Automatic vehicle trajectory data reconstruction at scale

Yanbing Wang [a,b,*], Derek Gloudemans [b,c], Junyi Ji [a,b], Zi Nean Teoh [c], Lisa Liu [d], Gergely Zachár [a,b], William Barbour [b], Daniel Work [a,b,c]

[a] *Department of Civil and Environmental Engineering, Vanderbilt University, United States*
[b] *Institute for Software Integrated Systems, Vanderbilt University, United States*
[c] *Department of Computer Science, Vanderbilt University, United States*
[d] *Department of Electrical Engineering, Vanderbilt University, United States*

ABSTRACT

In this paper we propose an automatic trajectory data reconciliation to correct common errors in vision-based vehicle trajectory data. Given "raw" vehicle detection and tracking information from automatic video processing algorithms, we propose a pipeline including **(a) an online data association algorithm to match** fragments that describe the same object (vehicle), which is formulated as a min-cost network circulation problem of a graph, and **(b) a one-step trajectory rectification procedure formulated as a quadratic program** to enhance raw detection data. The pipeline leverages vehicle dynamics and physical constraints to associate tracked objects when they become fragmented, remove measurement noises and outliers and impute missing data due to fragmentations. We assess the capability of the proposed two-step pipeline to reconstruct three benchmarking datasets: (1) a microsimulation dataset that is artificially downgraded to replicate the errors from prior image processing step, (2) a 15-min NGSIM data that is manually perturbed, and (3) tracking data consists of 3 scenes from collections of video data recorded from 16–17 cameras on a section of the I-24 MOTION system, and compare with the corresponding manually-labeled ground truth vehicle bounding boxes. All of the experiments show that the reconciled trajectories improve the accuracy on all the tested input data for a wide range of measures. Lastly, we show the design of a software architecture that is currently deployed on the full-scale I-24 MOTION system consisting of 276 cameras that covers 4.2 miles of I-24. We demonstrate the scalability of the proposed reconciliation pipeline to process high-volume data on a daily basis.

1. Introduction

1.1. Motivation

Vehicle trajectory data has received increasing research attention over the past decades. We draw particular attention to extracting high-resolution vehicle trajectory data from video cameras as traffic monitoring cameras are becoming increasingly ubiquitous. Video-based trajectory data collection in conjunction with image processing algorithms can offer a "bird's eye" perspective and, with a dense deployment of cameras, can achieve complete spatial and temporal coverage of a roadway segment. The capability to fully observe every vehicle for a long duration of time and over a long corridor is critical to advance research in areas such as training and calibration of traffic flow models, and driving behavior modeling (Li et al., 2020).

---

* Corresponding author at: Department of Civil and Environmental Engineering, Vanderbilt University, United States.
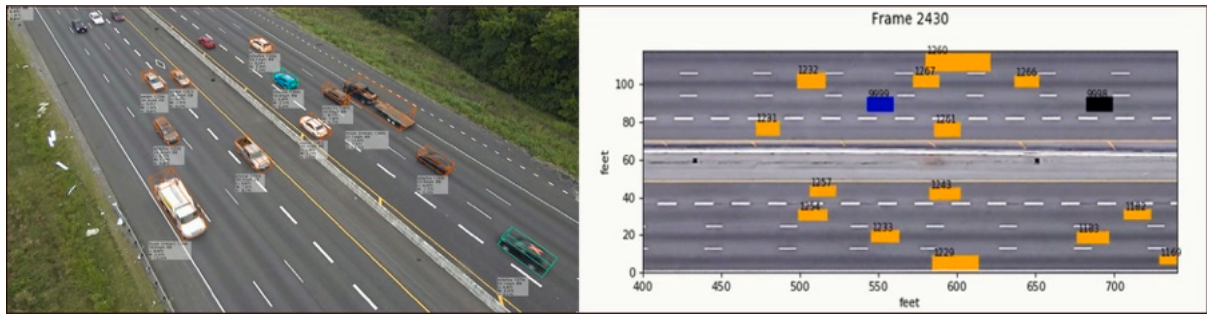*E-mail address:* yanbing.wang@vanderbilt.edu (Y. Wang).

Fig. 1. Left: the field of view from one of the 276 cameras with vehicles detected in 3D bounding boxes. Right: Projected positional measurements in roadway coordinates.

However, vision-based trajectory data by itself is of limited utility for traffic research because of noise and systematic sensing errors, thus necessitates proper processing to ensure data quality. Inaccuracies in such data – projection errors, occlusions, dimensions, and kinematic physicality (Punzo et al., 2011), for example – need to be addressed. Automatic data reconciliation methods are needed to deal with massive volumes of trajectory data from traffic video processing.

This paper describes the development of a two-step data reconciliation and postprocessing pipeline for a densely-instrumented camera network, I-24 MOTION (Gloudemans et al., 2023b). The physical infrastructure was built in collaboration with the Tennessee Department of Transportation (TDOT) and involved 276 fully-connected 4K cameras to monitor a 4.2-mile stretch of Interstate 24. The videos are subsequently processed by an anonymous computer vision algorithm (detailed in Gloudemans et al., 2023b,a, 2024) to produce 3D vehicle detection and tracking data in a roadway reference system (Fig. 1). The continuous field of view shows the entire roadway, providing the opportunity to potentially track nearly every vehicle.

A preliminary investigation on the raw trajectories revealed various errors arise from a combination of factors in system design and computer vision detection and tracking, such as object hand-off across cameras, homography estimation, and the coordinate transformation from imagery to roadway coordinates. We have intentionally designed the post-processing step to be independent of the tracking algorithm and the specific error structures it may contain.

1.2. Challenges

The first challenge in generating high-quality trajectory data through post-processing is the need to tackle a range of issues stemming from the system design and the preceding computer-vision algorithms. One of the foremost challenges is dealing with "tracking fragmentations", where the tracking of vehicles gets frequently disrupted, often due to errors in tracking algorithms, object occlusions, inaccuracies in the transformation from image space to roadway coordinates, and data segmentation due to distributed computation (see Fig. 2). Additional issues include missing detections, false positives, tracking inconsistencies, and inaccurate localization (see Fig. 3). These errors are common in video-processing algorithms, which necessitate the significance of having a comprehensive data post-processing procedure capable of addressing a broad spectrum of issues before the trajectory data can be employed for traffic analysis purposes.

A secondary challenge is that the data postprocessing algorithms should be efficient in order to handle the growing volume of data coming from large-scale real-world camera systems that are designed to enable real-time traffic monitoring and control. Most of the existing algorithms for fragment matching are batch processing algorithms and do not have capabilities to operate online or provide provable optimality, which prevent them from being deployed to a large-scale system with raw tracking data as streaming input. Data reconciliation must be performed in real-time or with low latency to facilitate continuous or even live generation of high-volume data. Therefore, fast and online data reconciliation methods are to be explored.

Last but not the least, while many prior studies in the literature have acknowledged the importance of data cleaning and have proposed various methodologies, there is a noticeable absence of open-sourced datasets and implementations suitable for method benchmarking. A reliable benchmarking dataset should contain raw tracking results and their corresponding ground truth. To initiate benchmarking endeavors, we are making our datasets available (both simulated data and raw tracking data with manually labeled ground truth), along with the post-processing repository used in this paper, accessible to the research community.

1.3. Problem statement and contributions

This paper focuses on the postprocessing component as part of the I-24 MOTION software system. The postprocessing takes the raw tracking information from the prior computer vision pipeline as input, and outputs trajectory data that is more complete and with kinematic noises and outliers removed. The postprocessing pipeline addresses two common tracking problems: the first is *fragmentation*, meaning an object (vehicle) is tracked multiple times, because tracking is disrupted due to occlusion by larger vehicles or bridges, or the vehicles move across cameras or computational boundaries, resulting in partially tracked short trajectories (or fragments); the second is the presence of *noises, outliers and missing data* on each trajectory (Fig. 3), due to miscellaneous detection
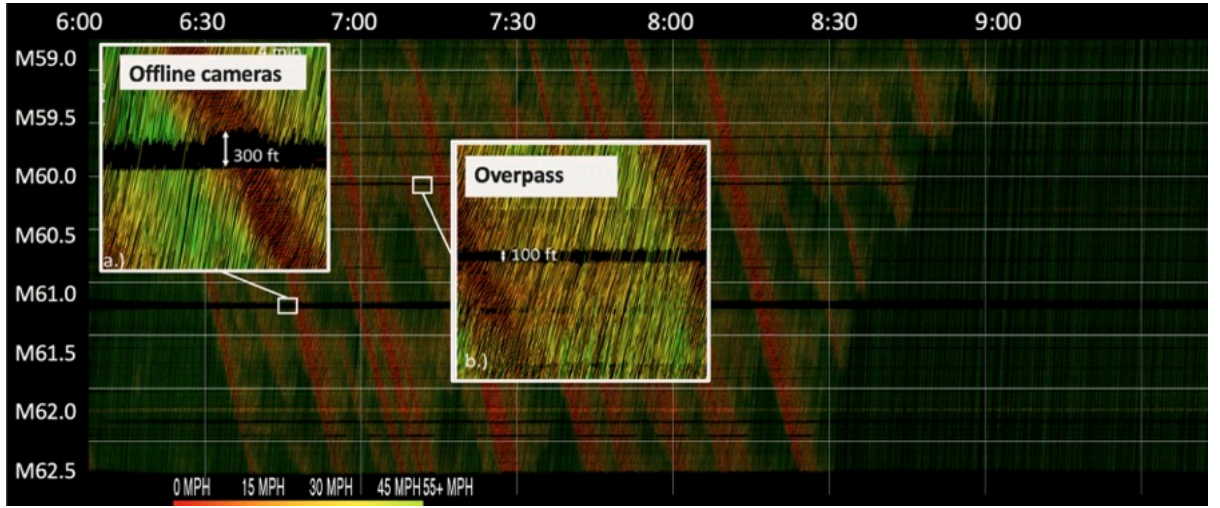
Fig. 2. Time-space diagram for four hours of I-24 westbound morning rush hour traffic on Nov 22, 2022, generated from I-24 MOTION vehicle trajectories. x-axis: time of day (HH:MM); *y*-axis roadway postmile (mi). Fragments due to artifacts from prior processes are highlighted.
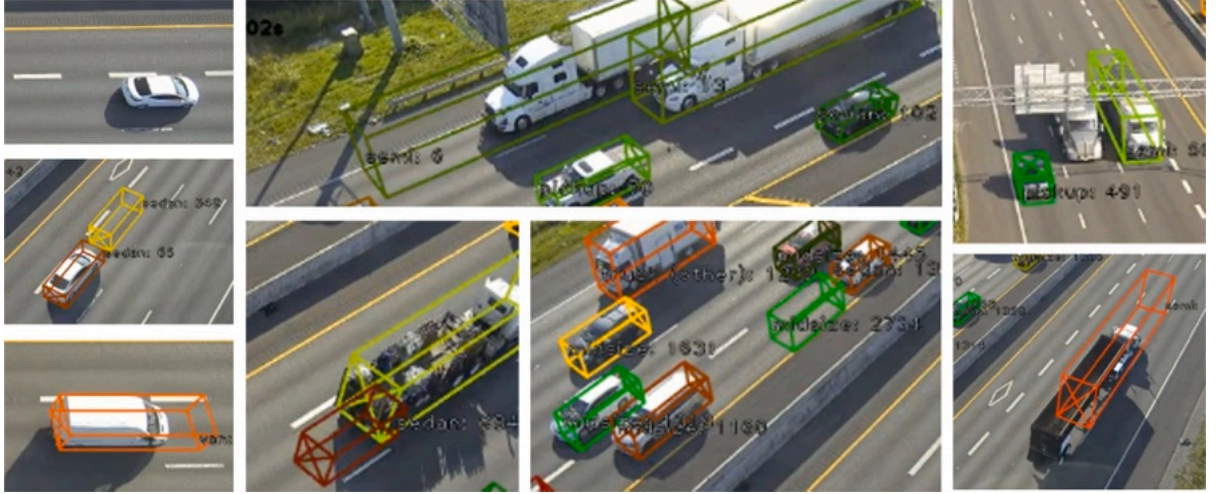

Fig. 3. Examples of common inaccuracies from prior video processing algorithms: missing detection, inaccurate localization, and false positives.

and tracking errors from preceding processes. The raw tracking input is in a 2D relative roadway coordinate system (a locally perpendicular coordinate system with the primary axis being the direction of travel and the secondary axis perpendicular to the direction of travel). The postprocessed output trajectories are in the same coordinate system.

We highlight the following contributions in this article:

- To solve the *fragmentation* issue, we develop an efficient online data association algorithm, *online negative cycle canceling*, which overcomes the limitations of batch methods for solving the data association problem as a network flow problem. The online algorithm ensures optimality of the batch algorithm, and is able to process up to half a million fragments per hour as required by the streaming data rate from the preceding tracking system. Benchmark experiments show that the proposed data association algorithm has the capability to combine together 60%–77% of fragments, resulting in trajectories that are 3–4.5 times longer.

- To address the *noises, outliers and missing data*, we introduce an one-step convex program to rectify trajectories. This method is capable of simultaneous imputation of missing data, denoising of signals, and removal of outliers. This approach, based on *elastic net regularization*, ensures kinematic feasibility and internal consistency of trajectories and is able to streamline the process compared to other multi-step methods.

- We design a software architecture for postprocessing that utilizes asynchronous processes and is currently deployed on the I-24 MOTION system to continuously generate a large volume of high-quality open-road trajectory data. The online nature of

the data association algorithm and the convex trajectory rectification formulation allow this deployment to potentially work with streaming raw tracking data from prior processes.

- We release the datasets used to benchmark the proposed postprocessing algorithms in this paper, including the ground truth, manually-perturbed and reconciled simulated data, the raw tracking data from three scenes and the corresponding manually labeled ground truth and reconciled data. Additionally, we provide the postprocessing repository used in our paper for the community to start the benchmarking efforts in the direction of trajectory data reconciliation (https://i24motion.org).

The rest of the paper is organized as follows: Section 2 summarizes available vehicle trajectory datasets, data acquisition methods and their scopes. Section 3 specifies the problem formulation for solving the fragmentation and trajectory rectification problems. In Section 4, we assess the capability of the proposed pipeline to reconstruct three benchmarking datasets: (1) a microsimulation dataset that is artificially downgraded to replicate errors from preceding processes, (2) a 15-min NGSIM data that is manually tampered, and (3) tracking data consists of 3 scenes from a subsection of the I-24 MOTION system. All of the experiments show that the postprocessed trajectories improve the accuracy on all the tested input data for a wide range of measures. Additionally we demonstrate the design of a software architecture that is capable of processing high-volume data from the full-scale I-24 MOTION system. Finally in Section 5, we summarize the methods, results and highlight potential future improvements.

2. Related work

2.1. Vehicle trajectory datasets

Vehicle trajectory data is integral to the study of traffic dynamics. They reveal the relationship between individual traffic participants and the resulting traffic flow phenomena. The increasing volume of data sources is enabling revolutionary research on, for example, traffic flow theory (Treiterer and Myers, 1974; Hoogendoorn and Knoop, 2013; Greenberg, 1959) and traffic state estimation (Seo et al., 2017; Wang and Papageorgiou, 2005; Tao et al., 2012). Trajectory data contains the microscopic fidelity (fine-grained vehicle positions) necessary for building and validating realistic microscopic models. Accurate estimation of energy consumption also relies on vehicle-level detailed dynamics (Fiori et al., 2016, 2020; Oh et al., 2020). The accurate quantification of the energy demand of vehicles is of critical importance for many applications, including the design of modern energy-centric intelligent transportation systems. Available high fidelity energy modeling tools such as Autonomie (Halbach et al., 2010), CarSim and TruckSim (Mechanical Simulation Corporation, 2014) relies on realistic acceleration and velocity dynamics for validation. High-quality trajectory data can close the gap for understanding microscopic traffic phenomena, such as lane-change and car-following (Gazis et al., 1959; Gipps, 1981; Schakel et al., 2012; Kesting and Treiber, 2008; Treiber and Kesting, 2013) and the impact of mixed autonomy in traffic (Stern et al., 2018; Gunter et al., 2021; Yang et al., 2019).

Small-scale trajectory data is usually collected via vehicles that have been equipped with high-accuracy GPS sensors (Jiang et al., 2014, 2015; Coifman et al., 2016; Huang et al., 2018) and other in-vehicle sensors. Unlike conventional probe vehicles or floating cars, vehicles currently collect their position data with significantly higher spatial and temporal resolution levels to obtain high-quality trajectory data (Guo et al., 2019). Data collected from a combination of in-vehicle sensors such as radars, LiDAR and cameras provides ambient information, which can be useful for traffic flow studies as well as for training and testing AI algorithms for autonomous driving. Readers are directed to references such as Makridis et al. (2020) and Kang et al. (2019) for a comprehensive discussion on such dataset.

Image sensors and video processing algorithms associated with overhead cameras or Unmanned Aerial Systems (UAV) have become increasingly powerful as a large-scale trajectory data source. The seminal work is the Next Generation Simulation (NGSIM) dataset (NGSIM, 2006) developed in early 2000s, which is a collection of real-world trajectories, based on the use of cameras mounted on tall buildings and covering approximately 600-meter long roadway sections with a frequency of 10 Hz in several US locations. More recently in the highD dataset (Krajewski et al., 2018), videos were recorded by overhead flying drones (4096 × 2160 pixels, 25 fps), which provided much higher-quality videos than that of the NGSIM dataset (Coifman and Li, 2017). HighD provides highway trajectories covering 420 m of road segment, for a total of 28,000 vehicle miles traveled. The pNEUMA large-scale field experiment (Barmpounakis and Geroliminis, 2020) recorded traffic streams in a multi-modal congested environment over an urban area using UAV. The dataset was generated by a swarm of 10 drones hovering over a traffic intensive area of 1.3 km^2 in the city center of Athens, Greece, covering more than 100 km-lanes of road network at 25 Hz. Zen Traffic Data (Seo et al., 2020) is obtained by cameras installed on some of the light poles of the Hanshin Expressway to observe all vehicles in target sections at a 0.1 s interval in Osaka, Japan. More recently, Automatum dataset became available (Spannaus et al., 2021), which covers 12 characteristic highway-like scenes from 30 h of drone videos. HIGH-SIM (Shi et al., 2021) is another high-quality highway trajectory dataset extracted from aerial video data. The videos were collected by three 8K cameras in a helicopter over an 8000 ft long segment of the I-75 freeway in Florida for 2 h. The dataset covers a wide range of traffic characteristics. A comparison across selected datasets can be viewed from Table 1. Other examples and discussion on their usage in traffic studies can be found in a comprehensive survey, such as Li et al. (2020).

Table 1

Video-based trajectory data comparison. VMT: vehicle miles traveled.

Dataset	Year	Context	Camera config	Road segment	VMT	Hours of recording
NGSIM (NGSIM, 2006)	2006	Highway	8 cameras	600 m	18,000	0.75
pNeuma (Barmpounakis and Geroliminis, 2020)	2018	Urban arterial	10 drones	10 km	N/A	59
highD (Krajewski et al., 2018)	2018	Highway	1 drone	420 m	28,000	147
Zen Traffic (Seo et al., 2020)	2020	Highway	6 cameras	2 km	N/A	5
Automatum (Spannaus et al., 2021)	2021	Highway	Overhead drones	N/A	18,724	30
HIGH-SIM (Shi et al., 2021)	2021	Highway	Aerial cameras	2438 m	N/A	2
I-24 MOTION (Gloudemans et al., 2023b)	2022	Highway	276 4K cameras	4.2 miles	200M/yr (expected)	daylight

2.2. Data extraction and quality assessment

Raw trajectories from video footage are often obtained from a *tracking-by-detection* framework (Kalal et al., 2012; Andriluka et al., 2008), where a set of object detections for all frames are linked across time to form continuous trajectories. The detected bounding boxes are then transformed to a common reference system for meaningful interpretation of dynamics. One of the errors we aim to reconcile is tracking discontinuity due to, for example, noisy detection, inevitable object occlusion and switching camera field of views, which results in track fragmentations. The first problem to address is data association, with is given a set of detections, identify their origin (objects that they are associated with). Data association problems can be difficult when the tracking produces large fragmentations or when the detections have false alarms or missing detections, which are common in real-world video-based detection systems.

Different approaches for data association have been proposed depending on the association criteria, complexity of object motion and the computation requirements. In general, these approaches differ from their choices of (1) matching cost (sometimes termed as probability, affinity, energy or confidence) and (2) matching criteria (Hungarian algorithm Perera et al., 2006 and bipartite graph matching Bewley et al., 2016a; Wojke et al., 2017 etc.), which result in various problem formulations. The matching cost reflects kinematic (Dicle et al., 2013; Yoon et al., 2015) and/or appearance information (Bewley et al., 2016b; Kim et al., 2015; Bae and Yoon, 2014; Geiger et al., 2013), and the matching criteria directly drives the algorithm to solve the data association problem.

Graph-based formulations provide efficient algorithms for finding the global min-cost tracking solutions. The tracks (or detections) can be naturally represented as the nodes of a graph and the pair-wise matching costs can be represented as graph edges. The general data association problem is finding the least-cost set cover on the track graph (Chong, 2012). Studies such as Castnnón and Finn (2011), Zhang et al. (2008) and Vyahhi et al. (2008) have explored efficient algorithms related to bipartite matching and min-cost flow. Interested readers are referred to a recent survey, such as Rakai et al. (2022) on this topic.

Although there has been a growing projects dedicated to generate large vehicle trajectory data, the quality issues remain insufficiently addressed. Many works (Coifman and Li, 2017; Punzo et al., 2011; Thiemann et al., 2008) raised the issue of the quality of the seminal NGSIM data, including nonphysical kinematics and inter-vehicle distance. The importance for data postprocessing is also emphasized in Kesting and Treiber (2008), Punzo et al. (2011), Treiber and Kesting (2013) and Li et al. (2020). Montanino and Punzo (2015) undertook one of the most thorough efforts of data reconciliation by applying a series of smoothing operations on the dataset. They posed the smoothing and reconstruction as a nonlinear, non-convex optimization problem, as to find the minimum local smoothing window subject to kinematic constraints. Other data postprocessing efforts (Thiemann et al., 2008; Hamdar and Mahmassani, 2008; Duret et al., 2008) also performed data reconciliation considering realistic vehicle kinematics and driving dynamics. Although with the general goal of improving overall quality of the trajectory data, these methods provide treatments for only limited types of errors, as summarized in Table 2. Most of the algorithms do not have capabilities to operate online or provide provable optimality, which prevent them from being deployed to a large-scale system with raw tracking data as streaming input. There are many practical challenges to benchmark these trajectory reconstruction methods. For example, we have not yet found a valid benchmark dataset that consists of both raw tracking results and the corresponding ground truth. In this work we demonstrate a more efficient alternative for trajectory reconciliation that is capable to treat a wide range of errors commonly seen in video processing algorithms, and process high volume of streaming data. We also release the datasets (both the simulated data and the raw tracking data with manually labeled ground truth) and the postprocessing repository used in our paper for the community to start the benchmarking efforts in this direction.

3. Problem formulation

We focus on addressing two types of issues stemming from the raw tracking data. The first is a *fragmentation*, in which an object (vehicle) is tracked multiple times, resulting in partially tracked short trajectories (or fragments). Fragments occur because tracking is disrupted due to occlusion by larger vehicles or bridges, or the vehicles move across cameras or computational boundaries (Fig. 2). It results in discontinuity in tracking and broken trajectories. The second type of errors include *noises, outliers and missing data* which

Table 2

A comparison on trajectory data reconstruction methods.

Reference	Treatments					Algorithm properties	
	Data association	Denoise	Data imputation	Outlier and false positives	Conflict resolution	Online?	Provable optimality?
Montanino (Montanino and Punzo, 2013)		✓		✓			
Montanino (Montanino and Punzo, 2015)		✓		✓	✓		
Coifman (Coifman and Li, 2017)		✓		✓	✓	Manual correction	
Punzo (Punzo et al., 2011)		✓		✓	✓		
Thiemann (Thiemann et al., 2008)		✓					
Hamdar (Hamdar and Mahmassani, 2008)		✓					
Fard (Fard et al., 2017)	✓	✓		✓			
Ji (Ji et al., 2023)		✓	✓		✓		✓
Ours	✓	✓	✓	✓		✓	✓

are caused by either a missing detection, errors in vehicle position estimation, or false positives produced by the object detector (shown in Fig. 3). Given these known errors, we propose a two-step trajectory reconstruction approach: 1. an *online negative cycle cancellation* algorithm to solve the fragment/data association problem formulated as a network flow problem, and 2. a *convex program* to rectify trajectories such that their high-order derivatives are kinematically feasible (i.e., velocities and accelerations are within a feasible range), and the gaps in between associated fragments are filled in and outliers are removed.

The methods proposed below are performed on the 2D raw trajectories in a relative roadway coordinates, assuming a fixed point on each vehicle travels in 2D planar motion. It follows a local coordinate system such that the x-axis is parallel to the lane lines, and the y-axis is perpendicular to x.

In this section, we outline the problem formulation for multi-object tracking as a minimum-cost circulation (MCC) of a graph. Solving for MCC on a track graph results in trajectory sets that have the highest *maximum a posteriori* (MAP) probability. The problem formulation is explained in literature such as Zhang et al. (2008), Wang et al. (2019, 2020) and Wang et al. (2023), and therefore only highlighted briefly in this section.

3.1. Preliminary: MCC and negative cycle cancellation

A fragment with index k is denoted as $\phi_k = \{p_1, \ldots, p_n\}$, which consists of a series of positional data ordered by time (frame). Each data point p_i is a vector containing timestamp, x and y position of a fixed point on the bounding box. We are given a set of fragments as input $\Phi = \{\phi_i\}$. A trajectory $\tau_k = \{\phi_{k_1}, \ldots, \phi_{k_n}\}$ consists of one or multiple fragments. A set of such trajectories form a trajectory set hypothesis $T = \{\tau_1, \ldots, \tau_K\}$. Assuming that fragments are conditionally independent, the fragment association step aims at finding T^*, the hypothesis with the highest MAP:

$$
\begin{aligned}
T^* &= \mathrm{argmax}_T \, P(T|\Phi) \\
&= \mathrm{argmax}_T \, P(\Phi|T)P(T) \\
&= \mathrm{argmax}_T \prod_i P(\phi_i|T) \prod_{\tau_k \in T} P(\tau_k) \\
&\text{s.t. } \tau_k \cap \tau_l = \emptyset, \ \forall k \neq l,
\end{aligned}
\tag{1}
$$

with a non-overlapping trajectory constraint, since each fragment can belong to at most one trajectory. The likelihood $P(\phi_i|T) = P(\phi_i) = \beta_i$ indicates the probability that fragment i is a false positive and thus should not be included in the trajectory hypothesis. The prior of a trajectory can be modeled as a Markov chain:

$$
P(\tau_k) = P_{enter}(\phi_{k_1}) \prod_{i=1}^{n-1} P(\phi_{k_{i+1}}|\phi_{k_i}) P_{exit}(\phi_{k_n}),
\tag{2}
$$

where $P_{enter}(\phi_{k_1})$ and $P_{exit}(\phi_{k_n})$ denote the probabilities that ϕ_{k_1} starts the trajectory and ϕ_{k_n} ends the trajectory, respectively. Taking the negative logarithm of (1), the MAP problem becomes equivalent to the following integer program:

$$
\underset{f_i, f_{i,j}, f_i^{en}, f_i^{ex}}{\text{minimize}} \sum_i c_i f_i + \sum_i c_i^{en} f_i^{en} + \sum_{i,j} c_{i,j} f_{i,j} + \sum_i c_i^{ex} f_i^{ex}
\tag{3a}
$$

$$
\text{s.t. } f_i, f_{i,j}, f_i^{en}, f_i^{ex} \in \{0, 1\},
\tag{3b}
$$

$$
f_i^{en} + \sum_j f_{j,i} = f_i = f_i^{ex} + \sum_j f_{i,j}, \ \forall i \in V \setminus \{s, t\},
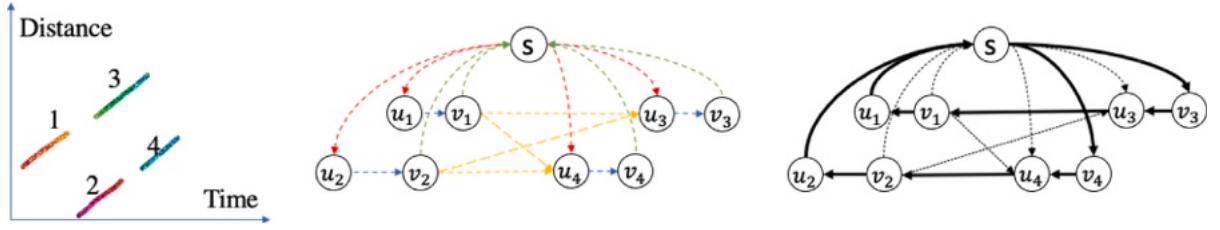\tag{3c}
$$

Fig. 4. Left: fragments in time-space coordinates. In this example the correct association is $\{\phi_1, \phi_3\}$ and $\{\phi_2, \phi_4\}$. The numbers indicate the order of last timestamp. Middle: fragments as a circulation graph. Red edges are the entering edges with cost c_i^{en}; blue edges are inclusion edges with cost c_i; green ones are exiting edges with cost c_i^{ex} and yellow ones are transition edges with cost c_{ij}. Right: the residual graph after running the negative cycle canceling algorithm to obtain the min-cost circulation. The residual edges that carry the min-cost circulation are highlighted in bold. The fragment association assignment can be obtained by tracing along the bold edges.

where

$$c_i^{en} = -\log P_{enter}(\phi_i), \;\; c_i^{ex} = -\log P_{exit}(\phi_i), \;\; c_{i,j} = -\log P(\phi_j|\phi_i), \;\; c_i = -\log \frac{1-\beta_i}{\beta_i}. \tag{4}$$

The decision variables are binary according to constraint (3b). f_i indicates whether ϕ_i should be included in any trajectory, f_i^{en} and f_i^{ex} determine whether a trajectory starts or ends with ϕ_i, respectively. $f_{i,j}$ indicates if fragment ϕ_j is an immediate successor of ϕ_i. Constraint (3c) ensures non-overlapping trajectories. The situations when an object is tracked by multiple fragments are addressed automatically in the MCC formulation: the fragments that have overlapped time intervals will not be associated into the same trajectory. The most likely association hypothesis is determined by a combination of $\beta_i, c_i^{en}, c_i^{ex}$ and $c_{i,j}$.

In seminal work (Zhang et al., 2008), it is shown that (3) has a natural graph interpretation, and solving for (3) is equivalent to solving the min-cost-flow of a tracklet graph, which has a polynomial solution (Ford and Fulkerson, 1956). Later in the work of Wang et al. (2020), it is proven that the min-cost-flow problem for MOT is equivalent to a min-cost-circulation problem on a slightly modified graph. Many efficient algorithms are developed to solve this problem (Ahuja et al., 1988; Sokkalingam et al., 2000; Klein, 1966; Goldberg and Tarjan, 1989). The graph is constructed such that each fragment ϕ_i is represented as two nodes u_i and v_i, with a directed edge $(u_i \rightarrow v_i)$ and a cost $\$(u_i \rightarrow v_i) = c_i$ indicating *inclusion* of ϕ_i; edges between two fragments ϕ_i and ϕ_j are represented as $(v_i \rightarrow u_j)$, with the cost $\$(v_i \rightarrow u_j) = c_{ij}$ related to the likelihood of ϕ_j following ϕ_i. The edge direction implies the sequential order between fragments. Furthermore, the graph has a dummy node s that has an incident edge to every u, and every v directs back to s. The resulting graph is therefore a directed circulation graph, see Fig. 4. We denote this circulation graph as $G(V, E)$, with node set V and edge set E. Each edge $e := (u, v) \in E$ has a unit capacity $r(e) = 1$, a cost $\$(e)$ and a binary flow $f(e) \in \{0, 1\}$. The data association problem can be formulated as finding a set of non-overlapping circulations f on G with the lowest total cost. The total cost of the circulations is $\sum_{e \in f} \$(e)f(e)$.

One efficient algorithm is the negative cycle canceling algorithm (NCC) proposed by Klein (1966) and later on optimized by Goldberg and Tarjan (1989) and Sokkalingam et al. (2000), based on the Ford-Fulkerson's method for incremental improvement. To understand the algorithm we first recall the definition of an important concept — a residual graph G_r:

Definition 1. The residual graph $G_r(V, E_r)$ for the original directed graph $G(V, E)$ with respect to a flow f is generated by replacing each edge $e = (u \rightarrow v) \in E$ by two residual edges $e' = (u \rightarrow v) \in E_r$ and $e_r = (v \rightarrow u) \in E_r$, with cost $\$(e') = \$(e)$ and residual capacity $r(e') = r(e) - f(e)$, while $\$(e_r) = -\$(e)$ and $r(e_r) = f(e)$.

In the context of MOT graph as shown in Fig. 4, the construction of residual graph can be simplified. The edges in the flow of the original graph simply needs to be reversed and costs on the edges negated, to form the corresponding residual graph.

The idea of NCC is to repeatedly find a cycle with negative cost in the residual graph G_r and push flow through the cycles. The algorithm terminates when no more negative cycles can be found (optimality condition). We direct interested readers the above reference for the details and proof of correctness of this algorithm, and only provide an outline in Algorithm 1.

First, a circulation graph $G(V, E)$ is constructed from the set of fragments Φ (ConstructTrackletGraph) and we iteratively look for a negative cycle in G_r based on, for example, Bellman–Ford algorithm. If such cycle exists, then update the residual graph according to Definition 1 (PushFlow). When the iteration stops (no more negative cycle can be found), the assignment, or the trajectories, can be extracted by traversing along all the cycles through the residual edges in G_r (FlowToTrajectories).

Next we show an online extension of the NCC algorithm.

3.2. Online negative cycle cancellation

The streaming data coming from I-24 MOTION testbed necessitates an online and memory-bounded version of Algorithm 1. In other words, the tracking graph G is dynamic: new fragments are added and older fragments are removed from the graph constantly. A naive online extension of Algorithm 1 is to run the same algorithm on each updated graph. However, it is inefficient because the majority of the graph remains the same and the majority of the computation on the shortest path is wasted. This opens opportunities for algorithm engineering improvements to improve performance. We show an online extension to the NCC algorithm briefly.

Algorithm 1 Negative cycle cancellation for min-cost-flow on a tracklet graph

---

Input: Set of fragments $\Phi = \{\phi_i\}$
Result: Set of trajectories $T = \{\tau_i\}$
  $G(V, E, C) \leftarrow$ ConstructCirculationGraph(Φ)
  $f \leftarrow 0$
  $G_r \leftarrow G$
while *a negative-cost cycle Γ exists in G_r* **do**
    // Update residual graph
    $G_r \leftarrow$ PushFlow(G_r, Γ)
end
$T \leftarrow$ FlowToTrajectories(G_r)

The proposed online algorithm is based on the assumption that fragments are added to the graph in the order of last timestamp, which is a reasonable assumption in practice as this is the order that fragments are generated from object tracking. The online algorithm proceeds by adding each fragment ϕ_k to the residual graph from the previous iteration $G_{r,k-1}^+$ one at a time, to obtain a new graph $G_{r,k}^-$ (AddNode($G_{r,k-1}^-, \phi_k$)). This step adds two nodes u_k and v_k to the graph along with edges $(s \to u_k), (u_k \to v_k), (v_k \to s)$ and possibly additional transition edges incident to u_k. Then, we search for the least-cost negative cycle Γ in $G_{r,k}^-$ (FindMinCycle($G_{r,k}^-$)) and push flow through the cycle to obtain the updated residual graph $G_{r,k}^+$. When all the fragments are processed, we output the trajectories T by tracing all the cycles in the final residual graph. It can be proved that pushing flow through Γ, $G_{r,k}^+$ contains the min-cost circulation because the flow is feasible and no further negative cycles can be found in G_r^+. We denote the residual graph after adding ϕ_k at iteration k to be $G_{r,k}^+$. The algorithm is shown in Algorithm 2.

Algorithm 2 Online NCC for MCC on a tracklet graph

Input: Set of fragments $\Phi = \{\phi_i\}$
Result: Set of trajectories $T = \{\tau_i\}$
  $f \leftarrow 0$
  $G_{r,0}^+ \leftarrow \{s\}$
  $k \leftarrow 1$
for *each ϕ_k (ordered by last timestamp)* **do**
    $G_{r,k}^- \leftarrow$ AddNode($G_{r,k-1}^+, \phi_k$)
    $\Gamma \leftarrow$ FindMinCycle($G_{r,k}^-$)
    $G_{r,k}^+ \leftarrow$ PushFlow($G_{r,k}^-, \Gamma$)
    $k \leftarrow k + 1$
end
$T \leftarrow$ FlowToTrajectories($G_{r,k}^+$)

The online algorithm is proved to be correct. To summarize, we proved that the circulation in $G_{r,k}^+$ is always optimal, i.e., there is no more negative cycles in $G_{r,k}^+$ for every k (optimality condition). Details on the proof can be found in Wang et al. (2023).

3.3. Trajectory rectification

After applying fragment association, the next step is to rectify the stitched, raw trajectories with denoising, imputation and smoothing operations. Instead of ad-hoc correcting each source of the detection errors shown in Fig. 3, we treat all noises and errors in a one-step approach to rectify them all at once. To simplify the problem, we consider the fixed-point positions of a trajectory, and the dimension of each vehicle is taken as the median from the bounding box trajectories. That way, the projected footprints of each vehicle in the roadway coordinate do not change shapes. We consider a 2D vehicle motion model, with independent longitudinal (x) and lateral (y) dynamics. This allows us to decompose the problem to solving two independent 1D reconciliation problems.

We use the rear bottom-center of the 3D bounding boxes as the points of interest. Each trajectory τ_i includes the following features (features are the same for all trajectories, therefore the index i is dropped for simplicity): let $\mathbf{x} = [x[1], x[2], \ldots, x[N]]^T$ be the x-positions for N timesteps, and similarly let $\mathbf{y}$ be the time-series of the y-position; $\mathbf{v}_x = [v_x[1], v_x[2], \ldots, v_x[N-1]]^T$ is the time-series of speed in the longitudinal direction (x-axis), and similarly $\mathbf{v}_y$ is the time-series of speed in the lateral component (y-axis). The acceleration $\mathbf{a}$ and jerk $\mathbf{j}$ use the same 2D representation. Lastly, l, w represent the vehicle length and width, respectively. The vehicle dynamics model can be seen in Fig. 5.

Consider the following discrete-time 3rd order 1D motion model in either longitudinal or lateral direction (the same method is applied for x-component and y-component dynamics. We demonstrate on x-component only):

$$x[t+1] = x[t] + v[t]\Delta T$$
$$v[t+1] = v[t] + a[t]\Delta T \tag{5}$$
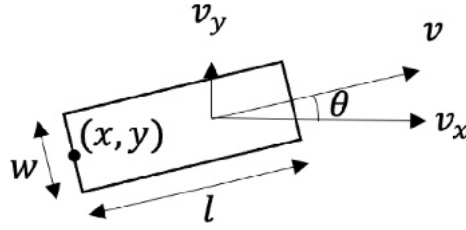$$a[t+1] = a[t] + j[t]\Delta T,$$

Fig. 5. A simple 2D vehicle model.

with ΔT as the timestep. Notice that the finite-difference method decrements the dimension of time-series as an increase of derivative order, i.e., $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{v} \in \mathbb{R}^{N-1}$, $\mathbf{a} \in \mathbb{R}^{N-2}$ and $\mathbf{j} \in \mathbb{R}^{N-3}$. Eq. (5) can be written in matrix multiplication form:

$$
\begin{aligned}
\mathbf{v} &= D^{(1)}\mathbf{x} \\
\mathbf{a} &= D^{(2)}\mathbf{x} \\
\mathbf{j} &= D^{(3)}\mathbf{x},
\end{aligned}
\tag{6}
$$

where $D^{(k)} \in \mathbb{R}^{(N-k)\times N}$ represents the kth-order differentiation operator. For example, $k = 1, 2$, and 3 can be written as:

$$
D^{(1)} = \frac{1}{\Delta T}
\begin{bmatrix}
-1 & 1 & 0 & \dots & 0 & 0 & 0 \\
0 & -1 & 1 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & -1 & 1 & 0 \\
0 & 0 & 0 & \dots & 0 & -1 & 1
\end{bmatrix}.
$$

$$
D^{(2)} = \frac{1}{\Delta T^2}
\begin{bmatrix}
1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & 1 & -2 & 1
\end{bmatrix}
$$

$$
D^{(3)} = \frac{1}{\Delta T^3}
\begin{bmatrix}
-1 & 3 & -3 & 1 & \dots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & -1 & 3 & 3 & 1
\end{bmatrix}.
$$

Consider corrupted measurement $\mathbf{z}$ which contains missing data (indicated by the observation operator H), noises $\mathbf{w}$ and outliers $\mathbf{e}$:

$$
\mathbf{z} = H\mathbf{x} + \mathbf{w} + \mathbf{e}, \quad \mathbf{z} \in \mathbb{R}^M, \mathbf{w} \in \mathbb{R}^M, \mathbf{e} \in \mathbb{R}^M, H \in \mathbb{R}^{M\times N},
\tag{7}
$$

and missing data exists if $M < N$. We aim to find the reconstructed position $\hat{\mathbf{x}}$ that is smooth in kth-order derivatives. The idea is to use a combination of Ridge and Lasso regression (Zou and Hastie, 2005), to simultaneously handle noises and outliers, assuming outliers are sparse and noises have small magnitude:

$$
\underset{\mathbf{x}, \mathbf{e}}{\text{minimize}} \ \|\mathbf{z} - H\mathbf{x} - \mathbf{e}\|_2^2 + \sum_{k=2}^{k=K} \lambda_k \|D^{(k)}\mathbf{x}\|_2^2 + \lambda_1 \|\mathbf{e}\|_1
\tag{8a}
$$

$$
\text{subject to} \ -D^{(1)}\mathbf{x} \preceq 0
\tag{8b}
$$

$$
\lfloor \mathbf{x} \rfloor^{(k)} \preceq D^{(k)}\mathbf{x} \preceq \lceil \mathbf{x} \rceil^{(k)}, k = 2, 3, \dots, K.
\tag{8c}
$$

The first term of the cost function (8a) penalizes the data-fitting error on the non-missing entries. The second term regularize the smoothness of the position vector, by penalizing the l_2-norm of higher-order derivatives (e.g., $k = 2$ and $k = 3$ correspond to accelerations and jerks in respective). The third term regularizes the sparsity of the outliers. The first constraint (8b) states that the speed has to be non-negative, i.e., no cars are traveling backward at any time. The second constraint (8c) sets the upper and lower bound for each high-order derivatives. For example, $\lceil \mathbf{x} \rceil^{(2)}$ is the largest possible acceleration. Note that (8a) can be written into a quadratic programming form by converting the l_1 penalization to a linear programming with linear inequality constraints (Chen et al., 2001; Boyd et al., 2004). The problem can be solved with a convex programming solver such as `cvxopt` (Vandenberghe, 2010). Note that above formulation rectifies the trajectory of each vehicle independently of another. After treatment (8), further investigation is needed to determine if, for example, considering vehicular interactions in the optimization formulation (e.g., with additional non-collision constraint) is required.

Solving this convex program produces derivative quantities that are internally consistent, i.e., numerical differentiation of the position vector gives the speed vector, and Euler forward integration of the speed vector produces the same position vector. The same internal consistency is satisfied between other derivative orders.

Table 3
Summary of benchmark experiments. MOT: multi-object-tracking.

Experiment	Ground truth data	Raw tracking data	Spatial size	Temporal size	# Cameras	Metrics/Goal
Exp1	TransModeler simulation	artificially downgraded from ground truth	2000 ft	15 min	3	MOT metrics Kinematic distributions
Exp2	NGSIM I-101	artificially downgraded from ground truth	2200 ft	13.3 min	7	MOT metrics Kinematic distributions
Exp3	Manually labeled I24-3D	tracking output from videos	2000 ft	3 min	16–17	MOT metrics Kinematic distributions
Exp4	No available ground truth	tracking output from videos	4.5 mi	4 h	276	Scalability test

The 2D dynamics can be obtained by solving two independent optimization problems of form (8) for both longitudinal and lateral dynamics. The lateral movement follows the same discrete-time dynamics as (5). Let the solution for solving (8) for the longitudinal movement be $(\hat{\mathbf{x}}, \hat{\mathbf{e}}_x)$, and the solution for the lateral movement be $(\hat{\mathbf{y}}, \hat{\mathbf{e}}_y)$. The rectified steering angle $\hat{\theta} = [\theta[1], \ldots, \theta[N-1]]^T$ can be calculated in the end as

$$\hat{\theta} = \tan^{-1}\left(\frac{D^{(1)}\hat{\mathbf{y}}}{D^{(1)}\hat{\mathbf{x}}}\right). \tag{9}$$

4. Experiments

In this section, we present the results of four experiments designed to evaluate the performance of our proposed 2-step data reconciliation pipeline:

- Experiment 1 (Exp1): In this numerical experiment, we utilize a microsimulation dataset as the ground truth. Raw tracking data is generated by intentionally degrading the ground truth to simulate errors from preceding processes. The objective is to assess the 2-step data reconciliation pipeline's ability to reconstruct the original ground truth data from this downgraded version.
- Experiment 2 (Exp2): This experiment employs a 15-minute NGSIM I-101 dataset as the ground truth. The raw tracking data is produced by manually introducing perturbations to the ground truth. The aim is to evaluate how effectively our 2-step data reconciliation pipeline can recover the ground truth in this scenario.
- Experiment 3 (Exp3): Real tracking data from 16–17 cameras recorded on a section of the I-24 MOTION system serves as the input for this experiment. The data captures three distinct traffic conditions. We use manually labeled 3D vehicle bounding boxes for each scene (I24-3D Gloudemans et al., 2023a) as the ground truth to validate our algorithm's performance.
- Experiment 4 (Exp4): This experiment assesses the scalability of our algorithms. The input is a 4-hour raw tracking data from the full-scale I-24 MOTION system, which includes 276 cameras covering a 4.2-mile highway segment.

A summary of these experiments is provided in Table 3.

It is important to note that these experiments serve as an initial step towards refining the algorithms. This includes identifying optimal parameters and cost models, among others. Rather than focusing on optimizing parameters to remove all errors in a given dataset, the results provide preliminary evidence that the methods are capable of reducing errors across a range of datasets.

4.1. Evaluation metrics

Vehicles tracking and trajectory reconciliation performance can be evaluated with standard *multi-object-tracking* (MOT) metrics, specified in Bernardin and Stiefelhagen (2008), Milan et al. (2016), Li et al. (2009) and Ristani et al. (2016). For all the following metrics, an intersection-over-union (IOU) of vehicle footprint with ground truth of 0.3 or higher is required to be considered as a true positive.

- *Precision*: number of detected objects over sum of detected and false positives (target: 1).
- *Recall*: number of detections over number of objects (target: 1).
- *Sw/GT*: total number of track switches per ground truth trajectory (target: 0).
- *Fgmt/GT*: total number of fragments (switches from tracked to not tracked) per ground truth trajectory (target: 0).
- *Multi-object-tracking accuracy* (MOTA): an aggregated measure to indicate tracking performance. It is detection errors (false negatives and false positives) and fragmentations normalized by the total number of true detections (target: 1).
- *Multi-object-tracking precision* (MOTP): the total error in estimated position for matched prediction and ground truth pairs over all time, averaged by the total number of matches made (target: 1).

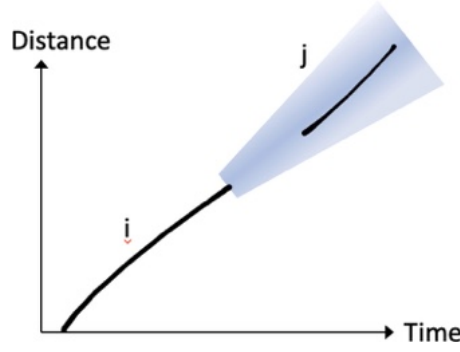Other statistics are computed to qualitatively assess the trajectories:

Fig. 6. Probability of fragment ϕ_i after the last measurement of ϕ_i is represented as a cone. The matching cost of ϕ_i and ϕ_j is determined by the negative log likelihood of ϕ_j given the cone probability.

- Trajectory lengths distribution
- Speed distribution (calculated using finite-difference)
- Acceleration distribution (calculated using finite-difference)

4.2. Implementation details

In this section, we describe the computation of edge costs for min-cost flow and the choice of parameters λ_i's for solving the trajectory rectification problem (8).

The probabilities of a trajectory starts or ends with a fragment P_{enter} and P_{exit} is determined by the ratio of the true number of trajectories and the predicted number of trajectories. The fragment linking costs $P(\phi_i|\phi_j)$ can be modeled by (a combination of) dynamics, shape affinity and the time interval between the two fragments. We implemented a version that considers the dynamics only, as it is shown to properly represent the matching probabilities.

The general idea is, if ϕ_j is indeed a continuation of ϕ_i but the object tracking is broken due to, e.g., object occlusion, then the projected position of ϕ_i at the time of ϕ_j has a high chance of overlapping with the actual detection of ϕ_j. The probability of track i at the time of ϕ_j follows a stochastic process, with the mean $\hat{p}_i$ being the projected position of track i given ϕ_i, and the variance increases with respect to Δt, the time elapsed since the end of ϕ_i, t_e^i, i.e.,

$$p_i(t_e^i + \Delta t) = \hat{p}_i(t_e^i + \Delta t) + \eta_i(\Delta t), \text{ and } \eta_i(\Delta t) \sim \mathcal{N}(0, \alpha + \beta \Delta t). \tag{10}$$

The uncertainty $\eta_i(\Delta t)$ follows a Brownian-like process, with zero-mean and variance $\alpha + \beta \Delta t$ growing linearly with Δt (see Fig. 6). The projected mean of position is:

$$\hat{p}_i(t) = v_i t + \bar{p}_i, \tag{11}$$

where $v_i = [v_{x,i}, v_{y,i}]^T$ and $\bar{p}_i = [x_i, y_i]^T$ can be determined using, for example, linear regression. Finally, the matching cost is the negative log likelihood of ϕ_j given the probabilities computed from ϕ_i:

$$\Lambda(\phi_i, \phi_j) = \frac{1}{2N_j} \sum_{t_j \in [t_s^j, t_e^j]} \log\left(\alpha + \beta(t_j - t_e^i)\right) + \frac{1}{2N_j} \sum_{t_j \in [t_s^j, t_e^j]} \frac{\left(p_j(t_j) - \hat{p}_i(t_j)\right)^2}{\alpha + \beta(t_j - t_e^i)}, \tag{12}$$

where N_j is the number of measurements of ϕ_j. t_s^j and t_e^j are the start and end timestamp of ϕ_j, and $p_j(t_j)$ is the measurement of ϕ_j at t_j.

After fragment association, the next step is to impute missing data, correct for outliers, and smooth the trajectories in a single step by solving an optimization problem (Eq. (8)). To balance the terms in the cost function, we conducted a grid search to determine the optimal parameters λ_i. We found that setting K up to 3, where the λ_2 term penalizes large accelerations and λ_3 penalizes large jerks, was sufficient. Empirical assessment of a subset of trajectories led us to pre-tune the parameters to $\lambda_1 = 1.2 \times 10^{-3}$, $\lambda_2 = 1.67 \times 10^{-2}$, and $\lambda_3 = 1.67 \times 10^{-7}$. The acceleration and jerk constraints were set to ± 10 ft/sec^2 and ± 10 ft/sec^3, respectively.

The parameter tuning is subject to the implicit assumption of Euler forward discretization, with the current sampling frequency of 25 Hz. We demonstrate the capacity the above choice of parameters to bring close the speed and acceleration to a more realistic range, but note that a detailed parameter optimization that generalizes to a range of datasets is part of our ongoing effort to operationalize the described methods.

Table 4
Simulation experiment results.

Metrics/Statistics		SIM-GT	SIM-RAW	SIM-REC
Precision ↑		1	0.81	0.98 (+21.0%)
Recall ↑		1	0.76	0.90 (+18.4%)
MOTA ↑		1	0.58	0.88 (+51.7%)
MOTP ↑		1	0.80	0.92 (+15.0%)
Fgmt/GT ↓		0	4.66	0.36 (−92.3%)
Sw/GT ↓		0	0	0.03 (+NA%)
No. trajectories		989	5597	1316
Trajectory lengths	min	61.8	42.5	72.0
(ft)	max	2001	802	2003
	avg	1976	327.0	1440
	stdev	14.0	225.2	77.1
Speed (ft/s)	min	0.00	0.00	0.00
	max	130.4	294.8	130.3
	avg	57.8	64.1	60.8
	stdev	33.1	42.6	32.1
Acceleration	min	−6.93	−7608	−6.93
(ft/s^2)	max	6.93	7446	6.94
	avg	0.78	0.96	0.83
	stdev	2.80	1591	2.84

4.3. Exp1: Numerical experiments with microsimulation data

The proposed 2-step data reconciliation pipeline is benchmarked using a micro-simulation dataset that showcases various traffic characteristics. In the absence of "ground truth" tracking data of sufficiently large scale to test our algorithms, numerical experiments with simulation data serve as a fundamental first step to refine algorithms and identify optimal parameters. The simulation dataset used in this experiment is the ground truth (dubbed as SIM-GT) and offers a broad range of traffic scenarios, including freeflow and congested traffic with various lane-change rates. In addition, this dataset is artificially perturbed to replicate the major errors observed in real-world tracking input data, such as noise, masks, and overlapped fields of view from multiple cameras. The perturbed data is referred to as SIM-RAW. The corresponding reconciled output after executing the 2-step postprocessing pipeline is termed as SIM-REC. To evaluate the accuracy of the proposed method, we compare SIM-RAW and SIM-REC with the ground truth SIM-GT using standard MOT metrics along with other statistics related to traffic dynamics, as mentioned in Section 4.1.

4.3.1. Simulation description

The simulation dataset is generated using TransModeler, a micro-simulation software that allows customization of highway traffic characteristics such as traffic density, lane blockage, car-following and lane-changing behaviors. TransModeler utilizes an extended GM model (Ahmed, 1999) that closely mimics real-world car-following behaviors. The dataset encompasses a 4-lane highway segment spanning 2000 ft, with a simulation duration of 900 s (as depicted in Fig. 7(a)). Notably, this simulation includes a lane blockage around the 800-ft mark, which persists for 400 s. Traffic downstream of the blockage and subsequent to its removal flows freely, while an induced congestion emerges upstream of the bottleneck. Additional details on the simulation dataset can be found in Appendix A. A summary is also provided in Table 4.

The simulation dataset has been intentionally polluted (we refer to the polluted data as SIM-RAW, see Fig. 7(b)) to mimic the inaccuracies observed in the I-24 MOTION system, which arise from factors such as overpasses, misaligned cameras, and offline cameras. This introduced perturbation encompasses several elements, including spatial and temporal masks, overlapping trajectory segments, and measurement point noise.

Specifically, the SIM-RAW dataset is generated from the ground truth dataset (SIM-GT) by incorporating a mask that spans the entire simulation timeframe and covers all trajectories within the distance range of 1550–1700 ft. This mask represents an overhead bridge measuring 150 ft wide, leading to tracking disruptions for vehicles passing beneath it. Another mask is placed upstream within the distance range of 100–300 ft, where vehicles are decelerating, entering a bottleneck, and frequently changing lanes. Several additional masks, each 50 ft wide, are introduced to simulate periods of camera package loss in various locations. This results in objects changing their IDs before and after the masks.

Furthermore, we introduce a 100-ft-wide overlap in trajectories at two specific locations, occurring at distances of 600 ft and 1200 ft. These overlaps are representative of two fields of view from three cameras that intersect. The perturbation procedure also includes the introduction of random noises and outliers.

All RAW datasets have two locations (at 700 ft and 1400 ft) with overlapped regions of adjacent cameras. Vehicles change IDs when they travel across cameras, resulting in fragmentation. The fragments also overlap for about 100 ft, representing inaccuracies due to camera misalignment or homography transformation. Lastly, white noises are added to all the fragments. The resulting trajectory length, speed, and accelerations are summarized in Table 4.

Table 4 summarizes the MOT metrics for the polluted dataset. The raw data has a detection precision of 0.81, recall of 0.76, and 4.66 fragments per ground truth vehicle trajectory. We also calculate the distributions of trajectory derivative quantities as additional statistics. The polluted dataset (SIM-RAW) shows a wide spread in these metrics due to the added noises on the position.
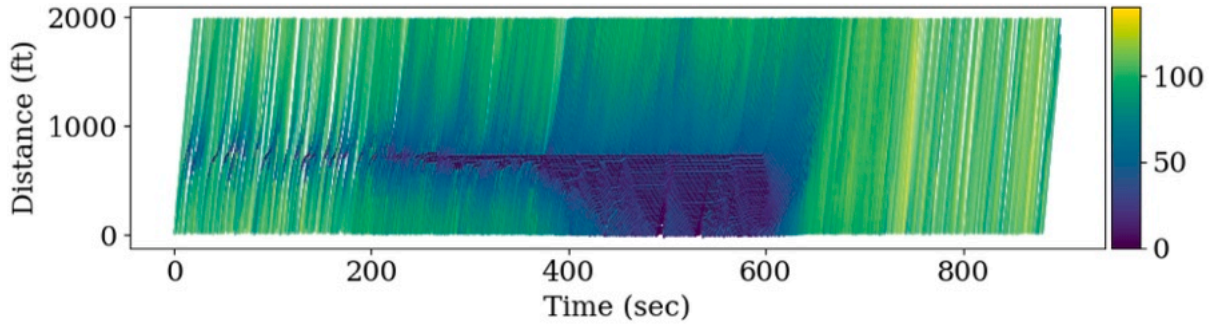
(a) Ground truth simulation (SIM-GT).
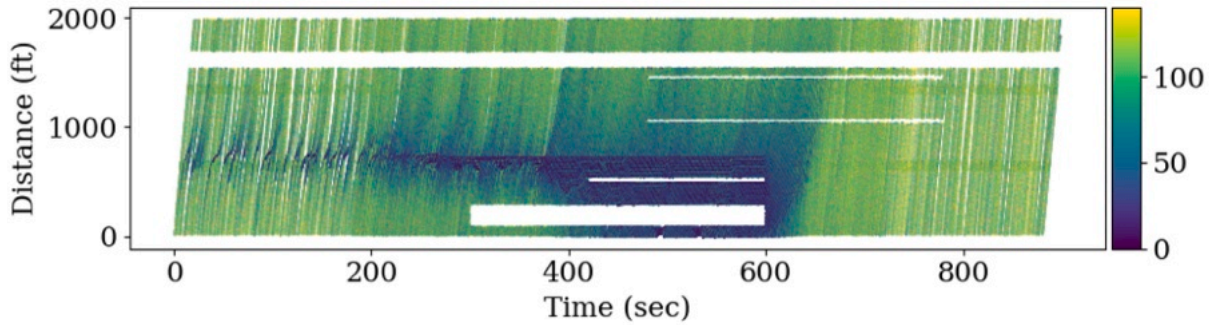
(b) Manually polluted simulation data (SIM-RAW) including noises, overlapped and non-overlapped fragments.
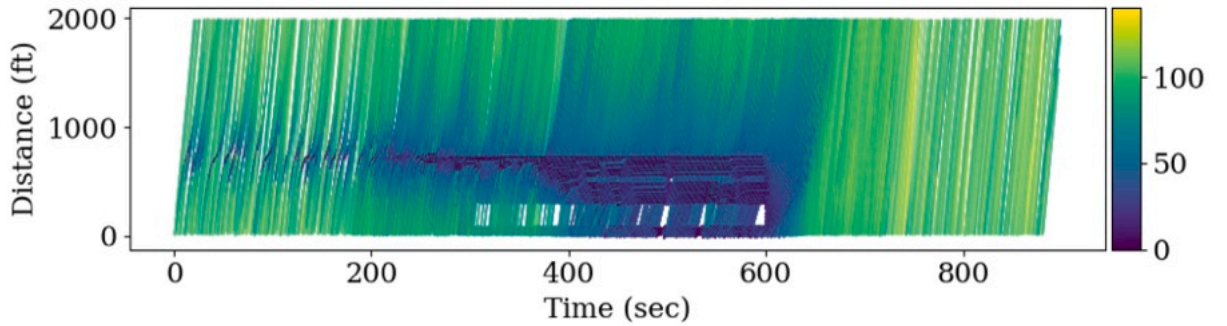
(c) Reconstructed trajectory data (SIM-REC).

Fig. 7. Time-space diagrams for SIM-GT, SIM-RAW and SIM-REC.

4.3.2. Results

The evaluation results are obtained by comparing SIM-RAW and the corresponding reconciled trajectories SIM-REC against the same ground truth SIM-GT (Fig. 7). The results are summarized in Tables 4. The table indicates that regardless of the traffic scenarios, the 2-step reconciliation output improves a wide range of metrics.

The proposed two-step data reconciliation approach enhances precision and recall by approximately 18%–21% when compared to the raw tracking data. This improvement is achieved by correctly associating fragments and imputing missing data caused by masks, resulting in an increase in true positives and a decrease in false negatives. False positives are reduced by merging overlapped fragments during the data-association step. Furthermore, noise is smoothed during the trajectory rectification step (8), which reduces false positives. As a result, the aggregated score MOTA improves, while MOTP primarily improves due to the smoothing effect of the trajectory rectification step.

The effectiveness of the data association is particularly evident in the higher-speed regions (downstream of the bottleneck and after the lane-blockage is removed), where all fragments are correctly matched together. Overall the majority of the fragments are correctly associated (92.3% improvement in Fgmt/GT), a few incorrectly-associated fragments lead to ID switches (i.e., when one

predicted trajectory maps to multiple ground truth trajectories). In total, we counted 30 ID switches in SIM-REC and identified two reasons for their occurrence: (1) a change in speed occluded by the mask, and/or (2) an abrupt lane-change maneuver occluded by the mask. Several illustrative examples of ID switches are shown in Fig. 8.

Each plot displays a single predicted trajectory from SIM-REC that mapped to two ground truth trajectories, along with the corresponding raw fragments from SIM-RAW. The left sides show time vs. longitudinal (x) positions of all trajectories, and the right sides are time vs. lateral (y) positions. The matched tracking trajectory (SIM-REC) is shown in an orange dashed line, while the two ground truth trajectories (in SIM-GT) are depicted in black and gray solid lines. All the fragments (in SIM-RAW) corresponding to those two ground truth trajectories are displayed in colored points, with each fragment represented by a distinct color. Fig. 8(a) is an example where an ID switch occurs due to an unobserved speedup at the masked region (longitudinal position 100–300 ft). Before entering the mask, both vehicles 84e8 and 84b9 traveled on the same lane at approximately 25.5 ft/s with 84e8 following closely behind 84b9. Tracking for both vehicles was discontinued for 8 s due to the mask, and when they reappeared at position 300 ft, the speed for both vehicles had increased to 42.8 ft/s. Our current cost function (Eq. (12)) assumes linear longitudinal and lateral dynamics, and thus favors matching two fragments that appear to be on a straight line in the time-space dimension. Consequently, fragment 84e8 has a lower cost to be matched with fragment 84ba. Matching becomes even more challenging when both longitudinal and lateral dynamics change during the occlusion, as shown in Fig. 8. In such cases, the lateral dynamics tend to dominate the association cost when fragments are "close" together in time vs. x dimension, causing the data association to fail because it prefers to match fragments that are in the same lane.

These examples highlight the difficulty of accurate motion modeling, particularly during non-trivial maneuvers such as speed changes and lane changes in long occlusions. This difficulty motivates the search for other representations of "similarity" in matching cost design that do not solely rely on motion information. One potential solution is to use appearance embedding from the preceding object detector. Another option is to train a classifier-like similarity measure model that learns the correct matching function from labeled data. For future work, we could explore a general data-driven approach outlined in Mathisen et al. (2020).

Moreover, we compute the statistics on the dynamical information of the trajectories as some unsupervised measures of data quality when ground truth labels are not present. Specifically, we analyze the statistical characteristics of trajectory dynamics. Ideally, the length distribution of trajectories should cover the entire road segment (excluding ramps or trajectories at the time boundaries). The speed distribution should reflect the traffic dynamics, including free flow, stop-and-go, and congestion, and the acceleration distribution should be within a reasonable range. We calculate speed and acceleration by taking the finite difference of positions and speed, respectively. A comparison of the traffic dynamics distributions for ground truth, polluted, and postprocessed data is presented in Table 4. Generally, SIM-REC generates longer trajectories due to the fragment association step. Notably, the regularization terms in (8) of data reconciliation contribute to a significantly narrower (or more consistent) and feasible acceleration distribution. Table 4 suggests that the statistics for speed and acceleration of the reconstructed data SIM-REC are much closer to the ground truth than those of the polluted datasets.

4.4. Exp2: Experiments on the NGSIM I-101 data

Similar to Exp1, in this experiment we run our algorithms on the manually polluted NGSIM data, and benchmarked it against the ground truth NGSIM data (NGSIM, 2006). The manual perturbation includes masks of widths between 30 and 50 ft, noises and outliers on position measurements. The metrics for ground truth, perturbed and reconstructed data are summarized in Table 5, and the time-space diagrams are shown in Fig. 9.

We observe that compared to NGSIM-RAW, NGSIM-REC improves on all the metrics. Particularly, 83% of the fragments are reduced. We notice a similar result as compared to the previous experiment: the accuracy of data association is particularly high when the traffic is at a higher speed. Whereas when the speed variation is occluded by the masks, it is much difficult to match fragments that are interrupted by the masks. Additionally, we observe the speed and acceleration distributions of the reconstructed data are closer to the ground truth.

4.5. Exp3: Experiments on the I-24 MOTION validation system

Next, we demonstrate the application of the proposed pipeline on the I-24 MOTION validation system, which was built in 2020 and functioned as a prototype for the design selections in the full system. The validation system consists of three poles that host 18 cameras to seamlessly cover 2000-ft highway segment. The detailed description of the design can be found in Gloudemans et al. (2023b).

4.5.1. Dataset description

All evaluations have been conducted using three datasets sourced from the I-24 MOTION validation system (I24-3D) (Gloudemans et al., 2023a), each describes a distinct traffic scenario. The ground truth datasets comprise a total of 877,000 manually-labeled 3D bounding boxes of vehicles, derived from 57 min of video data collected across 16–17 cameras. Notably, this dataset stands as the most extensive "ground truth" dataset directly obtained from the I-24 MOTION system's cameras. It captures real-world traffic scenes and includes raw tracking results produced by the system's current tracker. This data presents real-world challenges originating from preceding sources, including issues such as fragmentations due to occlusions, object handoffs between multiple cameras, projection errors when transitioning from camera image space to roadway coordinates, false positives arising from tracking and detection errors, and more.
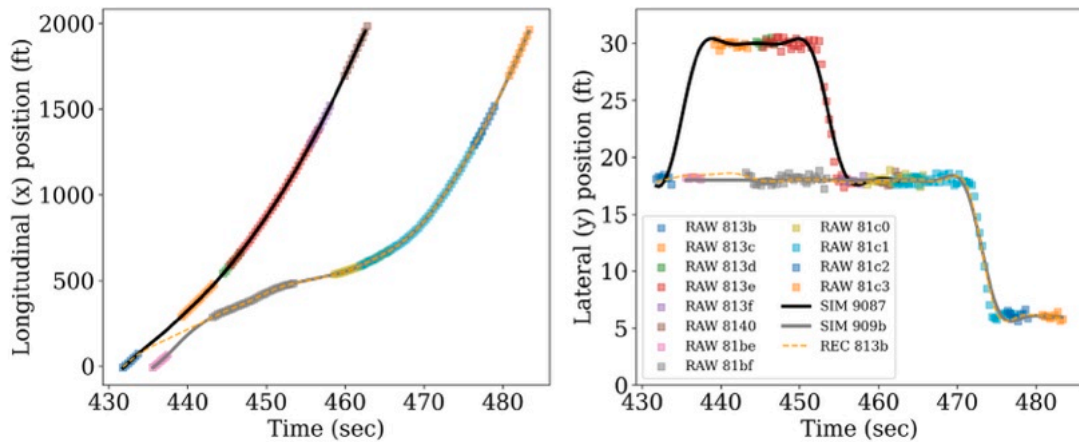
The three distinct scenes in the I24-3D include:
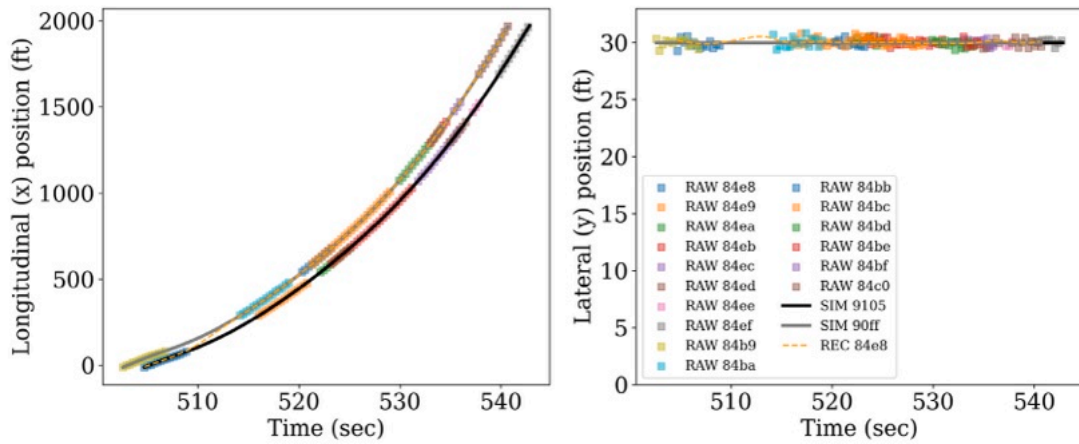
(a) ID switch example 1: occluded speedup, no lane change
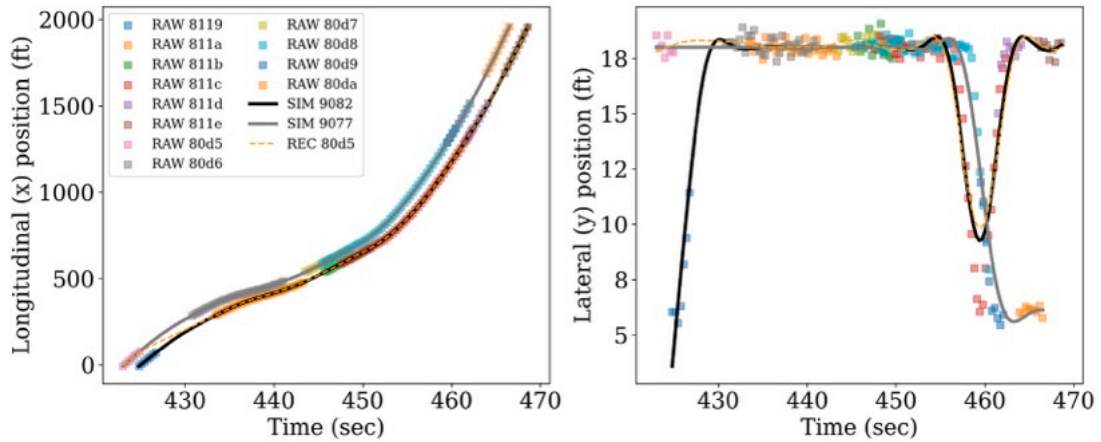
(b) ID switch example 2: occluded slowdown and lane-change

(c) ID switch example 3: occluded slowdown and lane-change

Fig. 8. Examples of ID switches.

Table 5

NGSIM experiment results.

Metrics/Statistics		NGSIM-GT	NGSIM-RAW	NGSIM-REC
Precision ↑		1	0.48	0.69 (+43.8%)
Recall ↑		1	0.46	0.67 (+45.7%)
MOTA ↑		1	0.81	0.86 (+6.17%)
MOTP ↑		1	0.84	0.90 (+7.14%)
Fgmt/GT ↓		0	8.93	1.52 (−83.0%)
Sw/GT ↓		0	2.69	0.96 (−64.3%)
No. trajectories		1533	13 685	2335
Trajectory lengths	min	877.7	3.90	17.7
(ft)	max	2173.4	1049.8	2172.9
	avg	2037.6	517.9	1030.6
	stdev	176.4	322.8	605.6
Speed (ft/s)	min	0.029	7.11	0.00
	max	81.6	158.9	81.5
	avg	29.8	29.94	29.9
	stdev	13.9	15.8	14.0
Acceleration	min	−6.82	−1784.4	−6.91
(ft/s^2)	max	6.80	2060.8	6.94
	avg	0.113	0.132	0.11
	stdev	1.82	265.8	2.06

Table 6

Exp2: Evaluation results using 3 manually-labeled ground truth datasets (Gloudemans et al., 2023a).

| Metrics		GT-i	RAW-i	REC-i	GT-ii	RAW-ii	REC-ii	GT-iii	RAW-iii	REC-iii
Precision ↑		1	0.71	0.90	1	0.87	0.88	1	0.76	0.76
Recall ↑		1	0.56	0.83	1	0.55	0.79	1	0.46	0.67
MOTA ↑		1	0.32	0.74	1	0.48	0.68	1	0.31	0.46
MOTP ↑		1	0.63	0.73	1	0.72	0.75	1	0.67	0.68
Fgmt/GT ↓		0	5.22	0.60	0	5.38	1.93	0	4.92	1.11
Sw/GT ↓		0	1.43	0.04	0	2.98	0.53	0	3.01	0.52
No. trajs		314	789	321	100	411	150	253	1250	282
Trajectory	min	25.6	24.0	17.6	36.9	5.1	5.1	5.0	0.01	0.45
lengths (ft)	max	2270.5	2096.5	2094.7	2183.8	568.9	2181.2	2278.1	2066.9	2289.0
	avg	1635.4	507.8	1455.1	1013.5	147.9	588.2	1042.0	157.8	721.2
	stdev	639.0	419.4	711.6	587.9	126.8	512.2	621.5	231.8	649.1
Speed	min	72.3	0.97	72.7	16.7	0.01	8.7	0.00	0.00	0.00
(ft/s)	max	147.7	437.8	142.9	62.0	276.4	66.36	136.3	235.2	141.2
	avg	106.0	106.4	105.9	39.4	39.1	9.3	37.6	40.6	37.7
	stdev	10.9	13.8	10.5	7.24	11.6	7.40	33.9	35.5	33.4
Acceleration	min	−5.6	−54155	−6.9	−6.58	−20965	−6.94	−6.90	−19233	−6.94
(ft/s^2)	max	4.95	55701.0	6.7	6.86	25017.1	6.94	6.90	54174	6.93
	avg	−0.005	−9.76	0.12	−0.04	39.2	−0.04	−0.157	9.1	−0.159
	stdev	1.29	1891.3	1.46	1.01	675.8	1.48	2.16	1002.3	2.21

- A 60-second free-flow traffic scenario.
- A 51-second slow traffic in snowy conditions.
- A 50-second scene of heavily congested traffic with stop-and-go waves.

The raw tracking data was obtained using a crop-based fast tracking algorithm described in Gloudemans and Work (2021). The raw tracking results obtained from the three video recordings are denoted as RAW-i, RAW-ii, and RAW-iii. The corresponding ground truth datasets are labeled as GT-i, GT-ii, and GT-iii, respectively. The output generated by our 2-step reconciliation pipeline is designated as REC-i, REC-ii, and REC-iii.

4.5.2. Results

The performance of the pipeline on the three tracking datasets is presented in Table 6. The results show that the proposed data reconciliation pipeline improves on all the MOT metrics for all three scenes. Noticeably the pipeline is able to match the 60%–77% of fragments. The data association step enhances precision and recall by accurately connecting fragments and filling in the gaps between the detections.

To view the performance in more details, Figs. 10–12 provide the time-space diagram for each lane colored by true positives (blue), false positives (red), and false negatives (yellow). The results show comparison between the raw tracking (RAWs) and the ground truth (GTs), as well as the postprocessed (RECs) and GTs. Specifically, we show that in all three scenes, the false negatives are significantly reduced due to the data association and the data imputation in between the associated fragments. The false negatives
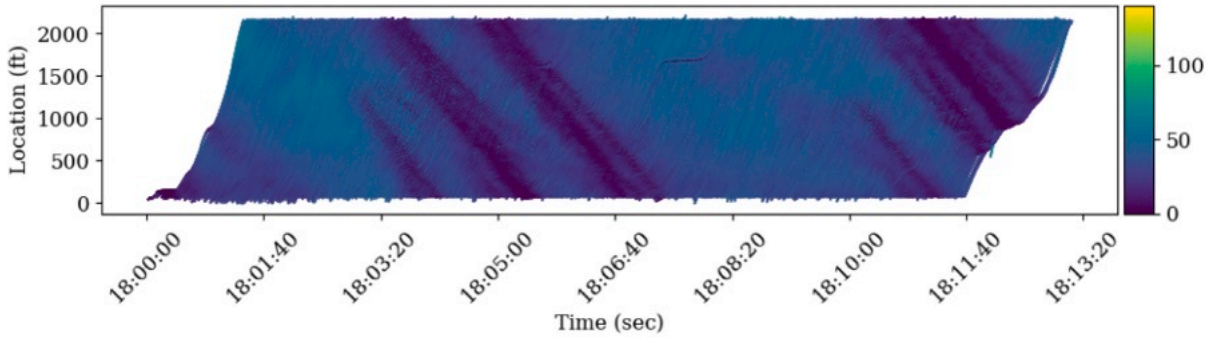
(a) Ground truth NGSIM-GT.
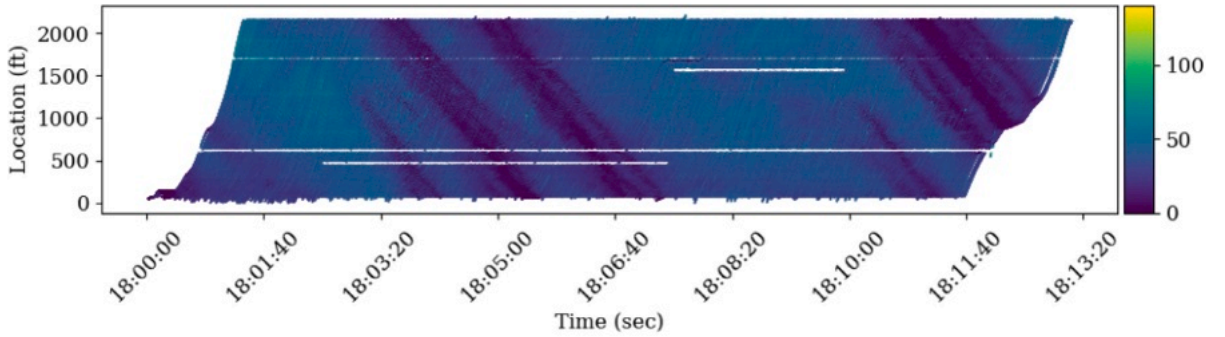
(b) Manually polluted NGSIM-RAW including noises and masks.
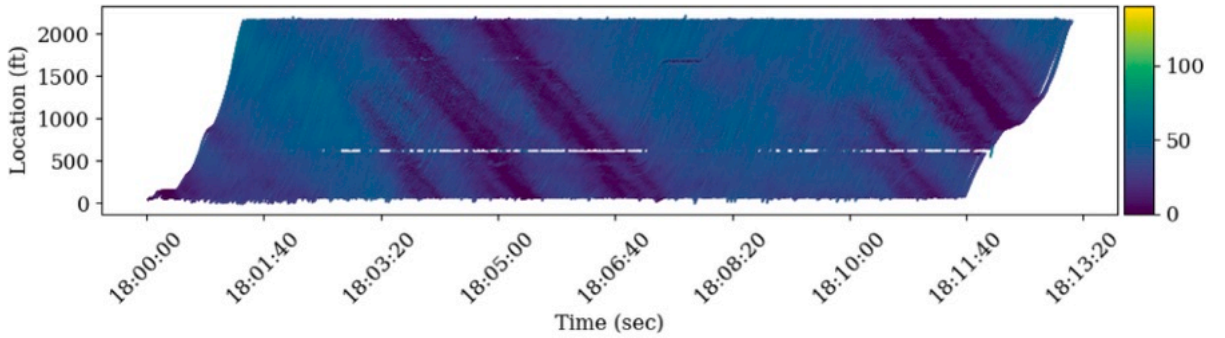
(c) Reconstructed trajectory data NGSIM-REC.

Fig. 9. Time-space diagrams for NGSIM-GT, NGSIM-RAW and NGSIM-REC.

at the boundaries remain after postprocessing because there is currently no mechanism to extrapolate trajectories to the boundaries of the time window. A few false positives are corrected automatically during the data association stage, and others are identified as outliers and are removed at the trajectory rectification stage. Most significantly, we show that most of the "choppy" fragments in RAW-iii westbound (Fig. 12(b)) are correctly associated together, leading to more complete trajectories. The fragmentation rate reduced from 4.92 to 1.11 Fgmt/GT. However, postprocessing is likely to reintroduce false positives if two fragments are matched incorrectly (causing ID switching). This error tend to occur when a vehicle undergoes speed variations and the tracking is fragmented, as the case in REC-iii westbound. The overall tracking and postprocessing accuracy is the highest in REC-i (freeflow), as the vehicle dynamics are the simplest. Overall, the benchmark experiments demonstrate a promising result that the postprocessed trajectories resolved the majority of the fragmentation issues, and significantly reduced false negatives.

Next we show the distributions of trajectory lengths, velocities and accelerations for all three scenes in Fig. 13. The data association step directly helps to produce longer trajectories, and the trajectory rectification step leads to a more realistic range of speed and acceleration, as compared to the raw tracking data which contains noises.
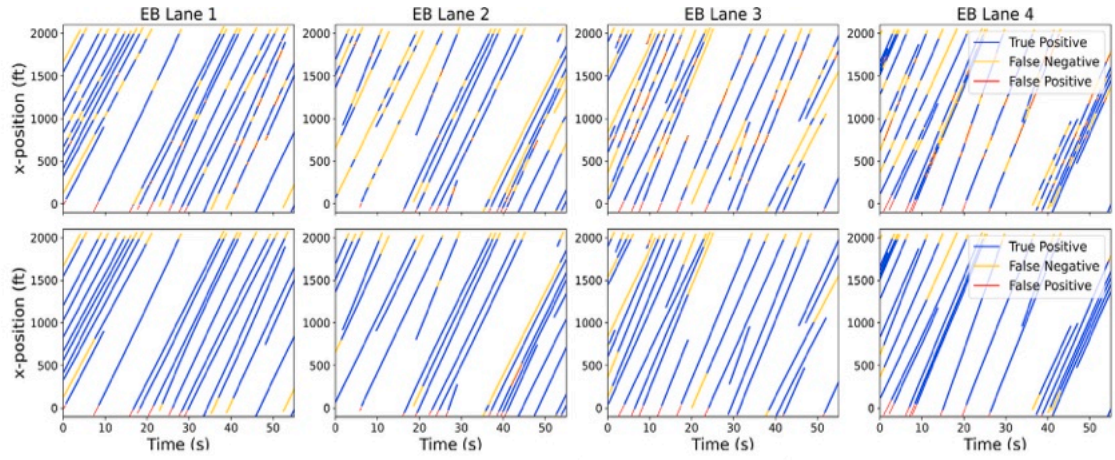
(a) Lane-specific time-space diagrams for eastbound RAW-i (top) and REC-i (bottom); lane 4 is rightmost lane in direction of travel. False negatives (yellow), false positives (red) and true positives (blue) are shown. False negatives indicate missing data due to disrupted tracking, and false positives indicate that an object was tracked below the IOU threshold.
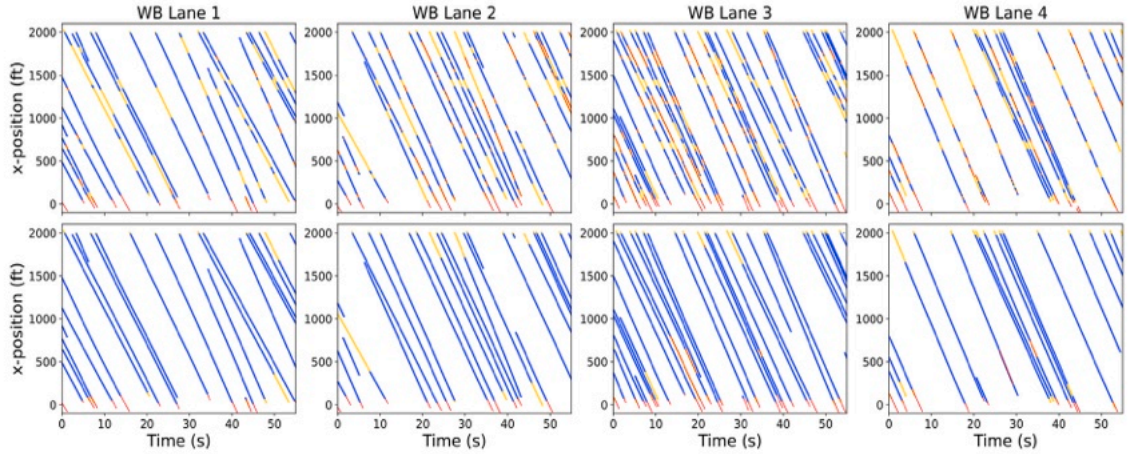
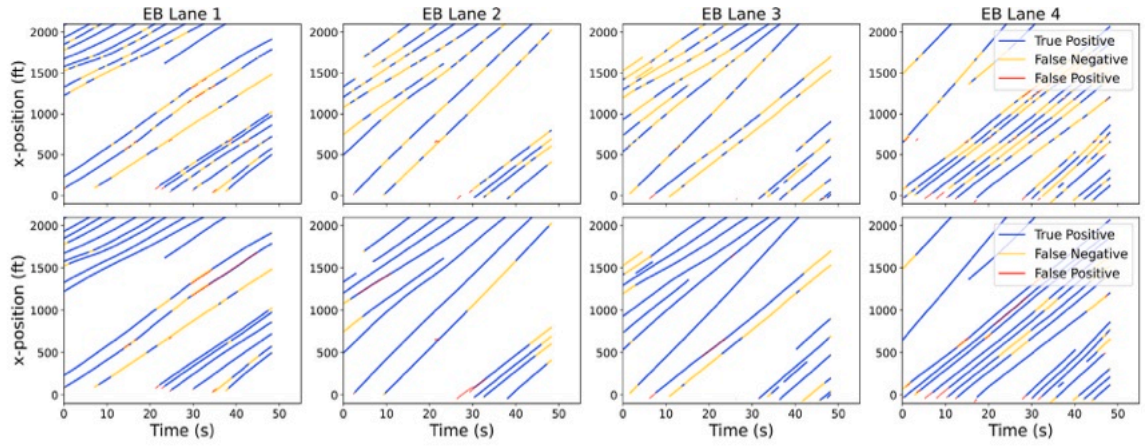(b) Lane-specific time-space diagrams for westbound RAW-i (top) and REC-i (bottom).

Fig. 10. Summary of evaluation for the first scene in I24-3D (compare RAW-i and REC-i against GT-i).

Table 7

A comparison of data postprocessing procedures. Ours: data association with the matching cost specified in (12), followed by trajectory rectification; SD: data association with the simple distance (SD) metric as the matching cost, followed by trajectory rectification; Reversed: trajectory rectification first, then data association.

Metrics	RAW-i	REC-i			RAW-ii	REC-ii			RAW-iii	REC-iii		
	–	Ours	SD	Reversed	–	Ours	SD	Reversed	–	Ours	SD	Reversed
Precision ↑	0.71	**0.90**	0.58	0.87	0.87	**0.88**	0.81	0.84	0.76	**0.76**	0.71	0.76
Recall ↑	0.56	**0.83**	0.79	0.81	0.55	**0.79**	0.47	0.78	0.46	**0.67**	0.47	0.65
MOTA ↑	0.32	**0.74**	0.40	0.69	0.48	**0.68**	0.36	0.65	0.31	**0.46**	0.28	0.44
MOTP ↑	0.63	**0.73**	0.67	0.70	0.72	**0.75**	0.73	0.74	0.67	**0.68**	0.67	0.67
Fgmt/GT ↓	5.22	**0.60**	1.97	4.28	5.38	**1.93**	3.55	2.65	4.92	**1.11**	3.32	5.28
Sw/GT ↓	1.43	**0.04**	1.35	0.11	2.98	**0.53**	2.75	0.55	3.01	**0.52**	3.05	0.60

Lastly, we provide an additional experiment (Table 7) to show the impact of the cost function $\Lambda(\phi_i, \phi_j)$ and data processing order on the reconstruction accuracy. Specifically, we compared our method with a simple distance (SD) metric as an alternative matching cost. The SD metric is specified as $\|p_i(t_e^i) - p_j(t_s^j)\|_2$, i.e., the Euclidean distance between the last measurement of fragment i and the first measurement of fragment j. Additionally, we compared our 2-step procedure with the steps reversed, i.e., first run

(a) Lane-specific time-space diagrams for eastbound RAW-ii (top) and REC-ii (bottom); lane 4 is rightmost lane in direction of travel. False negatives (yellow), false positives (red) and true positives (blue) are shown.
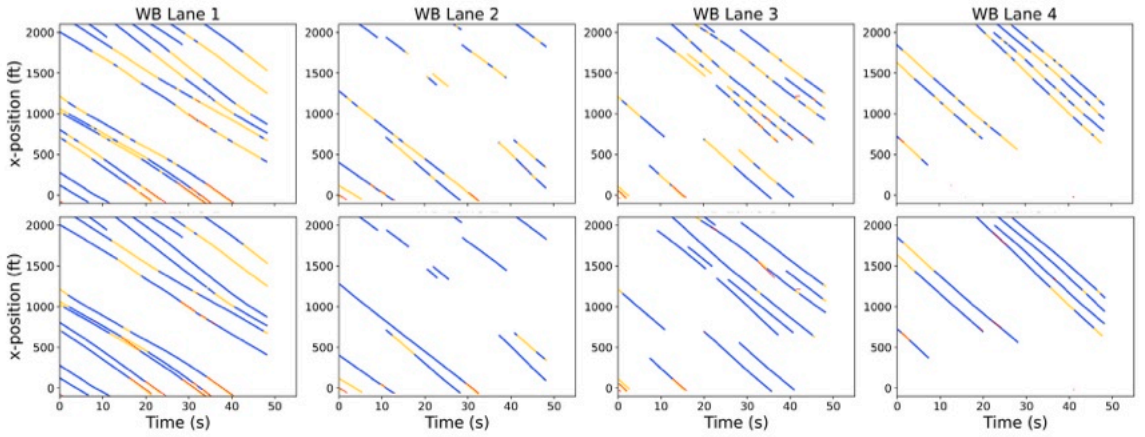
(b) Lane-specific time-space diagrams for westbound RAW-ii (top) and REC-ii (bottom).

**Fig. 11.** Summary of evaluation for the second scene in I24-3D (compare RAW-ii and REC-ii against GT-ii).

trajectory rectification followed by data association ("Reversed"). The results show that our original 2-step method with (12) as the matching cost performs the best.

In summary, the postprocessing improves all metrics on all the tested datasets. Particularly it produces longer trajectories with feasible dynamics. However, the most significant challenge occurs when associating fragments in dense and slow traffic, where the complex speed and lane-change behavior is not adequately captured by the current data association cost model. In the benchmark datasets released with this paper, we provide the origin of the fragments associated to each trajectory. Details can be found on the project website https://i24motion.org.

4.6. Exp4: Scalability test on the complete I-24 MOTION system

The complete I-24 MOTION testbed (Gloudemans et al., 2023b) consists of 276 cameras in total that seamlessly cover 8 lanes (4.2 miles) of I-24 segment near Nashville, TN. On a typical workday morning, the recorded raw tracking data spans 4 h, from 6:00 AM to 10:00 AM, making it 16 times larger in temporal scale, and 22 times larger in spatial scale than the NGSIM I-101 data. The dataset contains about 1.2 million tracklets (fragments). It is crucial to design an architecture that is scalable for size of the testbed.

To handle the volume of tracking data from this testbed, we design a postprocessing software architecture that employs parallel and asynchronous compute processes (see Fig. 14). This architecture utilizes multiple processes running concurrently, managed by a master scheduler, to handle tracking data from nine video processing nodes, with each process assigned to independently process fragments detected from 25–31 cameras. Each process performs local data association independently, and the locally processed results are passed to the master process of the corresponding direction of travel, which runs a second pass of data association
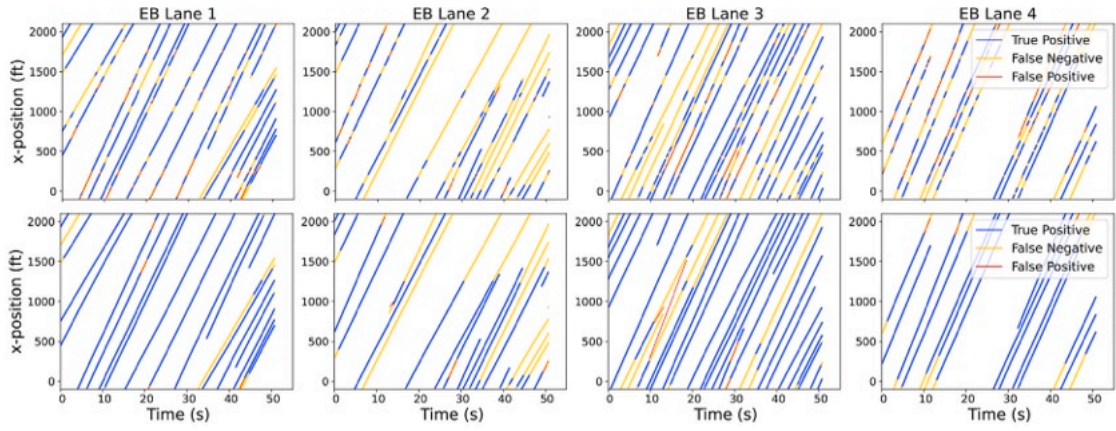
(a) Lane-specific time-space diagrams for eastbound RAW-iii (top) and REC-iii (bottom); lane 4 is rightmost lane in direction of travel. False negatives (yellow), false positives (red) and true positives (blue) are shown.
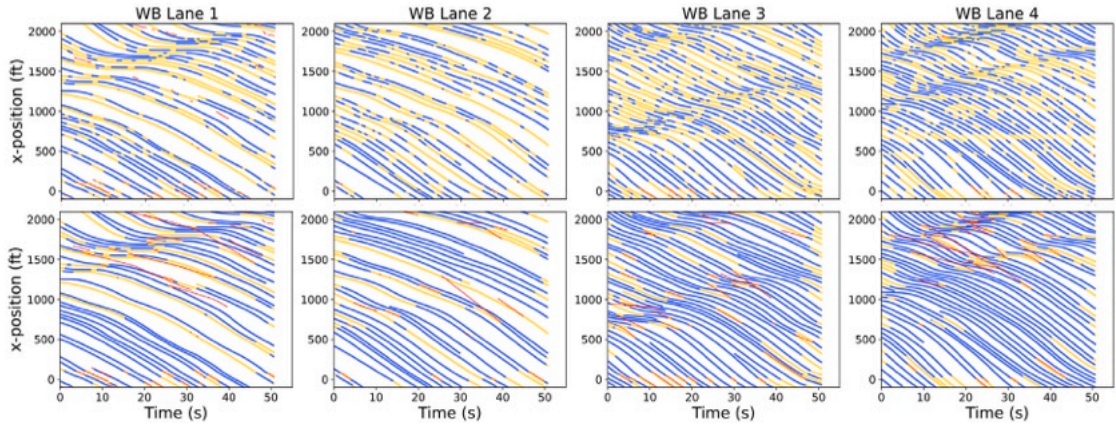
(b) Lane-specific time-space diagrams for westbound RAW-iii (top) and REC-iii (bottom).

Fig. 12. Summary of evaluation for the third scene in I24-3D (compare RAW-iii and REC-iii against GT-iii).

to connect the partial trajectory fragments across each adjacent local process. Finally, all resulting trajectories are smoothed and imputed in the "reconciliation module" before being written to the postprocessed database.

Our experiments demonstrate that this software architecture can effectively handle the volume of tracking data from this testbed. In addition to implementing an online version of the negative cycle cancellation algorithm mentioned in Section 3.2, our approach can process approximately 400 trajectories per minute for light traffic (flow is about 30 vehicles/lane/min), while for heavy traffic (120 vehicles/lane/min) it can process one minute of data in 50 s. For a full-scale 4-hour run with the asynchronous software structure illustrated in Fig. 14, local processes are completed in 1.5 h, and master processes in 2 h, resulting in a total runtime of 3.5 h to process 4 h of data. The server that hosts this architecture operates on Ubuntu 20.04, with 528 GB total memory and 64 CPU cores. These results indicate that our proposed algorithm is scalable and can efficiently process large volumes of real-time tracking data on I-24 MOTION.

5. Conclusion and future work

High-quality trajectory data can close the gap for understanding microscopic traffic phenomena. A real-world live testbed like I-24 MOTION helps researchers to understand the impact of mixed autonomy in traffic. However, data produced by cameras and prior computer vision algorithms still lacks high quality to be ready for research use. In this paper we demonstrate a two-step data postprocessing pipeline to automatically reconcile detection and tracking data. The pipeline includes a fragment association algorithm to solve an online min-cost flow problem, and a trajectory rectification approach formulated as a quadratic programming. The accuracy is benchmarked on both numerical experiments using miscrosimulation, NGSIM data, as well as on the raw tracking
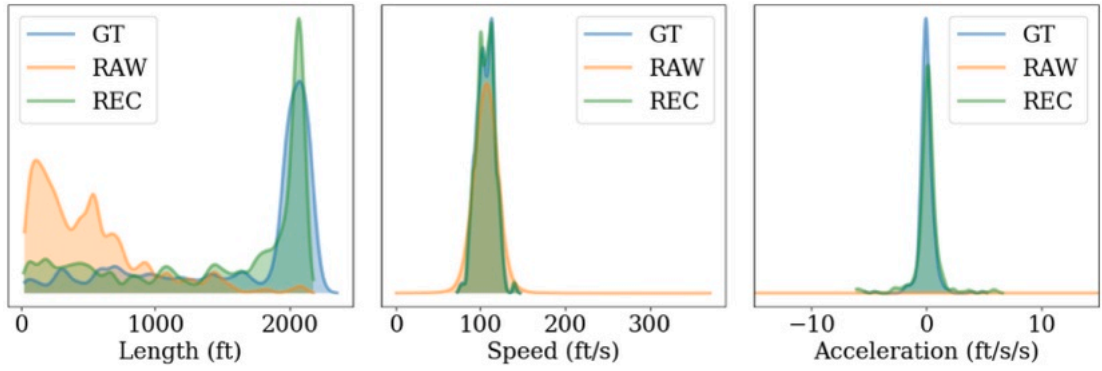
(a) Scene i (free flow).
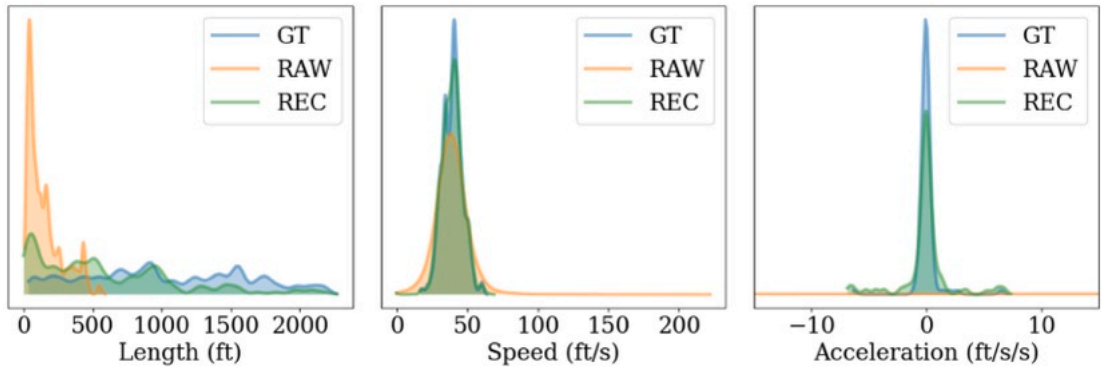
(b) Scene ii (snow weather and slow traffic).
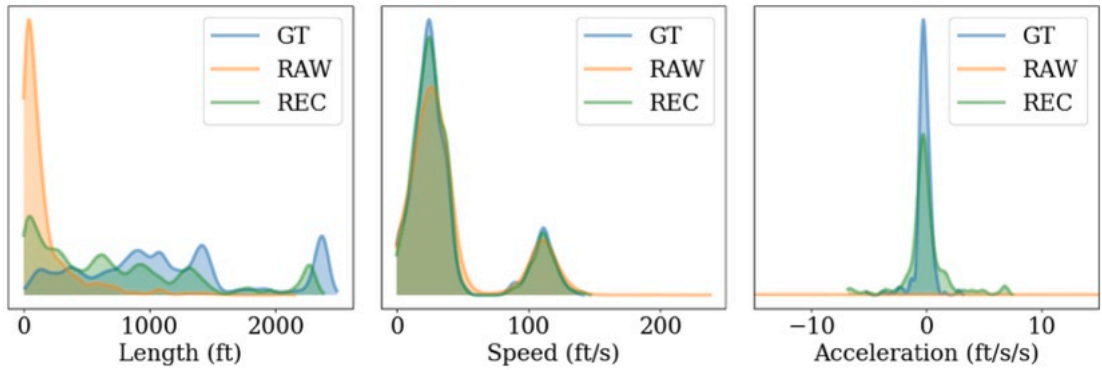
(c) Scene iii (congested traffic).

Fig. 13. Distributions of trajectory lengths (left), speed (middle) and acceleration (right) for GT (blue), RAW (orange) and REC (green) for all three scenes in I24-3D.

data from three scenes of the I-24 MOTION system and the corresponding manually labeled ground truth. Results show that the two-step treatments improve a variety of trajectory quality measures on all the testing cases, given different traffic scenarios. Noticeably, velocities and accelerations are within a feasible range, despite the parameter tuning and cost function design step are only preliminary. This proposed pipeline has high promises to replace previous manual efforts on data cleaning. Additionally, we show a design of a software architecture for postprocessing that utilizes asynchronous processes and is currently deployed on the I-24 MOTION system to continuously generate a large volume of high-quality open-road trajectory data.
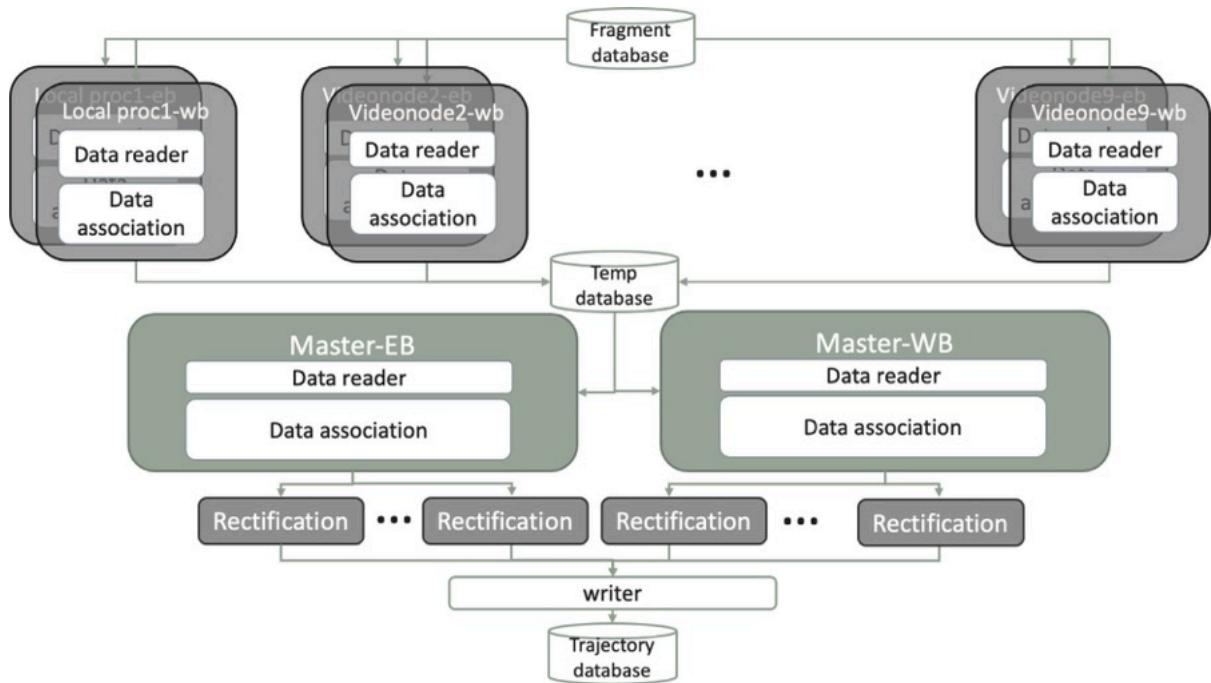
Fig. 14. An overview of the postprocessing system diagram. The architecture includes several parallel processes performing local data association in each direction (EB: east bound, WB: west bound). The results are then passed on to the master processes, which perform association across adjacent road segments. The final step involves the trajectory rectification module, which imputes and smooths all associated fragments and writes the results to the database.

For future work, a few algorithmic improvements can be made on existing methods to potentially address issues in real data. For example, a multi-vehicle reconciliation formulation can be considered to account for potential collisions. Furthermore, different cost models for data association and an exhaustive parameter tuning need to be performed to improve the current results.

CRediT authorship contribution statement

Yanbing Wang: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Derek Gloudemans:** Data curation, Methodology, Software, Writing – review & editing. **Junyi Ji:** Data curation, Software, Visualization, Writing – review & editing. **Zi Nean Teoh:** Data curation, Software, Validation, Visualization. **Lisa Liu:** Data curation, Software, Validation, Visualization. **Gergely Zachár:** Data curation, Software, Writing – review & editing. **William Barbour:** Data curation, Project administration, Resources, Writing – review & editing. **Daniel Work:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing.

Acknowledgments

Appendix A. Synthetic data

The synthetic data used to test the postprocessing pipeline was generated using TransModeler 6.1, a micro-simulation software. The simulation involved a one-way, 2000 ft-long four-lane highway, with a capacity of 2000 $veh/hr/lane$, and lasted 15 min (900 s). To evaluate the algorithms' performance under varying traffic conditions, the simulation incorporated time-varying traffic demand and a bottleneck. The traffic demand varied between 1200 $veh/hr/lane$ in 1–3 min and 13–15 min, 2400 $veh/hr/lane$ in 4–6 min and 10–12 min, and 3600 $veh/hr/lane$ in 6–9 min. The bottleneck was triggered by a lane closure signal, with lane-1 (leftmost) closed for the first 10 min. Fig. A.15 depicts the trajectory space–time diagram for the synthetic data. The lane-changing logic in TransModeler involves a decision process with multiple rules, including the target lane rule and the gap acceptance rule (Yang and Koutsopoulos, 1996). The target lane rule determines which lane to change to, while the gap acceptance rule determines whether the gap on the targeted lane is acceptable to the vehicle. The lane-changing action is triggered when both rules are satisfied. It
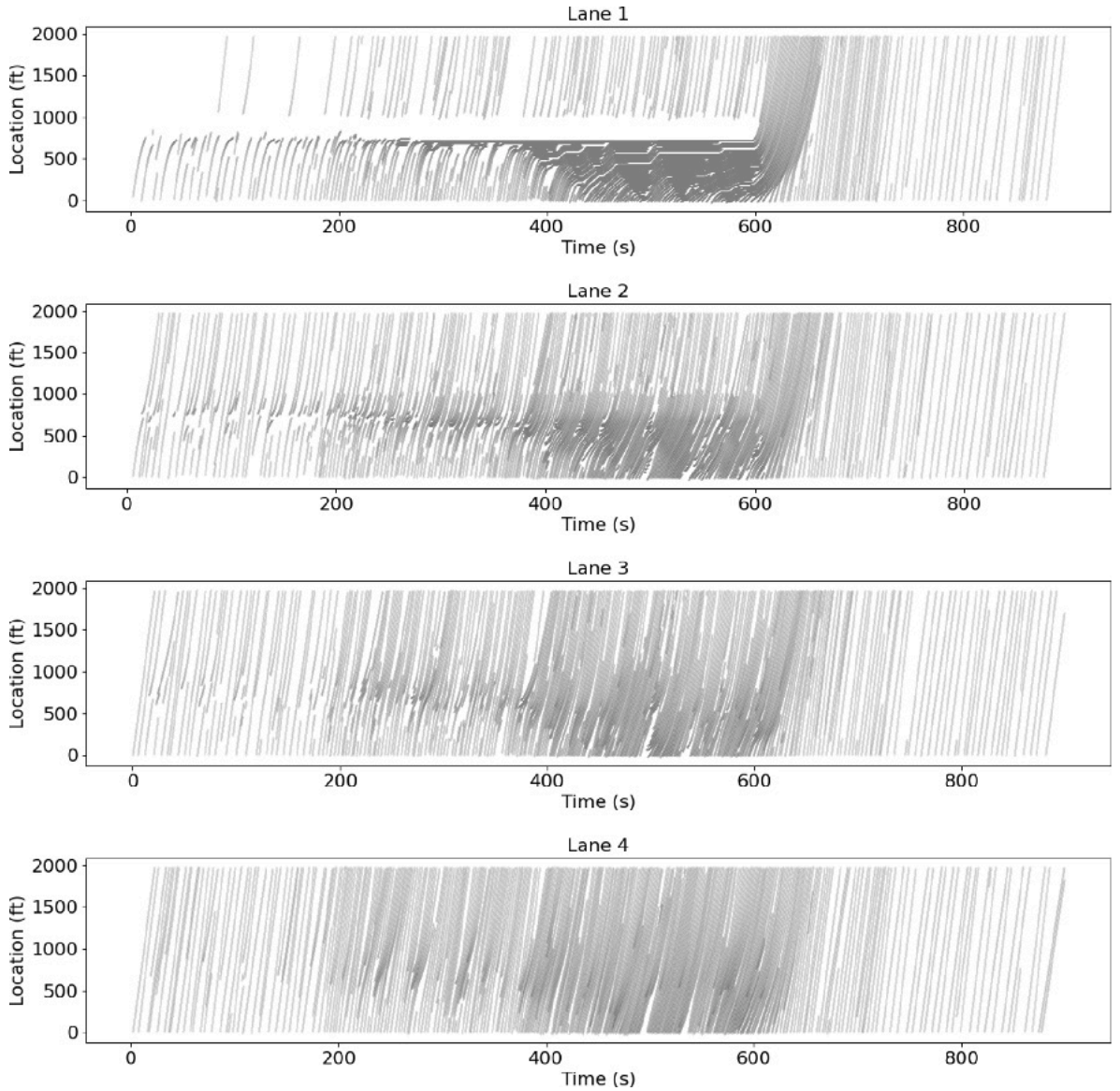
Fig. A.15. Trajectory space–time diagram for the synthetic data (SIM 3): figures from top to bottom correspond to traffic in lane 1–4, respectively.

should be noted that the car-following and lane-changing model parameters were set to default values in TransModeler. Figs. A.16 and A.17 are the polluted and reconstructed visualizations, respectively.

In order to simulate the traffic observation system that utilizes multiple cameras, the 2000 ft-long highway sketch is divided into 20 segments, each 100 ft long (see Fig. A.18). The highway is monitored by three cameras, namely P1C1, P1C2, and P1C3, with P1C1 covering segments 1 to 7, P1C2 covering segments 7 to 14, and P1C3 covering segments 14 to 20. Each camera assigns a different ID for the trajectory of the same vehicle, and there is an overlap area between adjacent cameras, such as segment 7 for P1C1 and P1C2, and segment 14 for P1C2 and P1C3. To replicate real-world scenarios, typical data loss situations such as overpass and camera packet loss (M1, M2, M3) are simulated. For instance, M1 (5–10 min, 100–300 ft) indicates the data in the space–time zone from 5 to 10 min at 100 ft to 300 ft is lost. As in real-world tracking tasks, when a vehicle passes through a missing data area, its ID will be changed and it will be re-initialized as a new object in the system. The overpass area mechanism is similar.

Appendix B. NGSIM data

Fig. B.19, B.20 and B.21 are the ground truth, polluted and reconstructed visualizations for the NGSIM data, respectively.
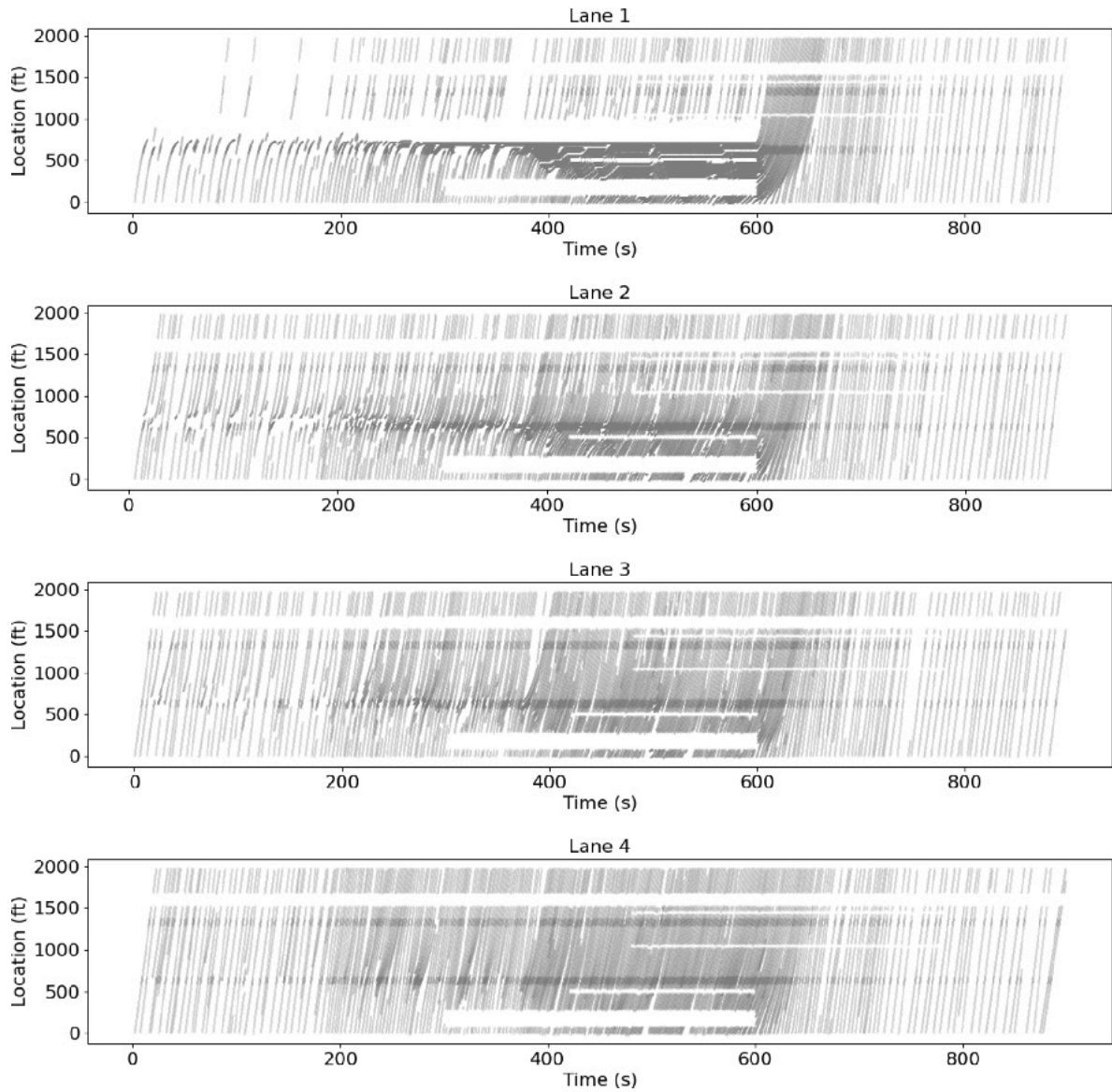
Fig. A.16. Trajectory space–time diagram for the polluted synthetic data (RAW 3).
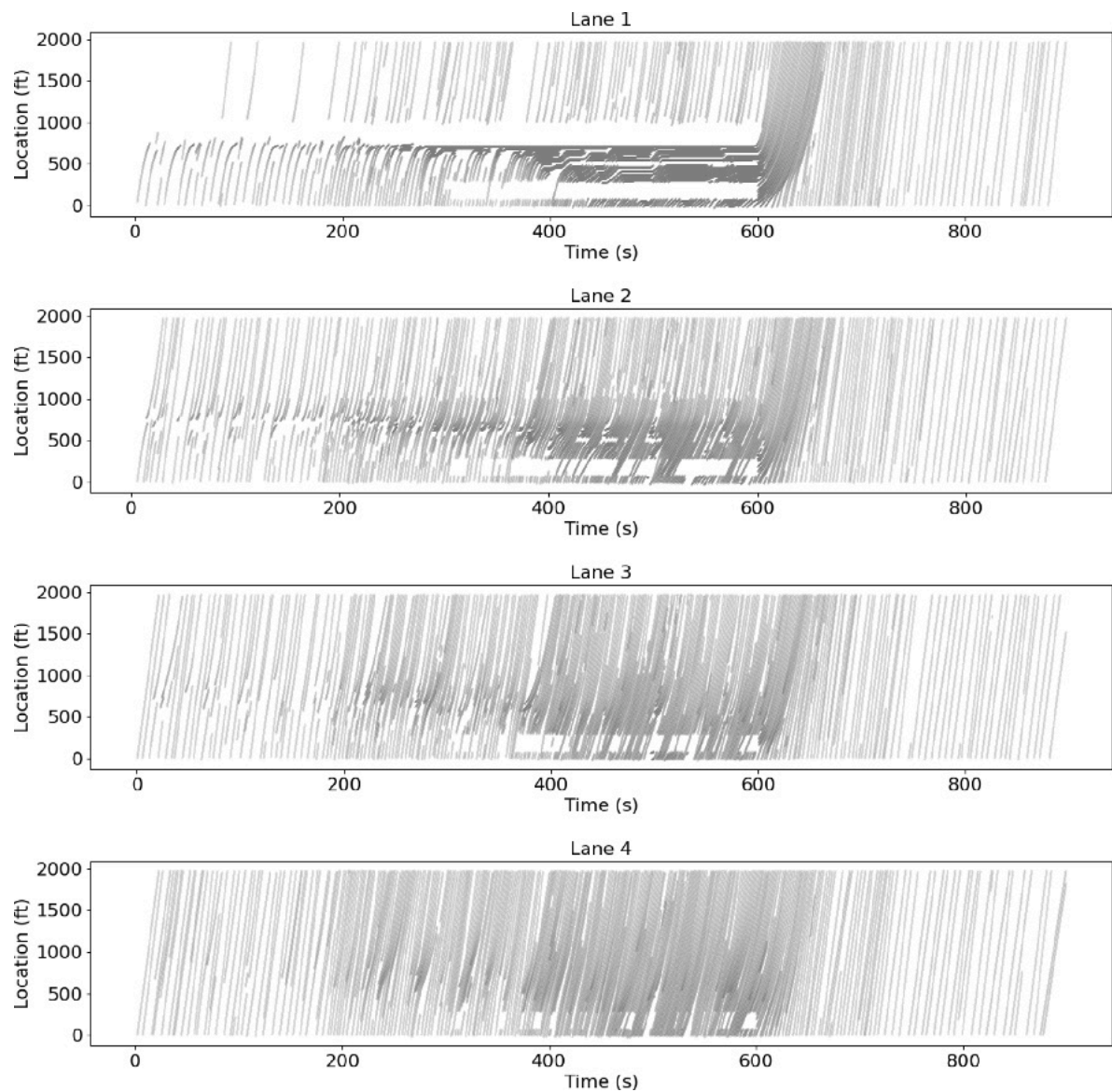
Fig. A.17. Trajectory space–time diagram for the reconstructed synthetic data (REC 3).
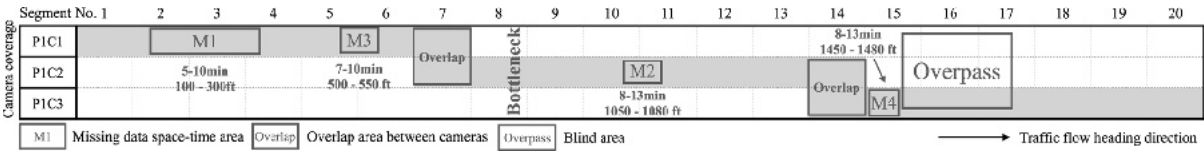

Fig. A.18. A segment with multiple cameras reproduced in micro-simulation.
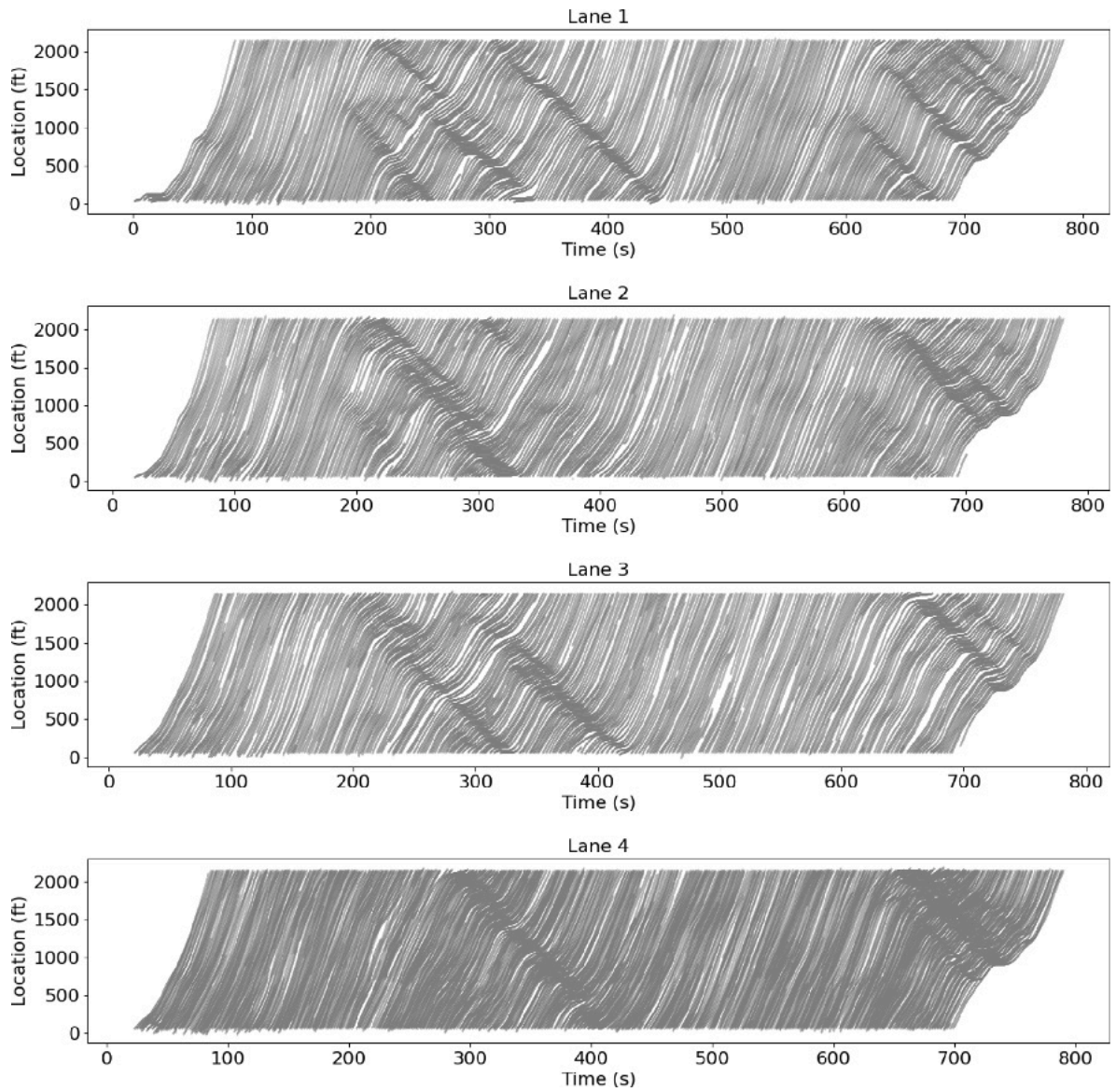
Fig. B.19. Trajectory space–time diagram for the NGSIM data (NGSIM-GT): figures from top to bottom correspond to traffic in lane 1–4, respectively.
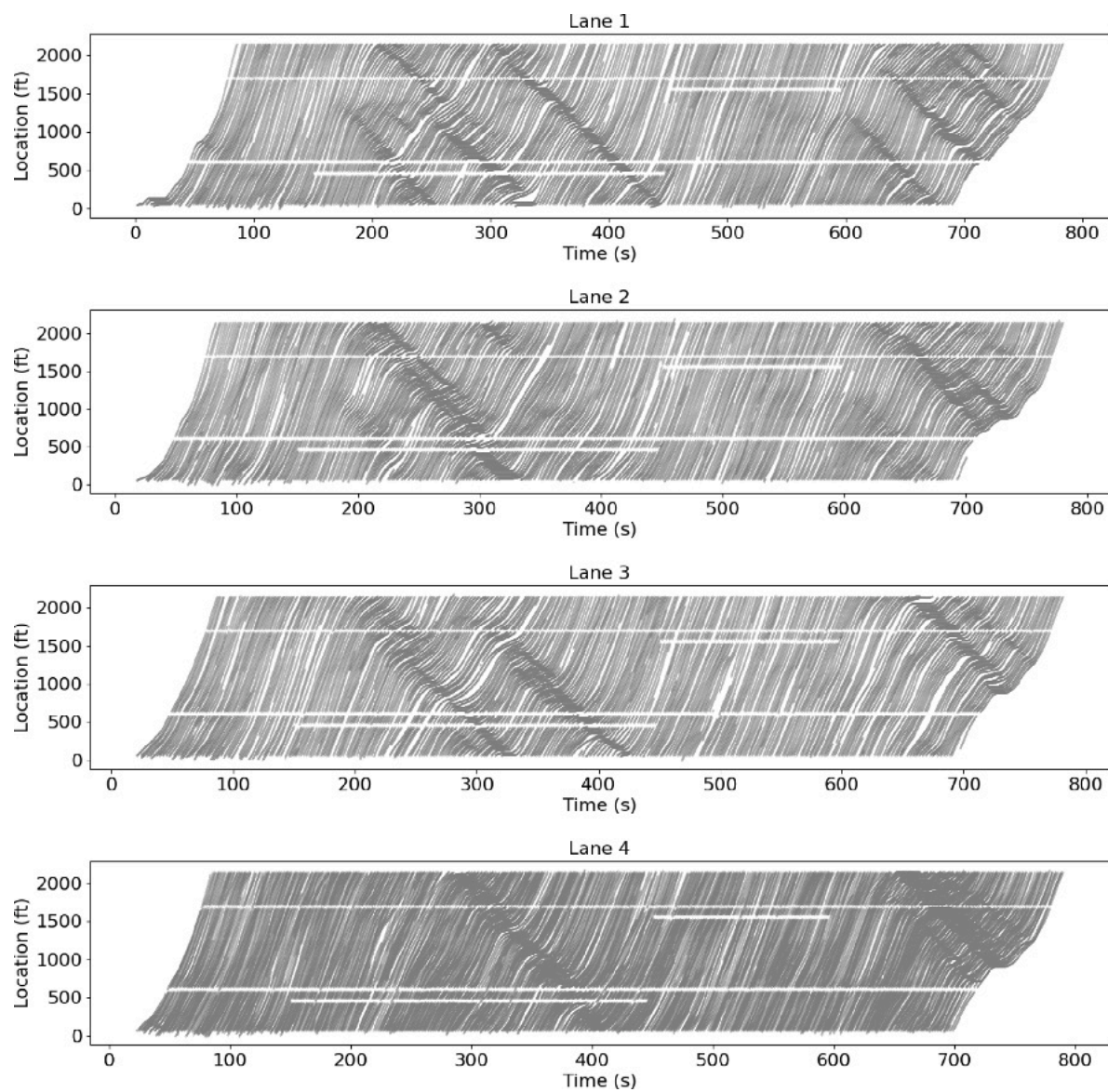
Fig. B.20. Trajectory space–time diagram for the polluted NGSIM data (NGSIM-RAW).
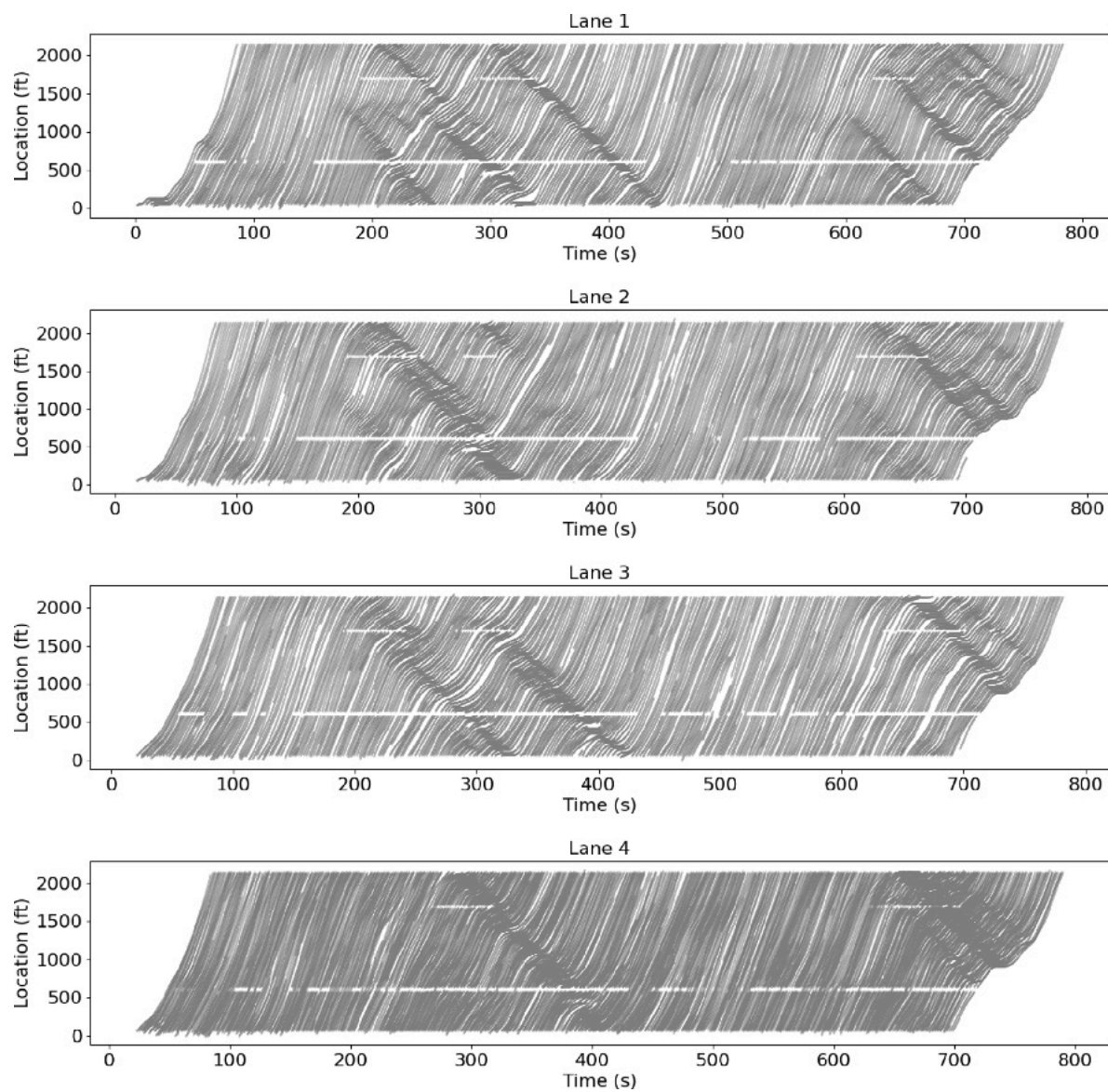
Fig. B.21. Trajectory space–time diagram for the reconstructed NGSIM data (NGSIM-REC).

References

Ahmed, K.I., 1999. Modeling drivers' acceleration and lane changing behavior. (Ph.D. thesis). Massachusetts Institute of Technology.

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1988. Network Flows, first ed. Pearson.

Andriluka, M., Roth, S., Schiele, B., 2008. People-tracking-by-detection and people-detection-by-tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Bae, S.-H., Yoon, K.-J., 2014. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1218–1225.

Barmpounakis, E., Geroliminis, N., 2020. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. Transport. Res. Part C: Emerging Tech. 111, 50–71.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. 2008, 1–10.

Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016a. Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3464–3468.

Bewley, A., Ott, L., Ramos, F., Upcroft, B., 2016b. Alextrac: Affinity learning by exploring temporal reinforcement within association chains. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2212–2218.

Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex Optimization. Cambridge university press.

Castnnón, G., Finn, L., 2011. Multi-target tracklet stitching through network flows. In: 2011 Aerospace Conference. IEEE, pp. 1–7.

Chen, S.S., Donoho, D.L., Saunders, M.A., 2001. Atomic decomposition by basis pursuit. SIAM Rev. 43 (1), 129–159.

Chong, C.-Y., 2012. Graph approaches for data association. In: 2012 15th International Conference on Information Fusion. IEEE, pp. 1578–1585.

Coifman, B., Li, L., 2017. A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. Transport. Res. Part B: Method. 105, 362–377.

Coifman, B., Wu, M., Redmill, K., Thornton, D.A., 2016. Collecting ambient vehicle trajectories from an instrumented probe vehicle: High quality data for microscopic traffic flow studies. Transport. Res. Part C: Emerging Tech. 72, 254–271.

Dicle, C., Camps, O.I., Sznaier, M., 2013. The way they move: Tracking multiple targets with similar appearance. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2304–2311.

Duret, A., Buisson, C., Chiabaut, N., 2008. Estimating individual speed-spacing relationship and assessing ability of Newell's car-following model to reproduce trajectories. Transport. Res. Record 2088 (1), 188–197.

Fard, M.R., Mohaymany, A.S., Shahri, M., 2017. A new methodology for vehicle trajectory reconstruction based on wavelet analysis. Transport. Res. Part C: Emerging Tech. 74, 150–167.

Fiori, C., Ahn, K., Rakha, H.A., 2016. Power-based electric vehicle energy consumption model: Model development and validation. Appl. Energy 168, 257–268.

Fiori, C., Marzano, V., Punzo, V., Montanino, M., 2020. Energy consumption modeling in presence of uncertainty. IEEE Trans. Intelligent Transport. Syst. 22 (10), 6330–6341.

Ford, L.R., Fulkerson, D.R., 1956. Maximal flow through a network. Canadian J. Math. 8, 399–404.

Gazis, D.C., Herman, R., Potts, R.B., 1959. Car-following theory of steady-state traffic flow. Operations Res. 7 (4), 499–505.

Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R., 2013. 3D traffic scene understanding from movable platforms. IEEE Trans. Pattern Anal. Machine Intelligence 36 (5), 1012–1025.

Gipps, P.G., 1981. A behavioural car-following model for computer simulation. Transport. Res. Part B: Method. 15 (2), 105–111.

Gloudemans, D., Wang, Y., Gumm, G., Barbour, W., Work, D.B., 2023a. The Interstate-24 3D Dataset: a new benchmark for 3D multi-camera vehicle tracking. arXiv preprint arXiv:2308.14833.

Gloudemans, D., Wang, Y., Ji, J., Zachár, G., Barbour, W., Hall, E., Cebelak, M., Smith, L., Work, D.B., 2023b. I-24 MOTION: An instrument for freeway traffic science. Transport. Res. Part C: Emerg. Tech. 155, 104311.

Gloudemans, D., Work, D.B., 2021. Vehicle tracking with crop-based detection. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp. 312–319.

Gloudemans, D., Zachár, G., Wang, Y., Ji, J., Nice, M., Bunting, M., Barbour, W.W., Sprinkle, J., Piccoli, B., Monache, M.L.D., Bayen, A., Seibold, B., Work, D.B., 2024. So you think you can track?. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 4528–4538.

Goldberg, A.V., Tarjan, R.E., 1989. Finding minimum-cost circulations by canceling negative cycles. J. ACM (JACM) 36 (4), 873–886.

Greenberg, H., 1959. An analysis of traffic flow. Operations Res. 7 (1), 79–85.

Gunter, G., Gloudemans, D., Stern, R.E., McQuade, S., Bhadani, R., Bunting, M., Delle Monache, M.L., Lysecky, R., Seibold, B., Sprinkle, J., Piccoli, B., Work, D.B., 2021. Are commercially implemented adaptive cruise control systems string stable? IEEE Trans. Intelligent Transport. Syst. 22 (11), 6992–7003.

Guo, Q., Li, L., Ban, X.J., 2019. Urban traffic signal control with connected and automated vehicles: a survey. Transport. Res. Part C: Emerging Tech. 101, 313–334.

Halbach, S., Sharer, P., Pagerit, S., Rousseau, A.P., Folkerts, C., 2010. Model architecture, methods, and interfaces for efficient math-based design and simulation of automotive control systems. Tech. Rep., SAE Technical paper.

Hamdar, S.H., Mahmassani, H.S., 2008. Driver Car-following Behavior: From Discrete Event Process to Continuous Set of Episodes. Tech. Rep., Transportation Research Board 87th Annual Meeting.

Hoogendoorn, S., Knoop, V., 2013. Traffic flow theory and modelling. Transport Syst. Transport Policy: An Introduction 125–159.

Huang, Y.-X., Jiang, R., Zhang, H., Hu, M.-B., Tian, J.-F., Jia, B., Gao, Z.-Y., 2018. Experimental study and modeling of car-following behavior under high speed situation. Transport. Res. Part C: Emerging Tech. 97, 194–215.

Ji, J., Wang, Y., Barbour, W., Work, D.B., 2023. Platoon trajectory reconstruction with conflict resolution using semidefinite relaxation. In: 2023 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, (in press).

Jiang, R., Hu, M.-B., Zhang, H., Gao, Z.-Y., Jia, B., Wu, Q.-S., 2015. On some experimental features of car-following behavior and how to model them. Transport. Res. Part B: Method. 80, 338–354.

Jiang, R., Hu, M.-B., Zhang, H.M., Gao, Z.-Y., Jia, B., Wu, Q.-S., Wang, B., Yang, M., 2014. Traffic experiment reveals the nature of car-following. PloS one 9 (4), e94351.

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. IEEE Trans. Pattern Anal. Machine Intelligence 34 (7), 1409–1422.

Kang, Y., Yin, H., Berger, C., 2019. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. IEEE Trans. Intelligent Vehicles 4 (2), 171–185.

Kesting, A., Treiber, M., 2008. Calibrating car-following models by using trajectory data: Methodological study. Transport. Res. Record 2088 (1), 148–156.

Kim, C., Li, F., Ciptadi, A., Rehg, J.M., 2015. Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4696–4704.

Klein, M.J., 1966. A primal method for minimal cost flows with applications to the assignment and transportation problems. Manag. Sci. 14, 205–220.

Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 2118–2125.

Li, Y., Huang, C., Nevatia, R., 2009. Learning to associate: hybridboosted multi-target tracker for crowded scene. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2953–2960.

Li, L., Jiang, R., He, Z., Chen, X.M., Zhou, X., 2020. Trajectory data-based traffic flow studies: A revisit. Transport. Res. Part C: Emerging Tech. 114, 225–240.

Makridis, M., Mattas, K., Anesiadou, A., Ciuffo, B., 2020. openACC. An open database of car-following experiments to study the properties of commercial ACC systems. arXiv:2004.06342.

Mathisen, B.M., Aamodt, A., Bach, K., Langseth, H., 2020. Learning similarity measures from data. Progress Artificial Intelligence 9 (2), 129–143.

Mechanical Simulation Corporation, 2014. CarSim® mechanical simulation.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.

Montanino, M., Punzo, V., 2013. Making NGSIM data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction. Transport. Res. Record 2390 (1), 99–111.

Montanino, M., Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. Transport. Res. Part B: Method. 80, 82–106.

NGSIM, 2006. The Next Generation Simulation Program. U.S. Department of Transportation Federal Highway Administation, http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

Oh, G., Leblanc, D.J., Peng, H., 2020. Vehicle energy dataset (VED), a large-scale dataset for vehicle energy consumption research. IEEE Trans. Intelligent Transport. Syst..

Perera, A.A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W., 2006. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 1. IEEE, pp. 666–673.

Punzo, V., Borzacchiello, M.T., Ciuffo, B., 2011. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. Transport. Res. Part C: Emerging Tech. 19 (6), 1243–1262.

Rakai, L., Song, H., Sun, S., Zhang, W., Yang, Y., 2022. Data association in multiple object tracking: A survey of recent techniques. Expert Syst. Appl. 192 (4), 116300, https://linkinghub.elsevier.com/retrieve/pii/S0957417421016031.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. Springer, pp. 17–35.

Schakel, W.J., Knoop, V.L., van Arem, B., 2012. Integrated lane change model with relaxation and synchronization. Transport. Res. Record 2316 (1), 47–57.

Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annual Rev. Control 43, 128–151.

Seo, T., Tago, Y., Shinkai, N., Nakanishi, M., Tanabe, J., Ushirogochi, D., Kanamori, S., Abe, A., Kodama, T., Yoshimura, S., et al., 2020. Evaluation of large-scale complete vehicle trajectories dataset on two kilometers highway segment for one hour duration: Zen Traffic Data. In: 2020 International Symposium on Transportation Data and Modelling.

Shi, X., Zhao, D., Yao, H., Li, X., Hale, D.K., Ghiasi, A., 2021. Video-based trajectory extraction with deep learning for High-Granularity Highway Simulation (HIGH-SIM). Commun. Transport. Res. 1 (12), 100014.

Sokkalingam, P.T., Ahuja, R.K., Orlin, J.B., 2000. New polynomial-time cycle-canceling algorithms for minimum-cost flows. Networks 36 (1), 53–63.

Spannaus, P., Zechel, P., Lenz, K., 2021. AUTOMATUM DATA: Drone-based highway dataset for the development and validation of automated driving software for research and commercial applications. In: 2021 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 1372–1377.

Stern, R.E., Cui, S., Delle Monache, M.L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Pohlmann, H., Wu, F., Piccoli, B., et al., 2018. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. Transport. Res. Part C: Emerg. Tech. 89, 205–221.

Tao, S., Manolopoulos, V., Rodriguez Duenas, S., Rusu, A., 2012. Real-time urban traffic state estimation with A-GPS mobile phones as probes. J. Transport. Tech. 2 (1), 22–31.

Thiemann, C., Treiber, M., Kesting, A., 2008. Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. Transport. Res. Record 2088 (1), 90–101.

Treiber, M., Kesting, A., 2013. Traffic flow dynamics. In: Traffic Flow Dynamics: Data, Models and Simulation. Springer-Verlag Berlin Heidelberg.

Treiterer, J., Myers, J., 1974. The hysteresis phenomenon in traffic flow. Transport. Traffic Theory 6, 13–38.

Vandenberghe, L., 2010. The CVXOPT linear and quadratic cone program solvers. Online: http://cvxopt.org/documentation/coneprog.pdf.

Vyahhi, N., Bakiras, S., Kalnis, P., Ghinita, G., 2008. Tracking moving objects in anonymized trajectories. In: International Conference on Database and Expert Systems Applications. Springer, pp. 158–171.

Wang, Y., Ji, J., Work, D.B., 2023. Online min cost circulation for multi-object-tracking on fragments. In: 2023 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE (in press).

Wang, Y., Papageorgiou, M., 2005. Real-time freeway traffic state estimation based on extended kalman filter: a general approach. Transport. Res. Part B: Method. 39 (2), 141–167.

Wang, C., Wang, Y., Wang, Y., Wu, C.-T., Yu, G., 2019. muSSP: Efficient min-cost flow algorithm for multi-object tracking. Adv. Neural Inform. Process. Syst. 32.

Wang, C., Wang, Y., Yu, G., 2020. Efficient global multi-object tracking under minimum-cost circulation framework. IEEE Trans. Pattern Anal. Machine Intell..

Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3645–3649.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transport. Res. Part C: Emerging Tech. 4 (3), 113–129.

Yang, D., Kuijpers, A., Dane, G., van der Sande, T., 2019. Impacts of large-scale truck platooning on dutch highways. Transport. Research Procedia 37, 425–432.

Yoon, J.H., Yang, M.-H., Lim, J., Yoon, K.-J., 2015. Bayesian multi-object tracking using motion context from multiple objects. In: 2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, pp. 33–40.

Zhang, L., Li, Y., Nevatia, R., 2008. Global data association for multi-object tracking using network flows. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. Royal Statist. Soc.: Series B (Statistical Methodology) 67 (2), 301–320.