ME5413-Homework Group number :16

LI BO : A0269522Y XUE HAOYAN: A0268454U 12/02/2023

Task 1

1a):

Since open3d facilitates the processing of pcd files, program bin2pcd.py comes to convert bin files to pcd files for subsequent processing.

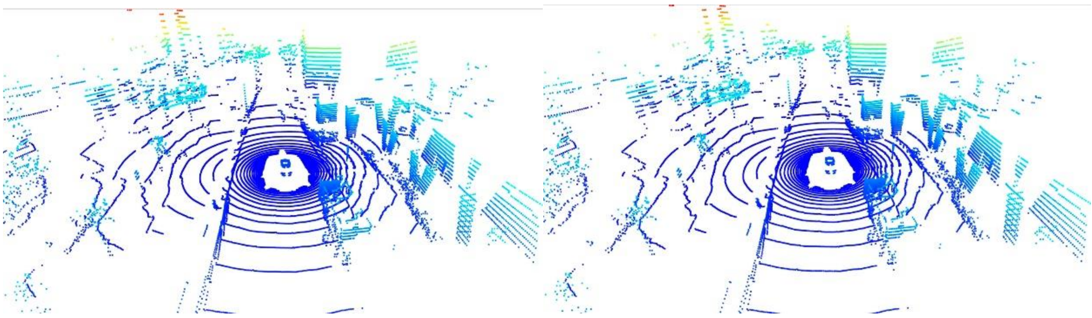


Figure1.1

The image on the left is the processed original point cloud image, and the one on the right is the down-sampled point cloud image. After down-sampling, the point cloud density becomes lower, which is helpful to improve the calculation speed and reduce the memory consumption.

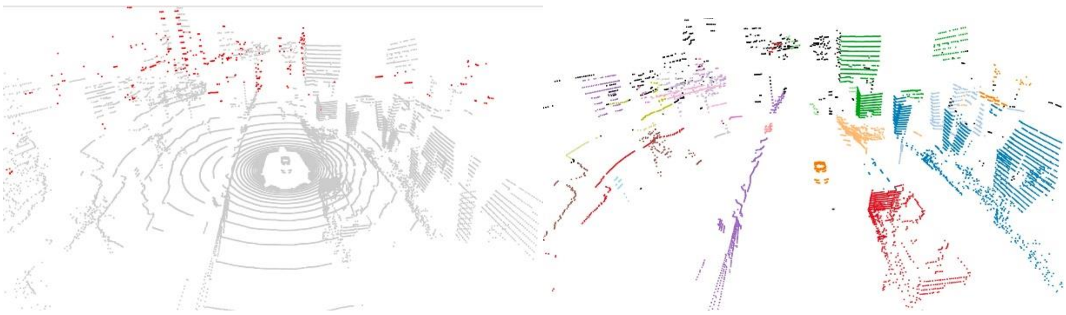


Figure1.2

The image on the left marks the outlier points as red and removes them. The point cloud ground is then removed in the program. After the above process, the point cloud is then clustered using the DBSCAN algorithm to obtain the results shown on the right.

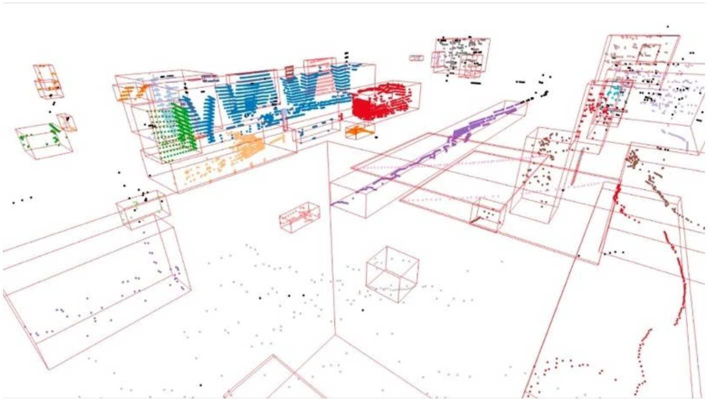


Figure 1.3 (parameters :eps=1.5 ,min_points=10)

The effect after the bounding box has been painted is shown in the picture(Using DBSCAN)

1b):

The K-means algorithm was chosen because it is a fast algorithm, suitable for large data sets, and because it is easier to choose the parameters by simply setting the number of clusters; in the K-Means clustering algorithm, the cluster centres represent the centre of each cluster and the points within each cluster are close to the centre, which is an intuitive and easy to understand structure. This structure is intuitive and easy to understand. It helps us to better understand the distribution pattern of the data and helps in decision making.

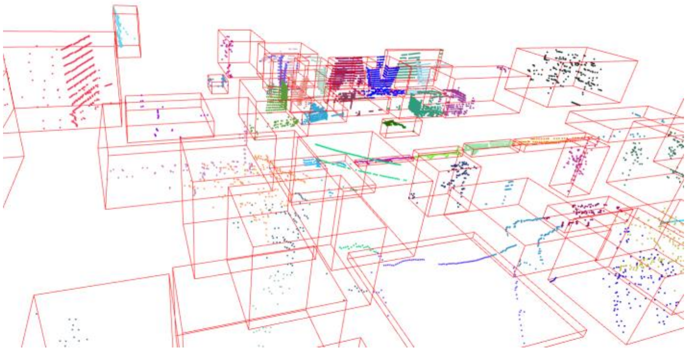


Figure 1.4 (parameters : n_clusters=60)

The effect after the bounding box has been painted is shown in the picture(Using K-means).

Task 2

2a);

Select one picture（Figure 2.1）with five labels in five pictures for analysis: according to the legend, the result of semantic segmentation of this pre-trained network can be seen.

Figure 2.1

Comparing the images before classification and after segmentation, we can find that the images after classification are excellent in finding out two kinds of things: drivable surface and others. However, the segmentation effect of this neural network is very poor for other objects like bicycles and motorcycles and pedestrians, which are relatively small in the figure. The net classifies bicycles and motorcycles as vehicles labels, and because the pedestrian is in front of the car, a large part of pedestrian is also attributed to the vehicles labels. The signboard in front of the vehicle is also included in the vehicle label.

2b):

Next, the quantitative analysis of the labeling effect of this picture is carried out:

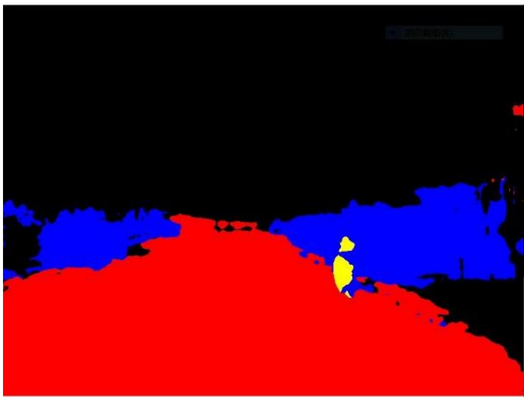


Figure2.2

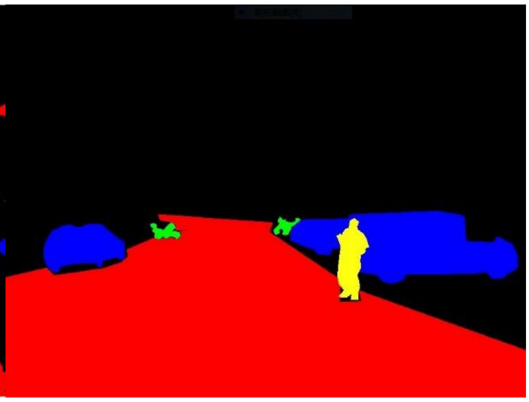


Figure2.3

The figure on the left is the labeling result obtained by pre-training network processing, and the figure on the right is the result of manual labeling. Compare the two figures with Matlab, the calculation results of relevant parameters are as follows:

```
>> judge
IOU = 0.805601
recall = 0.939737
precision = 0.849487
f1 = 0.892336
Dice = 0.892336
```

calculation results

Because the labels of drivable surface and others account for about 70% in the selected graph(photo2.jpg), and the pre-trained model performs well in identifying these two parts, the overall labeling results are not so bad.

2c):

There are some methods to improve the training model such as increasing the number of training data, it can improve the accuracy and robustness of the model. Or retraining network, using the pre-training model on other data sets as the initial model to speed up the training process and improve the accuracy. Or adjust the parameters to improve the training effect of the model such as learning rate.

We use Laplacian operator sharpening method to preprocess the image to enhance the image and improve the final classification effect. This method is more suitable when only the position of the edge is concerned, but the gray difference of pixels around it is not considered.The labeling results of the processed images are as follows:

Figure 2.4

It can be seen that after image preprocessing, the image segmentation greatly improves the marking of pedestrians, and correctly classifies billboards into others labels. But the marking of labels effect of vehicles is worse, and bicycles and motorcycles are still not marked. After image preprocessing, the pre-training network has enhanced its ability to mark near objects.

Task 3

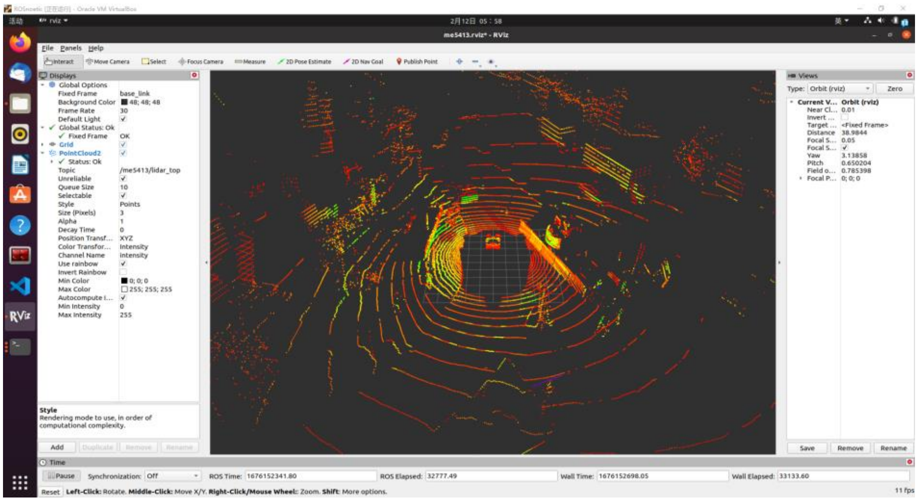
1. After downloading me5413_lidar.bag, open the terminal, run *Roscore*, and run *Rosbag Info* to view the information in the bag. You need to subscribe to the contents of */me5413/lidar_top* according to the requirements of the topic.

```
rosnoetic@rosnoetic-VirtualBox:~/homework_1/src$ rosbag info 'me5413_lidar.bag'
path:        me5413_lidar.bag
version:     2.0
duration:    19.1s
start:       Jul 24 2018 11:28:47.65 (1532402927.65)
end:         Jul 24 2018 11:29:06.80 (1532402946.80)
size:        206.5 MB
messages:    3345
compression: none [196/196 chunks]
types:       sensor_msgs/PointCloud2 [1158d486dd51d683ce2f1be655c3c181]
             tf/tfMessage            [94810edda583a504dfda3829e70d7eec]
topics:      /me5413/lidar_top     382 msgs    : sensor_msgs/PointCloud2
             /tf                  2963 msgs    : tf/tfMessage
rosnoetic@rosnoetic-VirtualBox:~/homework_1/src$
```

2. Open a separate terminal and start Rviz to visualize the data for easy observation of the processing. In the message shown above, you can see that the content of the sensor is Point Cloud2. At this time, you cannot subscribe to the topic content.

3. back to the first terminal, you can see that the file is about 19 seconds, for easy observation, use *Rosbag play -l* to loop through it.

4. Back to the space of Rviz, the topic of PointCloud2 can be selected manually at this time. Here the topic is selected, lidar_top, you can select Style as Points, and select global options . You can see the picture below. It can be clearly seen that it is a moving car and the scene scanned by LiDAR.



For the clustering of the data, our group has thought of two solutions, the first one is to use the point cloud clustering algorithms in the PCL library. We can use these algorithms to cluster the point cloud data and use the "Rosbag" tool in ROS to write the clustered point cloud data into a new bag file.

The second way is to use the functions in Ros to convert the bag file to a .pcd file, then use the clustering algorithm to process the pcd file, and then convert the processed result to a bag file for distribution. Since we were less familiar with the ROS environment, we spent a lot of time on the deployment of the environment.

The final algorithm we used was euclidean_cluster, which clusters point cloud data based on Euclidean distance. Approximate steps to handle point cloud data clustering.

Load point cloud data: Read point cloud data using PCL library.

Create the clustering object: Use pcl::EuclideanClusterExtraction to create the clustering object.

Set clustering parameters: Set the relevant parameters of the clustering algorithm, such as Euclidean distance threshold, minimum number of clusters, etc.

Cluster processing: Call the setInputCloud method of the clustering object to take the point cloud data as input, and call the extract method to perform the clustering process.

Get clustering result: Use the getClusters method of the clustering object to get the clustering result, which is an array of vector<PointIndices> type, where each array element corresponds to a cluster.

Save results: Save the clustering results to a new point cloud data or bag file. But in the end we cannot visualize the clustering results in RVIZ due to environmental issues