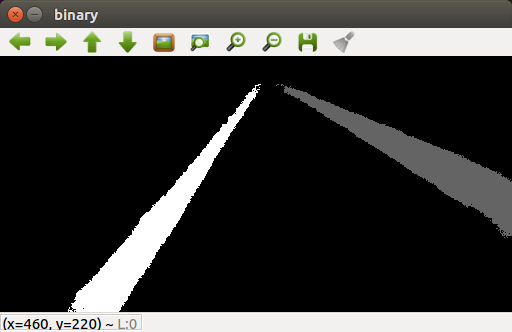


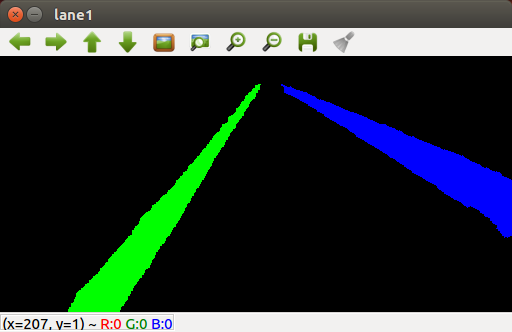
输出结果:

raw



 embedding result

3-value segmentation result

m+n+1-clustering result



保存模型步骤：

首先在 ./lanenet\_model/lanenet\_merge\_model.py中修改name:

binary\_seg\_ret = tf.argmax(binary\_seg\_ret, axis=-1, name='binary\_seg\_argmax')#2019.4.18

运行 ./tools/save\_model.py

运行 ./tools/freeze\_model.py

得到 frozen\_model.pb

**II 使用雷达数据和分割结果对车道线和道路边沿进行三维重建**

1. **理论部分**

① 从雷达点云数据中提取路面平面参数 （使用RANSAC算法）

pcl code: pcl\_test/random\_sample\_consensus.cpp

input: \*.pcd (radar cloud file)

output: plane coefficients A,B,C,D (Ax+By+Cz+D=0)

② 获得图像中的车道线和道路边沿点坐标

c++ code: lanenet-c/src/test\_lanenet

加载训练得到的frozen\_model.pb

test\_lanenet .cpp修改较多，未标注

LaneNetDetector.cpp中修改的代码标注了//2019.4.18

③ 获得车道线和道路边沿三维点坐标

原理：

雷达坐标系中平面方程A’x’+B’y’+C’z’+D’=0

相机坐标系中平面方程Ax+By+Cz+D=0

radar2camera 转换矩阵： T

计算相机坐标系中平面参数

平面约束下图像二维坐标→三维坐标

K为相机内参矩阵

1. **实现**

c++ code: lanenet-c/src/test\_lanenet

output: ane&roadside 3d point pcd file

show combine.pcd:



