COMAP: A SYNTHETIC DATASET FOR COLLECTIVE MULTI-AGENT PERCEPTION OF AUTONOMOUS DRIVING

Y. Yuan[1], M. Sester[1]

[1] Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany - (yuan, sester)@ikg.uni-hannover.de

Commission II, WG II/3

KEY WORDS: Collective Perception, Point Cloud, Simulation, Data Fusion, 3D Object Detection, Semantic Segmentation, Uncertainty Estimation.

ABSTRACT:

Collective perception of connected vehicles can sufficiently increase the safety and reliability of autonomous driving by sharing perception information. However, collecting real experimental data for such scenarios is extremely expensive. Therefore, we built a computational efficient co-simulation synthetic data generator through CARLA and SUMO simulators. The simulated data contain image and point cloud data as well as ground truth for object detection and semantic segmentation tasks. To verify the superior performance gain of collective perception over single-vehicle perception, we conducted experiments of vehicle detection, which is one of the most important perception tasks for autonomous driving, on this data set. A 3D object detector and a Bird's Eye View (BEV) detector are trained and then test with different configurations of the number of cooperative vehicles and vehicle communication ranges. The experiment results showed that collective perception can not only dramatically increase the overall mean detection accuracy but also the localization accuracy of detected bounding boxes. Besides, a vehicle detection comparison experiment showed that the detection performance drop caused by sensor observation noise can be canceled out by redundant information collected by multiple vehicles.

1. INTRODUCTION

Autonomous driving promises to release humans from tedious driving tasks and increase traffic efficiency, for example, by platooning and car-sharing. However, there are still many security problems waiting to be solved. The source of such issues comes either from the incompleteness, such as occlusions and limited field-of-view (fov), and imperfection, such as observation noise of the data collected by sensors or from the uncertainty introduced by data processing such as the black-box property of machine learning.

Many algorithms have been developed to aggregate and fuse the data collected by different sensors of the same vehicle (Fayyad et al., 2020) to address the first issue. However, they can not efficiently resolve the occlusion problem based on the fov of a single vehicle. To this end, smart-intersection (Niels et al., 2019; Yang et al., 2020a) is proposed to mitigate occlusion by sharing the Bird's Eye View (BEV) observation from infrastructure with traffic participants. This method is simple and efficient for intersections with busy traffic but not applicable to all driving scenarios considering their static property. Therefore, we propose COllective Multi-Agent Perception (COMAP), a distributed perception system that is based on vehicle-to-vehicle (V2V) scenarios and shares information across vehicles. Each autonomous vehicle is regarded as an agent in the V2V network, hence the term Multi-Agent. Data fusion of COMAP can happen in different stages of data processing. According to the processing stages, the methodologies of COMAP can be categorized into three types: raw data fusion (RDF), deep feature fusion (DFF), and fully processed data fusion (FDF).

Collective perception is not a new concept, it has been researched for about one decade. However, most of the developed methods under this context fuse the fully processed data, such as predicted objects (Obst et al., 2014; Allig and Wanielik, 2019; Miller et al., 2020). In the recent two years, along with the maturation of different high-performance perception algorithms and simulators for autonomous driving, some researchers start to pay attention to early data fusion (RDF, DFF) for collective perception in order to further improve the accuracy and reliability of perception system. Cooper (Chen et al., 2019a) fuses raw point cloud data based on the KITTI (Geiger et al., 2012) and the T&J data set. KITTI is a popular data set for different tasks of autonomous driving, but all data are collected by sensors mounted on a single vehicle. T&J is a data set the authors collected at a parking plot with two Lidars (Velodyne VLP-16 Puck). Cooper realized collective perception by fusing consecutive frames of point clouds and detecting the static vehicles on the side of the road. Using these data sets, they further investigated the benefit of collective perception in F-Cooper (Chen et al., 2019b) by fusing voxel features that contain the information aggregated from all points in a voxel and by fusing spatial features learned by a Deep Neural Network (DNN). Both Cooper and F-Cooper showed that cooperative perception with early fusion can significantly improve the detection performance compared with single-agent perception. However, they did not compare early data fusion with late fusion (FDF). Instead, (Marvasti et al., 2020) compared fusion performance in all three data processing stages by using the Volony data set which is a synthetic data set the authors produced by CARLA simulator (Dosovitskiy et al., 2017). In their work, it is proved that early data fusion outperforms late fusion by a large margin, especially when GPS noise exists. Besides, they also showed that DFF is more robust to GPS noise than RDF. V2Vnet (Wang et al., 2020) also conducted experiments to compare the performance of joint perception and prediction with these three fusion strategies similar to (Marvasti et al., 2020) based on the simulated data from LidarSim (Manivasagam et al., 2020). Results in this work also show that collective perception can significantly improve object detection performance, and they drew the same conclusion as (Marvasti et al., 2020) when comparing the performance between the three types of fusion strategies.

As discussed above, the data sets used for COMAP scenarios are either real data (KITTI, T&J) or simulated data (Volony, LidarSim). Same as KITTI, all currently available open-source data sets for autonomous driving collected from real scenes are based on a single-agent perception system. Under the concept of COMAP, these data sets can only be used to investigate static objects on the roadside or on a parking plot similar to T&J, which is different from the real driving scenarios of COMAP. On the other hand, it is extremely expensive to collect real data for COMAP with a fleet of autonomous vehicles, hence the compromise of generating such data with simulations. LidarSim (Manivasagam et al., 2020) extracts and fuses the 3D dynamic objects and the 3D static maps from real point cloud data of several cities and then randomly place these dynamic objects back to a selected scene so that they can perform ray casting over this new 3D scene to generate new point clouds. To make the generated point clouds more realistic, they further produced deviations for points using a DNN. However, this simulation method can not provide image data which is critical for some vision problems that the point-clouds-only system can not solve sufficiently, such as detecting road debris, identifying smoke reflections of laser beams. Moreover, data generated by virtual simulators, such as CARLA, can also be transferred to a realistic domain through DNNs. Volony is a data generator based on CARLA under the concept of COMAP. They use the ”autopilot” mode of CARLA to make the vehicles drive in the city in random behavior. However, we found that it is hard to scale up the simulation by introducing more CAVs (Connected Autonomous Vehicles) and sensors to bigger towns in CARLA with only limited computational resources so that the traffic density can reach a specific level similar to busy cities. Furthermore, the ”autopilot” mode makes the vehicles perfectly aligned to the lanes and waypoints (3D-directed points in drivable road areas) of CARLA, which is very unrealistic. Therefore, we perform CARLA-SUMO co-simulation in order to introduce the more realistic routing mechanics of SUMO (Simulation of Urban MObility (Lopez et al., 2018)) to CARLA and navigate the vehicles so that they can drive in a certain small area of a town to increase traffic density with a limited number of CAVs.

Based on the simulation data, two object detection experiments are performed to show the significant benefit of COMAP because object detection is an indispensable task of the perception system of autonomous vehicles. According to the network architecture, object detectors can be classified into single-stage detectors (Simon et al., 2018; He et al., 2020; Zheng et al., 2020) and two-stage detectors (Shi et al., 2020a; Shi et al., 2020b; Chen et al., 2019c). Single-stage detectors run faster because of fewer layers in their network architecture when two-stage detectors generate more accurate bounding box predictions benefiting from their box-refining process in the second stage. The low accuracy of bounding box regression is found to be related to the misalignment of object localization and classification confidence, so CIA-SSD (Zheng et al., 2020) introduced one more IoU (Intersection over Union) regression head to the detector and use the predicted IoU scores to rectify the classification confidence. Since CIA-SSD performs similar to the state-of-the-art two-stage detectors but much faster, we adopt this network for our first experiments of 3D object detection on point clouds. Besides, we conduct the second experiment with Bird’s Eye View (BEV) representation of point clouds as the input of the network. Detectors based on this format of the input are also prevalent for autonomous driving because they can avoid 3D convolutions, which makes them more computationally efficient. PIXOR (Yang et al., 2018) is a single-stage detector that takes BEV representations as input but different from the detectors mentioned above because of its proposal-free feature. This feature again spares the time for generating the anchor box hypothesis and assigning the anchors to the ground truth boxes to calculate the regression targets. Therefore, we use this model for the second experiment.

The contributions of this paper are as follows:

- First, we developed a data generator for COMAP, which can generate both images and point cloud data with ground truth for both object detection and semantic segmentation
- Second, this data generator is very efficient and can be easily scaled up for dense traffic scenarios without dramatically increasing the computational resources.
- At last, we conducted object detection experiments on the generated data set to show the benefit of COMAP over single-agent perception in the perspective of detection accuracy, bounding box localization accuracy and the robustness against sensor noise.

2. METHOD

2.1 Simulation

For the co-simulation, the main task of CARLA is to generate sensor data. We choose CARLA map ”Town05” (figure 1), a urban map with many cross junctions and ”T” junctions, as the simulation world because vehicles are more probable to be occluded by the buildings in this scenario and COMAP aims to solve this problem. Each CAV in CARLA is mounted with a *RGB camera*, a *semantic segmentation camera*, and a *semantic LiDAR sensor*. It is well known that the data generated by these simulated sensors have a big domain shift comparing to the data generated by real sensors. However, in this paper, we concentrate on researching the effectiveness of COMAP on improving the perception performance in comparison with single-vehicle perception, for which the experiments can be conducted only under the synthetic domain. Therefore the generated data are not transformed to real domain with any transfer learning technologies. Instead, we generate noisy data by building the sensor uncertainty model.

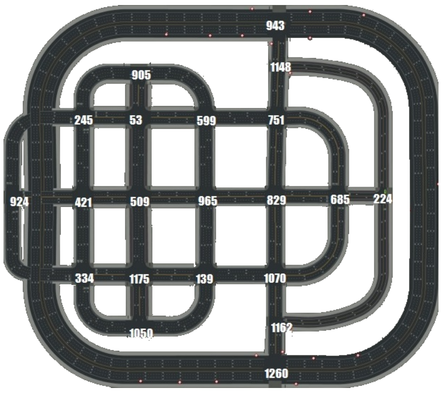

Figure 1. Map town 05 of Carla with junction IDs

For cameras, we use the default sensor parameters, such as horizontal field of view and lens parameters, from CARLA. For LiDAR, additional to the original points generated by ray-casting in the physical world, we build the sensor model with normal distributions for horizontal ($\mathcal{N} \sim (0, E_H^2)$) and vertical ($\mathcal{N} \sim (0, E_V^2)$) angular uncertainties as well as the reflection distance ($\mathcal{N} \sim (0, E_D^2)$) uncertainties. The standard deviation E_H and E_V for both angular uncertainties are defined as constant. However, according to the physical property of LiDAR, the standard deviation of LiDAR sensor noise is a distance-dependent variate defined by the equation (1), where $E_{D,min}$ and $E_{D,max}$ are the minimum and maximum standard deviation along the range dimension, respectively. The distance of a point to the origin of the LiDAR is represented as d and the maximum observation distance of the LiDAR is defined as d_{max}. In addition, we calculate the intensity of the reflected laser beam of each point with equation (2) where γ is the atmosphere attenuation rate. Based on this intensity we drop points if the sampled probabilities from a uniform distribution $\mathcal{U} \sim (0, 1)$ is smaller than the intensity I of corresponding points.

$$E_D = E_{D,min} + \frac{d}{d_{max}} \cdot (E_{D,max} - E_{D,min}) \tag{1}$$

$$I = e^{-\gamma \cdot d} \tag{2}$$

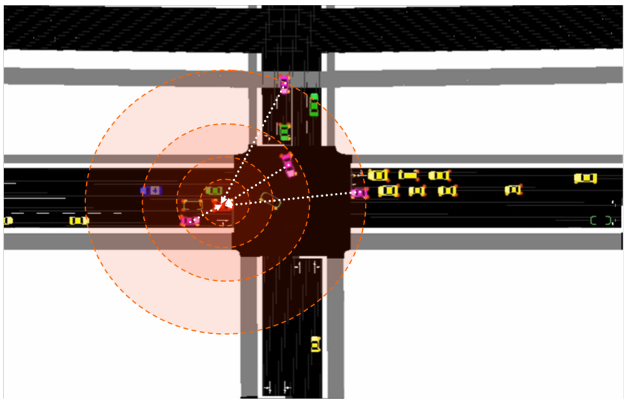

Figure 2. An example for COMAP scenario (red: ego-vehicle, magenta: data-sharing-vehicle, green: in-range-vehicles but not chosen for sharing data, yellow: not-in-range-vehicles)

SUMO is responsible for the traffic flow generation and the navigation of vehicles based on the same map transformed from CARLA "Town05" to SUMO xml format. In order to create enough meet up scenarios and traffic density for COMAP with limited number of vehicles and sensors, we run each simulation based only on one junction. An example of such simulation is given in figure 2. The vehicles randomly choose an entry road and an exit road that is connected to this junction and depart from the junction-opposite end of the entry road. Once they finish driving over the exit road, they will choose a entry and exit road of this junction again and find the optimal route back to the new entry road. This loop continues until the simulation reaches the pre-defined number of simulation frames or there is no CAV in the range of ego-CAV anymore. As shown in figure 2, although many vehicles (green and magenta) are in the communication range (e.g. 30m) of ego-vehicle (red), Furthest Point Sampling (FPS) algorithm is used to select a maximum number (e.g. 4) of cooperative vehicles in order to avoid too much redundant data. These selected vehicles are colored in magenta.

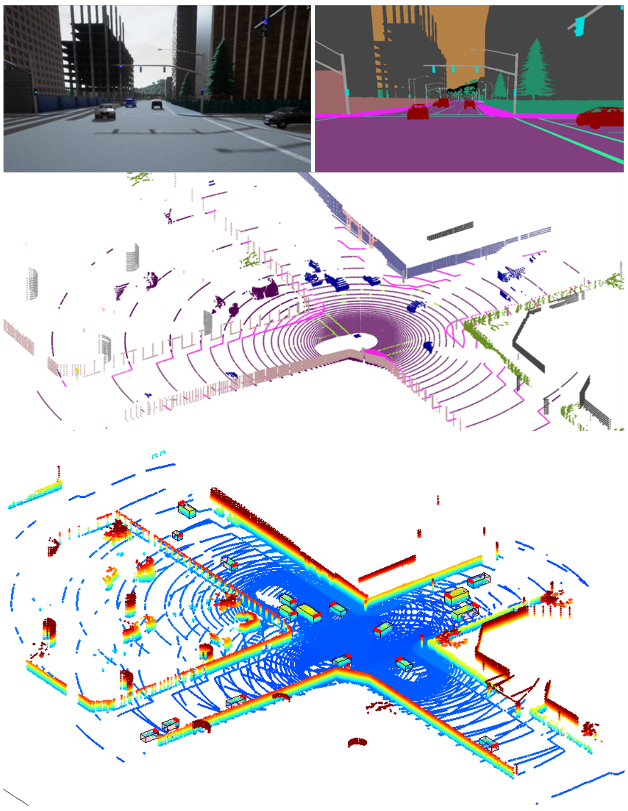

Figure 3. An example frame of the data collected by a single vehicle. Top left: RGB image; Top right: semantic segmentation label; Middle: point cloud colored by the point-wise label; Bottom: Fused point cloud data from 5 vehicles and the ground truth bounding boxes of all vehicles in the scene (height information is encoded into the point color, lines of bounding boxes are in black, the direction of bounding boxes are illustrated with a red bar on the upper front line of each box)

To obtain enough data for training neural networks, simulations are run based on 21 junctions labeled with IDs as shown in figure 1. This junction-based simulation can also be applied to any other big cities straightforwardly. An example frame of generated data is shown in figure 3. Ground truth annotations for both semantic segmentation and object detection are provided.

In the next section, we perform experiments on object detection with the generated COMAP data set, since object detection is one of the most important perception task of autonomous vehicles. For the experiments, we generate a data set with vehicle communication range of 60m with at most 10 selected cooperative vehicles that are in the range of the ego vehicle. For all simulation scenarios, 20 vehicles are randomly chosen from 21 prototypes to be spawned in the world. We use one 32-beam-LiDAR at a rotation rate of 10HZ for each vehicle because it is already sufficient enough to perform object detection through COMAP. In total, we generated 4391 frames of data. More detailed information about the simulation parameters such as sensor parameters can be found in the configuration file of our github repository[1].

[1] https://github.com/YuanYunshuang/cosense-simulation.git

2.2 Object Detection

To investigate the efficiency of COMAP based on our simulated data set, we conduct experiments to detect vehicles based on point clouds. We tested two single-stage detectors for both 3D vehicle detection with CIA-SSD (Zheng et al., 2020) and BEV vehicle detection with PIXOR (Yang et al., 2018).

CIA-SSD

CIA-SSD is the state-of-the-art 3D object detector that outperforms other single-stage detectors (Yang et al., 2020b; He et al., 2020) both in mean Average Precision (mAP) and inference time for car detection on KITTI benchmark. It takes voxels as input and uses 3D sparse convolution (Graham, 2015) to encode the input voxels into latent feature spaces. These spatial features are then be compressed from four dimensions ($C \times L \times W \times H$) to three dimensions ($C \cdot H \times L \times W$), where C is the number of features and L, W and H are the sizes of features in three spatial dimensions. Then 2D convolutions operate on these three-dimensional feature maps. Two successive convolution blocks are used to extract different abstract levels of semantic features. These two levels of features are then aggregated by weighted addition. The weights are learned by an attention fusion module constructed by convolution layers and a softmax layer. At last, several convolutional layers are attached to the head of the network to do classification and regression tasks.

We use the same network structure and loss functions as CIA-SSD, but instead of the Distance-Variant IoU-Weighted Non-Maximum-Suppression (DI-NMS) they proposed, we use the standard NMS for selecting final detected bounding boxes. Because DI-NMS is designed to reduce false-negative predictions and strong oscillations of regressed bounding boxes of distant objects in KITTI data set where the points are very sparse. However, the point sparsity in the fused point clouds of COMAP data set does not change regularly against the distance. DI-NMS would remove too many true-positive predictions and deteriorate the detection performance. Since we only need to detect vehicles, the classification is binary. In addition to bounding box regression, we follow CIA-SSD and also regress the IoUs and classify the BBox directions. The regression angles of bounding boxes are limited to $[0, \pi]$ which is direction-agnostic. The positive and negative angles are differentiated by the bounding box direction classifier.

PIXOR

With the consideration of expensive computation of high dimensionality of point clouds, PIXOR only uses 2D convolutions. Instead of projecting aggregated point features (e.g. height, intensity, occlusion) along the Z-axis to XY-plane, it divides 3D physical spaces into small 3D cells and regard height dimension as the input image channels for the 2D fully-convolutional network, which uses a backbone network of four residual convolution blocks to encode the input as well as down-sample it 16 times and then uses FPN-like (Lin et al., 2017a) structure to up-sample the feature maps 4 times by combining the low-resolution features and high-resolution features from the backbone network. The header of PIXOR contains four convolution layers followed by two separated branches of convolution layer, one for classification, one for bounding box regression. Different from CIA-SSD, which makes prior assumptions about the bounding boxes at each pixel of the (e.g. 4times) down-sampled feature maps and then classify the object category of these boxes and regress the encoded residuals between these proposed boxes and the ground truth boxes, PIXOR explicitly classifies these feature pixels similar to semantic segmentation, and regress the normalized box encodings $[cos\theta, sin\theta, log(dx), log(dy), log(l), log(w)]$ for each pixel, where θ, dx, dy, l, w are the bounding box orientation, the position offset to the pixel center, and the length and width of the box. However, in our experiment, we use $[cos\theta, sin\theta, dx, dy, log(l), log(w)]$ to encode box regression target in order to let the sign also play a role in the offset direction.

2.3 Experiments

Because the map is symmetric against the roads from junction ”924” to junction ”224” as shown in figure 1, we select the data generated in 12 junctions at the lower part of the map as training set and 9 junctions in the upper part as the test set. In total, we got 4655 training samples and 4013 test samples. In all experiments, we only detect the objects in ranges [-40, 70.4], [-40, 40], and [-3, 1] meters along the longitudinal:x, lateral:y and vertical:z direction of the ego-vehicle, respectively. All input points from the cooperative vehicles are transformed to the LiDAR coordinate system of the ego-vehicle. For both types of selected networks, we only train a single model for each network that can be used for all scenarios including the single-agent perception which only takes the point cloud of the ego-vehicle as input of the network, and the COMAP perception, which takes in the fused point clouds of multiple vehicles.

For CIA-SSD, we use voxel size of $[0.1, 0.1, 0.1]$, and only x, y, z coordinates are used as voxel features. In each pixel of 4 times down-sampled feature maps, we define 2 boxes of the same size (mean size of all boxes in training data set) and two perpendicular rotation angles ($0°$, $90°$) as the prior bounding boxes, also called anchors. The regression targets of bounding boxes are generated by assigning the anchors to ground-truth bounding boxes with an IoU threshold of 0.6 as matched (positive samples), 0.45 as unmatched (negative samples), and others as ignored anchors, which do not contribute to the training process. We follow the original work of CIA-SSD to use two heads of convolutional layers to do the binary classification for vehicle class and the direction of the vehicle, and two convolutional heads to regress the encoded residuals of the bounding box and the IoUs between the predicted bounding boxes and the ground truth bounding boxes. *Smooth L1 Loss* is used for regression of bounding boxes and IoUs and *cross entropy* for classifying bounding box directions. Classification head uses *focal loss* and the same hyperparameters as recommend in (Lin et al., 2017b). The losses for all these heads are normalized by the number of positive samples, but the weights for positive samples are enhanced 50 times to prevent the network from classifying all anchors as negative. The weights for balancing different losses are set to 1.0, 2.0, 1.0, and 0.2 for objectiveness classification, bounding box regression, IoU regression, and direction classification, respectively. During inference, we use a score threshold of 0.3 to select valid bounding box predictions for NMS module. The IoU threshold for NMS is set to a very small value (0.01) because two vehicles can not occupy the same spatial area. We give a small threshold to allow a small overlap because of the inaccuracy of the predicted bounding boxes.

For PIXOR, we use *cross entropy* for classification loss and normalize it by the number of all samples. The loss of positive samples is re-weighted by factor 20 to ensure that the network still learns from positive samples when the negative samples dominate. *Smooth L1 Loss* is used for bounding box regression and normalized by the number of positive samples. Only positive samples contribute to the regression losses. A re-weighting strategy is not used between classification and regression losses for this network. During the test phase, we threshold the classification scores by 0.8, and decode bounding boxes from regression values that correspond to the top-score pixels of the classification map. These decoded bounding boxes are then passed to NMS with the IoU threshold of 0.01.

We use the same data pre-processing and augmentation settings for training both networks on the original data set without adding sensor noise. In addition, we trained one more CIA-SSD model on the noisy data set by introducing sensor noise parameterized with $E_H = E_V = 0.05°$, $E_{D,min} = 0.02m$, $E_{D,min} = 0.06m$ and $d_{max} = 80m$ in order to investigate the effect of sensor noise on the detection performance. The difference between the no-noise point cloud and the noisy point cloud is shown in figure 8 in Appendix A.

In each frame, there might be a different number of cooperative point clouds. Therefore, we sample a varying number of cooperative point clouds with FPS and fuse them with the ego point cloud. The fused point clouds are then augmented by random flipping against the y-axis followed by rotation around the z-axis with a rotation angle uniformly sampled from the range $[-10, 10]$ degrees and scaling uniformly sampled from the range $[0.95, 1.05]$. During training, we remove all ground truth bounding boxes which do not have enough point observations to prevent the network from learning "ghost" objects and generating too many false positive detections during testing. We train both networks with ADAM (Kingma and Ba, 2015) optimizer parameterized with betas=[0.95, 0.999] on a single Nvidia 1080Ti GPU with constant learning rate of 10^{-4}. The batch sizes of CIA-SSD and PIXOR are set to eight and two, respectively. For all experiments, we train the networks with 50 epochs.

To investigate the influence of the number of cooperative vehicles on the fusion performance, we test the detection accuracy of different cooperative configurations on vehicle number, which differs from 0 (no cooperative vehicles available) to 4. We evaluate the mean Average Precision (AP) following the definition of Area Under Precision-Recall Curve (PR-AUC) by counting predictions at different IoU thresholds varying from 0.3 to 0.7 with an interval of 0.1. For one test of n cooperative vehicles, we calculate mAPs only on the frames that the ego vehicle has at least n cooperative vehicles. Besides, we performed one more test with the same models but the different method for selecting cooperative vehicles in order to study how the communication range influences the detection result. Instead of randomly choosing n vehicles from all cooperative vehicles, we choose n from the vehicles that are in a specific communication range of the ego vehicles. This range differs from 30m to 60m with an interval of 10m.

3. RESULTS AND EVALUATION

We first compare the overall vehicle detection performance of CIA-SSD and PIXOR. The mAP results are listed in table 1. The IoUs between predicted and ground truth bounding boxes are calculated directly over 3D boxes for CIA-SSD while they are calculated over BEV boxes for PIXOR. It is obvious in table 1 that CIA-SSD performs better than PIXOR in all test configurations of different IoU thresholds (2nd row) and the number of cooperative vehicles (1st column). This is, because CIA-SSD uses anchor-based target encoding which explicitly tells the network how to learn local features of a bounding box. Moreover, CIA-SSD is more robust against the object height shift along the Z-axis profiting from the weight-sharing mechanism of 3D convolution over all three physical dimensions. However, PIXOR still has its advantage over CIA-SSD because of its 2D representation of point clouds. This is critical for some cases for which the input dimension should be further increased by considering the sequential or temporal information, such as FaF (Luo et al., 2018) which achieves joint detection, tracking, and motion forecasting by taking successive frames of point clouds as input. Besides, we observed that some detected bounding boxes of CIA-SSD are flipped comparing to ground truth as marked with blue dashed eclipses in figure 5. This is caused by the unstable rotation angle encoding when angles are close to $-\pi$ and π. CIA-SSD classifies the rotation angles into positive and negative categories, a predicted negative angle close to $-\pi$ will be regarded as wrong if the ground truth angle is a positive angle close to π even they are nearly in the same direction. To this end, we conducted additional experiments by encoding angles with sine and cosine like PIXOR. This efficiently removed the the bounding box flipping problem but the mAP performance degraded as shown in figure 4. Therefore, we kept the original encoding of CIA-SSD with considering that orientations can be easily corrected in the vehicle tracking stage where the temporal sequence information is considered, such as FaF (Luo et al., 2018).

Network	CIA-SSD			PIXOR		
IoU	0.3	0.5	0.7	0.3	0.5	0.7
0	78.8	76.0	62.6	70.1	66.7	57.3
2	92.0	90.2	80.6	85.5	84.0	76.6
4	96.5	95.8	88.7	90.4	89.4	84.7

Table 1. Vehicles detection mAP of communication range 60m (in %). The second row defines the IoU threshold for calculating mAP. Number 0, 2, 4 in the first column indicates the number of cooperative vehicles.

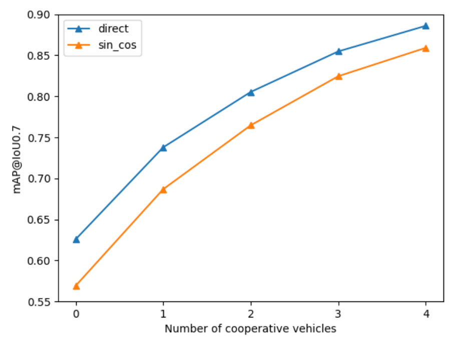

Figure 4. Performance of CIA-SSD with different bounding box target encodings at communcation range 60m. The legend "direct" indicates the original encoding by the residual of ground truth angle minus the pre-defined anchor angle, where "sin_cos" means the decoupled angle encoding like PIXOR.

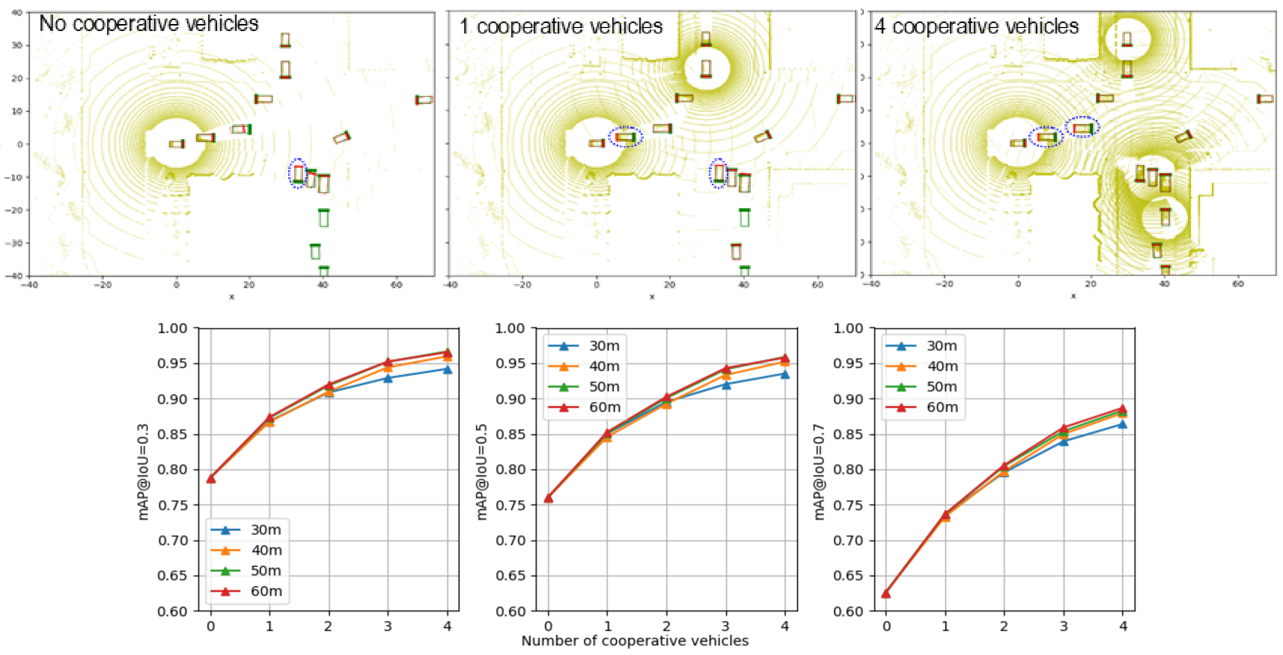

Figure 5. The vehicle detection performance of CIA-SSD. Top: detection result with/without cooperative vehicles, green indicates ground truth, red indicates predictions, the thick bar of each box tells the box orientation; Bottom: three plots show mAPs calculated with three IoU thresholds, each one shows the results with a different number of cooperative vehicles (horizontal axes, vary from 0 to 4) and communication ranges (plotted in different colors, vary from 30m to 60m)

For both CIA-SSD and PIXOR, mAPs have the same changing tendency when adjusting the IoU threshold and the number of cooperative vehicles as shown in table 1. Therefore, we only take the results of CIA-SSD as an example for further analysis as shown in figure 5. The top figures give a visual overview of the vehicle prediction results of different cooperative modes. In the top left image, three vehicles are occluded by the building on the right hand of the ego vehicle. Some vehicles are detected but with a very low IoU with the ground truth bounding box, in other words, low localization accuracy. With one cooperative vehicle, these low accurate boxes in the top left image are refined and one more vehicle is detected as shown in the top middle image. When the number of cooperative vehicles further increases to four, all vehicles are detected with a high box localization accuracy as shown in the top right image.

In the bottom row of figure 5, the horizontal axes give the number of cooperative vehicles, and 0 means that only the point cloud of the ego vehicle is used for object prediction. Each line in the figure gives the mAPs of a specific communication range and each plot shows the mAP result calculated at a different IoU threshold. In all communication ranges and at all IoU thresholds, the mAPs all increased dramatically as the number of cooperative vehicles increases. In comparison with the variant of cooperative vehicles, the communication range has a rather smaller influence on the performance. It only increased the accuracy by about 2% by increasing the communication range from 30 meters to 60 meters and the mAP gain stops at 50 meters. Therefore, we think it is beneficial to set the communication range at a lower level because it can not only save transmission power of the communication network but also increase the re-usage of radio frequency channels.

By comparing the bottom three sub-figures in figure 5, we conclude that all mAPs drop dramatically as IoU threshold increases. However, they drop less when there are more cooperative vehicles available. For example, without cooperative vehicles, the detection accuracy of the ego vehicle has dropped about 16% (from 0.79 to 0.63) where it only dropped about 7% when there are four cooperative vehicles (from 0.96 to 0.89) available. This effect can be better identified in figure 6 by evaluating the IoU scores. As expected, the changing tendency of IoU scores normalized over the number of all ground truth bounding boxes (figure 6: Top right) is identical to mAP metric. In order to investigate the localization accuracy of predicted bounding boxes, we also normalized the IoUs by the number of these predicted bounding boxes to get the average IoU of the true positive predictions. This is shown in the top left plot of figure 6. The average IoU has been dramatically improved as the number of cooperative vehicles increases. However, the communication range can barely influence the IoUs of predicted bounding boxes especially when there are enough cooperative vehicles (more than 2). This is because the growth of visible cooperative vehicles can not only increase the point observations of a specific object but also provide better special distribution of observation origins to ensure this object to be observed from different view directions. In contrast, increasing the communication range can only slightly refine the overall point distribution regarding the point density.

To further visualize the remarkable improvement of COMAP on the bounding box localization accuracy, which is also important for object tracking and motion prediction, we counted the true positive predictions that lie in different intervals of IoUs. The counted number in each interval is then normalized by the total number of the true positive predictions to obtain ratios as shown in the bottom plot of figure 6. Each color in the plot indicates a specific number of cooperative vehicles. As this number increases from 0 (blue) to 4 (purple), the ratio of the low-IoU ($0.3 \sim 0.8$) predictions all decreased and these decreased ones are replaced by the high-IoU ($0.8 \sim 1.0$) predictions hence the increased ratio of boxes in this interval. This

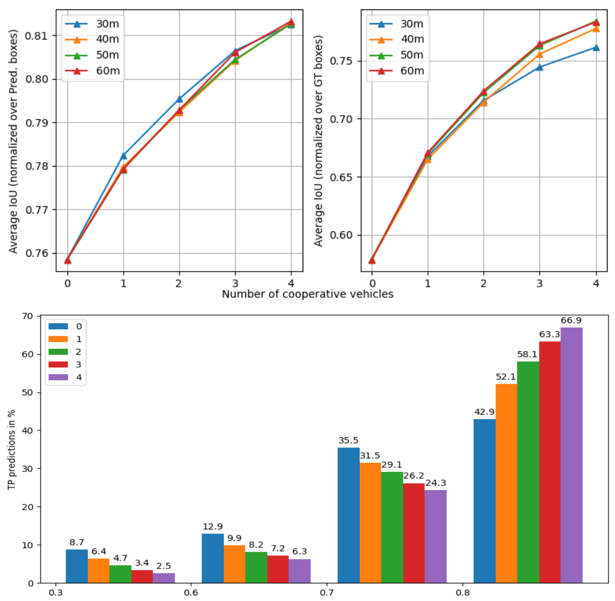

Figure 6. Influence of the number of cooperative vehicles and communication ranges on the localization accuracy (IoU) of predicted bounding boxes. Top: average IoU of all positive predicted bounding boxes with ground truth, they are normalized over the number of all true positives (left) as well as all ground truths (right); Bottom: ratios of true positive predictions in different IoU intervals for communication range 50m

means that the IoU of a detected object is less probable to be under 0.8 when increasing observing vehicles.

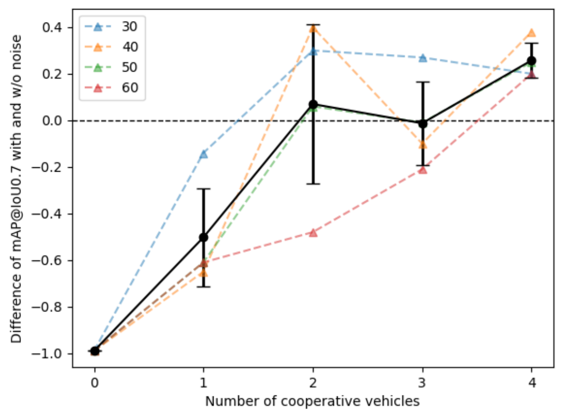

Figure 7. Influence of sensor noise on detection performance. Performance drop (mAP@IoU0.7 difference in %) of different communication ranges are plotted with the dashed line in different colors. The black solid line is the mean mAP drop of all results. Vertical bars indicate the standard deviation.

At last, we compare the detection performance on the data with and without noise. The result is shown in figure 7. Each point in the figure indicates the performance change by introducing sensor noise to the data. Negative values mean that the mAP at the IoU threshold of 0.7 has dropped, where positive values mean that the performance is improved by introducing noise. When there are no or zero cooperative vehicles, the mAP has dropped 1% (62.6% to 61.6%) because of noise. As the number of cooperative vehicles increases, the negative influence of noise on the performance of vehicle detection gradually disappeared. Starting from three vehicles, the average mAP for all communication ranges even increased with the existence of sensor noise. For example, with communication range 30m and 4 cooperative vehicles, the mAP increased from 86.3% to 86.5%. This has proved that COMAP can not only increase the detection accuracy but also suppress the sensor noise because the measurement accuracy can be increased by redundant observations.

4. CONCLUSION

In this paper, we proposed COMAP for connected autonomous vehicles and generated point cloud data as well as image data under the context of this concept through an efficient co-simulation with CARLA and SUMO. We then conducted experiments of object detection that have shown a significant performance gain of COMAP in comparison with single-agent perception. We also tested the detection performance with a different number of cooperative vehicles and different communication ranges. The results revealed that increasing the number of cooperative vehicles can dramatically improve the detection mAP as well as the box localization accuracy while enlarging the communication range can sightly improve mAP but not the localization accuracy of the predicted bounding boxes. Moreover, the comparison study of sensor uncertainty has shown that COMAP can suppress the negative influence of sensor noise on the detection performance.

All results show that COMAP can significantly increase the performance of the perception system of the autonomous vehicle. However, there are still lots of challenges that need to be overcome. In this paper, we did not consider the localization error of the vehicles which can deteriorate the fusion performance if this error is not appropriately handled. However, we see the potential of improving SLAM (Simultaneous Localization And Mapping) of autonomous vehicles by COMAP. Besides, safety is a critical issue for autonomous driving. With the simulated data set which has the innate feature of 100% certainty, this issue can be also better studied, for example, under the context of uncertainty estimation (Feng et al., 2020). Moreover, sharing visual data can also be a heavy load for communication networks. With ensuring the perception performance, better strategies are also needed for selecting as little sharing data as possible. This can be achieved by only sharing the necessary data or the data from higher processing stages.

REFERENCES

Allig, C., Wanielik, G., 2019. Alignment of perception information for cooperative perception. *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France,June 9-12, 2019*, IEEE, 1849–1854.

Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., Fu, S., 2019a. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, SEC '19, Association for Computing Machinery, New York, NY, USA, 88–100.

Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., Fu, S., 2019b. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, SEC '19, Association for Computing Machinery, New York, NY, USA, 88–100.

Chen, Y., Liu, S., Shen, X., Jia, J., 2019c. Fast point r-cnn. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

Fayyad, J., Jaradat, M. A., Gruyer, D., Najjaran, H., 2020. Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review. *Sensors*, 20(15). https://www.mdpi.com/1424-8220/20/15/4220.

Feng, D., Harakeh, A., Waslander, S. L., Dietmayer, K., 2020. A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving. *arXiv e-prints*, arXiv:2011.10671.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graham, B., 2015. Sparse 3D convolutional neural networks. X. Xie, M. W. Jones, G. K. L. Tam (eds), *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, 150.1–150.9.

He, C., Zeng, H., Huang, J., Hua, X.-S., Zhang, L., 2020. Structure aware single-stage 3d object detection from point cloud. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. Y. Bengio, Y. LeCun (eds), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.

Lin, T.-Y., Goyal, P., Girshick, R. B., He, K., Dollár, P., 2017b. Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999-3007.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using SUMO. *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE.

Luo, W., Yang, B., Urtasun, R., 2018. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3569-3577.

Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., Yang, B., Ma, W. C., Urtasun, R., 2020. Lidarsim: Realistic lidar simulation by leveraging the real world. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11164–11173.

Marvasti, E. E., Raftari, A., Marvasti, A. E., Fallah, Y. P., Guo, R., Lu, H., 2020. Feature Sharing and Integration for Cooperative Cognition and Perception with Volumetric Sensors. *arXiv e-prints*, arXiv:2011.08317.

Miller, A., Rim, K., Chopra, P., Kelkar, P., Likhachev, M., 2020. Cooperative perception and localization for cooperative driving. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 1256–1262.

Niels, T., Mitrovic, N., Bogenberger, K., Stevanovic, A., Bertini, R. L., 2019. Smart intersection management for connected and automated vehicles and pedestrians. *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 1–10.

Obst, M., Hobert, L., Reisdorf, P., 2014. Multi-sensor data fusion for checking plausibility of v2v communications by vision-based multiple-object tracking. *2014 IEEE Vehicular Networking Conference (VNC)*, 143–150.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H., 2020a. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10526-10535.

Shi, S., Wang, Z., Shi, J., Wang, X., Li, H., 2020b. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Simon, M., Milz, S., Amende, K., Groß, H., 2018. Complex-YOLO: An euler-region-proposal for real-time 3d object detection on point clouds. *ECCV Workshops*.

Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., Urtasun, R., 2020. V2VNet: vehicle-to-vehicle communication for joint perception and prediction. *ECCV*, Springer International Publishing, 605–621.

Yang, B., Luo, W., Urtasun, R., 2018. PIXOR: Real-time 3D object detection from point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7652–7660.

Yang, S., Bailey, E., Yang, Z., Ostrometzky, J., Zussman, G., Seskar, I., Kostic, Z., 2020a. COSMOS smart intersection: Edge compute and communications for bird's eye object tracking. *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 1–7.

Yang, Z., Sun, Y., Liu, S., Jia, J., 2020b. 3DSSD: Point-based 3D single stage object detector. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.-W., 2020. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. *arXiv e-prints*, arXiv:2011.08317.

APPENDIX

A: Original point cloud without noise and point cloud with noise

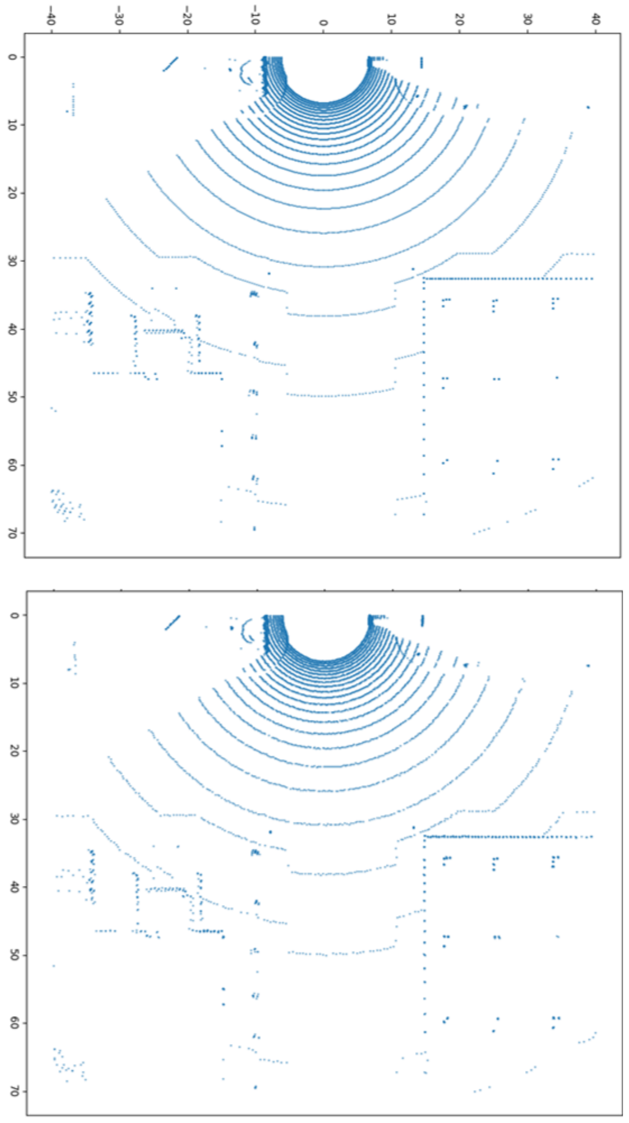

Figure 8. An example frame of point cloud data before and after adding sensor noise (only points in the front half-circle of the vehicle is visualized)