Improving Long-Range 3D Object Detection Methods for Autonomous Box Trucks Using Sensor Fusion

Ashutosh Bhown

University of Michigan

Engineering Honors Program

I. The Problem

The supply chain has experienced more stress than ever before due to driver shortages and increased demand of e-commerce due to the COVID-19 pandemic [1]. Thankfully, autonomy is showing to be an increasingly prevalent solution to this problem [2]. Automating delivery vehicles allows for operation of supply routes without a driver and allows for more deliveries over a given amount of time. There are several companies competing in the "long-haul" stage, automating 18-wheeler trucks carrying goods across interstate lines from manufacturers to warehouses, and several companies competing in the "local delivery" sector, autonomously delivering goods from retailers to consumers' residences with cars or smaller bespoke robots. However, nobody has automated the transportation logistics from warehouse to retailer, otherwise known as the "middle mile". That is, until Gatik AI was founded in 2017.

Gatik AI is the first autonomous trucking startup to focus on automating "middle mile" business-to-business delivery with driverless box trucks. They are currently operating with multiple partners such as Walmart in Arkansas, Sam's Club in Dallas, and Loblaw in Toronto, frequently travelling on urban and suburban streets. Because other autonomous vehicle companies are either focused on automating large trucks primarily on highway driving or smaller autonomous vehicles for local delivery in urban and suburban areas, Gatik is the first company to bring autonomous trucks into regular operation within urban and suburban areas, and this comes with a unique challenge that this team was asked to help solve.

II. The Problem

Trucks are large. They require more space and time to perform the same actions that a car would do (lane changes, turns, etc.). As such, it is important that the driver of the truck is able to

see far ahead to provide more time to make these decisions and then execute them, especially if there are vulnerable road users, or VRUs, in the vicinity of the truck. VRUs are entities that can be encountered on or near roads that do not have the protection of a motor vehicle or other form of protection in the event of an accident; these include pedestrians, cyclists, skateboarders, wheelchair users, traffic conductors, and road workers, to name a few examples [3].

This translates over to Gatik's autonomous box trucks. Because they operate in urban and suburban areas, they must be able to detect VRUs at a longer range than a smaller autonomous vehicle would due to their large size. Long-range 3D object detection methods to date have been used in the context of autonomous trucks mostly for detecting cars and not necessarily VRUs due to said trucks operating primarily on highways. Therefore, long-range 3D object detection with a focus on VRUs has not been a heavily explored topic within research; Gatik is one of the first companies to tackle this unique problem within autonomy.

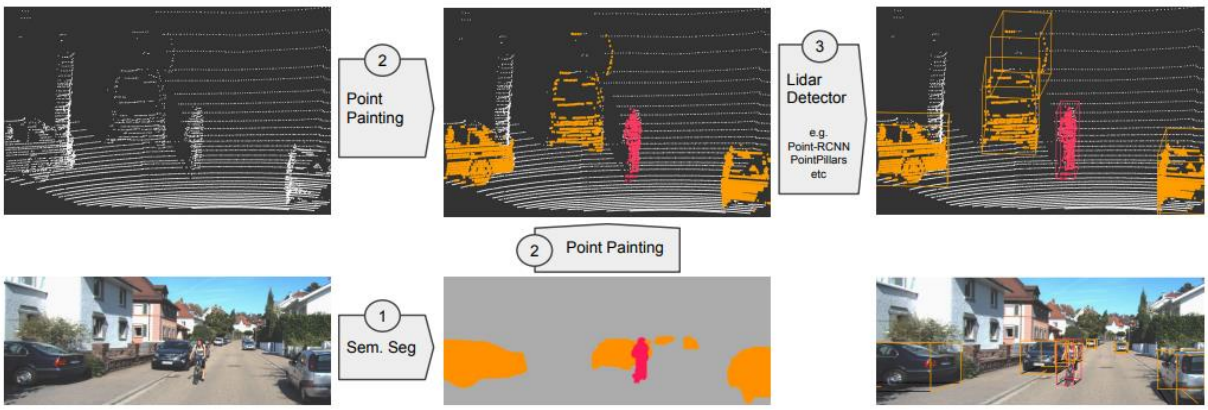
Over the course of the Winter 2022 and Fall 2022 semesters, this team worked on researching 3D object detection methods that can be applied to long ranges, specifically focusing on improving detection of VRUs (primarily pedestrians and cyclists) in the hopes of finding better ways for Gatik's autonomous box trucks to detect VRUs sooner and have an appropriate amount of time to make and execute decisions as a result, improving the efficiency, and more importantly, the safety of their autonomous box truck operations.

III. Solution Approach

The key to detecting 3D objects, especially at long range, is sensor fusion. It is important to have multiple modalities of sensing, in order to have high fidelity data that can withstand inclement conditions. In the context of this project, the sensor fusion is between LiDAR sensors and

monocular cameras. LiDAR uses laser projections to provide incredibly high-quality data about precise locations and distances of objects in an autonomous vehicle's surroundings. However, at long ranges, the point clouds generated by LiDAR become very sparse, and thus less information can be garnered from it. This is where monocular cameras come in: data from RGB images are rich with features that can be extracted to "fill in" the gaps of the LiDAR data and provide a more complete picture of what is happening around the vehicle.

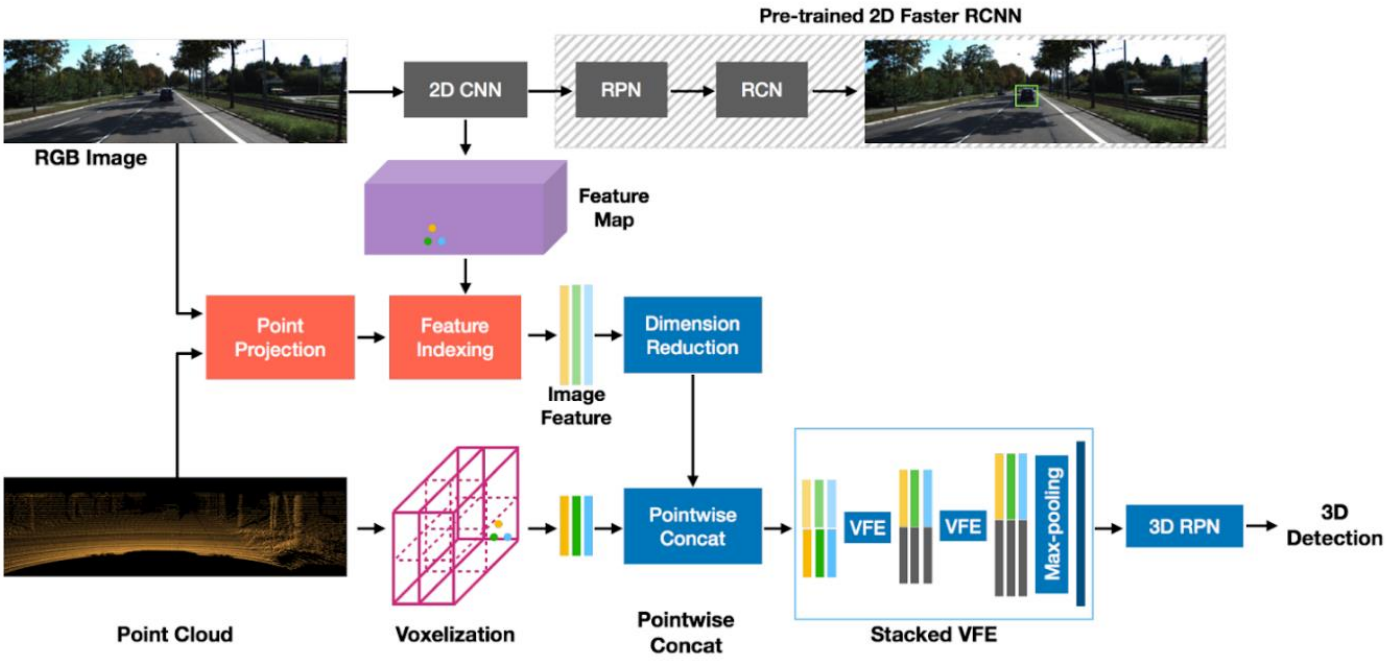
The general idea of LiDAR and sensor fusion techniques is to "enhance" one of the sensors' data with the data of the other sensor and then pass that enhanced data into a processing method to produce bounding boxes for the objects in the scene to be detected, such as cars, pedestrians, and bicyclists. For the purposes of this experiment, the selected method enhanced the LiDAR point cloud with features of the image and then the point cloud is processed to generate bounding boxes for pedestrians, bicyclists, and cars for the 2D images in the test dataset.



This was done through the PointPainting technique (visualized above), which first separates the different classes (cars, pedestrians, etc.) in the image through semantic segmentation, and then

"paints" that information onto the LiDAR point cloud points, hence the name "PointPainting". This enhanced LiDAR point cloud is then processed and bounding boxes are generated.

Due to the complexity of the codebase, however, we migrated about halfway through the project to MVXNet, a sensor fusion architecture that fuses camera and LiDAR data similarly to PointPainting, but with a codebase far easier to understand and modify.



The MVXNet architecture is similar to PointPainting in that it uses a neural network to extract features from the 2D RGB image, enhance the point cloud with that information, and then process the improved point cloud to get bounding boxes as a result. With this architecture, we intended to attain our goal of improving detection of pedestrians and cyclists at long range by modifying this baseline architecture to perform better at this task. The MVXNet architecture is available in the MMDetection3D repository, which hosts a collection of object detection architectures that can be experimented with, trained, and tested on any dataset the user desires.

IV. Setup For Development and Augmentation

The computational setup involved a high-power Amazon Elastic Cloud Computing (EC2) instance, which had a GPU for improved training and testing performance for our architectures, so that anyone could access the computer and develop code, train, and test at any time. To make the MMDetection3D setup easier to move around and/or duplicate for experimentation, it was containerized via Docker. One Docker container was used for "control" setup, which contained the stock MVXNet architecture within MMDetection3D, and an experimental container was used for experimentation, where the augmented MVXNet architecture would be developed.

V. Evaluating Architectures

While using Gatik AI's data recorded from routes run on their vehicles would be a better choice of dataset for this project, it was not properly packaged for training and testing usages (ground truth labels, etc.), so in the interest of time the best choice was a publicly available dataset. We elected to use the KITTI dataset for training, validation, and testing of our architectures.

The KITTI dataset is one of the most popular datasets in the computer vision field in the subcategory of detecting objects in road scenes, and it contains 7,424 images split across three difficulty levels: easy, medium, and hard, where an increased difficulty corresponds to objects in scenes further away from the vehicle's sensors, increased occlusion between the sensors and the objects in the scene, and increased truncation of objects. To focus on the "long-range" aspect of this project, we focused on performing well in the "hard" difficulty for 3D detection.

VI. Evaluating Architectures: Training and Testing

The KITTI dataset has 7,424 images in its training set to train architectures such that they can "learn" patterns and be able to make their own decisions (in this case, predicting agents in road scenes). It also has 3,769 unique images for its testing set, for evaluating the architecture's performance after it "learns" from the training set on images that the architecture has not seen yet. When training, the architecture trains on the training set multiple times, and each iteration is known as an "epoch". MMDetection3D by default setting trains its architectures for 40 epochs. This process on our EC2 instance would take approximately 30 hours as the architecture would train on an image 300,000 times in total. Testing only requires predictions to be made on the test set once, and this process would take approximately five minutes.

VI. Evaluating Architectures: Performance Figures

How can the performance of an object detection architecture be quantified? One of the most popular methods is calculating how well prediction bounding boxes overlap with ground truth bounding boxes. A measure for this IoU, or "Intersection Over Union", which, as the name implies, equals the volume of the bounding boxes' intersection divided by the volume of their union. The IoU is calculated by comparing a number of points between the prediction and ground truth bounding boxes; 11 points and 40 points are commonly used, we chose to use 40 points in our calculations. If the IoU is greater than or equal to a certain threshold value, it can be considered a correct classification. For this experiment, we chose a threshold value of 0.5. From these values, an overall mean average precision can be calculated, denoted as mAP; this is a common figure chosen to represent object detection networks on a scale of 0-100.

Augmented vs. Stock MVXNet Results

Augmentations were made to the MVXNet architecture. **Details of the augmentations made may not be discussed due to NDA**.

After testing a trained stock MVXNet architecture and a trained augmented MVXNet architecture, below is a comparison of their performance scores. Note that these are performance figures for the 3D/Hard detection performance subsection of the KITTI dataset, using 40 points for calculating IoU, with an IoU greater than or equal to 0.5 counting as a correct classification.

TABLE I. Performance Comparison of Stock/Augmented MVXNet Architectures

Class	Stock MVXNet mAP	Augmented MVXNet mAP
Ped.	61.69	**73.28**
Cyclist	51.60	**53.94**
Car	87.60	**90.09**

As can be seen from Table I, there is an improvement of detection performance across all classes, but especially so with pedestrians. Approximately two points of improvement are observable across the Car and Cyclist classes, and the Pedestrian class sees an improvement of over 11 points, a significant improvement. One question that may arise is why cyclist detection does not see as significant improvement as compared to pedestrian detection. There are a number of factors that could contribute to this; the two most prevalent are likely the quantity of cyclists in the dataset compared to pedestrians and bounding boxes being larger with cyclist detection; while this decreases mAP it can be seen as providing an increased “safety margin” for planning.

Below are three example scenes for visual results, with a base image, the stock MVXNet architecture's predictions, and the augmented MVXNet architecture's predictions. All prediction bounding boxes are shown in blue.

Scene I. Intersection

Base Image | Stock MVXNet | Augmented MVXNet

The base image of this scene on the left details an intersection with several pedestrians and cyclists crossing the street as well as a stopped car at said intersection. The stock MVXNet architecture only detects the stopped car, and none of the vulnerable road users. However, the augmented MVXNet architecture detects each of the pedestrians and cyclists. It is crucial the VRUs are detected such that the autonomous truck running this software can stop sooner and leave a greater margin of safety.

Using the augmented MVXNet architecture would result in detection of these VRUs and safer behavior of the truck, whereas the stock MVXNet architecture would not detect them, resulting in potentially riskier behavior and possibly increased danger to the VRUs in the intersection.

Scene II. Lone Cyclist

Base Image | Stock MVXNet | Augmented MVXNet

The base image of this scene on the left depicts nothing in front of the vehicle on the road aside from a lone cyclist on the side of the road. The stock MVXNet architecture does not detect the cyclist, but the augmented MVXNet architecture does.

With the stock architecture not detecting the cyclist, the truck believes it is free to drive however it wants in a legal manner, much like a human would. However, if there is a VRU detected, it must behave safer (i.e. slowing down, larger margin between the vehicle and VRU) to maximize the safety of the VRU. As the augmented MVXNet architecture ensures this happens but the stock architecture does not, this helps prove the augmented MVXNet architecture is better for long-range VRU detection.

Scene III. Parked Cars w/ Pedestrian in Street

Base Image | Stock MVXNet | Augmented MVXNet

The base image of the above scene depicts several parked cars and a pedestrian in the street.

The stock MVXNet architecture detects the parked vehicles, but not the pedestrian. The augmented architecture detects both the parked cars and the pedestrian.

Similarly to the previous scenario, a lone VRU is present in this scene and the autonomous vehicle detecting said VRU plays a critical role in their safety. The stock architecture, not detecting the VRU, would lead the truck into thinking there is nothing in front of it except for parked cars, potentially resulting in not enough time to detect and stop for the pedestrian as it gets closer. The augmented architecture's detection of the pedestrian further away allows more time for the truck to slow down, increasing safety margins for the VRU.

Challenges

These promising results did not come without challenges. A number were encountered over the course of the project, but two stand out as the most significant. First was our immense difficulty finding adequate computational resources during the first half of this project. Initially, we were provided a GPU laptop to develop these architectures on, but training a network on this architecture once would take several days, which was absolutely not practical. We searched for other sources of higher computational power. One option we looked at was the University of Michigan Great Lakes service, which although powerful, was shared among university clubs, classes, and so on, resulting in immense queues and limited time for individual jobs. Thankfully, Gatik provided a high-power Amazon EC2 instance with a GPU to use freely for training, testing, and development. Even still, however, it was not perfect. Complexity of the augmented network had to be reduced in some areas to fit the augmentations such that the architecture could still be trained and tested with GPU memory constraints, and even with powerful computing instance, training these large architectures on the KITTI dataset took ~30-36 hours on average.

Secondly, the KITTI dataset itself is not perfect. Many scenes look quite similar in terms of background and have different elements in the environment, so it could benefit from diversity. More importantly, however, the dataset is missing some ground truth labels.

Ground Truth Labels

Augmented MVXNet Predictions

Above is a scene from the visual results presented previously; the orange labels are ground truth and in blue are the augmented MVXNet predictions. The ground truth labels do not have all cyclists and pedestrians nor do they have one of the parked cars to the right side included. However, the augmented MVXNet detects all these. These predictions that visually are correct are not included in ground truth and thus may be reported as extraneous during evaluation, therefore the performance figures previously shown may in truth be underreporting the true performance of the architecture.

Future Work

This is only the first step to solving the long-range VRU detection problem. Next steps for future work into this include modifying different parts of the architecture, increasing the computational power of the host machine to reintroduce more complexity and learning ability to the model, and even porting this software to hardware in the loop to simulate performance on an actual truck.

Works Cited:

[1]. G. Friesen, “No end in sight for the covid-led global supply chain disruption,” *Forbes*, 14-Apr-2022. [Online]. Available: https://www.forbes.com/sites/garthfriesen/2021/09/03/no-end-in-sight-for-the-covid-led-global-supply-chain-disruption/?sh=49a4a7db3491. [Accessed: 15-Dec-2022].

[2]. “Autonomous vehicles could solve the U.S. truck driver shortage,” *Bloomberg.com*, 17-Nov-2021. [Online]. Available: https://www.bloomberg.com/opinion/articles/2021-11-17/autonomous-vehicles-could-solve-the-u-s-truck-driver-shortage?leadSource=uverify+wall. [Accessed: 15-Dec-2022].

[3]. “# 147 National Safety Council position/policy statement vulnerable road ...” [Online]. Available: https://www.nsc.org/getattachment/d5babee6-582d-4e66-804f-8d06f9b021a4/t-vulnerable-road-users-147. [Accessed: 16-Dec-2022].