Transportation Research Part C 158 (2024) 104416

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Partial trajectory method to align and validate successive video cameras for vehicle tracking

Benjamin Coifman [a,*], Lizhe Li [b]

[a] The Ohio State University, Joint Appointment with the Department of Civil, Environmental, and Geodetic Engineering, and the Department of Electrical and Computer Engineering, Hitchcock Hall 470, 2070 Neil Ave, Columbus, OH 43210, USA
[b] The Ohio State University, Department of Electrical and Computer Engineering, USA

ARTICLE INFO

Keywords:
Machine vision
Video image processing
Traffic flow theory
Highway traffic
NGSIM
Empirical data
Congested traffic

ABSTRACT

This paper develops the partial trajectory method to align the views from successive fixed cameras that are used for video-based vehicle tracking across multiple camera views. The method is envisioned to serve as a validation tool of whatever alignment has already been performed between the cameras to ensure high fidelity with the actual vehicle movements as they cross the boundaries between cameras. The strength of the method is that it operates on the output of vehicle tracking in each camera rather than secondary features visible in the camera view that are unrelated to the traffic dynamics (e.g., fixed fiducial points). Thereby providing a direct feedback path from the tracking to ensure the quality of the alignment in the context of the traffic dynamics. The method uses vehicle trajectories within successive camera views along a freeway to deduce the presence of an overlap or a gap between those cameras and quantify how large the overlap or gap is. The partial trajectory method can also detect scale factor errors between successive cameras. If any error is detected, ideally one would redo the original camera alignment, if that is not possible, one could use the calculations from the algorithm to post hoc address the existing alignment.

This research manually re-extracted the individual vehicle trajectories within each of the seven camera views from the NGSIM I-80 dataset. These trajectories are simply an input to the algorithm. The resulting method transcends the dataset and should be applicable to most methods that seek to extract vehicle trajectories across successive cameras. That said, the results reveal fundamental errors in the NGSIM dataset, including unaccounted for overlap at the boundaries between successive cameras, which leads to systematic speed and acceleration errors at the six camera interfaces. This method also found scale factor errors in the original NGSIM homographies. In response to these findings, we identified a new aerial photo of the NGSIM site and generated new homographies. To evaluate the impact of the partial trajectory method on the actual trajectory data, the manually re-extracted data were projected into the new coordinate system and smoothed. The re-extracted data shows much greater fidelity to the actual vehicle motion. The re-extracted data also tracks the vehicles over a 14% longer distance and adds 23% more vehicles compared to the original NGSIM dataset. As of publication, the re-extracted data from this paper will be released to the research community.

* Corresponding author.
E-mail address: coifman.1@osu.edu (B. Coifman).

https://doi.org/10.1016/j.trc.2023.104416
Received 23 December 2022; Received in revised form 10 November 2023; Accepted 11 November 2023

1. Introduction

This paper develops the partial trajectory method to align the views from successive fixed cameras that are used for video-based vehicle tracking across multiple camera views. The method is envisioned to serve as a validation tool of whatever alignment has already been performed between the cameras to ensure high fidelity with the actual vehicle movements as they cross the boundaries between cameras. The strength of the method is that it operates on the output of vehicle tracking in each camera rather than secondary features visible in the camera view that are unrelated to the traffic dynamics (e.g., fixed fiducial points). Thereby providing a direct feedback path from the tracking to ensure the quality of the alignment in the context of the traffic dynamics.

In general, tracking vehicles across successive cameras is an important tool for monitoring roadway traffic, the resulting vehicle trajectories are of great value for studying traffic dynamics and developing microscopic traffic flow models. We specifically focus on the general problem of video-based vehicle tracking from a high vantage point to extract vehicle trajectories over a long distance, e.g., NGSIM, HighD and pNEUMA datasets (respectively: Kovvali et al., 2007; Krajewski et al., 2018; Barmpounakis and Geroliminis, 2020). Such deployments present numerous challenges because the camera must have a sufficient resolution to track the vehicles, but achieving this resolution usually limits the field of view of a given camera. To cover a meaningful distance along the road it is necessary to use several cameras, each with a different focal length to capture the traffic with sufficient resolution at different locations along the road.

While many studies have addressed tracking vehicles within a single camera, surprisingly little has been published about handling the interface between cameras. As noted in the HighD documentation, "If multiple cameras are used at the same time, problems can also occur at the transitions [between cameras]." Most studies that track vehicles across successive cameras fail to account for all the issues that arise between the successive camera views. The most common approach is to use a homographic transformation for each camera to project the image plane into a common ground plane using an aerial photo or other rectified map as a common reference. Other approaches forego external references and turn to image stitching techniques that seek common features seen in the overlap area of two or more camera views. Typically, after alignment the successive camera views are trimmed to remove overlap and provide perfectly abutting coverage to facilitate tracking as if the combined views were contiguous.

Collecting accurate vehicle trajectories requires precise alignment between successive cameras. When done without error the conventional alignment methods provide a smooth transition between successive cameras, the trouble is, most existing methods lack any feedback path to assess whether an alignment error occurred. If there was an error in the alignment, it will induce systematic errors in vehicle speed and acceleration in the vicinity of the camera boundaries and degrade all subsequent applications that use the resulting trajectory data.

This paper develops a methodology that uses the extracted vehicle trajectories within successive camera views along a freeway to deduce the presence of two errors that are likely to arise from deficient alignment: quantifying the amount of overlap or gap between successive camera views, and identifying the presence of a scale factor error between successive cameras. By using the resulting trajectories, our method provides a new feedback path to assess the quality of the implemented alignment. While there are other ways to detect an overlap, the fact that the method can detect and quantify the presence of a gap between camera views is an important ability that cannot be realized using traditional visual based solutions that rely on seeing the same feature concurrently in two different cameras.

Typically, after alignment and trimming the successive views, multi-camera vehicle tracking is applied across the camera boundaries as if there were a single contiguous view. Our method requires a small deviation from this approach. For the validation it is critical that the analysis be done by tracking vehicles independently in each individual camera, if the successive views are not kept independent adjusting the trajectories in one camera based on features seen in an adjacent camera would disrupt the key features this methodology leverages. Vehicle tracking also typically involves smoothing the trajectory data, which is a necessary step to being able to measure instantaneous speed, but it can introduce other positioning errors (e.g., as will be demonstrated herein for the NGSIM data). For robustness, our method deliberately avoids the need to smooth the vehicle trajectories, and instead, we use the raw vehicle trajectories as they cross detection zones at the abutting boundaries of a given pair of successive cameras.[1]

Specifically, in each camera the portion of each trajectory passing the detection zone is retained (hence the name, "partial trajectory method"). The partial trajectories in the downstream camera are shifted longitudinally over a range of possible gaps relative to the partial trajectories in the upstream camera. Then for a given gap the upstream and downstream partial trajectories are matched using an exhaustive search that fits a third degree polynomial to each feasible pair and for each upstream partial trajectory the algorithm picks the downstream partial trajectory with the smallest residual error between the fitted curve and the two partial trajectories. Then the algorithm varies the assumed gap between cameras to find the spatial offset that maximizes the trajectory consistency between the two detection zones. This approach also accounts for the possibility of an overlap (simply a negative gap), so it is not necessary to know if there is a gap or an overlap a priori. The approach is evaluated over a range from an overlap of 20 ft to a gap up to 80 ft between cameras. The intent is to provide alignment in the range where it is difficult for a human to detect a misalignment from a cursory inspection. It is envisioned that the presence of a gap or overlap beyond this range would easily be detectable by a person, and upon seeing a large discrepancy, they could realign the cameras to get them within the target range of this work. It will also be shown that in the process of evaluating the range of overlap, the methodology can measure the scale factor between successive cameras.

[1] The method is also compatible with smoothed data, but by using the raw data we avoid potentially confounding smoothing errors.

Our alignment methodology cannot be evaluated in a vacuum. It requires vehicle trajectories as input and the final alignment must be evaluated using a common reference. To this end, we employ the original video from the Next Generation SIMulation (NGSIM) dataset to derive the input raw trajectory data from each camera. We do not use the original NGSIM trajectories, instead we track the vehicles anew (Coifman and Li, 2017 & 2022) and to ensure generalizability, we use the raw trajectories within each camera, i.e., instantaneous position as found in the individual frames, with no cleaning or smoothing between frames, and thus, instantaneous speed is not available for the trajectories. Several factors led to this choice: (1) most of the commonly used vehicle tracking algorithms should be able to outperform the noisy trajectory data used herein, (2) it should be more robust in the presence of degraded effective resolution at the distant end of camera views, and (3) some data smoothing methods require observations beyond the current decision point, which would increase the distance between the last tracked point and the boundary of the given camera view. For completeness, we must discuss the details of extracting the raw vehicle trajectories (Section 2) before we develop the alignment method (Section 3). In practice, the raw vehicle trajectories would come from whatever vehicle tracking methodology is used in the given deployment.[2]

When we apply the methodology to the NGSIM video data we discover and quantify overlap and scale factor errors that undermine the quality of the original NGSIM vehicle trajectory dataset. Then, for validation, we rely on an independent, orthorectified aerial photo of the NGSIM data collection site. As shown herein, the partial trajectory method performed very well on the NGSIM videos. Finally, we correct for these errors and smooth the re-extracted vehicle trajectory data from the NGSIM videos used in this study to demonstrate the benefits of the methodology. The re-extracted trajectory data and associated validation video will be shared as of publication of this paper (see Coifman, 2023).

1.1. Background

Before getting to the multiple camera problem, we first need to briefly consider the challenge of tracking vehicles within a single camera. The performance of most video-based vehicle tracking systems improve as the viewing angle approaches a *straight down* view since this overhead view minimizes projection errors due to the viewing angle and occlusions between vehicles; thus, improving segmentation and localization of the targets. Typically, such views are only attainable from aircraft (including unmanned drones) (e.g., Treiterer and Myers, 1974; Hoogendoorn et al., 2003; Coifman et al., 2006; Kim et al., 2008; Salvo et al., 2017; Coifman et al., 2018; Krajewski et al., 2018). Effective tracking can still be done using oblique angles from cameras mounted on tall poles or buildings after accounting for occlusions, often by using a homography to project the 2D camera frame into a 2D ground plane to mimic an overhead view (e.g., Coifman et al., 1998; Kanhere and Birchfield, 2008; Battiato et al., 2015). In either case, it is rare to have a camera view with an effective tracking region greater than a few hundred feet, and often the range is much less. At freeway speeds a vehicle could pass completely through a 300 ft field of view in under 3 sec. So, the use of video-based vehicle tracking to study traffic dynamics over space usually requires multiple cameras to capture meaningful results.

There is another body of research that seeks to reidentify vehicles between different cameras with large gaps of hundreds to thousands of feet between camera views (e.g., Makris et al., 2004; Jelača et al., 2011; Castañeda et al., 2011; Meng et al., 2020; Zhao et al., 2021), but these studies are focused on reacquisition for applications such as origin/destination flows or travel time measurement rather than microscopic vehicle trajectory extraction. The reidentification problem where vehicles are expected to go unseen as they travel between distant cameras does not seek to precisely align the cameras to extract contiguous trajectories, and thus, our paper is beyond the scope of those applications. It is also important to distinguish the present work that seeks to track vehicles along the roadway in 2D space using a monocular camera view, from other research that uses overlapping camera views to deduce positions in a 3D coordinate system within the overlap area (e.g., Hu et al., 2006; Zhou and Aggarwal, 2006).

The present work is aimed at applications that seek to measure instantaneous speed and acceleration for all tracked vehicles at all times, where it is necessary to track vehicles with localization accuracy within a few inches, including while transitioning between cameras. This level of precision is challenging enough within a single camera (Coifman and Li, 2017; Krajewski et al., 2018) that few have considered the challenges of achieving such precision across the boundaries between successive cameras.

Far more attention needs to be paid to the "hand-off" between cameras. The few studies that track vehicles over multiple cameras with high precision localization appear to take the hand-off for granted. Among these past studies, none has seen wider impact than the NGSIM study (Kovvali et al., 2007), where the resulting empirical vehicle trajectory dataset has become a de facto standard for microscopic traffic flow studies, e.g., Laval and Leclercq (2008), Thiemann et al. (2008), Chiabaut et al. (2009), Choudhury et al. (2009), Zheng et al. (2011), Hou et al. (2014), Punzo et al. (2014), and Coifman (2015). As well as training data for machine vision based vehicle tracking studies (e.g., Dixon et al., 2009, Zhang and Jin, 2019). Given the broad use of the vehicle trajectory data, it is of utmost importance to address overlap/gap and scale factor errors before using the trajectory data for other applications.

1.2. Overview

The remainder of the paper is as follows. Section 2 presents the pre-processing necessary for the analysis and evaluation. This starts with the video data and within-camera vehicle trajectory extraction- both of which serve as input to the methodology, but the specific inputs and their details are not intrinsic to the partial trajectory method. Then the discussion turns to ground truth camera alignment that ultimately relies on a new, higher resolution aerial photo of the data collection site. With the pre-processing complete, Section 3

[2] With the stipulation that no matter what tracker is used, the raw vehicle tracking is done independently in each camera view, even if the main tracking algorithm is applied as if there were a single contiguous view across more than one camera.

shifts to the partial trajectory method itself. This work begins with estimating the gap or overlap between two successive camera views of a given lane. The work then shifts to detecting and quantifying scale factors between a given pair of cameras. Along the way, this section uses the ground truth data from the pre-processing to evaluate the performance. Section 4 explores the results after combining the trajectories across cameras and then studies the fidelity of both the original NGSIM and newly re-extracted trajectories relative to the actual vehicle motion. Finally, the paper closes with a brief discussion and conclusions in Section 5.

2. Preprocessing- generating the input and validation data

The alignment methodology will be developed in Section 3, but the methodology requires vehicle trajectories as input and the final alignment must be evaluated using a common reference. Section 2.1 presents the vehicle trajectory data that will be used as input to the alignment methodology. Section 2.2 calculates the baseline camera overlap that will be used for evaluation and Section 2.3 recalculates the homographies using an independent reference to assess the possibility of any non-unity scale factors between adjacent cameras and to provide ground truth localization for the camera views, camera overlap and vehicle trajectories. In short, this section presents the preprocessing necessary to conduct and evaluate the alignment methodology in Section 3.

2.1. Video dataset and generating the Within-Camera vehicle trajectories

This section presents the vehicle trajectory data that will be used in the subsequent sections. While it is important to understand the data used for this research, from the perspective of the alignment methodology the vehicle trajectories are simply an input.

This work re-extracts vehicle trajectories from a portion of the original NGSIM videos. A little background helps set the context. The NGSIM project is one of the most ambitious efforts to collect vehicle trajectory data to date. There are four NGSIM datasets, two from freeways (I-80 and US-101) and two from arterial corridors (Lankershim and Peachtree). In each case the dataset includes complete vehicle trajectories for all vehicles from entrance to exit through a road link during an extended period. Each NGSIM dataset reports the status of each vehicle at 10 Hz, including: longitudinal and lateral position, speed, acceleration, and information about the vehicle (length and class). While the NGSIM datasets are among the largest available, they are still relatively small, with each set consisting of a few thousand vehicles observed over a short distance (roughly 1/3 mile per set) and duration (under an hour per set). The 10 Hz corresponds to the frame rate of the video, as such, throughout this work we will express time in terms of the frame number (at 10 frames per sec), consistent with the NGSIM trajectory dataset.

This paper uses the NGSIM I-80 freeway dataset that was collected in the Berkeley Highway Laboratory in Emeryville, CA (Coifman et al., 2000). The NGSIM I-80 data were split into three periods, each lasting a little more than 15 min. This study uses the first of these periods. The underlying video of eastbound traffic was recorded on April 13, 2005, by seven cameras located on the roof of a 30 story building. The cameras are numbered in the sequence that vehicles traversed them, where cameras 1–3 recorded traffic approaching the camera location, camera 4 covered the transition from approaching to departing, and cameras 5–7 recorded the vehicles as they departed the camera location. Combined, the original NGSIM extracted vehicle trajectories covered just under 1,600 feet of the 6 eastbound lanes. The NGSIM researchers projected the oblique camera views into the ground plane using standard homographic techniques and then cropped the field of view with the intention that successive cameras were perfectly abutted, i.e., the first pixel in one camera's view would fit immediately after the last pixel in the previous camera's view. After tracking all of the vehicles, the NGSIM researchers generated validation videos by superimposing pink boxes with green vehicle ID numbers onto the projected video. The original release of the NGSIM dataset included both the raw video before projection and the validation video with the pink boxes, it does not include the projected video without the pink boxes. Coifman and Li (2017) verified that the pink boxes are consistent with the final trajectories in the NGSIM dataset. Unfortunately, contrary to the assumptions of the NGSIM researchers, almost all successive camera pairs in the I-80 dataset overlapped by a few pixels, as found in Coifman and Li (2022) and illustrated in Fig. 1. The discovery of that problem was the inspiration for the present work- developing a tool to automatically detect the presence of such misalignment and quantify its magnitude.

While the projected NGSIM videos have a small amount of unintended overlap between cameras, this work seeks to include a larger range of overlap to develop the new methodology. Furthermore, the pink validation boxes disrupt vehicle tracking from the validation video. So this work has to reproject the NGSIM raw video into the ground plane. The first step is to recover the homography used by the original NGSIM projection. To this end, we generated a ***background image*** of the NGSIM roadway by finding the median value of each pixel over 1000 frames in both the raw and projected videos (e.g., Fig. 3D). In projective geometry a homography transformation is a bijection that maps lines to lines, therefore by locating all the matched feature points in the two spaces, we recalculate the original NGSIM homography between the raw and projected road. One advantage of the background image is that we can access all of the roadway features without any occlusion from vehicles. We mark and match the start and end points of all visible lane markers as feature points. By comparing all the feature points in the raw road and projected road, we find all of the matched pairs, revealing the homography between the two images. Using this same homography, this work reprojects the raw video.

For cameras 1, 2, 3 and 5 this work then takes the recalculated homography but extends the longitudinal distance that is retained, thereby deliberately increasing the overlap between successive camera views, e.g., Fig. 2. This extended surveillance region covers a longer range than the original NGSIM projected video, providing overlap between the pair of successive camera views of up to 147 ft. Camera 4 had a straight down view of the freeway while the adjacent cameras 3 & 5 had viewing angles that were close to straight down. Cameras 1, 2, 6 and 7 all had larger angles of incidence. The projection errors increase as the angle of incidence increases and on a related note, the homographic distortions increase on the far end of each camera view. These errors are most acute in camera 7, a short distance into the field of view of camera 7 the feature points become hard to recognize. So, in both the original and re-extracted

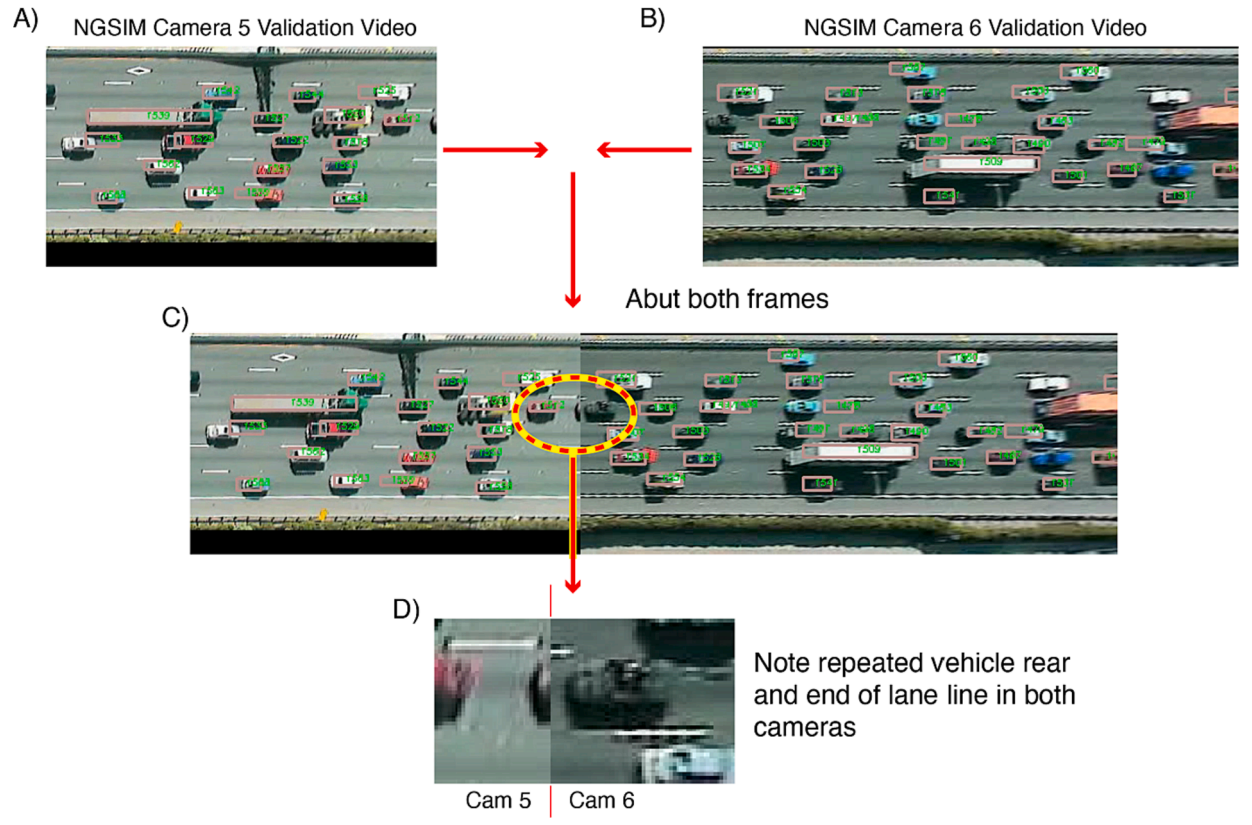

Fig. 1. NGSIM video frames taken from the same instant demonstrating the overlapping view error. NGSIM validation video from (A) camera 5, and (B) camera 6. (C) The two views are abutted together. (D) A detail of the abutment, now taken from the re-extracted video to eliminate the validation boxes, showing that the rear part of vehicle 1508 still can still been seen in camera 5 while is has fully entered camera 6, also note the right end of the lane line above this vehicle is also visible in both camera views.

data, the extracted trajectories in the range of cameras 1, 2 and 7 should be less reliable than those from the closer cameras.[3]

When calculating the homography between the two coordinate systems, raw and projected, after projection some areas of the projected image could have a resolution that is higher than the original raw image. In the case of the projected NGSIM videos, the projected images have a scale of approximately 0.5 ft/pixel. To find the effective resolution in the context of the raw video a new ratio is defined as the ratio of the physical distance in ft between two feature points in the projected image and their pixel distance in the raw image. Two different feature measurements are used, the length of the lane lines and the length of the gaps between them. In all cameras the effective resolution drops off as one moves from the near end to the far end of the field of view. Cameras 3–5 maintain an effective resolution on the order of 0.5 ft/pixel throughout. Cameras 2 and 6 show an effective resolution of 0.5 ft/pixel on the near end, dropping to 1.5 ft/pixel on the far end, while cameras 1 and 7 start at about 0.9 ft/pixel and degrade to almost 2 ft/pixel on the far end of the span used for tracking (with even further degradation in the unused region beyond this span). The drop in effective resolution is an important factor to keep in mind when considering measurements at the far end of a given camera, as will be done in following sections.

Meanwhile, the conventional practice of projecting the 2D image plane seen by the camera into the 2D ground plane of the roadway has the implicit assumption that all objects are actually in the ground plane. Obviously, 3D vehicles violate this assumption, resulting in projection errors. The shape of the projection of a given vehicle varies as it moves due to the evolving viewing angle. In this dataset, camera 1 can only view the front of vehicles while camera 7 can only view the rear of vehicles. These projection errors are like shadows, the higher the feature is off the ground, the further it projects away from its true location coordinates in the ground plane. The projection errors also increase as the vehicle moves further from the camera; thus, the projection of the top front of a departing vehicle should seemingly move faster over ground than the actual vehicle. To avoid the large projection errors arising from the far end of the vehicles, our trajectory re-extraction work tracks the bottom of the near side of the given vehicle[4] because that feature is always closest to the ground and therefore should exhibit the smallest projection error. Ultimately, we tracked over 2,500 vehicles during 16.7 min as they traversed roughly 1,800 ft of the I-80 test site (our coverage area is slightly longer than the original NGSIM tracking

[3] These regions roughly correspond to upstream of 300 ft and downstream of 1500 ft in the newly extracted data.

[4] For approaching traffic in cameras 1–4 we use the bottom front and for departing traffic in cameras 4–7 we use the bottom rear.

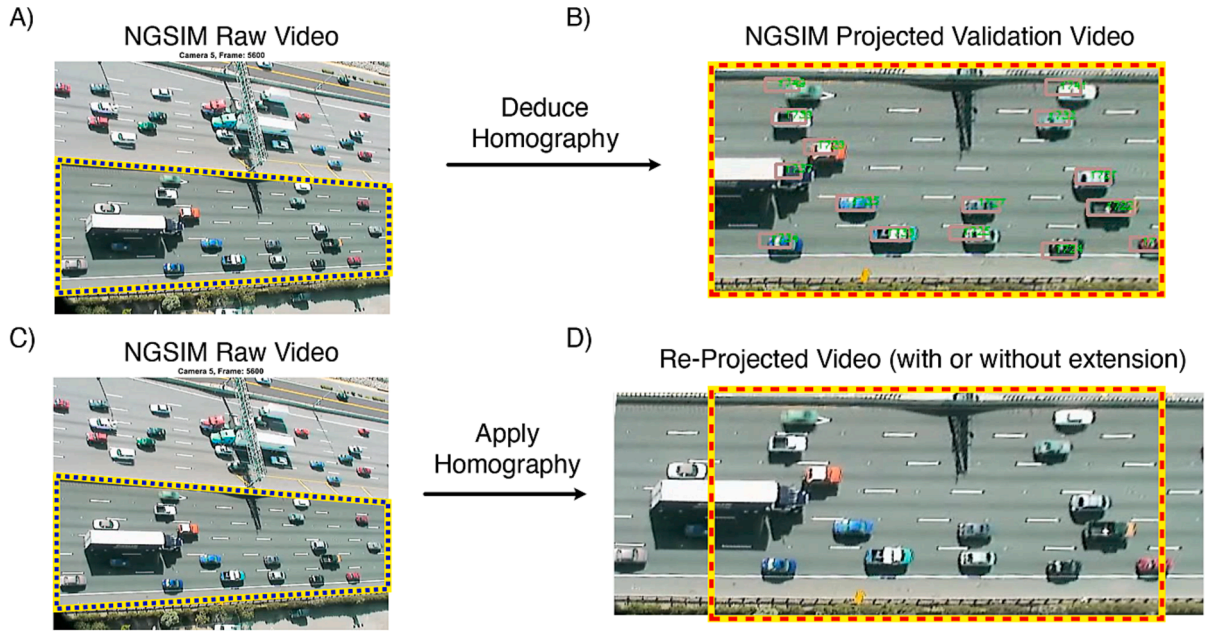

Fig. 2. An example from camera 5 of calculating the homography used in the original NGSIM projections from: (A) the raw video, and (B) the validation video (note the presence of the validation boxes over each vehicle). Using that homography to reproject: (C) the raw video into (D) the NGSIM coordinates with an extension beyond the original NGSIM bounds in this case. Note that some of the other cameras were not extended when reprojected. In either case, this reprojection eliminates the validation boxes present in the validation video.

distance). The front and rear trajectories in the straight down view of camera 4 are then used to measure each vehicle's length and associate the two ends of the given vehicle together.

The core of this paper assumes that the vehicle trajectories are an input to the current research rather than a product of the process. Still, it is necessary to have vehicle trajectories for the input. To this end, this work used the spatial temporal stack (ST stack) method from Coifman and Li (2022) to manually re-extract all of the individual vehicle trajectories in a given camera. Basically, a ST stack creates a 2D image that captures the motion of vehicles over time along the road's distance, as follows: first, for each lane in each camera we define a line of pixels along the middle of the given lane. For each frame we extract the pixels along the line to form a column in the ST stack. So, the successive columns in the ST stack correspond to the same line of pixels in successive frames of the video. For the given lane, this process is repeated for every frame in the video, with each successive column of pixels placed to the right of the corresponding column from the frame before, yielding the ST stack in the form of an image with length in pixels equal to that of the video frame and width equal to the total number of frames. Thus, the ST stack is effectively a graphically constructed time–space diagram from the video. The colorful tracks in this stack show the progression of the individual vehicles, and at times shadows from the adjacent lanes. This research uses a purpose-built graphical user interface (GUI) to facilitate the data reduction. Following Coifman and Li (2022), each vehicle track is flattened, first using a semi-automated edge detection and then manually cleaned, e.g., to address partial occlusions when two tracks intersect. This flattening process is then used to back-calculate the original vehicle trajectory over time and space. Of course, the manual tracking is only as accurate at the original video resolution. In other words, each individual measurement will exhibit noise on the order of one pixel due to the limits of the video resolution. However, these errors should not accumulate.

Within a given camera the trajectories were extracted without regard for the neighboring cameras, and once extracted no effort was made to clean the trajectories, so the individual position data are only accurate to a few feet and given the noise in the raw data it would be pointless to attempt to calculate instantaneous speed. We use these raw trajectories (i.e., without any smoothing or cleaning) extracted by lane in each camera as the input to for the camera alignment work throughout the remainder of this paper. The re-extracted data are noisy, and the quality is not good enough to measure speed or acceleration. The choice to retain the noise was deliberately done to highlight the robustness of the alignment tools developed in the rest of this work.

2.2. Baseline camera alignment and revelations about the original NGSIM data

This section defines a local coordinate system in each camera view. The origin of the longitudinal coordinate is set to the upstream start of the original NGSIM video (i.e., without the extended overlap). So, for the three camera views that were extended upstream (cameras 2, 3, and 5) the start of the extended camera view is at a negative distance in the local camera coordinate system. To illustrate this point, the origin for camera 5 is at the far left of Fig. 2B in the original NGSIM view, and the left vertical dashed line in the extended surveillance region of the reprojected camera 5 in Fig. 2D. Meanwhile, the start of the extended camera 5 view on the far left of Fig. 2D

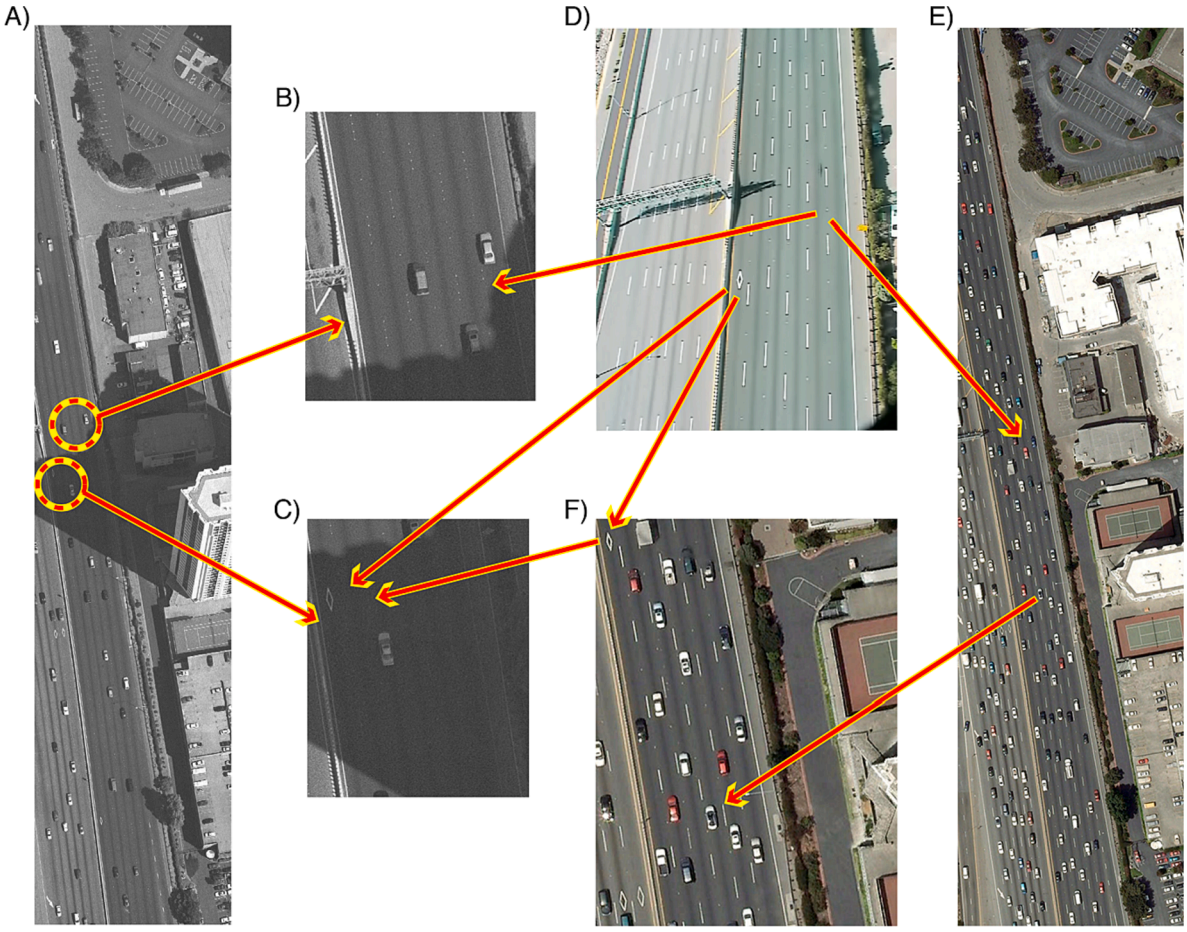

Fig. 3. (A) A portion of the NGSIM aerial photo, (B) detail of part A outside of the shadow- note the absence of lane lines, probably due to the spectrum used for the imagery, (C) detail of part A inside the shadow- note the absence of almost all features. (D) Raw camera 5 background image- note the corresponding areas in parts B & C, e.g., the HOV diamond marker. (E) The corresponding portion of the new aerial photo from Google Earth dated May 31, 2007, (F) detail of part E showing the clear pavement marking features.

is at roughly −40 ft in the new local coordinate system. At this point, the longitudinal distance in the NGSIM coordinate system is assumed to be true and is retained for analysis in Sections 3.1-3.3.[5]

To establish a baseline alignment, this work uses the background image for each of the seven cameras from the reprojected video (recall that each pixel in the background image is the median value of that pixel over 1000 frames). A human then shifted the background image from one camera against that of the adjacent camera in the overlap region to find the best underlying offset. This "best" offset is generated individually for each lane, as enumerated in Table 1. Then, the location of the downstream camera origin is found by subtracting the overlap from the furthest point of the upstream camera, as enumerated by lane in the six rightmost columns of Table 2.

Table 2 offers us an opportunity to quantify the overlap in the original NGSIM videos. As noted previously, NGSIM assumed that the successive cameras abutted one another, with no gap and no overlap, i.e., NGSIM assumed that in the coordinates of the upstream camera the downstream camera's origin falls exactly at the length of the upstream camera, as enumerated in the second column of Table 2. To illustrate, consider the camera 5–6 interface, NGSIM assumed there was zero overlap, thus, NGSIM assumes camera 6 coverage begins immediately after the full length of the camera 5 view from the camera 5 origin (194 ft). After accounting for the overlap, the last five columns in Table 2 shows that in fact the camera 6 coverage begins a shorter distance from the camera 5 origin (in this case 189 ft for all lanes). The top part of Fig. 1 shows a sample frame from the original NGSIM validation video from cameras 5 and 6 at the same instant. Fig. 1C shows the two views abutted against each other. Fig. 1D shows a detail after repeating the comparison using the reprojected video from the exact same frames, eliminating the pink boxes from the NGSIM validation video to clearly show the underlying vehicles and lane lines. Upon close inspection, there are a few columns of pixels that are repeated in both camera views,

[5] Note that Section 2.3 shows for ground truth and independently Section 3.3 "discovers" via the partial trajectory method that the original NGSIM distance is not accurate. But we do not use these facts at this point in the development.

Table 1
Overlap between the given extend camera pair.

Cameras	Overlap(ft)					
	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6
Camera 1&2	147	147	147.5	147.5	147.5	148
Camera 2&3	60.5	60.5	60	60	60	60
Camera 3&4	32	32	31.5	31.5	30.5	30
Camera 4&5	44	44	43	42.5	42	41
Camera 5&6	25	25	25	25	25	25
Camera 6&7	7	7	7	7	7	7

Table 2
Location of the downstream camera origin in the upstream camera's longitudinal coordinate system for the NGSIM video and the baseline calculations.

Cameras	NGSIM All lanes (ft)	Baseline (ft)					
		Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6
Camera 2 to 1	414	403	403.5	403	402.5	402.5	402
Camera 3 to 2	466	462	462	462	462.5	462.5	462.5
Camera 4 to 3	170	161.5	161.5	161.5	162.5	163	163
Camera 5 to 4	180	174	175	175.5	176.5	176.5	176.5
Camera 6 to 5	194	189	189	189	189	189	189
Camera 7 to 6	266	259	259	259	259	259	259

indicating the overlapping coverage between cameras, e.g., at this instant, vehicle 1508 is completely visible within the field of view of camera 6. However, at the same instant the last few feet of the same vehicle are still visible in camera 5, and the right end of the lane line above this vehicle is also evident in both cameras. This overlap of 10 pixels undermines the tracking between cameras when the two views are assumed to be non-overlapping. Table 3 parses the baseline overlap data to show the unaccounted overlap between successive cameras in the original NGSIM video. Coifman and Li (2017) found that the validation boxes followed the trajectories in the NGSIM dataset, while Coifman and Li (2022) showed that the original NGSIM trajectories were generated with the unaccounted-for overlap between successive cameras, i.e., the NGSIM trajectories will erroneously "speed up" while crossing a camera boundary because the distance of the overlap area is double counted in the original NGSIM coordinate system.

2.3. Establishing the ground truth alignment

While the analysis in Sections 3.1-3.3 will use the original NGSIM homographies, for validation we need an independent alignment between cameras. In the context of the NGSIM data, the natural solution would be to recalculate the homography for each camera from scratch. Unfortunately, one cannot do so effectively strictly using the NGSIM reference material. Although the NGSIM dataset included an aerial photo, there are two problems in using this photo to re-calculate the homographies. First, the NGSIM aerial photo has a large shadow where few roadway features are evident, as reiterated in Fig. 3A. The shadow falls over a large portion of the area viewed by camera 4 and camera 5, as evident in Fig. 3C. Second, as shown in Fig. 3B, the painted lane lines are not visible in this aerial photo, but the ceramic tile delineators used to supplement the lane lines are visible, suggesting that the aerial photo might not be from the visible spectrum. Whereas the background image for camera 5 in Fig. 3D shows that the opposite is true in the same region from the video, the lane lines are visible but most of the ceramic delineators are not.

In order to establish a common coordinate system and generate the ground truth homographies, a new reference image is needed. Fortunately, Google Earth includes a high resolution aerial photo of the site from May 31, 2007, which is about 2 years after the NGSIM video were collected on April 13, 2005. Thus, minimizing the possibility of changes due to repaving or repainting the lane markers. The Google Earth image was cropped to the region of the original NGSIM aerial photo. The new aerial photo has a resolution of approximately 4 pixels per ft. Fig. 3E shows the portion of the new aerial photo that corresponds to Fig. 3A. The detail image in Fig. 3F shows the clear lane line features in the area that was under a shadow in the original NGSIM aerial photo and shows that the painted lane lines are visible, providing a rich set of feature points for recalculating the homographies.

To generate a homography between a given video camera and the aerial photo it is necessary to match feature points that are clearly seen in both views. Since the vehicle trajectories were extracted in the original NGSIM image coordinates, this section finds the homography from the original NGSIM image coordinates to the new aerial photo.

The first step of the entire transform process is to calculate the homographies: H1 from the raw video plane to the original NGSIM projected image coordinates (which was done in Section 2.1), and now adding H2 that projects from the original NGSIM projected image coordinates to the new aerial photo. To ensure that there were no residual problems in the successive application of homographies H1 & H2, we also find H3 to transform directly from the raw image coordinates to the new aerial photo coordinate system. The chief distinction between the two paths is that H3 used all of the lane line features visible in raw video, whereas H1 only used the features within the region from the original NGSIM projected video.

Table 3
Overlap of the original NGSIM projected video.

Cameras	Overlap(ft)					
	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6
Camera 1&2	11	10.5	11	11.5	11.5	12
Camera 2&3	4	4	4	3.5	3.5	3.5
Camera 3&4	8.5	8.5	8.5	7.5	7	7
Camera 4&5	6	5	4.5	3.5	3.5	3.5
Camera 5&6	5	5	5	5	5	5
Camera 6&7	7	7	7	7	7	7

At this point there are three views in the aerial photo coordinate system: the new aerial photo; the raw NGSIM background image projected to the original NGSIM coordinate system and then to the new aerial photo coordinate system via successive application of H1 and H2; and the raw NGSIM background image projected directly to the new aerial photo via H3. These three views are repeated in Fig. 4A for each lane line in camera 6. The three strips for the given lane pair are sandwiched together to facilitate direct comparison, but for brevity, the view is limited to the vicinity of the ends of the lane lines visible in the new aerial photo to show the details while preserving the entire 267 ft span of camera 6. A distance reference has been added to the image with alternating red/black marks on the right of the given sandwich at 0.5 ft, corresponding to 2 pixels in the aerial photo per individual black or red stripe (or 1 ft per red/black pair, corresponding to 4 pixels in the aerial photo), and on the left at 1.25 ft (5 pixels in the aerial photo per individual black or red stripe). Fig. 4B repeats this process for all the delineators visible in the new aerial photo. The purpose of these two figures is to compare all of the matched lane features to ensure that after the homographic transformations they have the same location as the corresponding features in the new aerial photo. At close inspection it can be seen at the downstream end (image top) that the lane lines and lane delineators are "stretched" longitudinally in the projected video, which we believe is due to the lower effective resolution of the raw video in this region, as per section 2.1. Fig. 4 and the corresponding figures for all successive camera pairs (not shown) indicate that almost all of the features in the given projected background image are within 1 ft (or 4 pixels) of the corresponding feature in the new aerial photo. In other words, both transformation paths yield good performance: H1 to the original NGSIM coordinates followed by H2 to the new aerial photo coordinates, or H3 directly to the aerial photo coordinates.

Since the raw trajectories were recorded in the original NGSIM coordinate system, Fig. 4 shows that they only need to be projected to the new aerial photo coordinate system via H2. With the homography between the NGSIM image coordinates and the new aerial photo, this work projects all of the raw (i.e., without cleaning) re-extracted vehicle trajectories from Section 2.1 in the NGSIM image coordinates to the aerial photo coordinate system using the new H2 for the given camera.[6]

After projecting the NGSIM coordinates into the aerial photo coordinate system for both the camera boundaries and vehicle trajectories we now also have ground truth localization. The vehicle trajectories are recorded in both the global aerial photo coordinates and also in the respective corrected camera coordinate systems. Where the corrected camera coordinate system is simply the aerial photo coordinates after subtracting the aerial photo coordinate of the given camera's origin.

3. Methodology and evaluation

This section develops the partial trajectory methodology to align two successive cameras using the vehicle trajectories from the same lane as recorded separately in each of the camera views. The presentation alternates between one subsection developing a portion of the methodology and the next subsection evaluating the respective performance. This work uses the within camera trajectories from Section 2.1. Obviously, in practice one would use trajectories from the tracking system under evaluation. While we know from the validation data in Sections 2.2-2.3 that an overlap error and a scale factor both exist in the original NGSIM camera alignment, these facts are not known by the algorithm.

Section 3.1 develops the methodology to determine the amount of gap or overlap between two successive cameras. In practice, the partial trajectory method would then use the same analytical tools to proceed to calculating the scale factor between the two successive cameras. However, it is easiest to explain the scale factor error detection in the context of real data. So before proceeding to the scale factor part of the partial trajectory method, Section 3.2 evaluates the gap and overlap detection and in the process reveals the impacts of the scale factor error. Section 3.3 explores this error, shows that it arises from a scale factor error between successive cameras, and develops a method to detect and quantify the presence of the error. The presence of the scale factor error indicates the need for a recalibration of the original camera alignment (as was done for validation in the preprocessing, Section 2.3). Section 3.4 re-evaluates the method using the new homographies and the aerial photo coordinates from Section 2.3, which simultaneously represents recalculating the original camera alignment and verifies the results of the preceding subsections.

It is important to note that Sections 3.1-3.3 use re-extracted data in the original NGSIM coordinate system. It is only after detecting the various errors in these sections that Section 3.4 uses the data as projected into the new aerial photo coordinates.

[6] Note that the re-extraction discussed in Section 2.1 only measured the longitudinal location and lane, no effort was made to control for the lateral position within a lane, therefore, this projection assumes all the vehicles are moving in the middle of their respective lane. This assumption is consistent with the fact that the ST stacks were extracted using pixels from the middle of the lane.

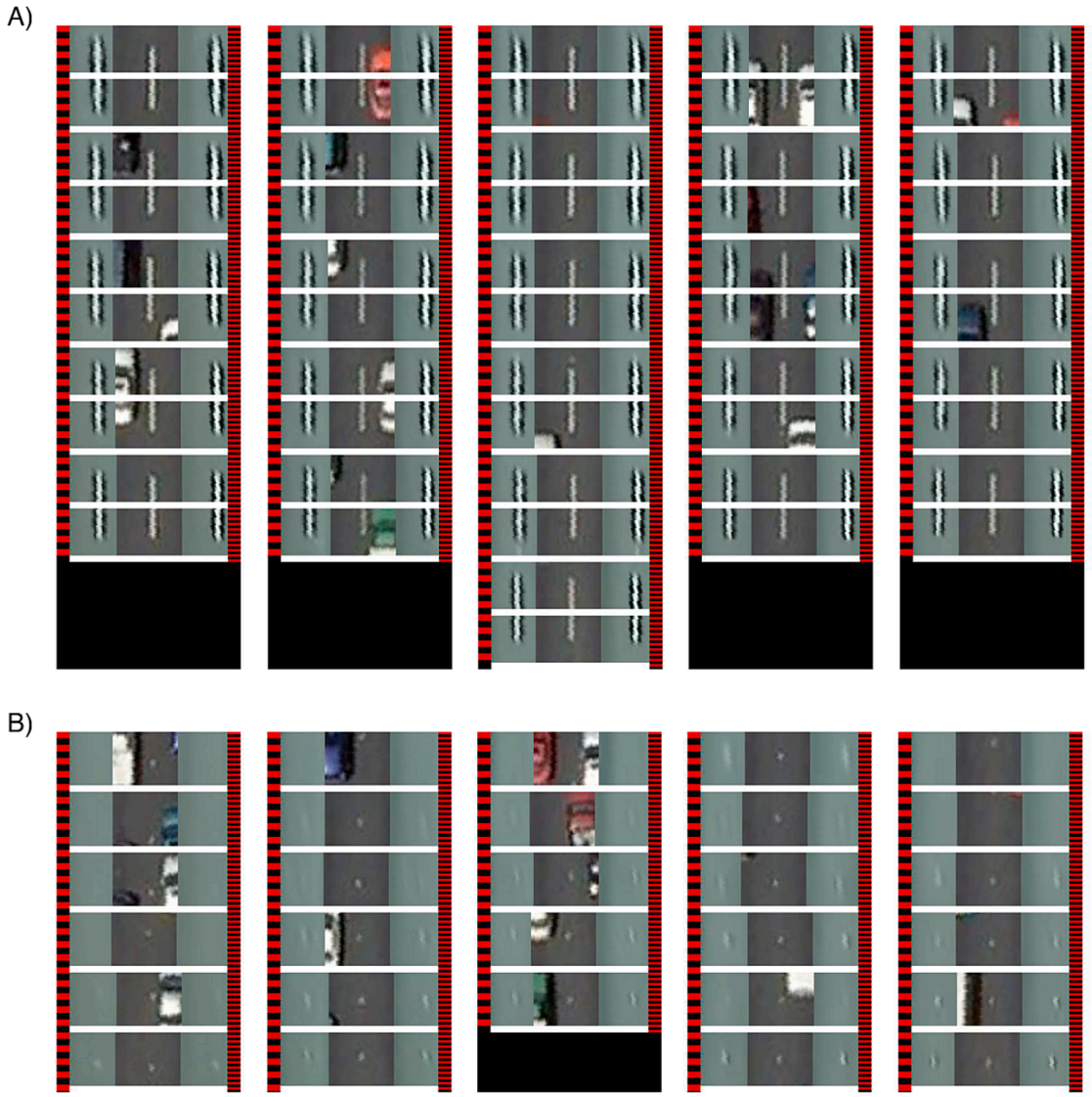

Fig. 4. Camera 6 road strip sandwich for each lane boundary. For each strip, the left column is the "projected NGSIM" coordinate frame subsequently projected to the aerial photo via H2, the middle column is the new aerial photo, and the right column is the raw NGSIM frame projected directly to the aerial photo via H3. The strips correspond progressively to lanes 1&2 on the left, to lanes 5&6 on the right. (A) The ends of all visible lane lines, and (B), ceramic lane marks between lines. A scale reference has been added with alternating red/black marks on the right of the given sandwich at 0.5 ft (2 pixels per individual stripe in aerial photo), and on the left at 1.25 ft (5 pixels per individual stripe in aerial photo). This process was applied to all seven cameras with similar results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.1. Successive camera alignment with unknown overlap or gap

This section develops the method for aligning successive video cameras that have a small but unknown gap or overlap between them. The work considers the range of gaps between −20 ft to 80 ft, where a negative gap is simply an overlap of the given magnitude. The intent is to provide alignment in the range where it is difficult for a human to detect a misalignment from a cursory inspection. It is

envisioned that the presence of a gap or overlap beyond this range would easily be detectable by a person, and upon seeing a large discrepancy, they could realign the cameras to get them within the (-20,80) ft range. Furthermore, recall that this method is intended to work with noisy trajectories, i.e., the instantaneous speed is expected to be unreliable. In this case the errors arise from uncleaned manual data reduction, but more generally, errors would arise from the unknown performance of whatever vehicle tracking method used to collect the vehicle trajectories.[7]

This approach is termed the ***partial trajectory method*** because it uses the portion of the vehicle trajectories as the vehicles pass through a detection zone near the boundary in each camera. Meanwhile all of the trajectory data outside of the given detection zone is ignored. Each detection zone is defined by a pair of cut lines: lines 1 and 2 at the end of the upstream camera, and lines 3 and 4 at the start of the downstream camera. Throughout this work, we use a spacing of 25 ft between the paired cut lines in a given camera.[8] The partial trajectories seen in the detection zones are then used to impute the amount of gap or overlap between the detection zones. The base method uses a single lane at a time.

Algorithm 1: Partial Trajectory Gap Estimation

INPUT upstream camera trajectories in target lane and location of line 2, both in upstream camera coordinates; downstream camera trajectories in target lane and location of line 3, both in downstream camera coordinates.
line_1 ← line_2 − 25′
line_4 ← line_3 + 25′
FOR each upstream camera trajectory
  UpstPtTraj ← all points in the trajectory over the detection zone between lines 1 & 2
END FOR
FOR Gap ← -20′ to 80′
  FOR each downstream camera trajectory
    DnstPtTraj ← all points in the trajectory over the detection zone between lines 3 & 4
    DnstPtTraj ← project all DnstPtTraj into upstream camera coordinates by adding (line_2 - line_3 + Gap) to the DnstPtTraj locations
  END FOR
  FOR CurUpst ← each UpstPtTraj
    v ← average speed between first point and last point in CurUpst
    IF v > 20 mph // exclude any UpstPtTraj moving slow enough that it might stop
      FOR CurDnst ← each DnstPtTraj within −10 to + 20 sec of CurUpst time
        Fit 3rd degree polynomial to the combined set of all points in CurUpst and CurDnst combined
        RSS(CurDnst) ← the RSS error across all points in the combined set via Eq. (1)
      END FOR
      Find CurDnst with the smallest RSS, keep as the match for CurUpst
      FinErr(CurUpst) ← RSS for the match
    END IF
  END FOR
  TotErr(Gap) ← sum(FinErr) across all CurUpst
END FOR
FinalGap ← Gap with the smallest TotErr

As noted earlier, the present work reprojected the NGSIM raw video to ensure that there was an overlap between successive cameras, which both allows us to establish the baseline alignment in the NGSIM coordinate system and allows us to evaluate scenarios where there really is overlap between cameras. To simulate gaps from these overlapping views we simply move the detection zone in one of the cameras away from the actual camera boundary, thus ignoring the portion of the trajectories in the intervening region to replicate the gap between non-overlapping cameras. To illustrate key elements of the partial trajectory method, Fig. 5A shows the ground truth in global coordinates for an example with a simulated 30 ft gap between a detection zone in NGSIM camera 5 and another detection zone in camera 6. In this case the gap between detection zones was artificially created by omitting the trajectories before 1250 ft in downstream camera 6 and after 1220 ft in upstream camera 5. By artificially creating gaps like this, the work preserves the true matches for validation of the partial trajectory method. This figure shows 12 vehicles that pass both cameras in the time period, and the algorithm does not know which trajectory in camera 6 is associated with which trajectory in camera 5. Although this figure shows the trajectories from each camera in global coordinates, the input to the algorithm only reports the vehicle trajectories in the respective camera's coordinate system.

Algorithm 1 presents the partial trajectory method for estimating the gap between two successive camera views in a given lane. The algorithm takes the entire set of trajectories seen in each camera, with the positions specified in the respective camera's coordinate

[7] When using another vehicle tracking method, one needs to be careful to ensure there are no systematic biases in the vicinity of the camera boundaries. For example, if one tracked the centroid of the bounding box of each target two systematic errors could arise: (1) the bounding boxes should move faster over ground than the actual vehicles because of projection errors (see Section 2.1), and (2) when a target enters/exits the given camera one end of the bounding box will capture the camera boundary rather than the respective edge of the vehicle, thus the centroid will not reflect the actual vehicle's motion. Data cleaning and smoothing techniques can also introduce systematic errors at the camera boundaries, so it is important to understand the behavior of the tracking algorithms and adjust accordingly to control for any biases.

[8] In general, the spacing between cut lines can be adjusted for a given deployment based on the trajectory time step and range of traffic speeds considered. Using a suboptimal spacing should lead to noise in the results, but this noise would readily be evident in the output of Algorithm 3 that steps through a range of possible gaps.

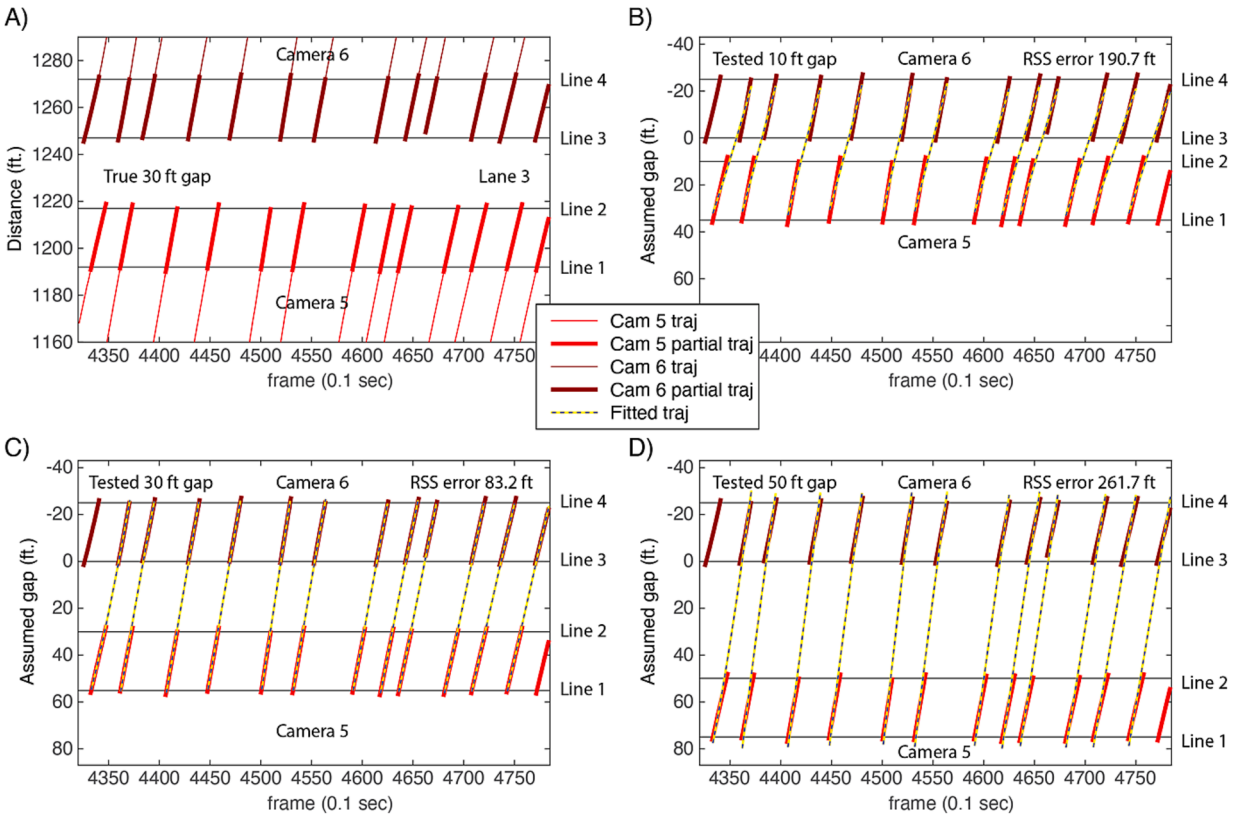

Fig. 5. (A) An example of real vehicle trajectories in lane 3 from cameras 5 and 6 in world coordinates simulating a "true" 30 ft gap. Only the bold partial trajectories are retained and the true gap shown in this plot is unknown by the algorithm. Using the partial trajectories from part A, the final matches and fitted curves are shown for these partial trajectories when testing (B) a 10 ft gap, (C) the true but unknown 30 ft gap, and (D) a 50 ft gap. The RSS error is found across the 12 fitted curves and is shown in the top right of the last three plots. In this case the RSS error is smallest when the tested gap matches the true gap. In practice, the actual RSS error calculation uses a much larger set of vehicles than this illustrative example.

system. Lines 1 and 2 are in the upstream camera coordinate system and fall near the end of the camera's view, while lines 3 and 4 are in the downstream camera coordinate system, falling near the start of the camera's view. The two camera views are assumed to be offset in global coordinates (e.g., the original NGSIM longitudinal coordinates) by an unknown gap between lines 2 and 3.

The core of Algorithm 1 consists of three nested loops, with the first loop stepping through the range of gaps under consideration (rows 7–24). The second loop (rows 12–22) steps through each of the upstream vehicles to find the "best" matching downstream vehicle via the third loop (rows 15–18). The third loop uses a search window of 30 s of data from the downstream camera to find the match to a given upstream vehicle. All of the downstream vehicles falling between 10 s before to 20 s after a given upstream vehicle passed line 1 are retained. All of the partial trajectories from the downstream camera that fall into this time window will be selected as possible matches and taken one at a time via the third nested loop. Each pass through the third nested loop compares a different downstream trajectory against the same upstream trajectory from the second nested loop. This comparison uses a third degree polynomial fit to estimate an entire trajectory under the assumption that the given downstream partial trajectory comes from the same vehicle as the upstream partial trajectory in the presence of the spatial offset from the first of the nested loops. This fitted curve is then compared to the corresponding partial trajectories recorded in each camera (i.e., between Lines 1 & 2 and separately, between Lines 3 & 4) to find the root sum squared (RSS) error over all times via Equation (1). Among this set of downstream partial trajectories, the one with the smallest RSS error is retained as the match for the given upstream vehicle. Note that the IF statement in row 14 uses the average speed in the upstream camera's detection zone to exclude slow moving vehicles, this is done because slower vehicles could come to a complete stop over the range of gaps considered and thus, result in a poor fit with a third degree polynomial even when using the true partial trajectories from the same vehicle over the true gap.[9] After finding the best match for each upstream vehicle by stepping through second loop, the final error from each of the upstream vehicles is then summed as the ***total error*** for the given gap for the first loop. Finally, the smaller the total error is across all gaps, the more compatible the partial trajectories are likely to be. Therefore, the

[9] The 20 mph threshold is a conservative value based on Wu and Coifman (2014), which found that vehicles traveling slower than 10 mph can come to a stop over a 20 ft gap between dual loop detectors. Also note that, this assessment is done only for the upstream camera, thus ensuring that no true matches in the downstream camera are discarded for the retained vehicles.

gap with the minimum total error is retained as the final gap.

$$RSSerror = \sqrt{\sum \left(X_{fitted}(t) - X_{original}(t)\right)^2} \quad (1)$$

Consider camera 5 in Fig. 5A, only the portion of the trajectories falling in the detection zone between line 1 and 2 are retained, as indicated with bold curves, while the discarded data are shown with thinner curves. Similarly, in camera 6 only the portion of trajectories in the detection zone between line 3 and 4 are retained. Thus, yielding two sets of "partial" trajectories available for alignment. The distance between line 2 and line 3 defines the effective gap between the two cameras and the data in this area are unknown by the algorithm. Since the location of the lines are known in the camera coordinates the effective gap between the pair of lines also defines the actual gap between the ends of the two cameras. As such, we drop the modifier "effective" from this point onward.

Fig. 5B-D show the results of Algorithm 1 applied to only the 12 upstream partial trajectories in Fig. 5A for three different assumed spatial offsets. Indeed, the assumed offset that corresponds to the true but unknown gap has the smallest RSS error.[10] In this case each of the 12 upstream trajectories had a true match in the downstream trajectories, and for all three of the offsets the algorithm selected the correct matching downstream trajectory, that will not always be the case, but we will return to that point in a moment. First though, note that although each upstream partial trajectory was matched to the correct downstream partial trajectory in this example, the third degree polynomial fit had the smallest total error when the assumed gap was equal to the true gap. When the assumed gap is further from the true gap the quality of the fit applied to the combined upstream and downstream partial trajectories degrades and the residuals increase. In this way, the *correct-match* vehicles serve to reduce the total error as the assumed gap approaches the true gap. More importantly, the correct-match vehicles resonate with each other, they should all reach their minimum RSS at roughly the same gap and exhibit the same systematic trends as the assumed gap varies from the best gap. As such, the fidelity greatly improves as the number of vehicles grows.

Note that throughout these calculations it is impossible to know whether a given matching pair is correct or not, but an incorrect match will usually yield a larger error than the correct one. Likewise, an incorrect gap will usually increase the RSS error for the correct matches. Algorithm 1 will always find a "best match" for each upstream partial trajectory provided that trajectory is faster than 20 mph (row 14) and there is at least one downstream partial trajectory in the search window (row 15). But the best match is not guaranteed to be the correct-match. Some upstream trajectories will not have a matching downstream trajectory, either due to that vehicle departing the lane or if the vehicle was not tracked in the downstream camera, we will refer to these as *"no-match vehicles."* Some upstream trajectories that do have a true match could be incorrectly matched to a different downstream trajectory, we will refer to these as *"wrong-match vehicles."* Consider an extreme hypothetical situation where no vehicle had a correct-match, each vehicle would contribute noise to the total error. Let b denote the expected RSS for each incorrectly matched vehicle, in this case b should simply be a function of the headway distributions in the upstream and downstream cameras, and the total error across the set of upstream trajectories should simply show random noise as the assumed gap varies. At a given gap the no-match and wrong-match vehicles are functionally the same- on average a vehicle in either set will contribute b to the total error. The key distinction between these two groups is that at some assumed gaps the wrong-match vehicles might change to correct-match vehicles, that no longer contribute noise but instead help to pull the gap to the true gap. Meanwhile, on average, the correct-match vehicles should contribute less than b to the total error. In this way, the no-match vehicles will always contribute roughly the same amount of noise to the total error, i.e., b; the correct-match vehicles will work to pull the assumed match towards the true gap, and the wrong-match vehicles have the potential to transition from the equivalent of a no-match vehicle to a correct-match vehicle as the assumed gap approaches the true gap, thereby increasing the performance of the true gap compared to other gaps.

Ultimately a single match does not give the final offset, it only provides a constraint. The power of this method comes from shifting all partial trajectories in one camera by the same amount, so that the true matches provide a strong response that is distinct from the background noise of the incorrect matches (similar in concept to the filtering process of Coifman and Krishnamurthy, 2007). Or to put it another way, the true matches will reinforce one another when the assumed gap is close to the true gap, yielding a smaller total error, while the false positives from the no-match and wrong-match vehicles will result in a larger total error due to the random noise and the expected value of this noise does not vary with the gap (unless a wrong-match vehicle transitions to a correct-match vehicle, which only improves the fidelity of the algorithm).

Regardless, since the algorithm steps through upstream trajectories, it is possible that some downstream trajectories will not be matched (e.g., the left-most trajectory in Fig. 5A) or that one downstream trajectory will be the best match to more than one upstream trajectory as a result of a wrong-match, but the algorithm as presented only considers matches in terms of the upstream trajectories so these oddities in the downstream trajectories do not directly impact the outcome. In most cases the unmodified Algorithm 1 that relies strictly on seeking matches for the upstream trajectories should be sufficient. If there are concerns about unaccounted for downstream trajectories (e.g., due to a large number of vehicles entering the lane in the vicinity of the gap) the algorithm could easily be reversed to step through the downstream trajectories in search of upstream matches. But before doing so, it is important to remember that thus far we are only considering a single lane, the first indicator of such a problem would be large discrepancies between the estimated gap

[10] In fact, the trajectories in Fig. 5 would all be rejected by the speed threshold in Algorithm 1 because their actual speeds range between 8 and 19 mph. We suppressed the speed threshold filtering for the illustrative example because it was hard to find trajectories with speeds above 20 mph that did not look like straight lines at the roughly 130 ft by 50 sec scale shown in Fig. 5. To be clear, Fig. 5 is meant strictly as an aid to explain how the algorithm works. The proof of the method is the resulting 36 independent alignments (one for each lane between each successive camera pair) over the large range of simulated true gaps, as per Section 3.4, which does use the minimum speed threshold.

from adjacent lanes. It is also important to recognize that one can use independent sample periods to evaluate the camera alignment, if the cameras are fixed Algorithm 1 should always yield the same estimated gap, so one could sample high flow periods, low flow periods, congested periods and non-congested periods to further improve the fidelity.

A few points should be noted here. First, the partial trajectory method is contingent on having an accurate homography in each camera. As will be discussed in Section 3.3, there is no guarantee that the homography will be sufficiently accurate. Fortunately, that section also illustrates how the partial trajectory method can be used to detect inconsistencies in the homographies between successive cameras. Second, this work assumes that there is no temporal offset. If we allow both time and distance to vary, then errors in time can offset errors in space, yielding a set of nearly equally strong spatial offsets that systematically drift over a small time window. To avoid the confounding impact of temporal errors, we verified there is no time offset between successive cameras and the methodology assumes there is no time offset between the two cameras.

For real-time deployments or any study that explicitly seeks to collect vehicle trajectories with sufficient fidelity to develop car following models or estimate vehicle emissions, it is reasonable to assume that the camera clocks will be synchronized. Of course, there will be other situations where it is desirable to collect trajectories across successive cameras post hoc from video that was collected for other reasons, without time synchronization. To that end, we are pursuing on-going research into a method that will simultaneously correct for spatial and temporal offsets, but that is beyond the scope of the current paper.

3.2. *Evaluating the partial trajectory method*

This section evaluates the performance of the partial trajectory method. From a practical standpoint, recall from Section 2.1 that the effective resolution within a given camera's field of view degrades as one moves further from the camera. So as a result, when cameras overlap, we keep the near field of the further camera fixed because it has a higher effective resolution in the overlap area. With this perspective in mind, the evaluation initially uses a 20 ft overlap (or as long as possible if less than 20 ft) from the extended reprojected views, and then iteratively removes 5 ft per pass from the far end of the near camera (i.e., the downstream end of the upstream camera for departing traffic) until the overlap becomes a gap and ultimately the gap reaches up to 80 ft. Fig. 6 shows a typical example of the actual cut lines used for three different gaps in a pair of successive cameras with departing traffic. Fig. 6A shows the pair of fixed cut lines for the downstream camera, i.e., lines 3 and 4 at the start of camera 6. The rest of this figure show the cut lines in upstream camera 5 as lines 1 and 2 are adjusted to create different values of the simulated true gap. Note that true location of cut line 3 in camera 5 is shown for reference but its location in the camera 5 coordinates is unknown by the algorithm. Fig. 6B shows the case when there is a 10 ft overlap between the views. Fig. 6C shows the case when line 2 and line 3 have a 20 ft gap. Fig. 6D shows the case when the gap is 50 ft. For most camera pairs the maximum gap is set to 80 ft, the most notable exception being camera 4 where the maximum gap is limited to 60 ft because this camera only spans a short range and vehicles transition from approaching to departing in this view.

Algorithm 2 summarizes the evaluation process. The true camera offset comes from the baseline columns in Table 2. On the first pass through the for loop in row 2 the algorithm takes the smallest gap considered, −25 ft, which corresponds to an overlap of 25 ft. In this implementation the cut lines are placed 5 ft from the edge of the camera, yielding a maximum overlap between the cut lines of at most 15 ft for cameras 5 and 6. This will be the starting point for evaluating the partial trajectory method. To simulate the specific true gap, rows 4–7 of the algorithm trim the trajectories in each camera so that in the given camera's coordinate system no returns are seen past the boundary of the simulated gap. Algorithm 2 then calls Algorithm 1 using the trimmed trajectories and records the final gap

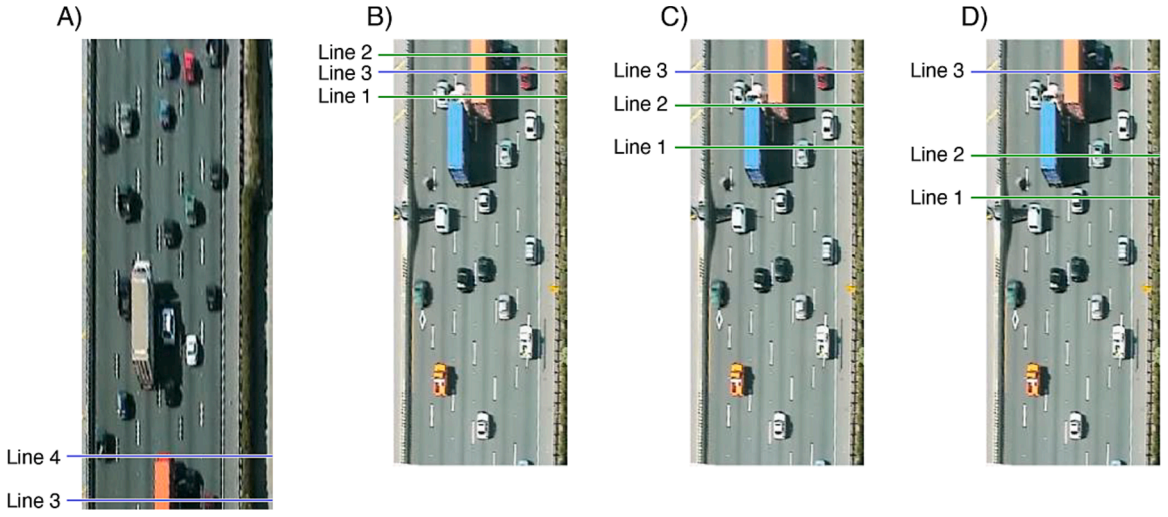

Fig. 6. Example showing the actual location of the four cut lines for camera pair 5&6. All four images come from the same instant. (A) lines 3&4 held fixed in downstream camera 6. Line 3 is also visible in the overlap area with upstream camera 5, while lines 1&2 in upstream camera 5 are shown at different simulated gaps, corresponding to (B) 10 ft overlap, (C) 20 ft gap, and (D) 50 ft gap.

reported by Algorithm 1 for the given simulated true gap (row 15), then the gap is enlarged (row 2) and the process is repeated until completing the maximum simulated true gap of 80 ft.

Algorithm 2: Evaluate Partial Trajectory Gap Estimation

```
1:  INPUT upstream camera trajectories in target lane in upstream camera coordinates; downstream camera trajectories in target lane in downstream camera coordinates; true_camera_offset
2:  FOR SimulatedTrueGap ← -25′ to 80′
      // simulate gap by deleting trajectories from the far end of the closest camera
3:    IF departing traffic
4:      line_2 ← true_camera_offset - SimulatedTrueGap − 5′
5:      Discard all points in upstream camera trajectories beyond line_2
6:      line_3 ← 5′
7:      Discard all points in downstream camera trajectories before line_3
8:    ELSE // approaching traffic
9:      line_2 ← true_camera_offset − 5′
10:     Discard all points in upstream camera trajectories beyond line_2
11:     line_3 ← SimulatedTrueGap + 5′
12:     Discard all points in downstream camera trajectories before line_3
13:   END IF
14:   RUN Algorithm 1
15:   FinalGapVector(SimulatedTrueGap) ← FinalGap
16: END FOR
17: PLOT (true_camera_offset - SimulatedTrueGap + FinalGapVector) versus SimulatedTrueGap
```

For ease of presentation, the last row of Algorithm 2 converts the calculated gaps into camera offsets. For example, if the final gap found by Algorithm 1 equaled the simulated true generated by Algorithm 2, then the resulting calculated camera offset would equal the true offset from Table 2. In this way, the curve plotted at the end of Algorithm 2 shows the calculated offset that corresponds to the calculated gaps from Algorithm 1 for the given camera pair over the entire span of simulated true gaps. The resulting calculated offset as a function of the gap between cameras 5 and 6 for each lane is shown by a bold curve in Fig. 7A, where an overlap is denoted as a negative gap. The original NGSIM assumed offset and the baseline offset from Table 2 are both shown with straight lines in the figure. Note that the vertical axis shows the offset of the camera 6 origin in the camera 5 coordinate system, so the true offset is constant regardless of the gap or overlap.

Fig. 7A shows that for a true gap between −15 ft and 0 ft all but one lane is within 1 ft of the baseline (two pixel difference) with the remaining lane just above 1 ft. As the true gap grows to 80 ft the offset error grows, reaching about 2 ft difference at a gap of 80 ft. This trend is similar for all lanes, yet one would expect there should be no gradient in the calculated offset, and especially no systematic gradient that is similar across all lanes since each lane is processed independent of all other lanes. These results are typical of all of the camera pairs: the relative error is within 1 % of the true offset for gaps below 20 ft and the partial trajectory method outperforms the offset used in the original NGSIM data over the entire range. However, the systematic gradient problem is evident in four of the six camera pairs, e.g., Fig. 7B repeats the comparison for the interface between cameras 2 and 3, only this time the curves generally show a negative slope.

3.3. *Successive camera alignment with unknown scale factors*

Fig. 7 shows that the calculated offset exhibits a gradient as the gap increases, but the optimal offset between the origin of one camera's coordinate system and the origin of the next camera's coordinate system should be constant and independent of how much of a gap is present between the two cameras. This section explores the source of the gradient and then develops a method to correct the error.

Consider what is going on in in Fig. 7 that might lead to the systematic gradients. To increase the gap, the cut lines in the further camera[11] are held constant while the cut lines in the nearer camera are moved away from the camera boundary. As the gap increases, the vehicles travel an increasing unseen distance between cut lines 2 and 3 in the near camera's coordinate system. Recall from Section 3.1 that for each vehicle Algorithm 1 fits a third degree polynomial to the matched partial trajectories from the two cameras. This polynomial fitting assumes that the two detection zones and the unseen gap between the zones are all at the same scale. If there is a small scaling difference between the two cameras the speed through the unseen gap will also be off by this scaling difference, and in turn, the estimated gap will exhibit a scaling error from the true gap. Thus, the increasing gap in the near camera should result in a gradient proportional to the scale factor error between the two cameras. An important aspect to keep in mind is that the key to the success of this scale factor analysis is to hold the detection zone fixed in one camera while varying its location in the other, thereby ensuring that the changing gap only comes from one camera.

When the gap is close to zero the impact of the scaled error will also be close to zero, and the impact will grow with the magnitude of the gap, consistent with the gradients seen in Fig. 7. The fact that the gradient is consistent across all lanes, even though each lane was processed independent of the other lanes, supports this hypothesis that the error has to do with the camera geometry. While the NGSIM documentation claims all of the cameras are at the same scale, this analysis shows that the pixel length appears to vary slightly from

[11] Downstream camera for departing traffic in camera pair 5&6, upstream camera for approaching traffic in camera pair 2&3.

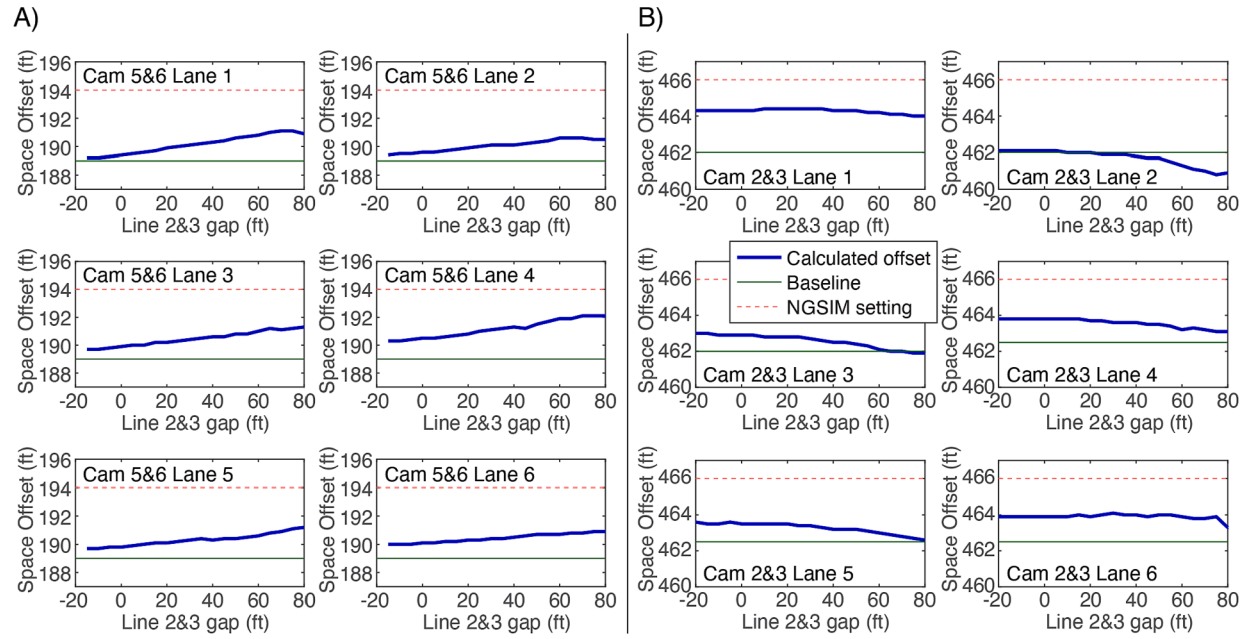

Fig. 7. Calculated spatial offsets for each lane estimated from the partial trajectory method over a range of true gaps −20 ft to 80 ft for (A) camera pair 5 and 6, and (B) camera pair 2 and 3. For reference the offsets from the baseline and the original NGSIM projections are shown with straight lines at the respective fixed offset value.

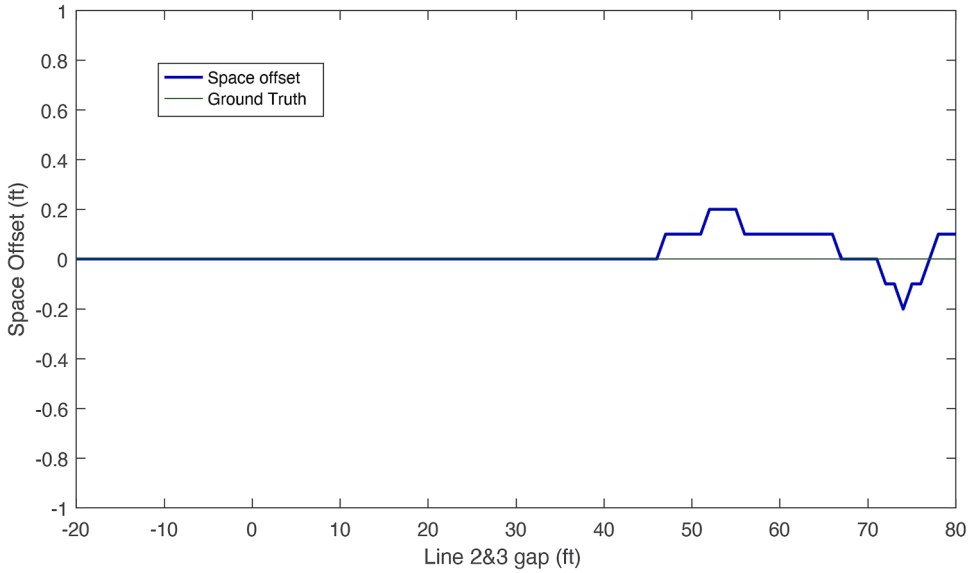

Fig. 8. Partial trajectory method to calculate the spatial offset when all four lines are within a single camera view, camera 5 lane 3 in this case. No error is seen up to a 45 ft gap and then the errors fluctuate but never beyond ± 0.2 ft.

one camera to the next. Fig. 8 puts this idea to the test by repeating the analysis from Fig. 7 via Algorithm 2, except now all four cut lines are from camera 5, lane 3, thereby ensuring that the scale factor is uniform throughout. Indeed, the resulting curve in Fig. 8 is nearly flat, with a drift of less than 0.25 ft (i.e., less than half a pixel) across the range of gaps from −20 ft to 80 ft, and what drift is evident is consistent with random noise at larger gaps.

At this point it is pertinent to establish whether the gradients seen in Fig. 7 could be due to a scale factor between successive cameras. To this end we linearly rescale the trajectories in the closer camera (i.e., the upstream camera for camera pairs 4–7 and the downstream camera for camera pairs 1–4). In the global coordinate system the rescale process employs Equation (2) where X_{raw} represents the trajectories that are to be rescaled in the corresponding NGSIM camera coordinate system, RF is the rescale factor that

will be used, and $X_{reference}$ is the reference location in that camera. Whereas in the camera coordinate system, the rescale process simply requires scaling the longitudinal points in one camera by RF. As such, the specific location of $X_{reference}$ is not critical, choosing a different location will simply result in the final calculated offset shifting by a constant value for all gaps. In light of this fact, this research arbitrarily places the reference line to be at the end of the upstream camera, recognizing that this location is typically within 20–30 ft of the true offset, and thus, will typically result in a very small fixed shift.[12]

$$X_{rescaled} = \left(X_{raw} - X_{reference}\right) \times RF + X_{reference} \tag{2}$$

Algorithm 3: Partial Trajectory Scale Factor Estimation

INPUT upstream camera trajectories in target lane in upstream camera coordinates; downstream camera trajectories in target lane in downstream camera coordinates; Xreference; minRF; maxRF.
FOR RF ← minRF to maxRF.
  Reset upstream and downstream camera trajectories to original values.
  FOR SimulatedTrueGap ← -25′ to 80′.
    // simulate gap by deleting trajectories from the far end of the closest camera.
    IF departing traffic.
      Rescale all points in the upstream camera trajectories via Eq. (2).
      line_2 ← Xreference - SimulatedTrueGap − 5′.
      Discard all points in upstream camera trajectories beyond line_2.
      line_3 ← 5′.
      Discard all points in downstream camera trajectories before line_3.
    ELSE // approaching traffic.
      line_2 ← Xreference − 5′.
      Discard all points in upstream camera trajectories beyond line_2.
      Rescale all points in the downstream camera trajectories via Eq. (2).
      line_3 ← SimulatedTrueGap + 5′.
      Discard all points in downstream camera trajectories before line_3.
    END IF.
    RUN Algorithm 1.
    FinalOffsetVector(SimulatedTrueGap) ← FinalGap - SimulatedTrueGap.
  END FOR.
  PLOT (Xreference - FinalOffsetVector) versus SimulatedTrueGap.
  StdevRF(RF) ← standard deviation of FinalOffsetVector.
END FOR.
FinalRF ← RF with the smallest StdevRF.

The resulting process to calculate the scale factor is captured in Algorithm 3, which is essentially Algorithm 2 embedded in a loop that steps through different values of RF in an effort to find the RF that corresponds to the flattest curve. Revisiting the comparison in Fig. 7A, Fig. 9A compares the resulting offset curves as a function of gap length for RF values ranging from 0.98 to 1.06 at 0.01 steps, where the bold dashed curve is the original non-scaled result. When there is no scale issue, the curve should be flat (as shown in Fig. 8); thus, if there was no scale error the distribution of offsets found at each evaluated gap should yield a standard deviation close to zero. Therefore, among these curves we choose the RF from the curve that has smallest standard deviation to be the optimal RF, which is marked by solid green curves in Fig. 9A. Note that the optimal RF is not the same for all lanes, for the interface between cameras 5 and 6 the optimal RF ranges between 1.02 and 1.04 for different lanes to flatten the respective curve, suggesting that the error process differs slightly across lanes.

This process was repeated for all successive camera pairs, e.g., Fig. 9B shows the results for cameras 2 and 3 and this time the resulting optimal RF ranges between 0.98 and 1.00. Across all lanes in all camera pairs the optimal RF spanned from 0.97 to 1.05. Most camera pairs exhibiting a spread of 0.02 across lanes, with the largest spread being 0.03 found in camera pair 6 and 7. Ultimately these gradients before rescaling reflect an error in the homographies that significantly degrade the quality of the extracted trajectories. These new-found homography errors undermine the use of any data in the original NGSIM coordinate system, including for evaluating the partial trajectory method between successive cameras.

The error evident in Fig. 7 proved to be a fortuitous discovery, yielding a tool to assess if there is an unaccounted RF between successive cameras. For simple camera alignment one could run Algorithm 1 (Section 3.1) using the full extent of the successive views to detect and quantify any gap or overlap. Then, to detect the scale factor error one would repeat the iterative process of this section: hold the detection zone in one camera fixed and progressively increase the gap by moving the detection zone in the other camera away from the end of that camera in small steps as per Algorithm 3. If the resulting trend over a range of gaps is flat (e.g., Fig. 8) then there is no evidence of a scale factor error, but if it exhibits a slope (e.g., Fig. 7) it would be indicative of a scale factor error. If any error is detected, the best path would be to redo the original camera alignment, as per Section 2.3; but if that is not an option, one could use the output of Algorithms 1 and 3 to generate correction factors and post hoc adjust the existing alignment.

[12] The one risk of blindly using the upstream camera length arises if the two camera views overlap by a lot more than 30 ft. In this case, the range of simulated true gaps in row 2 of Algorithm 3 is too high and needs to be adjusted to start below −25 ft. Fortunately, such a large overlap should be readily apparent to a cursory visual inspection.

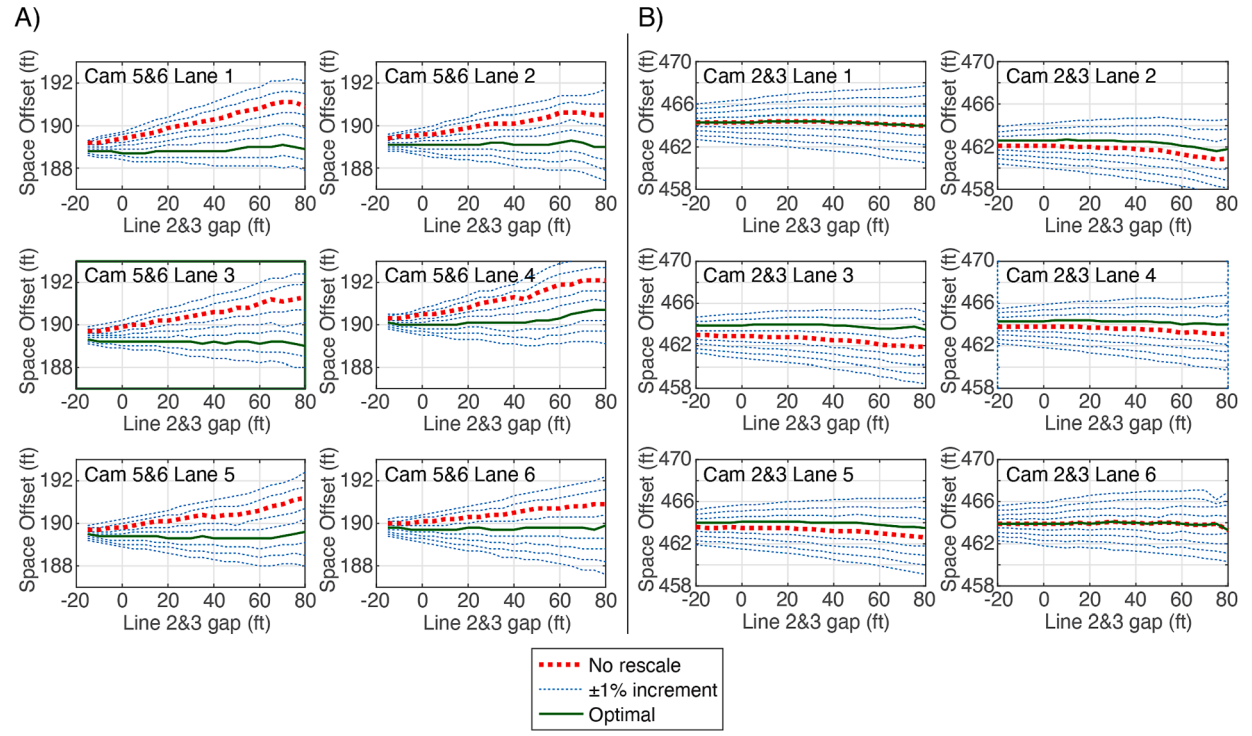

Fig. 9. Spatial offset estimations for a range of different RF values for each lane taken individually over a range of true gaps −20 ft to 80 ft for (A) camera pair 5 and 6 for RF values ranging from 0.98 (top curve) to 1.06 (bottom curve) at 0.01 steps, and (B) camera pair 2 and 3 for RF values ranging from 0.96 (top curve) to 1.04 (bottom curve) at 0.01 steps.

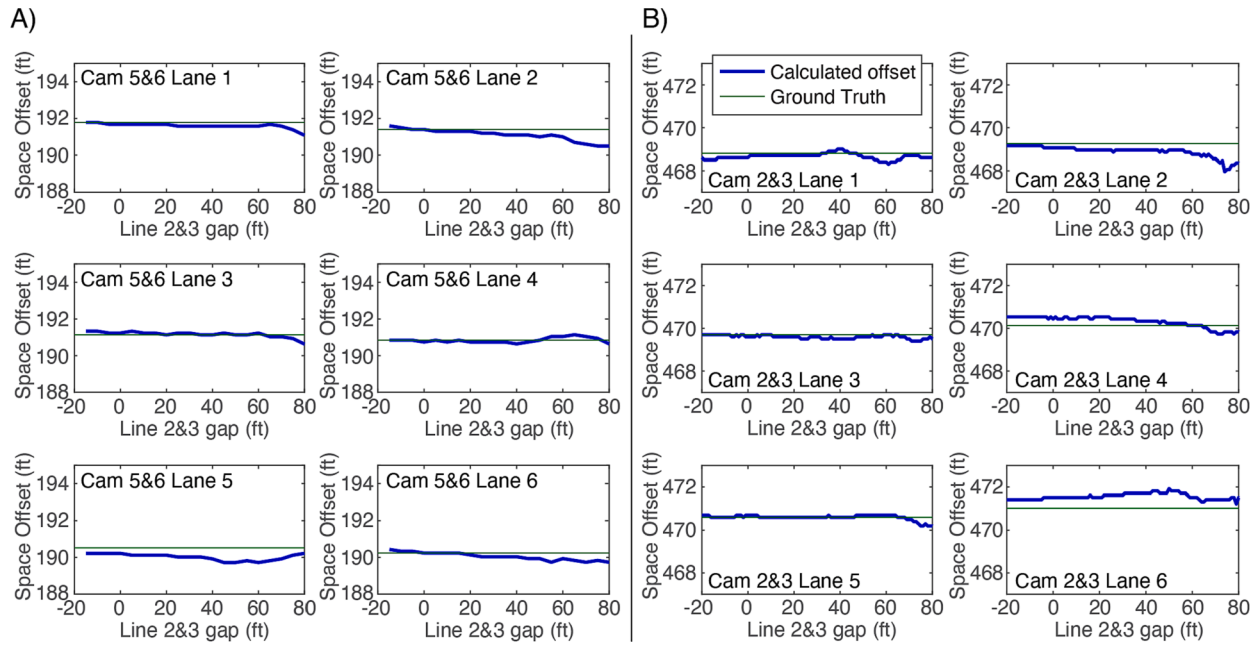

Fig. 10. After projecting all of the data into the aerial photo coordinates, the resulting calculated spatial offsets for each lane estimated from the partial trajectory method over a range of true gaps −20 ft to 80 ft for (A) camera pair 5&6, and (B) camera pair 2&3. For reference the offsets from the ground truth are shown with straight lines at the respective fixed offset value.

3.4. Re-Evaluation

As shown in Section 3.3, the partial trajectory method is sensitive to scale factor errors between the successive cameras because this method explicitly estimates travel over unseen gaps that can extend up to 80 ft in this study. This section repeats the analysis from Section 3.2 to evaluate the partial trajectory method, only now using the raw trajectories projected to the corrected camera coordinate system based on the new aerial photo.

As before, Algorithm 2 generates the full range of simulated true gaps and at each true gap Algorithm 1 is used to find the gap that yields the smallest total error. Fig. 10 shows the resulting estimated gap over the range of true gaps for each lane for two typical camera pairs: 5 and 6, as well as 2 and 3. Like Section 3.2, the bold curve shows the calculated offset as a function of the gap in the given lane and the thin solid line shows the ground truth camera offset as calculated from the global aerial photo coordinate system. Of note is the fact that the calculated offset curves in these plots no longer exhibit the systemic gradients that were so prominent when using the original NGSIM homographies. Fig. 11 shows the percent relative error for all camera pairs, all lanes using Equation (3). The magnitude of all of the relative errors are below 1 %, and almost all of them smaller than 0.6 %.

$$\text{percent relative error} = \frac{measurement - truth}{truth} \times 100 \tag{3}$$

4. Results

To evaluate the partial trajectory method, we begin by combining the manually extracted trajectories across cameras in the aerial photo coordinate system and then smoothing them. Section 4.1 provides the details of this processing. Then, using the smoothed re-extracted trajectories, Section 4.2 explores the quality of both the old and new data in the aerial photo coordinate system. With the full scope of the methodology now presented, Section 4.3 discusses the parameters and assumptions embedded in the methodology.

4.1. Trajectory smoothing

The manual vehicle tracking was conducted individually in each lane in each camera using the original NGSIM coordinate system and the extended views. While the original NGSIM dataset only tracked vehicles that traversed the entire segment (i.e., a given vehicle must both enter and exit the link) during the video recording period, ignoring any vehicles that remained in the link at the start or end of the period. Our work tracked all vehicles seen in the video, regardless of whether they were still in the link at the start or end of the video recording. To minimize projection errors from the camera view, the rear end of vehicles were tracked forward in time for cameras 4–7, while the front end of vehicles were tracked backward in time for cameras 1–4 (Coifman and Li, 2017). With the front and rear trajectories in camera 4, each vehicle's length is measured as it passes the "straight down" view in camera 4 (967–987 ft in the final coordinate system). Then to maintain the reference point on the front of a given vehicle (as per the original NGSIM dataset), the vehicle's measured length is added to the trajectory of its rear recorded in cameras 5–7. Meanwhile, in camera 4 the front trajectory is used upstream of 967 ft, the shifted rear trajectory is used downstream of 987 ft, and a spline fit is used for the handoff in between.

In the middle of the freeway segment the road curves slightly to the left. As a result, if one drew a line perpendicular to the roadway at the start and another at the end of the surveillance region, the distance between the two lines in the outside lane would be longer than the inside lane. In relative terms, the difference of this distance between lane 1 and lane 6 is less than 1 %. On the other hand, if one used separate longitudinal distance metrics on a lane by lane basis, a vehicle leaving one lane would enter the adjacent lane at a different longitudinal distance. We do not know if the original NGSIM dataset deliberately addressed this issue, but it ultimately used a global longitudinal distance across all lanes. Since our relative distance discrepancy is under 1 % it was decided to also use a common longitudinal distance for all lanes. Our reference being the lane boundary between lanes 3 and 4, at the center of the road. To find the longitudinal position in the aerial photo coordinates the raw vehicle positions are projected laterally to this centerline in the NGSIM coordinate system before the trajectory is projected to the aerial photo coordinate system.

After projecting the raw data into the aerial photo coordinates the camera overlap ensured that the trajectories from successive cameras would also overlap for a short distance during the handoff between cameras. In these overlap areas we retain the trajectories in the far camera since these have the highest effective resolution of the two overlapping views. At this point the longitudinal trajectory of a given vehicle is combined across cameras. To maintain fidelity during stop-and-go movements where conventional filtering techniques would blur the transition to/from a stop, the stop region is identified and isolated following Coifman and Li (2017). Then, using the technique from Chang et al. (2023), the given trajectory is smoothed using a Rauch-Tung-Striebel smoother (Rauch et al., 1965). Finally, most vehicles were tracked upstream of 0 ft and downstream of 1800 ft, but decreasing resolution, increased occlusions and increased projection errors reduces the tracking accuracy beyond these thresholds, so in the final data only the trajectories between 0 and 1800 ft are retained. The net result is that vehicles are tracked roughly 260 ft further than the original NGSIM dataset.[13] As noted previously, more vehicles were tracked. Like Coifman and Li (2017), we also tracked vehicles that were in the link at the start or at the end of the time period.

Although Coifman and Li (2017) also demonstrated lateral tracking, in the manual extraction most vehicles were only tracked longitudinally. So in the present work, we maintain the original NGSIM lateral position for vehicles that were tracked in the original

[13] The original dataset has an effective longitudinal tracking range from 65 ft to 1640 ft in the new coordinate system.

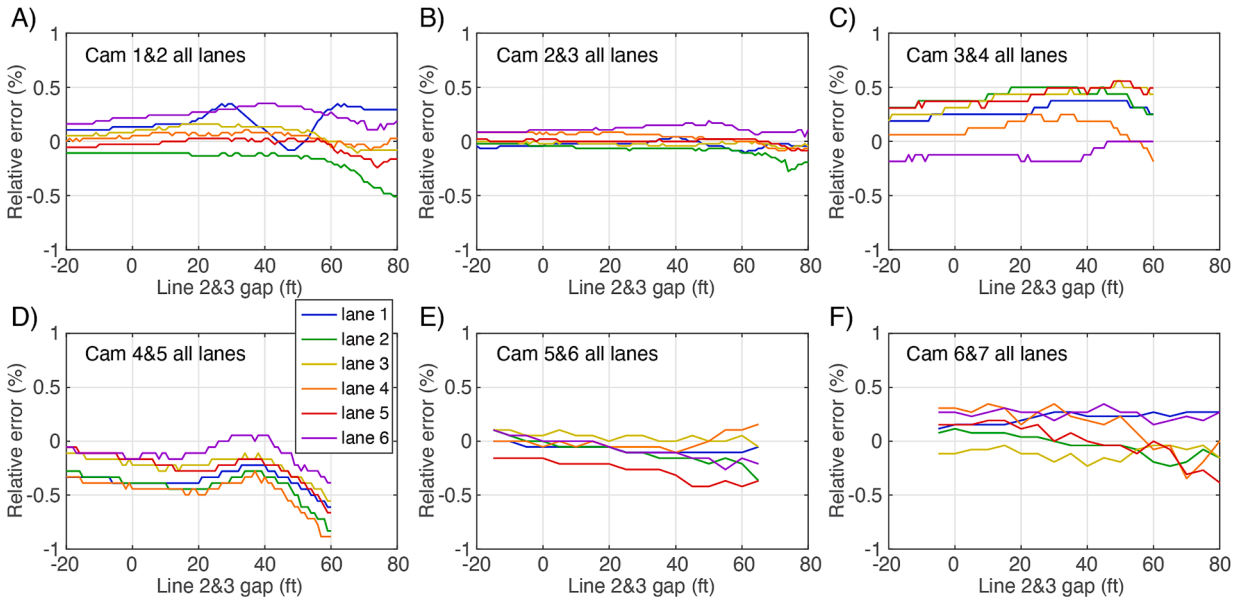

Fig. 11. Relative errors from the partial trajectory method applied to each lane over a range of true gaps −20 ft to 80 ft for all camera pairs, all lanes, showing that almost all errors are below 0.6%.

data, meanwhile, for the vehicles that were only tracked for the first time in this work the lateral position is simply recorded at the resolution of which individual lane the vehicle is in at the given instant. As a result, the time of any lane change recorded in the original dataset is unchanged in the new dataset. Obviously no lane change is instantaneous and there is evidence that vehicles would sometimes travel next to one another in a given lane (especially in the merge area where the on-ramp joins lane 6 and the lane width exceeds 18 ft). So there are times in lane 6 between roughly 600 and 900 ft when two vehicles are really in the same lane with an overlapping longitudinal distance, but are offset by a small lateral distance. Fortunately, these events are readily evident by a small spacing in the trajectory data and if they are critical to a given application, one can refer to the video to assess further. The resulting smoothed re-extracted data are used in Section 4.2 to assess the data quality. As of the publication of this paper the re-extracted data will be available at Coifman (2023).

4.2. Data quality

As noted in Section 1, the presence of an undetected gap or overlap will lead to large distortions in vehicles' speed and acceleration in the vicinity of the camera boundaries. Once we know what to look for, the NGSIM camera boundary errors become evident using even very coarse measures on the original NGSIM data, for example, measuring median speed and separately median acceleration of all vehicles as a function of longitudinal distance. For this work we use successive 1 ft bins and record the speed or acceleration from each vehicle as it passes a given longitudinal distance bin. Fig. 12A shows the median v(x) from the original NGSIM data while adding the individual camera coverage with vertical dashed lines denoting the respective camera boundaries. This plot shows one curve for each lane. The trends in the different lanes are very similar, with the exceptions that lane 1 (HOV lane) shows speeds much higher than the other lanes, while lane 6 shows speed drop in camera 2, in advance of the onramp. As expected, the original NGSIM speed systematically exhibits large peaks near each camera boundary in most lanes because the assumed distance traveled due to double counting the overlap area is greater than the actual distance traveled. In contrast, Fig. 12B shows that the systematic peaking is absent in the re-extracted data. Meanwhile, Fig. 12C shows median accel(x) for the original NGSIM dataset. Here we see large peaks near most of the camera boundaries, consistent with the speed data. Once more, Fig. 12D shows that the systematic peaking is absent in the re-extracted data. These plots underscore the fact that although it is a simple measure, median v(x) and median accel(x) are powerful tools for detecting systematic errors.

Inspection of the trajectories also reveals significant improvements. Fig. 13A shows a sample of the original NGSIM trajectories with dashed curves, superimposed on top of the re-extracted trajectories with sold curves. The boundary between a given pair of successive cameras is indicated with a horizontal dashed line and the specific camera numbers are shown on the right of the plot. The original data roughly spanned from 65 to 1,640 ft, while the new tracking is truncated at 0 and 1,800 ft, as evidenced by the re-extracted data extending beyond the spatial range of the original data. Several errors are evident in the original NGSIM data, with many of the large errors highlighted with shaded regions, numbered 1–11. Region 1 shows an example of the piecewise constant speeds that Coifman and Li (2017) found in camera 6, while regions 2, 3, 4 & 6 show this problem persists in other cameras. However, too much fluctuation in the original NGSIM data is also evident in region 5. Several discontinuities are evident near camera boundaries (regions 4 & 7–11), but random jumps also evident (regions 2 & 6). Looking closer at the data, Fig. 14A shows a detail of region 3 while

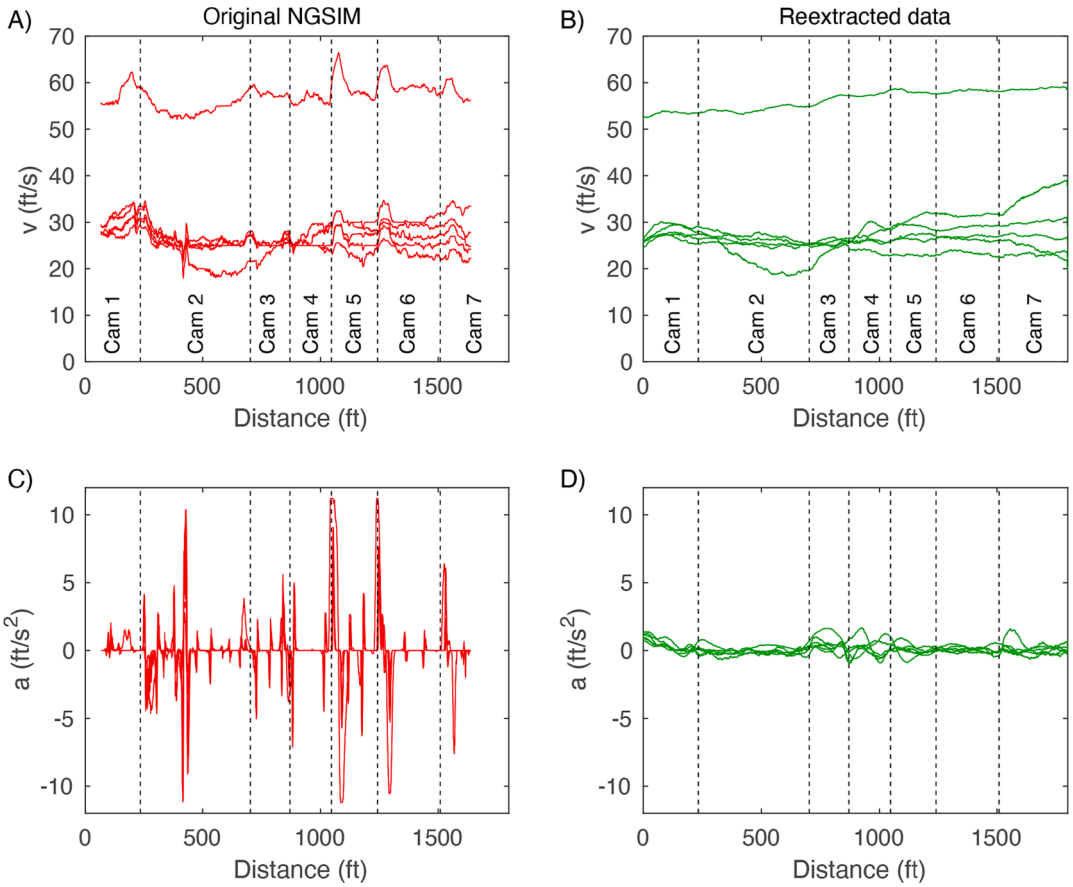

Fig. 12. Median speed across all vehicles by lane as a function of longitudinal position from, (A) the original NGSIM data, and (B) the re-extracted data. The corresponding median acceleration by lane from, (C) the original NGSIM data, and (D) the re-extracted data.

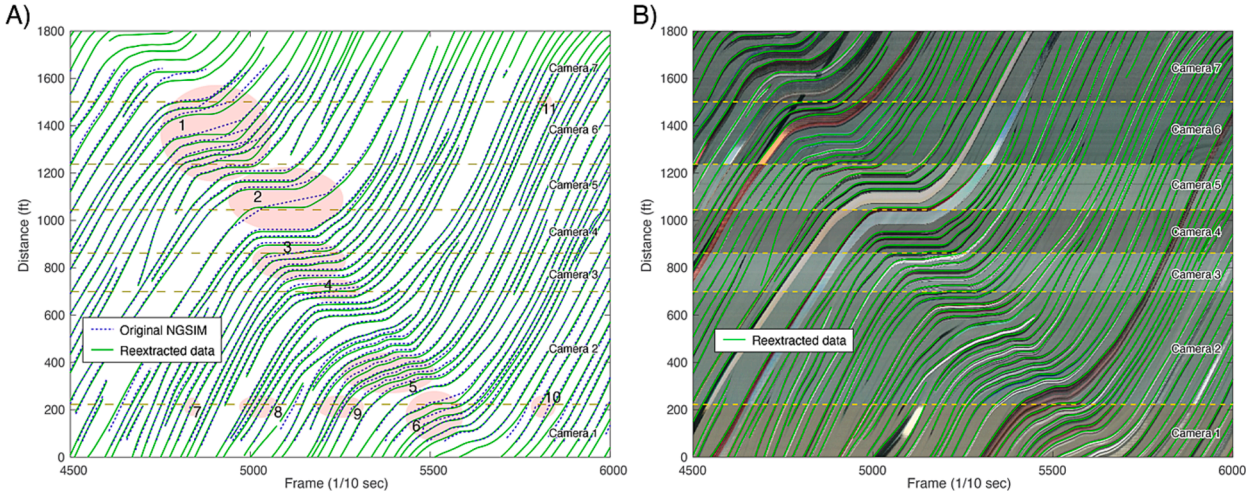

Fig. 13. (A) A comparison of typical vehicle trajectories over the entire segment from the original NGSIM and re-extracted data, in this case from lane 3. The re-extracted data extends an additional 225 ft and (not shown) adds trajectories for the vehicles that were in the link at the start and end of the video sequence. The shaded regions highlight various errors in the original data, as discussed in the text. The camera used for tracking over the given spatial range is shown on the right while the boundaries between cameras are denoted with dashed horizontal lines. (B) A comparison of typical re-extracted vehicle trajectories (green curves) superimposed on the underlying spatio-temporal stacks, illustrating the high fidelity of the re-extracted trajectories. To facilitate evaluation, this figure shows the same lane and region of the time–space plane that was used in part A. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

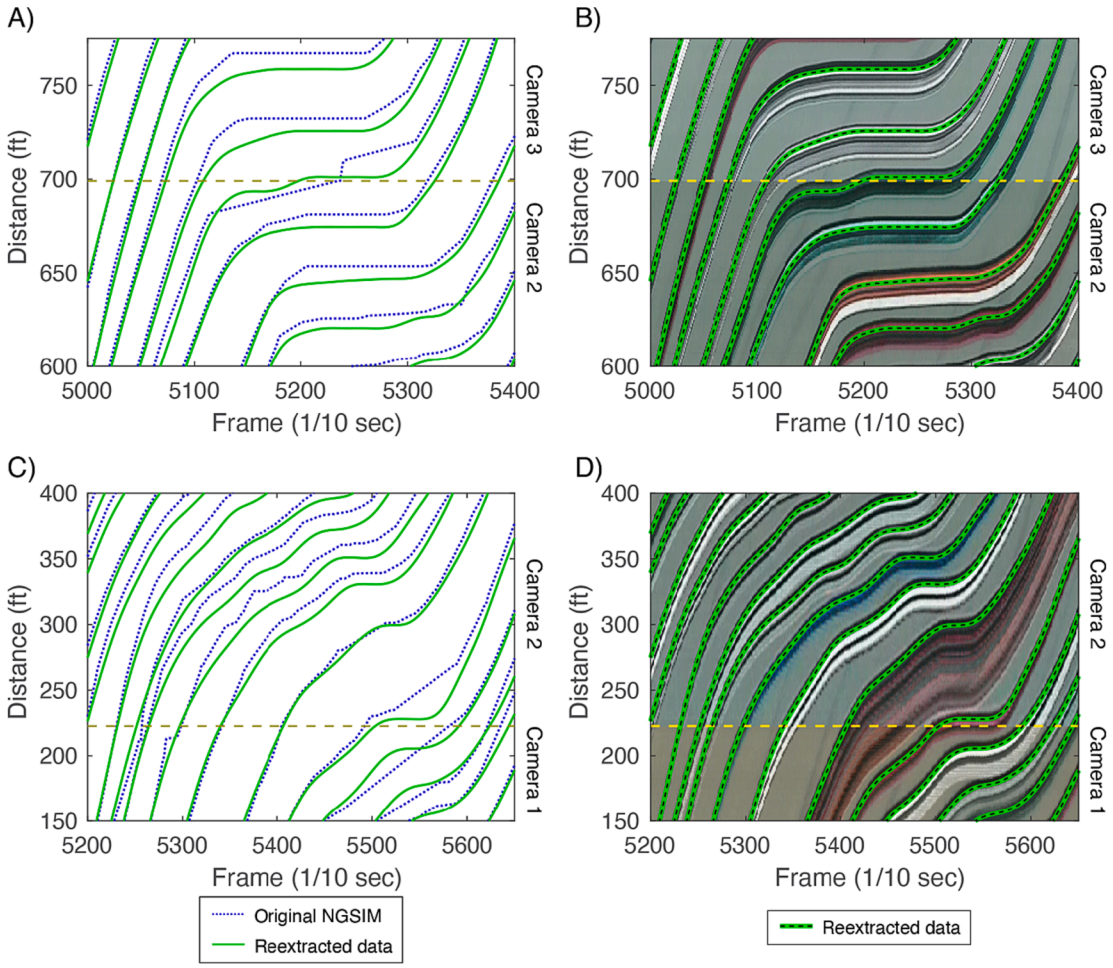

Fig. 14. Details from Fig. 13 highlighting the improvements in the re-extracted data (A) region 3 from Fig. 13A, (B) the corresponding region of Fig. 13B, (C) portions of regions 5, 6 and 9 from Fig. 13A, (D) the corresponding region of Fig. 13B.

Fig. 14C shows portions of regions 5, 6, and 9.

Meanwhile, Fig. 13B superimposes the re-extracted trajectories on top of the ST stacks (Coifman and Li, 2022). To facilitate comparison, this plot repeats the same time–space region shown in Fig. 13A. Fig. 14B&D repeat the respective comparisons for the detail examples. The trajectories show high fidelity to the underlying ST stacks and these results are typical of the entire dataset. The benefits of correcting the errors become even more apparent in terms of the individual vehicle speeds. Fig. 15 highlights the improved fidelity, where Fig. 15A shows the time series v for all of the original NGSIM trajectories from lane 3 that were tracked in frame 5000. Fig. 15B shows the same vehicles only now from the re-extracted data. The noise in the original NGSIM data speed measurements blurs the overlapping time series together, rendering it impossible to pick out individual vehicles, meanwhile, the time series v is clear and distinct in the corresponding re-extracted data.

4.3. Parameters and assumptions

At low speeds a vehicle trajectory can exhibit a very complex shape over a short distance, which is why Algorithm 1 skips any upstream trajectory moving slower than 20 mph. The third degree polynomial offers sufficient flexibility to fit most trajectories that can be observed given the minimum speed threshold of 20 mph and maximum gap of 80 ft, e.g., a 20 mph vehicle will take less than 3 sec to cross an 80 ft gap at a constant speed, which is not enough time for the driver to make a major change in acceleration (excluding rare extreme events such as an emergency braking maneuver). This time window is an upper bound, most retained vehicles will be traveling faster and most gaps considered will be shorter. It is also worth noting that at the time scale necessary to show multiple vehicle trajectories, e.g., as done in Fig. 5, the real fluctuations in an individual vehicle trajectory traveling at speeds above 20 mph while traversing 100 ft will be difficult to perceive by eye.

In undertaking this work we did not seek to find optimal parameter settings. The initial development was based upon the basic equations of motion and reasonable assumptions of maximum absolute acceleration. As such, the methodology should be insensitive to

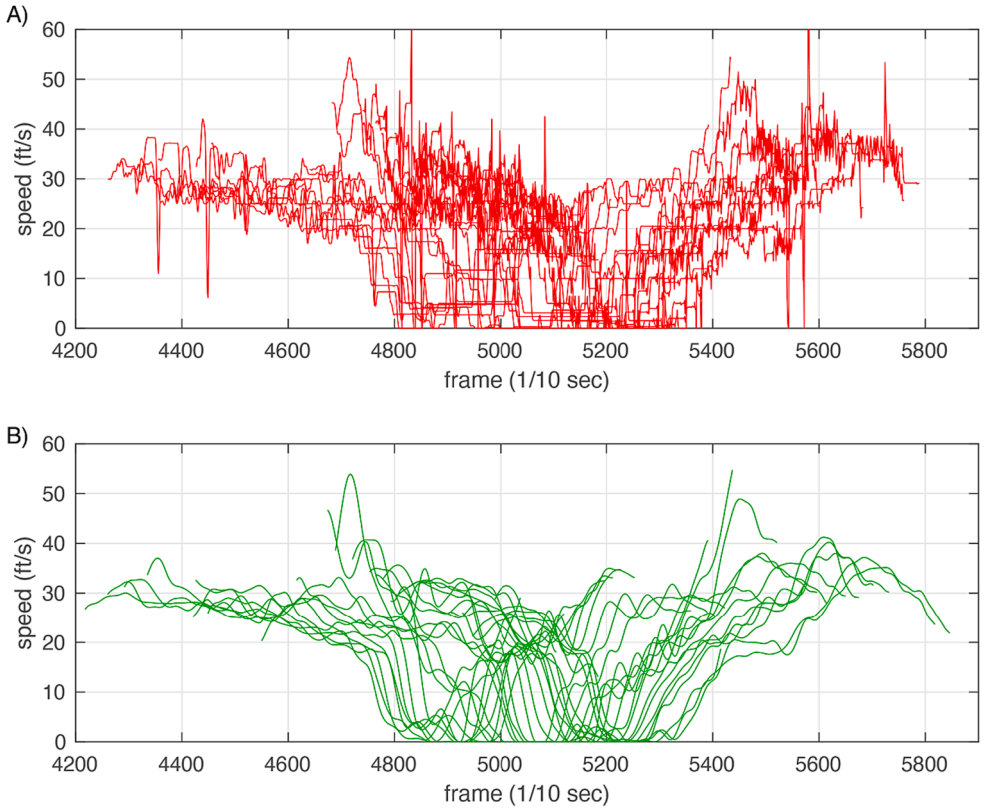

Fig. 15. (A) Time series speed for all of the original NGSIM trajectories that were in lane 3 during frame 5000 (see Fig. 13). (B) Repeating the plot for the same vehicles only now using the re-extracted trajectories. The noise in the original data blurs the overlapping vehicle data together whereas the progression of the stop wave through successive vehicles is clearly evident in the re-extracted data.

small changes in the parameters, e.g., the performance should not change significantly if the cutoff speed was set to 25 mph or 15 mph, or if the maximum gap considered by the algorithm was extended to 100 ft. However, the larger the real gap becomes, the more likely the algorithm performance will degrade simply because the longer duration of a given vehicle's partial trajectory the more likely acceleration could cause it to deviate from the fitting assumptions. One can already see these impacts in the results, with performance drop at larger true gaps, e.g., the fluctuations seen in Fig. 8 for gaps above 40 ft. Although the performance of estimating the exact gap degrades as the true gap expands, the error is still much smaller than the true gap, so the algorithm should be able to detect the presence of a gap between successive cameras far beyond the 80 ft range considered herein even if it loses the ability to accurately quantify the exact size of a large gap.

If one wanted to also extend the ability to accurately quantify the size of a large true gap beyond 80 ft, a simple task to mitigate the increased variability arising from longer dwell times between lines 1 and 4 would be to follow basic sampling theory and use more vehicles for the alignment, i.e., a longer sample period. Such a move is not unreasonable, the results presented herein were limited to the trajectories seen in less than 17 min due to the short duration of the NGSIM video. If more data were available, it would be reasonable to require 30 or 60 min of data for calibration rather than 17 min. Although we do not have the data to test it, we suspect performance could greatly improve during lower flow free flow periods because the larger and irregular headways between vehicles will provide unique patterns that will reinforce the true matches in Algorithm 1. Coifman and Cassidy (2002) applied this concept to vehicle lengths to reidentify vehicles between loop detector stations while Lee and Coifman (2015) used vehicle headways instead of vehicle lengths to synchronize clocks between concurrent LIDAR and loop detectors. While Lee and Coifman assumed the spatial offset fixed to find the optimal time offset, the present work holds time fixed to find the optimal spatial offset.

At any rate, the possibility that the algorithm's performance might degrade as the real gap grows beyond 80 ft is already addressed by the stated scope of this work, namely, the intent is to provide alignment in the range where it is difficult for a human to detect a misalignment from a cursory visual inspection. The presence of a true gap greater than 80 ft should easily be detectable by eye, without the need for the present algorithm.

5. Conclusions

This paper developed the partial trajectory method to align the views from successive fixed cameras that are used for video-based vehicle tracking across multiple camera views. The method is envisioned to serve as a validation tool of whatever alignment has

already been performed between the cameras. The strength of the method is that it operates on the output of vehicle tracking in each camera rather than secondary features visible in the camera view that are unrelated to the traffic dynamics (e.g., fixed fiducial points). Thereby providing a direct feedback path from the tracking to ensure the quality of the alignment in the context of the traffic dynamics. The methodology seeks to address situations where it is not clear whether successive cameras exhibit a small gap or a small overlap. The method uses vehicle trajectories within successive camera views along a freeway to deduce the presence of an overlap or a gap between those cameras and quantify how large the overlap or gap is. Vehicle trajectories are sampled at the abutting boundaries of a given pair of successive cameras and the assumed gap between cameras is varied to find the gap that maximizes the trajectory consistency between the two cameras. This approach also accounts for the possibility of an overlap (simply a negative gap), so it is not necessary to know if there is a gap or an overlap a priori.

The partial trajectory method was then expanded to test for scale factor errors between successive cameras. While it is envisioned that for simple camera alignment one would run the partial trajectory method in Algorithm 1 using whatever the actual gap might be, by progressively moving one of the detection zones away from the other camera to expand the effective gap (e.g., lines 1 and 2 in Fig. 5) via Algorithm 3, it can reveal the presence of scale factor errors. If there are no scale factor errors the resulting alignment between cameras should not change with an expanding gap (e.g., Fig. 8 and Fig. 10), but if the scale differs between cameras by even a small amount, the alignment will seemingly drift with the expanding gap (e.g., Fig. 9). To quantify the new-found scale factor error in the NGSIM homographies and verify our analysis worked, we found a new aerial photo for the NGSIM site and recalibrated the homographies to both verify our results in the original NGSIM coordinates and to demonstrate that the methodology works well on well calibrated cameras, correctly showing the absence of any error.

No matter what system is used to align cameras, when there is camera overlap the quality of the final alignment can be verified by comparing the common or referenced features (e.g., Fig. 4). For this reason, we strongly recommend any field deployment that seeks to collect vehicle trajectories across multiple cameras should also deliberately employ overlapping fields of view so that the overlap area can be used to refine and validate the final alignment. When the views perfectly abut there is no redundancy available to validate the alignment. If there is a gap between the views then at best, a portion of the vehicle trajectories will go unseen as they traverse this gap. Regardless, if an alignment error goes undetected it will distort the vehicle trajectories (e.g., Fig. 12). The best solution is to undertake a thorough and accurate georeferencing before aligning cameras.

There is always some true alignment between cameras. For this paper we make a first order approximation that this alignment is linear in the projection space, as is implicitly assumed whenever using homographic projection. Of course, the assumptions necessary for homographic projections do not always hold, e.g., when the terrain is not sufficiently flat or there are significant non-linear distortions by the camera. Our method does not seek to address any of these non-linear distortions.

Provided conventional homographic projection is appropriate for the given camera deployment, our method seeks to assess the quality of the alignment between successive cameras. In general, when using the partial trajectory method, if any error is detected, the best path would be to redo the original camera alignment; but if redoing the alignment is not an option, one could use the output of Algorithms 1 and 3 to generate correction factors and post hoc adjust the existing alignment. After applying the method to the first period of the NGSIM I-80 data Fig. 11 shows that all of the relative errors are below 1 %, and almost all of them are smaller than 0.6 %.

To evaluate the impact of the partial trajectory method on the actual trajectory data, the manually re-extracted data were projected into the aerial photo coordinates and then smoothed. It was shown that the boundary errors evident in the original NGSIM speed and acceleration data are rectified in the re-extracted data. The re-extracted data shows much greater fidelity to the actual vehicle motion. The re-extracted data also tracks the vehicles over a 14 % longer distance and adds 23 % more vehicles that were not tracked in the original NGSIM because they were in the segment at either the start or end of the video sequence (472 new / 2034 old). As of publication, the re-extracted data from this paper will be released to the research community (see Coifman, 2023).

While this paper relied on the NGSIM video datasets to develop and evaluate the analytical tools, the trajectories are simply an input to the algorithm. The resulting partial trajectory method transcends the dataset and should be applicable to most methods that seek to extract vehicle trajectories across multiple cameras. The fact that this work was effective on the low resolution NGSIM data is particularly encouraging, since most contemporary video datasets are likely to be at a much higher resolution, which will only serve to improve the fidelity of the trajectory data used as input.

Finally, this work assumes that there is no temporal offset between successive cameras. If we allow both time and distance to vary, then errors in time can offset errors in space, yielding a set of nearly equally strong spatial offsets that systematically drift over a small time window. To avoid the confounding impact of temporal errors, the manual evaluation verified there was no time offset. Accommodating an unknown time offset in addition to the unknown spatial offset is the subject of on-going research.

CRediT authorship contribution statement

Benjamin Coifman: Conceptualization, Methodology, Formal analysis, Investigation, Visualization, Supervision, Funding acquisition. **Lizhe Li:** Conceptualization, Methodology, Formal analysis, Investigation, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This material is based in part upon work supported in part by the National Science Foundation under Grant No. 1537423 and 2023857. The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. This report does not constitute a standard, specification or regulation.

References

Barmpounakis, E., Geroliminis, N., 2020. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. Transp. Res. C 111, 50–71.
Battiato, S., Farinella, G.M., Furnari, A., Puglisi, G., Snijders, A., Spiekstra, J., 2015. An integrated system for vehicle tracking and classification. Expert Syst. Appl. 42 (21), 7263–7275.
Castañeda, J.N., Jelaca, V., Frías, A., Pizurica, A., Philips, W., Cabrera, R.R., Tuytelaars, T., 2011. In: Non-Overlapping Multi-Camera Detection and Tracking of Vehicles in Tunnel Surveillance. IEEE, pp. 591–596.
Chang, Y., Xiao, W., Coifman, B., 2023. Using Spatiotemporal Stacks for Precise Vehicle Tracking from Roadside 3D LiDAR Data. Transp. Res. C 154. https://doi.org/10.1016/j.trc.2023.104280.
Chiabaut, N., Buisson, C., Leclercq, L., 2009. Fundamental diagram estimation through passing rate measurements in congestion. IEEE Trans. Intell. Transp. Syst. 10 (2), 355–359.
Choudhury, C., Ramanujam, V., Ben-Akiva, M., 2009. Modeling Acceleration Decisions for Freeway Merges. *Transportation Research Record: Journal of the Transportation Research Board*, Issue 2124, 45–57.
Coifman, B., 2015. Empirical Flow-Density and Speed-Spacing Relationships: Evidence of Vehicle Length Dependency. Transportation Research- Part B 78, 54–65.
Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance. Transp. Res. C 6 (4), 271–288.
Coifman, B., Cassidy, M., 2002. Vehicle Reidentification and Travel Time Measurement on Congested Freeways. Transp. Res. A 36 (10), 899–917.
Coifman, B., McCord, M., Mishalani, R., Iswalt, M., Ji, Y., 2006. Roadway Traffic Monitoring from an Unmanned Aerial Vehicle. IEE Proc. - Intell. Transp. Syst. 153 (1), 11–20.
Coifman, B., Krishnamurthy, S., 2007. Vehicle Reidentification and Travel Time Measurement Across Freeway Junctions Using the Existing Detector Infrastructure,". Transportation Research- Part C 15 (3), 135–153.
Coifman, B., Li, L., 2017. A Critical Evaluation of the Next Generation Simulation (NGSIM) Vehicle Trajectory Dataset. Transp. Res. B 105, 362–377.
Coifman, B., Li, L., 2022. A new method for validating and generating vehicle trajectories from stationary video cameras. IEEE Trans. Intell. Transp. Syst. 23 (9), 16227–16236. https://doi.org/10.1109/TITS.2022.3149277.
Coifman, B., Lyddy, D., Skabardonis, A., 2000. The Berkeley Highway Laboratory- Building on the I-880 Field Experiment. Proc. IEEE ITS Council Annual Meeting, IEEE, pp. 5–10.
Coifman, B., Li, L., Xiao, W., 2018. Resurrecting the Lost Vehicle Trajectories of Treiterer and Myers with New Insights into a Controversial Hysteresis. Transp. Res. Rec. 2672 (2018), 25–38.
Coifman, B., 2023. Data Sets. https://u.osu.edu/coifman.1/data-sets/.
Dixon, M., Jacobs, N., Pless, R., 2009. An efficient system for vehicle tracking in multi-camera networks. In: In *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–8.
Hoogendoorn, S., Van Zuylen, H., Schreuder, M., Gorte, B., Vosselman, G., 2003. Microscopic traffic data collection by remote sensing. Transp. Res. Rec. 1855 (1), 121–128.
Hou, Y., Edara, P., Sun, C., 2014. Modeling Mandatory Lane Changing Using Bayes Classifier and Decision Trees. IEEE Trans. Intell. Transp. Syst. 15 (2), 647–655.
Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S., 2006. Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans. Pattern Anal. Mach. Intell. 28 (4), 663–671.
Jelača, V., Castañeda, J.O.N., Frías-Velázquez, A., Pižurica, A., Philips, W., 2011. Real-time vehicle matching for multi-camera tunnel surveillance. In: Real-Time Image and Video Processing 2011, vol. 7871. International Society for Optics and Photonics, p. 78710R.
Kanhere, N.K., Birchfield, S.T., 2008. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. IEEE Trans. Intell. Transp. Syst. 9 (1), 148–160.
Kim, J.S., Hwangbo, M., Kanade, T., 2008. In: Motion Estimation Using Multiple Non-Overlapping Cameras for Small Unmanned Aerial Vehicles. IEEE, pp. 3076–3081.
Kovvali, V., Alexiadis, V., Zhang, L., 2007. Video-Based Vehicle Trajectory Data Collection, *Proc*. Of the 86th Annual TRB Meeting.
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: In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125.
Laval, J., Leclercq, L., 2008. Microscopic Modeling Of The Relaxation Phenomenon Using A Macroscopic Lane-Changing Model. Transp. Res. B Methodol. 42 (6), 511–522.
Lee, H., Coifman, B., 2015. Using LIDAR to Validate the Performance of Vehicle Classification Stations. J. Intell. Transp. Syst. 19 (4), 355–369.
Makris, D., Ellis, T., Black, J. (2004). Bridging the gaps between cameras. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (Vol. 2, pp. II-II). IEEE.
Meng, D., Li, L., Liu, X., Li, Y., Yang, S., Zha, Z.J., Gao, X., Wang, S., Huang, Q., 2020. Parsing-based view-aware embedding network for vehicle re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7103–7112.
Punzo, V., Montanino, M., Ciuffo, B., 2014. Do we really need to calibrate all the parameters? Variance-based sensitivity analysis to simplify microscopic traffic flow models. IEEE Trans. Intell. Transp. Syst. 16 (1), 184–193.
Rauch, H.E., Tung, F., Striebel, C.T., 1965. Maximum likelihood estimates of linear dynamic systems. AIAA J. 3 (8), 1445–1450.
Salvo, G., Caruso, L., Scordo, A., Guido, G., Vitale, A., 2017. Traffic data acquirement by unmanned aerial vehicle. European Journal of Remote Sensing 50 (1), 343–351.
Thiemann, C., Treiber, M., Kesting, A., 2008. Estimating acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. Transp. Res. Rec. 2088, 90–101.
Treiterer, J., Myers, J., 1974. In: The Hysteresis Phenomenon in Traffic Flow, *Proc., Sixth International Symposium on Transportation and Traffic Theory*, pp. 13–38.
Wu, L., Coifman, B., 2014. Vehicle Length Measurement and Length-Based Vehicle Classification in Congested Freeway Traffic. Transp. Res. Rec. 2443, 1–11.
Zhang, T., Jin, P.J., 2019. A Longitudinal Scanline Based Vehicle Trajectory Reconstruction Method for High-Angle Traffic Video. Transportation Research Part c: Emerging Technologies 103, 104–128.

Zhao, J., Qi, F., Ren, G., Xu, L., 2021. Phd learning: Learning with pompeiu-hausdorff distances for video-based vehicle re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2225–2235.
Zheng, Z., Ahn, S., Chen, D., Laval, J., 2011. Freeway traffic oscillations: Microscopic analysis of formations and propagations using Wavelet Transform. Transport. Res. Part b: Methodol. 45 (9), 1378–1388.
Zhou, Q., Aggarwal, J.K., 2006. Object tracking in an outdoor environment using fusion of features and cameras. Image Vision Comput. 24 (11), 1244–1255.