Object Detection using LiDAR

→ Objects are detected & stored in bounding boxes.

→ fit cuboid bounding boxes around detected objects.

→ Lidar toolbox uses L-shape fitting approach to detect objects.

→ PointSeg network: predict the class of object

→ Joint probabilistic data Association (JPDA) and Interacting multiple model (IMM): track vehicles.

→ workflow:

Lidar point cloud data.

↓

(x, y, z, intensity, range) i.e ← 5-channel of images input

↓

Semantic Segmentation
- 1) Ground plane segmentation
- 2) Object detection + classification

↓

Oriented Bounding Box fitting

↓

Bounding Box tracking
- 1) JPDA tracker
- 2) IMM filter

Steps:

(1) Ground plane Segmentation.

uses: a) segmentGroundFromLidarData and
b) pcfitplane function (piece wise plane fitting → this method is robust)

→ segmentGroundFromLidarData function is to get an estimate of the ground plane.

→ The estimated ground plane is divided into strips along the direction of the vehicle so as to fit the plane using pcfitplane function.

(2) Semantic Segmentation

→ uses PointSeg n/w model, this model is trained for object classes like Cars, truck & background

→ The o/p from n/w is masked image with each pixel labelled as per its class.

→ pcsegdist function: to obtain different object clusters.

{ for training n/w, refer yesterday's document }

(3) Bounding Box fitting.

→ Now, each object cluster is fitted in an L-shape bounding box.

→ State vector of detected object is
$[x, y, z, \phi, l, w, h]$

(x, y, z) → centroid of bounding box.
ϕ → Yaw angle of " " (rad).
(l, w, h) → Dimensions of bounding box.
(i.e length, width & height).

→ getObjectDetections object.
1) extracts oriented bounding box informat.
2) Corners of bounding boxes.
3) Segment point clouds in different colours depending on class info.
4) it takes 5-channel lidar data & Current time as i/p.

(4) Visualization

→ helper/LidarObjectDetectionDisplay class is used to setup the visualization window.

Visualization window shows:
1) Point Cloud image in 2-D.
2) Detected labels generated from Semantic Segmentation n/w.
3) 3-D point cloud with oriented bounding boxes
4) Top View of point cloud with " "

(5) - Loop through data.

→ display class

→ Initialize HelperLidarObjectDetection Display class.

→ getObjectDetections object: extracts oriented bounding box information.

→ initialize display class by creating its own object.

for (inputdata) :

① extract BoundingBox information

② update point Cloud

③ update Bounding Box.

④ update Segmented Image.

% Now display class is upto date.

end.

(6) Adding trackers.

→ we use trackers to stabilize the detections.

→ trackerJPDA object is used for the same

→ properties [x, y, z, θ, l, w, h]

→

a) Initialize parameters for JPDA tracker.

b) Initialize IMM filter.

c) Now, initialize tracker JPDA object (it takes parameters, IMM filter as arguments). which is

d) Now, use the tracker initialized above to process the data.

for - - :

end.

HelperLidarObjectDetection : It is the main object detection class that handles

1) preprocessing of point cloud.
2) Ground plane fitting.
3) Segmentation
4) clustering with L-shape fitting (Bounding box).

→ This class has 5 sections :

1) Semantic Segmentation
2) Ground plane estimation
3) Clustering & Bounding box fitting
4) Conversation of detections into object detection class object.

5) Main function to be called.
& It calls all the functions ?

→ getObjectDetections is Main function.