The Development of a Smart Intersection Mobility Testbed (SIMT)

FINAL
REPORT
December 2020

Submitted by:

Peter J. Jin
Associate Professor

Tianya Zhang
Research Assistant

Yizhou Wang
Research Assistant

Yufei Huang
Research Assistant

Mohsen Jafari
Professor

Yi Ge
Research Assistant

Center for Advanced Infrastructure and Transportation,
Department of Civil and Environmental Engineering,
Rutgers, The State University of New Jersey,
100 Brett Road, Piscataway, NJ 08854

External Project Manager
Khalid Anjum, Middlesex County

In cooperation with

Rutgers, The State University of New Jersey
And
US Department of Transportation
Federal Highway Administration

Disclaimer Statement

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The US Government assumes no liability for the contents or use thereof.

The Center for Advanced Infrastructure and Transportation (CAIT) is a Regional UTC Consortium led by Rutgers, The State University. Members of the consortium are Atlantic Cape Community College, Columbia University, Cornell University, New Jersey Institute of Technology, Polytechnic University of Puerto Rico, Princeton University, Rowan University, SUNY - Farmingdale State College, and SUNY - University at Buffalo. The Center is funded by the U.S. Department of Transportation.

1. Report No. CAIT-UTC-REG23	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle **The Development of a Smart Intersection Mobility Testbed (SIMT)**		5. Report Date December 2020	
		6. Performing Organization Code CAIT/Rutgers University	
TECHNICAL REPORT STANDARD TITLE PAGE 7. Author(s) Peter J. Jin https://orcid.org/0000-0002-7688-3730, Mohsen Jafari https://orcid.org/0000-0003-4776-0632, Tianya Zhang https://orcid.org/0000-0002-7606-9886, Yizhou Wang https://orcid.org/0000-0003-3234-4548, Yufei Huang https://orcid.org/0000-0002-9996-9190, Yi Ge https://orcid.org/0000-0002-7186-9024		8. Performing Organization Report No. CAIT-UTC-REG23	
9. Performing Organization Name and Address Rutgers, The State University of New Jersey, 100 Brett Road, Piscataway, NJ 08854		10. Work Unit No.	
		11. Contract or Grant No. 69A3551847102	
12. Sponsoring Agency Name and Address Center for Advanced Infrastructure and Transportation Rutgers, The State University of New Jersey 100 Brett Road Piscataway, NJ 08854		13. Type of Report and Period Covered Final Report September 2019 – September 2020	
		14. Sponsoring Agency Code	
15. Supplementary Notes US Department of Transportation/OST-R 1200 New Jersey Avenue, SE Washington, DC 20590-0001			
16. Abstract This project developed the key modules and models for building a prototype smart mobility intersection testbed. The project proposed the key system design, developed the computer-vision based sensing module, created 3D infrastructure model, and explored a risk-based intersection traffic conflict analysis application as an application of the developed intersection smart mobility testbed platform.			
17. Key Words Connected and Automated Vehicles, Computer Vision, Conflict Analysis		18. Distribution Statement	
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages Total 59	22. Price

Form DOT F 1700.7 (8-69)

Acknowledgments

The team would like to thank City of New Brunswick, Middlesex County, and New Jersey Department of Transportation in assisting the data collection and Connected and Automated Vehicle (CAV) application exploration in the project.

Table of Contents

List of Figures

List of Tables

DESCRIPTION OF THE PROBLEM

The emerging ECAV (Electrical, Connected, and Automated Vehicle) technologies will bring profound changes to future mobility. The private sector, such as Waymo, Uber, Tesla, and other self-driving startup companies, has accelerated road testing and accumulated significant vehicle miles daily. However, those datasets are not readily available to the public and academic sector for research and evaluation of their potential impact on and interaction with existing transportation infrastructure. With the accelerated deployment of connected and automated vehicle (CAV) technologies, public agencies have urgent needs (1) to understand how to prepare their transportation infrastructure, traffic operations, and management system to support CAVs, (2) on how to design traffic control systems for mixed Connected Automated, and Human-driven vehicles and (3) how to utilize the new data sources and application platforms of ECAVs to improve traffic mobility, safety, and environmental and energy impact. Meanwhile, the rapid development of the AV (Automated Vehicles)-grade sensing technologies, including LiDAR (Line Detection and Ranging) and computer vision technologies, have made it possible for the public and academic sectors to collect industrial-grade high-resolution vehicle, pedestrian, and infrastructure data. Research in this project focuses on planning, developing, and piloting a Smart Intersection Mobility Testbed (SIMT) for high-resolution smart mobility and infrastructure data sharing and testing of ECAV applications.

APPROACH

This project builds the basic platform modules for a proposed pilot Smart Intersection Mobility Testbed (SIMT) in downtown New Brunswick. The testbed will be equipped with high-resolution sensors to collect the real-time vehicle, pedestrian, and infrastructure change data. Data sharing and testing platforms will be built for testing and evaluating different mobility, safety, environmental, and energy applications. The proposed testbed will include the following key components.

- Roadside Sensing and Computing Layer
- V2I (Vehicle-to-Infrastructure) and I2C (Infrastructure-to-Center) Communication Layer
- Central Computing, 3D Modeling, and Visualization Layer
- Traffic, Pedestrian, and Infrastructure Analytic and Modeling Layer
- In-Vehicle Smart Mobility Service Application Layer.
- Testing interfaces for mobility, safety, environment, and energy applications.

METHODOLOGY

Proposed Smart Intersection Mobility Testbed System Design and Components

This project will establish the pilot Smart Intersection Mobility Testbed (SIMT) at the intersection of Albany (Route 27) and George Street in downtown New Brunswick for the collection, processing,

visualization, and application testing with the smart mobility sensor data. The proposed testbed includes the following key components.

Roadside sensors

LiDAR and computer vision sensors are installed at the roadside infrastructure at or near the two intersections to collect the real-time vehicle, pedestrian, and infrastructure data. LiDAR sensors will be mounted to the existing roadside infrastructure, such as signal or light poles. Computer vision sensors consist of a high-resolution video to be deployed on the top of buildings around the testbed sites.

An array of advanced sensors that will collect high-resolution data will form the foundation of the STMG. Sensors will range from autonomous-grade LiDAR to differential GPS base stations to HD (High-Definition) surveillance cameras. An initial data capture using LiDAR will be used to capture roadway, transportation facilities, exterior, and in some cases, interior building infrastructure will be collected to serve as a "base map" of the SMTG.

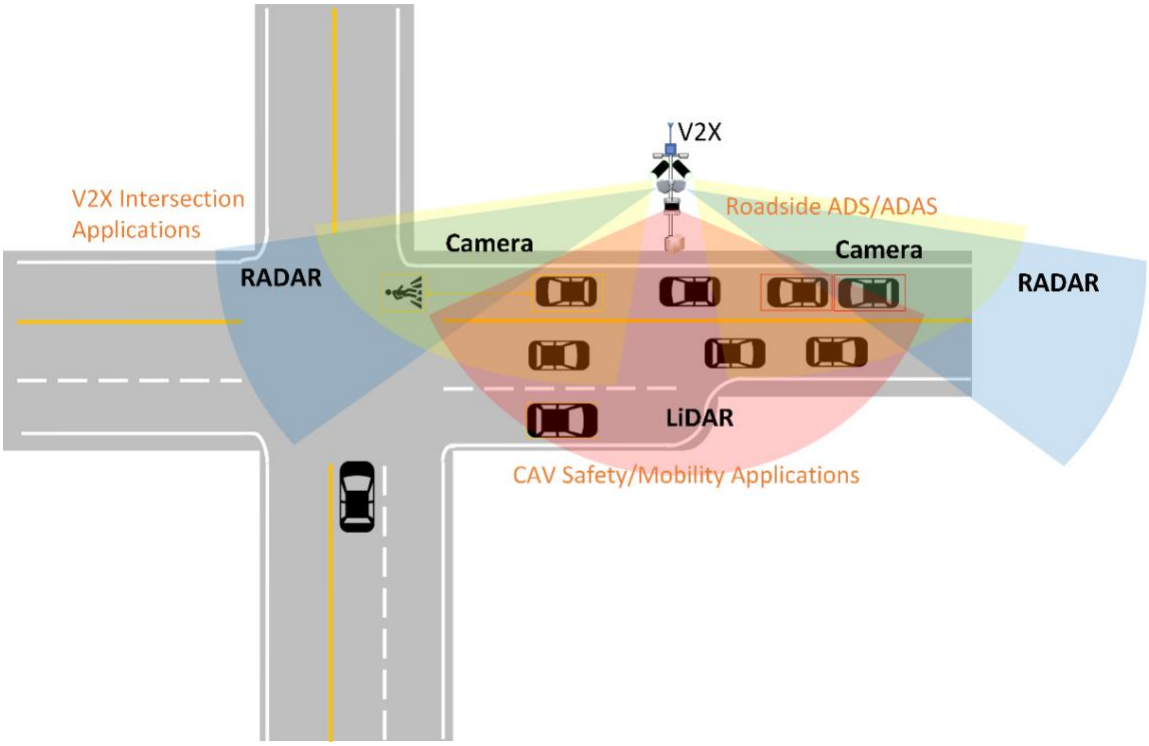

Figure 1. Sensor Configuration and Enabling Applications for the SMTG Smart Roadside Unit

As depicted in Figure 1, the multi-resolution sensors are summarized below.

- **LiDAR –** Light Detection and Ranging sensors utilize lasers to create 3D images of the surrounding area within 100 – 240m, depending on the type of sensor. High-angle, building-mounted LiDAR sensors will be used to collect high-resolution vehicle trajectory and velocity information to support a variety of CV and other applications. They will also be used at intersections to support smart intersection applications.
- **HD Camera –** HD cameras that can provide 4K resolution will be used with LiDAR sensors to

collect information regarding vehicle operations, pedestrian/cyclist behavior, and interaction at smart intersections. HD cameras will be supplemented with network cameras at locations where HD resolution is not required.

- **HD GPS/Bluetooth Beacon –** Highly accurate vehicle and pedestrian positioning will be provided using differential GPS base stations and Bluetooth beacons deployed near the test site to increase the accuracy levels of GPS data. The sensors can also provide last-position data by triangulating beacon signals around buildings and other infrastructure corners.
- **Millimeter Wave Radar Detector –**Millimeter Wave Radar Detectors can be used to collect trajectory data of moving objects (vehicles and pedestrians) to supplement the LiDAR sensor detection range. It will also provide redundancy for data collection during inclement weather, such as snow and fog, when LiDAR sensors have their limitations.
- **Parking Sensor –** Sensors that monitor parking spaces will be utilized to support the development of smart parking applications.
- **Weather Sensor –** Environmental conditions will be closely monitored; these sensors will also be used to support applications designed to provide road-weather condition alerts.

Edge/Fog Computing Units:

Edge/Fog computing units will be built to process the roadside sensor data in the field and transfer the processed data to the computing center at Rutgers Smart Mobility Lab to explore next-generation traffic management centers (TMCs) functionalities. If approved, the edge/fog computing units will also include the connectivity and data feed from the traffic signal controllers at the site.

The roadside computing layer supports application delivery for test ground users and travelers in the project corridor. Given the low latency required to support connected and advanced driving system applications, data processing will occur near drivers, pedestrians, and bicyclists at the roadside level. Key roadside computing technologies proposed for use at SMTG are summarized below.

- **Edge Processors -** LiDAR and other sensors generate significant amounts of data; pushing this data to a central location/back office would introduce significant latency into application processing. In addition, the large amounts of data generated by LiDAR or camera systems would require significant bandwidth for the communications network. Edge processing can alleviate much of this challenge by performing the processing locally via direct connections to the sensors. Data that will be used locally is immediately available; data to be aggregated with other data for use nearby can be pushed to roadside "Fog[1]" processor nodes.
- **Fog Processor Node –** This element serves as an intermediary server at the roadside. It aggregates data from edge processors and sends the data to servers found in the Central Data Management Environment for storage and further processing/computing. These nodes can also disseminate real-time local traffic and infrastructure information to test vehicles and pedestrians/bicyclists.
- **RSU –** The Roadside Unit is the fixed DSRC unit that serves as the main application platform. These units are highly capable Linux computers with radios enabling low latency communications, enabling CV applications over distances of 400 – 500 meters with clear lines of sight.

[1] "Fog" is the term used to refer to intermediate data processing prior to the "cloud."

- **Network Security Appliance –** Network security will be enhanced through the use of industrial-grade security appliances that will be installed at nodes with public-facing communications. Similar security measures will also be considered in traffic signal controllers involved in the testing or equipped with public-facing applications.
- **Wi-Fi/Cellular Connectivity –** Local wireless connectivity will be provided via Wi-Fi nodes to minimize processing times and as an element of potential pedestrian/cyclist applications. Cellular connectivity will be used for connections to the Central Data Management Environment and between Fog Processor Nodes for a specific type of data.

Communication network

The communication architecture realizes the two-way communication between sensing and edge computing units and the central computing units at the Rutgers Smart Mobility Lab. The communication architecture may include the combination of wired and wireless communication through cellular or University network infrastructure.

The Connected Vehicle technologies should include the conventional C-V2X(Cellular Vehicle-to-Everything) physical and the emerging virtual roadside units (RSUs). Virtual RSU services are provided via Verizon's VZMODE Safety Messaging Framework, resident on MEC will provide access to safety data information collected from vehicles and infrastructure for data analysis and subsequent viewing. These virtual RSUs will be configurable via a simple user interface and can be placed anywhere in the Innovation Hub. Safety messages can be collected from OBUs mounted in the test vehicles and messages generated by ITS services and roadside sensors such as the Computer Vision (CV) software processing the live video streams of Bosch traffic safety cameras. Sensor messages include Basic Safety Message (BSM), which are generated for vehicles detected in the FoV, and Personal Safety Messages (PSM) generated for Pedestrians and other Vulnerable Road Users (VRUs) detected at crosswalks.

Cloud/Center Computing for 3D Data Processing and Modeling

The server or cloud-based 3D data analytic platforms will process, archive, and model the collected 3D LiDAR and Computer Vision data. The data will be denoised, anonymized, and mapped to the 3D space of the entire testbed. Object data such as vehicles, pedestrians, temporary road infrastructures (e.g., work zones), and infrastructure changes will be extracted and archived. The Central Data Management Environment layer represents the storage and application servers housed at the Traffic System Lab at Rutgers University for initial processing. Additional processing, as well as cloud-based storage capacity, will be provided by Amazon Web Services (AWS).

- **Central Computing Node –** Central computing node will process system-wide vehicle and object trajectories, consolidate, and reconcile overlapping objects from different sensors, and apply anonymization algorithms to mask pedestrians' personal identification information (PII).
- **Amazon Cloud Storage –** Additional storage and 3D rendering support will be provided through AWS cloud services to supplement local storage at Rutgers University.
- **Smart Mobility Data Portal –** Researchers and practitioners can access public data collected by test ground sensors and any applications developed by Rutgers researchers. Data will be cleansed and anonymized before being shared.
- 3D Holistic-View Video Wall System: The 3D object data generated will be imported into 3D

visualization and simulation models that will be displayed in a futuristic 360-view visualization theater for real-time inspection of traffic scenes from different angles.

In-Vehicle Smart Mobility Service Applications: The in-vehicle test platform will be developed based on the research team's existing mobile applications. The research team has developed a naturalistic driving data collection and adaptive traffic signal control. The team will integrate both platforms and establish a cellular-based V2I (Vehicle-to-Infrastructure) and C2V (Center-to-Vehicle) smart mobility application test platform.

Testing Interfaces for Mobility, Safety, Environment, and Energy Applications:

The SIMT will include the mobility, safety, environment, and energy application testing interfaces with service/controller-end and user-end interfaces. The interfaces will include testing interfaces for applications, such as adaptive traffic signal control and intersection safety applications, to improve the safety and mobility of drivers, pedestrians, and bicyclists. Initial implementations will take place within the testing ground, serving as pilot sites for NJDOT. The applications implemented or investigated by the Rutgers Team encompass the range of connected/advanced safety and mobility applications, including Smart Intersections, Smart Pedestrian Assistance, Incident Management Assistance, Infrastructure Maintenance Assistance support, Road-Weather Intelligence applications, and Advanced Roadside Driver Assistance applications.

Proposed Deployment Strategies

The proposed deployment includes the following four stages.

- **Stage 1**: On-Demand Platform and Application Development and Testing: In the first stage, the research team will focus on developing the critical sensing, computing, and communication systems while building all components on an on-demand platform that can be moved between the roadside and the labs.
- **Stage 2**: Roadside Instrumentation and Network Configuration: Once the full prototype system is developed. The research team will work with Middlesex County, the City of New Brunswick, and NJDOT to fully deploy the proposed systems.
- **Stage 3**: Computing, Analytics, and Visualization Platform Development: The research team will develop prototype applications to process the collected real-time high-resolution smart mobility data and create the 3D model for the proposed visualization theater for futuristic TMCs. This stage includes building both the static high-resolution environment of the test sites and mapping all dynamic 3D objects detected into the environment.
- **Stage 4**: Data Sharing and Application Testing Platform Development: The data sharing platform will focus on developing 3D high-resolution data packaging, archiving, and retrieval systems. Testing interfaces from both the user and service/controller end will also be developed for application testing.

Proposed Computer Vision Algorithms for SIMT

Existing Commercial CCTV Analytic Products

The CCTV (Closed Circuit Television) traffic surveillance systems are one of the essential assessments of traffic management centers (TMCs) to monitor congestion and incidents in daily operations. Computer vision sensors based on CCTV traffic cameras have been explored over the last four decades. Many platforms have the capability of crowdsourcing and on-demand video analytics. However, those systems often offer heavy-weight all-inclusive packages of smart city functionalities such as TSM&O 21 (Transportation System Management and Operations), energy and utility management, etc.

Existing traffic video analytic systems can be classified into three categories: integrated camera and analytic solutions, universal virtual sensor solutions, and cloud-based smart city video analytic solutions. The integrated camera and analytic solutions use image processing algorithms developed and fully calibrated for specific types of video cameras at specific scenes. Many of those systems even encode the image processing algorithms within the processing units directly connected to the cameras. Such tight integration allows those algorithms to take full advantage of the full resolution and quality of the raw Video to generate needed traffic data. Representative platforms include Autoscope and Gridsmart.

Universal virtual sensor platforms are essentially hardware independent and developed for typical intersection, or roadside traffic scenes often observed in existing CCTV traffic cameras. Some platforms can be deployed to existing traffic data sources, even PTZ (Pan-Tilt-Zoom) cameras (e.g., TrafficVision ()), with the capability of adjusting the "scanlines" or "detector zones" in PTZ operations. Representative systems include CitiLog, TrafficVision, MetroTech, GoodVision, Miovision, and KiwiVision.

The advance in central (CPU) and graphics processing unit (GPU) technologies and cloud computing triggers the current wave of cloud-based smart city video analytic solutions. Many platforms, such as Placemeter, MicroFocus (ex-HP), IBM Intelligent Video Analytics, Cisco Smart City Cloud, and BriefCam, have the capability of crowdsourcing and on-demand video analytics.

CCTV Traffic Surveillance/Smart City Applications for Intersections

Traffic operations and management heavily rely on the traffic data collected from intersections and roads. Efficient and reliable traffic sensing and detection empower traffic managers to objectively measure and assess traffic conditions, mitigate traffic congestion, and adjust traffic signal timing. We can effortlessly estimate intersection delay, travel time, and Level of Service (LOS) with high-resolution vehicle trajectory data. The trajectory data can contribute to incident management, active traffic control, and speed harmonizing, improving safety and mobility from individual intersections to the entire roadway network. In the era of connected and autonomous vehicles (CAV), vehicle trajectories obtained from roadside sensors play a critical role in many V2I (Vehicle to Infrastructure) applications such as Cooperative Adaptive Cruise Control, Dynamic Merge Assistance, and Eco-Traffic Signal Timing.

Compared to many other traffic sensing devices, traffic cameras have apparent advantages, which can be a cost-effective solution in the service of the infrastructure-based detector. Due to its capability to provide rich information and a large coverage area, CCTV Camera has been applied for vehicle speed measurement, traffic analytics, near-miss reporting, and incident review. The main disadvantage is that

vision sensors are sensitive to illumination changes. The video image processing relies on external illumination, resulting in lower accuracy at night. The video image detector is often used with other types of detectors to provide some back-ups, such as Radar or Remote Traffic Microwave Sensors (RTMSs).

The existing video-based vehicle detection and recognition methods can be categorized as motion- and model-based methods. The motion-based method employs motion information to segment moving objects from the traffic scene between consecutive frames. In contrast, using a pre-trained template, the model-based approach identifies objects based on their appearances. Typical motion-based methods include frame differencing, background subtraction, and optical flow. Model-based methods include a Histogram of Gradient (HOG) feature detector, Deformable Part Model, and deep learning models. The rise of AI (Artificial Intelligence) and deep learning has significantly advanced image object detection in recent years. Video-based intersection surveillance will become an indispensable part of smart city applications. In the future, CAV technology such as Eco-intersection Approach and Intelligent Signal Control will lead to more harmonized speed characteristics. The performance metrics generated from trajectory data are critical for CAV-based traffic operation, as they can provide proactive solutions and depict a better picture of traffic networks.

Another commonly seen traffic video analysis is scanline-based. A scanline is a group of pixels on selected lanes used for object detection and tracking. There are two types of scanlines: the latitudinal scanline, which is defined across the traveling path (Tseng, Lin & Smith, 2002; Zhang et al., 2007; Mithun, Rashid & Rahman, 2012; Ren et al., 2014). The other is the longitudinal scanline that is defined along the traveling direction (Cho & Rice, 2007; Malinovskiy et al., 2009; Ardestani et al., 2016; Zhang and Jin, 2019). Previous scanline-based vehicle detection can only produce spot-specific traffic parameters, such as volume, vehicle type, and spot speed. In this study, we used longitudinal scanlines to extract the vehicle trajectory in the urban arterial with signal-controlled intersections, which deals with more complex scenarios than previous scanline applications.

Traffic Camera Calibration

Camera calibration provides a mapping relationship between real-world coordinates and 2D images, which is the foundation for extracting vehicle trajectories, measuring speeds, and acquiring other traffic information from video footage. Some camera calibration techniques require to detect the vanishing point (VP) in the 2D image, which is the point on the image plane formed by the convergence of mutually parallel lines in three-dimensional. Others are using reference objects for calculating the camera-pose-based perspective transformation. Dailey, Cathey, and Pumrin (2000) developed an algorithm to estimate mean traffic speed from uncalibrated cameras without knowing information such as camera focus, tilt, or angle. Their algorithm is constrained to several assumptions, such as the limitation of the speed of the vehicle, motion constraints on the road plain, linear change of scale factor, and known vehicle length distribution. Schoepflin and Dailey (2003) presented a three-stage method to calibrate the roadside camera into a speed sensor for traffic management. Their model used the motion of the vehicle to estimate the camera position and calibrate the camera by determining the vanishing point of the roadway. Cathey and Dailey (2005) proposed an algorithm to calibrate PTZ (pan, tilt, zoom) camera, consisting of three phases, (1) lane boundary detection, (2) computation of vanishing point and

image straightening transformation, (3) calculation of the image-to-highway scale factor (feet per pixel). Grammatikopoulos, Karras, and Petsa (2007) developed an approach for automatically estimating camera parameters (camera constant, principal point location, and two radial lens distortion coefficients) from images with three vanishing points of orthogonal directions. Dubská et al. (2015) proposed a fully automatic camera calibration method without the manual setting under various road conditions. Their approach detects and tracks local feature points of moving vehicles and uses the trajectories of tracked points to obtain vanishing points corresponding to the direction of moving vehicles. Luvizon et al. (2016) used the planar of the inductive loop detector as a reference object to construct a homography matrix for measuring vehicle speed from license plate detection. Do et al. (2015) developed a method of calibration to measure traffic speed by drawing an equilateral triangle on the ground as a 2D reference object. Then they solve the three configuration parameters of height h, the tilt angle ψ, and the focus distance f. You and Zheng (2016) developed a dynamic calibration method by obtaining two vanishing points: the vanishing point in the direction of the traveled lane and the orthogonal vanishing point.

More recently, Sochor, Jranek, and Herout (2017) developed a deep learning model to assign a 3D bounding box for the detected vehicle. Based on the outputs of the deep learning model, they can obtain two vanishing points for camera calibration. Their result reduced the distance ratio error of vanishing point detection from 0.18 to 0.09, which beat the previous state-of-the-art model. Sochor et al. (2018) established a benchmark dataset for evaluating different traffic camera calibration methods. The speed of vehicles in the dataset was collected using LiDAR and verified through GPS trackers. Bhardwaj et al. (2018) proposed the AutoCalib system for scalable, automatic calibration of traffic cameras, using a deep learning model to extract selected key-point features from vehicle images to produce a robust estimate of the camera calibration parameters automatically. Their model relies on the car's known geometric parameters (e.g., the distance between the two taillights).

Most of the camera calibration algorithms are based on prior knowledge. Some traffic camera calibration methods mentioned above are based on vanishing points inferred from moving objects, which causes those models to be sensitive to environmental variations. In practice, other models based on reference objects are hard to deploy because traffic operators cannot move the reference object every time the PTZ-camera scene changes.

LiDAR Data Analytics

Sun, Y. et al. proposed a vehicle trajectories extraction system that used region of interest (ROI) to roughly filter the LiDAR data and then used mean distance to separate the inliers and outliers, significantly reducing the processing workload. In addition, Random Sample Consensus (RANSAC) was used for ground plane segmentation. In 2018, Wu et al. proposed a novel method for vehicle tracking and an automatic background filtering method with 3-dimensional density statistic background filtering (3D-DSF), frame aggregation, point statistics, threshold learning, and real-time filtering. In 2019, Wu et al. developed a novel algorithm for ground points identification and excluding, which includes four parts: density-based spatial clustering of applications with noise (DBSCAN)-based points clustering, slope-based filtering, shape-based filtering, and ground points matrix extraction. Wu et al. proposed a new vehicle classification approach. The paper firstly used 3D-DSF for background filtering, lane

identification, multirectified density-based spatial clustering of applications with noise (MCDBSCAN) for lane identification, DBSCAN for object clustering, global nearest neighbor (GNN) for object tracking, and then selected six features for vehicle classification based on Naive Bayes, KNN, decision trees, SVM, and ANN. Wu et al. developed a semi-automatic approach for roadside LiDAR points registration, which could not only improve the effectiveness and accuracy of data integration but also help reduce the impact of the occlusion issue on object detection.

Mobile LiDAR Technology and LiDAR-Camera Integration

LiDAR sensors, including mobile LiDAR, airborne, and static LiDAR, have been extensively used in transportation studies like vehicle and pedestrian detection, object localization, and trajectory tracking. With the rise of the self-driving car, LiDAR-based mapping service and sensing technology will play a critical role in self-driving vehicles routinely executing complex maneuvers. A wide range of spatial information can be extracted from LiDAR point cloud data, including road level (e.g., road surface, lane markers, driving lines, cracks, and utility holes), object-level analysis (e.g., buildings, trees, vehicles, and power line) to building-structure element level (e.g., façade, doors, windows, roofs) analysis.

Much research has explored the use of LiDAR for automated urban on-road object detection and extraction (Williams et al., 2013; Guan et al., 2016; Ma et al., 2018). For example, Zai et al. (2018) proposed an effective 3-D road boundary extraction by employing super-voxels and graph cuts on MLS (Mobile LiDAR System) data. Other studies, like Xu et al. (2017), developed a method for automatic extraction of road curbs. They evaluated their method on a large scale of residential and urban area mobile LiDAR point clouds. Additionally, Yang (2012) presented a technique that can realize the automated extraction of road markings from mobile LiDAR point clouds. In this study, 3-D point clouds were converted into 2-D geo-referenced feature images, and road markings were filtered by controlling LiDAR intensity and elevation value. Finally, road marking outlines were extracted based on prior knowledge of road marking shape and arrangement. Yu et al. (2015) proposed an algorithm that uses a multi-thread computing strategy to detect urban road utility hole covers with MLS data. Other published studies focus on automated urban object extraction, including traffic signs, trees, buildings, vehicles, powerlines, etc. (Engelmann et al., 2017; Ye et al., 2018; Che et al., 2019; Wu et al., 2017). Yang et al. (2015) proposed a method for urban object extraction with Mobile LiDAR data. They generated multi-scale super-voxels and reduced computing costs by segmenting super-voxels. Finally, their approach was validated with large data sets and achieved accuracy between 90% and 96%. Some studies focus on building element extraction from MLS data. For example, MLS data have been successfully used in window and façade detection studies by Wang et al. (2011) and Arachchige et al. (2012).

Another critical topic about LiDAR is the sensor fusion of LiDAR and camera, which has received increasing attention over the years. Cameras provide rich texture and color information, while LiDAR provides accurate spatial data. Fusing them can provide depth information for the pixels in the camera image with reliable 3D point clouds, which are helpful in velocity estimation for precise vehicle tracking and autonomous driving. Extensive studies have explored the registration between LiDAR and camera imagery. The most common approaches require the existence of known targets in the scene (Zhang et al., 2004; Fremont and Bonnifait, 2008; Li et al., 2007; Naroditsky et al., 2011; Vel'as et al., 2014). These studies used checkerboards and other types of targets (e.g., triangles, circles, or white-to-black

transitions) that are observable by both LiDAR and the camera. For example, Zhang et al. (2004) exploited a planar checkerboard and used nonlinear least-squares optimizations to calibrate a single optical camera with a 2D scanner. In Narodistsky et al. (2011) 's study, the calibration problem is described as a set of polynomial equations, and six correspondences are minimally required for the alignment of the LiDAR-camera system. Recently, more researchers have attempted to automate the calibration process using features present in the observed scene without markers or targets. For example, Pandey et al. (2015) addressed automatic targetless extrinsic calibration by maximizing mutual information between image and 3D LiDAR camera. It used the known intrinsic value of the camera and estimated the extrinsic parameters to project LiDAR onto camera imagery. The mutual information value is computed by comparing the LiDAR reflectivity with the intensities value from camera images. In another study proposed by Li et al. (2018), the registration of panoramic image sequences and mobile laser scanning point clouds in the urban environment was estimated using parked vehicles as registration primitives.

In contrast, there has been minimal research on PTZ camera calibration using infrastructure 3D LiDAR data for vehicle trajectory detection. Previously-published studies of the combination of LiDAR (for range information) and camera systems ("for better recognition") have focused on the dynamic data fusion between image objects detected in traffic video and the corresponding 3D point cloud clusters identified in mobile LiDAR data. In this paper, however, the focus is on using static LiDAR 3D point cloud to assist the physical trajectory extraction. Moreover, those methods assume the LiDAR-Camera alignment can be accurately estimated when the camera and LiDAR capture the same scene on a mobile platform. Such an assumption cannot apply to the proposed LiDAR-Camera system for vehicle trajectory detection. The camera and LiDAR capture data at the same time and location in existing papers (Vel'as et al., 2014; Li et al., 2018; Pandey et al., 2015). However, in this paper, the traffic cameras capture the dynamic roadway condition, while the pre-collected static LiDAR 3D model is used as the basis for perspective and coordinate transformation.

Proposed 2D Video and 3D Infrastructure Data Integration Workflow

It takes two steps to acquire useful traffic trajectory data from video sensors. The first step is extracting vehicle movements from continuous video frames. The second step is converting the video coordinates to real-world GPS coordinates to get the real traffic parameters, such as speed, density, and queue length.

Figure 2 illustrates the overall workflow for LiDAR-assisted Traffic Video Analysis. The main Video analytic workflow is depicted on the left branch, where raw video data are processed and analyzed to generate pixel trajectory. In contrast, the right branch uses LiDAR data to conduct 2D-3D feature matching to convert the pixel coordinates into world coordinates to generate physical trajectories.

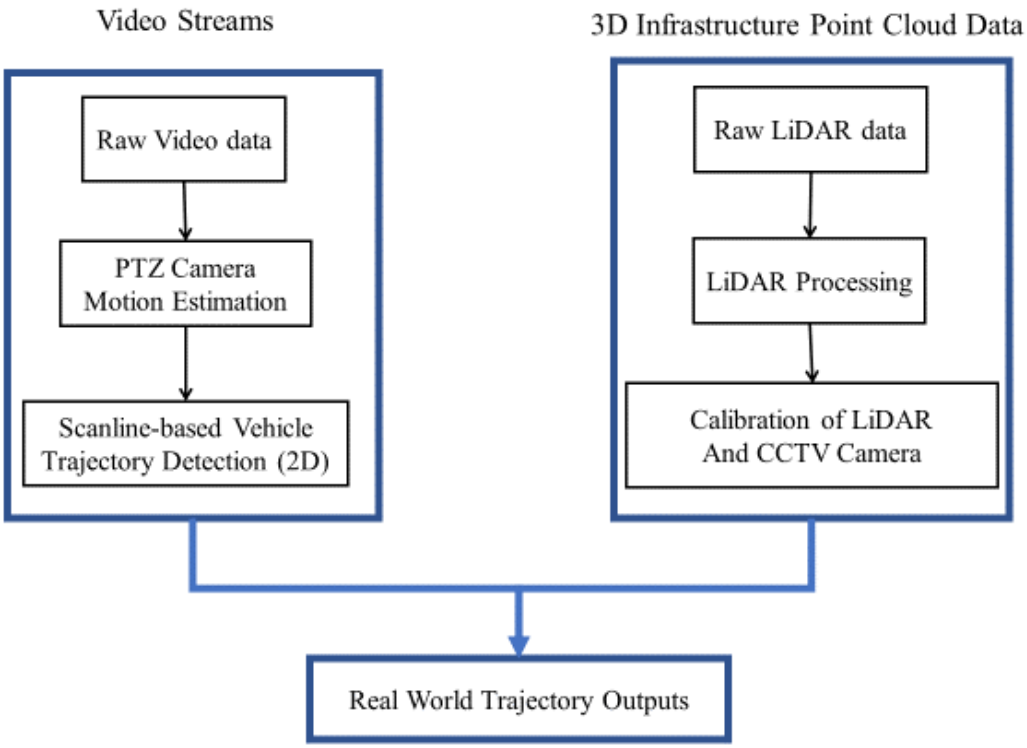

Figure 2. Dataflow of LiDAR-assisted Traffic Video Analysis

Scanline-based Trajectory Extraction

The scanline methods in the literature review are used on straight highway segments. While for this paper, the scanline method was modified for an urban signalized intersection to deal with more complex scenarios in terms of road geometry, crossing vehicles, and roadside infrastructure. The scanline-based trajectory extraction consists of four major steps: spatial-temporal map (STmap) generation, preprocessing, vehicle strands detection, and pixel trajectory detection. Using STmap for vehicle trajectory extraction, the conventional two-step trajectory extraction algorithm consisting of object detection and tracking is simplified as one step algorithm by segmenting out the vehicle strands on STmap.

STmap is stacked scanline pixels from each video frame. The vertical axis of the STmap can be converted into the pixel distance. The horizontal axis represents the frame number. The STmap preserves trajectories of any moving objects passing along the scanline. Figure 3 illustrates the scanlines in our testing videos, which covered nine lanes at the New Brunswick train station intersection.

Figure 3. Scanline-Based Vehicle Trajectory Detection

- **Spatial-temporal Map Generation**

First, a piecewise scanline was defined to generate a spatial-temporal diagram by selecting turning points along the traveled lanes. Bresenham's line algorithm is used to get the pixel coordinates between two turning points. The following is the algorithm.

Bresenham's Line Pixel Algorithm

==========

Input: (X_{start}, Y_{start}) and (X_{end}, Y_{end})

Outputs: pixel coordinates of the straight line between (X_{start}, Y_{start}) and (X_{end}, Y_{end})

Algorithm:

Calculate constants: $\Delta x = X_{end} - X_{start};\ \Delta y = Y_{end} - Y_{start}$

Assign value to the starting parameters: $k = 0,\ p_0 = 2\Delta y - \Delta x$

Obtain (X_{start}, Y_{start})

while $x_k < X_{end}$ (note that x_k is integer x-coordinate along the line)

if $p_k < 0$

obtain pixel at $(x_k + 1, y_k)$

$p_{k+1} = p_k + 2\Delta y$

Else

obtain pixel at $(x_k + 1, y_k + 1)$

$p_{k+1} = p_k + 2\Delta y - 2\Delta x$

$k \mathrel{+}= 1$

==========

After obtaining the pixel coordinates of scanline points, the scanline points are stacked together from every frame, eventually showing the movements of objects passing along the scanline. On completion of the stacking scanlines from every video frame, the STmap is created. Each strand on the STmap represents a unique object (Figure 4).

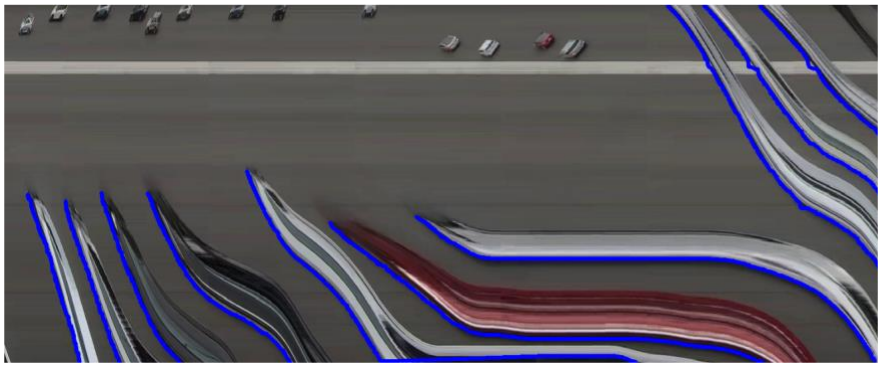

Figure 4. STmap and Vehicle Trajectory

- **Preprocessing**

Preprocessing is necessary to remove the noises before trajectory extraction. Preprocessing has three two modules, static noise removal and shadow detection.

- **Static Noises Removal**

A significant challenge in the urban scenario is static noises caused by lane markers, traffic signs, poles, and other infrastructure objects that might leave multiple horizontal marks on STmap. To remove the static noises, we first identify static object areas with the horizontal lines on STmap. Then we replace the horizontal line with the value of the nearest non-noise pixels. The results are shown in Figure 5.

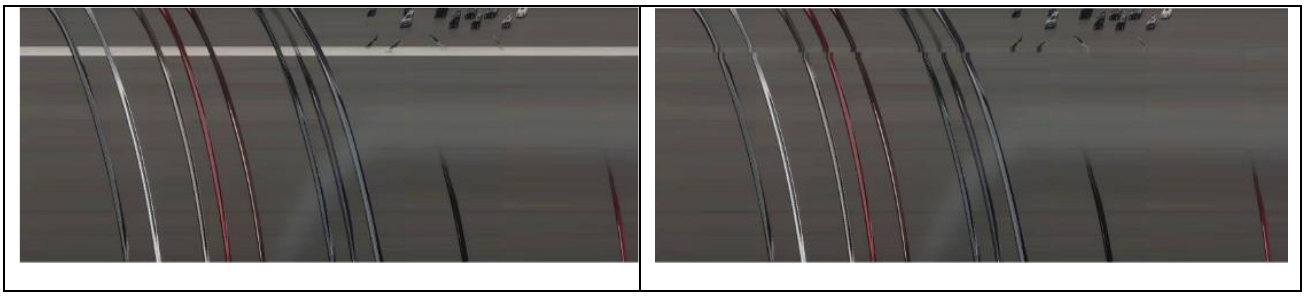

Figure 5. STmap and Vehicle Strands

Algorithm 1: Remove Static Noise

Input:

RGB Spatial-Temporal Diagram: S

Parameter: noise_threshold

Output:

Noise-Free Spatial-Temporal Diagram: S

R is row number of $\mathrm{S}(r,t)$

T is column number of $S(r,t)$

$\mathrm{S_e}$ is Canny Edge detection results of S

Do Morphological Dilation on $\mathrm{S_e}$

For every row $\mathrm{r_0}$ **in** $S(r,t)$ **do**

If $(\sum_{t=1}^{t=T} \mathrm{S_e}\ (\mathrm{r_0},\mathrm{t}))/\ \mathrm{T} >$ noise_threshold **then**

Save the Row Number in $\mathrm{R_{noise}}$

End if

End for

For every row $\mathrm{r_1}$ in $\mathrm{R_{noise}}$ **do**

Replace row $\mathrm{r_1}$ in S with mean value of K-nearest neighbor pixels that are not in $\mathrm{R_{noise}}$

End For

Return S

- **Remove Shadows**

Shadow detection and suppression is an active research field of image processing. Shadows cast from objects often contaminate STmap and result in errors for trajectory detection. However, shadow pixels have their attributes, often characterized as lower intensity and fewer textures (see Figure 6).

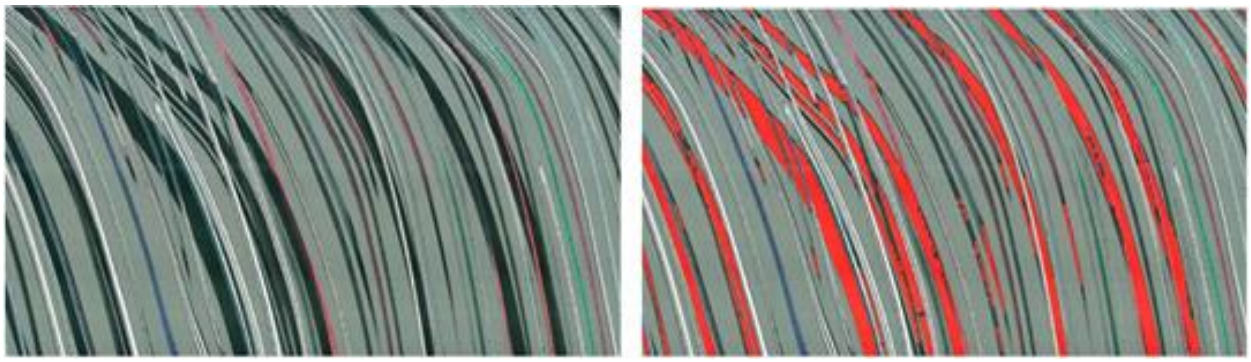

Figure 6. Shadow Detection on STmap

Algorithm 2: Detect and Remove Shadow

Input:

RGB Spatial Temporal Diagram S

Parameters: shadow_mean, shadow_variance

Output:

Shadow Free matrix: S_{new}

W: n by n moving window

G: Grey level STmap

For every pixel (r, t) **in** ST Map **do**

Consider a 3 by 3 window: **W = G** (r-1: r+1, t-1: t+1)

Compute mean value of window **W**: Mean(**W**)

Compute variance of window **W**: Var**(W)**

If Mean**(W) <** shadow_mean **and** Var**(W) <** shadow_variance **then**

Replace $S(r, t)$ with a median RGB value of row r of S

End if

End for

$S_{new} = S$

End for

Return S_{new}

- **Stands Detection**

A group of pixels on STmap that shows the moving object's trajectory is named vehicle strands. During this step, three main features of STmap are utilized to segment the vehicle strands, including edge, color, and motion.

- **Edge Detection**

The first feature is the edge feature, and the Canny edge detector is used due to its robustness. However, the outputs of the Canny Edge detector are incomplete and often lead to cracked segments, as shown in Figure 7.

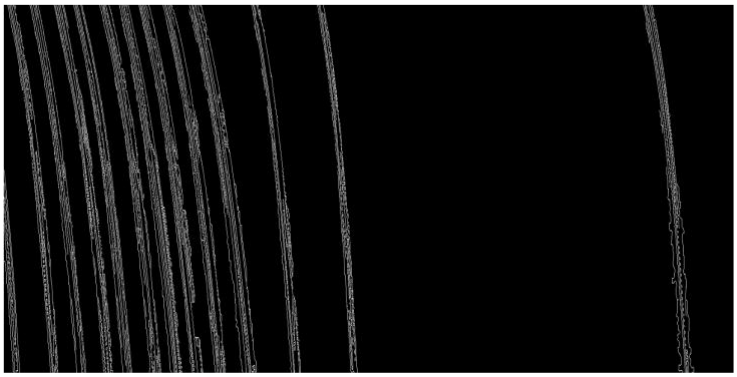

Figure 7. Edge Detection Results for Vehicle Strands

- **Background Detection**

Its color information is utilized to analyze further the STmap, which uses histogram thresholding for Background detection. This step builds the background model and compares the pixel value with the background model to separate moving objects. For the background detection method, we assume the intensity level of roadway pavement follows a normal distribution, while the vehicle textures are usually randomly distributed. The roadway pavement and vehicle strands often occupy different ranges, as shown in histogram Figure 8(a).

The probability of a pixel z given by the intensity distribution is expressed in equation $\boldsymbol{P(z) = P_b * p_b(z) + P_v * p_v(z)}$ (**1**).

$$\boldsymbol{P(z) = P_b * p_b(z) + P_v * p_v(z)} \quad (1)$$

Where $p_b(z)$ is the probability distribution of the background, $p_v(z)$ is the probability distribution of vehicles, P_b is a priori probabilities of background, and P_v is a priori probability of vehicles.

Considering the background roadway is the majority, the road pixel intensities can be defined by background thresholds (T_1, T_2). To find the optimum background threshold is described in algorithm 3. Figure 8(b) shows the results of replacing all background pixels with a uniform color.

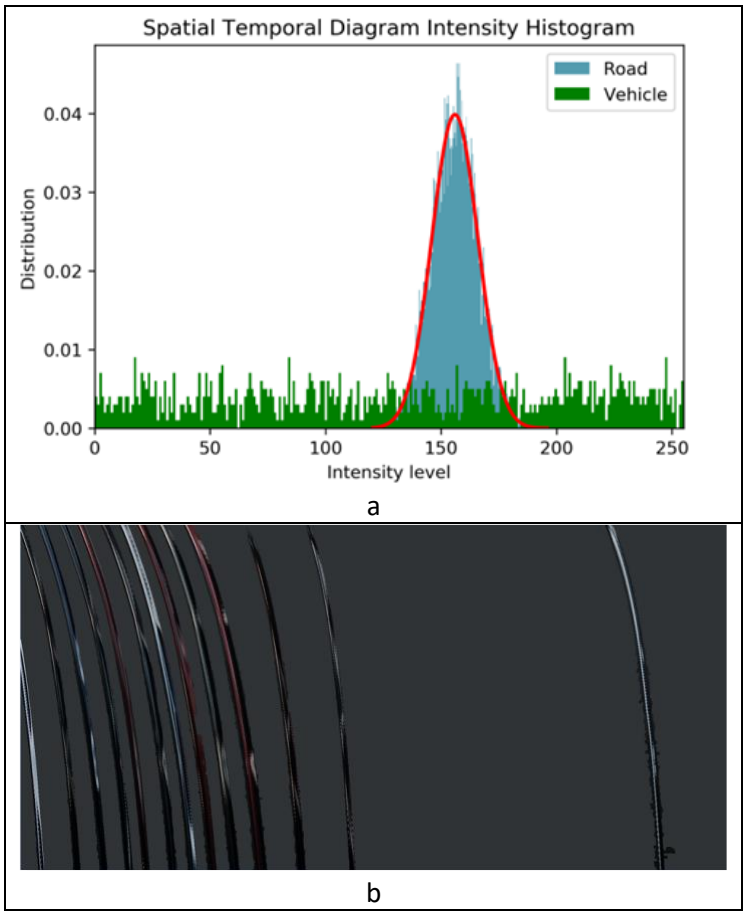

Figure 8. Background Model and Thresholding Based Histogram Analysis

Algorithm 3: Histogram Based Background Detection

Input:

RGB Spatial-Temporal Map: S

Output:

Spatial-Temporal Map with uniform Background: S_s

Compute the median RBG value of STMap (R_m, G_m, B_m).
Convert S to Grey level image G.

For each row r in G: **do**
Compute the histogram of intensity distribution H(r)
Find the valleys of H(r) on both sides as (T_1, T_2)
If $pixel(r) \geq T_1$ **and** $pixel(r) \leq T_2$
Set $pixel\ (r)$ on S as (R_m, G_m, B_m).
End
for $S_s = S$
Return S_s

- **Time Differencing**

Time differencing is to use the motion feature on the STmap, which essentially does the frame differencing method on the STmap, as the STmap is the progression of a group of pixels over time. The following equation describes how we obtain the time differencing result. Detection results are shown in Figure 9.

$$ST_t(r,t) = \begin{cases} 1, & \left|ST_{Gray}(r,t) - ST_{Gray}(r,t-\Delta t)\right| \geq thrld \ \ and \ \left|ST_{Gray}(r,t) - ST_G(r,t+\Delta t)\right| \geq thrld \\ 0, & otherwise \end{cases} \quad (2)$$

where ST_t is the binary image output from time differencing method. Vehicle strands are labeled as 1, and background pixels are labeled as 0. (r,t) are row and column indexes for each pixel on ST diagram. ST_{Gray} is a gray-level STmap, Δt is the parameter, and its suggested value is 2. $Thrld$ is the threshold parameter for differencing detection, and the suggested value is 10.

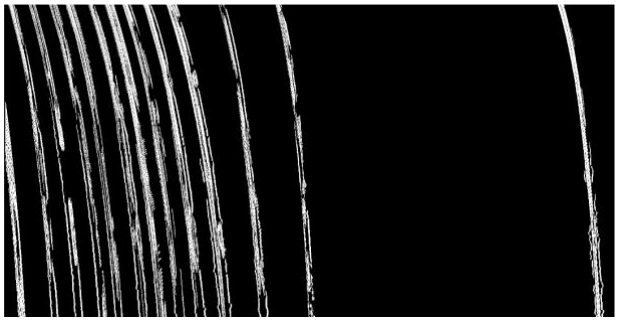

Figure 9. Strands Detection Results Using Time Differencing

After edge detection, background detection, and time-differencing, we combine the three results to get clean and complete vehicle strands. There are also crossing vehicles on STmap, frequently observed for intersection scenarios. However, the crossing cars can be removed based on occupancy (duration) or height/width ratio.

- ***Pixel Trajectory Extraction***

Since we draw the scanline in the same direction as the direction of traffic, all trajectories always go from the top-left to the right-bottom on the STmap. The edges of vehicle strands accurately record the front bumpers of vehicles. The goal of this step is to identify the boundaries of each vehicle strand. Therefore, the complete movement of the car along the scanline can be obtained. On completion of trajectory profiles on STmap, we can acquire the vehicle trajectories in video image coordinates, as we know the selected video pixel coordinates of all points of the scanline. The results of generated trajectory profiles on STmap are plotted in Figure 10.

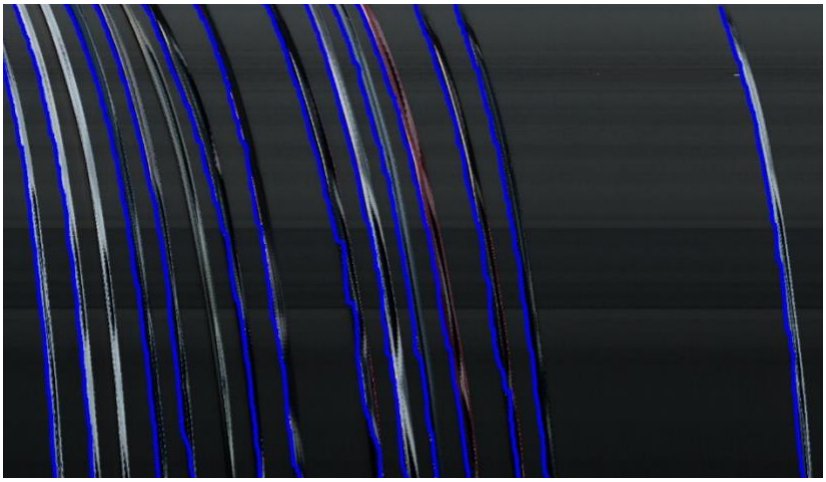

Figure 10. Detected Pixel Trajectory on STmap

LiDAR Processing and Camera Calibration

- ***Estimating Video Distortion***

Correcting lens distortion is critical to an accurate projection result. Without a reasonable estimate of the camera distortion, it is difficult to calculate the precise projection between the video frame and point cloud. The camera calibration and lens un-distortion steps are implemented with the OpenCV toolbox.

- ***Raw LiDAR Processing***

The New Brunswick mobile LiDAR dataset is hosted in the online mapping system (Figure 11b). And LiDAR data can be retrieved by entering the GPS information of the study area (40.496326 N and -74.446131 W). The raw LiDAR point cloud obtained from the online mapping system is shown in Figure 11d. After this step, we removed the highlighted building in Figure 11c, which blocks the studied area (see Figure 11e). Then, we removed the unnecessary point cloud from the camera view and cleaned the

target area by eliminating the noise points. The points belong to vehicles, pedestrians, trees, etc. The study area of the point cloud model after cleaning is shown in Figure 11f.

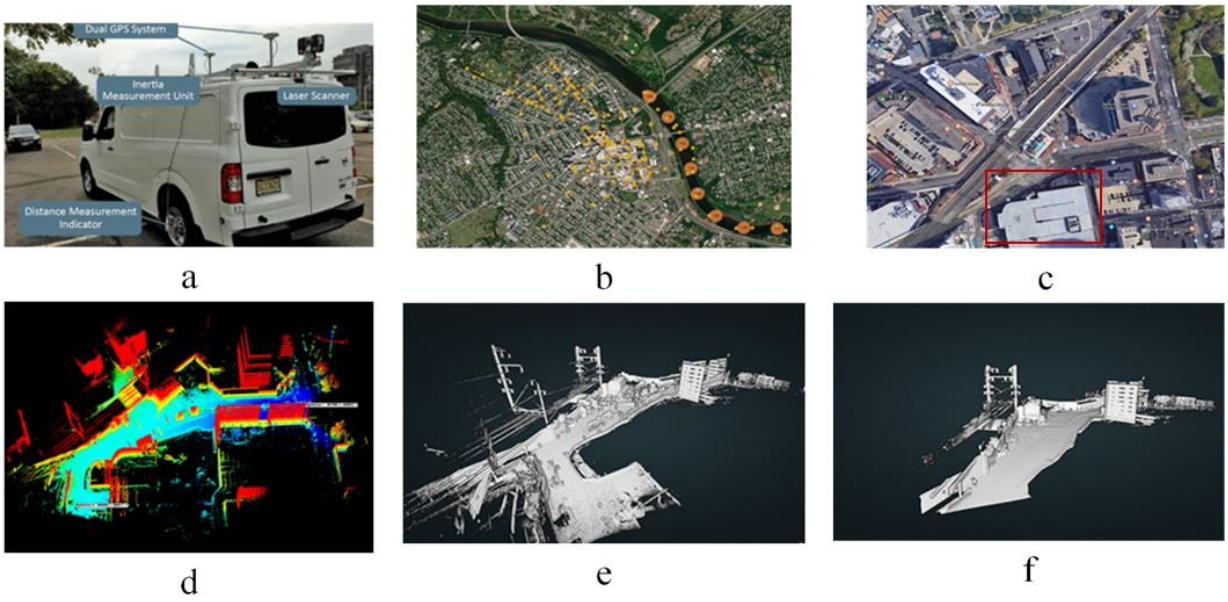

Figure 11. LiDAR Data Collection and Processing Procedure
(a) Rutgers Mobile LiDAR System; (b) New Brunswick Mobile Mapping Database;(c) Study Area on Google Map; (d) Raw LiDAR Data (e) LiDAR Data of Test Site Before Cleaning; (f) LiDAR Data After Cleaning

- ***Camera Calibration with 3D LiDAR Data***

The camera calibration process is to identify the relationship between image pixels with real-world coordinates, where both intrinsic and extrinsic parameters determine the relationship. Intrinsic parameters are fixed values composed of focal length, optical center, and screw coefficients. Extrinsic parameters are usually decomposed to rotation and translation concerning world coordinates.

Figure 12 shows how to relate the trajectory points from STmap to video image coordinates and then transform the trajectory points into real-world coordinates. The STmap coordinates and video image translation is already known during the STmap Generation step because the row coordinate (r) in STmap (r, t) is the order of generated points using Bresenham's line algorithm.

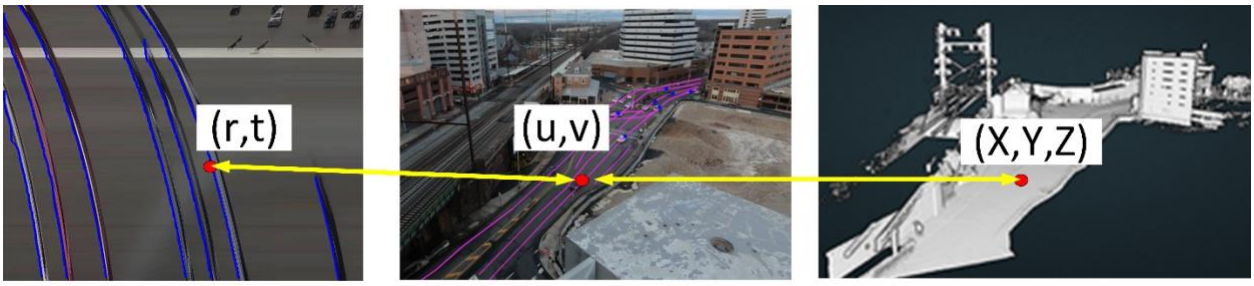

Figure 12. Three Coordinate Systems in LiDAR-Camera System Using Scanline Method

The following section will explain how to link video coordinates (u, v) to real-world GPS coordinates (X, Y, Z) using matched features on both the 2D camera and 3D LiDAR model. The relationship between 2D points and 3D points is represented as the equation (2):

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (3)$$

Where,
(u, v) video image pixel coordinate for a reference point
$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ is world GPS coordinates for a reference point
λ is a scalar
$K_{int} =$ intrinsic parameters
$K_{ext} =$ extrinsic parameters
$P = K_{int} k_{ext}, \;\; P \; is \; 3 * 4 \; projective \; matrix$

The Intrinsic parameter can be obtained through camera calibration in the lab or from known camera model parameters. The method to compute matrix P given intrinsic parameter is called the PnP problem. "Given n (n ≥ 3) 3D reference points in the object framework and their corresponding 2D projections, to determine the orientation and position of a fully calibrated perspective camera is known as the perspective-n-point (PnP) problem" (Hartley and Zisserman, 2003). The following equations describe how to solve the PnP problem using reference points.

Equation (2) can be rewritten as equation (3)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} X \qquad (4)$$

Where P_i is the ith row in P, X is the world coordinate of reference point.

$$u = \frac{P_1 X}{P_3 X} \qquad (5)$$

$$\mathrm{v} = \frac{P_2 X}{P_3 X} \quad (6)$$

Equations (4) and (5) can be written as:

$$(\mathrm{P}_1 - uP_3)X = 0 \quad (7)$$

$$(P_2 - vP_3)X = 0 \quad (8)$$

By rearranging the items, we obtain equation (8) as:

$$\begin{pmatrix} X^T & 0^T & -uX^T \\ 0^T & X^T & -vX^T \end{pmatrix} \begin{pmatrix} P_1^T \\ P_2^T \\ P_3^T \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (9)$$

For n points, we can stack equation (8) for all reference points into a big equation:

$$\begin{pmatrix} X_1^T & 0^T & -u_1 X_1^T \\ 0^T & X_1^T & -v_1 X_1^T \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ X_n^T & 0^T & -u_n X_n^T \\ 0^T & X_n^T & -v_n X_n^T \end{pmatrix} \begin{pmatrix} P_1^T \\ P_2^T \\ P_3^T \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{pmatrix} \quad (10)$$

Equation (9) can be simply represented as the equation (10)

$$\mathrm{AX} = 0 \quad (11)$$

Where A is a 2n * 12 matrix, which is known from 3D and 2D reference points;
X is 12 by one matrix that contains all parameters in projection matrix P.

The problem to solve parameter in P is converted to the problem to minimize $\|AX\|^2$, which can be considered as the least square problem.

As we know, the projection matrix $\mathrm{P} = K_{int} * [\mathrm{R}\ \mathrm{t}]$, where R is the rotation matrix, t is the translation vector. So, the rotation matrix can be recovered through equation (12).

$$\mathrm{R} = K_{int}^{-1}\ \mathrm{P}_{1:3} \quad (12)$$

Where $\mathrm{P}_{1:3}$ is the first three columns of the projection matrix P.

To enforce the orthogonal property of rotation matrix R, we need to do the Singular Value Decomposition (SVD) in equation (12).

$$UDV^T = R \quad (13)$$

Then we obtain the optimized rotation matrix R and translation vector t through the equation below.

$$\mathrm{R}^+ = UV^T \quad (14)$$

$$\mathrm{t} = \mathrm{K}^{-1}P_4 \,/\sigma_1, \text{ where } \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3) = D. \quad (15)$$

Therefore, we reconstruct the projection matrix P through the equation below.

$$\mathrm{P} = \mathrm{K}\,[\,\mathrm{R}^+ \; t] \quad (16)$$

OpenCV's Camera Calibration and 3D Reconstruction API (Application Programming Interface) is used in this research to obtain all projection matrix parameters (Docs.opencv.org, 2019).

PTZ Camera Recalibration using Motion Estimation

One crucial issue for traffic monitoring is the ever-changing remote-controlled PTZ cameras. In our system, The LiDAR-Camera model mentioned above is initially well-calibrated when the traffic camera is used. To restore the 3D/ 2D relationships of the PTZ camera, the relative camera motion between the pre-calibrated camera and zoomed/rotated camera is identified. Motion estimation has many applications, including object tracking, human-computer interaction (HCI), Visual Odometry, etc. Motion estimation methods can be classified into direct and indirect methods. Direct methods include phase correlation, block matching, and optical flow. Indirect methods often refer to feature-based methods (Katsaggelos, 2019). This study uses the indirect motion estimation method to estimate the camera movement.

Figure 13 shows the matched SIFT (scale-invariant feature transform) features between a calibrated camera and a moving camera. Once the matched features are found, we can establish the coordinate system transformation between the calibrated camera image and the real-time camera using projective transformation. Any pixel from the moved PTZ camera will be projected onto the pre-calibrated camera image. Therefore, the PTZ camera 2D coordinates can be transformed into 3D coordinates using the calibrated LiDAR-Camera system.

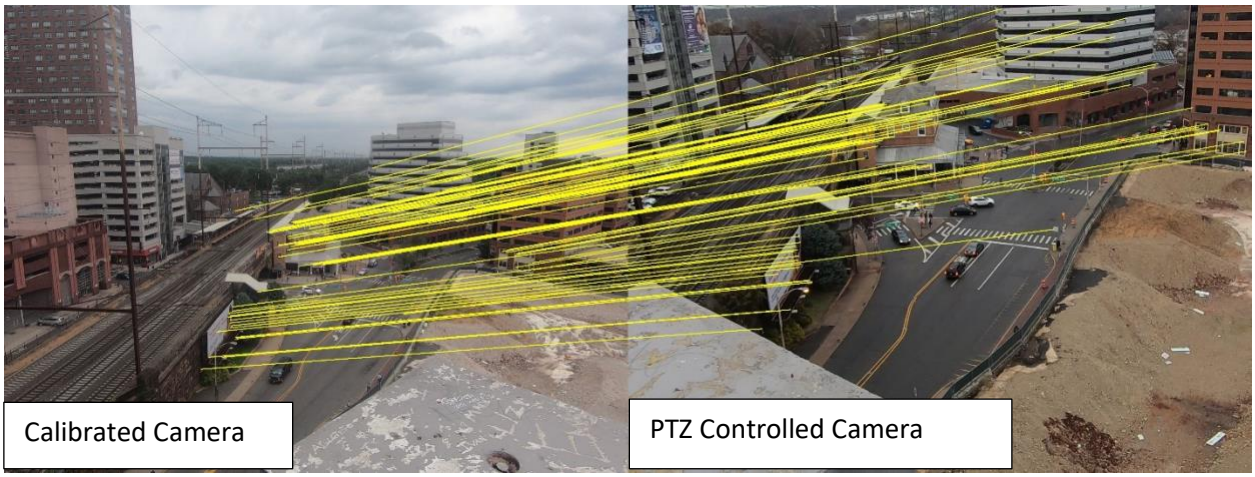

Figure 13. Feature matching between the original image and Rotated-Zoomed image

Multiple images from different angles will be pre-calibrated using the LiDAR model during the initial stage to cover the entire surveillance area. The pre-calibrated camera images will be used as static data. When the traffic operator moves the PTZ camera, the program automatically finds the best match from candidate pre-calibrated images to build a new 2D/3D transformation. This method indirectly recalibrates the PTZ camera by matching the new camera scene with pre-calibrated photos, resulting in better accuracy and quick response.

SIMT Digital Twin Modeling

Infrastructure Data Collection and Modeling

The infrastructure data was collected through FARO Laser Scanner Focus3D (Figure 14), which offers numerous advantages over conventional procedures. The scanner is lightweight and robust, so it can easily be transported to any site. The scanner can be set up in minutes and operated using a user-friendly touch screen. The laser scanner uses ball-shaped target spheres as a reference to connect individual scans. These spheres are placed in various locations and should be seen in each scan.

Figure 14. FARO Laser Scanner Focus3D

Building Digital Twin Model based on LiDAR Data

A game engine named Unity is used to establish the Digital Twin Model for the potential needs of autonomous vehicle simulation. Unity is the creator of the world's most widely used real-time 3D development platform that gives developers around the world the tools to create rich, interactive 2D, 3D, VR (Virtual Reality) and AR (Augmented Reality) experiences. It has been used in various industries, including games, film and entertainment, automotive, architecture, and construction and engineering (AEC). Unity's 1,000-person development team keeps the company at the bleeding edge of technology by working alongside partners such as Google, Facebook, Magic Leap, Oculus, and Microsoft. The platform's renowned flexibility gives developers the power to target and optimize their creations for over 25platforms. Games and experiences with Unity have reached 3 billion devices worldwide and installed more than 28 billion times in the last 12 months. Unity offers solutions and services for connecting with audiences, including Unity Ads, Unity Analytics, Unity Asset Store, Unity Cloud Build, Unity Collaborate, Unity Connect, and Unity Certification. Unity is also closely collaborating with autonomous vehicle companies like Baidu Inc. Unity can develop a real-time simulation product that creates virtual environments allowing developers to test autonomous vehicles in simulated real-world situations. The Unity-based real-time simulation will now be available to developers taking part in Baidu's Apollo platform: an open, comprehensive, and reliable platform geared towards accelerating the development, testing, and deployment of Levels 3, 4, and 5 autonomous vehicles.

Using simulated scenarios for testing in the automotive industry is a well-established practice. However, the scenarios used in the past, for example, to train ABS brake systems, do not suffice for autonomous vehicle training. Essentially, autonomous vehicles must be trained to behave like humans, requiring highly complex simulations. A vital part of any autonomous vehicle training simulation is the simulation environment, aka. the digital twin model. Unity, the real-time 3D rendering platform, is being used by engineering teams from Baidu, BMW, Toyota, etc., to efficiently create simulation environments for

autonomous vehicle training rich in sensory and physical complexity, provide compelling cognitive challenges, and support dynamic multi-agent interaction.

Our team collected high-resolution LiDAR point cloud data in this study to assist our 3D model development. Our team focuses on manually developing the Digital Twin Model of the infrastructure of the studied intersection. The Autodesk Recap and Revit are usually used to model shapes, structures, and systems in 3D with parametric accuracy, precision, and ease, which helps us create a high-accuracy digital twin model.

The final digital twin model in Unity is developed through the following steps:

1. Once the LiDAR point cloud data is collected, the point cloud can be loaded into Autodesk Recap as the base 3D map of the model creation.
2. The Autodesk Recap file can be loaded into Autodesk Revit for manual modeling.
3. The 3D model will be manually drawn in the Revit based on scanned 3D point cloud data, which can be exported to various formats for other applications such as Unity.
4. Unity's built-in tools add other details, such as model textures, traffic signs, and trees, to make the Digital Twin model more realistic. Figure 2 shows the final static intersection digital twin model imported to Unity, containing the roadways, road markings, signal lights, trees, and buildings.

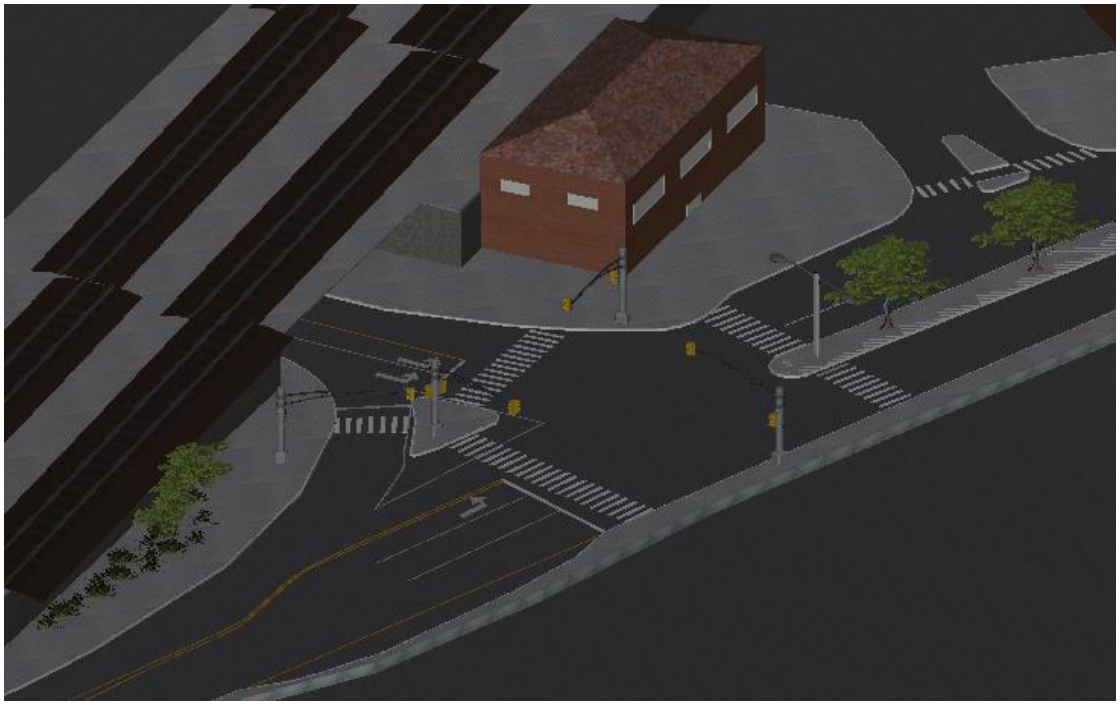

Figure 15. Final Digital Twin Model in Unity

The Unity Asset Store provides an expansive library of high-quality content that you can use to quickly and easily build complex virtual environments for any simulation. Our digital twin model is easy to extend and modify with new elements.

Additionally, Unity's machine learning initiative ML-Agents can be integrated into AirSim's capabilities, allowing even more experimentation. The open source ML-Agents are available through GitHub and have been positively received with well over 4,000 stars. With the release of AirSim on Unity, the two communities now have a common ground to experiment, develop, and evolve.

Vehicle Geometry and Movement Modeling

To visualize the vehicle movement in the simulation, we have developed our real-time procedures to detect vehicles through roadside LiDARs and Cameras. And the vehicle trajectories with longitudinal and latitudinal information will be transformed and fed into Unity to visualize the vehicle movements.

Figure 16 shows a snapshot of vehicles moving in the Digital Twin Model in their specific lanes. Vehicles, traffic lights, road lanes, trees, and light pods are placed according to the collected LiDAR data, highly close to their real positions in the field. The building in the snapshot is the studied New Brunswick Train Station.

Figure 16. Vehicle Movements in Digital Twin Model

All vehicles, sensors, and related coordinate systems are placed in the world coordinate system. A world coordinate system is essential in global path planning, localization, mapping, and driving scenario simulation. The vehicle coordinate system used in Unity is anchored to the ego vehicle. The term ego vehicle refers to the vehicle with sensors that perceive the environment around the vehicle.

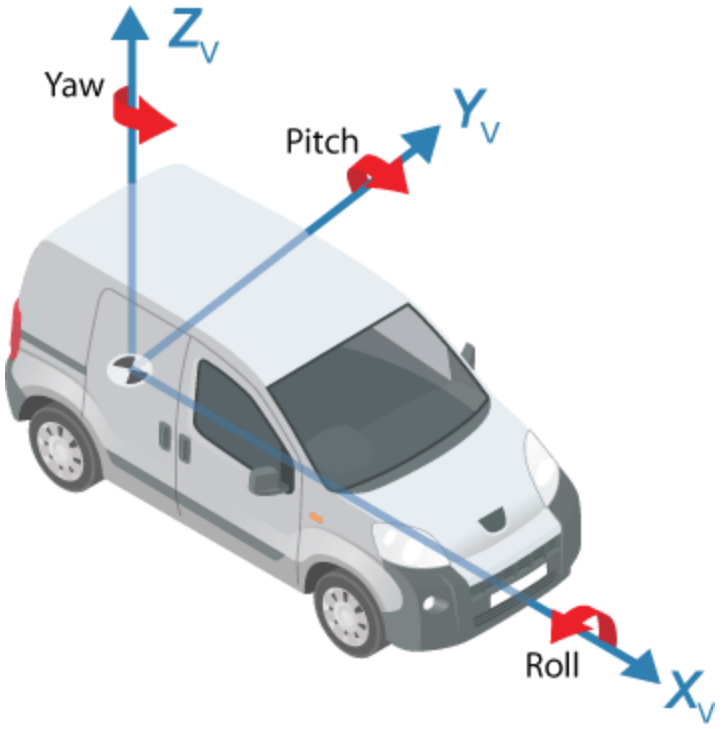

Figure 17. Vehicle Coordinates in Digital Twin Model

As shown in Figure 17, the X_V axis points forward from the vehicle, the Y_V axis points to the left, as viewed when facing forward, and the Z_V axis points up from the ground to maintain the right-handed coordinate system. Each axis is positive in the clockwise direction when looking in the positive direction of that axis. The origin of the vehicle coordinate system is on the ground, below the midpoint of the rear axle. In 3D driving scenario simulations as shown in Figure 18, the origin is on the ground, below the longitudinal and lateral center of the vehicle. Locations in the vehicle coordinate system are expressed in world units, typically meters. Values returned by the RSU are transformed into the vehicle coordinate system to be placed in a unified frame of reference. For global path planning, localization, mapping, and driving scenario simulation, the vehicle's state can be described using the vehicle's pose. The steering angle of the vehicle is positive in the counterclockwise direction.

With the coordinate's transformation tools used, Algorithm 1 describes the logic of vehicle movements in the developed Digital Twin Model. The movements of vehicles are determined by the position data processed based on the camera and LiDAR data from the roadside unit. It should be noted that there are still some instant-shift issues when visualizing the vehicle movements, as the output for the vehicle detection is not very stable and smooth.

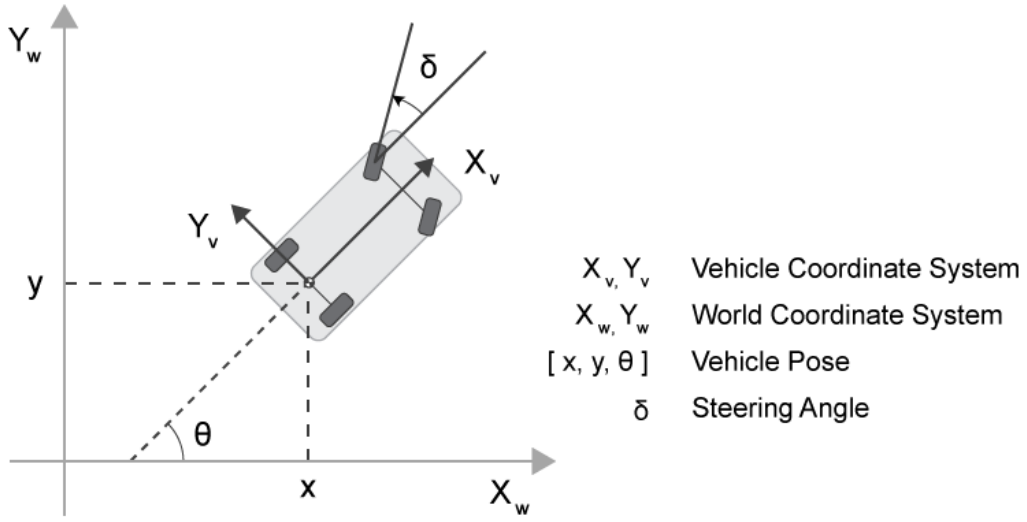

Figure 18. Vehicle Coordinates and Steering Angle in Digital Twin Model

Algorithm 1: Vehicle Movements

Data: vehicle position (x,y) of each frame
Results: vehicle position ($x(t_0)$,$y(t_0)$), facing direction ϑ of current frame
FOR each frame t_0 **DO**
  FOR each vehicle i **DO**
  Transform position from world coordinates ($x(t_0)$,$y(t_0)$) to vehicle coordinates ($x_v(t_0)$,$y_v(t_0)$)
  Update position for current frame t_0 and next frame t_0+1
  Calculate vehicle facing direction ϑ facing to position at next frame ($x_v(t_0+1)$,$y_v(t_0+1)$)
  Transform position back to world coordinates ($x(t_0+1)$,$y(t_0+1)$)
  ENDFOR
ENDFOR

SIMT Prototype Application Demonstration: Intersection Nearmiss Analysis

With rare traffic crashes, near-misses have been commonly used as surrogates for traffic conflicts. Timely detection of near-misses can provide advanced warnings onboard to drivers using smartphone apps or onboard devices. At the same time, statistical data on near-misses can be used to designate safe corridors and help planners mitigate accident risks.

The data from infrastructure instruments and onboard devices can detect near-misses and alert drivers before accidents happen. Near-miss data and alerts can also be communicated to bicycles and pedestrians through personal devices or public billboards. For the proposed hub, near-miss data can be collected from infrastructure cameras, onboard devices, and smartphones. For detection and prevention purposes, near-miss risks must be calculated in short intervals in real-time (e.g., in milliseconds or seconds). Hence, processing images at intersections and roadway segments would require extensive edge computing capacity. Recent works published by one of the Co-PIs developed a statistical risk model to alter drivers on very short time scales. These works use onboard data from the

SHARP-II Naturalistic Driver Behavior database. Four data types are used: event data, time series data, driver behavioral data, and vehicle attributes. No image data was used. These works provide coloring schemes (e.g., red, yellow, and green) for alerting drivers of potential risks. They also suggest schemes with higher resolutions but emphasize that more research is needed to analyze drivers' reactions to these real-time alert schemes.

Further improvements of near-miss risk models would require a close capturing of interactions between vehicles in real-time (using cameras, LIDAR, etc.), real-time driving data (e.g., speed, acceleration, distance from adjacent cars), real-time roadway conditions, driver behavior, and vehicle attributes, and traffic controls (signalized roads vs. non-signalized; 4-way stops, etc.). The hub's instruments will capture these data types with vehicles' digital twins. Using this data and further extending the idea of a vehicle digital twin, we plan to build 2D and 3D digital frames that enclave a vehicle's digital twin and dynamically change in shape (length and width) according to speed, traffic, and external conditions (including weather), see Figure 19. The limits set by the digital frame can be communicated to drivers through onboard devices or smartphones. These digital frames can be shared among vehicles in the case of smart autonomous or connected systems. Near-miss risk can be integrated into vehicles' digital twins as a dynamically changing key safety measure. Communicating and preventing safety risks in mixed traffic (different vehicle intelligence) environments remains a significant challenge we plan to address with the proposed digital twin technology. Autonomous vehicles are trained in isolation, i.e., all the reactive and proactive responses originated from the trained vehicle. We envision that in the future, using these safety frames, collaborative training will be possible. The data from this testbed will enable us to examine some of the earlier theories we developed in game-theoretic distributed control systems.

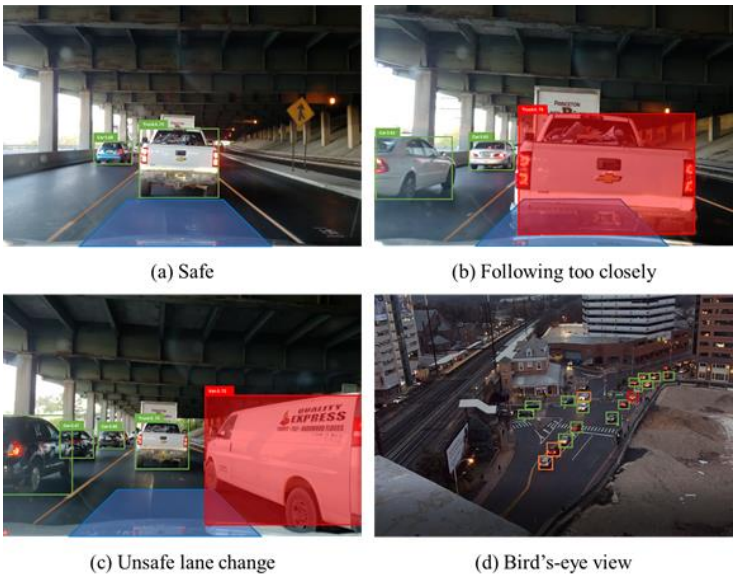

Figure 19. Sample Near-misses

Aggregating near-miss data in short time scales (e.g., hours) provides valuable traffic safety data to law enforcement and traffic controls. Hot spot designation of roadways using systematic and systemic techniques with historical crash data and qualitative measures is not new. Traffic authorities and law enforcement have practiced it for many years. But using real-time individual trip data to build dynamic road safety heat maps is novel and new. The Co-PIs have already developed a preliminary risk heat mapping model of roadway sections, where a roadway section is divided into safety-risk uniform segments [9]. Risk attributes for each such segment are calculated from aggregating drivers' risk profiles that use the segment over time. Heat mapping is sensitive to the time of the day, weather conditions, traffic conditions, and the mix of drivers and vehicles.

See Figure 20 for an example. A 700-feet long roadway subsection is divided into three segments: Link 1, Link 2, and Link 3. Driving behaviors are taken into consideration. According to the study done by CDM Smith, we define three types of drivers based on driving speed, acceleration, following distance, and so on. Type 1 drivers are cautious while driving, while type 3 drivers are aggressive. The driving behavior of type 2 drivers is between type 1 and type 3 drivers.

Figure 20. Sample roadway segments

Intersection Near-miss Applications

A near-miss has the potential to cause but does not result in a traffic accident. Studying near-miss events will help find the causal factors of future accidents and prevent them. The Intersection Near-miss Application is developed based on a data-driven methodology named Safe Route Mapping (SRM), as shown in Figure 21. SRM can combine multiple data sources, predict the conflict risk, and produce heat maps in real-time. These heat maps can be used by authorities to designate safe corridors, dispatch law enforcement, and strategize safety projects.

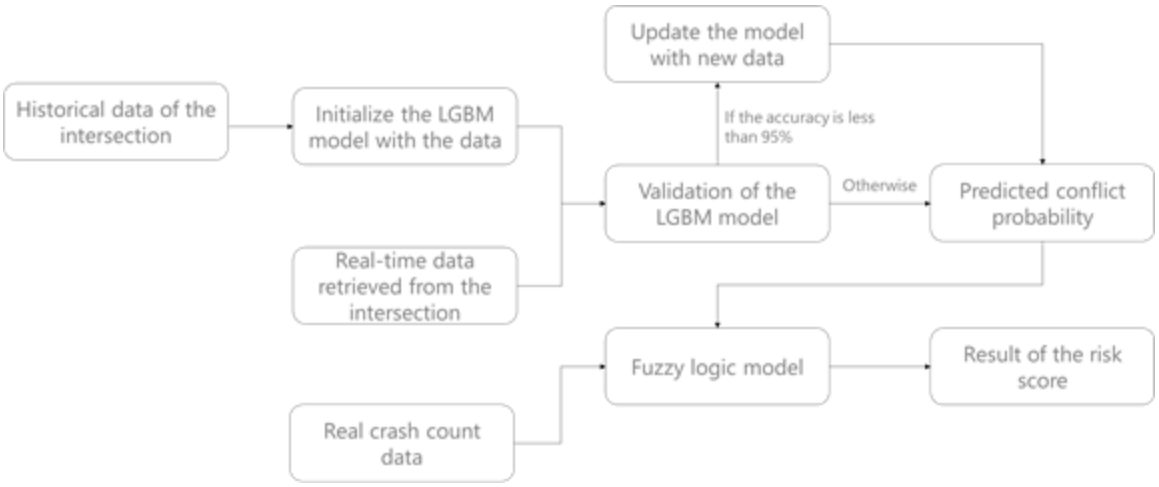

Figure 21. Flow chart of the SRM

Firstly, the data required by the methodology can be collected from the roadside camera. The scanline-based algorithm helps recognize the vehicles at the intersection and calculate each vehicle's GPS information, speed and acceleration, and the inter-relation between them, as shown in Figure 22.

Figure 22. The scanline-based image recognition algorithm

The next step is to determine the traffic conflict indicators, including Time to Collision (TTC), modified Time to Collision (MTTC), and Minimum Deceleration to Avoid Collision (DRAC). These indicators are often used for crash estimation. However, we are not just interested in the real-time estimation of the probability that a crash would occur but also in predicting the conflict probability in the next few seconds. Light Gradient Boosting Machine Classifier (LGBM) is introduced to predict whether the indicators will be held within the safety threshold taking driver behaviors, roadway characteristics, vehicle status, distance, and speed as our inputs, as shown in Figure 23.

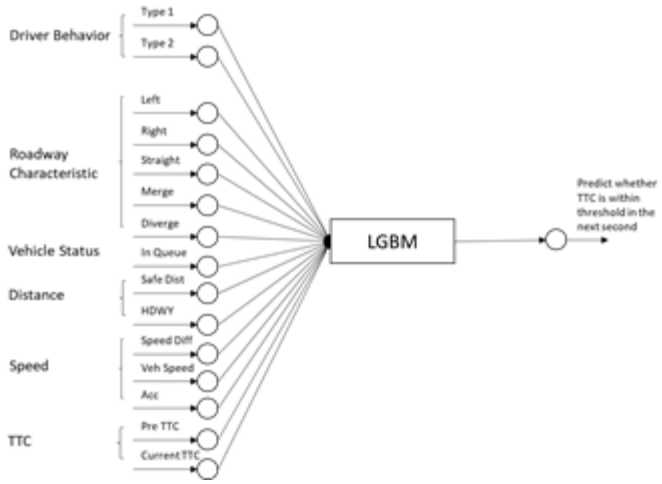

Figure 23. Inputs and outputs of an LGBM model to predict TTC in the next second

Since different indicators would overestimate or underestimate the crash probability under different circumstances, and historical crash data is also a good reference for conflict risk, Fuzzy Logic is applied to combine all the outcomes and generate our final risk score. The inputs and outputs are shown in Figure 24.

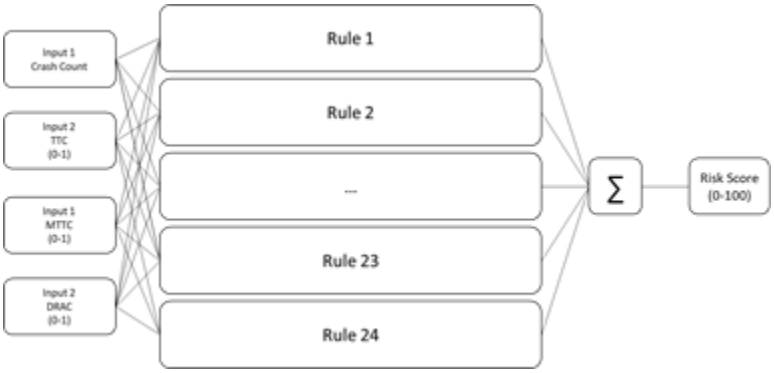

Figure 24. Structure of the fuzzy logic model

SIMT Living Lab Application: Physics-Informed Deep Learning (PIDL)

PIDL was first proposed as an alternative solver of partial differential equations (PDE) in the pioneering work. Since its inception, PIDL has become an increasingly popular data-driven tool for data-driven solutions or discovering nonlinear dynamical systems. Recent years have seen a growing interest in applying PIDL to various engineering areas. There is also a gradually rising effort to encode ordinary differential equations (ODE) into neural networks to approximate ODE solutions [8] or identify dynamical system models.

In transportation, a related study used neural networks of different structures to capture the general behavior of a car-following model and predict the acceleration of a vehicle using velocity and distance information. Yang et al. (2018) combined the Gipps model with DL-based models (back-propagation

neural networks and random forest, respectively) and trained the optimal weights of these two models for collision avoidance of predicted trajectories. Shi et al. (2020) encoded traffic flow models into neural networks for traffic state estimation. However, the application of PIDL to a data-driven solution of car-following dynamical equations or system identification of such behavior is a largely unexploited area.

The generic architecture of a PIDL consists of two deep neural networks: one (i.e., the data-driven component) for predicting the unknown solution, while the other (i.e., the model-driven component), in which physics in the form of PDEs or ODEs are encoded, for evaluating whether the prediction aligns with the given physics. Human driving exhibits highly unstable and nonlinear behaviors, and accordingly, physics-based models may not suffice to reveal the highly nonlinear nature of driving behavior, leading to high bias. Moreover, the physics-encoded computational graph can be treated as a regularization term of the other deep neural network to prevent overfitting, i.e., high-variance.

In summary, the hybrid of both components overcomes high bias and high variance induced by the individual ones, rendering it possible to leverage the advantage of both the model-based and data-driven methods in terms of model accuracy and data efficiency.

FINDINGS

2D Computer Vision and 3D Infrastructure Data Integration Results

Scanline-Detection Validation

The trajectory outputs of the scanline-based vehicle detection algorithm were validated based on trajectory-level and point-level performance metrics. The ground-truth traffic volume data were provided through a commercial video analysis platform. We validate the trajectory-level performance by comparing the ground-truth traffic volume with the proposed scanline-based traffic volume at four cross-sections, as shown in Figure 25(a).

The point-level performance of the scanline-based trajectory model is conducted by a manual video counting tool developed with VLC (VideoLAN Client) media player API. The tool collects the exact video timestamps of vehicles passing some pre-determined scanline points, as shown in Figure 25(b). Two points were pre-defined along each scanline. One is the entry point, representing the point where vehicles are getting on the scanline. The other point is the exit point, representing the point where vehicles get off the scanline. When a vehicle hits the entry point or endpoint along its traveling direction, we click the button of lane number on the VLC interface to record the timestamp of that event. Then we compare the manually collect trajectory points with trajectory points using the proposed method to evaluate the accuracy of our proposed model.

The two-level trajectory detection results are presented in the result analysis section.

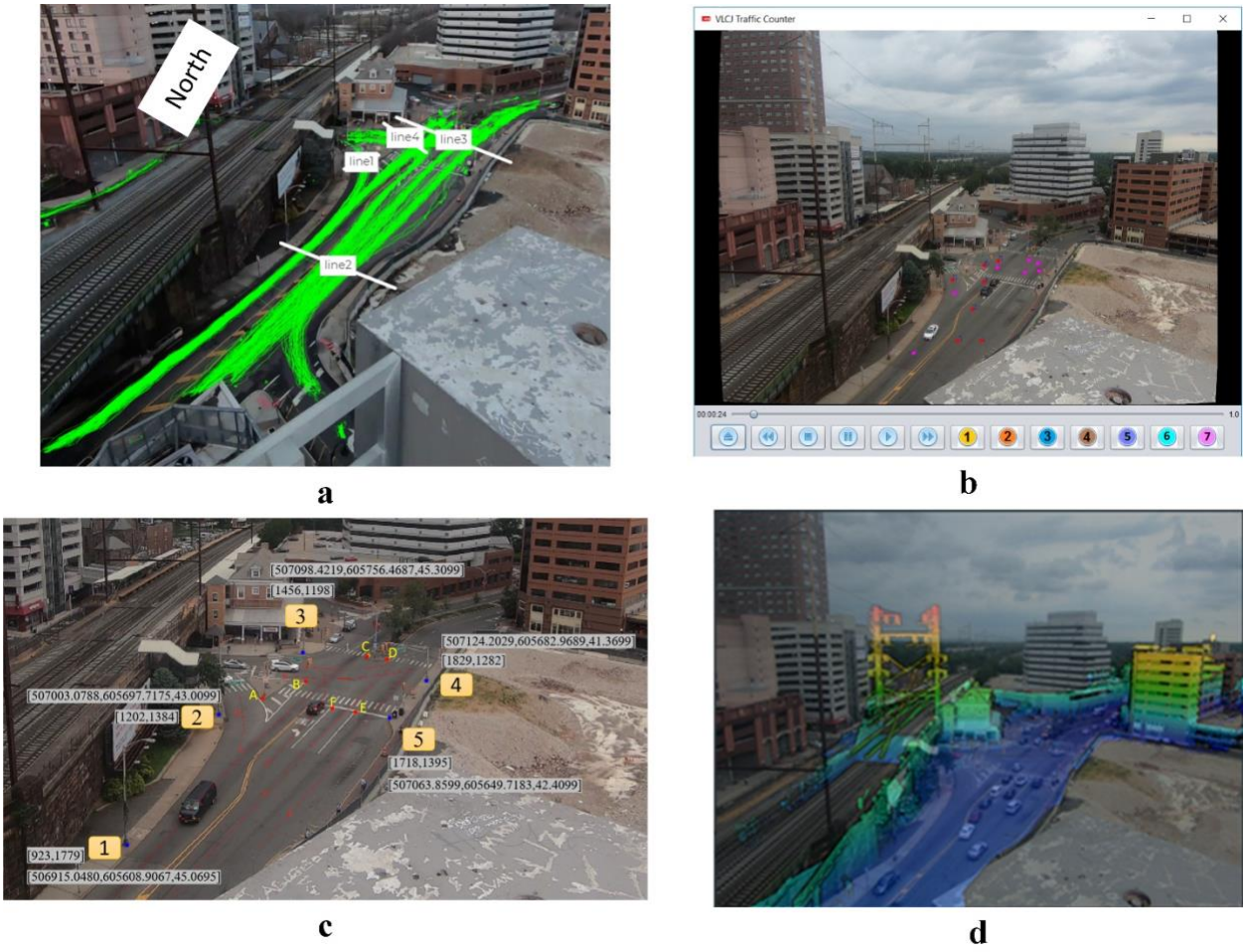

Figure 25. 2D Video Detection and 3D LiDAR Model Validation
(a. Ground-Truth Volume data; b. Video Lane Counter (VLC) for Trajectory Point Validation; c. Reference Points for Camera Calibration in both video Coordinates and GPS coordinates; d. 2D/3D Matching Results)

LiDAR-Camera Projection Validation

We calculated the projective transformation matrix between LiDAR point cloud and CCTV video by picking five key points in the study area (selected locations can be seen in Figure 25(c)). To quantify the performance and accuracy of the 3D/2D matching algorithm, we prepared a validation dataset consisting of six points. Each feature point's pixel coordinates and GPS coordinates in the validation set were recorded. Then we applied the computed 2D/3D projection matrix to transform the 3D coordinates back to 2D pixel coordinates. We used the Mean Squared Error (MSE) to estimate the difference between the values of the projection result and the recorded pixel coordinates. The Mean Squared Error for validation feature points is 1.7025 pixels given 2.7K image resolution, indicating good accuracy.

Table 1 shows the validation data, project errors, and calibrated projection parameters for this LiDAR-camera system.

Table 1. Validate Proposed Calibration Method with Ground-truth GPS and Photo Information

Point No.	Feature Pixel Coordinates	Actual World GPS	Calculated pixel with calibration
A	[1334, 1343]	[507030.1097,605701.4065,43.9999]	[1334.11,1343.65]
B	[1464, 1285]	[507067.4849,605714.5925,43.6899]	[1462.95 1286.45]
C	[1650, 1208]	[507130.5481,605736.7185,42.3899]	[1653.95 1206.58]
D	[1712, 1218]	[507135.3601,605726.0935,41.8599]	[1713.00 1218.45]
E	[1613, 1378]	[507054.9209,605662.6563,42.9499]	[1614.62 1378.71]
F	[1545, 1366]	[507050.2029,605673.6563,43.2799]	[1546.61 1365.96]
Projection Error	MSE = 4.7106 pixel; Average Pixel Discrepancy = 1.7025 pixel		

Calibration Parameters	Tx(m)	Ty(m)	Tz(m)	α(deg)	β(deg)	μ(deg)
	-280.279	149.841	141.0	1.83592	-1.11952	0.934099

Figure 26 illustrates two performance metrics that can be used as intersection performance measurements for operational analysis. Figure 26(A) is the frequency heat map showing detected vehicles' frequencies. The brighter areas indicate the higher detected vehicle frequency, implying a queuing/congestion issue, long waiting time, and limited capacity. As we can see, the waiting time for the westbound lane is the highest, which is consistent with our observation from the Video. This detection frequency map can diagnose traffic congestion to accommodate fluctuating traffic.

Figure 26 is created after calculating the moving speed of each object using the GPS position of each trajectory point. The average speed heat map is a useful performance metric for intersection safety management because many crashes are speed related. Speed profile is critical to optimize signalized intersection based on traffic flow theory, as there is a fundamental relationship between speed, queue, and volume. However, without LiDAR assistance, it is usually too difficult to know the real-world speed characteristic of traveling vehicles from just CCTV cameras.

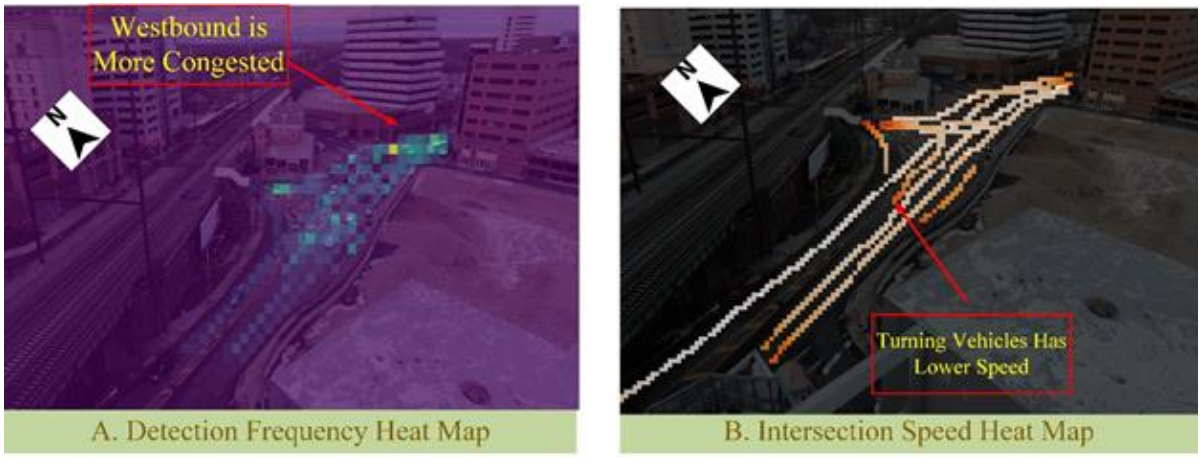

Figure 26. Traffic Analysis Using Vehicle Trajectories Detected from CCTV Camera

Trajectory Detection Accuracy Analysis

This section will discuss the scanline-based vehicle trajectory detection result, present the projected physical trajectory with a 3D LiDAR roadmap, and demonstrate the potential benefits of using LiDAR-assisted video traffic analysis.

Scanline-based Vehicle Trajectory Detection Results in Table 2 show the detected vehicle and ground-truth data at both trajectory-level and point-level. The total volume detection accuracy is 90.87% for all four main approaches. Due to the tilted camera angle, the scanline on one lane might capture vehicles from the adjacent lane. The invasions of adjacent-lane vehicles lead to duplicated counts of vehicle volume. A potential solution to remove duplicated counts is to find the concurrent detections on adjacent lanes. Generally, the scanline method tends to over-counting because it may count vehicles on the other lanes due to occlusions.

The second half of Table 2 shows the point-level trajectory detection results by comparing manually extracted points with the model trajectory. An event is a vehicle hitting either the enter or exit points on the scanline. We can see that the weighted average detection rate is 92%, indicating a good model performance for trajectory detection.

Figure 27 (a), (b), (c), and (d) show the detected trajectories based on travel distance along the scanline from four major directions. Trajectories from each direction are color-coded and demonstrate two signal cycles. The VLC checkpoints are plotted on the trajectory diagrams, demonstrating that our algorithm can provide high-resolution results.

Table 2. Scanline Vehicle Detection Validation Results

Trajectory Level Comparisons				
Direction Number	**Direction**	**Scanline Detection Volume**	**Ground-Truth Data Volume**	**Traffic Count Accuracy**
1	Southbound Right	55	55	100.00%
2	Southbound Left	132	120	90.00%
3	Eastbound Through	119	126	94.44%
4	Westbound Through	148	115	71.30%
	Total Count	454	416	90.87%
Point Level Comparisons				
Lane Number	Direction	Number of Sampled Trajectory Points	Point Level Time Accuracy	
1	Southbound Right Turn	30	86.667%	
2	Southbound Left Turn	94	80.85%	
3	Westbound Through	57	100%	
4	Eastbound Through	66	84.85	
Average			88.09%	

LiDAR-Camera Projection Results

Figure 27 illustrates the physical trajectory projected on the 3D urban infrastructure map using the 3D/2D camera calibration method. Such visualization can give traffic managers a holistic view of traffic flow collected through a 2D infrastructure camera. This picture demonstrates many prominent features using the LiDAR system than other camera calibration models. With the growth of large-scale digital mapping systems, we can acquire more realistic and extensible trajectory data from the proposed system to build traffic flow profiles and promote various traffic studies.

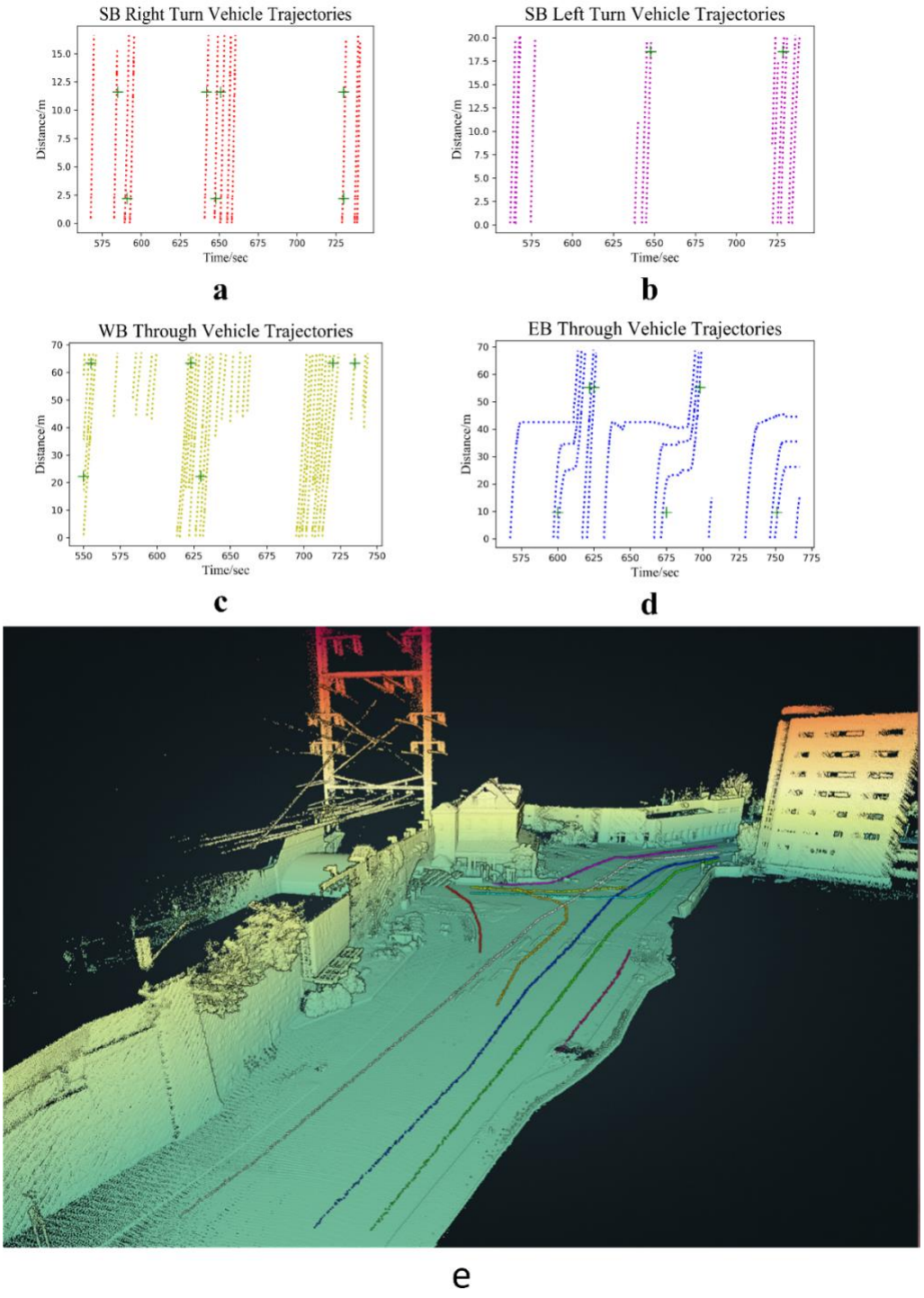

Figure 27. Trajectory Results Investigation
(a. Southbound Right-turn; b. Southbound Left-turn; c. Westbound Through; d. Eastbound Through; e. Projected Physical Trajectory on High-resolution 3D Street Model.) (Note: blue dots on the image a, b, c, d are sampled trajectory points using VLC counter)

In the future, CAV technology such as Eco-intersection Approach and Intelligent Signal Control will lead to more harmonized speed characteristics. The performance metrics generated from trajectory data are critical for CAV-based traffic operation, as they can provide proactive solutions and depict a better picture of the traffic network.

Regarding image processing speed, our scanline-based model shows advantages over the state-of-the-art deep learning model tested on an Intel i7-8750H 2.20 GHz CPU computer. It takes 5.48 seconds on average to process each frame and 47 hours to run a 20-minute video clip using the Mask-RCNN model for vehicle detection. However, using the same computer, the scanline-based model takes roughly 12 minutes to process one STmap from a 20-minute video with competitive accuracy.

Near-miss Analysis Results

Figure 28(a) illustrates the risk profile for a Type 1 driver passing through three differently colored links in L1 traffic. Figure 28(b) shows conflict risks for many Type 1 drivers who pass through the three links. Finally, in Figure 28(c), we have average conflict risks for all sampled drivers. We note that these samples are taken within a given time period on a given day. One can always repeat this sampling for the same period for many days to receive aggregate risk measures over drivers and days. Figures 29 and 30 show how average risk profiles change with driver types and traffic conditions, respectively.

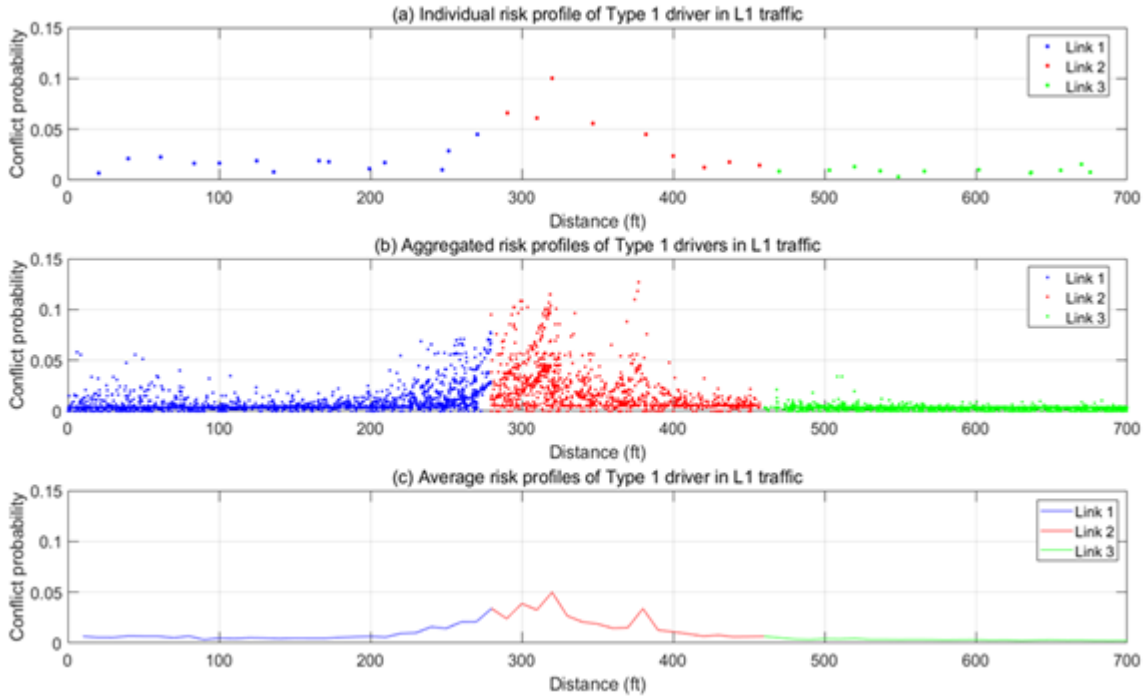

Figure 28. Risk profile of the sample roadway section.

Figure 29. Average risk profiles of different types of drivers in L2 traffic.

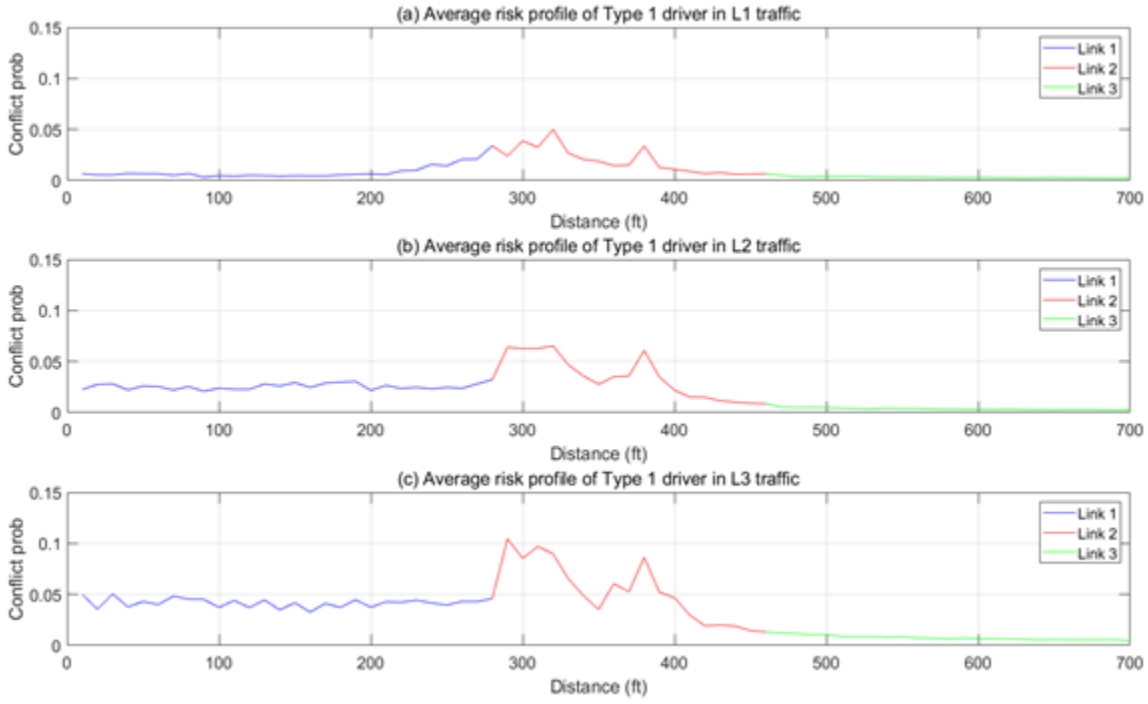

Figure 30. Average risk profiles of Type 1 drivers in different traffic levels.

In all cases, Link 2 shows a higher conflict probability than others. As the traffic level changes, the overall risk profiles fluctuate. Besides, Type 2 driver seems to be more conflict-prone. Next, we fuse crash count estimates with driver-based risk measures to obtain a more comprehensive risk heat map.

These heat maps can provide valuable information to drivers in terms of safe navigation and safe route planning and law enforcement in terms of dispatching resources and configuring traffic controls. We imagine that, in the future, these heat maps will be used by autonomous vehicles for in-vehicular controls (e.g., speed, acceleration, etc.). In this proposal, we will demonstrate, through the simulation of digital twins, the use of these dynamic heat maps for traffic safety and impacts on traffic congestion.

We used three videos filmed at the intersection in New Brunswick to train the LGBM model and validate our SRM algorithm. We took one of the videos as the training dataset, which has 1,425 cars and 17,560 records. The other two videos were used as the testing dataset, one with 2,289 cars and 21,589 records and the other with 1,107 cars and 10,180 records. We defined the records as the information (speed, acceleration, etc.) for each vehicle every 1 second at the intersection.

Figure 31. Road segments

As illustrated in Figure 31, the intersection was divided into 7 segments, and we predicted the risk levels for each every 1 second. The risk levels were defined as: Very small (Risk score is less than 20); Small (Risk score is between 20 and 40); Medium (Risk score is between 40 and 60); Large (Risk score is between 60 and 80); Very Large (Risk score is larger than 80).

The heat map demonstrated in Figure 32 is created to illustrate the difference between the actual risk level and the predicted risk level in the next 1 second and 2 seconds.

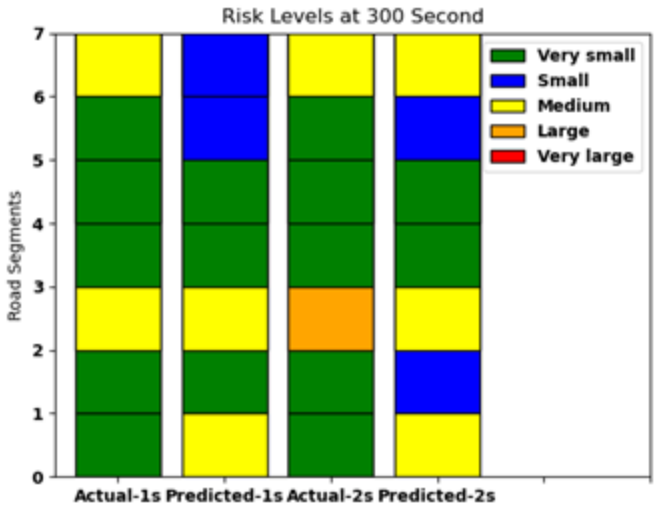

Figure 32. Heatmap of the actual and predicted risk level at 300 seconds in the Video

Assuming our Null hypothesis is that the risk level is very small, small, or medium, the Type I and Type II errors for two testing videos are listed in Table 3 and Table 4.

Table 3. Type I and Type II errors of the predicted risk levels in video 1

N=21589	Accept the Null	Reject the Null
Null hypothesis is true	21260 (98.5%)	113 (0.5%)
Null hypothesis is false	164 (0.8%)	52 (0.2%)

Table 4. Type I and Type II errors of the predicted risk levels in video 2

N=10180	Accept the Null	Reject the Null
Null hypothesis is true	10176 (99.9%)	1 (0%)
Null hypothesis is false	1 (0%)	2 (0%)

The result shows that our algorithm can distinguish between low-risk levels (the risk score is very small, small, or medium) and high-risk levels (the risk score is large or very large) in most situations. The outcome can be used in traffic safety applications to warn the drivers if the conflict risk level on a specific road segment at the intersection is high.

Currently, the application relies on the data collected from roadside devices. The data we can get is limited because we may require more information on driver behavior to understand and predict the conflict risks fully. The next phase of developing the application is to build a hybrid model for nearmiss and warning using data from both infrastructure and driver-based devices such as cell phones.

The Android application we have built in a different project can be useful for us to extend the hybrid model. The App, shown in Figure 33, can collect the trip information and upload it to our database. It can also analyze the safety issues during the trip and inform the drivers of unsafe behaviors.

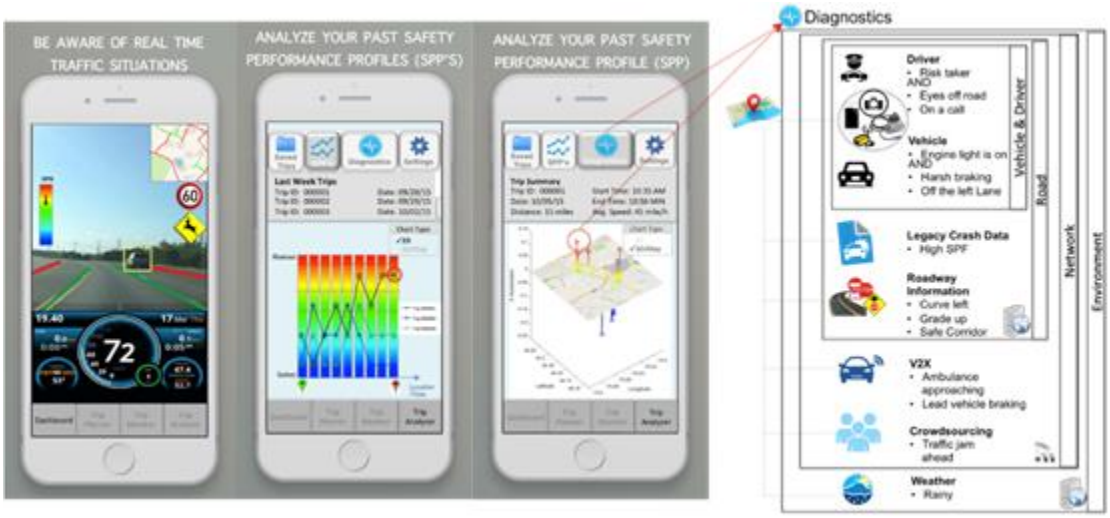

Figure 33. Screenshots of the smartphone application

Figure 34 and Figure 35 reveal the user interface of our database. The information collected from the driver side can supplement the data retrieved from roadside cameras. Combining these data resources with a hybrid model will help us improve the SRM algorithm.

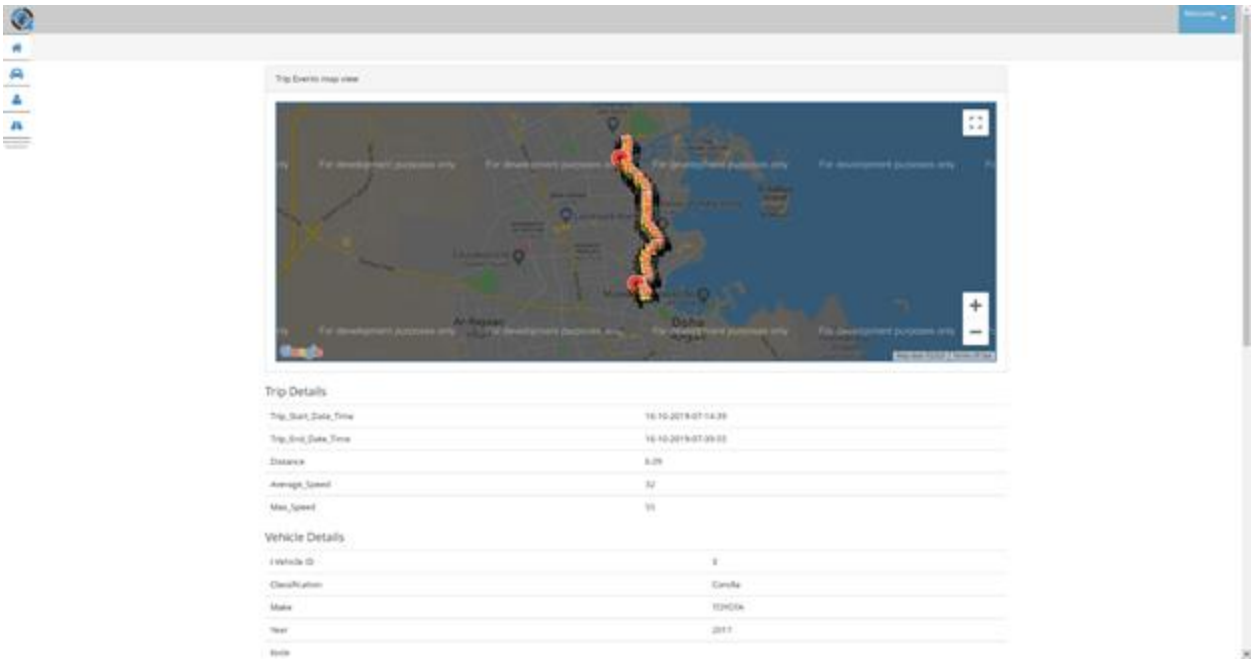

Figure 34. Database for the trip information

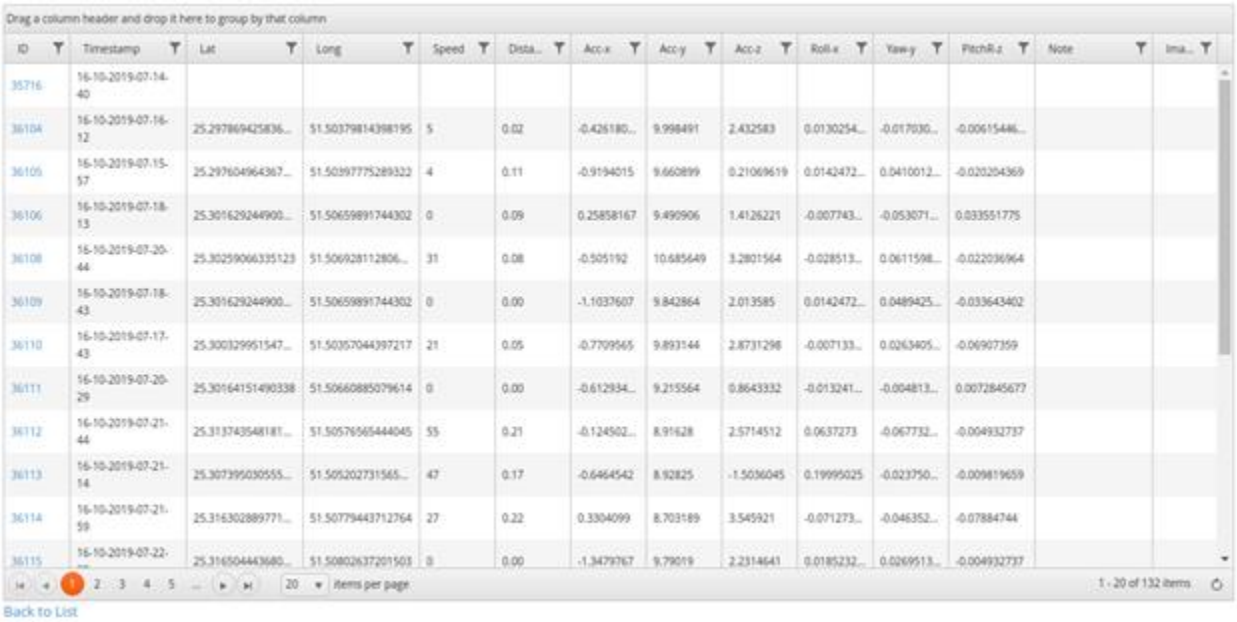

Figure 35. Database for detailed travel records

CONCLUSIONS

The outcome of the proposed research is a small-scale intersection smart mobility testbed. The developed hardware-software-and-cloud systems can be scaled to develop a full-size smart mobility testing ground in the City of New Brunswick.

The project developed the detailed system design and specifications for prototype smart mobility testbed facilities, including the sensing, computing, communication, data, and application testing modules.

This project successfully developed the key components of a prototype smart intersection mobility testbed with components including high-resolution sensing, 3D infrastructure modeling, and CAV application testing.

The research outcome also validated that pilot safety and mobility applications can be implemented and tested with the capabilities of the proposed testbed for application testing, 3D data visualization, and sharing. Furthermore, the team will develop plans and strategies for scaling up the proposed testbed concept toward a full arterial testing corridor.

RECOMMENDATIONS

The research outcome leads to the following recommendations for building arterial smart mobility testbeds. The recommended key components of the system are as follows:

- **Traveler Interface**: The Traveler Interface Layer represents the technologies or means through which travelers in the SMTG and test vehicles would interact with the system. The overall approach for the SMTG does not preclude the use of devices not shown in the graphic; other devices, such as CV On-Board Units (OBU), could be used for application testing.
- **Sensor Infrastructure**: An array of advanced sensors that will collect high-resolution data will form the foundation of the STMG. Sensors will range from autonomous-grade LiDAR to differential GPS base stations to HD surveillance cameras. An initial data capture using LiDAR will be used to capture roadway, transportation facilities, exterior, and in some cases, interior building infrastructure will be collected to serve as a "base map" of the SMTG.
- **Roadside Computing Environment**: The roadside computing layer supports application delivery for test ground users and travelers in the project corridor. Given the low latency required to support connected and advanced driving system applications, data processing will occur near drivers, pedestrians, and bicyclists at the roadside level.
- **Central Data Management Environment**: The Central Data Management Environment layer represents the storage and application servers housed at the Traffic System Lab at Rutgers University for initial processing. Additional processing, as well as cloud-based storage capacity, will be provided by Amazon Web Services (AWS).
- **Public Agencies Operations Support**: The Public Agencies layer addresses applications and resources that will be developed to provide tangible benefits to NJDOT and local jurisdictions that support the SMTG.
- **Living Laboratory**: The Living Laboratory will consist of four individual but related research, development, testing, and transfer (RDT2) labs, which will focus on Technology, Data, Applications, and Knowledge Transfer

REFERENCES

Abadi, M. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.

Amini et al. A Decision-Theoretic Approach to Autonomous Decision Making,

Ardestani, S.M., Jin, P.J. and Feeley, C. Signal Timing Detection Based on Spatial-Temporal Map Generated from CCTV Surveillance Video. Transportation Research Record: Journal of the Transportation Research Board, 2016. 2594: 138-147.

Bhardwaj, R., Tummala, G.K., Ramalingam, G., Ramjee, R. and Sinha, P. Autocalib: automatic traffic camera calibration at scale. ACM Transactions on Sensor Networks (TOSN), 2018 Nov 27. 14(3-4):19.

Cathey, F.W. and Dailey, D.J. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. Intelligent Vehicles Symposium, 2005. p. 777–82.

Chan, T., & Lichti, D. (2015). Automatic in situ calibration of a spinning beam lidar system in static and kinematic modes. Remote Sensing, 7(8), 10480-10500.

Che, E., Jung, J. and Olsen, M.J. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. Sensors, 2019. 19(4):810.

Cho, Y., and Rice, J. Estimating Velocity Fields on a Freeway from Low-Resolution Videos. IEEE Transactions on Intelligent Transportation Systems, 2006. 7(4): 463-469.

Dailey, D.J., Cathey, F.W. and Pumrin, S. An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE transactions on intelligent transportation systems, 2000.1(2):98-107.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. 2005.

Dailey, D.J., Cathey, F.W. and Pumrin, S. An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Transactions on Intelligent Transportation Systems, 2000. 1(2):98–107

Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005.

De la Escalera, A., Armingol, J., Pastor, J. and Rodriguez, F. Visual Sign Information Extraction and Identification by Deformable Models for Intelligent Vehicles. IEEE Transactions on Intelligent Transportation Systems, 2004. 5(2), pp.57-68.

Di Giovanni, M. et al. "Finding Multiple Solutions of ODEs with Neural Networks." AAAI Spring Symposium: MLPS. 2020.

Do, V.H., Nghiem, L.H., Thi, N.P. and Ngoc, N.P. A simple camera calibration method for vehicle velocity estimation. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on 2015. p. 1–5.

Docs.opencv.org. Camera Calibration and 3D Reconstruction — OpenCV 2.4.13.7 documentation, 2019. [online] Available at: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html [Accessed 30 Jul. 2019].

Dubská, M., Herout, A., Juránek, R. and Sochor, J. Fully automatic roadside camera calibration for traffic surveillance. IEEE Transactions on Intelligent Transportation Systems, 2014. 16(3):1162–71.

Engelmann, F., Kontogianni, T., Hermans, A. and Leibe, B. Exploring spatial context for 3d semantic segmentation of point clouds. Proceedings of the IEEE International Conference on Computer Vision, 2017.

Fang, Z., and Justin Z. "A Physics-Informed Neural Network Framework for PDEs on 3D Surfaces: Time Independent Problems." IEEE Access 8 (2019): 26328-26335.

France, J. and VanderPol, J. (2017). LiDAR - A Key Element of DOT's Move to Civil Integrated Management (CIM). [online] Available at: http://www.rieglusa.com/pdf/WHITEPAPER-LIDAR%20AND%20CIM Fremont, V., & Bonnifait, P. (2008, August). Extrinsic calibration between a multi-layer lidar and a camera. In 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (pp. 214-219). IEEE. .final.pdf [Accessed 31 Jul. 2019].

Gordon, R. Traffic control systems handbook. Springfield, Va.: Reproduced by U.S. Dept. of Commerce, National Technical Information Service, 1996.

Grammatikopoulos, L., Karras, G. and Petsa, E. An automatic approach for camera calibration from vanishing points. ISPRS Journal of Photogrammetry and Remote Sensing, 2007. 62(1), pp.64-76.

Guan, H., Li, J., Cao, S. and Yu, Y. Use of mobile LiDAR in road information inventory: A review. International Journal of Image and Data Fusion, 2016. 7(3):219-42.

Guido, G., Saccomanno, F., Vitale, A., Astarita V., and Festa, D. "Comparing safety performance measures obtained from video capture data," Journal of Transportation Engineering, vol. 137, 2011.

He, D., Shi, Y., and Song, X. "Weight-free multi-objective predictive cruise control of autonomous vehicles in integrated perturbation analysis and sequential quadratic programming optimization framework." Journal of Dynamic Systems, Measurement, and Control 141.9 (2019).

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask R-CNN. [online] arXiv.org. Available at: https://arxiv.org/abs/1703.06870 [Accessed 25 Jul. 2019].

Jiang, S., et al. Safe route mapping of roadways using multiple sourced data. IEEE Transactions on Intelligent Transportation Systems, 2020. Jiang, M. Jafari and M. Jalayer, "Safe Route Mapping of Roadways Using Multiple Sourced Data," in Transportation Research Board, Washington, D.C., 2020.

Jiang, S., Jafari, M., and Jalayer, M. "Developing a Two-Dimensional Key Performance Indicator of Safety and Mobility for Intersections," in Transportation Research Board, Washington, D.C., 2020.

Jin, P., Ge, Y., Zhang, T., Wang, Y., and Martinez, J., "Cloud-Based Virtual Traffic Sensor Network with 511 CCTV Traffic Video Streams," in Transportation Research Board, Washington, D.C., 2019.

Klein, L. "Traffic Detector Handbook." Federal Highway Administration Report, Washington, DC, 2006.

Kumar, M., et al. "Genetic algorithm: Review and application." Available at SSRN 3529843 (2010).

Lan, J., Li, J., Hu, G., Ran, B. and Wang, L. Vehicle speed measurement based on gray constraint optical flow algorithm. Optik - International Journal for Light and Electron Optics, 2014.125(1):289 –95.

Li, G., Liu, Y., Dong, L., Cai, X., & Zhou, D. (2007, October). An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3854-3859). IEEE.

Li, J., Yang, B., Chen, C., Huang, R., Dong, Z., & Xiao, W. (2018). Automatic registration of panoramic image sequence and mobile laser scanning data using semantic features. ISPRS journal of photogrammetry and remote sensing, 136, 41-57.

Luvizon, D.C., Nassu, B.T. and Minetto, R. A video-based system for vehicle speed measurement in urban roadways. IEEE Transactions on Intelligent Transportation Systems, 2016. PP(99):1–12.

Ma, L., Li, Y., Li, J., Wang, C., Wang, R. and Chapman, M. Mobile laser scanned point-clouds for road object detection and extraction: A review. Remote Sensing, 2018.10(10):1531.

Malinovskiy, Y., Wu, Y. and Wang, Y. Video-Based Vehicle Detection and Tracking Using Spatiotemporal Maps. Transportation Research Record: Journal of the Transportation Research Board, 2009. 2121: 81-8 Raissi, Maziar. "Deep hidden physics models: Deep learning of nonlinear partial differential equations." The Journal of Machine Learning Research 19.1 (2018): 932-955.

Nageshrao, S., Eric Tseng, H., and Filev, D. "Autonomous highway driving using deep reinforcement learning." 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, 2019.

Naroditsky, O., Patterson, A., & Daniilidis, K. (2011, May). Automatic alignment of a camera with a line scan lidar system. In 2011 IEEE International Conference on Robotics and Automation (pp. 3429-3434). IEEE.

Ossen, S., and Hoogendoorn, S.P. "Car-following behavior analysis from microscopic trajectory data." Transportation Research Record 1934.1 (2005): 13-21.

Pandey, G., McBride, J. R., Savarese, S., & Eustice, R. M. (2015). Automatic extrinsic calibration of vision and lidar by maximizing mutual information. Journal of Field Robotics, 32(5), 696-722.

Process Improvement Institute, Inc., "Gains from Getting Near Misses Reported," in 8th Global Congress on Process Safety, Houston, 2012.

Raissi, Maziar, and George Em Karniadakis. "Hidden physics models: Machine learning of nonlinear partial differential equations." Journal of Computational Physics 357 (2018): 125-141.

Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational Physics 378 (2019): 686-707.

Raissi, Maziar, et al. "Deep learning of vortex-induced vibrations." Journal of Fluid Mechanics 861 (2019): 119-137.

Raissi, Maziar, Alireza Yazdani, and George Em Karniadakis. "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations." Science 367.6481 (2020): 1026-1030.

Roehrl, Manuel A., et al. "Modeling System Dynamics with Physics-Informed Neural Networks Based on Lagrangian Mechanics." arXiv preprint arXiv:2005.14617 (2020).

Schoepflin, T.N. and Dailey, D.J. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. IEEE Transactions on Intelligent Transportation Systems, 2003.4(2):90–8.

Shi, R., Mo, Z., Di, X., 2020. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. 35th Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence.

Sina, I., Wibisono, A., Nurhadiyatna, A., Hardjono, B., Jatmiko, W. and Mursanto, P. Vehicle counting and speed measurement using headlight detection. In 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 149-154). IEEE, 2013. p. 149–54.

Sirignano, Justin, and Konstantinos Spiliopoulos. "Scaling Limit of Neural Networks with the Xavier Initialization and Convergence to a Global Minimum." arXiv preprint arXiv:1907.04108 (2019).

Sochor, J., Herout, A. and Havel, J. BoxCars: 3D boxes as CNN input for improved fine grained vehicle recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016a.

Sochor, J., Juránek, R. and Herout, A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. Computer Vision and Image Understanding, 2017. 161, pp.87-98.

Sun, Y., Xu, H., Wu, J., Zheng, J., & Dietrich, K. M. (2018). 3-D data processing to extract vehicle trajectories from roadside LiDAR data. Transportation research record, 2672(45), 14-22.

Treiber, Martin, and Arne Kesting. "Microscopic calibration and validation of car-following models–a systematic approach." Procedia-Social and Behavioral Sciences 80 (2013): 922-939.

Turner, S. Travel time data collection handbook. Washington, DC: Federal Highway Administration, Office of Highway Information Management, 1998.

Veľas, M., Španěl, M., Materna, Z., & Herout, A. (2014). Calibration of rgb camera with velodyne lidar.

VISSIM Development and Calibration Report (CDM Smith), October 2014.

Wang, R., Bach, J. and Ferrie, F.P. Window detection from mobile LiDAR data. IEEE Workshop on Applications, 2011.

Williams, K., Olsen, M., Roe, G. and Glennie, C. Synthesis of transportation applications of mobile LiDAR. Remote Sensing, 2013.5(9):4652-92.

Wu, F., Wen, C., Guo, Y., Wang, J., Yu, Y., Wang, C. and Li, J. Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach. IEEE Transactions on Intelligent Transportation Systems, 2016.18(2):292-305.

Wu, F., and Daniel B. Work. "Connections between classical car following models and artificial neural networks." (2018) 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018.

Wu, J. (2018). An automatic procedure for vehicle tracking with a roadside LiDAR sensor. Institute of Transportation Engineers. ITE Journal, 88(11), 32-37.

Wu, J., Xu, H., Lv, B., Yue, R., & Li, Y. (2019). Automatic ground points identification method for roadside LiDAR data. Transportation research record, 2673(6), 140-152.

Wu, J., Xu, H., Sun, Y., Zheng, J., & Yue, R. (2018). Automatic background filtering method for roadside LiDAR data. Transportation Research Record, 2672(45), 106-114.

Wu, J., Xu, H., Zheng, Y., Zhang, Y., Lv, B., & Tian, Z. (2019). Automatic vehicle classification using roadside LiDAR data. Transportation Research Record, 2673(6), 153-164.

Wu, J., Xu, H., & Liu, W. (2019). Points registration for roadside LiDAR sensors. Transportation research record, 2673(9), 627-639.

Xu, S., Wang, R. and Zheng, H. Road curb extraction from mobile LiDAR point clouds. IEEE Transactions on Geoscience and Remote Sensing, 2016.55(2):996-1009.

Yang, B., Fang, L., Li, Q. and Li, J. Automated extraction of road markings from mobile LiDAR point clouds. Photogrammetric Engineering & Remote Sensing, 2012.78(4):331-8.

Yang, B., Dong, Z., Zhao, G. and Dai, W. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 2015.99:45-57.

Yang, D., et al. "A novel car-following control model combining machine learning and kinematics models for automated vehicles." IEEE Transactions on Intelligent Transportation Systems 20.6 (2018): 1991-2000.

Yang, Y., and Perdikaris, P. "Adversarial uncertainty quantification in physics-informed neural networks." Journal of Computational Physics 394 (2019): 136-152.

Ye, X., Li, J., Huang, H., Du, L. and Zhang, X. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. Proceedings of the European Conference on Computer Vision (ECCV), 2018.

You, X. and Zheng, Y. An accurate and practical calibration method for roadside camera using two vanishing points. Neurocomputing, 2016.

Yu, Y., Guan, H. and Ji, Z. Automated detection of urban road manhole covers using mobile laser scanning data. IEEE Transactions on Intelligent Transportation Systems, 2015.16(6):3258-69.

Zai, D., Li, J., Guo, Y., Cheng, M., Lin, Y., Luo, H. and Wang, C. 3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts. IEEE Transactions on Intelligent Transportation Systems, 2017.19(3):802-13.

Zhang, Q., & Pless, R. (2004, September). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566) (Vol. 3, pp. 2301-2306). IEEE.

Zheng L., and Sayed T. "Comparison of Traffic Conflict Indicators for Crash Estimation using Peak Over Threshold Approach," Transportation Research Record, 2019.