SDC Final Competition

0510727 張博凱, 0860078 黃瑞得, 0756172 鍾嘉峻

A. Introduction:

In this competition, we are asked to join one of the mainstream competition of tracking or prediction. Considering the deadline for each competition and our knowledge of all the fields, we choose the 3D tracking competition held by Argo in CVPR 2020 WAD workshop. In the 3D tracking task, we need to detect every object in the scene by giving the location the object we detect, and then combine the result of each timestamp together to track the object. You can see the example below(Figure.A). The input datasets which are provided by Argo contain several types of data, such as lidar point cloud, ring image, map, and etc. We may need to choose which data we want to use and design the method to achieve the goal.

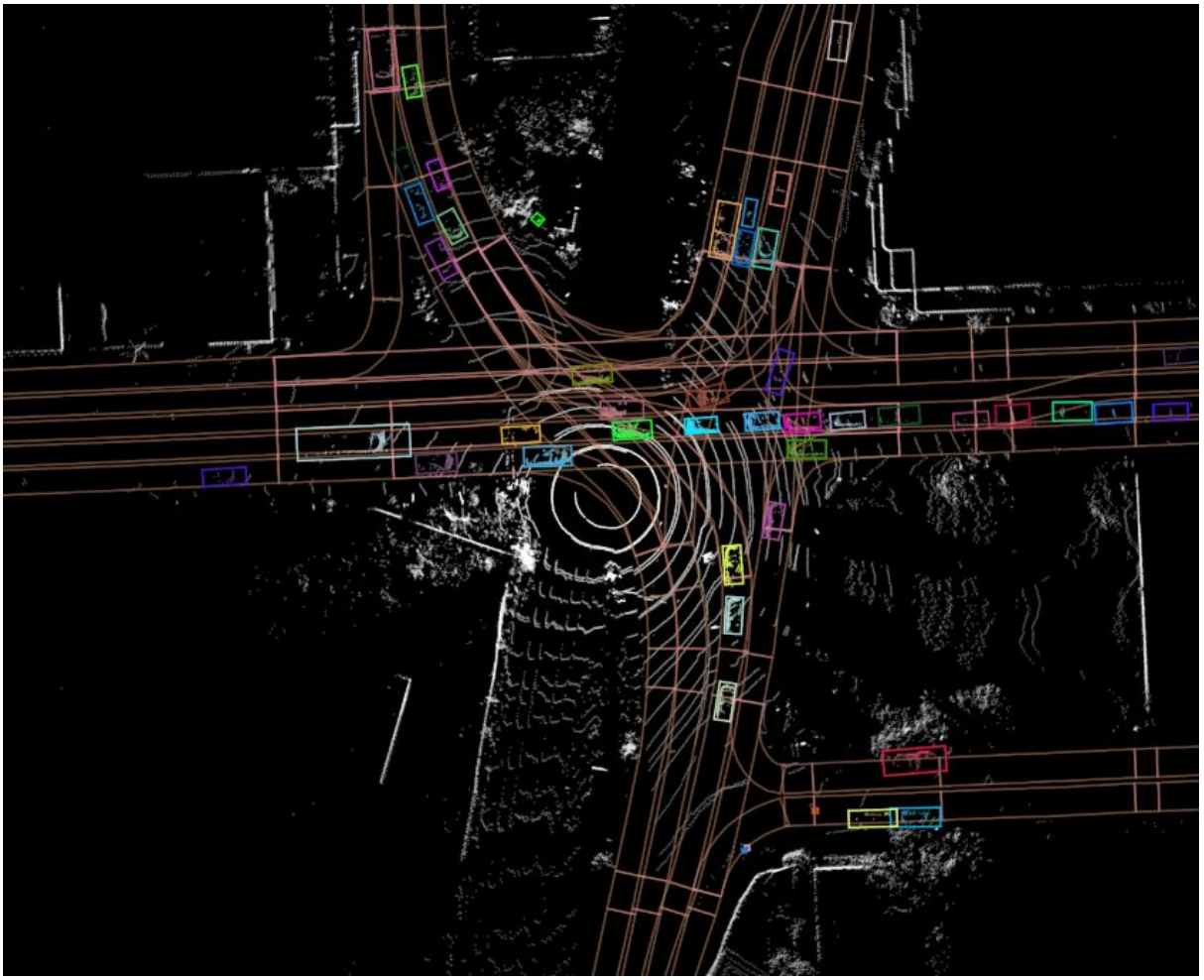

Figure.A Tracking Example

We sperate the workload to each member 張博凱 and 黃瑞得 try to use other data association method, and 鍾嘉峻 try to train the detection model. All team

members tune the system to have the best tracking result and write this report together.

B. Background:
   a. Dataset:
      There are several types of data, like Lidar, ring camera image, map and etc. You can see the explanation of the data below(Figure.B). We only use the Lidar data as detection model input.

Figure.B ARGO Dataset Generate Example

We will describe the detail of each type of data:

- Lidar sensor
  - 2 roof-mounted LiDAR sensors
  - Overlapping 40° vertical field of view
  - Range of 200m
  - On average, our LiDAR sensors produce a point cloud with ~ 107,000 points at 10 Hz

- Cameras
  - Seven high-resolution ring cameras (1920 x 1200) recording at 30 Hz with a combined 360° field of view
  - Two front-view facing stereo cameras (2056 x 2464) sampled at 5 Hz

- Localization
  - Argo uses a city-specific coordinate system for vehicle localization. We include 6-DOF localization for each timestamp, from a combination of GPS-based and sensor-based localization methods.
- Calibration

- Sensor measurements for each driving session are stored in “logs.” For each log, Argo provides intrinsic and extrinsic calibration data for LiDAR and all nine cameras.

b. CBGS detection result:
   - The detection result provided by the Argo. They adopt a famous model to train a detection model and produce the detection result for the competitor, which is based on the CVPR 2019 WAD nuscene detection challenge winner’s report paper, which’s called ***Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection***

C. Proposed Method:

a. Proposed Idea:
We based on the NeurIPS 2019 Argo winner’s architecture(Figure.C) The whole architecture can be depart as two parts, one is the 3D Object Detector, and the other is the Data Association part. In the 3D Object Detector part, the module will provided the detection results of each timestamp, and the data association part will take these detection result and combine them to achieve the tracking. We try two methods in the 3D detection module, one is the detection result provided by Argo, and the other is that we try to train a model by ourselves, which is based on PointRCNN. In the data association part, we try three different ways to measure the distance between two timestamps, one is 2d IOU, another is 3d IOU, and the other is Mahalanobis distance.

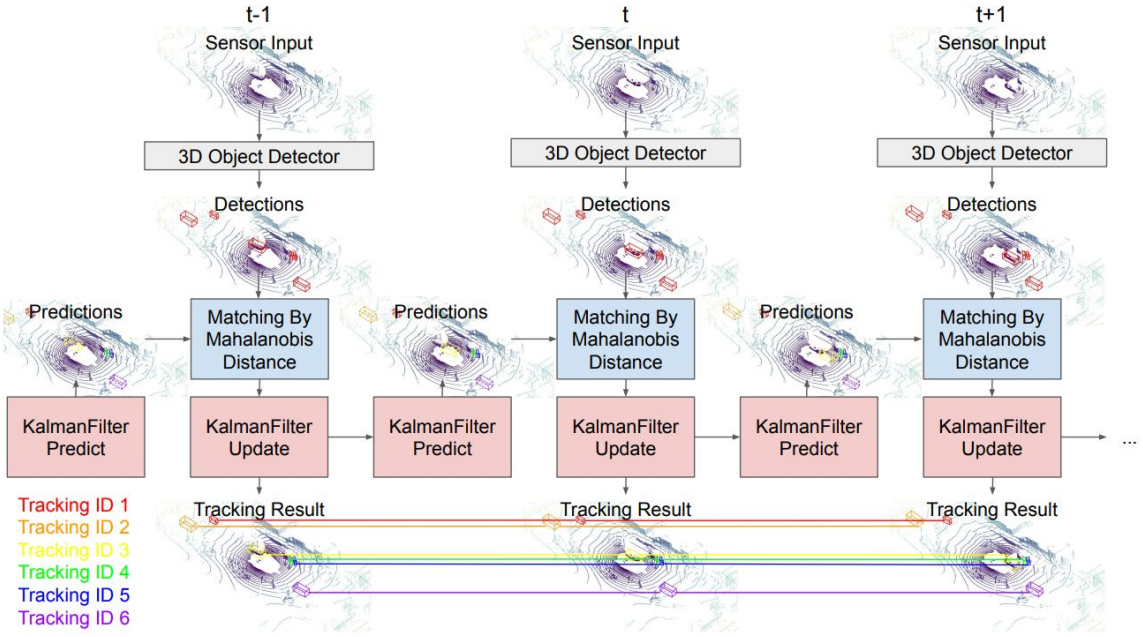

Figure.C The architecture we based on

b. Implementation:

Our code is mostly based on the code“https://github.com/johnwlambert/argoverse_cbgs_kf_tracker” released by Argo. And we will discuss our implementation from two-part:

1. Detection part:
   In this part we try to way to get our detection result, one is using the CBGS detection result provided by Argo, which means we only need to download the zip file form the competition page and use it, and the other is we want to try to train our own 3D detection model. The reason why we want to train our own model is that we think the key point of getting a better result of tracking is that you need to get a better detection first. You can the example below (Figure.C)which we open the detection result provided by Argo and use the visualization tool provided by TA. You can see the detection model misses the whole big bus in the middle of the image. In that case, no matter how good is our tacking module, we will never track this bus successfully.

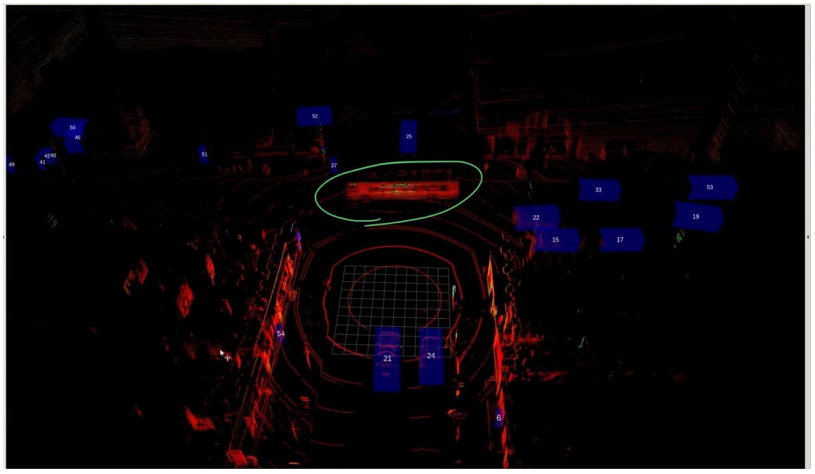

Figure.C CBGS Detection Fail Example

   So we use two-step to try to train a 3D detection model, which we want to base on a famous model, PointRCNN. PointRCNN is a famous model that has a nice detection result on the KITTI dataset leaderboard. So our strategy is first to convert the Argo dataset into KITTI dataset format, and then use the original official PointRCNN repo to train our model :

   a. Prepare the dataset:
      We first download the dataset from the Argo official website, then we convert the dataset to the KITTI dataset. We use the training dataset for training, using a validation dataset for testing, and using a test dataset for final detection performance testing.

b. Train the Model:
We modified the officially released code for training. Unfortunately, we got a bad result from our model. We only got AP = 0.36 when we threshold set to 0.7. It's too low for us, maybe we had some bug here, but due to the deadline, we cannot turn back for debugging and retraining the model. So we will use the detection result provided by Argo

2. Data association part:
In this part, we try three different methods to compute the distance between two detections:
   a. 2D - IOU:
   It's using by the baseline.
   b. 3D - IOU:
   Directly compute the Euclidean Distance distance between bounding boxes
   c. Mahalanobis distance(M_distance):
   The formula is listed below:

$$m = \sqrt{(\mathbf{o}_{t+1} - \mathbf{H}\hat{\mu}_{t+1})^T {\mathbf{S}_{t+1}}^{-1} (\mathbf{o}_{t+1} - \mathbf{H}\hat{\mu}_{t+1})}.$$

   This distance m measures the difference between predicted detections ***H*** and actual detections ***o*** weighted by the uncertainty about the prediction as expressed through the innovation covariance ***S***

D. Evaluation Metrics:
The total rank computes by the formula:
- **AVG-RANK**: Average ranking (updating ever hour). We calcuclated ranking over each metrics and average them using (rMOTA)/2 + (rMOTP + rMOTP-D + rMOTP-O + rMOTP-I)/16 + rIDF1/2 + (rMT + rML + rFP + rFN + rSW + rFRG)/24

The formula combine with those scores:
1. **MOTA,** We use the bounding box centroid distance between tracker output and ground truth as detection range (the threshold for the missed track at 2 meters, which is around half of an average family car length in US).
2. **MOTP**, We use three distance metrics for MOTP

3. **MOTP-D**, (centroid distance) the bounding box centroid distance same as MOTA detection range
4. **MOTP-O**, (orientation error) the smallest angle difference about z (vertical) axis
5. **MOTP-I**, (Intersection-over-Union error) the amodal shape estimation error, computed by the 1-IoU of 3D bounding box projections on XY plane after aligning orientation and centroid
6. **IDF1**: F1 score, denotes as 2(precision * recall)/(precision+recall)
7. **MT (Mostly Tracked)**: the ratio of trajectories tracked more than 80% of its lifetime.
8. **ML (Mostly Lost)**: the ratio of trajectories tracked for less than 20% of object
9. **FP**: Total number of false positives
10. **FN**: Total number of false negatives
11. **SW**: number of identity ID switches.
12. **FRG**: Total number of switches from "tracked" to "not tracked"

E. Result & Discussion:
So we try to combine the CBGS detection result and the three different methods to compute the distance, see the Appendex:

F. Conclusion:
In the end, we beat the baseline just a little bit. We believe that the reason here is because we cannot train the detection model successfully. We also compare the result by using two computing methods in the data association part, and we found we will get the best result when using M_distance.

G. Appendex:

	2D-IOU	M_distance
C:MOTA	65.86	0
P:MOTA	48.31	0
C:MOTPD	0.34	0.37
P:MOTPD	0.37	0.64
C:MOTPO	15.76	16.45
P:MOTPO	24.16	51.22
C:MOTPI	0.2	0.21
P:MOTPI	0.18	0.32
C:IDF1	0.79	0.49
P:IDF1	0.58	0.4

C:MT	0.51	0.49
P:MT	0.28	0.38
C:ML	0.21	0.19
P:ML	0.31	0.24
C:FP	15,715.00	130,538.00
P:FP	4,933.00	37,342.00
C:FN	23,594.00	25,017.00
P:FN	25,780.00	22,307.00
C:SW	221	1,031.00
P:SW	424	1,688.00
C:FRG	393	1,586.00
P:FRG	387	1,353.00
C:MT-OCC	0.42	0.42
C:MT-FAR	0.17	0.14
C:ML-OCC	0.09	0.08
C:ML-FAR	0.53	0.51
P:MT-OCC	0.4	0.55
P:MT-FAR	0.01	0.01
P:ML-OCC	0.12	0.07
P:ML-FAR	0.73	0.73
AVG-RANK	10.64	8.58