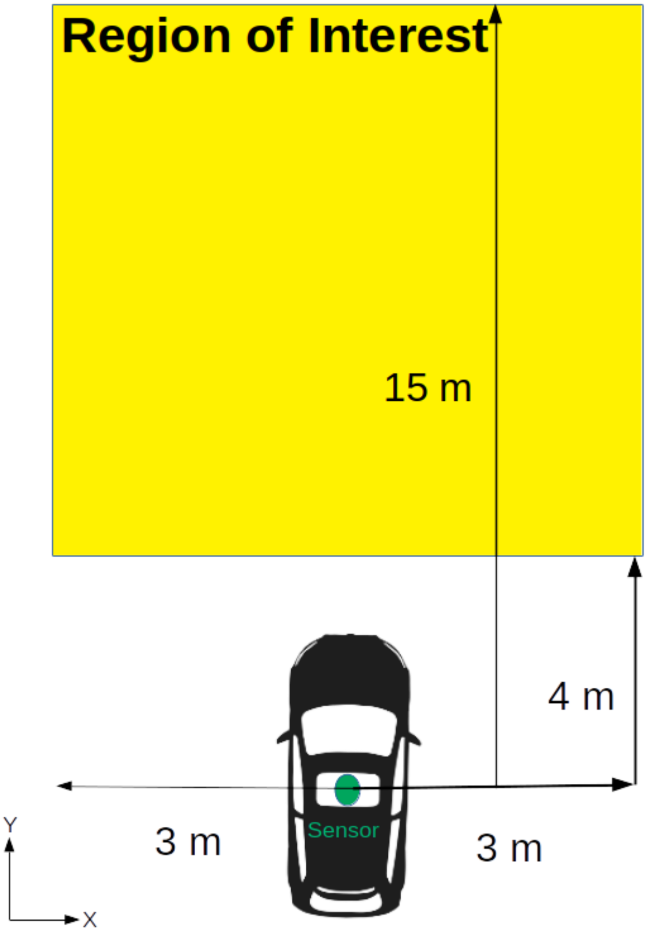
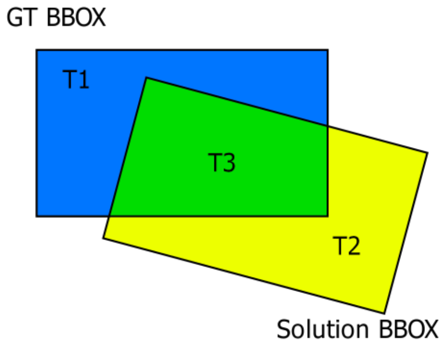
# Scoring for parking car detection

The solutions given by you will be scored using an automated system. The estimated bounding boxes (BBOXs) will be compared to the Ground Truth (GT) ones. Parking cars should be detected only in the **last LiDAR/camera frame (timestamp) of the sequence.**

The Region of Interest (RoI) is visualized below. Parking cars should be detected in a rectangle on the ground. The union of the RoI and the detected BBOX should not be empty, otherwise the detection is discarded from the scoring. For the RoI, the distances of the closest and farest borders are 4 and 15 meters, respectively. To left and right, 3 meters should be examined. The origin is given by the LiDAR sensor.



The pairing of the bounding boxes is done automatically by a greedy algorithm: pairing the GT BBOX with the one given that can reach the highest score. One GT bounding box is paired with only one provided bounding box. The individual score for every BBOX is in the interval [0.0, 1.0] depending on how it fits with the paired GT. The score is calculated in the following way:



Precision: T3 / (T2 + T3)

Recall: T3 / (T1 + T3)

**Score: (1 * Precision + 2 * Recall) / 3**

For the definition of recall and precision, see the related Wikipedia page.

Given the above weighting of the precision and recall, we punish the false-negative results more. It's a higher risk if one says that there is not car where there is really, compared to the reverse case, when he/she predicts that there is a car where there is not.

The final score is the sum of the scores of the provided bounding boxes, divided by the maximum of the number of bounding boxes given and the number of GT data.

Final score: $\left(\sum_{i=1}^{k} Score(BBOX_i,\ GT_i)\right) / \max(k, l)$, where *k* is the number of bounding boxes given in the solution, and *l* is the number of GT bounding boxes.

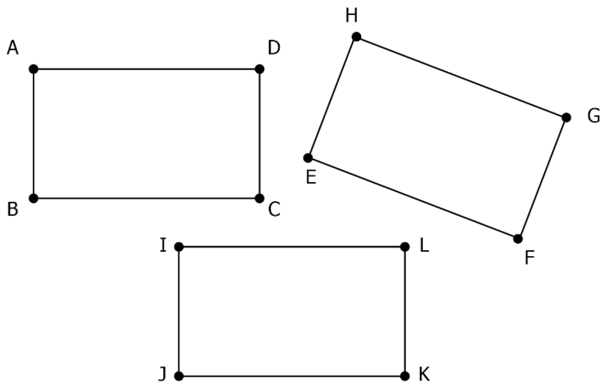
# File format for the submission

During the solution of the problem, for every *scene,* you determine *k* bounding boxes (BBOXs). We ask for a different file for every scene that contains the description of bounding boxes. For the *n*$^{th}$ scene, the name of the file should be '*scene_n.txt'*. As you can choose to provide the bounding boxes in the point cloud or on the images, the first line should contain on which sensor the bounding boxes are given continued by the description of the bounding boxes you found. *But only one prediction is allowed per scene. If you send more files for the same scene, the score is zero.*

Put a new line after every bounding box description. The first line can contain the word '*lidar*' in case the bounding boxes are given in the point cloud or the word '*camera'* followed by the ID number of the camera (*0, 1 or 2*).

A bounding box can be given by the coordinates of its four corners. The description of a bounding box is composed of four lines containing the coordinates of the four corners. The order of the corners should be given in anti-clockwise/direct order. The order of the bounding boxes doesn't matter. The coordinates of the corners should be given as decimal fractions.

**Example.** The image below contains the bounding boxes determined in the first scene (*n=1*). In total *k=3* bounding boxes were found:



The file *'scene_1.txt'* contains the description of the bounding boxes with a new line after every bounding box. The first line depends on the sensor type. It can be 'lidar' or 'camera j' where j is the index of the camera.  Note the ordering of the corners should be counterclockwise.

| lidar | camera 1 |
|---|---|
| $X_A$ $Y_A$ $Z_A$ | $X_A$ $Y_A$ |
| $X_B$ $Y_B$ $Z_B$ | $X_B$ $Y_B$ |
| $X_C$ $Y_C$ $Z_C$ | $X_C$ $Y_C$ |
| $X_D$ $Y_D$ $Z_D$ | $X_D$ $Y_D$ |
| | |
| $X_E$ $Y_E$ $Z_E$ | $X_E$ $Y_E$ |
| $X_F$ $Y_F$ $Z_F$ | $X_F$ $Y_F$ |
| $X_G$ $Y_G$ $Z_G$ | $X_G$ $Y_G$ |
| $X_H$ $Y_H$ $Z_H$ | $X_H$ $Y_H$ |
| | |
| $X_I$ $Y_I$ $Z_I$ | $X_I$ $Y_I$ |
| $X_J$ $Y_J$ $Z_J$ | $X_J$ $Y_J$ |
| $X_K$ $Y_K$ $Z_K$ | $X_K$ $Y_K$ |
| $X_L$ $Y_L$ $Z_L$ | $X_L$ $Y_L$ |