

CSC490 Write Up

Part 1. Object Tracking Questions

Q2.1

- a) An example cost matrix where the greedy algorithm does not lead to optimal total cost:

0.4	0.5	0.3
0.5	0.3	0.1
0.2	0.1	0.3

The Greedy algorithm would

- 1) Pick 3 in the first row, eliminate the third column
- 2) Pick 3 in the second row eliminate the second column
- 3) Pick 2 in the third row

This gives us the total cost of $0.3 + 0.3 + 0.2 = 0.8$

However the optimum case is actually

- 1) Picking 4 in the first row, eliminate first column
- 2) Picking 1 in the second row, eliminate third column
- 3) Picking 1 in the third row

This gives us the total cost of $0.4 + 0.1 + 0.1 = 0.6$

- b) A scenario with two sets of bounding boxes where the resulting cost matrix does not lead to optimal assignment with the greedy algorithm

There two bounding boxes will be

A: {Bounding Box 1, Bounding Box 2, Bounding Box 3}

B: {Bounding Box 4, Bounding Box 5, Bounding Box 6}

Where the cost matrix provided above is the iou scores between the different combinations of bounding boxes

Q2.3

Hungarian: Final evaluation metric results for each of the 12 validation logs

0%| | 0/12 [00:00<?, ?it/s][Sequence: 002]
EvaluationResult(mota=0.47574626865671643, motp=0.5720018896444496,

mostly_tracked=0.6761133603238867, mostly_lost=0.30364372469635625,
 partially_tracked=0.020242914979757054)
 8%|██████████ | 1/12 [00:02<00:31, 2.85s/it][Sequence: 003]
 EvaluationResult(mota=0.3520481237467774, motp=0.5691669916774194,
 mostly_tracked=0.7560975609756098, mostly_lost=0.24390243902439024, partially_tracked=0.0)
 17%|██████████ | 2/12 [00:04<00:19, 1.98s/it][Sequence: 004]
 EvaluationResult(mota=0.257529463116543, motp=0.4848192464529397,
 mostly_tracked=0.34108527131782945, mostly_lost=0.6124031007751938,
 partially_tracked=0.04651162790697683)
 25%|██████████ | 3/12 [00:05<00:16, 1.84s/it][Sequence: 005]
 EvaluationResult(mota=0.15724703737465817, motp=0.6282787261346726,
 mostly_tracked=0.34269662921348315, mostly_lost=0.6404494382022472,
 partially_tracked=0.016853932584269593)
 33%|██████████ | 4/12 [00:07<00:13, 1.72s/it][Sequence: 017]
 EvaluationResult(mota=0.5284132841328413, motp=0.6533165449234641,
 mostly_tracked=0.6716417910447762, mostly_lost=0.31716417910447764,
 partially_tracked=0.01119402985074619)
 42%|██████████ | 5/12 [00:09<00:14, 2.01s/it][Sequence: 019]
 EvaluationResult(mota=0.375, motp=0.6163033797100557, mostly_tracked=0.521551724137931,
 mostly_lost=0.47413793103448276, partially_tracked=0.004310344827586188)
 50%|██████████ | 6/12 [00:11<00:11, 1.94s/it][Sequence: 021]
 EvaluationResult(mota=0.24525085241110567, motp=0.6150086168082591,
 mostly_tracked=0.7658227848101266, mostly_lost=0.20253164556962025,
 partially_tracked=0.031645569620253194)
 58%|██████████ | 7/12 [00:15<00:12, 2.43s/it][Sequence: 028]
 EvaluationResult(mota=0.40312499999999996, motp=0.6923336873802782,
 mostly_tracked=0.6197183098591549, mostly_lost=0.36619718309859156,
 partially_tracked=0.014084507042253558)
 67%|██████████ | 8/12 [00:17<00:10, 2.55s/it][Sequence: 032]
 EvaluationResult(mota=0.23974445191661065, motp=0.6631157888715614,
 mostly_tracked=0.5670731707317073, mostly_lost=0.4024390243902439,
 partially_tracked=0.030487804878048808)
 75%|██████████ | 9/12 [00:20<00:07, 2.40s/it][Sequence: 033]
 EvaluationResult(mota=0.26314575645756455, motp=0.630889413381609,
 mostly_tracked=0.6462264150943396, mostly_lost=0.32547169811320753,
 partially_tracked=0.028301886792452824)
 83%|██████████ | 10/12 [00:23<00:05, 2.78s/it][Sequence: 034]
 EvaluationResult(mota=0.32564037534871926, motp=0.6794239757397786,
 mostly_tracked=0.5786163522012578, mostly_lost=0.3836477987421384,
 partially_tracked=0.037735849056603765)
 92%|██████████ | 11/12 [00:25<00:02, 2.54s/it][Sequence: 035]
 EvaluationResult(mota=0.2681650246305419, motp=0.6093248317043642,
 mostly_tracked=0.5514018691588785, mostly_lost=0.4392523364485981,
 partially_tracked=0.009345794392523421)
 100%|██████████ | 12/12 [00:28<00:00, 2.37s/it]

Hungarian: Median and mean results for all 12 validation logs

[Results (mean) EvaluationResult(mota=0.3242546364826732, motp=0.617831924369071, mostly_tracked=0.5865037699057484, mostly_lost=0.3926033749332956, partially_tracked=0.02089285516095595)
[Results (median) EvaluationResult(mota=0.2969026999896306, motp=0.6222910529223642, mostly_tracked=0.5991673310302064, mostly_lost=0.374922490920365, partially_tracked=0.018548423782013324)

Greedy: Final evaluation metric results for each of the 12 validation logs

```

0%|██████████| 0/12 [00:00<?, ?it/s][Sequence: 002]
EvaluationResult(mota=0.44296375266524524, motp=0.5715834404929407,
mostly_tracked=0.7182044887780549, mostly_lost=0.2718204488778055,
partially_tracked=0.00997506234413964)
8%|███████| 1/12 [00:03<00:40, 3.64s/it][Sequence: 003]
EvaluationResult(mota=0.33056430822114, motp=0.5692712606873683,
mostly_tracked=0.8333333333333334, mostly_lost=0.16666666666666666,
partially_tracked=-2.7755575615628914e-17)
17%|██████████| 2/12 [00:05<00:27, 2.73s/it][Sequence: 004]
EvaluationResult(mota=0.24268878219118284, motp=0.4844891222665807,
mostly_tracked=0.3969849246231156, mostly_lost=0.5728643216080402,
partially_tracked=0.03015075376884413)
25%|██████████| 3/12 [00:07<00:22, 2.48s/it][Sequence: 005]
EvaluationResult(mota=0.14220601640838648, motp=0.6282787261346726,
mostly_tracked=0.3952569169960474, mostly_lost=0.5889328063241107,
partially_tracked=0.015810276679841917)
33%|██████████| 4/12 [00:09<00:17, 2.24s/it][Sequence: 017]
EvaluationResult(mota=0.48856088560885613, motp=0.648127580232876,
mostly_tracked=0.7398720682302772, mostly_lost=0.24946695095948826,
partially_tracked=0.01066098081023456)
42%|██████████| 5/12 [00:13<00:19, 2.72s/it][Sequence: 019]
EvaluationResult(mota=0.3461895910780669, motp=0.6128211813143862,
mostly_tracked=0.5761194029850746, mostly_lost=0.417910447761194,
partially_tracked=0.005970149253731405)
50%|██████████| 6/12 [00:15<00:14, 2.43s/it][Sequence: 021]
EvaluationResult(mota=0.22308816366293227, motp=0.6071815231621085,
mostly_tracked=0.8166089965397924, mostly_lost=0.1695501730103806,
partially_tracked=0.013840830449827035)
58%|██████████| 7/12 [00:18<00:13, 2.67s/it][Sequence: 028]
EvaluationResult(mota=0.3880681818181818, motp=0.6930725931283851,
mostly_tracked=0.6619718309859155, mostly_lost=0.3145539906103286,
partially_tracked=0.023474178403755874)
67%|██████████| 8/12 [00:20<00:10, 2.60s/it][Sequence: 032]
EvaluationResult(mota=0.2259583053127101, motp=0.6640394767257336,
mostly_tracked=0.6044444444444445, mostly_lost=0.37777777777777777,
partially_tracked=0.01777777777777778)
75%|██████████| 9/12 [00:22<00:07, 2.45s/it][Sequence: 033]
EvaluationResult(mota=0.24630996309963105, motp=0.633005121040128,
mostly_tracked=0.6796116504854369, mostly_lost=0.30097087378640774,
partially_tracked=0.01941747572815533)

```

83% [REDACTED] | 10/12 [00:27<00:06, 3.03s/it][Sequence: 034] EvaluationResult(mota=0.311437991377124, motp=0.6794239757397786, mostly_tracked=0.648068669527897, mostly_lost=0.33905579399141633, partially_tracked=0.012875536480686678)
92% [REDACTED] | 11/12 [00:29<00:02, 2.79s/it][Sequence: 035] EvaluationResult(mota=0.24353448275862066, motp=0.6083505996875997, mostly_tracked=0.6067073170731707, mostly_lost=0.38109756097560976, partially_tracked=0.012195121951219523)
100% [REDACTED] | 12/12 [00:32<00:00, 2.72s/it]

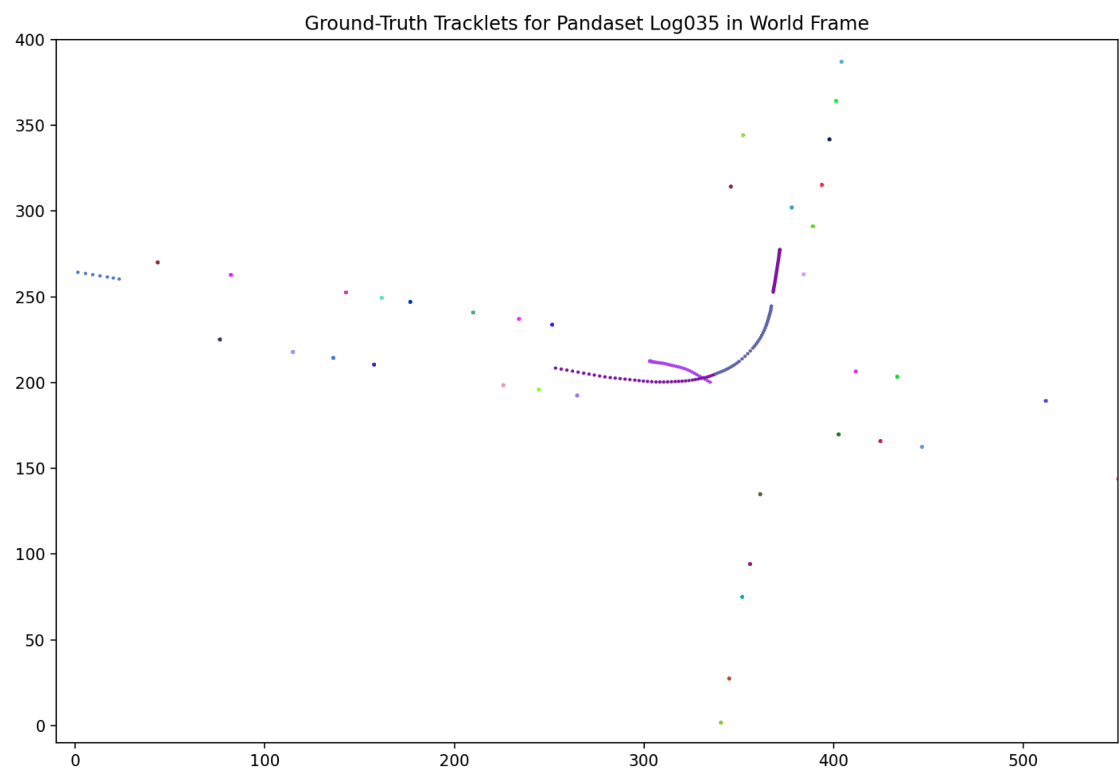
Greedy: Median and mean results for all 12 validation logs

[Results (mean) EvaluationResult(mota=0.30263086868350647, motp=0.6166370500510465, mostly_tracked=0.6397653370002133, mostly_lost=0.34588898436243554, partially_tracked=0.014345678637351153)
[Results (median) EvaluationResult(mota=0.27887397723837754, motp=0.6205499537245294, mostly_tracked=0.6550202502569062, mostly_lost=0.3268048923008725, partially_tracked=0.013358183465256857)

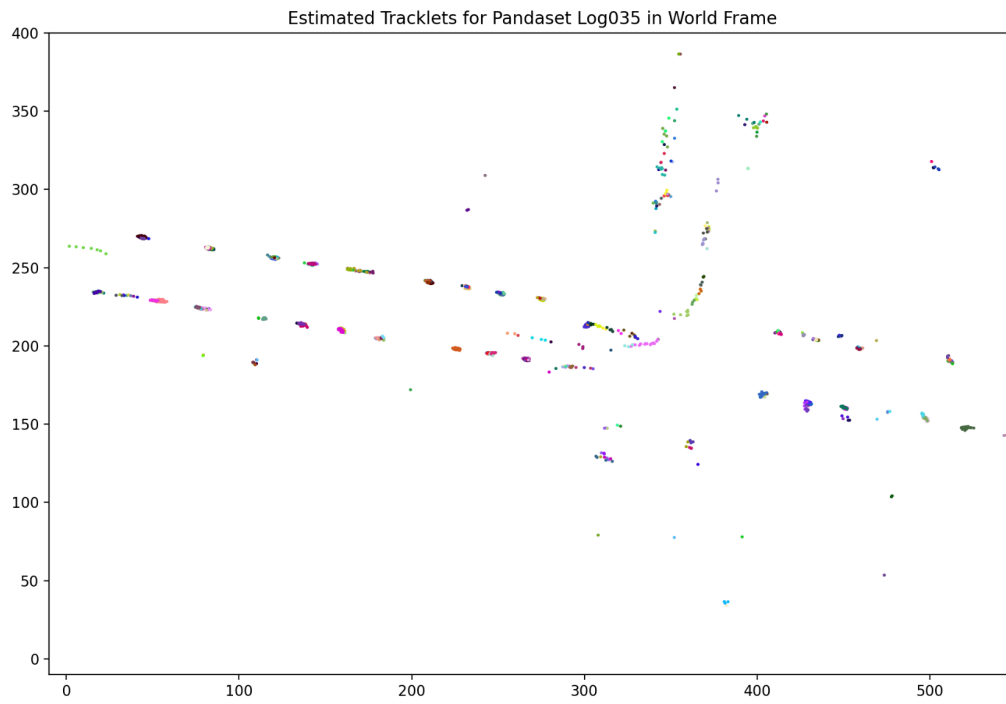
How does the greedy association compare with the Hungarian association on our dataset?

The Hungarian association does better on our dataset with higher MOTA and MOTP mean results for the 12 validation logs. For example in sample #35 (visualized below), the Hungarian algorithm is able to match the detections for the same actor (the vehicle that's turning) across the frames. In comparison, the greedy matching establishes new tracklets for the same vehicle.

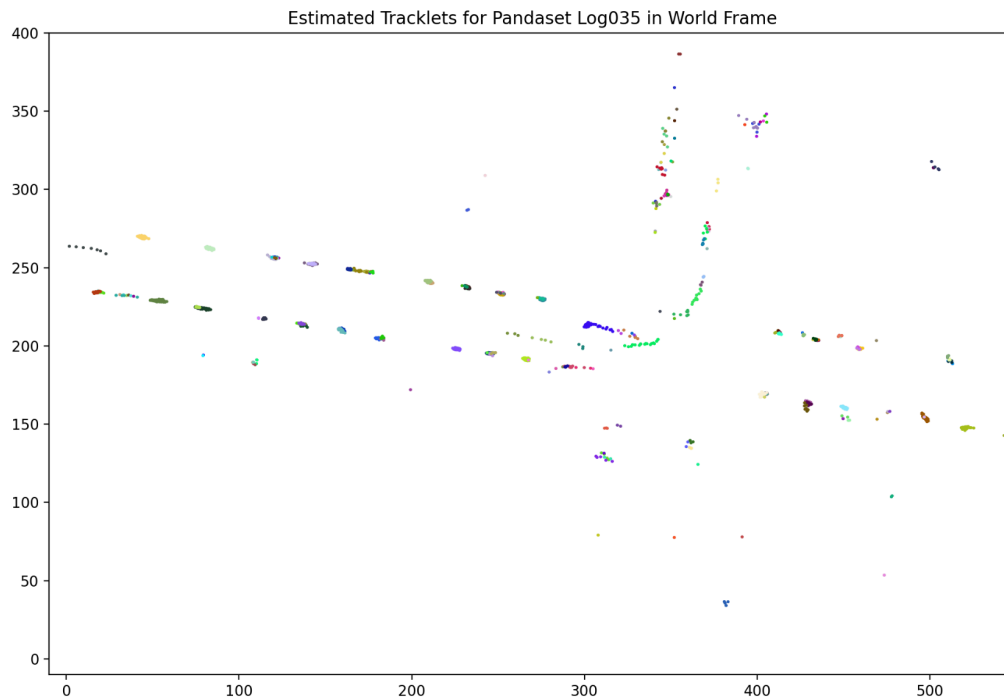
Ground truth:



Greedy:



Hungarian:



Part 2. Improved Object Tracking Report:

Using more sophisticated cost functions

Motivation

What is the problem being tackled

The problem that we are interested in tackling is to come up with a more sophisticated cost function such that the tracking accuracies can be improved. Our hypothesis is that, suppose we have two vehicles in the original dataset that were very close to each other, it would be difficult for the model to distinguish them and track them properly. Another issue that might also arise when we use IOU to track vehicles, is the failure to track the vehicle after it has been occluded for some periods of time. Therefore, we would like to do some research and find out other ways to measure the costs that can solve this issue.

Why is this problem relevant

This problem is relevant because from some of the visualization results, we can see that routes sometimes overlap with each other, which implies there might be cases like occlusion or

vehicles being very close to each other. For cases like that, as explained above, our original tracking algorithm would not be able to perform well. Therefore, we think that if our new cost functions are implemented successfully, these problems could be taken care of, which will ultimately result in an increase in the final tracking accuracy.

Briefly explain other approaches to this problem and why you chose your approach

Apart from the motion feature and the geometry distance approach that we are using, there are also several other ways to tackle this problem, for example, measuring the distance between the feature embeddings generated from the detection model. The feature embeddings can be obtained from the second last layer from the model, and we can compute the distance between the embeddings using L2-norms. Using this deep neural network feature as the cost might be better than IoU because the features might contain more information.

Another way to solve this problem is to also add visual information into our model. [2] For example, we might also be able to track the vehicles using their special features such as colors or car plates. Adding visual information on top of data collected by the lidar sensor would definitely give us higher accuracy, since there is more data available. However, this would also require high computation power, and long processing time. Therefore, it might not be a good solution if we want everything to be running in real time.

Techniques

Describe the intuition for why your approach would help

We experimented with two ways of solving the problem. The first loss function is to measure the geometric loss between the two bounding boxes. We measured the difference in keypoints, the difference in sizes and the differences in headings. Our intuition of why this method could help is because we think that this loss function gives us higher flexibility to tune the weights of all the attributes that account for the intersection of bounding boxes. For example, we can assign a higher weight to the difference in sizes, which tells the system that if the sizes between the two boxes are not the same, a higher loss should be returned. This is because the size of the vehicle at two consecutive frames should be relatively the same. With this example, it might help to distinguish vehicles that are different in sizes but close together.

The second loss function that we have attempted is the motion feature loss, where we measure the displacement difference. If the displacements between bounding box 1 and the previous bounding boxes of the assigned tracklet has a huge difference from the displacement between bounding box 1 and 2, then it is likely that they are not the same vehicles, as the displacements should be relatively the same in consecutive frames. There shouldn't be drastic changes in displacements from one frame to the next frame, as each frame is taken at the interval of a few

seconds. That said, we think that this might help us to identify vehicles that move at different speeds.

Articulate your approach mathematically / algorithmically

Geometric Cost:

- Keypoint

$$\hat{\mathbf{t}} = \left(\frac{\hat{p}_1 - x_1}{x_2 - x_1}, \frac{\hat{p}_2 - y_1}{y_2 - y_1} \right).$$

$\hat{\mathbf{t}}$ here refers to the keypoints of the bounding box 1, where (\hat{p}_1, \hat{p}_2) is the centroid and $(x_1, y_1), (x_2, y_2)$ refers to the top left and bottom right coordinates of the box.

Similarly, we computed keypoints \mathbf{t} for bounding box 2.

Then, we obtain the L1 loss for the bounding boxes' keypoints.

$$L_{kp} = L_1(\hat{\mathbf{t}}, \mathbf{t})$$

- Yaw:

We obtain the L1 loss between the yaw of the first bounding box and that of the second bounding box.

- Size:

We obtain the size of the bounding boxes by multiplying the length by the width. Then after computing the sizes of the two bounding boxes, we measure the L1 loss between the sizes of the two bounding boxes.

$$\text{Total Loss: } L = \lambda_{kp} L_{kp} + \lambda_{size} L_{size} + \lambda_{yaw} L_{yaw}$$

After obtaining the losses for yaw, keypoints and sizes, we compute the weighted sum of these losses. The values of lambda that we tested were:

[lam_t = 5, lam_size = 3, lam_yaw = 5], and

[lam_t = 5, lam_size = 1, lam_yaw = 3].

Reference: (Shi et al.) Geometry-based Distance Decomposition for Monocular 3D Object Detection

https://openaccess.thecvf.com/content/ICCV2021/papers/Shi_Geometry-Based_Distance_Decomposition_for_Monocular_3D_Object_Detection_ICCV_2021_paper.pdf

Motion Feature Cost:

Let B_{t-1} is the bounding box in the previous frame and B_t is the bounding box in the current frame. The cost algorithm will follow the steps below.

1. Find the associated tracklet of the actor detected in bbox B_{t-1} and obtain the tracklet's second last item for B_{t-2} . Calculate the displacement D_1 between bounding boxes B_{t-1} and B_{t-2} .
2. Calculate the displacement D_2 between bbox B_{t-1} and bbox B_t .
3. Calculate $\text{np.linalg.norm}(D_1 - D_2)$ as the motion cost

We used the Hungarian matching algorithm for all our experiments in this section.

Evaluation

Articulate your evaluation plan mathematically / algorithmically

Experiment 1: Optimizing performance of Geometric Cost Function

Step 1: Run the model with different hyperparameter sets ([lam_t = 5, lam_size = 3, lam_yaw = 5], [lam_t = 5, lam_size = 1, lam_yaw = 3])

Step 2: Run the model with different ways of normalization: dividing values for each row by the max value of each row versus dividing the maximum value of the matrix.

Step 3: After running the model with selected hyperparams and methods of normalizing, we evaluate the model performance by the resulting MOTA, MOTP, Mostly tracked, and mostly loss values.

Step 4: We consider the model with the highest MOTA, highest MOTP, highest mostly tracked and lowest mostly loss values to be the best model.

Experiment 2: Optimizing performance of Motion Feature Cost Function

Step 1: Run the model with different ways of normalization: dividing values for each row by the max value of each row versus dividing the maximum value of the matrix

Step 2: After running the model with selected hyperparams and methods of normalizing, we evaluate the model performance by the resulting MOTA, MOTP, Mostly tracked, and mostly loss values.

Step 3: We consider the model with the highest MOTA, highest MOTP, highest mostly tracked and lowest mostly loss values to be the best model.

Experiment 3: Combining IOU cost functions and the cost function that has a better performance

Step 1: Run the model with $0.5 * \text{IOU loss} + 0.5 * \text{motion feature loss}$

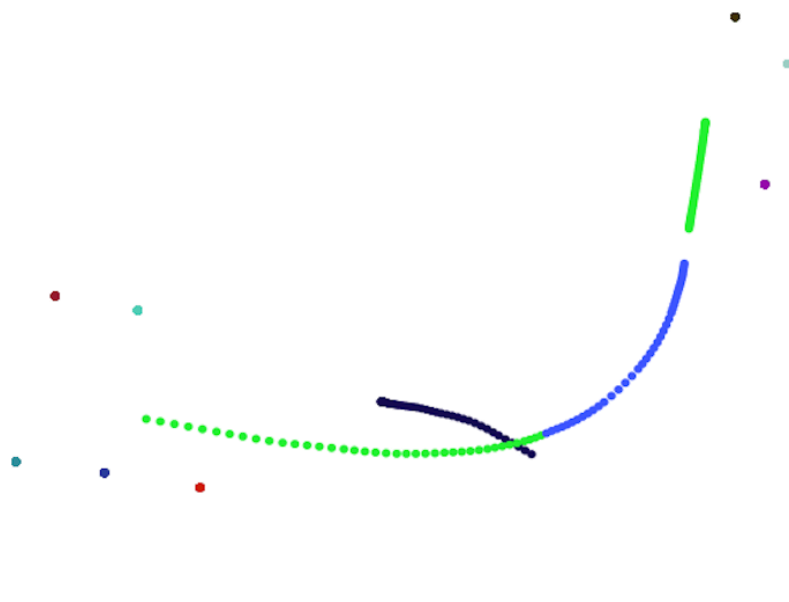
Step 2: After running the model with selected hyperparams and methods of normalizing, we evaluate the model performance by the resulting MOTA, MOTP, Mostly tracked, and mostly loss values.

Step 3: We consider the model with the highest MOTA, highest MOTP, highest mostly tracked and lowest mostly loss values to be the best model.

Show compelling results

Example using Log#35

Ground truth:



Part 1 baseline tracking estimation using IoU cost:



	MOTA	MOTP	Mostly tracked	Mostly lost
Part 1 baseline	0.3243	0.6178	0.5865	0.3926

Tracking estimation using geometric 2D cost:

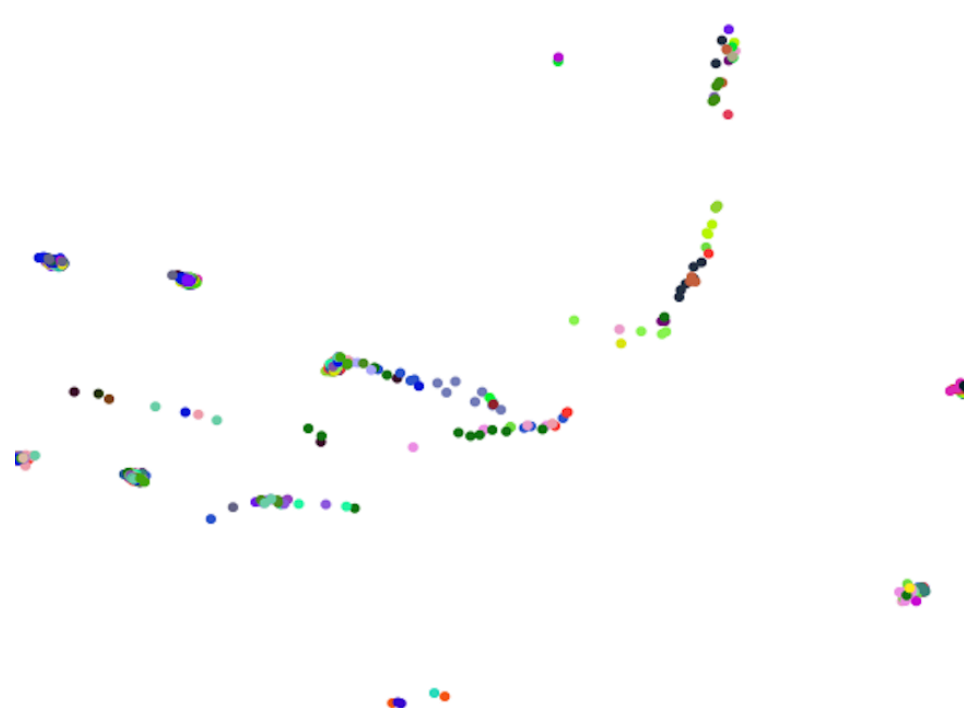


Figure A. raw geom cost [lam_t = 5, lam_size = 3, lam_yaw = 5]	Figure B. cost_geom_normalized_lambda_tuned

Geometric cost mean metrics across the 12 sequences:

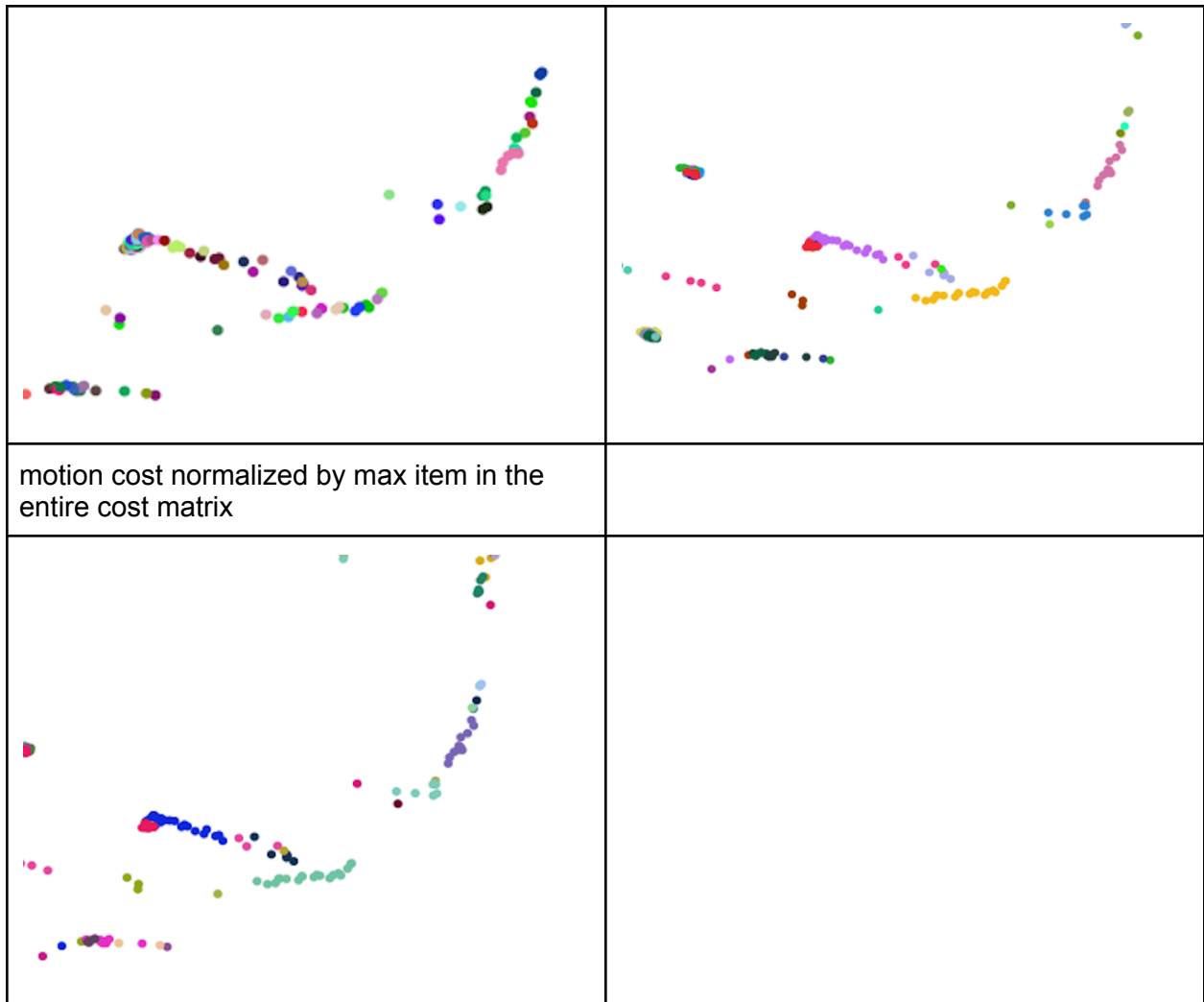
	MOTA	MOTP	Mostly Tracked	Mostly Lost
1: raw	-0.0408	0.6203	0.8135	0.1746
2: normalized by row max	0.0778	0.6163	0.6084	0.0967
3: normalized by max matrix item	0.0795	0.6168	0.6095	0.1070
4: normalized & lambda tuned	0.0871	0.6161	0.6074	0.0958

We see that after normalization and tuning weights for the geometric cost function, there is less fragmentation in the tracklets (e.g. the dark green dots in Figure B illustrate a longer tracklet than in Figure A). The decrease in tracklet fragmentation is also shown by the higher proportion of Mostly Tracked tracklets and lower proportion of Mostly Lost tracklets.

Tracking estimation using motion feature cost:



raw motion cost	motion cost normalized by row max
-----------------	-----------------------------------

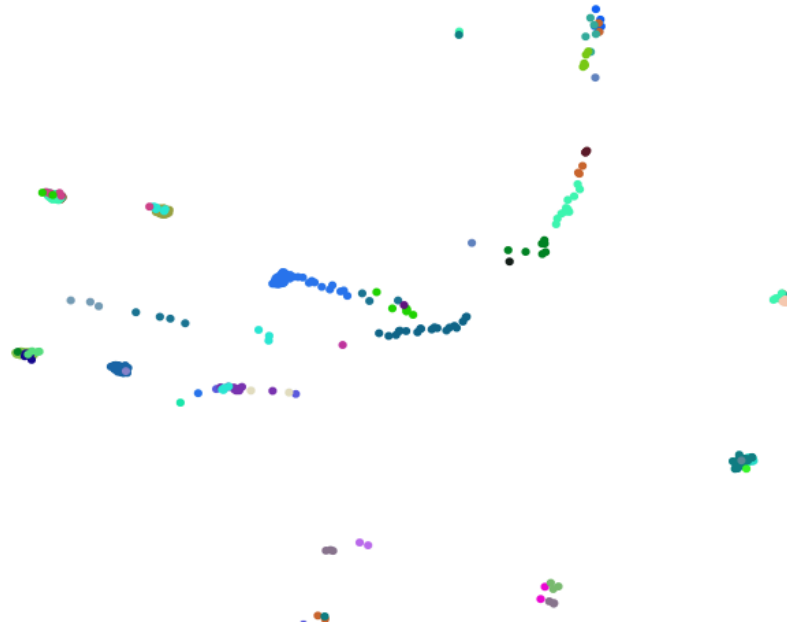


Motion cost mean metrics across the 12 sequences:

	MOTA	MOTP	Mostly Tracked	Mostly Lost
1: raw	0.1393	0.6205	0.7876	0.2106
2: normalized by row max	0.3035	0.6159	0.5497	0.2376
3: normalized by max matrix item	0.2910	0.6139	0.5596	0.2115

Using the motion cost instead of the original IoU cost didn't improve tracking.

Combined IoU and motion feature costs:



Combined IoU & motion cost mean metrics across the 12 sequences:

	MOTA	MOTP	Mostly Tracked	Mostly Lost
Normalized and cost weights tuned	0.3247	0.6176	0.5456	0.2739

The MOTA for tracking using this combined IoU-motion cost (0.3247) is higher than using Part1's only IoU cost approach (0.3243). Also, the Mostly Lost tracklets for the combined approach is 0.2739, which is better than only IoU cost approach of 0.3926. This decrease in Mostly Lost suggests that the tracklets are less fragmented.

Limitations

Are there setups in which your approach fails?

Using the more sophisticated cost function by combining IoU and motion feature costs did not give better performance than using the IoU cost function alone. This could be due to the motion feature cost function being unable to obtain historical velocity at the beginning of a new tracklet where there is only one bbox trajectory within this tracklet. Hence, we are unable to take the difference between any historical velocity and the current velocity.

Are there potential paths of future work to improve your approach?

Reducing the fragmentation of tracklets would eliminate unnecessary new tracklets being created. Thus, we would be able to obtain historical velocity from the tracklet (containing multiple trajectory boxes). Consequently, the motion feature cost would better capture the change in velocities and contribute more meaningfully to our combined cost function, improving the accuracy of tracking. Furthermore, better occlusion handling across frames would improve the accuracy as currently IoU cost is large for vehicles that don't get bbox detection during and after occluded frames.

Part 3. Motion Forecasting Questions

Q3.1.1

The code also contains an option to feed the current and past yaw as input to the model, in addition to x and y. Does adding yaw information help? Why / why not?

Yes, adding yaw information would help with encoding the direction of the vehicle because they can be used to predict the future direction of the vehicle, and hence the future position of where the car will be located after a known period of time.

Q3.1.2

Right now our prediction has a similar output space to our detector - it predicts a box location at each future time. Can you think of other output parametrizations which can be used? What are their pros and cons?

We can also include use yaw and velocity. Adding them to our model will increase the accuracy for the predictions, but particularly with velocity, since we would need extra time to compute the displacement between current frame and previous frame, the inference time will be so much longer.

Bonus: Occlusion Handling for Improved Object Tracking

Motivation

What is the problem being tackled and why is this problem relevant?

When the sensors are unable to capture vehicle information due to occlusion, we lose track of the occluded vehicle for this frame. In Part 1, the tracking algorithm would create a new tracklet when the occluded vehicle reappears in the future frame and will assign a new actor ID. This caused an issue and will always lead to low accuracy because this never matches the ground truth, which labels the cars in the two frames as the same actor ID.

This problem is relevant to us because it is always beneficial for our SDV to gain a better understanding of the surroundings. Moreover, in our dataset, there are also occlusion scenarios which eventually result in fragmented tracking trajectories due to the misassignment of the actor IDs.

Techniques

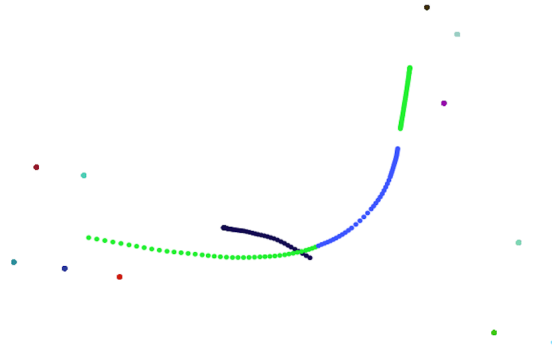
In the post-processing of tracklets, go through each tracklet and compute the iou between this tracklet A's last bbox and other tracklet B's first bbox. If this IoU > 0.1 , we extend this tracklet A to contain the other tracklet B, and then delete the other tracklet B.


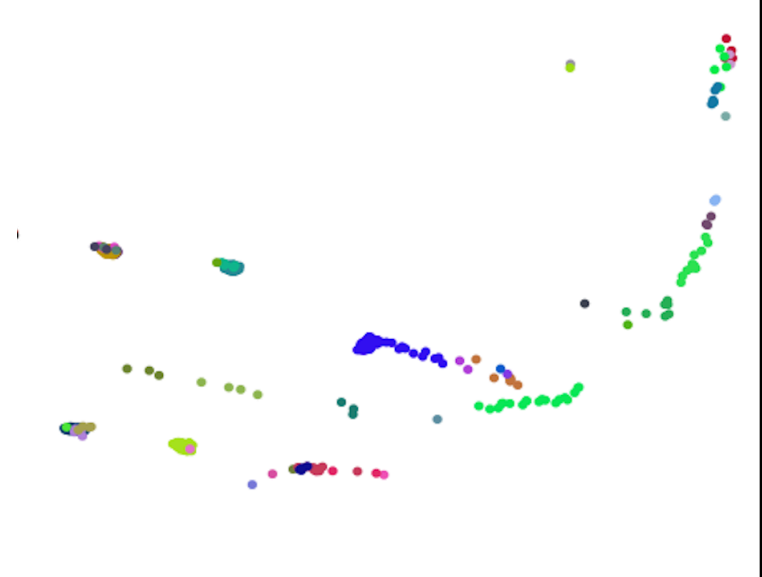
By estimating the bounding boxes for the occluded vehicle to be stationary across the frames where no detections exist for this vehicle, we will be able to identify any other tracklet that is a continuation for this occluded vehicle. Then, append the potential tracklet continuation to the same actor's tracklet.

Evaluation

Example using seq#35

Ground truth:



Our improved tracker	Part 1 baseline tracker
	

We see that each tracklet has a distinct color for an actor in the scene when using our improved tracker. There is less fragmentation: the tracklets are longer and connected, reflecting the actor movements. For example, the stationary vehicle blobs are in one color per vehicle (one tracklet per actor), which is better than the multiple different colored dot blobs (representing different short tracklets and multiple actors) in the baseline visualization.

Furthermore, there is a 0.003 increase in MOTA and MOTP for the sequence#35:

Sequence #35	MOTA	MOTP
Baseline	0.2682	0.6093
Our improved tracker	0.2712	0.6123

There is a 0.002 increase in the median MOTA and MOTP for the 12 validation sequences:

Median across 12 validation sequences	MOTA	MOTP
Baseline	0.2969	0.6223
Our improved tracker	0.2984	0.6244

Limitations

The stationary motion assumption only works well for vehicles that are, for example, waiting at the traffic light. It doesn't improve tracking for fast moving vehicles. A better way of occlusion handling is to use a motion assumption (e.g. constant velocity) to estimate the vehicle location upon reappearance, and then compute the IoU and compare the similarity in velocity between the tracklets would be more appropriate for moving vehicles. However, for a longer occlusion period, even the constant motion assumption would not work. Therefore, it is better to use motion forecasting to estimate the more realistic location of the occluded vehicle, and then combine tracklets for that vehicle for better overall tracking results.