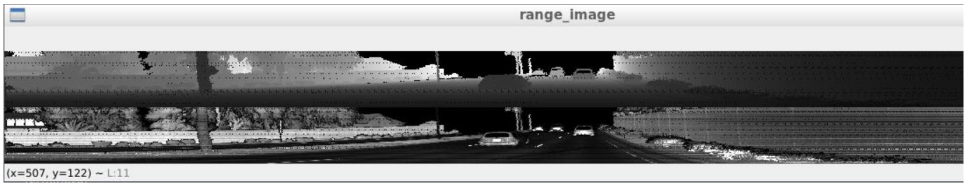
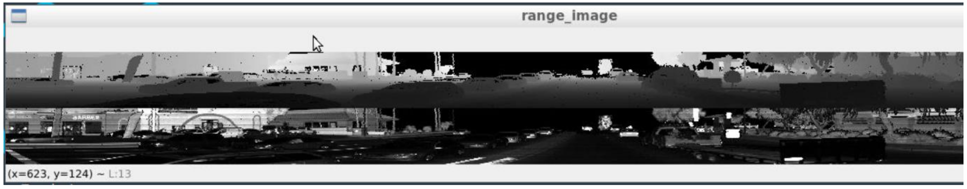
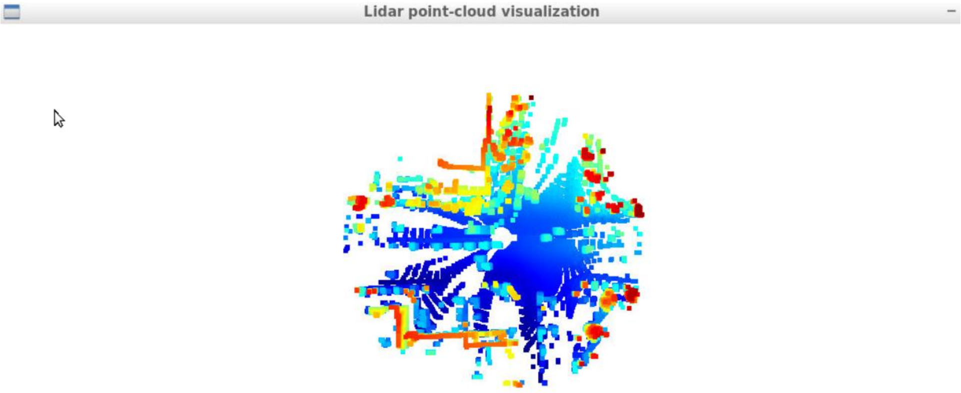
Compute Lidar Point-Cloud from Range Image

Visualize range image channels (ID_S1_EX1)



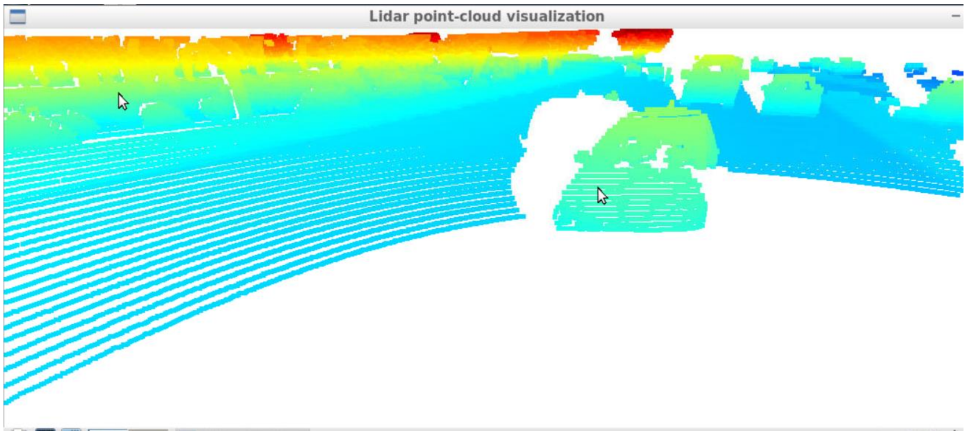


Visualize point-cloud (ID_S1_EX2)

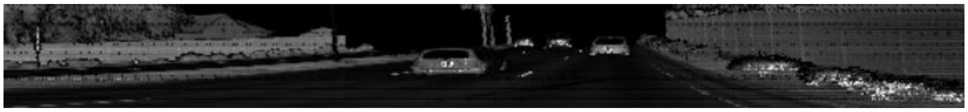


Following are the examples of vehicles as viewed in the plotted 3d point cloud

Example 1

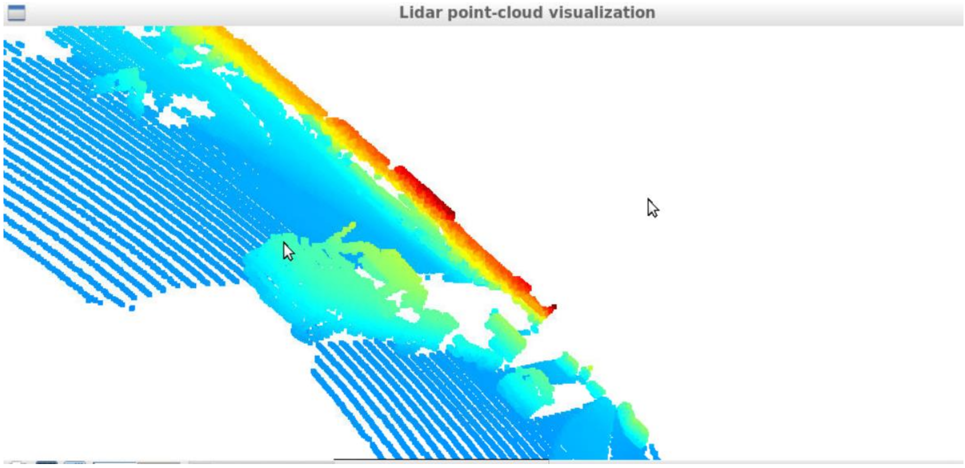


Vehicle features identified include: - Front tires - Front left mirror - front windshield- bumper



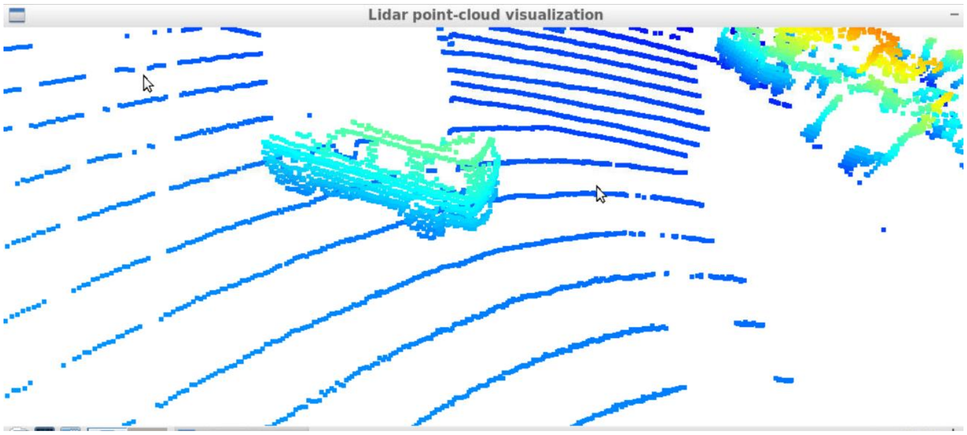
The above image shows three cars with varying level of details. The associated intensity image corroborates the lidar. Front and Rear right tires - Passenger door/windows - Side mirrors

Example 2:

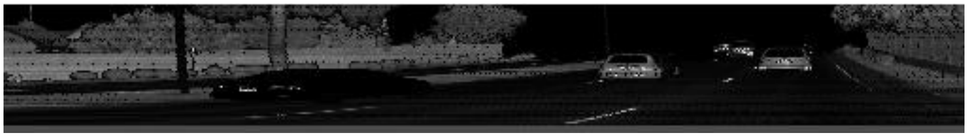


Identifiable vehicle feature in the above lidar : - Truck Front Bumper, Truck Roof - Front and Rear right tires - Truck rear, trailer

Example 3:

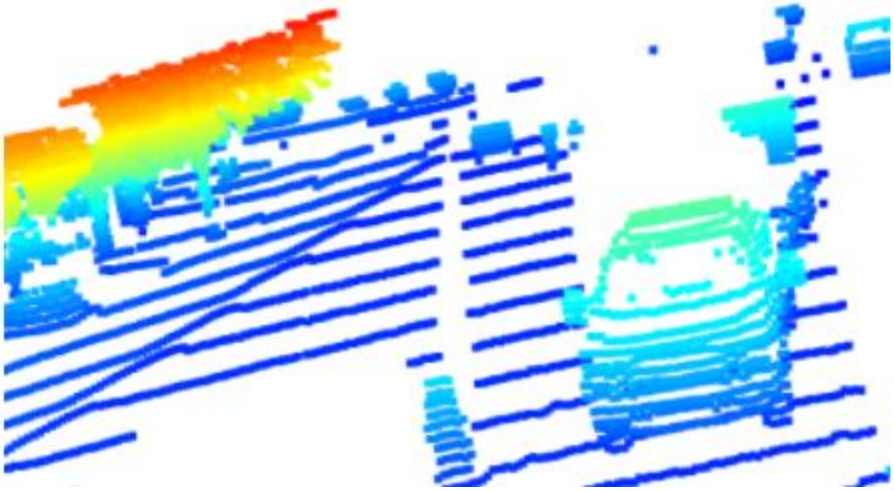


Identifiable vehicle feature in the above lidar : - SUV Side view, with side doors & windows - Front and Rear right tires - Rear bumper and taillights



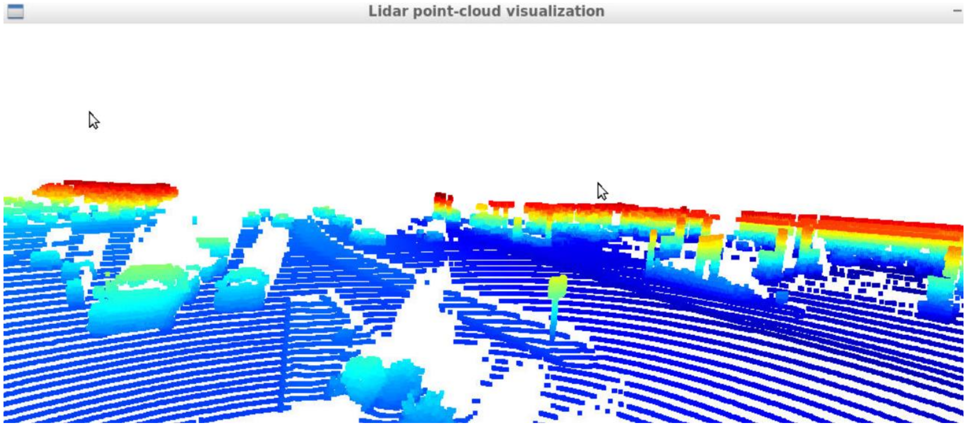
Identifiable vehicle feature in the above lidar and associated intensity image : - Front and Rear right tires - Passenger door/windows - Side mirrors

Example 4:



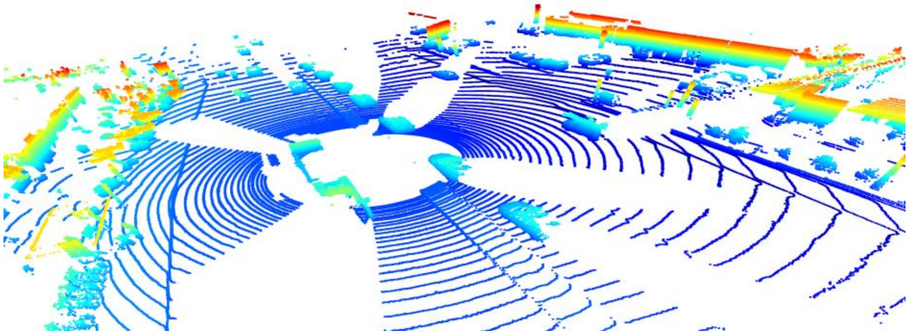
Identifiable vehicle features: – Front windshield- Front bumper - Front and rear tires – Side mirrors

Example 5:



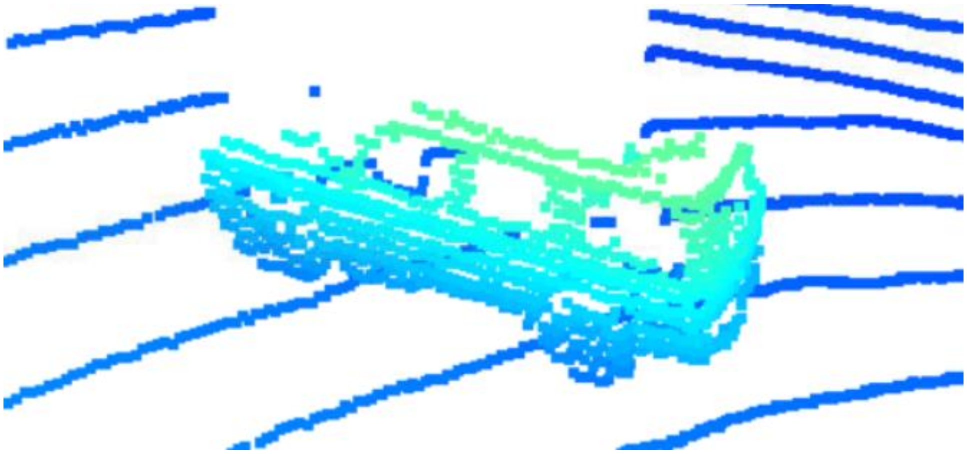
Identifiable vehicle feature in the above lidar : - Front windshields of multiple cars - rear window - Front and Rear right tires - Front and rear bumpers – Side mirrors

Example 6:



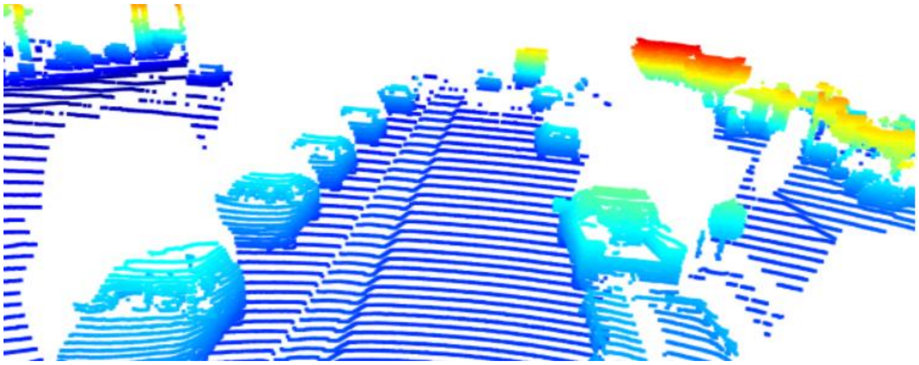
Identifiable vehicles traveling in multiple directions in the above lidar. Identifiable vehicle features: - Front windshields – Rear windows – side mirrors

Example 7:



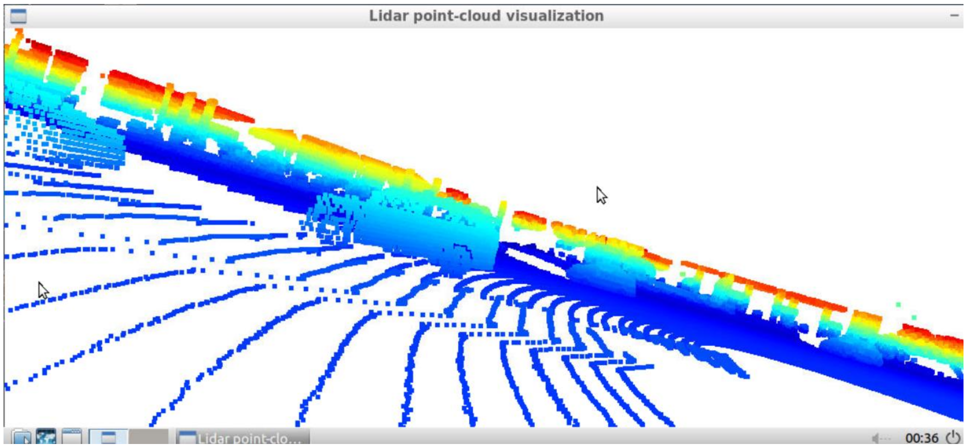
Identifiable vehicle features: – Rear window- rear bumper - Front and rear tires – windows – doors

Example 8:



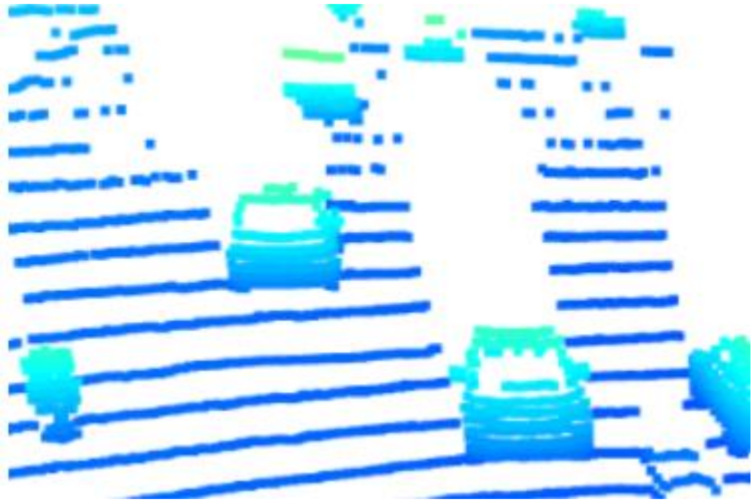
Identifiable vehicles traveling in multiple directions in the above lidar. Identifiable vehicle features: - Front windshields – Rear windows – side mirrors – Front and rear tires – windows – doors – Truck - trailer

Example 9:



Identifiable vehicle feature in the above lidar : - Front bumper - left mirrors – windows –front and rear left tires

Example 10:



Identifiable vehicle feature in the above lidar : - Front windshield – Rear window - Front and rear bumper - side mirrors –front and rear left tires

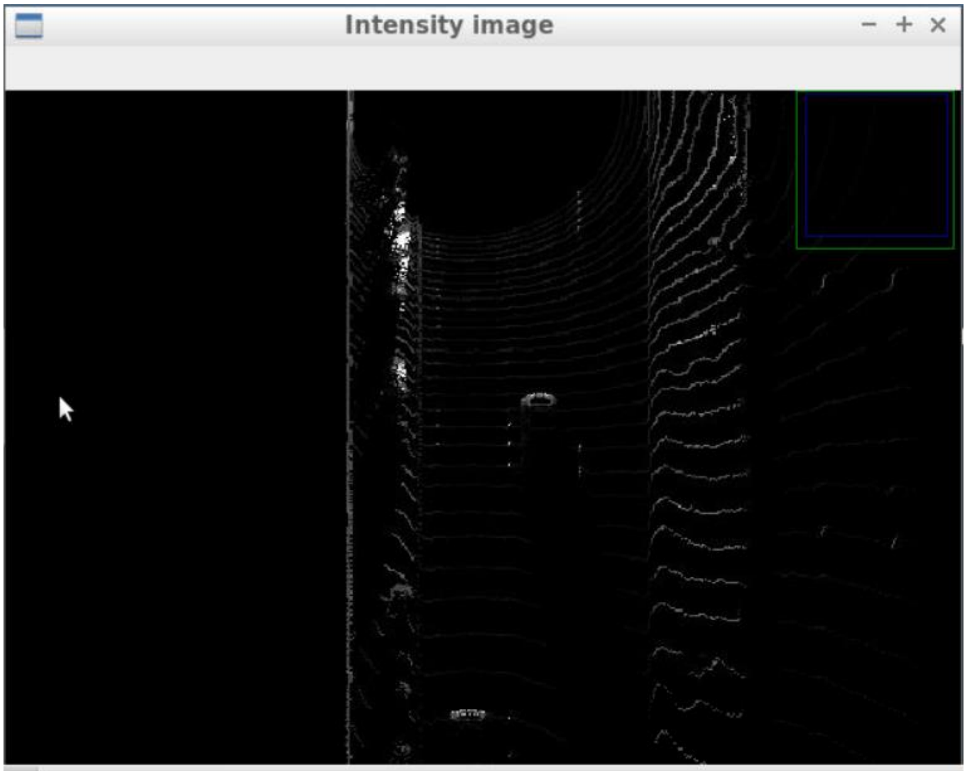
Create Birds-Eye View from Lidar PCL

Convert sensor coordinates to bev-map coordinates (ID_S2_EX1)

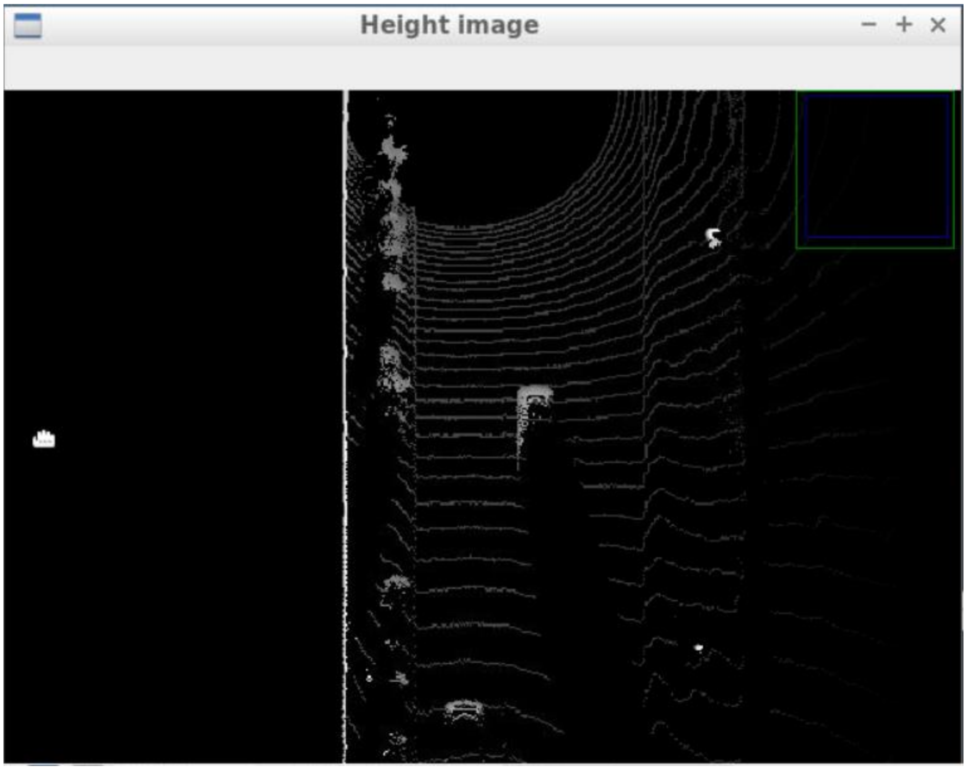
In file `loop_over_dataset.py`, set the attributes for code execution in the following way:

- `data_filename = 'training_segment-10050810020241296553_5313_150_5333_150_with_camera_labels.tfrecord`
- `show_only_frames = [50, 51]`
- `exec_data = ['pcl_from_rangeimage', 'load_image']`
- `exec_detection = ['bev_from_pcl', 'detect_objects']`
- `exec_tracking = []`
- `exec_visualization = ['show_objects_in_bev_labels_in_camera']`
- `configs_det = det.load_configs(model_name="fpn_resnet")`

Compute intensity layer of bev-map (ID_S2_EX2)

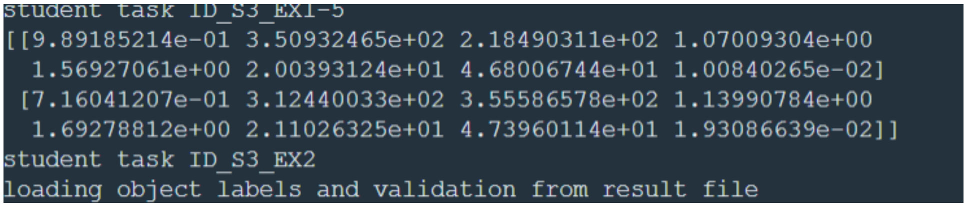


Compute height layer of bev-map (ID_S2_EX3)



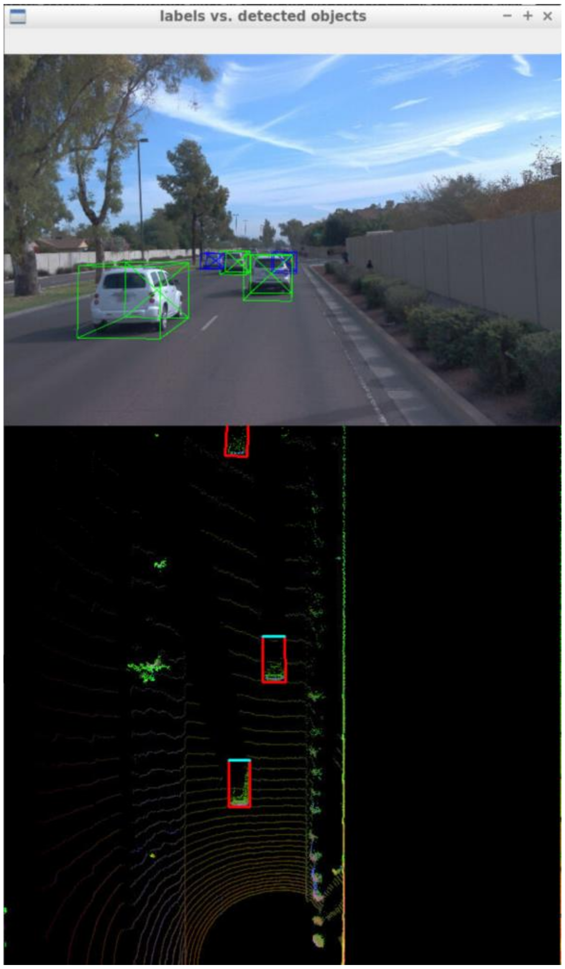
Model-based Object Detection in BEV Image

Add a second model from a GitHub repo (ID_S3_EX1)



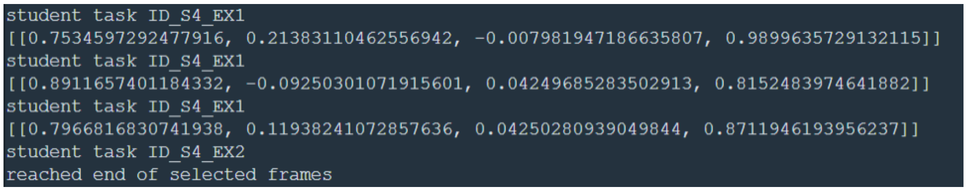
```
student task ID_S3_EX1-5
[[9.89185214e-01 3.50932465e+02 2.18490311e+02 1.07009304e+00
  1.56927061e+00 2.00393124e+01 4.68006744e+01 1.00840265e-02]
 [7.16041207e-01 3.12440033e+02 3.55586578e+02 1.13990784e+00
  1.69278812e+00 2.11026325e+01 4.73960114e+01 1.93086639e-02]]
student task ID_S3_EX2
loading object labels and validation from result file
```

Extract 3D bounding boxes from model response (ID_S3_EX2)



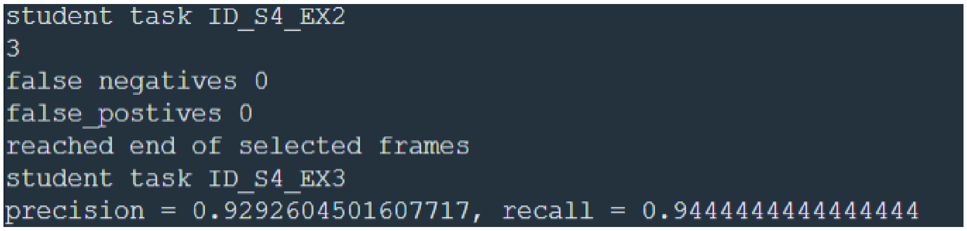
Performance Evaluation for Object Detection

Compute intersection-over-union (IOU) between labels and detections (ID_S4_EX1)



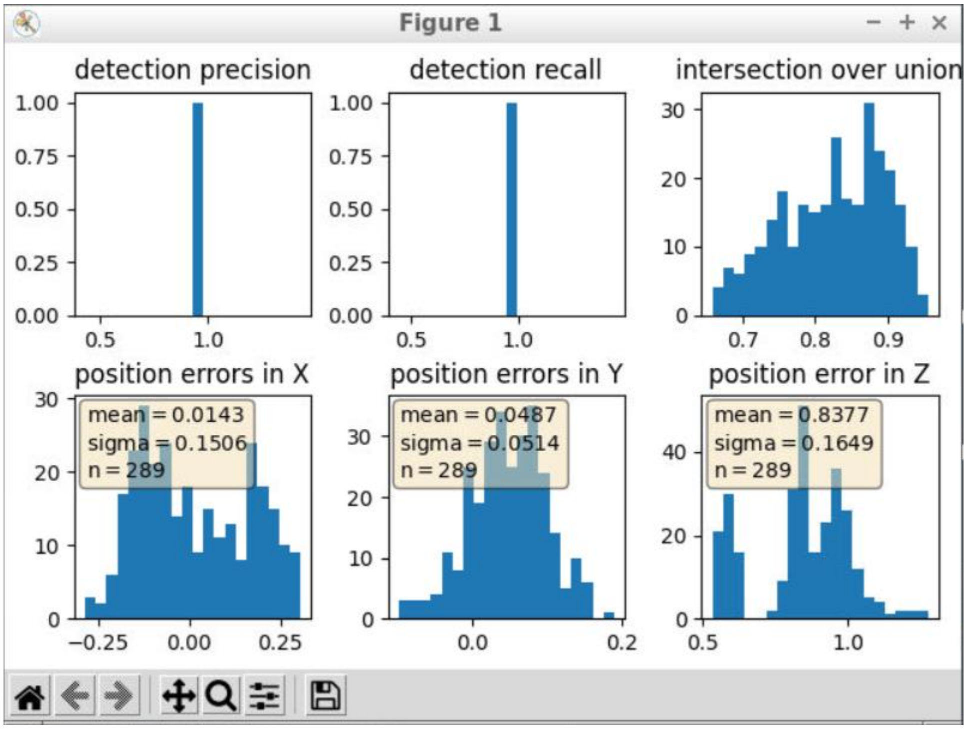
```
student task ID_S4_EX1
[[0.7534597292477916, 0.21383110462556942, -0.007981947186635807, 0.9899635729132115]]
student task ID_S4_EX1
[[0.8911657401184332, -0.09250301071915601, 0.04249685283502913, 0.8152483974641882]]
student task ID_S4_EX1
[[0.7966816830741938, 0.11938241072857636, 0.04250280939049844, 0.8711946193956237]]
student task ID_S4_EX2
reached end of selected frames
```

Compute false-negatives and false-positives (ID_S4_EX2)



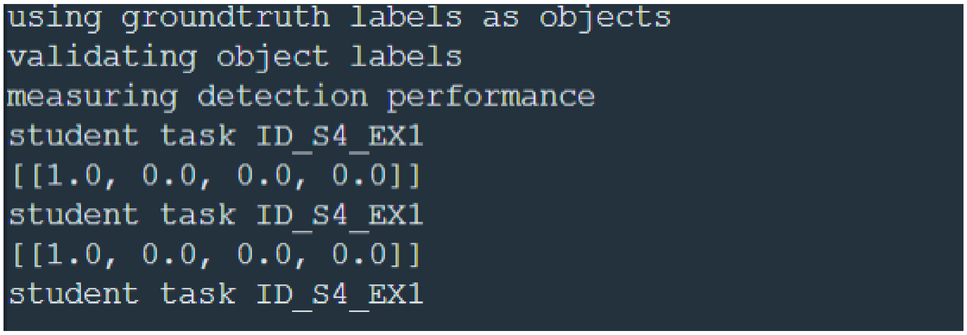
```
student task ID_S4_EX2
3
false negatives 0
false_postives 0
reached end of selected frames
student task ID_S4_EX3
precision = 0.9292604501607717, recall = 0.9444444444444444
```

Compute precision and recall (ID_S4_EX3)

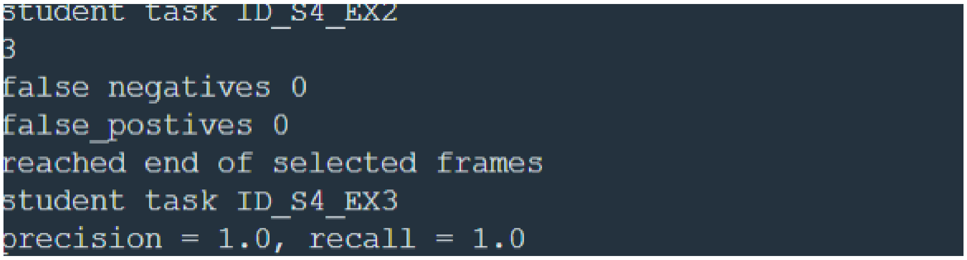


```
configs_det.use_labels_as_objects = True
```

Ideal Precision/Recall using Groundtruths as labels



```
using groundtruth labels as objects
validating object labels
measuring detection performance
student task ID_S4_EX1
[[1.0, 0.0, 0.0, 0.0]]
student task ID_S4_EX1
[[1.0, 0.0, 0.0, 0.0]]
student task ID_S4_EX1
```



```
student task ID_S4_EX2
3
false negatives 0
false_postives 0
reached end of selected frames
student task ID_S4_EX3
precision = 1.0, recall = 1.0
```

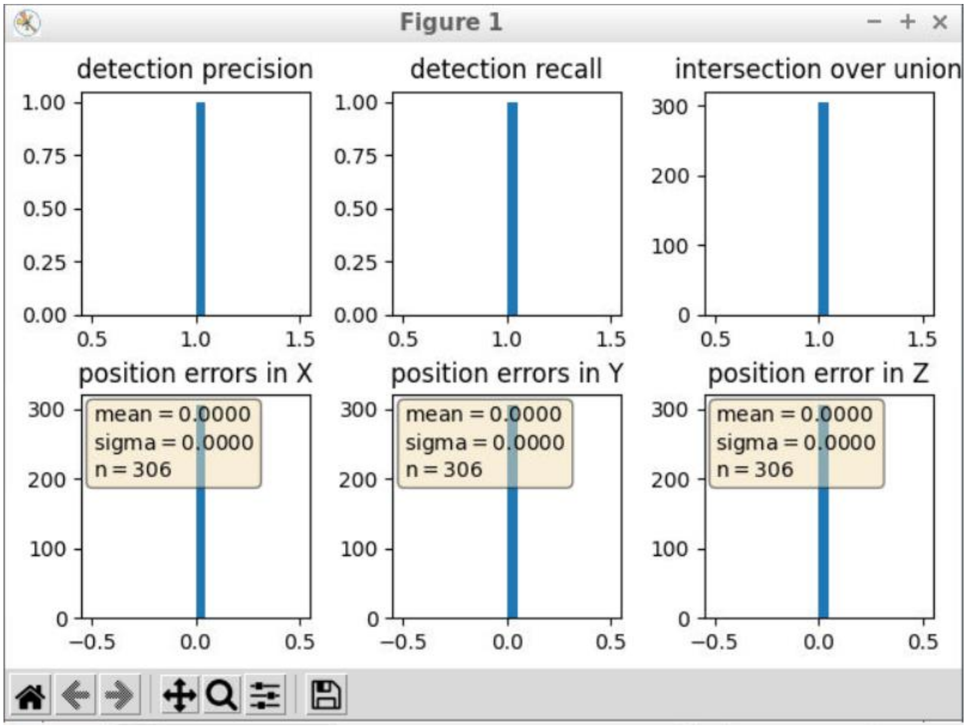


Figure 1
detection precision
detection recall
intersection over union
position errors in X
mean = 0.0000
sigma = 0.0000
n = 306
position errors in Y
mean = 0.0000
sigma = 0.0000
n = 306
position error in Z
mean = 0.0000
sigma = 0.0000
n = 306