Continuous Emulation and Multiscale Visualization of Traffic Flow Using Stationary Roadside Sensor Data

Haowen Xu, Anne Berres, *Member, IEEE*, Sarah A. Tennille, Srinath K. Ravulaparthy, Chieh (Ross) Wang, *Member, IEEE*, and Jibonananda Sanyal, *Member, IEEE*

***Abstract*—With the advent of the next-generation traffic monitoring systems, there has been a significant increase in the spatial-temporal resolution of vehicle mobility data in many cities. Effective analysis and visualization of such data can provide transportation planners with data-driven insights, which can facilitate the understanding of multiscale traffic dynamics. In this paper, we present a web-based traffic emulator for emulating and visualizing near-real-time and historical traffic flows on highways using data from road-side sensors. To construct a continuous traffic flow, the emulator adopts an analytical pipeline that can (a) integrate traffic data collected from discrete road-side radar detection sensors, (b) interpolate traffic conditions (vehicle speed and volume) on unmeasured road segments based on traffic flow theory, and (c) generate lane-specific vehicle trajectories and movements using a mathematically optimized representation of the road network. Our app also provides an integrated visual workflow that allows users to explore the interconnected traffic dynamics using an appropriate traffic flow visualization selected based on the level of detail. We devise two innovative geo-visualization techniques that utilize an animated strips-network representation and a lane usage matrix to visualize lane performances. To ensure a smooth emulation of large-scale traffic flow in an easy-to-access web environment, we implement the emulator using client-side GPU-accelerated techniques. Finally, we close with a case study that visualizes traffic dynamics of two scenarios – an afternoon peak hour and a traffic accident – in Chattanooga, Tennessee. Our app visualizes the responses of traffic dynamics during different traffic conditions, and to the presence of the traffic accident at different spatial scales.**

***Index Terms*—Traffic flow visualization, level of detail, situational awareness, traffic sensor network, urban mobility, traffic monitoring**

I. Introduction

RAPID urbanization of cities around the globe has resulted in the large-scale proliferation of motor vehicle use [1, 2], leading to severe traffic congestion problems that bring about considerable economic loss in time and fuel, increase in travel times, severe air pollution, and urban heat island effect [3, 4]. Recent research efforts have been focusing on *smart city* solutions to alleviate traffic congestion through information technologies [5] that rely on networks of road-side sensors (e.g., inductive loop, video camera, microwave radar, and infrared detector) to achieve the real-time situational awareness of traffic conditions in urban areas. Ideally, these technologies are designed to help transportation planners better understand the underlying mechanism of traffic congestion and lane performances, with the intent of supporting informed decisions for maintaining transportation infrastructure and optimizing traffic operations [6]. However, there are some limitations in the traditional monitoring systems: (1) traffic data collected from road-side sensors are limited to discrete points along major roadways [7]; (2) trajectories that are collected from on-board sensors (e.g., Global Navigation Satellite Systems) can record continuous vehicle movement but are only available from a small portion of vehicles [8]; and (3) most mobility datasets measured through on-board sensors are managed by commercial entities and are not freely available for transportation practitioners and researchers. Subsequently, it is challenging for traditional traffic monitoring systems to produce practical insights for more in-depth mobility management, as most of these systems are not designed to capture large-scale vehicle movements as continuous flows, as well as to present lane usage. Additionally, due to their limited availability and commercial nature, these systems can not be easily extended to monitor traffic in other cities. Replicating a realistic traffic flow in a shareable web environment enables the situational awareness of transportation systems, which is critical for many transportation system management applications, such as identifying traffic bottlenecks, monitoring lane performance [9], and optimizing traffic controls [10, 11]. Due to the complex nature of traffic flow [12], smaller-scale traffic dynamics (e.g., individual vehicle driving behavior and lane design) may lead to traffic incidents and congestion that may affect the mobility pattern at the regional scale. Nevertheless, comprehensive applications that can realistically mirror and visualize these interconnected traffic dynamics from multiple spatial scales are still rare in the mobility management sector.

In the interest of exploring the interconnected dynamics behind traffic congestion from a holistic perspective, we present a web-based application (web app) to emulate real-world traffic flows using near-real-time and historic traffic data collected

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

H. Xu, A. Berres, S. A. Tennille, C. Wang, and J. Sanyal are with the Oak Ridge National Laboratory Oak Ridge, TN 37831 USA (email: xuh4@ornl.gov; berresas@ornl.gov; bleakneysa@ornl.gov; cwang@ornl.gov; sanyalj@ornl.gov).

S. K. Ravulaparthy is with the Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, CA 94720 (email: srinath@lbl.gov).

Original manuscript received October 26, 2020.

from stationary radar detectors. Moreover, we demonstrate visualizations of the emulated traffic flow from microscopic to macroscopic scale through a visual workflow (depicted in Fig. 1). This web app is an integral part of the CTwin web platform [13], a *digital twin* for transportation at regional scale for the Chattanooga Metropolitan Area, which serves as a geospatial virtual representation of traffic in the area.

This work is divided into three research steps, which are reflected in Fig. 1. Our major contributions associated with each research step are listed as the following:

- *Data Preparation and Integration*: We developed an automated pipeline for integrating traffic data from discrete road-side sensors and reconstructed a continuous traffic flow that captures traffic dynamics at multiple spatial scales. We also optimized the road network representation using composite Bézier curves, an parametric curve-based representation, to optimize the computational performance of the emulation and visualization of traffic flow in a shareable web environment.
- *Traffic Flow Routing and Emulation*: We employed an algorithm that can interpolate the traffic state on unmeasured links using basic principles from the traffic flow theory. We refer to this algorithm as *traffic flow routing*. Based on the interpolated traffic state, we developed a *traffic flow emulation* algorithm that utilizes the composite Bézier curves representation of the road network to calculate large-scale vehicle movements in near-real-time.
- *Multiscale Flow Visualization*: We designed an exploratory visual workflow to visualize the emulated traffic flow and its pertaining traffic dynamics. This workflow selects the appropriate traffic flow visualization based on the Level of Detail (LoD) technique, which can help users explore the complex traffic dynamics and their interconnection at different spatial scales. Within the workflow, two novel geo-visualizations, which include an animated strips-network representation and a lane-usage matrix, were devised to display lane-level traffic dynamics and performance.

II. RELATED WORK

Traffic flow emulation is a comprehensive endeavor that requires the integration of interdisciplinary subjects: processing and understanding traffic data, traffic state interpolation, and traffic flow visualization. Based on these effort types, we divide this section into the following subsections.

A. Traffic Data

There are two main categories of traffic sensors that are widely used to monitor traffic: on-road sensors and on-board sensors [14]. On-road sensors are integrated with road infrastructure and can detect and classify vehicles in nearby areas [15]. Since on-road sensors are often deployed next to the road, these sensors are also known as ”road-side” sensors. Examples of these sensors include inductive loops, video-detectors, magnetometers, radar detectors, etc. [16]. On-road sensors provide the conventional method for monitoring traffic. Their main advantage is their technological maturity and reliability, as they have been continuously improved by government agencies and automobile industries for many decades [14]. Although on-road sensors are widely used, they are inflexible and can only monitor traffic at fixed positions along the road [17, 18]. Thus, they can only produce observations at discrete locations and cannot capture the traffic flow continuously along a road.

On-board sensors, also known as “probes”, have been more recently developed by automotive manufacturers and can, in theory, provide detailed location tracking and produce a continuous representation of vehicle trajectories. These sensors include GPS, mobile devices, and wireless beacons [19, 20]. Although these on-board sensors can be used to compensate for the limitation faced by on-road sensors [21], they are not available in every vehicle. Additionally, these sensors lack commonly accepted standards among different brands, and the continuity of their measurements are limited to the quality of wireless/GPS signals [14]. In addition, the majority of traffic data collected from on-board sensors is managed by commercial entities, thus are not freely available to transportation agencies and researchers. On-board sensors also raise concerns about encroachments on individual privacy as third parties access vehicle-tracking information.

B. Traffic Flow Emulation

In this work, we define traffic flow emulation as the endeavor to physically replicate real-world vehicle movement in a digital environment using measurements from traffic sensor, which is different from the concept of traffic model-based simulation. To the best of our knowledge, there are no existing applications that emulate the traffic flow at different spatial scales using data collected from road-side traffic sensors. The most relevant application would be the driving Intelligence system developed by INRIX [22], and the models that are developed based on the framework of space-time prisms [23] to evaluate the uncertainty of a moving vehicles location in between sample points that are measured by vehicle tracking devices (e.g., GPS and mobile apps). Literature on constructing continuous traffic flow using discrete roadside sensors is still scarce. Currently, traffic modeling and simulation applications are widely used to simulate the traffic flow based on hypothetical scenarios and ad hoc street measurements [24]. Examples of these applications include VISSIM [25], and SUMO [26]. However, many of these models are computationally intensive and are difficult to execute in near real-time using dynamic traffic sensor data. Additionally, most of these simulation models are implemented as desktop applications, limiting the capability for disseminating results and informing transportation practitioners about the most updated traffic states. Web-based applications that conduct traffic flow routing and emulation to mirror authentic traffic dynamics in digital twins are still rare.

Previous efforts have been made to characterize traffic flows using the traffic conservation law [27, 28], which describes macroscopic changes in both the time and location of vehicles along a road. In an enclosed highway system, this principle can

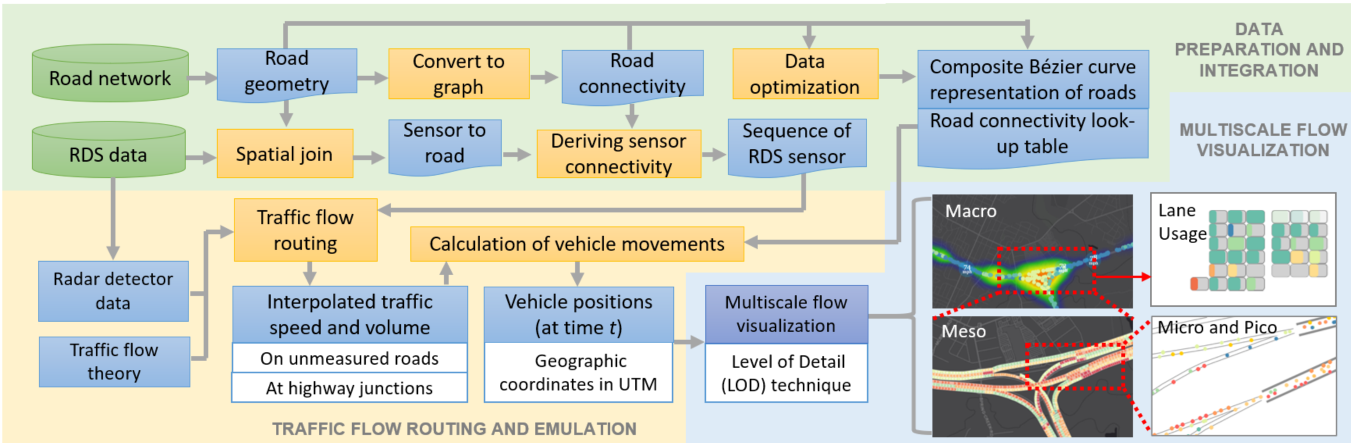

Fig. 1: The overall workflow of our system: First we process and integrate the data to create a piecewise continuous representation of the traffic state along the highways (on measured roads) by incorporating the road-side RDS sensor data. Next, we emulate and route traffic flow following traffic flow theory paradigms and interpolate traffic state between sensor locations (on unmeasured roads). Finally, we provide visualizations at different scales from macroscopic to microscopic scale to provide a zoom-appropriate level of detail.

be applied to estimate lane-specific vehicle count and speed on unmeasured roads using the traffic observation from adjacent roads. The continuity equation [29] is another important concept in the traffic flow theory, which emphasizes that no vehicles are lost or created in a road network. When applied to the traffic observations collected from the surrounding roads of a highway junction, the continuity equation is especially useful for determining the number of vehicles entering/exiting the highway through on/off-ramps.

C. Traffic Flow Visualization

Most traffic flow visualizations rely on vehicle trajectory data. These visualizations can be divided into three categories based on their spatial scales: macroscopic, mesoscopic, and microscopic.

Macroscopic scale visualizations aim to provide a regional overview of traffic patterns, such as flow density and hotspots [30], summarized trajectories through spatial or spatial-temporal aggregation [31, 32], and origins and destinations [33, 34]. Over the past decade, a wide variety of spatial visualization techniques have been designed to address these needs, including heatmaps [35, 36], edge splatting [37], density-based clusters and particles [38, 39], glyphs [40, 41], flow maps [42, 43], and amoeba representations [44].

Flow visualizations at the *mesoscopic* scale serve as the middle ground between the microscopic and macroscopic scales. At this scale, the traffic flow community focuses on analyzing a small group of transportation elements and their mobility patterns that are normally appearing at the corridor level [45]. Current interest in mesoscopic scale traffic flow visualization is centered on presenting the speed-density relationship in a traffic corridor. Subsequently, combined visualizations of microscopic and macroscopic scales are developed to achieve this goal, such as the integration of (1) density map and particle representations [46]; (2) heatmap and 3D particle movements [47]; and (3) variants of glyphs [40].

At the *microscopic* scales, traffic visualizations primarily focus on depicting the accurate position of individual vehicle movements, as well as revealing interactions between traffic entities (e.g., vehicle-vehicle, and vehicle-incident). Subsequently, animated trails and particles representation [48, 49] are widely used to present detailed vehicle trajectories and discrete choice (e.g. lane-changing and gap-acceptance) in the lateral and longitudinal directions [12].

In general, traffic flow and mobility visualization is a well-established field except for a few knowledge gaps:

- Studies have focused on visualizing spatial-temporal patterns at the regional and corridor level [36], while visualization techniques for exploring lane usage and performance in near-real-time are still sparse. Traditional Space-time diagrams are static and have limited capability for visualizing vehicle movements in different lanes.
- Most traffic flow visualizations are designed to visualize vehicle trajectory data that is collected directly from on-board sensors. In order to emulate traffic flows derived from road-side sensors, an optimized geo-processing method is needed to generate lane-specific trajectory for individual vehicles using the center line geometry of a generic road network.
- Many existing visualizations only address traffic dynamics and patterns at a specific spatial scale. A visual workflow for depicting the interactions and evolution of multiscale dynamics is needed.

III. OVERVIEW

In this section, we introduce the input data used in our traffic emulator and present the three major components of the present system design.

A. Description of Input Data

In this paper, we present the emulation and visualization of traffic flow on highways in Chattanooga, Tennessee, as a case study. There are two types of input data that are extensively used in this work, namely the radar detector data and digital road network. In addition, we use Hamilton County's 911 data to identify traffic incidents in the system.

1) Radar Detector Data: Radar Detection System (RDS) sensors transmit low-energy microwave radiation that is reflected by all vehicles within the detection zone. Currently, there are 214 radar sensors deployed at about half a mile intervals along I-24, I-75, US-27, and SR-153 in the Chattanooga area. These road-side sensors are operated by the Tennessee Department of Transportation (TDOT) and can provide lane-specific traffic data (e.g., vehicle counts, average speeds, and occupancy) at 30-second intervals. In the absence of RDS data, the presented traffic emulation methodology can use other types of road-side sensors as the input, as long as they provide lane-level speeds and volumes at regular time intervals.

2) Road Network: This work relies on a generic digital road network provided by the Chattanooga Department of Transportation (CDOT), containing detailed geometric representation and connectivity of all the roads in Chattanooga. The dataset also describes the number of lanes for each link, which is later used to generate lane-level trajectories for the traffic emulation. In this work, we studied traffic flow in the Chattanooga Metropolitan Area, which consists of approximately 95 center-line miles of highways.

3) 911 Data: In order to include the incidents that affect highway traffic in this area, we use Hamilton county's 911 data and filter it down to those incidents that are related to highway traffic. For each incident, we have information about date, time, location (e.g., street address and geocoordinates), incident type, and response unit types (e.g., law, fire, medical).

IV. DATA PREPARATION AND INTEGRATION

We present the overall data processing pipeline in Fig. 2 and describe individual data processing tasks in the the following subsections.

A. Radar Detection Sensor Measurements

We collected RDS data from TDOT and accumulated them into a relational database system (PostgreSQL), which we access through web services. Before ingesting the data into the traffic emulation app, data preprocessing is needed to fulfill three major data requirements:

- RDS observations need to be spatially joined to the links they measure.
- The sequence of RDS sensors needs to be established based on road topology, which enables the querying of observations from adjacent sensors. This is essential for interpolating continuous traffic flow along the highway.
- Data cleaning is required to remove RDS records with (1) unrealistic vehicle speed (e.g., 300 mph); (2) duplicate timestamps; and (3) ambiguous mapping of sensor locations to links.

In this work, we generated a spatial join table using ArcToolbox, which created a direct mapping between the RDS sensor ID and its pertaining road ID. This step is a one-time data pre-processing task that we conducted manually. The sensor topology graph contains the sequence of RDS sensors. The topology was created through a customized NodeJS geo-processing script with the following steps: (a) import the digital road network and convert it into a graph in which adjacent links are represented as connected edges; (2) map the sensor ID to the connected edges in the newly created graph; and (3) derive the sequence of RDS sensors through the road topology represented in the graph. As the traffic flow direction is encoded in each link (from the original road network), we can easily identify the adjacent sensors of a specific link by searching the sensor ID that is mapped to its connecting edges. The output of the sensor-to-road spatial join and the sensor topology are represented through the key-value pair data structure and are stored in static JavaScript Object Notation (JSON) files.

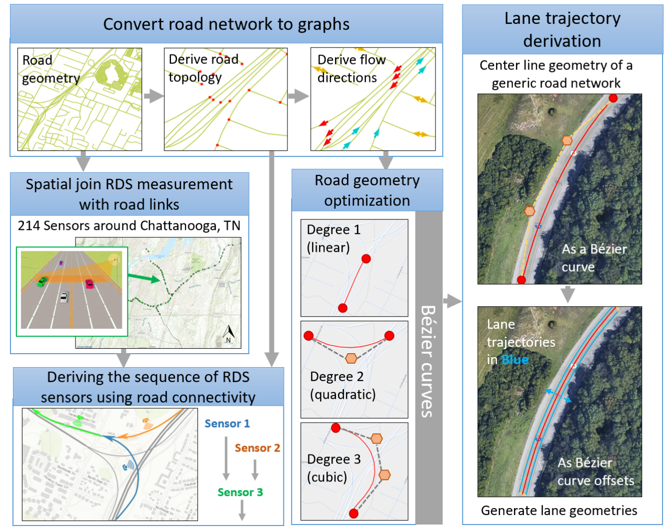

Fig. 2: Processing of the the road network and sensor data. The road network is processed to determine topology and flow directions and fit Bézier splines to the links. We then spatially join sensor locations to their nearby links, generating the sequence of sensors used to determine the topology of the sensor network.

B. Road Network Optimization

The digital road network used in this work is a GeoJSON file acquired from CDOT that contains the center lines of $36,895$ links in Chattanooga. From a traffic emulation perspective, there are several data requirements for the road network, including (1) reliable road connectivity for interpolating traffic states at ungauged roads; (2) spatial information that allows the generation of lane-level trajectories for individual vehicles in an automated fashion; and (3) a road geometry that enables smooth interpolation of vehicles positions on the road at any time frame. To fulfill these requirements, we fit composite Bézier curves to obtain a smooth representation of the road network [42]. Individual Bézier curves within a spline can be defined for any degree n to effectively represent different geometric components in a link. As shown in Fig. 2, linear Bézier curves ($n = 1$) are the best fit for modeling linear links;

quadratic Bézier curves ($n = 2$) are good fits for representing simple curves and circle arcs at road turns; and and higher-order Bézier curves ($n > 3$) can be used to represent more complex links that have a meandering shape. There are three major advantages that are associated with modeling roads using composite Bézier curves:

- These curves can be efficiently stored using 3-4 control points, which takes up less storage space than traditional polylines that consist of 10-20 vertices.
- The coordinates of any point (or vehicle) on the curve, as well as their direction of movement can be readily calculated using parametric equations, enabling the animation of a smooth movement along the curve.
- Trajectories of individual lanes can be easily derived mathematically as parallel curves along the roads center line.

Since our traffic flow emulation is designed as a client-side web app in the user's browser, the adoption of Bézier curve representation can effectively optimize the performance of the emulation and visualization processes in a web environment, where the memory and computation power are limited. In this setting, we could simultaneously emulate and render millions of vehicle movements at the regional level.

In this work, we applied a least square-based parametric curve-fitting technique to derive the Bézier curve parameters for individual links that are projected using the Universal Transverse Mercator (UTM) coordinate system, which provides a locally Euclidean space. These parameters were directly mapped to the link ID and are stored in a key-value pair data structure, which enables a fast retrieval and evaluation of road geometry during traffic flow emulation.

V. Traffic Flow Routing and Emulation

We present the concept of *traffic flow emulation* as the reconstruction of vehicle movements along the road network using traffic observations provided through road-side sensors and traffic flow theory. This effort involves two steps: (1) traffic flow routing and (2) emulation of vehicle movements, as shown in Fig. 3. A large-scale emulation of traffic flow is a sophisticated endeavor, especially in urban areas where road connectivity and directionality at intersections are complex. In this research, we only emulated traffic flow on interstate highways where road connectivity is more straightforward. In the digital environment, a time counter was set up to simulate the progression of time in a day at millisecond intervals. To emulate traffic flow for a given timeframe, the traffic emulation module was executed to calculate vehicle positions along the road network at every millisecond, beginning at a user-selected timestamp. In this setup, we were able to animate individual vehicle movements on the I-75 and I-24 highways in the entire Chattanooga region.

A. Traffic Flow Routing

As the RDS observations are not available at every link in this study, as such, emulation of a continuous flow of traffic along the highway requires a spatial-temporal interpolation technique (shown in Fig. 3) that can predict (1) vehicle counts

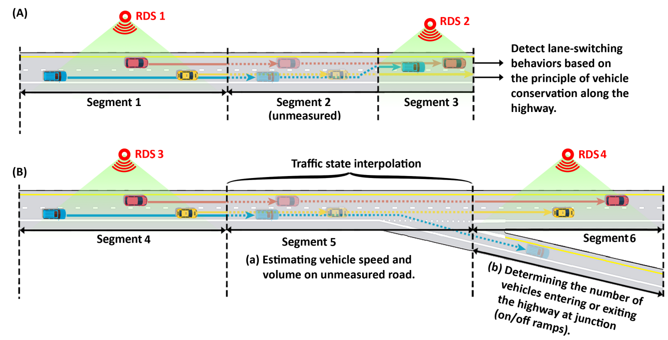

Fig. 3: Infographic of our lane changing, traffic flow routing and interpolation methodology. The top portion (A) shows a simple section of road which has no ramps and consists of 3 links. Two of the links are measured by an RDS sensor, and traffic on the unmeasured link is interpolated. The bottom portion (B) presents a more complex case, in which one of the links has a ramp through which vehicles may leave the road. Based on the difference between vehicle counts of the previous and next sensor, we determine the number of exiting vehicles, as well as lane changes between sensors.

and speeds on unmeasured links and (2) vehicle movements at highway junctions (such as tracking how many vehicles enter and exit a highway). We named this technique *traffic flow routing* because it is similar to the flow routing concept in hydrology [50], which utilizes the flow continuity theory [51] to interpolate flow conditions on ungauged streams using measurements from gauged streams. Conceptually, both stream and highway traffic conserve quantity (e.g., water and vehicles) in a confined network where the quantity can only enter or exit through the network nodes, such as watershed pour points and highway ramps. The flow equilibrium concept of hydrologic routing can be applied to derive the traffic volume on the unmeasured road link by subtracting inflow traffic from adjacent highway links and junctions with outflow traffic at the current road link.

1) Method: The use of the flow continuity principle for traffic flow routing is based on the following conditions. First, it is hypothesized that the highway network is an enclosed system in which vehicles can only exit and enter the network through ramps. It is also assumed that highway links are unidirectional, so vehicles can only move in a single direction. In this regard, a law of vehicle conservation [29] can be applied to analyze the traffic volume on individual links in the highway system. Second, there are abundant and freely available traffic observations being collected along the highway and at ramps. In the given scenario, RDS sensors are placed at approximately half-mile intervals along the highway. Third, the sequence of RDS sensors can be derived through the road topology that describes the upstream and downstream connection between RDS sensors along the highway.

With all the premises satisfied through the data integration and preprocessing step, one can apply simplified traffic flow theory [28, 27] to conduct traffic flow routing on the highways in the Chattanooga area. Fig. 3a provides an example of traffic flow routing, in which lane-level vehicle counts and average driving speed in Segments 1 and 3 are measured by RDS sensors, and Segment 2 is unmeasured. Vehicle trajectories obtained from the RDS measurements are highlighted using solid lines, and the interpolated vehicle trajectories are represented by dashed lines. On measured roads, we apply an in-road approach to represent the detected vehicle from the RDS observation made at a time t, and we simulate its movements as travel distance along each link. For each detected car on the measured road (Segment 1) at a time t_0, an anticipated travel time (Δt) is calculated by assuming the vehicle is driving at a constant speed measured by the corresponding sensor (RDS1). This anticipated travel time describes the required time span for a vehicle to travel from one sensor to the next at a constant speed. In this example, it represents the time from when it is detected by RDS1 until it is detected by RDS2. The vehicle is expected to reach Segment 3 and be detected by its associated sensor (RDS2) at the time $t_1 = t_0 + \Delta t$.

Based on the conservation of flow and the traffic stream characteristics proposed in [28], we can back-calculate the traffic volume, speed, and lane-switching behavior in unmeasured links. Depending on the road configuration, our traffic state interpolation can be divided into two scenarios shown in Fig. 3: (A) highway links without on ramps or off ramps (Segments 1–3), and (B) highway conjunctions with ramps (Segments 4–6). In the first scenario, the number of vehicles detected by RDS1 at time t should be the same as the vehicles detected by RDS2 at the time $(t_0 + \Delta t)$ on Segment 3. The second scenario presents a more complex situation since some vehicles may enter or exit the highway using the junction located between Segment 4 and Segment 6, thereby breaking the law of flow conservation. Subsequently, the difference in total vehicle counts between RDS3 and RDS4 allows us to make inferences on the number of vehicles (e.g., the blue car) that exit through the ramp, helping us interpolate traffic state on unmeasured segments. Lane-switching behaviors can also be deduced along a *series* of links without on or off ramps (as illustrated in Fig. 3a) by applying the flow conservation law at the individual lane level.

2) Result Validation: In this setting, the traffic flow routing technique utilizes measurements from consecutive sensors to validate the interpolation of the vehicle count and driving speed. This traffic flow routing technique is implemented as a module using NodeJS. During a traffic emulation task, the traffic flow routing module is directly connected to the RDS data feed and is executed automatically to update the interpolation of traffic state using the most recent RDS observation. The final output of the traffic flow routing is an integrated data table that reports traffic conditions across all links in the study area. For each link, the traffic condition, describing the traffic volume and average driving speed in individual lanes, is either observed from the RDS sensor or interpolated using traffic flow routing.

We conduct a series of verification checks to validate the interpolated traffic conditions using RDS measurements. During the validation, we selected a combination of three consecutive RDS sensors along a section of highway (labeled as RDS1-3 from the start to end of the highway link along the driving direction), and estimated traffic conditions on the link that is monitored by RDS2 (the middle sensor), as if it was an unmeasured link. Here we demonstrate the validation conducted using measurements from 10 different combinations of adjacent RDS sensors along I-75 and I-24 in Chattanooga, Tennessee. After using three days of RDS measurements collected at a total of 30 sensors on 2018-12-07, 2019-10-15, and 2020-9-24 to interpolate traffic conditions (e.g., traffic volume and speed in each lane) for the middle sensor, we compare the interpolated results with the actual measurement acquired by RDS2 and achieved an accuracy of 87.5% in traffic volume and standard error of ± 4.26 mph between interpolated and measured speeds. The interval for each comparison ranges from 50-90 seconds based on the average total time required for a vehicle to travel from RDS1 to RDS3 with a total driving distance of 1.47 miles, and a total of 11,500 sets of interpolated results were validated against the reality measured by RDS2.

B. Emulation of Vehicle Movements

In this task, the required data inputs include (1) the traffic condition table derived from traffic flow routing; and (2) the optimized road geometry and topology, produced in the data integration and preparation step.

1) Calculation of Vehicle Positions: We use an Agent-based Modeling (ABM) environment and traffic conditions created in the traffic flow routing step to calculate vehicle positions across the region. The environment represents individual vehicles as agents to model their movements at the microscopic scale, and it calculates lane-specific vehicle positions using the road geometry encoded through the optimized road network model as a foundation. Through an animation of vehicle positions calculated at a consecutive series of timestamps, the ABM environment is able to generate continuous trajectories for individual vehicles. To animate a smooth traffic flow, we repeat this calculation every millisecond and use the latest traffic condition table, which is updated every second.

The detailed calculation is executed as shown in Fig. 4. At the beginning of the traffic emulation, the ABM environment is initialized and starts to analyze the traffic condition data for each link. On each link, vehicles are created as agents that are uniformly distributed along the link. Each agent uses three attributes to pinpoint its position. These attributes are (1) travel distance along the link; (2) link ID; and (3) lane number. As time elapses, we use a displacement and acceleration equation to calculate the increment of travel distance for every agent using the average driving speed for its pertaining lane. The average driving speed is defined (1) using direct RDS measurement for agents that represent vehicles on measured links; and (2) through linear interpolation using two consecutive RDS measurements for agents that represent vehicles on unmeasured links, as shown in Fig. 3.

Once the travel distance of an agent is greater than the length of its current link, the agent is then placed at the origin of the immediate downstream link. To ensure the data quality, we visually cross-check the derived connectivity (especially their topological relationship) for 541 links and 214 RDS sensors in QGIS by overlaying their geometry with ESRI street and aerial imagery map (through HCMGIS plugins). Since we use composite Bézier curves [52] to represent the link's geometry, the geographic positions of any vehicle agent can be calculated through parametric curve equations, which express the 2D position (the X and Y coordinate) as a function of travel distance along the curve.

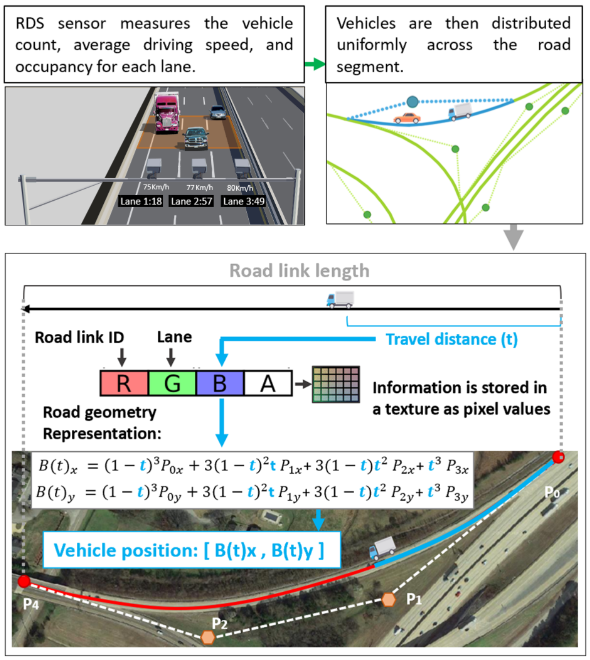

Fig. 4: To calculate vehicle positions, we distribute the vehicles that are measured by the sensors along the corresponding links. Then, we determine their absolute position on the Bézier curve by setting t to the relative travel distance. Finally, we store all required information in an RGBA texture for fast rendering with GPGPUs.

To ensure interactive performance of a large-scale traffic flow emulation that involves movements of hundred thousands of vehicles at the regional level, we employ a common general-purpose computing on graphics processing units (GPGPU) design pattern to speed up the calculation of individual vehicle speeds. Graphics cards are very efficient at processing and displaying image data which is stored as numbers in so-called *RGBA textures*, which have red, green, blue, and alpha (transparent) values for each pixel. We encode attributes of vehicle agents (represented as floating point numbers) into four non-negative integer values of an RGBA texture that is defined using WebGL. These vehicle attributes include (1) link ID; (2) lane number; and (3) the vehicle position as the 1D longitudinal distance along their pertaining links. In this setup, the calculation of the travel distance for each vehicle can be performed in WebGL using parallel rendering at every millisecond. The final output is an array of coordinates that represents individual vehicle positions.

2) Car-Following and Lane-Changing Behaviors: To emulate interactions between vehicles at the microscopic scale, we employed the simplest form of the psycho-physical vehicle-following model in the ABM environment to control individual vehicle driving behavior. Each vehicle defined in the ABM environment is associated with a mathematically defined safety buffer that surrounds the vehicle position. This safety buffer spans the vehicle, a safe following distance, and an additional look-ahead buffer ahead of the vehicle to emulate drivers looking ahead to evaluate the traffic in front of them. This look-ahead buffer is set to 0.12 miles ahead of the vehicle.

In every emulated scenario, we define the distance between vehicles at the beginning of the emulation when vehicles are uniformly distributed along a link as agents. As the emulation proceeds, speeds and distances between vehicles will begin to vary when vehicles experience speed changes across different links. When two vehicles are driving in the same lane of a link, both vehicles are traveling at the derived driving speed that is either directly measured by a RDS sensor (on measured links) or linearly interpolated using two RDS speed measurements (on unmeasured roads between two measured links). Without the presence of traffic incidents, the two vehicles are traveling at a uniform speed with a constant distance in between, which is greater than or equal to a defined safe following distance.

When the pair of predecessor and follower approach slower traffic ahead of them, they will initially adapt to the slower traffic. As the predecessor's safety buffer touches such an obstacle, its driving speed will be gradually reduced to zero. Once the follower's safety buffer intersects with the predecessor's safety buffer, the desired speed of the following vehicle is gradually changed to the same speed as its predecessor when it is driving across its search distance, which is zero in this scenario.

When the a vehicle is stopped in its lane, the following vehicle will detect the stopped traffic once its safety buffer touches the safety buffer of its slow-moving predecessor, and in response, is programmed to change to adjacent lanes that are not occupied by other vehicles' safety buffer. All other followers will repeat this lane-changing behavior.

VI. MULTISCALE FLOW VISUALIZATION

In this section, we first present the overall design of our visual interface and then show the details of each visualization technique within our visual interface at different spatial scales. We designed a visual workflow to visualize the emulated traffic flow from the ABM. This workflow was designed to present the traffic dynamics from macroscopic to microscopic scale using a combination of aggregation and visualization techniques.

A. Macroscopic Scale: High-Level Region Overview

Vehicle density at the macroscopic scale can help transportation agencies detect traffic jams and bottlenecks. We devised an animated trail and particle representation [48, 49], presenting each vehicle as a circle, along with a modified kernel density map that simultaneously encodes the magnitude of vehicle congestion and density at the regional level. Specifically, the modified kernel density map calculates the weight of individual vehicles based on their driving speed, as defined in (1), thereby automatically highlighting clusters of slow-moving vehicles and detecting traffic jams across the region.

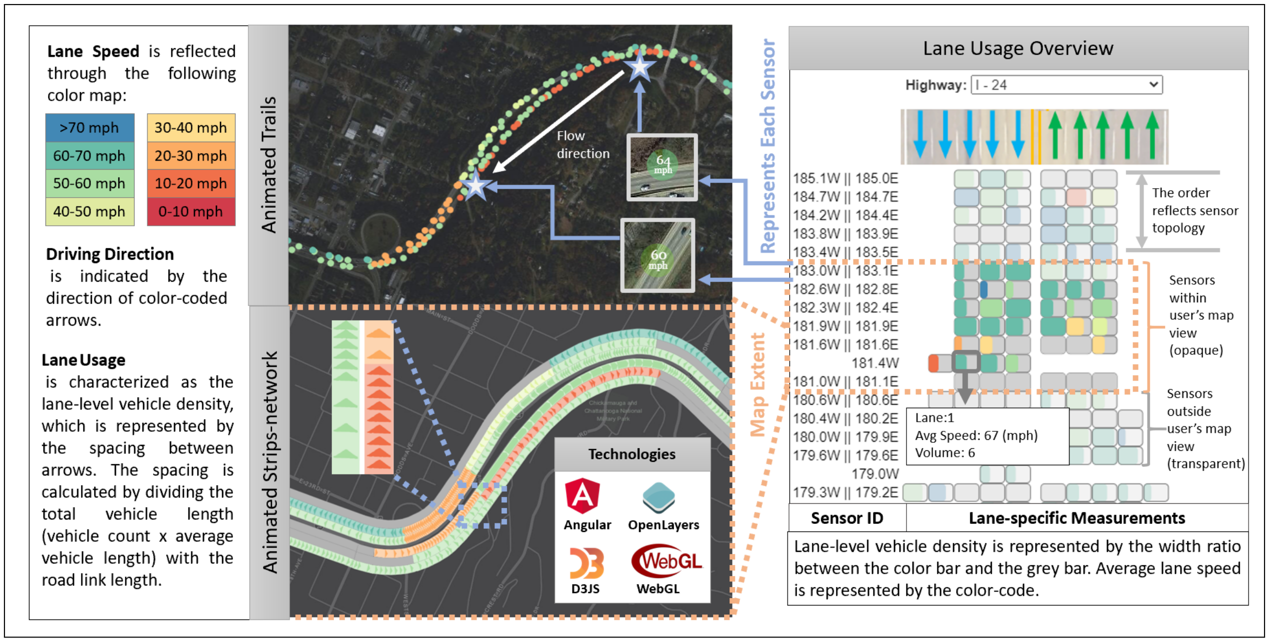

Fig. 5: This shows a comparison of an animated trails visualization (used at microscopic scale) and our proposed animated strips-network for visualizing lane usage at mesoscopic scale for a section of road that has a speed limit of 55 mph. Major advantages of animated strips-networks include indication of driving direction in static views, and the display of traffic density in individual lanes. A color-coded lane-usage matrix representation is developed and integrated with the animated strips-network to provide a coordinated view for visualizing detailed traffic condition in individual lanes.

In Fig. 1, for example, we observe the significant increase in vehicle density near highway interchanges and ramps.

$$\text{Weight} = 1 - \frac{\text{vehicle speed}}{\text{freeflow speed}} \tag{1}$$

To display the accurate traffic state for each vehicle cluster in the vehicle density map, we adopt a color-coded bubble representation (presented in the legend box ”RDS sensors” in Fig. 5) to visualize average vehicle driving speeds measured by RDS sensors. The color of the bubble is based on the speed of the vehicle (as illustrated by the legend in Fig. 5).

B. Mesoscopic Scale: Lane Usage and Performance at Corridor Level

In this work, we adopted an animated strips-network representation to visualize vehicle density, speed, and driving direction within individual lanes. An integrated visualization of these parameters can be used to reflect individual lane-usage and performance in each highway link. The emulated traffic flow data were aggregated to provided summary statistics for each lane of a link. We also devised a lane-usage matrix, which is connected to the animated strips-network representation through a coordinated view, to provide an overview of lane-specific traffic conditions and road topology at the road corridor level. In Fig. 5, we present the design and interpretation of these innovative visualization techniques to demonstrate their effectiveness for visualizing lane-level traffic dynamics at mesoscopic scale.

1) Animated Strips Network: The representation consists of color-coded strips and moving arrows. In each lane (represented as a strip), the vehicle driving direction is indicated by the arrow direction and the vehicle density is encoded by the spacing between arrows. The color designation is created based on the average vehicle driving speed in the lane. Compared with the traditional animated trails, the strips-network can explicitly visualize multiple lane attributes that reflect lane-usage and performance for multiple connected links simultaneously.

For example, a green strip with dense arrow placement would indicate a well-performing lane, packed with vehicles driving at free-flow speed with no congestion. A red strip with very sparse arrows would indicate that the lane has low usage and contains only a few slow-moving vehicles.

This style of visualization addresses the issue of traditional animated trails and particle representations that can only visualize lane-level driving behavior at a very small spatial scale (e.g., a single link). At a zoomed-out view, which covers multiple links, the animated trails cannot accurately map vehicle positions to individual lanes, whereas animated strips-networks can provide a much more complete view of the network state at lane-level.

2) Lane-usage Matrix: In the vertical bar on the right side of the interface (i.e., lane usage view in Fig. 5), we provide detailed sensor data for each lane that is observed using a novel lane-usage matrix. The design of this visualization is loosely based on the concept of a heatmap. Each row represents data from one or two sensors, sorted North to South (for I-75) or West to East (I-24). We “paired” each roadside sensor with its nearest counterpart in the opposite traffic direction as much as we could. Then, in the "columns" of this matrix, we visualized the 2 to 5 lanes of data collected by each sensor, starting from the median in the middle and going toward both sides to mirror the physical road layout (i.e., lane configuration), which is illustrated at the top of this display. Each lane is represented by a cell, which is updated every 30 seconds with the latest data. The cell's color represents the average speed of vehicles in this lane, and it's "fill level" represents the volume. For low volumes, only a small part of the lane is filled, whereas for high volumes, the lane is full. The fill level is calculated by dividing the traffic volume by the capacity of the lane. If there were no vehicles in the selected time interval (past 30 seconds or a rolling window of a specified length, e.g., one hour), the lane is colored gray. If the user zooms in on the map, the sensors that fall within the map view remain at full opacity, whereas all other sensors are rendered less opaque.

The resulting visualization provides a high-level overview of speeds and volumes. Despite its heatmap ancestry, this new visualization is more powerful as it (1) shows the topology and lane-changes along the road; (2) uses cell's fill level and color to represent both speed and volume; and (3) highlights sensor cells based on users' map view. Hover-over reveals additional information, such as the lane number.

3) User Interaction: The visual interface also applies coordinated view to allow users to explore the traffic flow holistically through linked visualizations. For example, the lane-usage matrix is linked to the animated strip-network on the map to connect the overview of lane usage at road corridor level with detailed lane-level usage and performance displayed through the animated strips-networks on the map. When a user clicks individual cells within the matrix, the map automatically zooms to the corresponding sensor location and displays more detailed lane-level dynamics of the highway links near the sensor, using the animated strips network.

Moreover, the lane-usage matrix de-emphasizes cells that represent sensors outside the user's map view (bounding box) by applying transparency to them, so that the user can easily understand what portion of the matrix is displayed as the animated strips-networks in the map (as shown in Fig. 5)

C. Microscopic Scale: Driving Behaviors at Individual Roads Level

At the microscopic scale, the traffic flow community is particularly interested in monitoring individual vehicle driving behaviors that include car-following activities in the longitudinal direction and lane-changing activities at the lateral direction [12]. In our visual workflow, we adopt an animated trail representation to visualize the trajectory of each vehicle. The longitudinal vehicle behaviors are emulated based on basic principles of traffic flow theory during the *traffic flow routing* step described in Section V-A. Meanwhile, the discrete lateral choices of vehicles are represented by lane-specific trajectories, which are generated in the *emulation of vehicle movements* step depicted in Section V-B.

VII. CASE STUDY AND RESULTS

The following case study aims to demonstrate the capabilities and effectiveness of our visual storytelling workflow for enabling the situational awareness and improving the understanding of complex traffic dynamics that occur at different spatial scales.

By combining RDS measurements with traffic incident location data, we constructed the emulated traffic flow to reveal how traffic at different spatial scales behaved during the afternoon peak, and how it responded to a traffic incident. This flow emulation is based on the RDS data and 911 incident data collected on September 24, 2020. The 911 incident data recorded a major traffic accident near exit 11 on I-75 southbound around 15:45. Fig. 6 provides examples of these scenarios:

The first example shows the traffic on I-24 in the middle of the afternoon peak from 14:30 to 16:40. The lane usage matrix on the right displays the raw data from each sensor. The right-hand side of the matrix dispays eastbound traffic, and there are two areas that are bottlenecks: the first one is near mile-marker 183.5E. This bottleneck is close to the I-24/I-75 junction. This area is a known hotspot for local traffic agencies. After a steep hill with two truck lanes and one no-truck lane, drivers have 2 miles to merge into their travel lanes prior to the junction. Further down in the matrix, there are several sensors with high traffic volumes and slow traffic. This is an area after a junction of I-24 (from the bottom left of the image) and US-27 (from the top left of the image), with many exits within a short distance. One interesting note is that traffic in the right-most lane moves faster than in the other lanes (179.2E to 179.9E). Further investigation showed that this lane turns into an exit lane, so most traffic is expected to merge into the travel lanes. It is interesting to note that several sensors did not record any traffic during this time frame. Nonetheless, this does not affect the animated trails visualization or the kernel density map on the left, as the individual vehicles are interpolated smoothly between sensors. However, it has some adverse effects on the animated strips representation, as the road network has segments of many varying lengths and the sensors without data result in additional interpolation. More densely placed sensors would help address this issue.

The second example shows traffic on I-75 around a very serious traffic accident that was reported at 15:09. This traffic accident occurred near exit 11 in the southbound direction, and it included an entrapment. As a result of this accident, the entire southbound direction of the highway was closed while the crews from 18 response vehicles (11 law enforcement, 2 medical, 5 fire) secured the scene, rescued and treated entrapped and/or injured individuals, and coordinated lane closure and reopen over the course of few hours (the incident was finally resolved at 17:55). During this early phase of the incident, the animated strips view is very informative: one can already see that traffic in the right two lanes was at a complete standstill (solid red), while it was still moving very slowly in the other two lanes. Beyond the accident, there was very little traffic, most of which was likely from down-stream on-ramps. In the opposite travel direction, one can see traffic slowing down near the scene of the accident (strip color changing from blue to green). In the macroscale view (left), the accident shows up as a very bright spot on the kernel density map, and vehicles in the area are all colored red, indicating very slow-moving traffic. Just outside of the viewport (grayed out) on the south, the lane usage matrix reveals another congestion, which was caused by another traffic accident occurred near mile marker 9 in the southbound direction. Upstream traffic (also grayed out) was moving briskly until it approached the incident.

A. *User Evaluation*

An evaluative survey was sent to a panel of four transportation research scientists at the National Transportation Research Center (NTRC) to assess the design of the novel traffic visualizations presented in this work. The survey was not designed to assess the usability of the emulation app using formal protocols and cognitive walk-through. Instead, the objective of the survey was to (1) collect qualitative feedback on the usefulness of the visualizations and user-friendliness of the visual workflow; and (2) learn if these visualizations can be applied to additional transportation applications. All users commented positively on the usefulness of the platform and offered new visions and use cases for these traffic visualizations. Examples of the feedback included "road types and some aggregated evaluation metrics (hotspot locations, level of services, traffic accidents) can be visualized through these visualizations," and "these dynamic visualizations can effectively complement the traditional space-time diagram of traffic flow." Critique from the survey included "the aesthetic design of the interface can be improved" and "the responsiveness and the coordination between these visualizations and the map can be further improved". In summary, all users provided some positive feedback. To address the critique we received, we applied corrective improvements to the visual workflow.

VIII. CONCLUSION AND FUTURE WORK

The paper presents the design and implementation of a web-based traffic flow emulation and visualization app to improve the situational awareness and management of urban mobility. Our main contributions are:

- **Data Workflow**: We created an integrated pipeline that integrates traffic data from discrete road-side sensors and optimizes the data storage of the highway networks using a composite Bézier curve-based data representation.
- **Methodology**: We developed an innovative traffic flow routing method, which was inspired by the hydrologic routing method; we also developed an ABM-based vehicle movement calculation method to reconstruct/emulate a continuous traffic flow on highways using near-real-time RDS data.
- **Visualization**: We designed a visual workflow that combines various traffic flow visualizations using LoD techniques to reveal traffic dynamics of the emulated traffic from macroscopic to microscopic scale. In addition, two novel geo-visualization techniques were proposed to visualize lane usage and performance at the mesoscopic scale.

With the capability of replicating a continuous traffic flow using traffic conditions measured by discrete road-side RDS sensors, the proposed traffic emulator maintains a holistic systems approach for achieving the situational awareness at the regional scale, and it achieves this with cost-efficient means compared to conventional investigative alternatives, which involve the use of computational-intensive traffic simulation models and the acquisition of a limited amount of vehicle-in-motion data (e.g., in-vehicle GPS data and wireless beacon). The proposed method can be best applied on highways that are monitored by an array of sensors and do not have complex road geometry and connectivity, such as roundabouts, shared turning lanes, access points, and roadside parking. These complexities can break the conservation of vehicular flow in the enclosed highway network and increase the uncertainty of the traffic emulation. A well-established GIS road network is also required for the emulation. Fortunately, these data types are increasingly available through the advancement of traffic monitoring technologies, as well as the expanding traffic sensor network. These technologies can measure traffic conditions over large scales at a fraction of the cost compared to conventional observational means.

To make the emulation and visualization easily accessible, we implement the emulator as a web app and incorporated the app into the CTwin platform. As it is a web-based platform, the traffic emulation can be updated as more RDS observations are retrieved, allowing for a centralized situational awareness tool that can provide information on highway traffic in Chattanooga. The system is developed in a flexible and extendable structure to address similar transportation management concerns at the national level. The platform was built with open-source technologies that make the system light-weight, low cost, and adaptive.

Our future work includes (1) improving the traffic flow routing algorithm to enable the interpolation of traffic conditions in complex urban road networks; (2) applying a more sophisticated version of traffic flow theory to improve the accuracy of the interpolation of traffic conditions on unmeasured roads; (3) supporting fusion of data from more data sources, such as GPS trace data, to emulate and validate the on-going traffic flow; and (4) utilizing additional sensors (e.g., CCTV cameras, smart traffic cameras) along the highways and at ramps to further validate traffic flow routing and emulation, as well as to further incorporate vehicle types (e.g., passenger vehicles and trucks) into consideration.

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy (U.S DOE), Office of Energy Efficiency and Renewable Energy, Vehicle Technologies Office. We would like to thank the Tennessee Department of Transportation for providing radar data, the Chattanooga Department of Transportation and the Chattanooga Transportation Planning Organization for their guidance and providing the road network data, and the Hamilton County 911 Services for providing incident data. Finally, we would like to thank Timothy J. LaClair, Wan Li, Yunli Shao, and Jackeline Rios-Torres for their feedback and participation in the user evaluation.

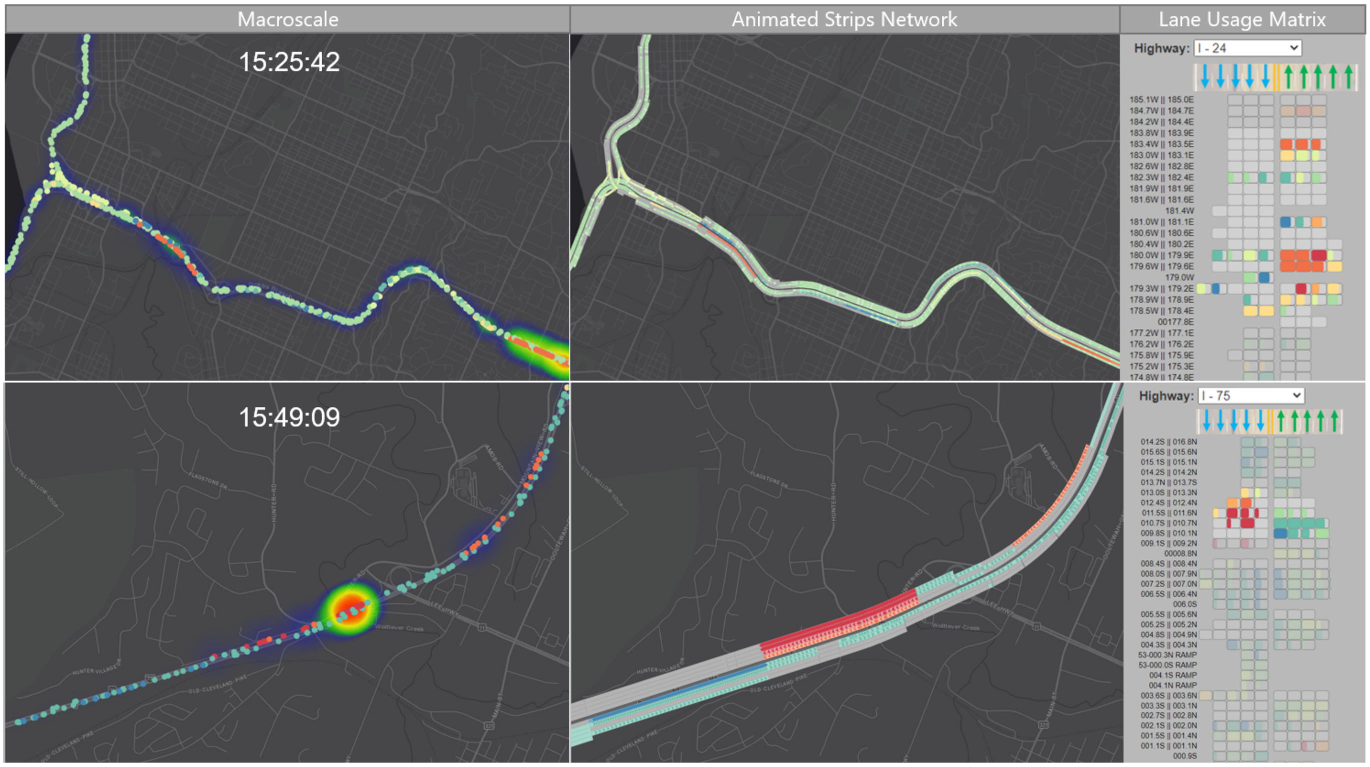

Fig. 6: A case study that reveals how traffic dynamics at different scales respond during two different scenarios. The first row shows traffic on I-24 Eastbound (away from the city center) during the afternoon peak, with a large hotspot at the bottom right of the viewport and a smaller hotspot near a junction on the left. The second row shows traffic near a major traffic accident on I-75 Southbound, a few minutes after the accident occurred. In each row, we present a macroscale visualization with animated trails on top of the kernel density map (left), the animated strips network (middle), and the lane usage matrix (right).

REFERENCES

[1] S. Riffat, R. J. Powell, and D. Aydin, "Future cities and environmental sustainability," *Future Cities and Environment*, vol. 2, pp. 1–23, 2016.

[2] United Nations, *More than half of world's population now living in urban areas, UN survey finds*, July 2014.

[3] D.-B. Nguyen, C.-R. Dow, and S.-F. Hwang, "An Efficient Traffic Congestion Monitoring System on Internet of Vehicles," 2018. [Online]. Available: https://www.hindawi.com/journals/wcmc/2018/9136813/

[4] C. B. . D. N. Beth Conlan, Duncan Kay, *Air Quality and Road Transport Impacts and solutions Guy Hitchcock*. RAC Fundation, June.

[5] Federal Highway Administration, *Traffic Detector Handbook Third EditionVolume II*, October 2006.

[6] W. Zhang, G. Tan, N. Ding, and G. Wang, "Traffic Congestion Evaluation and Signal Control Optimization Based on Wireless Sensor Networks: Model and Algorithms," *Mathematical Problems in Engineering*, vol. 2012, pp. 1–17, 2012.

[7] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. v. d. Wetering, "Visual traffic jam analysis based on trajectory data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2159–2168, Dec 2013.

[8] Y. Gu, L.-T. Hsu, and S. Kamijo, "Towards lane-level traffic monitoring in urban environment using precise probe vehicle data derived from three-dimensional map aided differential gnss," *IATSS Research*, vol. 42, no. 4, pp. 248 – 258, 2018.

[9] Y. Tsukamoto, M. Ishizaki, A. Hiromori, H. Yamaguchi, and T. Higashino, "Multi-lane detection and tracking using vision for traffic situation awareness," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308)*. IEEE, 2020, pp. 1–6.

[10] S. Jianjun, W. Xu, G. Jizhen, and C. Yangzhou, "The analysis of traffic control cyber-physical systems," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2487–2496, 2013.

[11] H. Wang, M. Zhu, W. Hong, C. Wang, G. Tao, and Y. Wang, "Optimizing signal timing control for large urban traffic networks using an adaptive linear quadratic regulator control strategy," *IEEE Transactions on Intelligent Transportation Systems*, 2020. [Online]. Available: https://doi.org/10.1109/TITS.2020.3010725

[12] D. Ni, "Multiscale modeling of traffic flow," *Mathematica Aeterna, Hilaris Ltd*, vol. 1, 01 2011.

[13] Oak Ridge National Laboratory, *Chattanooga Digital Twin*, January 2019. [Online]. Available: "https://ctwin.ornl.gov/"

[14] J. Guerrero Ibáñez, S. Zeadally, and J. Contreras Castillo, "Sensor technologies for intelligent transportation sys-

tems," *Sensors*, vol. 18, no. 4, 2018.

[15] European Environment Agency, *Investment in Transport*, Sep 2010.

[16] B. Płaczek, "A self-organizing system for urban traffic control based on predictive interval microscopic model," *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 75 – 84, 2014.

[17] M. Lewandowski, B. Płaczek, M. Bernas, and P. Szymala, "Road traffic monitoring system based on mobile devices and bluetooth low energy beacons," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 3 251 598:1–3 251 598:12, 2018.

[18] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. v. d. Wetering, "Visual traffic jam analysis based on trajectory data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2159–2168, Dec 2013.

[19] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, Sep. 2012.

[20] A. S. M. Karpiński and V. Cahill, "Sensor networks for smart roads," in *The 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '06)*, March 2006, pp. 310–314.

[21] R. Prabha and M. G. Kabadi, "Knodet: A framework to mine gps data for intelligent transportation systems at traffic signals," in *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, March 2017, pp. 85–89.

[22] A. Sharma, V. Ahsani, and S. Rawat, "Evaluation of opportunities and challenges of using inrix data for real-time performance monitoring and historical trend assessment," Transportation Research Board, Tech. Rep. No. SPR-P1 (14) M059, 2017.

[23] B. Kuijpers, *Space-Time Prism Model*. Cham: Springer International Publishing, 2017, pp. 1926–1932. [Online]. Available: https://doi.org/10.1007/978-3-319-17885-1_1599

[24] J. G. Shanthikumar and R. G. Sargent, "A Unifying View of Hybrid Simulation/Analytic Models and Modeling," *Operations Research*, vol. 31, no. 6, pp. 1030–1052, Dec. 1983.

[25] *PTV VISSIM 11 User Manual*, PTV AG, Karlsruhe, Germany, 2019.

[26] D. Krajzewicz, "Traffic simulation with SUMO–simulation of urban mobility," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 269–293.

[27] S. Logghe and B. Immers, "Multi-class kinematic wave theory of traffic flow," *Transportation Research Part B: Methodological*, vol. 42, pp. 523–541, 07 2008.

[28] F. L. Hall, "Traffic stream characteristics," *Traffic Flow Theory. US Federal Highway Administration*, vol. 36, 1996.

[29] B. Seibold, *Traffic Flow Theory*, September 2015.

[30] N. Willems, H. Van De Wetering, and J. J. Van Wijk, "Visualization of vessel movements," *Computer Graphics Forum*, vol. 28, no. 3, pp. 959–966, 2009.

[31] N. Andrienko and G. Andrienko, "Spatial generalization and aggregation of massive movement data," *IEEE transactions on visualization and computer graphics*, vol. 17, pp. 205–19, 03 2011.

[32] A. S. Berres, H. Xu, S. A. Tennille, J. Severino, S. Ravulaparthy, and J. Sanyal, "Explorative visualization for traffic safety using adaptive study areas," *Transportation Research Record*, 2021. [Online]. Available: https://doi.org/10.1177/0361198120981065

[33] G. Andrienko, N. Andrienko, G. Fuchs, and J. Wood, "Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2120–2136, Sep. 2017.

[34] T. Sobral, T. Galvão, and J. Borges, "Visualization of urban mobility data from intelligent transportation systems," *Sensors*, vol. 19, no. 2, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/2/332

[35] C. F. Huacón and L. Pelegrin, "Surv: A system for massive urban data visualization," in *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*, Nov 2017, pp. 1–4.

[36] A.-D. Shamal, F. Kamw, Y. Zhao, X. Ye, J. Yang, and S. Jamonnak, "An open source TrajAnalytics software for modeling, transformation and visualization of urban trajectory data," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 150–155.

[37] R. Krueger, G. Sun, F. Beck, R. Liang, and T. Ertl, "Traveldiff: Visual comparison analytics for massive movement patterns derived from twitter," in *2016 IEEE Pacific Visualization Symposium (PacificVis)*, April 2016, pp. 176–183.

[38] A. Keler, J. M. Krisp, and L. Ding, "Detecting vehicle traffic patterns in urban environments using taxi trajectory intersection points," *Geo-spatial Information Science*, vol. 20, no. 4, pp. 333–344, 2017.

[39] R. Scheepens, C. Hurter, H. Van De Wetering, and J. J. Van Wijk, "Visualization, selection, and analysis of traffic flows," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 379–388, 2015.

[40] G. Sun, R. Liang, H. Qu, and Y. Wu, "Embedding spatio-temporal information into maps by route-zooming," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 5, pp. 1506–1519, May 2017.

[41] W. Zeng, C. Fu, S. Mller Arisona, S. Schubiger, R. Burkhard, and K. Ma, "Visualizing the relationship between human mobility and points of interest," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2271–2284, Aug 2017.

[42] K. Vrotsou, G. Fuchs, N. Andrienko, and G. Andrienko, "An interactive approach for exploration of flows through direction-based filtering," *Journal of Geovisualization and Spatial Analysis*, vol. 1, 05 2017.

[43] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao, "Visual analytics of mobility and transportation: State of the art and further research directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, April 2017.

[44] Y. Hyougo, K. Misue, and J. Tanaka, “Directional aggregate visualization of large scale movement data,” in *Proceedings of the International Conference on Information Visualisation*, 07 2014, pp. 196–201.
[45] A. S. Berres, T. LaClair, C. Wang, H. Xu, and J. Sanyal, “Multiscale and multivariate transportation system visualization for shopping district traffic and regional traffic,” *Transportation Research Record: Journal of the Transportation Research Board*, Dec. 2020. [Online]. Available: ”https://doi.org/10.1177/0361198120970526”
[46] R. Scheepens, C. Hurter, H. Wetering, and J. Wijk, “Visualization, selection, and analysis of traffic flows,” *IEEE transactions on visualization and computer graphics*, vol. 22, 09 2015.
[47] V. Cristie, M. Berger, P. Bus, A. Kumar, and B. Klein, “Cityheat: Visualizing cellular automata-based traffic heat in unity3d,” in *SIGGRAPH Asia 2015 Visualization in High Performance Computing*, ser. SA 15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1–4. [Online]. Available: https://doi.org/10.1145/2818517.2818527
[48] G. A. M. Gomes, E. Santos, and C. A. Vidal, “Interactive visualization of traffic dynamics based on trajectory data,” in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Oct 2017, pp. 111–118.
[49] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan, “TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection,” in *2011 IEEE Pacific Visualization Symposium*. IEEE, mar 2011, pp. 163–170.
[50] C. Tague and L. Band, “Evaluating explicit and implicit routing for watershed hydro-ecological models of forest hydrology at the small catchment scale,” *Hydrological Processes*, vol. 15, no. 8, pp. 1415–1439, 2001.
[51] Z. Chen, G. Huan, and Y. Ma, *Computational methods for multiphase flows in porous media*. SIAM, 2006.
[52] A. Masood and S. Ejaz, “An efficient algorithm for robust curve fitting using cubic bezier curves,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, D.-S. Huang, X. Zhang, C. A. Reyes García, and L. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 255–262.

Haowen Xu is a Research Scientist in the Computational Sciences and Engineering Division at Oak Ridge National Laboratory. He received his Ph.D. in Civil and Environmental Engineering and a graduate certificate in Geoinformatics at the University of Iowa in 2019. His research interests include watershed management, sediment transport, erosion controls, hydro-informatics, developing web-based environmental Decision Support Systems (DSS), and visual analytics.

Anne Berres is a Research Scientist in the Computational Science and Engineering Division at Oak Ridge National Laboratory. She received her Ph.D. in computer science from the University of Kaiserslautern, Germany in 2015 after completing a BS (2009) and MS (2011) in computer science. Her research focuses on data science and visualization to solve domain problems in transportation, buildings, population dynamics, and smart cities.

Sarah Tennille is a Data Engineer in the Computational Sciences and Engineering Division at Oak Ridge National Laboratory (ORNL). She graduated from the University of Tennessee, Knoxville, with a Master of Science degree in Geography in 2016 and a Bachelor of Science degree in Geography in 2014. Her work has applications in resiliency, critical infrastructure, transportation, energy, climate, and disaster management.

Srinath Ravulaparthy is a Research Scientist with Berkeley Lab with research interests in transportation systems modeling & simulation, travel behavior analysis, and computational statistics. He is a Ph.D. graduate from Department of Geography, University of California, Santa Barbara in Transportation Modeling and Simulation (2013) with M.S in Civil Engineering from Arizona State University (2008).

Chieh (Ross) Wang is an R&D Associate in the Building and Transportation Science Division at Oak Ridge National Laboratory. He earned his Ph.D. in Civil and Environmental Engineering from the Georgia Institute of Technology in 2017. He received his BS and MS degrees in Civil Engineering from the National Taiwan University in 2005 and 2007. His research revolves around transportation data analytics and visualization, intelligent vehicles, smart cities, and infrastructure management.

Jibonananda (Jibo) Sanyal serves as the Group Leader for Oak Ridge National Laboratorys Computational Urban Sciences research group. His work falls at the intersection of HPC, extreme scale data and analytics, modeling and simulation, AI, visualization, and sensors and controls. He has extensive experience applying these techniques towards developing insights and solutions for smart city applications, urban mobility, energy applications, situational awareness tools, as well as emergency response and resiliency across local, regional, and national scales.