ROADSIDE LIDAR VEHICLE DETECTION AND TRACKING USING RANGE AND INTENSITY BACKGROUND SUBTRACTION

Tianya Zhang, Ph.D., Candidate

Department of Civil and Environmental Engineering

Rutgers, The State University of New Jersey

500 Bartholomew Rd,

Piscataway, NJ 08854-8018, United States

Email: tianya.zhang@rutgers.edu

Peter J. Jin, Ph.D., Associate Professor,

Department of Civil and Environmental Engineering

Rutgers, The State University of New Jersey

500 Bartholomew Rd,

Piscataway, NJ 08854-8018

Email: peter.j.jin@rutgers.edu

ABSTRACT
In this paper, we present the solution of roadside LiDAR object detection using a combination of two unsupervised learning algorithms. The 3D point clouds data are firstly converted into spherical coordinates and filled into the azimuth grid matrix using a hash function. After that, the raw LiDAR data were rearranged into spatial-temporal data structures to store the information of range, azimuth, and intensity. Dynamic Mode Decomposition method is applied for decomposing the point cloud data into low-rank backgrounds and sparse foregrounds based on intensity channel pattern recognition. The Triangle Algorithm automatically finds the dividing value to separate the moving targets from static background according to range information. After intensity and range background subtraction, the foreground moving objects will be detected using a density-based detector and encoded into state-space model for tracking. The output of proposed model including vehicle trajectories that can enable many mobility and safety applications. The method was validated against a commercial traffic data collection platform and demonstrated to be an efficient and reliable solution for infrastructure LiDAR object detection. In contrast to the previous methods that process directly on the scattered and discrete point clouds, the proposed method can establish the less sophisticated linear relationship of the 3D measurement data, which captures the spatial-temporal structure that we often desire.

INTRODUCTION

Light Detection and Ranging (LiDAR) is a high-precision sensor that uses a laser transmitter and receiver to detect the distance of surrounding object and provide a 3D information of the environment. The LiDAR sensor can meet the requirements for most scenarios, particularly suitable for moving object detection and localization. More recently, the LiDAR sensor has gained escalating traction for smart city and connected infrastructure applications such as intelligent intersections to ensure the pedestrian and bicycle safety, parking and construction management, and drone-based traffic monitoring, etc. LiDAR technology is very useful for object motion detection especially under some low light conditions, as it can see the surrounding environment in both day and night. Compared to radar sensor technologies, LiDAR sensor can provide high-resolution results while radar does not have enough resolution. Although the accuracy of LiDAR sensor could be impacted by certain scenarios due to phantom reflections, for instance, the fog weather, the LiDAR sensor is very reliable for data collection in every lighting for multimodal traffic monitoring system. LiDAR sensors generate data for scene depth understanding, whereas the camera-based system lacks the capability to directly generate precise depth estimation. Another advantage of LiDAR sensor is that the 3D point cloud data do not introduce any privacy concerns which is a significant matter for security purpose. The detection results will be used for real time traffic signal optimization to reduce pedestrian/cyclists' waiting time and protect vulnerable road users at the signalized intersection. With LiDAR data collection tool, the traffic manager can learn the mobility patterns, understand causes for non-recurrent or recurrent congestion. Connected Vehicle applications also rely on real-time data acquisition capability to enable Connected Vehicle (CV) applications through V2X (Vehicle-to-Everything) communications to address safety and mobility challenges.

The usage of LiDAR technologies has also been questioned whether the LiDAR sensor has enough value to be a good infrastructure investment. LiDAR is viewed as a complementary sensor to the camera or radars for the connected infrastructure solution. Critics have argued that the LiDAR sensor is a crutch to the Vision Zero of future traffic. In general, combing different sensors will increase the reliability and deliver beneficials to the analyzing layer. It also serves as a non-intrusive approach to provide sufficient coverage. If powered with next generation communication network, the 3D data can be accessed in the real time to enable many safety-critical applications as well as cyber-physical modeling for computational decision making.

The majority of the LiDAR-based object detection models were developed for self-driving cars. Infrastructure-based LiDAR has its unique characteristics compared to mobile LiDAR or airborne LiDAR that could lead to different approaches for object detection. First, the infrastructure-based LiDAR were installed on a higher position covering wide detection area, while the on-vehicle LiDAR is usually mounted on the roof of vehicle. Secondly, the object detection task only pertains to a small portion of data in a fixed background, as the roadside LiDAR contains mostly static background infrastructure. On the contrary, the mobile LiDAR sensor captures all input data to recognize various types of objects while being carried through complex environments. Based on the unique scenario of infrastructure-based LiDAR sensor, a background modeling approach is proposed in this paper. Compared to existing LiDAR object detection model, the proposed models are more efficient and require less computational resources. The outcome of this research will help to integrate the infrastructure-based LiDAR into smart mobility applications.

More specifically, our proposed methods comprise of two unsupervised learning algorithms that can automatically extract background features through intensity and range information. Supervised learning approach for LiDAR object detection is not adequate because the models often make ineffective predictions on new scenarios where there are no training data available. In the Figure 1, we present the object detection results using PointPillars [1] deep neural networks trained on a subset of PandaSet [2], which contains 2560 preprocessed organized LiDAR scans of various driving scenes. The data set provides 3-D bounding box labels for different object classes, including car, truck, and pedestrian. The PointPillars deep learning model can extract robust features from sparse point clouds using a similar architecture to PointNet, then the object detection layers will perform localization and classifications. Despite the excellent speed and accuracy achieved by the PointPillars model on many LiDAR benchmark dataset, it cannot be directly used on roadside scenarios. As shown in the Figure 1, although the PointPillars attains great results on its mobile LiDAR dataset, the well-calibrated model displays significant miss detections and false alarms on our LiDAR data. Given the issues of supervised learning approaches, in this paper, we developed unsupervised learning method for roadside LiDAR object detection that have better explanatory capability and does not need any training data.

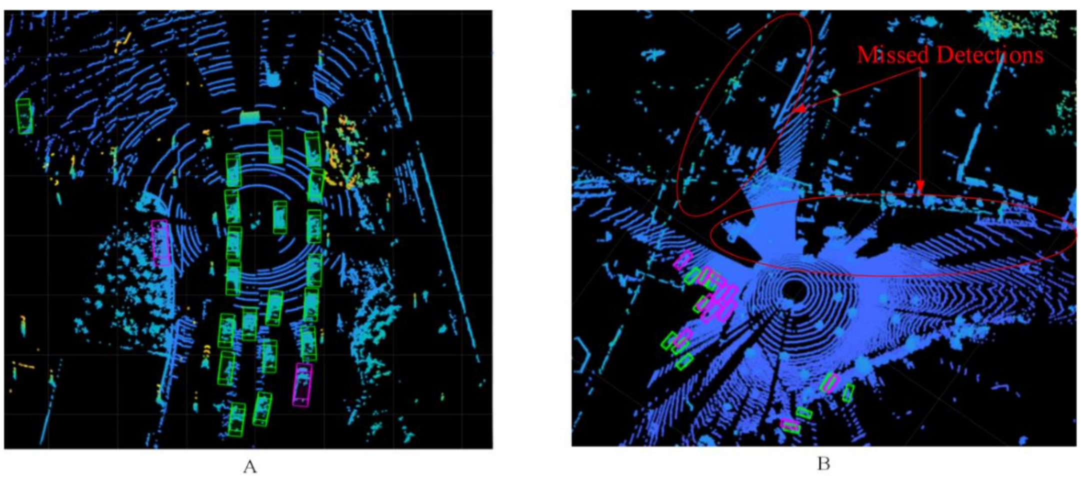

Figure 1 Trained Deep Neural Network on Mobile LiDAR and Roadside LiDAR Data (A. PointPillars Model on PandaSet Test Data; B. PointPillars Model on our own LiDAR Data)

RELATED WORK

LiDAR Object Detection

LiDAR based object detection algorithms are classified as static LiDAR object detection method and mobile LiDAR object detection method. Most of the LiDAR data processing algorithms were developed for autonomous driving vehicles to use the precise 3D geometric information. Li, Zhang and Xia [3] applied Fully Convolutional Neural Networks to detect vehicles from 2D point map that was projected from 3D coordinates. Zhou and Tuzel [4] developed Voxelnet as a generic architecture for object detection, which mainly contains three blocks 1. the Feature learning network, 2. the Convolutional middle layers and 3. the Region proposal network. The feature learning network divide the 3D point cloud data into uniform voxel grids and use random sampling to further reduce dimension, then each voxel is encoded into feature vector for the convolution neural networks to perform object detection and localization. Wirges et al., [5] proposed a deep convolutional network for 3D LiDAR data object detection and classification through occupancy grid map. The 3D LiDAR data were converted into bird eye view angle, and then the surface segmentation was removed, after that the objects are detected on the grid map. Zheng et al. [6] developed real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving by projecting 3D point cloud on the 2D grid and applied pre-RoI-pooling to accelerate the process speed. A BirdNet [7] was proposed for LiDAR 3D point cloud object detection, consists of three steps: First, original point cloud data is projected into bird's eye view angle. Later, a convolutional neural network is used for object localization. Finally, detection results were recovered in a post-processing phase. PointNet [8] is proposed to directly consume the point cloud data instead of transforming data to voxel or collections of images. The PointNet is a unified architecture can perform object classification, part segmentation, to scene semantic segmentation. Yin, Yang and He [9] developed a mobile LiDAR object detection algorithm for intelligent vehicles by converting LiDAR data into spherical coordinates. Their method separate ground LiDAR points from road users by identifying breaking points of the radial distances curve. To address the computational challenge of large volume LiDAR data, Lyu, Bai and Huang [10, 11] developed a convolutional neural network to perform semantic segmentation and evaluated on KITTI dataset, which has achieved fast and high accuracy of road segmentation using an efficient hardware design. Wu et al. [12, 13] proposed SqueezeSeg convolutional neural network for semantic segmentation of road-objects from 3D LiDAR data. The neural network takes projected LiDAR data and outputs point-wise label map, which is then refined by conditional random field (CRF). Milioto et al. [14] developed RangeNet++ for fast and accurate LiDAR semantic segmentation, which runs with range image transformed from LiDAR data. Chen et al. [15] propose simultaneous location and mapping model using LiDAR semantic named SuMa++ to reliably filter moving objects.

Most of roadside LiDAR data processing methods are based on background/foreground separation method. The first step is to separate foreground and background. The second step is to cluster the moving points into vehicles or non-vehicles. Then tracked road users are used for speed estimation and safety analysis. Zhang et al. [16] developed a tracking refinement module to optimize the centroid tracking of vehicle classification. The refinement module transforms the 3D point cloud into 2D images, and then applied image matching to determine the optimum location of tracked objects. Then the 2D matched object is converted back to 3D coordinates. Their algorithm is implemented to extract vehicle speed and verified using manually processed dataset. Lv et al. [17] proposed a LiDAR-enhanced connected infrastructure solution to collect traffic data of traffic participants using roadside LiDAR and broadcast the message through DSRC to enable connected vehicle application. They developed procedure to extract high-resolution micro traffic data from LiDAR data, including background filtering, object clustering, object classification, lane identification and target tracking. DBSCAN method was used for clustering object point clouds and Global Nearest Neighbor (GNN) was used to track the moving objects in different frames. Zhao et al. [18] explored infrastructure-based LiDAR sensors to detect and track pedestrians and vehicles at intersections. The proposed method was conducted in the following orders, background filtering, object clustering, pedestrian and vehicle classification, and tracking. The performance of their proposed model is impacted by the density of points, occlusions, and perspective shadow. Wu [19] implemented the similar procedure to track vehicles using roadside LiDAR sensor for the CV applications. The algorithms include background filtering, lane identification, and vehicle speed tracking. In the background filtering step, the 3D space is divided into multiple cubes to estimate point density. The tracking step uses average point as tracking point to represent detected vehicle. Zhang et al. [20] propose an automatic background construction and object detection method for roadside LiDAR data. The proposed method considers the horizontal and vertical angular value as spherical coordinates. The background dataset is constructed with the farthest distance in each angle. By comparing the background dataset with new data according to the same horizontal and vertical angular value, object points were extracted and clustered to pedestrian and vehicle detection. Zhao et al. [21] conducted research for lane and movement-based traffic volume data collection using infrastructure-based LiDAR under different congestion levels and traffic compositions, covering signalized intersections, pedestrian crossings, work zones, stop-sign intersections, metered/unmetered ramps, and rural highways.

The previous background modeling methods mainly make use of basic summaries such as maximum value, gradients, or density. In our new methods, we applied advanced techniques through pattern decomposition and dynamic clustering. This research for the first time developed 3D point cloud background modeling approaches that relate the rich body of 2D image-based background modeling techniques to the LiDAR point cloud using data driven dynamic classification techniques.

Background Modeling

Background modeling method is used as the first step of many video surveillance applications to understand video sequences. Each video frame is compared with the background video model to identify foreground objects with precise localization information. Bouwmans [22] provide a comprehensive survey paper. The background modeling method has three main steps: 1. background initialization using first N-frames; 2. Classification of pixels into foreground and background; 3. Background model maintenance over time. The paper also identified 13 challenging situations for background modeling: Noisy image; Camera jitter; Camera automatic adjustment; Illumination changes; Bootstrapping; Camouflage; Foreground aperture; Moved background objects; Inserted background objects; Dynamic backgrounds; Beginning moving object; Sleeping foreground object; Shadows. Bouwmans [23] classified background modeling method into the following categories: Basic Background Modeling; Statistical Background Modeling; Fuzzy Background Modeling; Background Clustering; Neural Network Background Modeling; Wavelet Background Modeling; Background Estimation. The basic model uses mean [24], median [25], or histogram [26] to describe background pixels. The statistical model uses statistical variables such as Gaussian distribution [27, 28], Kernel Density Estimation [29] to classify pixels. The fuzzy background model [30, 31] uses fuzzy running average or Type-2 fuzzy mixture of Gaussian. The background clustering model uses K-means algorithm [32] or

Codebook [33]. The neural network background modeling [34, 35] trains a set of weights on N clean background frames. The wavelet background model uses discrete wavelet transformation (DWT) [36]. The background estimation model is estimated with a filter such as Wiener filter [37], a Kalman Filter [38] or a Tchebyche Filter [39]. Goyal and Singhai [40] reviewed several Gaussian Mixture Model for background/foreground detection and conducted comparative analysis and analyzed the scope to improve them.

METHODOLOGY

The roadside LiDAR object detection has different characteristics than mobile LiDAR object detection. First, most of the point clouds in roadside LiDAR model are static background, while the mobile LiDAR point clouds model contains mostly the changing environment; Second, with the increasing distance between road users to the LiDAR sensor, the gaps among laser beams get larger, resulting in more blind area. The amounts of Laser points on the pedestrians and vehicles are fewer than mobile LiDAR sensor, which can have only one ring of laser beam. Due to the different application purposes, more efficient background subtraction methods can be used to infrastructure LiDAR sensor. In this section, two data-driven algorithms, Dynamic Mode Decomposition (DMD) and Triangle Algorithm, were applied for Roadside Lidar moving object detection and tracking. Before applying the background modeling methods, the LiDAR data need to be transformed and reorganized from a packet of $(X, Y, Z, \text{intensity})$ to matrices of Azimuth, Elevation, $\text{Range}, \text{Intensity}$.

Data Transformation

The LiDAR sensor record distance relative to itself as well as intensity values (depend on the reflectivity of the object and the wavelength used by the LIDAR). Two types of packets were created, data packets and position packets. The position packets are referred to as GPS packets. The LiDAR system uses spherical coordinate system originally. LiDAR data are stored in the format of XYZ coordinate. The Figure 2 shows the relation between Spherical coordinates (r, ω, α) and Cartesian coordinates (X, Y, Z).

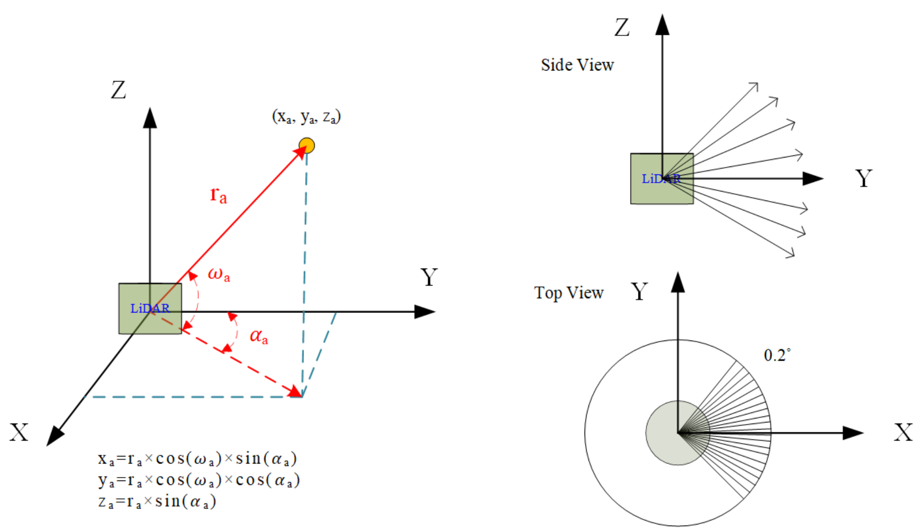

Figure 2 LiDAR Sensor Coordinate Systems

For the testing Velodyne LiDAR sensor, the sensor generates one data frame after completing a 360° scanning. The theoretical azimuth resolution is 0.18°. However, the azimuth angles between two emits often vary and always deviates from the resolution of 0.18°. The number of Laser beams emitted at each data frame often vary due to the manufacturing model and LiDAR brands. In past experiments, we found that the Velodyne HDL-32E LiDAR sensor emits 1806 to 1809 times per spin at the rotation frequency of 20 Hz. For the 128 Alpha Prime Velodyne model, it radiates 1772 ~ 1779 times per spin at 10 Hz but with higher vertical density. In this step, the original LiDAR data packet will be arranged into 1800 grids, based on azimuth resolution of 0.2°. The range of azimuth α is from $-179.99° \sim 179.99°$, which needs to convert to $0° \sim 359.99°$. A hash function is used to rearrange the LiDAR point to its corresponding grid.

$$h(\alpha) = \mathrm{mod}\left(\left\lfloor \frac{\alpha}{0.02°} \right\rfloor + 1,\ 1800\right) \tag{1}$$

if $h(\alpha)$ has collision, we will compare the range between two points that crash into the same azimuth grid. The smaller-range point will be preserved, because the background points are often farther than foreground objects and we want to keep the most useful data.

As shown in Figure 3, the LiDAR streaming data in .pcap file format store the data in cartesian coordinates with additional information of intensity value and/or RGB value by each channel. The number of channels and the elevation of each LiDAR channel are fixed for the infrastructure LiDAR after installation. Therefore, we use the elevation of each beam as known values and don't need to keep a set of matrices for elevation data.

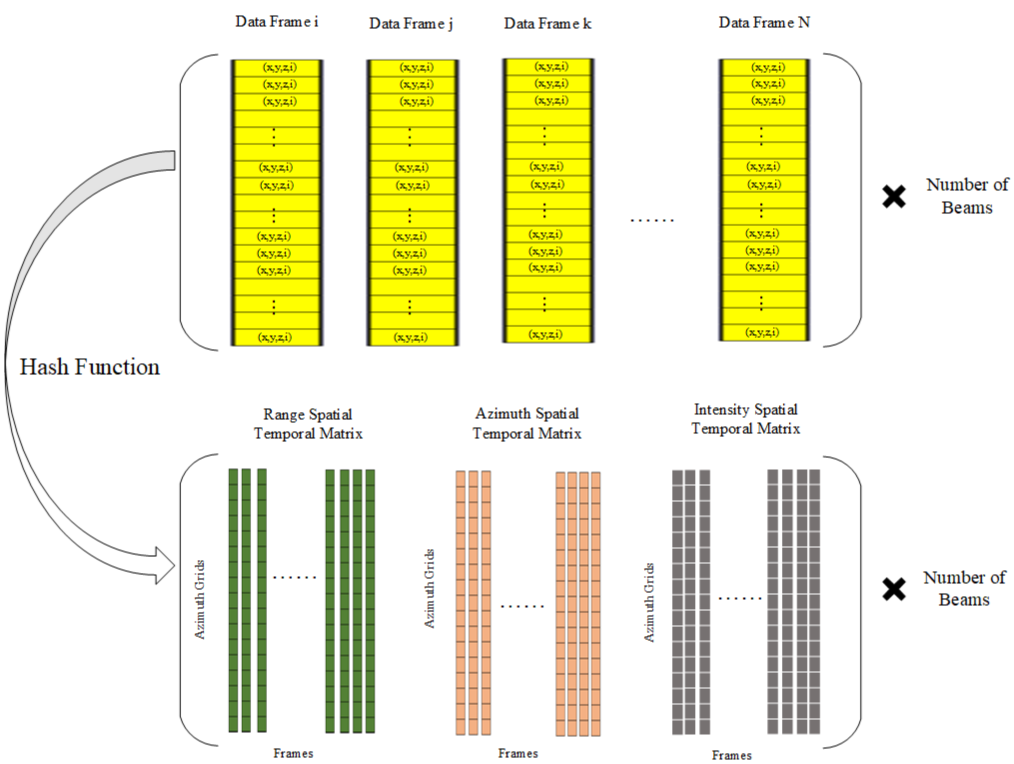

Figure 3 Transforming LiDAR Point Cloud into Azimuth Unit Spatial Temporal Matrix

Dynamic Mode Decomposition

Dynamic mode decomposition (DMD) is a data-driven technique for discovery of underlying patterns from high-dimensional data. It was firstly defined by Schmid and Sesterhenn [41, 42] to extract dynamic information from flow fields that can be used to describe the physical mechanisms captured in the data sequence. The DMD methods are connected to mathematical foundation that is readily interpretable using standard dynamic system techniques. The goal of DMD method is to extract the background mode for each channel of LiDAR. Then we will use the background mode to match the background points and filter out moving objects.

The LiDAR data at each channel can be thought of as scanning of environmental information for each spin. The intensity data of i_{x+1} at frame $(x+1)$ is assumed to relate to previous intensity measurement of i_x by linear operator A. the linear operator A is time-independent operator that reflects the time evolution of each beam's intensity value.

$$\mathbf{i}_{x+1} = A\,\mathbf{i}_x \qquad (1)$$

The DMD algorithm is a regression method to estimate A that can characterize the channel intensity changes captured by each frame. The problem is formulated as follows:

$$I = \begin{bmatrix} | & \cdots & | \\ \mathbf{i}_1 & \ddots & \mathbf{i}_{m-1} \\ | & \cdots & | \end{bmatrix}, \quad I' = \begin{bmatrix} | & \cdots & | \\ \mathbf{i}_2 & \ddots & \mathbf{i}_m \\ | & \cdots & | \end{bmatrix} \qquad (2)$$

Where I is called left intensity diagram, I' is called the right intensity diagram. I' has one frame difference compared to I. I' represents the time evolution of matrix I. The DMD algorithm seeks to find the best fit between the two matrices I and I' using a linear operator A.

$$I' = AI \qquad (3)$$

In order to solve A, the problem is converted to the following least-square problem.

$$\hat{A} = \underset{A}{\operatorname{argmin}} \|I' - AI\|_F^2 \qquad (4)$$

By using Moore-Penrose pseudoinverse, we obtain the estimator $\hat{A}$:

$$\hat{A} = I'I^{\dagger} \qquad (5)$$

The DMD mode that contains intensity information is eigen vector of $\hat{A}$, and each DMD mode corresponds to an eigenvalue of $\hat{A}$. By finding the eigenvectors and eigenvalues of matrix $\hat{A}$, we obtain the DMD mode $\Phi = W$.

$$\hat{A}W = W\Lambda \qquad (6)$$

The column of W are eigenvectors comprising of dominant mode $\emptyset_j$ and Λ is the diagonal matrix of eigenvalues λ_j. The Spatial-temporal Intensity Matrix be reconstructed using first k^{th} modes, where $k \leq \min(n, m)$.

$$\text{ST Intensity Matrix} \approx \Phi B \mathcal{V} = \underbrace{\begin{bmatrix} \emptyset_{11} & \cdots & \emptyset_{1k} \\ \vdots & \ddots & \vdots \\ \emptyset_{n1} & \cdots & \emptyset_{nk} \end{bmatrix}}_{\textbf{modes}} \underbrace{\begin{bmatrix} b_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_k \end{bmatrix}}_{\textbf{amplitudes}} \underbrace{\begin{bmatrix} 1\ \lambda_1 & \cdots & \lambda_1^{m-1} \\ 1\ \vdots & \ddots & \vdots \\ 1\ \lambda_k & \cdots & \lambda_k^{m-1} \end{bmatrix}}_{\textbf{dynamics}} \qquad (7)$$

Where, Φ are dominant modes from spatial-temporal map. Matrix B is the matrix of amplitudes. $\mathcal{V}$ is the Vandermonde matrix representing time evolution of DMD modes.

An intensity measurement $\mathbf{i}_t$ at frame $t \in 1, \dots, m$ can be estimated as follows:

$$\widetilde{i_t} = \sum_{j=1}^{k} b_j \emptyset_j \lambda_j^{t-1} \qquad (8)$$

Where b_j is amplitude, $\emptyset_j$ is each DMD mode, and λ_j^{t-1} is the time evolution of each intensity mode.

Let $t = 1$, we obtain the following equation.

$$\widetilde{i_1} = \sum_{j=1}^{k} b_j \emptyset_j \qquad (9)$$

So that the matrix B can be estimated as least-square problem using first scanline i_1 as initial state.

$$\widetilde{B} = \underset{B}{\operatorname{argmin}} \|i_1 - \Phi B\| \qquad (10)$$

Any DMD mode that does not change in time will have $\lambda_j = 1$, which forms the background of the intensity diagram.

In the intensity diagram, the intensity values of background are highly correlated from one column vector to the next, suggesting the low-rank structure. The DMD algorithm separate background and foreground by decomposing the intensity diagram into low-rank (background) and sparse (foreground) components.

$$I_{DMD} = \text{background} + \text{forground} = \sum_p b_p \phi_p \lambda_p^{t-1} + \sum_{j \neq p} b_j \phi_j \lambda_j^{t-1} \qquad (11)$$

Where $|\lambda_p| = 1$. $t \in 1, \dots, m$ is the data frame sequence.

The separation results were shown in the Figure 4. The y-axis shows azimuth units of LiDAR beam, the x-axis is for each data frame. The left figure is original intensity image, the middle image is the background that is time independence, and the right figure is the foreground moving objects. After obtaining the background intensity modes for all channels, we can use the background intensity value as filter to detect the moving objects.

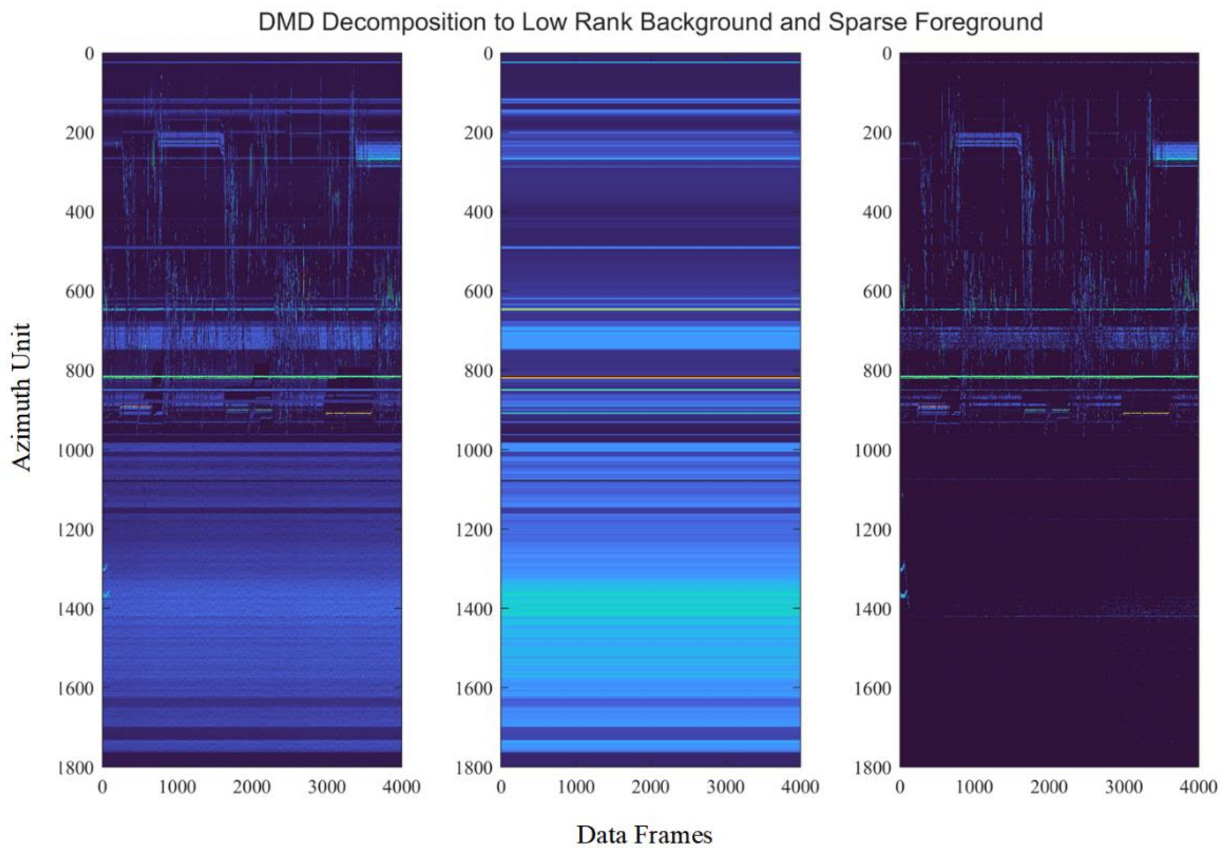

Figure 4 DMD Decompose Spatial-Temporal Intensity Map into Low-Rank Background and Sparse Foreground

The DMD method will be applied on each beam to build a background filter, which can be used to separate moving objects from background objects.

Algorithm: Compute DMD Background Modes of Intensity

==

Input: Intensity Channel Azimuth Unit Diagram

Outputs: Background Intensity Mode for Each LiDAR Beam

For Every Beam Intensity spatial temporal matrix I

1. Calculate operator that fits between the following two matrices using Moore-Penrose pseudoinverse: $I' = AI \rightarrow A \approx I'I^{\dagger}$
2. Take SVD of I: $I \approx U\Sigma V^*$
3. Reduced matrix and obtain of $\tilde{A}$ by projecting A onto U_r: $\tilde{A} = U_r^* A U_r = U_r^* X' V_r \Sigma_r^{-1}$
4. Eigen decomposition: $\tilde{A}W = W\Lambda_r$
5. Compute modes: $\Phi = X'V\Sigma^{-1}W$
6. Obtain the low-rank background mode ϕ_p, whose corresponding eigenvalue is asymptotically close to 0.

End For

==

Triangle Thresholding Algorithm

The Triangle algorithm is a dynamic clustering method based on the histogram analysis. To implement the triangle algorithm, the first step is to construct a histogram of ranges vs frequency for all beams and all azimuth units. Draw a line between the highest range value of the histogram h_{max} and the minimum range histogram h_{min} (for LiDAR data, the minimum histogram is by default at distance 0, because the emitted laser points either hit an object and returned with positive range value or never return). Then the algorithm will calculate the point to line distance d and increase h_{min} and repeat for all histogram h until $h = h_{max}$. The threshold value becomes the bin edge of range value for which the distance d is maximized. Like in Figure 5 histogram, the background range value is around 19 meters, while the foreground moving object's ranges are at about 15 meters.

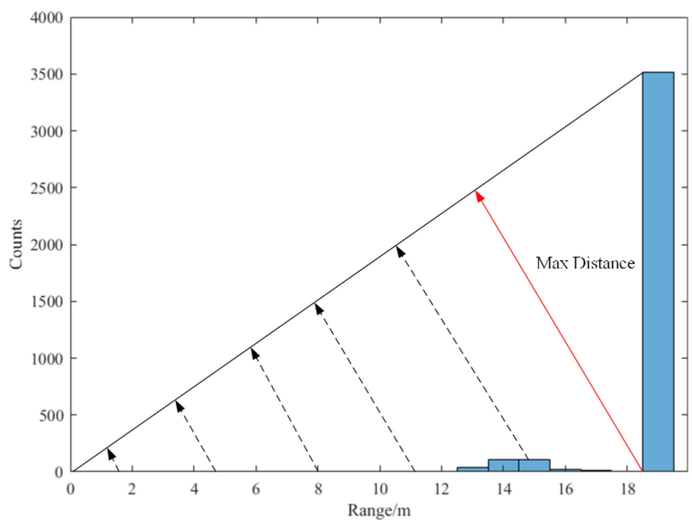

Figure 5 The triangle algorithm finds the threshold value that maximizes the distance d

The triangle thresholding aims to classify point clouds into either background points or moving objects based on the range information. Because the static infrastructure backgrounds are the farthest objects hit by the laser beams and have greater distance than the moving targets in the 3D point clouds, the triangle method can automatically select the threshold range values for the moving vehicle detection. The method is developed on two assumptions: 1. The static background objects occupied most the frequencies in the LiDAR point clouds. 2. The background points have farthest distance and distributed normally with standard measurement errors. This technique is particularly effective when the background objects point clouds produce a dominant peak in the histogram. Figure 6 illustrates the laser beam was intercepted when moving targets present in the monitored space. The automatic triangle thresholding method is a reliable and efficient model due to the highly unequal population size ratio between moving objects and backgrounds.

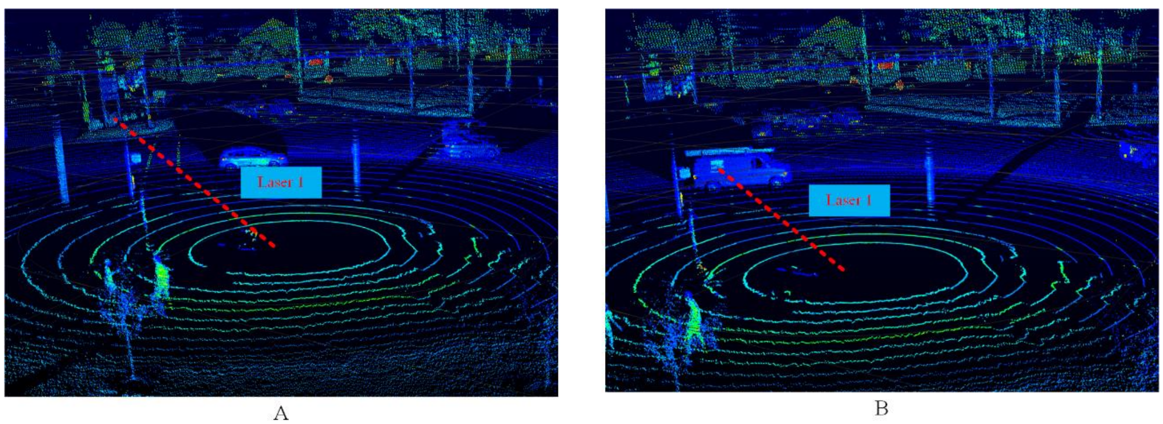

Figure 6 The LiDAR Laser Beam was Intercepted by Moving Vehicle

EXPERIMENT DESIGN

This data collection is to test Intelligent Transportation System Infrastructure at intersections through immediate data collection and analysis capability, which is part of the NJDOT project Real-time Traffic Signal System Performance Measurement. This project aims to testing innovative sensing and detecting methods to evaluate and monitor signal performance in real time. The outcome of the project will be used to improve intersection safety, reduce congestion and environmental impact. The testing site is selected from a key New Jersey arterial corridor on October 20th, 2021, from 3-6 pm at US1 at Bakers Basin. The 3 hour data include high-resolution GoPro video, 128 beam Velodyne Alpha Prime LiDAR data, connected vehicle SPaT and MAP data. The camera was mounted on the roadside pole and LiDAR sensor was mounted with tripod at the walkway and were powered by high-capacity batteries and solar panel. During the three hour periods, the GoPro generated 80G data at 60 frames per second, and the Alpha Prime Velodyne LiDAR generated 70G data at 10 Hz. The LiDAR has similar efficiency of data storage and somewhat decreases the data transfer demand compared to video data.

In the Figure 7, the experiment set up was displayed to show the coverage for video and LiDAR detection at the same timestamp. The vehicle detection and tracking are processed using Yolov5 and DeepSort for comparative analysis. The LiDAR Sensor was only installed at the height of 1.7 meter, and already can sufficiently provide a wide range of coverage and holistic 3D measurements of surrounding environment. Although the camera is installed at the height of about 5 meters, it is still difficult to cover the entire intersection area from all directions. Compared to camera detector, the LiDAR sensor shows great potential and will play a significant role as an intelligent infrastructure solution in the following years.

Figure 7 Testing Site for Traffic Data Collection Using Camera and LiDAR Sensor

MODEL EVALUATION

In this session, we will break the entire solution into sequential steps and examine the model results in detail. The overall workflow is shown in the Figure 8. The ROI filter, noise removal, clustering, bounding box detector and tracking are considered as general approaches. The two new algorithms, including DMD intensity background subtraction and triangle thresholding for range background subtraction, are integrated as one module. The extracted vehicle movements can be applied to many mobility or safety applications. For example, the vehicle counts for each turning movement could be used for signal optimization to assess whether the phase split is efficient. With real-time vehicle trajectories, we can also obtain near-miss conditions using surrogate safety measures to prevent the potential conflicts.

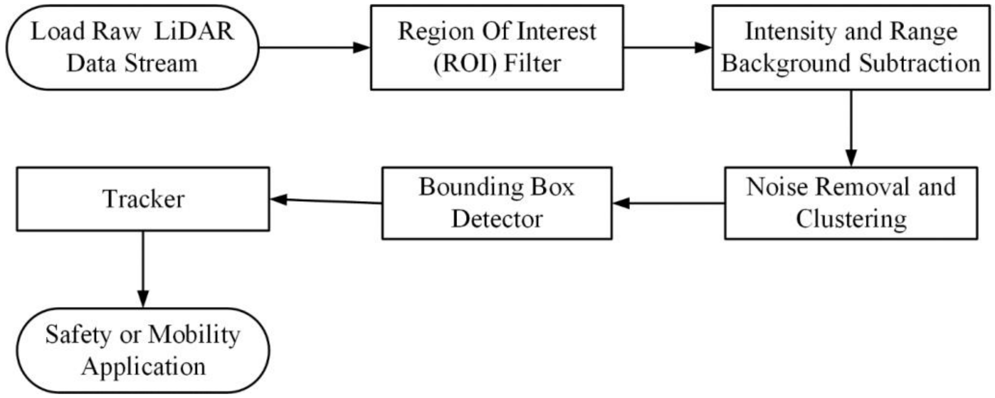

Figure 8 Workflow of Infrastructure LiDAR Background Subtraction

ROI Filter

As the infrastructure LiDAR are static, the accurate GPS coordinates can be obtained in practice. Therefore, the non-drivable space within the monitored area could be filtered out using geo-fencing methods. In the Figure 9, the raw LiDAR data are filtered by projecting all points into X-o-Y plane using a binary mask.

Figure 9 Region of Interest (ROI) filter

Background Subtraction

The background subtraction methods are directly performed on Spherical coordinates, which is the original coordinates of data collected from the LiDAR sensor. Therefore, it would save the computation to convert spherical data from the sensor to Cartesian coordinates within sensor chip. The mainstream LiDAR processing methods are based on the Cartesian coordinates and the data are saved in sparse matrices. With methods of background subtraction, more than 90% of data can be eliminated. Using the spherical coordinates system could significantly improves the LiDAR point cloud acquisition and transmission efficiency.

Noise Removal and Clustering

The noise removal is based on Local Outlier Factor (LOF) Algorithm. A point will be considered as noise if its K-distance, which is the distance between the point and it's Kth nearest neighbor, is smaller than a threshold. The point cloud clustering step is also distance based that segment all point cloud data into clusters and returns cluster labels of all cloud points.

Bounding Box Detector and Tracking

Upon finishing clustering, we then fit bounding box to each cluster that is greater than the minimum threshold number of points and its height, length and width are also between minimum and maximum value from each dimension. Then the detected object is encoded into state-space model that contains the objects' corresponding measurements and transition of state (speed in x, y z dimension, and turning rate). A joint probabilistic data association (JPDA) tracker is applied to update the tracked list of objects for each frame.

In the Figure 10, the model results after all steps are presented, showing the vehicle detection and tracking results from three phases of the signalized intersection. In the first column, the foreground moving vehicles are colored as green and the background LiDAR point clouds are colored in purple. The middle column pictures show tracking module outputs, where the red boxes are detected objects after distance-based clustering from the detector model, and the green box is confirmed tracks with certain confidence after the tracking module. We also presented the video detections using Yolov5 that was trained on coco dataset and DeepSort for online and real-time vehicle detection and tracking. As you can see, the LiDAR sensor provide broader field of view than the GoPro camera. For Phase B, the pretrained deep learning model missed three vehicles passing the intersection. The proposed LiDAR model showed more reliable detection and tracking capability than one of the most advanced computer vision models.

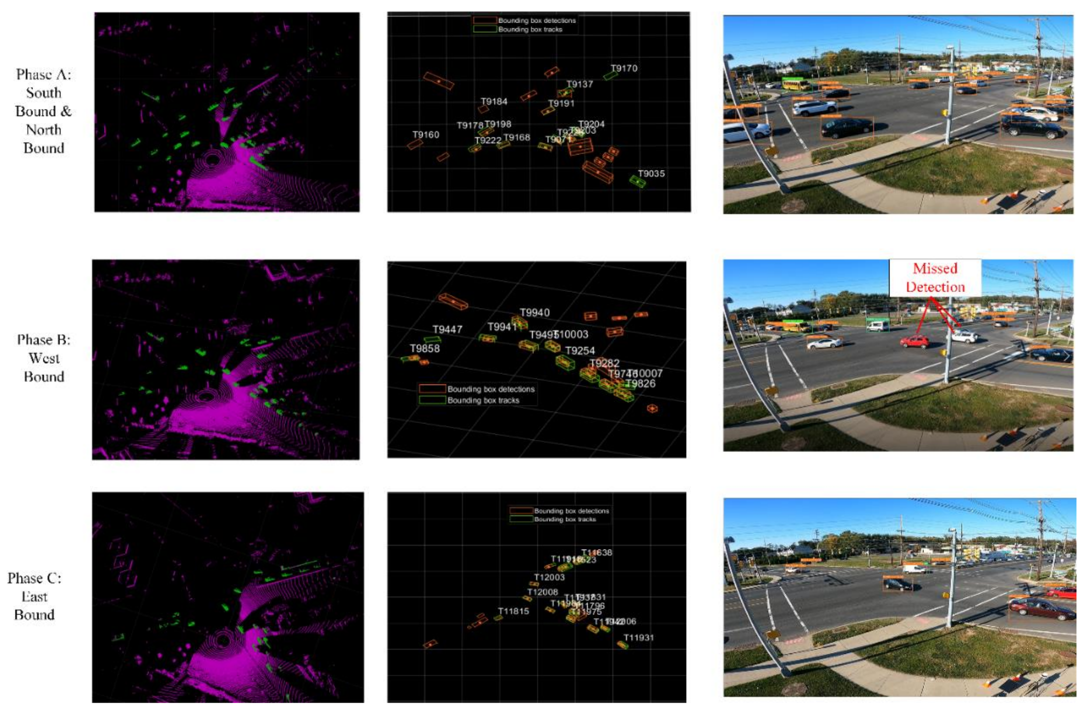

Figure 10 Proposed LiDAR Object Detection and Tracking Compared to Deep Learning Video Detection

Movement counting is an essential input for signalized intersection to optimize the timing parameters. The following figure shows LiDAR detected vehicle trajectories grouped by traveling directions in different colors. The second half of picture

is vehicle detection and tracking results from a commercial AI traffic data collection platform [43], which can create traffic movement counts at 15 minute intervals with 95% accuracy.

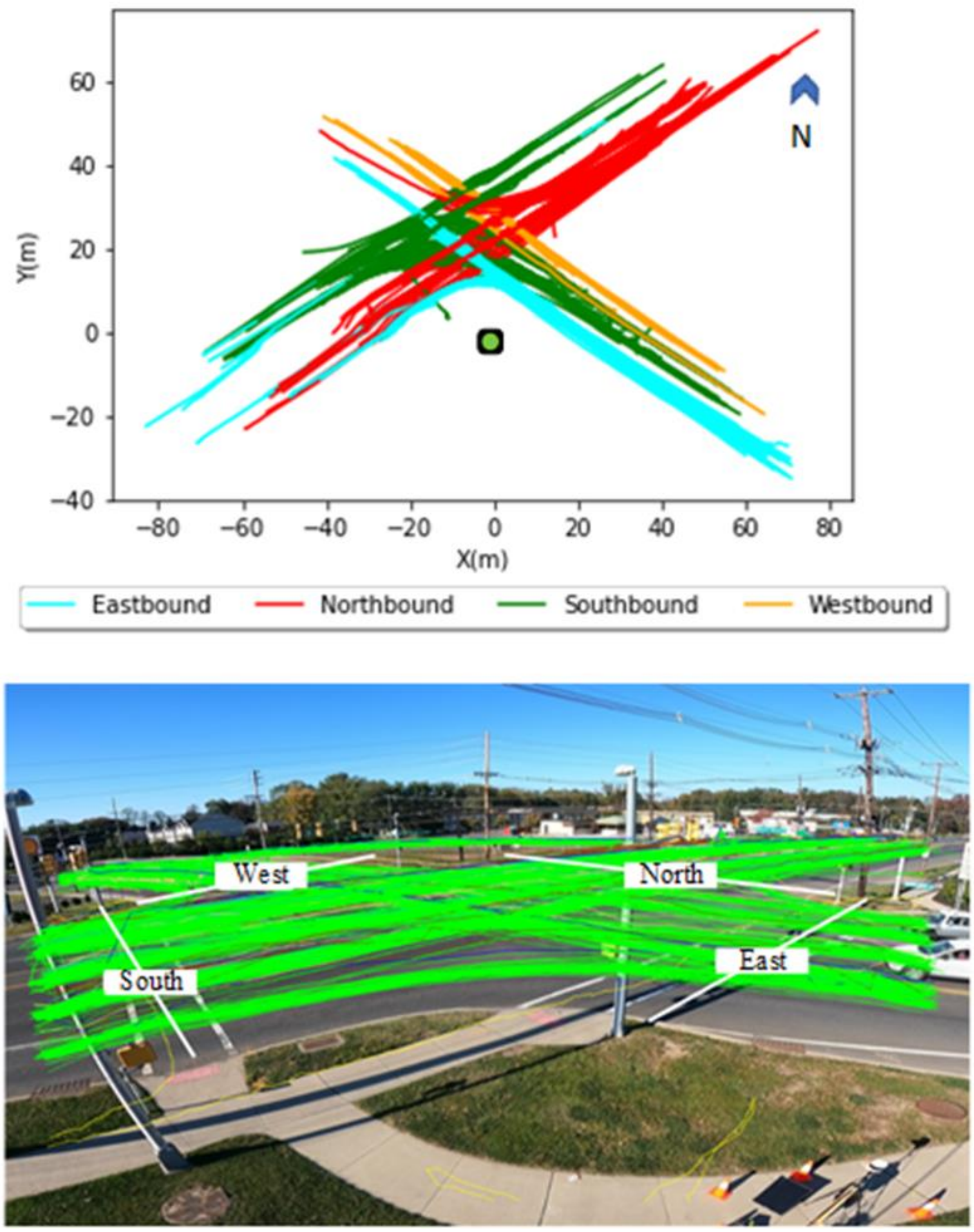

Figure 11 LiDAR Detected Trajectory and Video Detected Trajectory as Benchmark

In the Table 1, vehicle movements count for all four directions are presented. The overall counting accuracy is 92.76%. The reason that causes the largest counting errors on westbound is that the westbound vehicles are in the farthest lane to the LiDAR sensors and blocked by left turn vehicles and inbound vehicles from other paths. In this experiment, we only have portable tripod and cannot hoist the LiDAR to an ideal place. With better setting up and construction efforts to reduce the blind zones, the accuracy of our model performance is expected to become better.

Table 1 Vehicle Movements Counting Evaluation

	Total Count	Eastbound	Westbound	Northbound	Southbound
LiDAR	987	147	52	397	391
Video Benchmark	1064	125	78	448	413
Error Rate	7.24%	17.60%	33.33%	11.38%	5.33%
Accuracy	92.76%	82.40%	66.67%	88.62%	94.67%

Concluding Remarks

In this paper, we developed a novel background subtraction method with unsupervised learning algorithms for infrastructure LiDAR object detection and tracking. Main contributions of this paper to the existing literatures are summarized as follows:

1. Our method integrates both the range information and intensity information for point cloud object detection. This method can reduced 90% redundant background points and increase the data acquisition efficiency.
2. Instead of converting the point clouds into 3D voxels, our methods transform LiDAR data into 2D matrices and run more efficiently with reduced dimension. With proper data transformation, we bridge the gap between image based background modeling and point clouds background modeling, making a rich body of well-studied image-based techniques suitable for LiDAR data.
3. The proposed methods are built on unsupervised learning that automatically discovers the structures from data. The two algorithms require very few parameters, which means more robust and easier for auto-calibration and deployment. For intensity based algorithm, the only parameter is the intensity threshold that differentiates sparse foreground points from the low-rank intensity modes. For the triangle algorithms, the only parameter is the bin size of the histogram to make sure the background counts will fall into the same bin, at the same time separable from moving objects.
4. Compared to the deep learning based LiDAR object detection methods, our method shows more heuristics and better expandability. It does not need to collect large amounts of training data, sophisticated network design, and GPU to support the functionality. The background model is easy to maintain.

In the Figure 12, the lane by lane queue length could be easily acquired using the proposed methods. The LiDAR measurements can be turned into input to adaptive traffic signal control and enable traffic managers to monitor systems in real-time. Beside mobility applications, the roadside LiDAR sensor can also be used for safety-critical applications. For instance, the LiDAR detection could identify near-miss situations at intersections and generate safety performances for signalized intersections. As opposed to conventional detector (e.g., Loop Detector, Radar) that are installed at fixed locations and only produce spot information, the LiDAR sensor generates much larger coverage as an ideal digital solution for smart infrastructure.

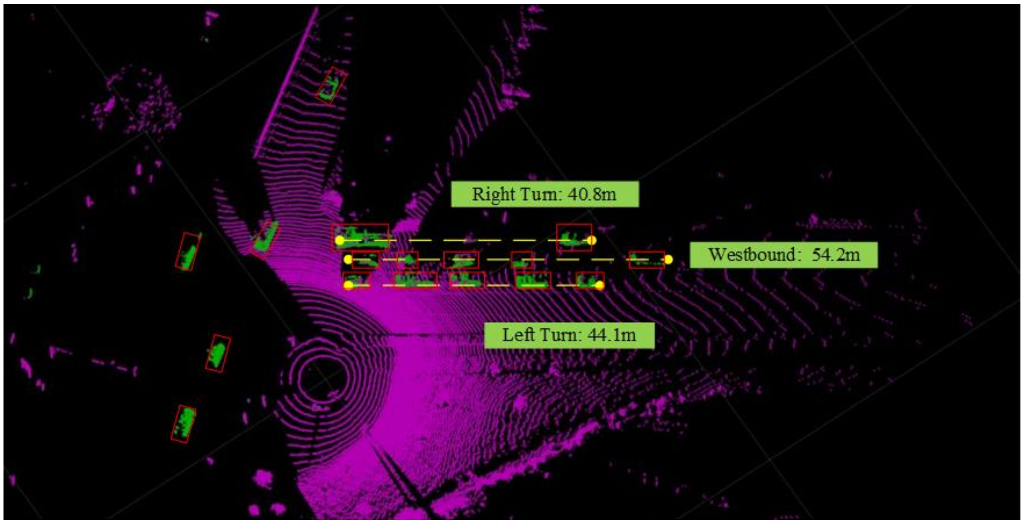

Figure 12 Lane by Lane Queue Length Measurement from Roadside LiDAR

The main problem of the LiDAR model is the occlusion due to the limited height of the tripod in this experiment setup. The vehicles on closer lane often project shadows to the middle and outer lanes. The problem could be mitigated by installing LiDAR sensor at a proper angle/height to make the laser radiate from top-down angle.

The high-resolution point cloud data will support the next-generation research on 3D big data sensing and analytics by creating digital twin of infrastructure systems in a holistic 3D environment. The roadside LiDAR object detection will be used to explore the many underlying scientific problems including transportation, infrastructure, energy, public service, and human activity systems and their interactions. For the next step, the research will be conducted in an urban environmental as part of the Middlesex County Smart Mobility Testing Ground (SMTG) to establish a living laboratory for smart mobility and smart city technology research in downtown New Brunswick, New Jersey. Our model will be further improved and evaluated on roadside LiDAR Sensor with permanent power supply and communication cables for real time connected and autonomous vehicles applications.

REFERENCE:

1. Lang, Alex H., Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. "PointPillars: Fast Encoders for Object Detection From Point Clouds." In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12689-12697. Long Beach, CA, USA: IEEE, 2019. https://doi.org/10.1109/CVPR.2019.01298.
2. Hesai and Scale. PandaSet. https://scale.com/open-datasets/pandaset.
3. Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network. arXiv preprint arXiv:1608.07916. 2016 Aug 29.
4. Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3d object detection. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 4490-4499).
5. Wirges S, Fischer T, Stiller C, Frias JB. Object detection and classification in occupancy grid maps using deep convolutional networks. In2018 21st International Conference on Intelligent Transportation Systems (ITSC) 2018 Nov 4 (pp. 3530-3535). IEEE.
6. Zeng Y, Hu Y, Liu S, Ye J, Han Y, Li X, Sun N. Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving. IEEE Robotics and Automation Letters. 2018 Jul 4;3(4):3434-40.
7. J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García and A. De La Escalera, "BirdNet: A 3D Object Detection Framework from LiDAR Information," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3517-3523, doi: 10.1109/ITSC.2018.8569311.
8. Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 652-660).
9. Yin H, Yang X, He C. Spherical coordinates based methods of ground extraction and objects segmentation using 3-D LiDAR sensor. IEEE Intelligent Transportation Systems Magazine. 2016 Jan 18;8(1):61-8.
10. Lyu, Yecheng, Lin Bai, and Xinming Huang. "Real-time road segmentation using lidar data processing on an fpga." In 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5. IEEE, 2018.
11. Lyu, Yecheng, Lin Bai, and Xinming Huang. "Chipnet: Real-time lidar processing for drivable region segmentation on an fpga." IEEE Transactions on Circuits and Systems I: Regular Papers 66, no. 5 (2018): 1769-1779.
12. Wu, Bichen, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud." In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1887-1893. IEEE, 2018.
13. Wu, Bichen, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud." In 2019 International Conference on Robotics and Automation (ICRA), pp. 4376-4382. IEEE, 2019.

14. Milioto, Andres, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. "RangeNet++: Fast and accurate LiDAR semantic segmentation." In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4213-4220. IEEE, 2019.
15. Chen, Xieyuanli, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. "Suma++: Efficient lidar-based semantic slam." In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4530-4537. IEEE, 2019.
16. Zhang J, Xiao W, Coifman B, Mills JP. Vehicle Tracking and Speed Estimation From Roadside Lidar. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2020 Sep 18;13:5597-608.
17. Lv B, Xu H, Wu J, Tian Y, Zhang Y, Zheng Y, Yuan C, Tian S. LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities. IEEE Access. 2019 Jun 17;7:79895-907.
18. Zhao J, Xu H, Liu H, Wu J, Zheng Y, Wu D. Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors. Transportation research part C: emerging technologies. 2019 Mar 1;100:68-87.
19. Wu J. An automatic procedure for vehicle tracking with a roadside LiDAR sensor. Institute of Transportation Engineers. ITE Journal. 2018 Nov 1;88(11):32-7.
20. Z. Zhang, J. Zheng, H. Xu, X. Wang, X. Fan and R. Chen, "Automatic Background Construction and Object Detection Based on Roadside LiDAR," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, pp. 4086-4097, Oct. 2020, doi: 10.1109/TITS.2019.2936498.
21. Zhao J, Xu H, Zhang Y, Tian Y, Liu H. Traffic Volume Detection Using Infrastructure-Based LiDAR under Different Levels of Service Conditions. Journal of Transportation Engineering, Part A: Systems. 2021 Nov 1;147(11):04021080.
22. Bouwmans T. Traditional and recent approaches in background modeling for foreground detection: An overview. Computer science review. 2014 May 1;11:31-66.
23. Bouwmans T. Recent advanced statistical background modeling for foreground detection-a systematic survey. Recent Patents on Computer Science. 2011 Sep 1;4(3):147-76.
24. Horprasert T., Haritaoglu I., Wren C., Harwood D., Davis L. Pentland A., Real-time 3D Motion Capture, Workshop on Perceptual User Interfaces, PUI 1998, San Francisco, California, pages 87-90, November 1998.
25. McFarlane N., Schofield C. Segmentation and tracking of piglets in images, BMVA 1995, pages 187-193, 1995.
26. Zheng J., Wang Y., Nihan N., Hallenbeck, E. Extracting Roadway Background Image: A mode based approach, Journal of Transportation Research Report, No 1944, pages 82-88, March 2006.
27. Wren C., Azarbayejani A., Darrell T., Pentland A. Pfinder: Real-Time Tracking of the Human Body, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 19, No. 7, pages 780-785, July 1997.
28. [14] Stauffer C., Grimson W. Adaptive background mixture models for real-time tracking, CVPR 1999, pages 246-252, 1999.
29. Elgammal A., Harwood D., Davis L. Non-parametric Model for Background Subtraction, ECCV 2000, pages 751-767, Dublin, Ireland, June 2000.

30. Sigari M., Mozayani N., Pourreza H. Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Application, International Journal of Computer Science and Network Security, Volume 8, No. 2, pages 138-143, 2008.
31. El Baf F., Bouwmans T., Vachon B., Type-2 fuzzy mixture of Gaussians model: Application to background modeling, ISVC 2008, pages 772-781, Las Vegas, USA, December 2008.
32. Butler D., Sridharan S. Real-Time Adaptive Background Segmentation, ICASSP 2003, 2003.
33. Kim K., Chalidabhongse T., Harwood D., Davis L. Real-time Foreground-Background Segmentation using Codebook Model, Real-Time Imaging, 2005
34. Culbrik D., Marques O., Socek D., Kalva H., Furht B. Neural network approach to background modeling for video object segmentation", IEEE Transaction on Neural Networks, Volume 18, No. 6, pages 1614–1627, 2007
35. Maddalena L., Petrosino A. A self organizing approach to background subtraction for visual surveillance applications, IEEE Transactions on Image Processing, Volume17, No. 7, pages 1729–1736, 2008
36. Biswas S., Sil J., Sengupta N. Background Modeling and Implementation using Discrete Wavelet Transform: a Review, JICGST-GVIP, Volume 11, Issue 1, pages 29-42, March 2011.
37. Toyama K., Krumm J. Brumitt B., Meyers B. Wallflower: Principles and Practice of Background Maintenance, International Conference on Computer Vision, pages 255-261, Corfu, Greece, September 1999.
38. Messelodi S., Modena C., Segata N., Zanin M. A Kalman filter based background updating algorithm robust to sharp illumination changes, ICIAP 2005, Volume 3617, pages 163-170, Cagliari, Italy, September 2005.
39. Chang R., Ghandi T., Trivedi M., Vision modules for a multi sensory bridge monitoring approach, ITSC 2004, pages 971-976, October 2004.
40. Goyal K, Singhai J. Review of background subtraction methods using Gaussian mixture model for video surveillance systems. Artificial Intelligence Review. 2018 Aug;50(2):241-59.
41. Schmid PJ, Sesterhenn J. Dynamic mode decomposition of experimental data. In8th International Symposium on Particle Image Velocimetry, Melbourne, Victoria, Australia 2009 Aug 25.
42. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. Journal of fluid mechanics. 2010 Aug;656:5-28.
43. GoodVision Video Insights - Advanced Traffic Analytics Platform. GoodVision, 2021. https://goodvisionlive.com/. Accessed Dec 15, 2021.