Beamforming and Scalable Image Processing in Vehicle-to-Vehicle Networks

Hieu Ngo[1, 2], Hua Fang[1, 2], Honggang Wang[1]
[1]University of Massachusetts Dartmouth, 285 Old Westport Rd, North Dartmouth, MA 02747
[2] University of Massachusetts Medical School, 368 Plantation St. Worcester, MA, 01605
hngo1@umassd.edu, hfang2@umassd.edu, hwang1@umassd.edu

***Abstract*— Vehicle to Vehicle (V2V) communication allows vehicles to wirelessly exchange information on the surrounding environment and enables cooperative perception. It helps prevent accidents, increase the safety of the passengers, and improve the traffic flow efficiency. However, these benefits can only come when the vehicles can communicate with each other in a fast and reliable manner. Therefore, we investigated two areas to improve the communication quality of V2V: First, using beamforming to increase the bandwidth of V2V communication by establishing accurate and stable collaborative beam connection between vehicles on the road; second, ensuring scalable transmission to decrease the amount of data to be transmitted, thus reduce the bandwidth requirements needed for collaborative perception of autonomous driving vehicles. Beamforming in V2V communication can be achieved by utilizing image-based and LIDAR's 3D data-based vehicle detection and tracking. For vehicle detection and tracking simulation, we tested the Single Shot Multibox Detector deep learning-based object detection method that can achieve a mean Average Precision of 0.837 and the Kalman filter for tracking. For scalable transmission, we simulate the effect of varying pixel resolutions as well as different image compression techniques on the file size of data. Results show that without compression, the file size for only transmitting the bounding boxes containing detected object is up to 10 times less than the original file size. Similar results are also observed when the file is compressed by lossless and lossy compression to varying degrees. Based on these findings using existing databases, the impact of these compression methods and methods of effectively combining feature maps on the performance of object detection and tracking models will be further tested in the real-world autonomous driving system.**

Keywords—V2V, Deep Learning, LIDAR, Beamforming (*key words****)***

I. Introduction

A. Autonomous Driving Systems

An autonomous driving system is a driving system that is capable of driving safely without human input by using a variety of sensors to perceive the environment. Human drivers normally cannot drive cars at a speed of 200-250 mph in their daily lives for the sake of safety. The speed limits are set to guarantee that human drivers have enough reaction time to stop their cars or change lanes if needed. As such, the driving speed is constrained by the reaction time of a normal human driver [1]. The autonomous driving system solves this restriction by using multiprocessor implementation of intelligent real-time control systems in the car to replace human drivers.

The greatest benefits of autonomous driving systems are increased traveling speed and more free time with little attention needed from the passengers. Additionally, autonomous driving also helps decrease traffic and reduce emissions.

However, the safety issue is still a major concern for the mass adoption of autonomous driving vehicles. One solution to this problem is the collaborative communication of vehicle to vehicle. Each vehicle can communicate with each other to provide the information of their surroundings, e.g., vehicles on the road that might be hidden from line-of-sight, obstacles on the road, or in cases where unexpected events happen. To accomplish this goal, there needs to be a stable uninterrupted highspeed connection between the vehicles.

B. Vehicle-to-Vehicle communication

Vehicle-to-Vehicle (V2V) communications are designed to transmit information without a centralized networking architecture. The objective of V2V communication is to provide communication between vehicles when there is a risk of an accident and to enable vehicles to take preventive action to avoid collisions. Therefore, to guarantee the safety of passengers inside autonomous vehicles, autonomous driving requires a higher data rate in communication between vehicles. The information communicated between vehicles can belong to many different streams of information gathered from multiple sensors on the car such as cameras, radar transceivers, as well as LIDAR. Thus, the communication data rate needs to be in the Gb/s range instead of the current Mb/s range. However, in real-world applications, such high bandwidths can be difficult to achieve. Therefore, we look into improving the bandwidth available in V2V communication as well as reducing the information to be transmitted.

First, to improve the connection power, we examine the use of beamforming in V2V communication. Traditionally, radio communication is often equipped with omnidirectional antennas, which have decreased signal quality at the receiving

vehicles due to signal power loss during transmission. To solve this issue, we can use beamforming and beam steering technology for higher channel gains. Both beamforming and beam-steering use directional antennas to avoid the signal power loss in undesired directions, which can improve the network capacity as well as the stability and quality of the V2V communication links.

To form a correct and stable beam connection in V2V communication, we need to obtain the accurate locations of the vehicles on road. One possible solution to obtain the locations of vehicles is to use the Global Positioning System (GPS). However, the localization accuracy of the GPS signals can be affected by the surrounding environment, such as buildings, weather, forests, and many other sources of interference. Therefore, in this paper, we aim to integrate the use of deep learning-based object detection and tracking techniques for the localization and tracking of vehicles to establish stable wireless V2V communication.

Furthermore, as the number of vehicles within a radius of 50 meters can reach 20 or 30 during rush hour, and the amount of data to be transmitted can become impossible to manage. Thus, we focus on scalable transmission, where the data is compressed in a lossy compression manner such that the reliability and safety of the system can remain the same.

C. *Scenarios of collaborative communication between multiple vehicles*

The main objective of this research is to understand the communication challenges to achieving cooperative perception among autonomous vehicles, and thus increased the safety of autonomous driving. One of the major challenges of autonomous driving is the problem of blind spots and occlusion, which restricted by the line of sight and field of view of autonomous vehicles. Therefore, with a cooperative perception among autonomous vehicles, each vehicle can have a more complete perception of the environment that is normally prevented when an occlusion occurs. However, to achieve this goal, we need to establish a stable connection between the vehicles to allow a channel for the exchange of information. For this reason, we propose the use of object detection and tracking from a multisensory system to assist in the beamforming process. The object detection and tracking system use the information from the camera, LIDAR, and radar sensors as the base to identify and locate other vehicles on the road. This location information can then be used to form mmWave connection beams and steer these beams towards the predicted/tracked location of these vehicles.

Particularly, we examine the effectiveness of the beamforming and beam-steering V2V system communication in the following two scenarios using vehicle detection and tracking techniques. In a single vehicle to vehicle scenario, the 2 vehicles can travel in the opposite direction as shown in Fig. 1. On the other hand, the two vehicles can also travel in the same direction and one of the vehicles is traveling at a faster speed. Both vehicles have cameras that can record the environment around the car, radar transceivers such as the Universal Software Radio Peripheral (USRP) on top of the roof for communication, and LIDAR sensors for precise and fast depth information. The information collected from these sensors helps with collaborative communication and autonomous driving of the vehicles by integrating deep learning-based object detection and tracking to help with beamforming and steering. For camera information, we use image-based object detection and tracking to locate and predict the directions of where other objects are from the current vehicle position. Additionally, using LIDAR sensors, we can gather the depth information, which can be used to further detect and track vehicles precisely during traffic conditions. These tracking results give us a prediction of vehicle position in real-time traffic conditions and let us move the beam connection preemptively to maintain a stable communication condition.

Thus, the process of collaborative communication includes detecting vehicles and their locations on the streets via a multitude of data such as image-based object detection or LIDAR's 3D object detection, tracking these vehicles during traffic conditions, and finally collaborative perception and communication between vehicles using these locations. Additionally, due to the high bandwidth required for this collaborative communication system, we propose scalable transmission, which compresses and reduces the amount of data during communication while guarantees the reliability and safety of the autonomous driving system.

II. VEHICLE DETECTION

Vehicle detection aims to identify the location and size of the vehicle. It is the starting point before the tracking tasks can be applied.

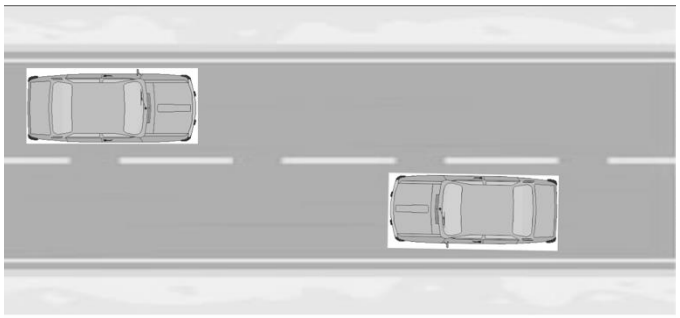

Fig. 1. Scenario 1

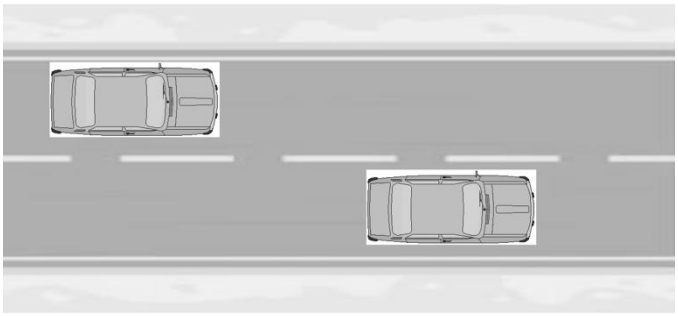

Fig. 2. Scenario 2

A. *Image-based vehicle detection*

Symbol	Description
$\boldsymbol{x_{pred}}$	Predicted state
$\boldsymbol{P_{pred}}$	Predicted State covariance
$\boldsymbol{P}$	State covariance
$\boldsymbol{F}$	State transition matrix
$\boldsymbol{Q}$	Process covariance
$\boldsymbol{B}$	Control function
$\boldsymbol{u}$	Control input
y	Residual
$\boldsymbol{H}$	Measurement function
$\boldsymbol{z}$	Measurement from the object detector
$\boldsymbol{x'}$	Update state
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{F}$	File size
$\boldsymbol{w}$	Image width
$\boldsymbol{h}$	Image height
$\boldsymbol{b}$	Image bit depth
$\boldsymbol{Z_n}$, $\boldsymbol{Z_n}$	DCT input and output

TABLE I

Object detection has been one of the key abilities in most computer and robot vision systems. Especially during recent years, there has been rapid and successful development and breakthrough in the field of computer vision, which is largely Neural Networks (CNNs).

The goal of object detection is to identify the location and scale of all object instances, such as vehicles, that are present in an image. Thus, a vehicle detector job is to detect all vehicle instances regardless of their scale, location, pose, view with respect to the camera.

In most object detection systems, the sliding window scheme is used. In a sliding window scheme, an exhaustive search is applied to detect the objects appearing in the image at different scales and locations [2]. A classifier will determine whether a given image patch corresponds to the object or not. In this scheme, because the classifier works for a given scale and patch size, the classier is usually used to classify all possible patches of a given size, for each of the downscaled version of the input image. One alternative to the sliding window scheme is the use of bag-of-words [3], which can be used for verifying the presence of the object and then iteratively refine the image region that contains the object. Another alternative is to sample image patches and iteratively search for the region of an image where the object is likely to stay [4]. Additionally, the detector can find key-points and match them to perform the detection [5]. The latter three schemes reduce the computation complexity of the classification by seeking to avoid an exhaustive search of all image patches. However, they cannot always guarantee that all object's instances will be detected, which is not desirable for the safety of the autonomous driving system.

Current state-of-the-art object detection methods rely on Deep Convolutional Neural Networks (DCNNs). Within DCNN, there exist two popular frameworks for detection [6]. The first is the single stage detection framework that uses a single network to produce object detection locations and class predictions simultaneously. The second is the region proposal detection framework that uses two stages, one to propose the general regions of interest and one to categorize them by a separate classifier network. Typically, the region proposal methods have higher performance on the detection benchmarks at the cost of high computational complexity and hard to implement or fine-tune. On the contrary, the single stage detection methods are generally faster with low memory cost while still achieving competitive performance [6]. Due to this reason, single stage detection methods are very suitable for real-time autonomous driving systems. Some of the popular single stage detection models are YOLO (You Only Look Once) [7] and Single Shot Detector (SSD) [8]. Of the two, the SSD method performs faster and has competitive results on public benchmarks.

B. Vehicle detection from LIDAR

Compared to image-based object detection, LiDAR's point cloud can be used to locate the objects and their trajectories even better due to the precise depth information. Similar to object detection in an image-based application, we focus on object recognition and classification, especially of other vehicles, using LIDAR's point cloud data. The popular method in LIDAR object detection is first separating the non-ground objects from the ground, and then classifying the vehicles against other non-ground objects using supervised learning techniques [9]. For example, Hernandez and Marcotegui [10] apply the λ-flat zones labeling to the 2D rasterized point cloud image to generate a ground mask, which makes the scan holes detected as non-ground objects. After this, they used SVM to classify cars, lampposts, and pedestrians. In the case of occlusions from the point cloud data, Xiao et al. [11] proposed a complete reconstruction and localization of vehicles by modeling the vehicles then reconstructing them in case of occlusions. In this approach, the point cloud data is first classified into the ground and non-ground objects similar to the previous method. After reconstruction, SVM and Random Forest are used to classify the objects into vehicles and non-vehicles. This classification is based on geometric features such as size and shape as well as the parameters of the models that fit an object. However, in the current model-based approaches, the training models are generated manually for limited types of objects and not generic.

C. Dataset for autonomous driving system testing:

The KITTI dataset [12, 13] is one of the most popular benchmark datasets for autonomous driving perception tasks, including image-based monocular and stereo depth estimation, optical flow, semantic, and instance segmentation, as well as 2D and 3D object detection. The data is collected from a car equipped with an Inertial Navigation System (GPS/IMU), a laser scanner, 2 grayscale cameras, 2 forward-facing cameras, and a LIDAR sensor. The location of the collected data is in Karlsruhe, Germany. In this dataset, each training example is a labeled 3D scene that is captured by the LIDAR sensor and the two front-facing cameras. In total, there are 7481 training examples and 7581 testing examples in this dataset. Each training example contains a 100 milliseconds snapshot of the 3D world around the car. The LIDAR sensor used in this dataset

is the Velodyne HDL-64E LIDAR sensor. This LIDAR has 64 channels (64 laser beams) and an azimuth resolution of 0.08 degree, which means that the generated point cloud photos are images with 64 rows and 4500 columns. In KITTI, the point cloud data is an unordered set of LIDAR point where each point is a 4-tuple (x, y, z, p) where x, y, z are the cartesian coordinates and p is the intensity.

For object detection training and testing, we use the Pascal Visual Object Classes Challenge 2007 (VOC07) [14], 2012 (VOC12) [15], and the Common Object in Context (COCO) [16] dataset. The PASCAL VOC project aims to provide standardized image data sets for object class recognition, a common set of tools for accessing the datasets, and enable evaluation and comparison of different methods. Similarly, COCO is also a large-scale object detection, segmentation, and captioning dataset. This dataset has over 1.5 million object instances and 330,000 images with all objects appearing in their natural contexts. Both of these datasets are widely popular in the field of object detection.

Fig. 3. AnnieWAY car used to collect the KITTI dataset[12]

III. Collaborative Communication

V2V communication imposes a high bandwidth requirement due to the plethora of information collected from the multisensory system attached to each vehicle. This bandwidth requirement is further enlarged by the current trend towards Vehicles-to-everything (V2X), which allows vehicles to communicate with other moving parts of the traffic system around them. Thus, the use of beamforming should be a promising solution to this problem.

Beamforming is a technique that focuses the wireless radio signal toward a specific direction, rather than having them spread in all directions from the broadcast antenna, thus can result in a faster and more reliable connection compared to the scenario without beamforming. The beam is formed by combining multiple antennas transmitting the same signal and reinforcing the waves in a specific direction. To achieve this, there needs to be multiple antennas at proximity, all broadcasting the same signal at a slightly different time on the transmitter's end. The beam width depends on the number of antennas, where the more antennas, the narrower the beam. The advantages of beam connections are that they are faster, more reliable, and provide more secure connections [17].

In an autonomous driving system, the obvious obstacles for beamforming are the rapid changes of beam directions due to relative positional change between the vehicles. To resolve this, the use of collaborative perception and multisensory fusion between vehicles is necessary to achieve vehicle tracking [18]. When the car can be tracked accurately, we can use this information for the radio beam formation as well.

The principle of vehicle tracking is based on the prediction of the current vehicle position based on the previous position. Due to the complex and high-speed scenarios of autonomous driving, estimating location alone is insufficient and may lead to low connection power. Therefore, the driving systems also need to estimate the direction and velocity of the vehicles such that a dynamic motion model can be applied to track the vehicles and predict their future location/trajectory. The most suitable data option for this task is the 3D LIDAR sensors, which capture the precise depth information in the 360-degree area around the vehicle. However, concerning the uncertainties of sensor functions in a complex system, a better solution is to employ a sensor fusion strategy that has the potential to reach a higher accuracy and better reliability.

Typical tracking systems work by associating data from the same class together (bounding box) then applying filtering methods [6]. For data association, in image-based object detection, these are usually done by DCNNs that produce bounding box and objects' classification. In the LIDAR's 3D points cloud case, the nearest neighbors methods can be used for establishing an association between data points. Additionally, point density or Hausdorff distance can also be used for data association in points cloud [19, 20].

For filtering, the Kalman filtering method is a popular method of tracking objects/vehicles. With Kalman filter, the vehicle tracker can predict the car's future location, correct the prediction based on new measurements, reduce the noise introduced by inaccurate detection, and facilitate the process of association of multiple vehicles with their tracks. This Kalman filter is also the filter method used in our simulation.

There are two steps in using the Kalman filter for vehicle tracking: prediction and update. The prediction step uses previous states to predict the current state. The update step uses the current measurement, which is the bounding box location in this case, to correct the state. In the prediction phase, the Kalman filter calculates the predicted state using eq. (1) and the predicted state covariance using eq. (2).

$$x_{pred} = F\bar{x} + Bu \quad (1)$$

$$P_{pred} = FPF^T + Q \quad (2)$$

Where $\bar{x}$ is the state mean, P is the state covariance, F is the state transition matrix, Q is the process covariance, B is the control function (matrix), and u is the control input. In this context, the state is the bounding box coordinates. The control

inputs can be used when there are known properties that can better estimate the system's state. However, in the case where there are no known control inputs, we can assume the control inputs u = 0.

In the update phase, the residual y is calculated using eq. (3):

$$y = z - Hx_{pred} \quad (3)$$

Where z is the measurement from the object detector, and H is the measurement function (matrix). The updated state x' is calculated using eq. (4):

$$x' = x_{pred} + Ky \quad (4)$$

Where the updated state x' is the summation of the predicted state and the Kalman gain (K) from the residual; and K is calculated as follow:

$$K = P_{pred}H^T(HP_{pred}H^T + R)^{-1} \quad (5)$$

where R is denoted as the measurement noise covariance, P_{pred} and H are the predicted state covariance and the measurement function respectively.

Finally, the state covariance is also updated using eq. (6):

$$P = (I - KH)P_{pred} \quad (6)$$

However, for a nonlinear model, it is better to use particle filter-based vehicle tracking, which is a generalization of the traditional Kalman filtering methods. The particle filter uses a set of discrete particles to approximate the distribution of the object, meaning that each particle tests the likelihood that the object is at the position where the particle is. At each generation, the good particles are multiplied and the bad particles are removed. In this way, the particle filter can better model non-linear object motion better than the Kalman filter.

Once a good tracking system can perform well in the autonomous driving system, the information of the system can also be used in the formation of wireless radio beam connections between vehicles.

IV. Scalable transmission

The effectiveness of autonomous vehicles depends largely on their sensors. These sensors allow the vehicles to see and sense everything on the road and collect information needed for safe drives. On top of this, the information collected from the sensors, such as road conditions, upcoming intersections, traffic jam, and obstacles on the road, need to be shared between vehicles on the roads. As such, this information can be processed and analyzed to map out the vehicles' path from the starting point to the destination, as well as the appropriate steering, turning, accelerating, and braking instructions to the controls of the car. However, it is not always possible to communicate this information in a fast and precise manner between vehicles, especially in an autonomous system with multiple vehicles collaborating together. The solution we propose for this problem is scalable transmission, which integrates lossy compression of a multitude of data between vehicles, which significantly reduces the amount of data exchanged among vehicles while maintaining the important information crucial to the safety of the passengers and the traffic flow efficiency.

In autonomous driving system, the two criteria that the vehicles must allow collaboration are: (a) the autonomous vehicles have a wide variety of sensors that provide a comprehensive recognition of the surrounding environment; (b) the vehicles also have a stable connection to communicate this information for better collaboration with other vehicles on the road.

The first criterion is usually satisfied in real-world applications. A typical autonomous vehicle has multiple radar sensors, LIDAR unit, and camera units around the body to guarantee a complete capture of the surrounding environment. These sensors are usually sufficient enough for the perception of the environment. Furthermore, they also enable precise cooperative perception of autonomous vehicles. Such cooperative perception is extremely helpful to extend the line of sight and field of view of autonomous vehicles, which otherwise suffers from blind spots and occlusions. The first step toward this cooperation is transmitting the information from all of these sensors from vehicles A to vehicles B, C, D which are driving on the same road. This transmission, however, requires very high bandwidth. In real-life applications, such bandwidth may not be available all the time due to the unstable connection between vehicles which is caused by environmental conditions, which can cause a lack of bandwidth.

To resolve this problem, we propose the use of scalable transmission. Scalable transmission's main goal is to minimize the information during transmission while still maintaining the necessary information for the safety of the passengers in autonomous vehicles. Such examples can be seen in online video streaming, when the bandwidth is not good enough, the quality of the video automatically reduces to low resolution. The same principle can be applied toward vehicle to vehicle communication in the collaborative autonomous driving system. When an environmental change happens such as rainy days or the change of line-of-sight connection to non-line-of-sight connection, the communication can become unstable and unable to transmit the high-resolution version of the sensor data. In this case, instead of transmitting a lossless version of the data in a noisy channel, we can transmit a lossy compressed version of the data that is relevant to the other vehicles to reduce the bandwidth requirements while still maintaining the appropriate information for them to operate safely. For example, in the case of LIDAR or radar, we can transmit only the key feature, such as the object information. While in the camera sensors case, we can transmit only part of the video that is relevant. Particularly, we would transfer only the box that contains the detected object, which is the vehicles being tracked, to the other collaborative vehicles in the network. Additionally, while transmitting part of the information is not possible, the vehicle can also transmit fewer frames instead. Thus, in good condition, all frames from the sensors can be transmitted between vehicles. However, in a suboptimal condition, the vehicles can choose to transmit fewer frames or parts of the image relevant to the other vehicles to significantly reduce the amount of data exchanged among vehicles.

The benefits of data compression are obvious, as it enables agile and precise cooperative perception on connected and autonomous vehicles. Due to a smaller bandwidth requirement, scalable transmission allows data sharing on a massive scale

among autonomous vehicles that can help with a safer driving as well as the overall traffic flow efficiency of the network.

V. Simulation

A. *Simulation of car detection from a mounted camera on vehicles:*

In this simulation, we apply a vehicle detection algorithm on a short video recorded from a front-facing camera attached to the car. The first step of the simulation is to input a captured image into the vehicle detection algorithm. The algorithm will output the image with the bounding boxes on the detected vehicles. The tool we used is the TensorFlow Object Detection API, an open-source framework built on top of TensorFlow to construct, train, and deploy object detection models. The model for object detection is the Single Shot Multibox Detection (SSD) framework that comes from the collection of pre-trained object detection models in the API. The training data comes from the COCO dataset. The SSD method detects objects in images using a single deep neural network. In particular, it discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. Then, at prediction time, the network scores the fitness of each object category in each default box and gives adjustments to the box to better match the object shape. Additionally, the network also combines prediction from multiple feature maps and different resolutions to account for the various sizes of detected objects.

The performance of the TensorFlow implementation of the SSD models is listed in table 1 below [21].

TABLE I. SSD MODEL PERFORMANCE

Model	Training Data	Test Data	mAP
SSD-300	VOC07+12 trainval	VOC07 test	0.778
SSD-300	VOC07+12+COCO trainval	VOC07 test	0.817
SSD-512	VOC07+12+COCO trainval	VOC07 test	0.837

The SSD-300 is a VGG-based SSD network with 300 inputs whereas SSD-512 takes 512 inputs. The architecture of the SSD-300 is described in Fig. 5, where the input is the captured image of the road, and the output is the bounding boxes of the detected vehicles. The training data comes from the Pascal Visual Object Classes Challenge 2007 (VOC07), 2012 (VOC12), and the COCO dataset. The testing dataset comes from the VOC07 testing data. Some examples of the outputs produced by this model are presented in Fig. 4. The mean Average Precision (mAP) measures the accuracy of the prediction of the bounding box. The mAP is calculated as the mean of the Area Under Curve (AUC) of the precision-recall curve overall categories. In the case of this dataset, there is no distinction between AP and mAP. The mAP metric is widely used to assess the performance of object detection model. As listed in the table above, the SSD method can achieve a mAP score of up to 0.837. There are other models that can perform even better than the SSD model, such as the DetectorRS model [22] or the SpineNet-190 [23]. However, the SSD provides a better tradeoff between accuracy and the detection speed, which is crucial in the autonomous driving application.

Based on this simulation, we confirm the possibility of using object detection and tracking in V2V beamforming. The first reason is that the autonomous driving system requires the application of object detection and tracking regardless of its potential utility in beamforming. Without these detections, the vehicle is unable to recognize and react to other vehicles on the road, disabling it to maintain safety for passengers. Thus, there is no computational overhead needed in finding the beam direction as these detections are already running constantly during a trip. The only calculation needed is to compute beam directions, which are fairly simple and can be performed with basic geometry. Secondly, from the literature survey and in simulation, object detection and tracking model proved to have good performance with a mAP score of up to 0.837 and a fast inference speed. Coupled with the rapid improvement of commercial GPU hardware and embedded systems that increase the computation power in autonomous vehicles, the accuracy and inference speed of these models will continue to improve in the near future.

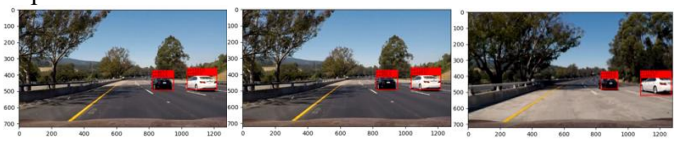

Fig. 4. Simulation of image-based Car detection and tracking

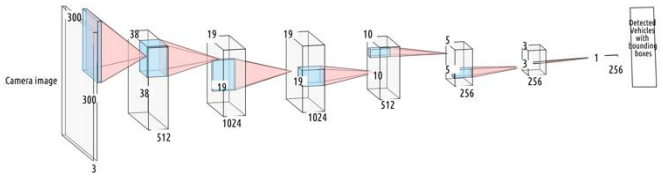

Fig. 5. SSD VGG-based model diagram [8]

B. *Scalable transmission of camera information:*

In the other direction, we propose scalable transmission that aim to reduce the amount of data in V2V communication. To illustrate the effectiveness of scalable transmission, we simulate the transmission using the video from the KITTI Vision Benchmark Suite as aforementioned. In this benchmark dataset, the image resolution of the video is 1382 x 512 pixels while the frame rate is 10 frames per second. We can calculate the size of one image with this resolution using:

$$F = w \times h \times b \quad (7)$$

Where F is the file size in bits, w, h are the width and height of the image respectively, and b is the bit depth of the camera. Using this formula, the total number of bits of one image in the KITTI dataset is 11,321,344 bits or 1.35 Megabytes. Thus, the file size of a 1-hour long video recorded on the car is 3036.6 Megabytes (3 Gigabytes). In Fig. 4, we examine the effect of the size of the vehicle's bounding box towards the file size. From Fig. 3, we can safely assume that the typical size of the bounding box ranges from 100x100 to 300x300 whereas the size of the whole frame is 1382x512. Thus, we can see that compared to sending each frame in raw data, sending only the

important information from the image, such as the vehicle's position, requires 10 times less bandwidth.

Additionally, we also analyze the effect of compression algorithms on the size of the camera data. The first compression technique is the JPEG method. JPEG is a lossy compression format that was created for digital images. It works by first converting the RGB-based image collected from digital camera to the YCbCr color space, then applying the Discrete Cosine Transform (DCT) that transforms the frequencies of the original values along each row and column in terms of a sum of cosine function oscillating at different frequencies. The DCT is expressed as:

$$Z_k = \sum_{n=0}^{N-1} Z_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \quad k = 0, \dots, N-1 \qquad (9)$$

This transformation enables the higher frequencies to be minimized or zeroed out, which is important for the compression of the image. The compression of the data happens in the quantization step, where more higher frequencies coefficients are zeroed out. The strength of the compression, thus, depends on the strength of the quantization matrix. The more compression the more information will be lost in this process. Finally, the final matrix is encoded using the Huffman-Coding, which reorder the data such that the lower spatial frequencies come before the higher spatial frequencies. Because the higher frequencies are very likely to be zeroed out after compression, the image can store them in "10x0" instead of "0 0 … 0" and reduce the amount of data significantly. The strength of this compression varies from 10:1 with little perceptible loss in image quality to 500:1 with very poor quality. In this experiment, we tested the two ratios of 10:1 and 50:1.

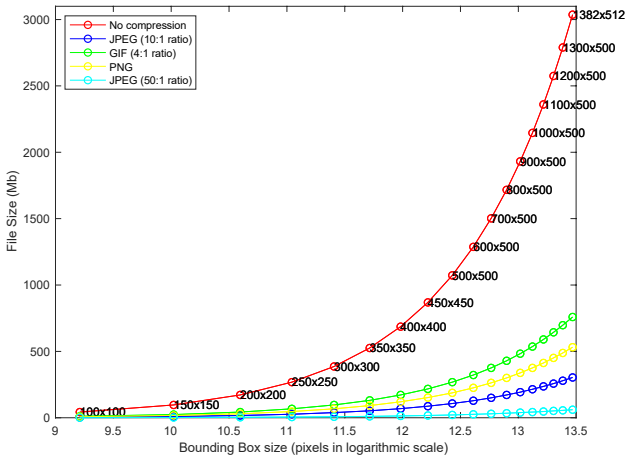

Fig. 6. The effect of transferring part of the image and compression techniques on the file size of an 1-hour long KITTI video

Another very popular method of compression is the Graphic Interchange Format (GIF) method. Unlike JPEG, GIF is a lossless compression method that keeps the quality of the image when the image has less than 256 colors. However, if the image has more than 256 colors, GIF is also a lossy compression. The technique used in GIF compression is the Lempel-Ziv-Welch (LZW) lossless data compression technique [24]. Using the LZW compression algorithm, the GIF image is compressed in two ways. First, it replaces the commonly occurring patterns in the image, such as a large area of uniform color, and saves them in a dictionary. Secondly, it also reduces the number of colors in color-rich images by approximating the colors in the image using the nearest color to represent each pixel or "error diffusion" to adjust the nearby pixels to account for the error in each pixel. The typical compression ratio for GIF compression format is 4:1 to 10:1. In this experiment, we assume the compression rate to be in the lower end with the 4:1 ratio.

The third compression technique that we analyzed is the Portable Network Graphics (PNG) compression. PNG is a lossless compression method that looks for patterns in the image such that it can compress these patterns into a dictionary, similar to GIF. The first of the two steps in the PNG compression process is the delta encoding filtering process, where a pixel is represented by its relation to the neighboring pixels. The encoded value is the difference between x and the predicted value based on the neighboring pixels. Therefore, when the value of the pixels has a low difference, the image is transformed into having lots of duplicate, low values, and making it much more compressible. In this second stage, the LZ77 compression algorithm compresses the encoded data. Similar to the LZW algorithm, the LZ77 algorithm is also a dictionary coder. It replaces the repeated occurrences of data with references to the existing earlier copy in the input. The match occurrences are encoded by a pair of length-distance numbers, which describe the length of the match and the distance to its previous appearance. The effectiveness of PNG compression is usually 10-30% better than GIF.

Based on Fig. 6, we see that even in the case of image compression, sending only part of the image instead of the whole frame can still reduce the bandwidth requirement by up to 5 times. In real-world scenarios, this method can save a tremendous amount of data transmission. For example, using camera data, instead of sending the whole picture frame with likely unhelpful background, we can send only the bounding boxes that contain the detected vehicles, shrinking the size of the data needed in communication by up to 10 folds.

C. Scalable transmission of LIDAR information:

A point cloud dataset can contain hundreds of millions or even more points with geometric, colorimetric, and radiometric attributes [6]. Thus, it can require a great amount of bandwidth to allow for a cooperative driving system. Currently, we are mostly using traditional compression system to compress the LIDAR's data. For example, using lossless compression tool such as GZIP [25], the power of compression can achieve up to 10:1 ratio. The level of compression can also be adjusted based on the tradeoff of application requirements and the computing time. Similarly, other compression methods such as BZIP2 [26], LZMA [27], and LZ4 [28] can all be applied and adjusted with different compression levels and computing time. Nevertheless, more investigation is needed for the tradeoff of the effect of lossy compression and the computing time-compression level in object detection and tracking using LIDAR sensor information.

To reduce this, instead of transmitting the raw point cloud data, we propose that vehicles utilize object recognition and

classification and group the discrete points in the points cloud into objects. Thus, the process of transferring data can be framed to a small subset of the data, which significantly reduces the required bandwidth. The size of the data can be further reduced by modeling these objects, such that the transmission can be completed on the simplified model rather than the bulky point cloud files. In this case, the transmitted data only contain the important information (e.g. road obstacles, other vehicles) and remove all noises and unwanted objects.

VI. Conclusion

In this paper, using existing databases our goal is to provide a stable and reliable method for V2V autonomous system communication. Thus, we investigated two areas: increasing the bandwidth available during communication and decreasing the amount of data in transmission.

To increase bandwidth, we propose the use of object detection and tracking in beamforming. Particularly, we utilize the application of object detection and tracking using camera and LIDAR data to establish accurate and stable collaborative beam connection between vehicles on the road. The results show that the current object detection and tracking technology provide competitive performance with the mean Average Precision of 0.837 and fast inference speed. Additionally, this process requires minimal overhead due to the nature of object detection and tracking in the autonomous driving system.

To decrease the amount of data during transmission, we propose scalable transmission to decrease the amount of data to be transmitted, thus reduce the bandwidth required for the collaborative perception of autonomous driving vehicles. Our simulations showed that transferring only (moving) part of the images can reduce the data size by 10 times with no compression, or 2 to 5 times with image compression techniques. Similarly, with LIDAR points cloud data, it is possible to transfer only the points associated with the detected vehicles to decrease data transmission.

Along this line of research, in future work we aim to develop a novel method of effectively combining feature maps from multiple sensor information, test its performance on object detection and tracking models in the real-world autonomous driving system.

Acknowledgement

This research was partly supported by NSF ECCS # 2010366 to Drs. Wang and Fang.

References

[1] Dong, Z., Shi, W., Tong, G., & Yang, K. (2020). Collaborative Autonomous Driving: Vision and Challenges.

[2] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International journal of computer vision, vol. 104, no. 2, pp. 154–171, 2013.

[3] Tsai, Chih-Fong. "Bag-of-words representation in image annotation: A review." *International Scholarly Research Notices* 2012 (2012).

[4] L. Liang, C. Liu, and H. Y. Shum. Real-time texture synthesis by patch-based sampling. *Technical Report MSR-TR-2001-40, Microsoft Research*, 2001.

[5] Eltanany A.S., SAfy Elwan M., Amein A.S. (2020) Key Point Detection Techniques. In: Hassanien A., Shaalan K., Tolba M. (eds) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019. AISI 2019. Advances in Intelligent Systems and Computing, vol 1058. Springer, Cham

[6] Yurtsever, Ekim, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. "A survey of autonomous driving: Common practices and emerging technologies." *IEEE Access* 8 (2020): 58443-58469.

[7] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," Dec. 2015.

[9] Che, Erzhuo; Jung, Jaehoon; Olsen, Michael J. 2019. "Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review." *Sensors* 19, no. 4: 810.

[10] Hernández, J.; Marcotegui, B. Filtering of artifacts and pavement segmentation from mobile lidar data. In Proceedings of the 2009 ISPRS Workshop on Laser Scanning, Paris, France, 1–2 September 2009.

[11] Xiao, W.; Vallet, B.; Schindler, K.; Paparoditis, N. Street-side vehicle detection, classification and change detection using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 2016, 114, 166–178. [

[12] Andreas Geiger, Philip Lenz, and Raquel Urta-sun, ""Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: Conference on Computer Vision and Pattern Recognitioon (CVPR). 2012

[13] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: International Journal oof Robotics Research (IJRR). 2013.

[14] M. Everingham et al. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. https://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[15] M. Everingham et al. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. https://www.pascal-network.org/challenges/VOC/voc2012/

[16] Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham

[17] Ali, E., Ismail, M., Nordin, R. *et al.* Beamforming techniques for massive MIMO systems in 5G: overview, classification, and trends for future research. *Frontiers Inf Technol Electronic Eng* **18,** 753–772 (2017). https://doi.org/10.1631/FITEE.1601817

[18] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al., "Autonomous driving in urban environments: Boss and the urban challenge," Journal of Field Robotics, vol. 25, no. 8, pp. 425–466, 2008.

[19] J. Lambert, L. Liang, Y. Morales, N. Akai, A. Carballo, E. Takeuchi, P. Narksri, S. Seiya, and K. Takeda, "Tsukuba challenge 2017 dynamic object tracks dataset for pedestrian behavior analysis," Journal of Robotics and Mechatronics (JRM), vol. 30, no. 4, Aug. 2018.

[20] M. . Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in Proceedings of 12th International Conference on Pattern Recognition, vol. 1, Oct. 1994, pp. 566–568 vol.1.

[21] P. Balanca, SSD: Single Shot MultiBox Detector in Tensorflow. SSD-Tensorflow. https://github.com/balancap/SSD-Tensorflow. 2017.

[22] Qiao, Siyuan, Liang-Chieh Chen and Alan L. Yuille. "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution." *ArXiv* abs/2006.02334 (2020): n. pag.

[23] Zoph, Barret, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk and Quoc V. Le. "Rethinking Pre-training and Self-training." *ArXiv* abs/2006.06882 (2020): n. pag.

[24] Ziv, J.; Lempel, A. (1978). "Compression of individual sequences via variable-rate coding". *IEEE Transactions on Information Theory*. **24**(5): 530. Doi: 10.1109/TIT.1978.1055934

[25] DEUTSCH, Peter. GZIP file format specification version 4.3. 1996.

[26] BURROWS, Michael et WHEELER, David J. A block-sorting lossless data compression algorithm. 1994.

[27] PAVLOV, Igor. 7-Zip. URL http://www.7-zip.org/, 2012

[28] COLLET, Yann, et al. Lz4: Extremely fast compression algorithm. URL http://lz4.github.io/lz4/, 2013