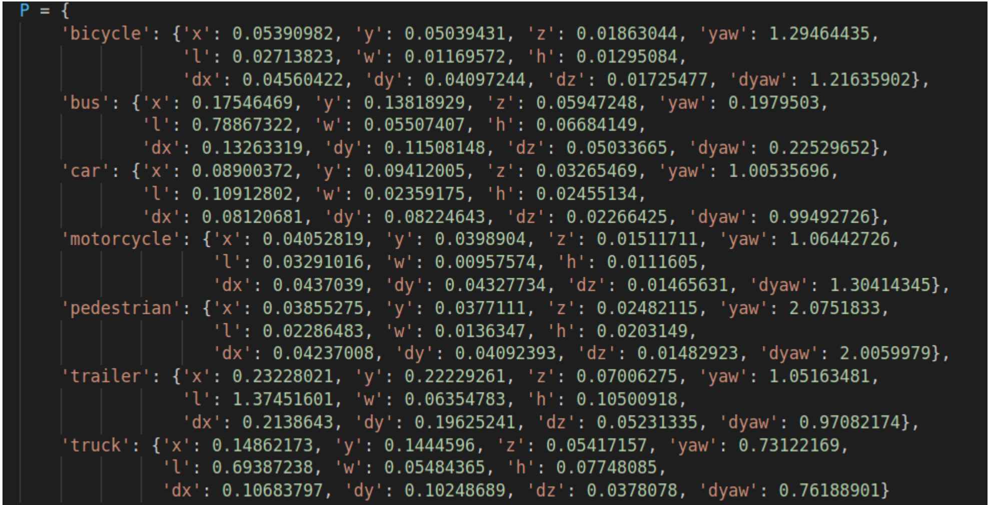
# NuScenes Based Tracking Project Status Report

January 14, 2022

# Measurement-Track association NOT a performance constraint

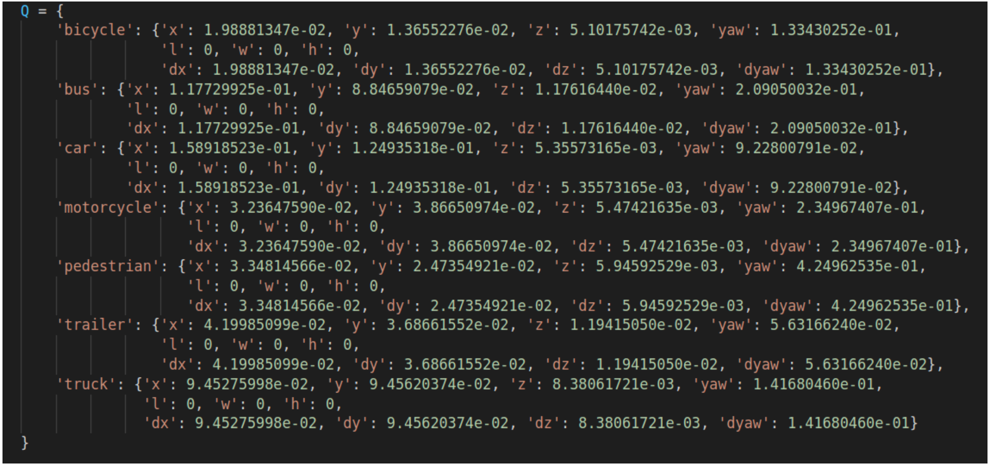
People who study computer vision are aware of the techniques provided by the fusion community, yet they choose not to use any because the association step is **NOT** a performance constraint as of now.

# Measurement-Track association NOT a performance constraint



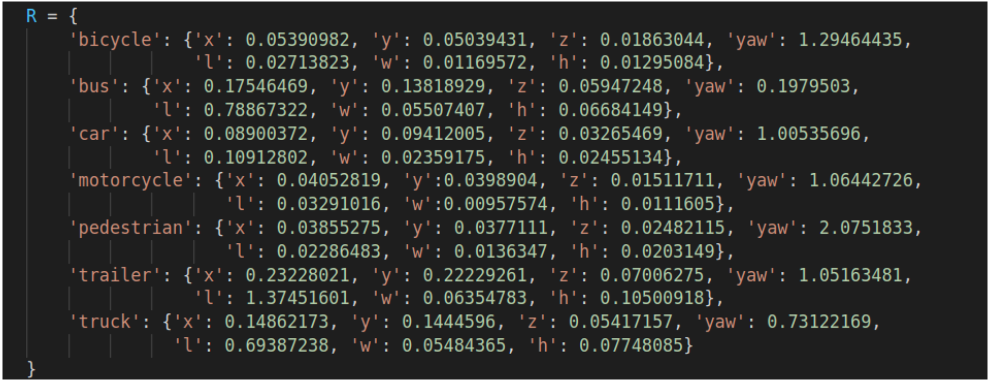
```
P = {
    'bicycle': {'x': 0.05390982, 'y': 0.05039431, 'z': 0.01863044, 'yaw': 1.29464435,
                'l': 0.02713823, 'w': 0.01169572, 'h': 0.01295084,
                'dx': 0.04560422, 'dy': 0.04097244, 'dz': 0.01725477, 'dyaw': 1.21635902},
    'bus': {'x': 0.17546469, 'y': 0.13818929, 'z': 0.05947248, 'yaw': 0.1979503,
            'l': 0.78867322, 'w': 0.05507407, 'h': 0.06684149,
            'dx': 0.13263319, 'dy': 0.11508148, 'dz': 0.05033665, 'dyaw': 0.22529652},
    'car': {'x': 0.08900372, 'y': 0.09412005, 'z': 0.03265469, 'yaw': 1.00535696,
            'l': 0.10912802, 'w': 0.02359175, 'h': 0.02455134,
            'dx': 0.08120681, 'dy': 0.08224643, 'dz': 0.02266425, 'dyaw': 0.99492726},
    'motorcycle': {'x': 0.04052819, 'y': 0.0398904, 'z': 0.01511711, 'yaw': 1.06442726,
                   'l': 0.03291016, 'w': 0.00957574, 'h': 0.0111605,
                   'dx': 0.0437039, 'dy': 0.04327734, 'dz': 0.01465631, 'dyaw': 1.30414345},
    'pedestrian': {'x': 0.03855275, 'y': 0.0377111, 'z': 0.02482115, 'yaw': 2.0751833,
                   'l': 0.02286483, 'w': 0.0136347, 'h': 0.0203149,
                   'dx': 0.04237008, 'dy': 0.04092393, 'dz': 0.01482923, 'dyaw': 2.0059979},
    'trailer': {'x': 0.23228021, 'y': 0.22229261, 'z': 0.07006275, 'yaw': 1.05163481,
                'l': 1.37451601, 'w': 0.06354783, 'h': 0.10500918,
                'dx': 0.2138643, 'dy': 0.19625241, 'dz': 0.05231335, 'dyaw': 0.97082174},
    'truck': {'x': 0.14862173, 'y': 0.1444596, 'z': 0.05417157, 'yaw': 0.73122169,
              'l': 0.69387238, 'w': 0.05484365, 'h': 0.07748085,
              'dx': 0.10683797, 'dy': 0.10248689, 'dz': 0.0378078, 'dyaw': 0.76188901}
```

# Measurement-Track association NOT a performance constraint



```
Q = {
    'bicycle': {'x': 1.98881347e-02, 'y': 1.36552276e-02, 'z': 5.10175742e-03, 'yaw': 1.33430252e-01,
                'l': 0, 'w': 0, 'h': 0,
                'dx': 1.98881347e-02, 'dy': 1.36552276e-02, 'dz': 5.10175742e-03, 'dyaw': 1.33430252e-01},
    'bus': {'x': 1.17729925e-01, 'y': 8.84659079e-02, 'z': 1.17616440e-02, 'yaw': 2.09050032e-01,
            'l': 0, 'w': 0, 'h': 0,
            'dx': 1.17729925e-01, 'dy': 8.84659079e-02, 'dz': 1.17616440e-02, 'dyaw': 2.09050032e-01},
    'car': {'x': 1.58918523e-01, 'y': 1.24935318e-01, 'z': 5.35573165e-03, 'yaw': 9.22800791e-02,
            'l': 0, 'w': 0, 'h': 0,
            'dx': 1.58918523e-01, 'dy': 1.24935318e-01, 'dz': 5.35573165e-03, 'dyaw': 9.22800791e-02},
    'motorcycle': {'x': 3.23647590e-02, 'y': 3.86650974e-02, 'z': 5.47421635e-03, 'yaw': 2.34967407e-01,
                   'l': 0, 'w': 0, 'h': 0,
                   'dx': 3.23647590e-02, 'dy': 3.86650974e-02, 'dz': 5.47421635e-03, 'dyaw': 2.34967407e-01},
    'pedestrian': {'x': 3.34814566e-02, 'y': 2.47354921e-02, 'z': 5.94592529e-03, 'yaw': 4.24962535e-01,
                   'l': 0, 'w': 0, 'h': 0,
                   'dx': 3.34814566e-02, 'dy': 2.47354921e-02, 'dz': 5.94592529e-03, 'dyaw': 4.24962535e-01},
    'trailer': {'x': 4.19985099e-02, 'y': 3.68661552e-02, 'z': 1.19415050e-02, 'yaw': 5.63166240e-02,
                'l': 0, 'w': 0, 'h': 0,
                'dx': 4.19985099e-02, 'dy': 3.68661552e-02, 'dz': 1.19415050e-02, 'dyaw': 5.63166240e-02},
    'truck': {'x': 9.45275998e-02, 'y': 9.45620374e-02, 'z': 8.38061721e-03, 'yaw': 1.41680460e-01,
              'l': 0, 'w': 0, 'h': 0,
              'dx': 9.45275998e-02, 'dy': 9.45620374e-02, 'dz': 8.38061721e-03, 'dyaw': 1.41680460e-01}
}
```

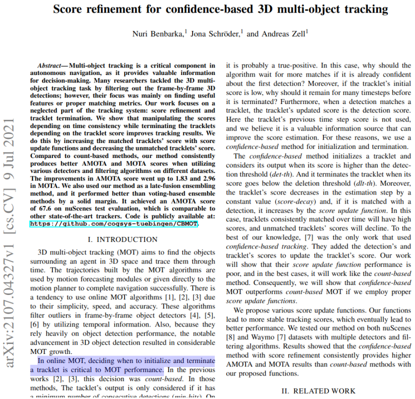
# Measurement-Track association NOT a performance constraint



```
R = {
    'bicycle': {'x': 0.05390982, 'y': 0.05039431, 'z': 0.01863044, 'yaw': 1.29464435,
                'l': 0.02713823, 'w': 0.01169572, 'h': 0.01295084},
    'bus': {'x': 0.17546469, 'y': 0.13818929, 'z': 0.05947248, 'yaw': 0.1979503,
            'l': 0.78867322, 'w': 0.05507407, 'h': 0.06684149},
    'car': {'x': 0.08900372, 'y': 0.09412005, 'z': 0.03265469, 'yaw': 1.00535696,
            'l': 0.10912802, 'w': 0.02359175, 'h': 0.02455134},
    'motorcycle': {'x': 0.04052819, 'y':0.0398904, 'z': 0.01511711, 'yaw': 1.06442726,
                   'l': 0.03291016, 'w':0.00957574, 'h': 0.0111605},
    'pedestrian': {'x': 0.03855275, 'y': 0.0377111, 'z': 0.02482115, 'yaw': 2.0751833,
                   'l': 0.02286483, 'w': 0.0136347, 'h': 0.0203149},
    'trailer': {'x': 0.23228021, 'y': 0.22229261, 'z': 0.07006275, 'yaw': 1.05163481,
                'l': 1.37451601, 'w': 0.06354783, 'h': 0.10500918},
    'truck': {'x': 0.14862173, 'y': 0.1444596, 'z': 0.05417157, 'yaw': 0.73122169,
              'l': 0.69387238, 'w': 0.05484365, 'h': 0.07748085}
}
```

# how to kill a tracklet

Quite a lot teams are investigating the issue of when to die.



## Score refinement for confidence-based 3D multi-object tracking

Nuri Benbarka,[1] Jona Schröder,[1] and Andreas Zell[1]

arXiv:2107.04327v1 [cs.CV] 9 Jul 2021

***Abstract*— Multi-object tracking is a critical component in autonomous navigation, as it provides valuable information for decision-making. Many researchers tackled the 3D multi-object tracking task by filtering out the frame-by-frame 3D detections; however, their focus was mainly on finding useful features or proper matching metrics. Our work focuses on a neglected part of the tracking system: score refinement and tracklet termination. We show that manipulating the scores depending on time consistency while terminating the tracklets depending on the tracklet score improves tracking results. We do this by increasing the matched tracklets' score with score update functions and decreasing the unmatched tracklets' score. Compared to count-based methods, our method consistently produces better AMOTA and MOTA scores when utilizing various detectors and filtering algorithms on different datasets. The improvements in AMOTA score went up to 1.83 and 2.96 in MOTA. We also used our method as a late-fusion ensembling method, and it performed better than voting-based ensemble methods by a solid margin. It achieved an AMOTA score of 67.6 on nuScenes test evaluation, which is comparable to other state-of-the-art trackers. Code is publicly available at: `https://github.com/cogsys-tuebingen/CBMOT`.**

### I. INTRODUCTION

3D multi-object tracking (MOT) aims to find the objects surrounding an agent in 3D space and trace them through time. The trajectories built by the MOT algorithms are used by motion forecasting modules or given directly to the motion planner to complete navigation successfully. There is a tendency to use online MOT algorithms [1], [2], [3] due to their simplicity, speed, and accuracy. These algorithms filter outliers in frame-by-frame object detectors [4], [5], [6] by utilizing temporal information. Also, because they rely heavily on object detection performance, the notable advancement in 3D object detection resulted in considerable MOT growth.

In online MOT, deciding when to initialize and terminate a tracklet is critical to MOT performance. In the previous works [2], [3], this decision was *count-based*. In those methods, The tacklet's output is only considered if it has a minimum number of consecutive detections (*min-hits*). On

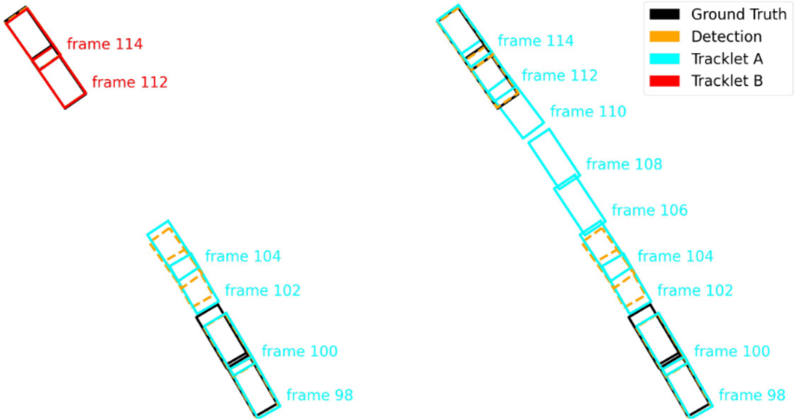
it is probably a true-positive. In this case, why should the algorithm wait for more matches if it is already confident about the first detection? Moreover, if the tracklet's initial score is low, why should it remain for many timesteps before it is terminated? Furthermore, when a detection matches a tracklet, the tracklet's updated score is the detection score. Here the tracklet's previous time step score is not used, and we believe it is a valuable information source that can improve the score estimation. For these reasons, we use a *confidence-based* method for initialization and termination.

The *confidence-based* method initializes a tracklet and considers its output when its score is higher than the detection threshold (*det-th*). And it terminates the tracklet when its score goes below the deletion threshold (*dlt-th*). Moreover, the tracklet's score decreases in the estimation step by a constant value (*score-decay*) and, if it is matched with a detection, it increases by the *score update function*. In this case, tracklets consistently matched over time will have high scores, and unmatched tracklets' scores will decline. To the best of our knowledge, [7] was the only work that used *confidence-based tracking*. They added the detection's and tracklet's scores to update the tracklet's score. Our work will show that their *score update function* performance is poor, and in the best cases, it will work like the *count-based* method. Consequently, we will show that *confidence-based* MOT outperforms *count-based* MOT if we employ proper *score update functions*.

We propose various score update functions. Our functions lead to more stable tracking scores, which eventually lead to better performance. We tested our method on both nuScenes [8] and Waymo [7] datasets with multiple detectors and filtering algorithms. Results showed that the *confidence-based* method with score refinement consistently provides higher AMOTA and MOTA results than *count-based* methods with our proposed functions.

### II. RELATED WORK

# how to kill a tracklet



(a) CenterPoint++

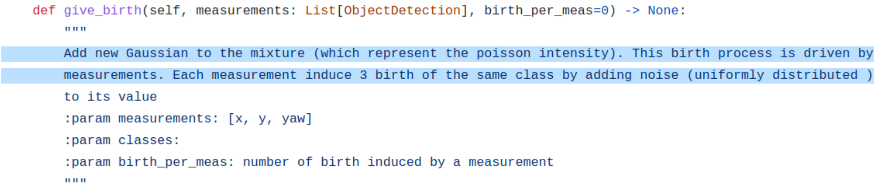
(b) Immortal Tracker

# PMBM adjustment

So unlike a standard PMBM filter, we incorporate the detection confidence score into the update step of **objects detected for the first time**. For detections with confidence scores larger than a threshold, we generate a potential new target by adding **a new Bernoulli process**, and plug the negative logarithm weight in the right m × m blocks diagonal in cost matrix L discussed in Section IV-C. For detections with lower confidence score, since we are not certain about their existences and require more evidences from the future, an undetected track with PPP density is generated for each of them

# PMBM adjustment

Add new Gaussian to the mixture (which represent the poisson intensity). This birth process is driven by measurements. Each measurement induce 3 birth of the same class by adding noise (uniformly distributed )



```
def give_birth(self, measurements: List[ObjectDetection], birth_per_meas=0) -> None:
    """
    Add new Gaussian to the mixture (which represent the poisson intensity). This birth process is driven by
    measurements. Each measurement induce 3 birth of the same class by adding noise (uniformly distributed )
    to its value
    :param measurements: [x, y, yaw]
    :param classes:
    :param birth_per_meas: number of birth induced by a measurement
    """
```

# RFS has a lot to contribute NOT in association but in track management

- how to strike a balance between Immortal track and Early Termination
- silent maintanance of the tracklet

# Incorporating Detection Score provided by the detector

- two stage association
- counters for the track