A New Method for Validating and Generating Vehicle Trajectories From Stationary Video Cameras

Benjamin Coifman and Lizhe Li

Abstract—**Image processing based vehicle tracking is a powerful tool for monitoring traffic, but it is error prone. Relatively small errors that impede measuring time-series speed and acceleration can be hard to detect, e.g., 1 m positioning error in a 100 m long trajectory. This paper presents an efficient approach to separate the positioning errors from vehicle travel for evaluating image processing based vehicle trajectories. The approach starts with a spatiotemporal slice, STS, which is effectively a visual time-space diagram sampled from the video. This work skews the STS to flatten a given trajectory, eliminating the vehicle travel recorded in the trajectory. Positioning errors that were imperceptible relative to the distance traveled become readily apparent in the flattened track. Thus, providing a means to quickly assess reported trajectories from almost any image processing system against the true vehicle positions in the original video data. Recognizing that the flattening process works both ways, if errors are evident in a given trajectory, the STS method can also be used to quickly fix them. Thereby providing a path to accurate instantaneous speed and acceleration throughout the given trajectory. Alternatively, one can use this process to generate vehicle trajectories directly from the STS. While the main focus is longitudinal tracking, the process can also be used to assess (extract) the lateral position of a given vehicle. The method is evaluated using the NGSIM, Cityflow and UA-DETRAC datasets, in each case it is shown how this work can increase the fidelity of the given dataset.**

Index Terms—**Freeway traffic, image processing, microscopic traffic, road transportation, traffic flow theory, vehicle trajectory.**

I. Introduction

THIS paper presents a simple and efficient approach for evaluating image processing based vehicle trajectories. The first objective of this work is to provide a method to verify the performance of almost any image processing based vehicle tracking system from a stationary camera (potentially after image stabilization). The second objective is to show how the methodology can be used to correct errors in the trajectories and yield precise localization with sufficient fidelity to measure instantaneous speed and acceleration. The tertiary objective of this work is to demonstrate that this method could be used to generate trajectories outright.

Image processing is an important tool for tracking roadway vehicles over long distances, be it 50 m through the view of one camera or 500 m across the views from multiple successive cameras. The resulting trajectories have become a cornerstone for the empirical study of traffic dynamics and driver behavior. Car following models, fuel consumption models and vehicle emissions models all require positioning accurate enough to measure instantaneous speed and acceleration. Unfortunately, it is very difficult to assess the accuracy of empirically collected vehicle trajectories since image processing is typically the only sensor used in these studies. Even when multiple sensors are employed, image processing is usually the most accurate sensor over the large spatial range. Although automated image processing is a powerful tool, it is also imperfect, suffering from localization, grouping and segmentation errors that are exasperated by shadows, lighting conditions, and projection errors. The large distance traveled in the trajectories only serves to hide many tracking errors, e.g., a positioning error of 1 m could easily become imperceptible at the scale of a 100 m long trajectory. Only large errors are evident at the typical scale used to present a given vehicle trajectory, while smaller errors can persist undetected. To date the only viable evaluation tools are "reasonableness" tests of the resulting trajectories, e.g., assessing whether the vehicle spacing or instantaneous acceleration is feasible [1]–[4] but these tests will not detect errors that fall within the region of "reasonable" behavior. Alternatively, one could turn to a labor intensive process of manually following each tracked vehicle [5], [6].

This paper develops a technique for evaluating image processing based vehicle trajectories by separating the spatial positioning errors from the actual travel of a given vehicle. The approach starts with the spatiotemporal slice, ***STS***, method, which effectively constructs a visual time-space diagram, ***TXp***, by sampling a specific line of pixels along the roadway in every frame. The sampled scan-line from a given frame becomes a column of pixels in the STS image, i.e., a "slice" in the spatial dimension, and slices from successive frames are "stacked" in temporal order to give rise to the visual TXp that includes a visual track for each vehicle that passes through the video. Throughout this paper we use the term, ***trajectory***, to denote the extracted coordinates of a given vehicle over time, and the term, ***track***, to denote the colored stripe in the STS that corresponds to a given vehicle.

Manuscript received 31 May 2020; revised 30 December 2020, 9 June 2021, and 21 September 2021; accepted 2 February 2022. Date of publication 10 June 2022; date of current version 12 September 2022. This work was supported in part by the National Science Foundation under Grant 1537423 and Grant 2023857. The Associate Editor for this article was V. Punzo. *(Corresponding author: Benjamin Coifman.)*

Benjamin Coifman is with the Department of Civil, Environmental and Geodetic Engineering, and the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: coifman.1@osu.edu).

Lizhe Li is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: li.4120@osu.edu).

Digital Object Identifier 10.1109/TITS.2022.3149277

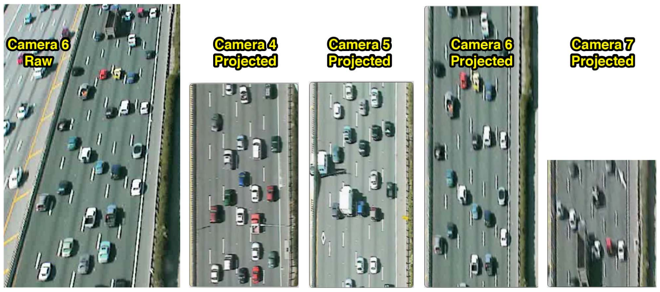

Fig. 1. Sample frames of the original video.

To illustrate the construction of an STS, consider the "camera 6 projected" image in Fig. 1. This image is one frame of a 10 Hz video of freeway traffic collected from a fixed-mount camera [7]. Fig. 2 focuses strictly on the third lane from the left, with three sample frames shown in Fig. 2A. Taking just one column of pixels from the center of the lane shows the instantaneous location of all vehicles in the given frame. Repeating this process by exacting the exact same column of pixels from each frame in the video and presenting them in sequential order gives rise to the STS in Fig. 2B. Now vehicle 1456 sweeps out a vehicle track that captures its progression along the road in pixels. This process can then be repeated independently for each lane. In this case the video was projected into the ground-plane from the raw camera view shown on the left of Fig. 1, thus the ordinate of the STS is proportional to distance in the world; however, as will be shown later, the method can also be applied in the image plane.

The STS methodology was originally developed to study natural phenomena and has gone by many names, including: "spatiotemporal slice," [8], [9]; "picture lines," [10]; and "time stack," [11]. In the context of vehicle tracking it has been called: "Spatio-Temporal Images," [12]; "spatio-temporal slices", [13]; "intensity flow," [14]; "spatiotemporal map," [15], [16]; or "cross-section imagery," [17]. Most of these papers focus on tracking the movement of features in one dimension along a straight line or pre-specified curve in a fixed view, but some consider tracking in two spatial dimensions [12], [13], [17]. The STS method is not limited to fixed cameras, many others have applied the same concept to moving cameras [18], [19]. While the STS method produces a visual TXp along the length of the scan-line, the STS method retains the spatial extent of a given trajectory, and as previously noted in the context of general image processing approaches, many positioning errors are not evident at this resolution.

The present research recognized that the STS can be transformed to eliminate the actual travel of the vehicle recorded in the trajectory, and only retain the positioning errors. Thereby providing a means to assess and validate the trajectories from almost any image processing system against the true vehicle positions seen in the original video data. Specifically, as presented in Section III.A, for a given vehicle of interest we use its reported trajectory as a reference to spatially skew the STS in such a way that in the coordinate system of the skewed STS image the reported trajectory becomes a flat line parallel with the time axis, e.g., the dashed line in Fig. 2C. In other words, we effectively skew the STS image to view it from the perspective of a "moving observer" that travels along the reported trajectory. If the reported trajectory is correct, then the corresponding track in the skewed STS image should also be perfectly flat. Otherwise, the remaining undulations in the track reflect errors in the reported trajectory. This approach has the beneficial feature that in the transformed coordinate system of the skewed STS, one only needs to consider a spatial range on the order of the vehicle's length, e.g., Fig. 2D, rather than the entire distance that the vehicle traveled. Now small positioning errors that were imperceptible relative to the extent of the actual travel become readily apparent in the form of wobbles and ripples in the flattened track of the subject vehicle in the skewed STS image. Since the track and trajectory are flattened, there is no limit to the spatial extent along the road over which this validation could be applied.

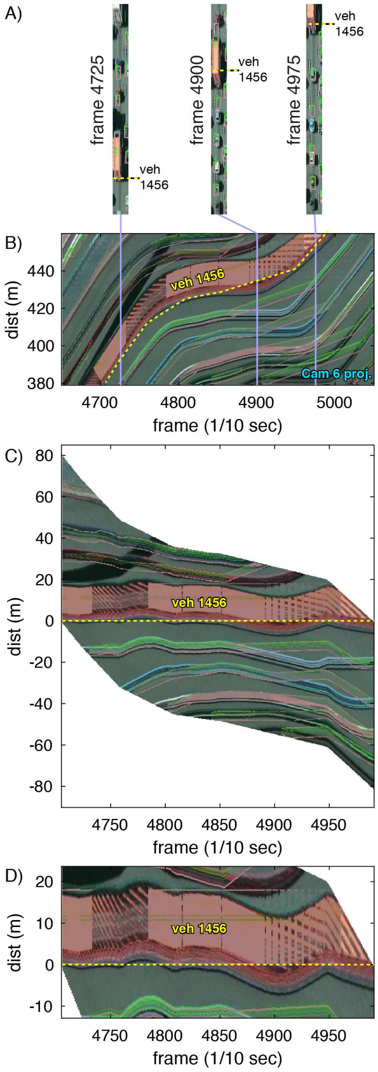

Fig. 2. Single camera STS, (A) sample frames, (B) STS, (C) skewed STS, (D) detail of C showing trajectory performance.

Recognizing that the flattening process works both ways, if any errors are evident in a given reported trajectory, the STS method can also be used to quickly fix them. Specifically, the flattened track in the skewed STS image can be used to improve the trajectory by shifting select frames up or down to eliminate any wobbles in the track and improve the flushness of the flattened track. Of course the exact same spatial shifts in each frame need to also be applied to the corresponding time points in the reported trajectory that was used to skew the STS in the first place, thereby improving the accuracy of the given trajectory.

Alternatively, one can use this process to generate the vehicle trajectories directly from the STS, without any prior vehicle tracking. For each vehicle, simply manipulate the STS to flatten the given track. The resulting shifts made to the STS to flatten the track yield the trajectory of the vehicle over the corresponding time window.

Once the longitudinal positions have been cleaned (or extracted if starting from the STS) the process can be repeated to assess (extract) the lateral position of a given vehicle using a moving row of pixels perpendicular to that used in the original STS. Obviously, this moving row of pixels should follow the given longitudinal trajectory, but it can be offset by a fixed number of pixels, e.g., to follow the center of a vehicle rather than an arbitrary feature on the front or rear of the vehicle.

The remainder of this paper presents the detailed process. Section II presents the necessary background. Section III develops the methodology in the context of validating reported trajectories across single and multiple camera views and then extends the methodology to clean the reported trajectories or generate new trajectories outright. It is shown how the work can be extended to evaluate the lateral component of trajectories, as illustrated for a lane change maneuver. The paper closes in Section IV with a brief discussion and conclusions.

II. Background-Setting the Context

This section sets the context for the analysis. Section II.A presents the data used in this study. Then Section II.B discusses which features should be tracked.

A. Data for This Study

This paper uses a portion of the Next Generation SIMulation, ***NGSIM***, data set [7] and the associated video files to illustrate the methodology, but the principles could be used on almost any video image processing based vehicle tracking where the original video is available. The choice of using NGSIM for illustration is based on the fact that the NGSIM data is the largest set of empirical microscopic traffic data available to the research community. The NGSIM project released four data sets and this paper uses one of them: the I-80 data set, which was collected in the Berkeley Highway Laboratory (BHL) [20]. The I-80 study used seven fixed mount cameras on top of a 30-story building to collect vehicle trajectories at 10 Hz along 500 m of freeway. The cameras were numbered in order from upstream to down. Cameras 1-3 viewed eastbound vehicles as they approached the building. Near the start of the view from camera 4 these vehicles passed the building and transitioned to a departing view that continued through the remaining cameras, 5-7. With the release of the data the NGSIM researchers shared the raw video, e.g., as shown for camera 6 on the left side of Fig. 1 and the video projected into the ground-plane, as shown for cameras 4-7 on the right side of Fig. 1. All five of the images in Fig. 1 are from the same instant. The projected video includes pink boxes superimposed on top of the video for all tracked vehicles for validation purposes. Note that Fig. 1 shows the first frame of the respective video, so it does not include any of the pink boxes since they are assigned to vehicles while they are in cameras 1 and 2. These pink boxes are evident in the sample frames of Fig. 2A. The pink boxes have been verified to correspond to the trajectories in the NGSIM database, and the pixels in all of the projected I-80 videos were sized to be roughly 0.15 m in the longitudinal direction [5]. The NGSIM researchers believed that they cropped the camera views precisely such that there was no gap and no overlap between successive projected camera views, i.e., they thought that the first row of pixels in one camera picks up exactly 0.15 m after the last row pixels in the previous camera. For the NGSIM examples we use the video shared from the original study. It is not known if any camera calibration was done by those researchers beyond calculating the homography from the image plane to the ground-plane.

B. Gaining Perspective

One of the first steps in collecting high quality trajectory data is to know where to measure the trajectories. In the empirical study of traffic dynamics and driver behavior, typically image processing based vehicle trajectory measurement collects the video from a high vantage point and then projects the recorded video-stream to the "ground-plane" using a homographic projection. Implicit in this ground-plane projection is the unrealistic assumption that the entire view is of features that are strictly in the ground-plane. Consider the "camera 6 raw" frame on the left of Fig. 1 to the corresponding projection to the ground-plane in "camera 6 projected" frame, second from right in the figure. The vertical number of pixels spanned by a given vehicle in the *raw* image is a function of both the height and length of the vehicle. The projection to the ground-plane ignores the vehicle height, resulting in projection errors. Thus, features that are actually above the ground-plane get projected like shadows on the ground-plane further away from their actual location above the road. The projection error increases with the height of the feature, i.e., the closer the feature is to the ground the smaller the projection error will be. So for this work we use the edge of the projected vehicle that is closest to the camera since this feature will generally be the closest feature to the ground, i.e., the bottom of the front of an approaching vehicle or the bottom of the rear of a departing vehicle. In this way the front and rear are best tracked separately. If one were tracking over the transition from approaching to departing vehicles the front and rear are both visible with a short duration of overlap as a vehicle passes the camera location. During this overlap period the trajectories of a given vehicle's front and rear can be associated with one another, separated by the vehicle's length. Since the front and rear provide separate trajectories, for the NGSIM data we arbitrarily chose to strictly track the rear of departing vehicles (NGSIM cameras 4-7, as per Fig. 1). One can easily reverse the process to track approaching vehicles (NGSIM cameras 1-4).

For this study we also break from conventional image processing techniques. Rather than try to eliminate shadows we leverage them. If the shadow is always visible it is a good candidate for our tracking since it is explicitly in the ground-plane and thus, cannot exhibit a projection error from the viewing angle and any casting error from the sun will not change over the typical period a vehicle is in view. So for the NGSIM video this work exploits the shadows that form the upstream end of the tracks in the NGSIM STS to minimize most of the projection errors.

III. Methodology

When capturing traffic dynamics, relatively small positioning errors in a vehicle trajectory can be severely detrimental to measuring instantaneous speed and acceleration. Because they can be small, the displacement from these errors can persist in reported vehicle trajectories without being detected. We will illustrate these ideas using the NGSIM data set. However, the analysis developed herein is transferrable to any similar trajectory data collected using video image processing tools, as will be illustrated in Section III.B.

Section III.A explains the method for skewing the STS and evaluating the reported trajectory. Section III.B demonstrates how to reduce the positioning errors by flattening the given track and improving the fidelity of the positioning data. This flattening can be done starting from the skewed STS in III.A or directly from the STS without any reference trajectory. Finally, Section III.C shows how the method can be extended to measure lateral trajectories too.

A. Back on Track- Validating Tracking Performance

We now return to the task of separating the real spatial travel from the spatial positioning errors. We start by considering just a single camera and arbitrarily pick camera 6 for this purpose. Since we are interested in validating ***reported trajectories*** that came from any image processing technique, we use the existing NGSIM database for this purpose. In tandem, we use the existing projected videos from NGSIM for our validation (example frames shown on the right side of Fig. 1). Each lane is processed separately, as shown for lane 3 in Fig. 2A. As noted previously, the projected video includes pink boxes denoting the location of the tracked vehicles in that frame as recorded in the trajectories.

The pink boxes are evident in Fig. 2A but might be hard to discern given the quality of the image. For illustration purposes throughout Fig. 2 the rear of the pink boxes for vehicle 1456 are shown with a dashed line. In any event, the pink boxes are not critical to our analysis, what matters is the reported trajectory that the pink boxes represent. If one were to watch the projected video the positioning error in frame 4900 of nearly 5 m would be readily apparent; but watching validation video is subjective and labor intensive. Despite the fact that the NGSIM data have been used in hundreds of traffic dynamics studies, these positioning errors have largely gone undetected.

The first step in our analysis is to sample the same column of pixels in each frame and stack these columns sequentially to form an STS, Fig. 2B, which as noted previously is effectively a visual TXp. The NGSIM trajectory for vehicle 1456 is superimposed on the STS with a dashed line. This figure starts to replace the need for manually watching the projected video to catch longitudinal tracking errors, since the reported trajectory pulls away from the upstream end of the STS track. Plotting the trajectory on top of the STS like this will reveal large deviations, but at the scale of 80 m shown in Fig. 2B smaller deviations are not apparent. As the span of road shown in the STS increases, the harder it becomes to discern the deviations between the reported trajectory and associated track.

In this case there is one position measurement in the trajectory per frame in the video, and by extension, per column of pixels in the STS. Flattening the reported trajectory for vehicle 1456 in Fig. 2B and for each point in the trajectory the associated column of pixels is shifted with it, yielding Fig. 2C. The resulting skewed STS is twice as tall as the original STS, but all of the information about vehicle 1456 falls within a range of about 20m. Zooming in to focus strictly on the vertical range of this vehicle, the deviations between the reported trajectory and the corresponding track in the STS become readily apparent in Fig. 2D. By projecting the STS to the perspective of a moving observer that travels with the reported trajectory like this, we have effectively removed the actual travel of the vehicle recorded in the trajectory, and only retain the positioning errors. If there were no errors in the reported trajectory the upstream end of the track (corresponding to the shadow directly below the rear of the truck) should perfectly follow the reported trajectory. Instead, the flattened trajectory ranges from -1.2 m to +7.3 m deviant from the associated undulating track. This simple plot allows for a rapid assessment of the accuracy of the reported trajectory in a single glance. Although this example only captures 80 m of travel, since the vertical range in the skewed STS excludes the actual travel distance by the vehicle, there is no limit to the length of roadway that can be validated. Even in the presence of lane change maneuvers, one could use a weighted average of the STS in adjacent lanes to follow the target vehicle as it maneuvers across lanes.

Now consider the case of tracking vehicles across successive cameras to demonstrate the fidelity over longer distances. Combining the concurrent STS from the four cameras in Fig. 1 yields Fig. 3A, where the horizontal dashed lines show the boundary between a given pair of successive camera views. The reported trajectory from vehicle 1456 is shown once more, along with the trajectory of vehicle 1539. First consider vehicle 1456, the combined STS spans roughly 230 m and at this scale the deviations between the reported trajectory and corresponding track evident in Fig. 2B are harder to see. So once more we skew the STS by flattening trajectory 1456 and take the respective columns of pixels with it, yielding Fig. 3B. Now the trajectory is flat and the boundaries between cameras have become monotonically decreasing curves. Fig. 3C zooms in to the vertical range of this vehicle to study the longitudinal deviations between the reported trajectory and the observed track. The undulating behavior seen in Fig. 2D persists through all four cameras in Fig. 3C. The trajectory rarely travels with the upstream end of the track, indicating a positioning error in almost all time steps, and the trajectory almost never moves parallel to the upstream end of the track, indicating that it is not a simple fixed offset.

A new problem reveals itself in Fig. 3C where the upstream end of the track crosses a camera boundary, as highlighted with the three rectangles. At each camera boundary the track jumps downstream by as much as 2 m. But the track simply captures what is in the video, the trajectory used to skew the track must be wrong. In other words, these discontinuities indicate that the reported trajectory jumps upstream as it goes from one camera to the next. Assuming the clocks are synchronized between the cameras, this upstream jump indicates that the vehicle covers

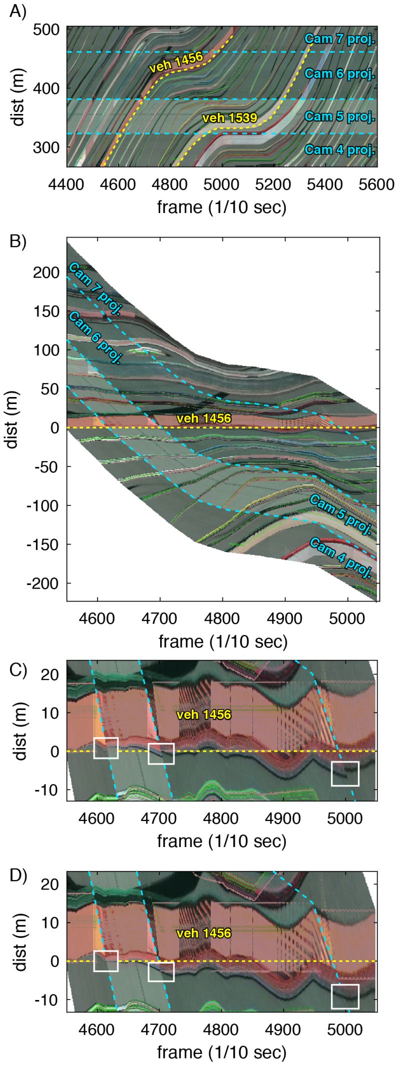

Fig. 3. Multi-camera STS (A) STS, (B) skewed STS, (C) detail of B showing trajectory performance and camera overlap, (D) after fixing overlap problem.

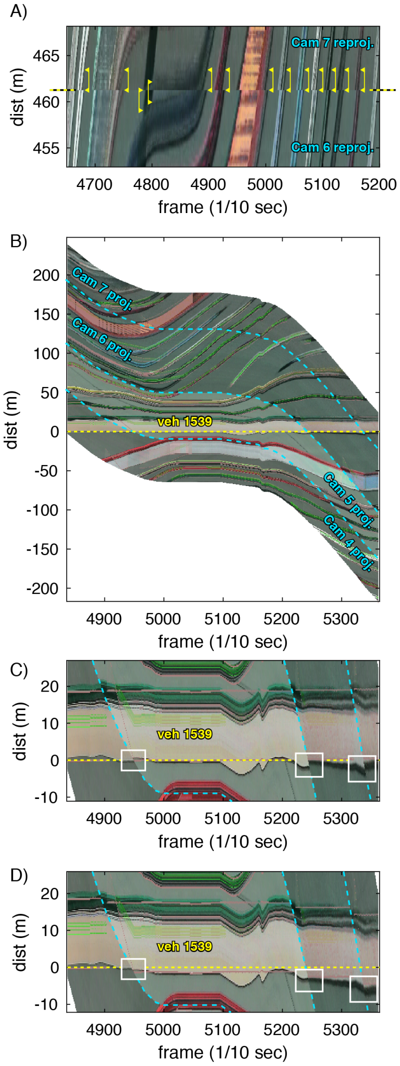

Fig. 4. Multi-camera STS. (A) Camera overlap in the STS, (B) skewed STS, (D) detail of B showing trajectory performance and camera overlap, (D) after fixing overlap problem.

the same short stretch of roadway twice because the camera views actually overlap. Indeed, zooming in to the combined STS from Fig. 3A, Fig. 4A shows that one can visually see that either cameras 6 and 7 overlap or that camera 7's clock is ahead of camera 6 by a few time steps.

Borrowing ideas from [21], [22], we extracted the time series of headways at the very end of one camera and compared these sequences against each of the first 20 pixels in the next camera to find the best match by shifting in time and space. It turns out that the cameras are synchronized at the 10 Hz resolution (i.e., no time offset) and the left-pointing yellow bars along the camera boundary show the resulting match point between the end of camera 6 and the location within camera 7. In this case the two cameras overlap by 14 pixels, corresponding to 2.1 m. The two right pointing bars show features seen in both cameras that are 2.1 m apart. These features were not used for the alignment, and thus, provide verification for the analysis. Note that this simple longitudinal offset arises because all of the cameras are projected into the same ground-plane. Our ongoing research is developing a formal process to automatically identify the amount of overlap between successive cameras.

Since the NGSIM tracking assumed perfect alignment between cameras (no overlap and no gap), the reported trajectories should appear to travel twice as fast across the overlap region than the vehicles actually traveled. For this paper, upon diagnosing the overlap problem, we deleted the overlapping rows from the upstream camera STS and regenerated the combined STS for the four cameras. We deliberately chose to remove the rows from the upstream camera because the distance per pixel in the raw video increases as you move downstream in a given camera (e.g., as evident in the left-most image of Fig. 1), thus, the raw video at the end of one camera should be lower resolution than at the start of the next camera (e.g., you can see these impacts in the camera 6 projected frame in Fig. 1, the vehicles at the start of the frame show far more detail than those at the end of the frame).

After removing the duplicate rows from the STS we once more skew the combined STS using the reported trajectory, resulting in Fig. 3D. First notice that now the flattened track is continuous at the camera boundaries, as highlighted by the three boxes. Although the track is continuous, we should still expect a vertical shift as the vehicle crosses the camera boundary since the trajectory still includes the overlap even though the STS does not. But the shift should be gradual since it represents the speed being off by a factor of 2 across the retained portion of the overlap region rather than an abrupt jump in space. Away from the camera boundaries the overlap should represent a constant vertical offset. In other words, if the only problem was the camera overlap, away from the camera boundaries the track should be perfectly flat and at the camera boundaries it should show a brief transition upstream to a different vertical offset from the flattened trajectory.

Fig. 4B-D repeat the analysis from Fig. 3 for vehicle 1539 that was shown in Fig. 3A. The results are similar-the flattened track undulates in Fig. 4C-D, indicating that the reported trajectory exhibits a continually varying positioning error. Meanwhile, using the exact same combined STS with the duplicate rows removed yields a continuous track across the camera boundaries in Fig. 4D.

This section has shown that by flattening the trajectory and skewing the STS it is possible to quickly see sub-meter errors in the reported trajectory even though the vehicle traveled 230 m. Since the vertical range in the skewed STS excludes the actual travel distance by the vehicle there is no limit to the length of roadway that can be validated this way.

B. Cleaning Trajectories or Tracking Vehicles Outright

Up to this point the paper has only demonstrated the skewed STS method to evaluate reported trajectories generated through other image processing techniques. This section demonstrates how the skewed STS can be used to clean those reported trajectories or generate new trajectories outright. In either case the process is the same, start with an incorrectly skewed STS (including no skew) and shift the columns of pixels to flatten the target vehicle track in the STS. For this example we will generate the trajectories from the original STS since the flattening only requires shifting all columns in a single direction (whereas cleaning a reported trajectory will likely require shifting some columns up and others down). The process can be done by hand, as per our work, or automated using some form of machine vision or edge detection, e.g., one way to automate this process would be to use a Hough transform to find the optimal alignment in each successive column.

Recall that the projected video used up to this point has pink boxes superimposed on the video. These boxes move roughly with the vehicles and thus, interfere with tracking the actual vehicles. So the raw video for each camera (e.g., the left frame in Fig. 1) is re-projected to the ground-plane without the pink boxes. For cameras 4-6 the rows of pixels that overlap the subsequent camera are removed (as per the discussion of Fig. 3C), next the STS is generated for each of the four of the cameras, and then stacked to form the combined STS, e.g., Fig. 5A (compare to the original combined STS in Fig. 3A).

For this work we manually flatten the track for each vehicle as follows. First, the STS is cropped to only span the duration that the given vehicle is within the STS. Then starting with the first column of pixels, all subsequent columns are shifted downward to bring the bottom of the track in one column in line with that of the previous columns. This process is repeated until the track in all columns has been brought in line with the first column. Fig. 5B shows a detail of this process for vehicle 1456, with all frames up to 4700 flattened while those after 4700 are still being brought in line.

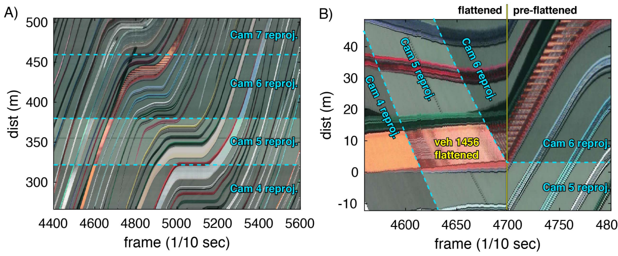

Fig. 5. (A) Combined re-projected STS, (B) flattening in progress.

This method emerged from a need to extract precise localization with sufficient fidelity to measure instantaneous speed and acceleration for the study of traffic dynamics and driver behavior. In this context, if the trajectories were extracted using automated techniques it would still require manual validation anyway. Since the human would be in the loop regardless, we chose to do the alignment manually from the start. The process was aided by a graphical user interface (GUI) to let the human operator quickly process the trajectories. The GUI actually semi-automates the process, employing a simple edge detector in each column to find the vehicle-road transition point that is subsequently shifted to align with the previous columns at 0 distance (e.g., frame 4,700 in Fig. 5B). The greatest time demands arise during stop waves where the gaps between tracks shrink or sometimes disappear altogether in the most distant camera, when a vehicle passes into a shadow (most commonly due to a tall truck in the adjacent lane) or is otherwise partially occluded. When these events occur, the operator would shift to flattening a different feature on the vehicle's track until the rear was once more visible.

Fig. 6A shows the final skewed STS for vehicle 1456. As before, all the critical information for the flattened track falls in a small vertical range, as shown in Fig. 6B. For reference the three camera boundaries are shown with dashed lines in Fig. 6B Note how the track elongates as the vehicle travels downstream due to the increasing projection error on the far end. Fig. 6C shows vehicle 1456 in one frame from each camera, while the lines connected to Fig. 6B show where each frame falls in the skewed STS. The progression in Fig. 6C shows how the multi-unit truck goes from a nearly top-down view in camera 4 to a rear view where the cab is completely occluded by the trailer in camera 7, reaffirming the importance of following the closest point on the track to minimize the projection errors.

While the track in Fig. 6B is truly flattened and compares favorably to the evaluation of the reported trajectory in Fig. 3D, a flat track is not itself of much value. What we really want is the trajectory. The trajectory itself is recorded in the translations made to the combined STS in order to flatten the track, and it is in fact hiding in plain sight in Fig. 6A as the upper or lower boundary of the skewed STS. This skewed boundary is repeated in Fig. 6D to show the trajectory from the skewed STS in isolation, albeit with a negative magnitude due to the flattening process. After correcting for the sign, Fig. 6E shows a portion the new trajectory superimposed on the STS

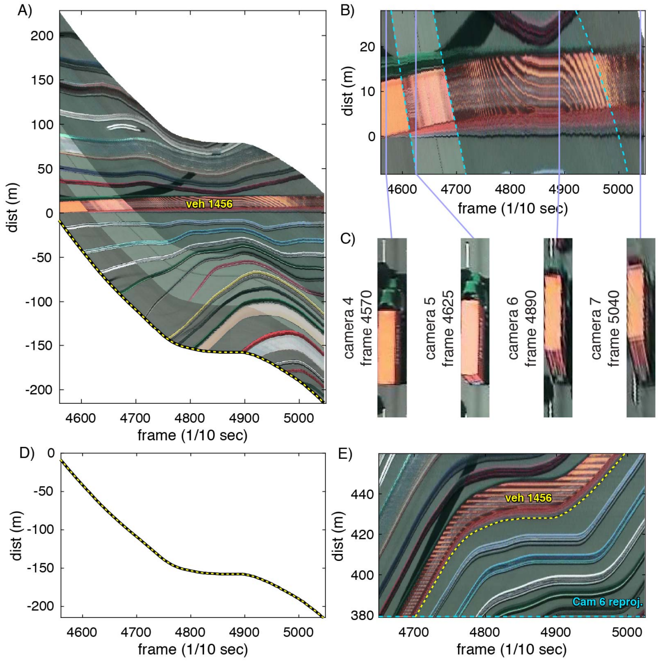

Fig. 6. Generating trajectories from the STS. (A) Skewed STS, (B) detail of A showing flattened track, (C) sample frames, (D) resulting trajectory, (E) a portion of the trajectory from E superimposed on the original STS.

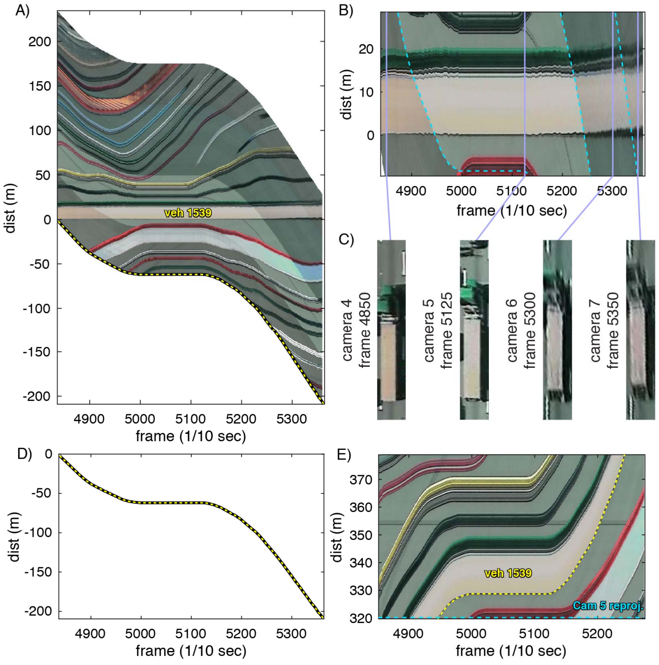

Fig. 7. Generating trajectories from the STS. (A) Skewed STS, (B) detail of A showing flattened track, (C) sample frames, (D) resulting trajectory, (E) a portion of the trajectory from E superimposed on the original STS.

from just one of the cameras to see the details (compare to the reported trajectory in Fig. 2B). The full trajectory spans all four cameras, i.e., the spatial range of Fig. 5A. Fig. 7 repeats the analysis from Fig. 6 for vehicle 1539 shown previously in Fig. 4. This process was used to regenerate all of the longitudinal trajectories in camera 6 in the first period of the NGSIM I-80 data set [5].

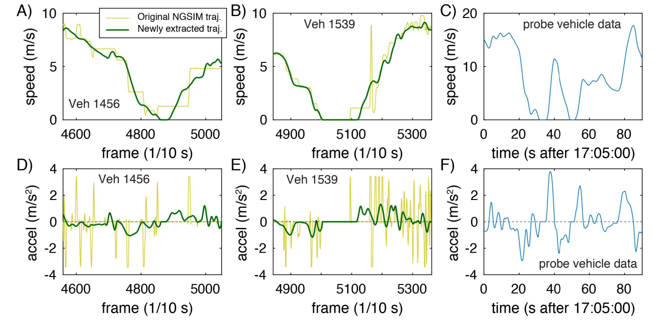

Fig. 8. Original raw and newly extracted v(t) and a(t) for: Veh. 1456, Veh. 1539, and for reference a probe veh.

To illustrate the benefits of this cleaning process, Fig. 8 compares the original NGSIM speed and acceleration against the newly extracted trajectories from Fig. 6 and 7. Fig. 8A-B show the time series speed for vehicles 1456 and 1539, respectively. Note how the original reported trajectories (yellow curves) exhibit piecewise constant speed. While this feature may seem innocuous, from our work with probe vehicles we know that very few drivers ever maintain a constant speed below 10 m/s except when stopped. For example, Fig. 8C shows typical time series speed from a probe vehicle in similar traffic conditions [23] the smoothly varying curves from the probe vehicle show little resemblance to the stair-stepped curves from the original NGSIM trajectories. Meanwhile after smoothing the sub-pixel noise [5] the newly extracted trajectories (green curves) in Fig. 8A-B exhibit smoothly varying speeds consistent with the probe vehicle data in Fig. 8C. Fig. 8D-E show the original NGSIM data have time series acceleration that is zero most of the time, punctuated by large peaks when the speed transitions between the discrete steps. This behavior is unrealistic, the newly extracted trajectories rarely have zero acceleration except when stopped and exhibit a slowly varying behavior that is consistent with the probe vehicle data shown for reference in Fig. 8F. The greatly improved time series acceleration is particularly important for the study of driver behavior, where car following models typically seek to model the driver's acceleration.

Repeating this analysis on benchmark vehicle tracking data sets, Fig. 9A shows a frame from a Cityflow sequence [24]. We do not have the homography to project into the ground-plane so using the unmodified video from the data set this example tracks the vehicles in the original image plane, along the nonlinear path of pixels shown in the frame. Comparing the geometry to Google Maps the total visible distance spans more than 650 m along the road. The resulting STS for the entire sequence is shown in Fig. 9B with a vertical line highlighting the frame used in Fig. 9A. Fig. 9C shows five of the Cityflow raw trajectories superimposed on a detail of the STS along the scan line in Fig. 9A. Vehicles 7 & 8 come to a stop with blue veh 7 occluding white veh 8 behind. At this spatial resolution the raw trajectories appear to do a good job following the tracks. After skewing the STS to flatten veh 7's raw trajectory in Fig. 9D one can see the noise manifest as ripples in the flattened track before and after the stop period, and smaller ripples during the stop period (frames 630-930). In this case the noise is also evident in the raw trajectory's speed, Fig. 9F shows v(t) in pixels/frame from the raw trajectory. Flattening the track rather than the trajectory can greatly reduce the noise, Fig. 9E shows the results after skewing the STS to flatten the near edge of the track. The flattened track is a lot smoother than the raw trajectory and

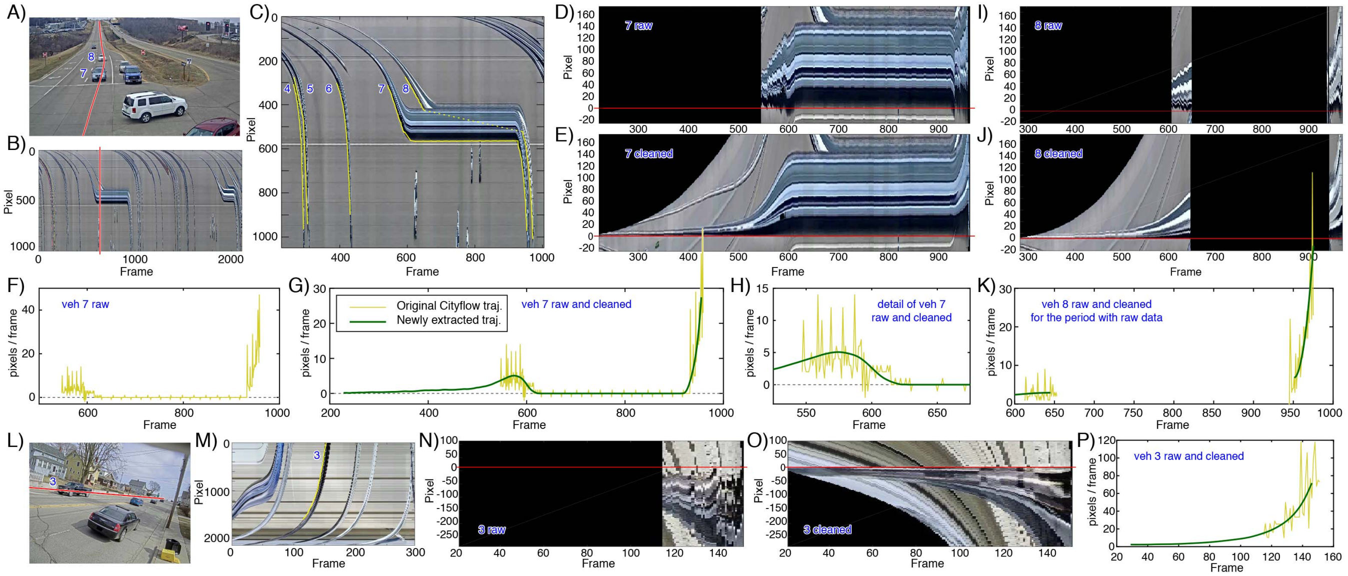

Fig. 9. Cityflow example. (A) Sample frame and path, (B) STS, (C) detail of A also showing original raw trajectories, (D) skewed STS from raw veh 7 traj, (E) D after cleaning the track, (F) v(t) from raw veh 7 traj., (G) adding newly extracted traj., (H) detail of G, (I)–(K) skewed STS for veh 8 raw and cleaned, (L)–(P) repeating the analysis on a Cityflow example from a side view captured close to the ground.

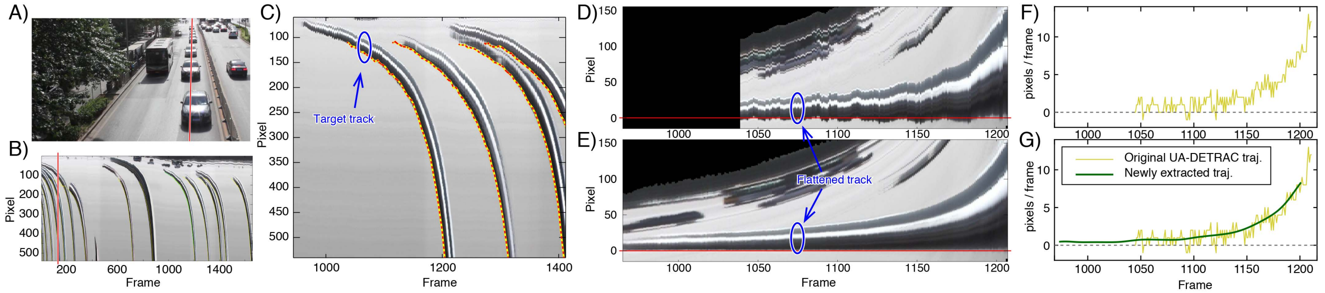

Fig. 10. UA-DETRAC example (A) sample frame and path, (B) STS, (C) detail of A also showing original raw trajectories, (D) skewed STS from raw traj, (E) D after cleaning the track, (F) v(t) from raw traj., (G) adding newly extracted traj.

extends much further upstream, persisting 77% further than the reported trajectory and reaching the full 650 m visible in the frame. Fig. 9G expands the time range and decreases the vertical range from Fig. 9F to show the corresponding v(t) from the newly extracted trajectory. The original v(t) peaks at 47 pixels/frame while the newly extracted v(t) peaks at only 27.5 pixels/frame. Fig. 9H zooms in further to show the noise in the original data, including several samples of with $v(t) < 0$. The strictly non-negative, smooth v(t) from the newly extracted trajectory provides greater fidelity from the raw trajectory in Fig. 9E. Fig. 9I flattens the raw trajectory for veh 8. The near end of veh. 8 was occluded in frames 652-948 and no position is reported in the raw data for this vehicle. Fig. 9J shows the results after skewing the STS to flatten the near edge of the track and Fig. 9K shows the resulting speeds. Fig. 9L-P repeats this exercise on another Cityflow sequence for the black pickup truck in the far lane (veh. 3) and scanline in Fig. 9L. This time from a camera with a side view captured close to the ground. In this case Google Maps shows that the raw and cleaned trajectory travel roughly 50 m and 180 m, respectively.

Fig. 10 repeats the evaluation on one of the UA-DETRAC sequences [25] with similar results from the trajectories in the image plane. As illustrated in Fig. 9 and Fig. 10, the skewed STS could be used to extend the range and improve the fidelity of the ground truth data in benchmark data sets like Cityflow and UA-DETRAC.

C. Lateral Tracking

The STS-based trajectory validation and generation methodology is not limited to longitudinal trajectories. To extend to lateral tracking the method is modified to build a lateral STS that moves longitudinally with a point on the vehicle, e.g., the midpoint. Fig. 11A shows a single frame from the camera 6 projected video. Vehicle 685 is explicitly labeled and a vertical line is drawn across the frame at the midpoint of this vehicle. This set of pixels is captured and is used as one slice in the lateral STS for this vehicle. Following the longitudinal trajectory for vehicle 685 generated using the techniques in Section III.B, this midpoint sampling is repeated in every frame associated with the longitudinal trajectory. Fig. 11B shows the resulting lateral STS for vehicle 685 in camera 6. The frame from Fig. 11A is highlighted with a vertical line. Notice how vehicle 685 has become an extended track that corresponds to the movement of its midpoint across the camera's field of view while the other vehicles (and pink boxes) have become distorted based on their relative speed to the subject vehicle when they were parallel with its midpoint. The reported lateral trajectory for this vehicle is superimposed on top of the lateral STS. Both the trajectory and the STS show

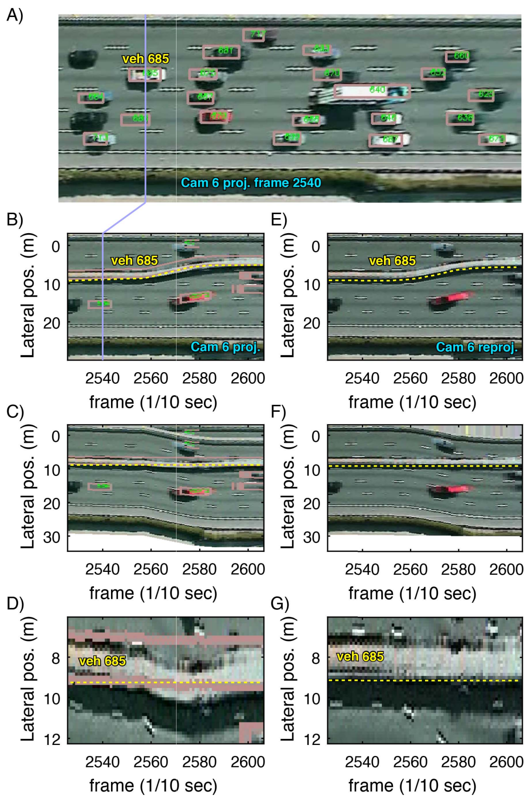

Fig. 11. Sample frames of the original video.

that vehicle 685 undertook a lane change maneuver to the left while traversing camera 6.

Repeating the process from Section III.A, in Fig. 11C the lateral STS is skewed in such a manner to flatten the reported trajectory. Fig. 11D zooms in to see the details of the skewed STS. In this case the pink box from the projected camera 6 video is evident throughout, giving rise to the two horizontal lines in the figure, with the bottom of these lines corresponding to the near side of the vehicle. As with the longitudinal trajectories, Fig. 11D shows that the STS undulates, with the flattened track leaving the reported trajectory by as much as 1 m near frame 2580. Repeating the process from Section III.B, Fig. 11E shows the resulting lateral STS from the re-projected camera 6 video. The lateral STS is skewed in the process of flattening the track in Fig. 11F, as shown in detail in Fig. 11G. The large deviation of Fig. 11D has been eliminated. The resulting lateral trajectory is then superimposed on the STS in Fig. 11E. Like the longitudinal tracking, the lateral tracking should use the near edge (or possibly shadows) to minimize the impact of lateral projection errors. Also like the longitudinal tracking, in the event of a partial occlusion of the primary feature of interest, use secondary features to bridge the gap.

IV. Discussion and Conclusion

Image processing is an important tool for tracking roadway vehicles over long distances, but it is imperfect, suffering from various errors. These errors can be hard to detect since the large distance traveled in the reported trajectories only serves to hide many tracking errors, e.g., a positioning error of 1 m could easily become imperceptible at the scale of a 100 m long trajectory. So this paper develops a simple and efficient approach for evaluating image processing based vehicle trajectories that separates the spatial positioning errors from the actual travel of a given vehicle.

The approach starts by constructing a spatiotemporal slice, STS, from the same video that was also used to generate the reported trajectory. Since the STS is essentially a visual time-space diagram composed of vehicle tracks it is used to evaluate the reported trajectory. This research recognized that the STS can be transformed to eliminate the actual travel of the vehicle recorded in the trajectory. Specifically, the STS is skewed in such a way that the reported trajectory is flattened. By projecting the STS to the perspective of a moving observer that travels with the reported trajectory in the skewed STS we have effectively removed the actual travel of the vehicle recorded in the trajectory, and only retain the positioning errors. At which point sub-meter positioning errors that were imperceptible over the extent of the actual travel become readily apparent in the form of small wobbles in the associated flattened track. Thereby providing a means to quickly assess and validate the trajectories from almost any image processing system against the true vehicle positions seen in the original video data. Since the vertical range in the skewed STS excludes the actual travel distance by the vehicle there is no limit to the length of roadway that can be validated. This paper showed examples of up to 650 m of travel. On the other hand, many applications do not require the fidelity needed to measure acceleration, and validation might only need to go as far as the STS without skewing, e.g., Fig. 2B, 3A, 9C and 10C all allow for quickly verifying that all of the trajectories roughly follow their respective tracks.

Recognizing that the flattening process works both ways, if any errors are evident in a given trajectory, the STS method can also be used to quickly fix them. Alternatively, one can use this process to generate the vehicle trajectories directly from the STS in the first place by shifting the columns of pixels to flatten the target vehicle track in the STS. The process can also be used to assess (or extract) the lateral position of a given vehicle, e.g., Fig. 11.

In Section III.B this work uses manual flattening to generate vehicle trajectories in a semi-automated process, whereby an edge detector is used in each frame to find the location of the vehicle-road transition point that is then shifted to align with the previous columns at 0 distance. It should be clear that it would be a simple extension to develop a more robust automated process to generate the trajectories in this fashion, e.g., automatically tracking multiple features on a track and devise a metric for optimal “flatness” across these features. Ultimately the present work is a proof of concept. The exact nature of automating this process depends on the needs of the application and the quality of the original video, e.g., resolution, view angle, light vs. dark pavement, different approaches for handling occlusions, illumination, etc.. Alternatively, one could start with the output of a conventional tracker, e.g., Fig. 3C, and use a road detector to simply shift the columns to align the vehicle-road boundary into a single row of the skewed STS. Since the road does not rapidly change appearance, this approach could use conventional background subtraction and shadow handling algorithms to identify when a given pixel corresponds to the road.

There are a few key points in the method to keep in mind, these include taking care to track the lowest point on the near side of the target vehicle to minimize the impacts of projection errors. If there are partial occlusions one can use other features in the track to bridge the periods where the preferred feature is unobservable. The features on a given target may change, evolve, appear, or disappear over the span of the reported trajectory, but the methodology is robust to the evolving appearance of the target vehicle (e.g., see Fig. 6-7). Most of this paper considered the case when the imagery is projected into the ground-plane, but the work can easily be extended to any other viewing plane (e.g., see Fig. 9-10). The method presented herein was limited to a single lane, for lane change maneuvers one could use a weighted average of the STS in adjacent lanes to follow the target vehicle as it maneuvers across lanes.

This paper illustrated the methodology on a portion of the NGSIM data set. Although the NGSIM data have been exhaustively studied over the past 15 years, this paper identified a newfound problem in the NGSIM data. While the NGSIM researchers believed that they cropped the camera views precisely such that there was no gap and no overlap between successive projected camera views, it turns out that the successive cameras overlapped by up to 2 m. This process was also used to regenerate all of the longitudinal trajectories in camera 6 in the first period of the NGSIM I-80 data set [5], in the process the methodology turned up several more previously unknown problems in the NGSIM data. As such, this work has demonstrated the value of using the skewed STS to evaluate reported trajectories and then using the methodology to clean the errors. Our ongoing research is applying this process to all of the vehicles across all seven of the cameras in the first period of the NGSIM I-80 data set.

While much of the paper used the NGSIM video, the tools transcend the NGSIM study. Accurate speed and acceleration measurement is important for many traffic applications, no matter what the tracking system might be. For example, Fig. 9 and Fig. 10 show that the skewed STS could be used to extend the range and improve the fidelity of benchmark data sets like Cityflow and UA-DETRAC.

References

[1] T. Christian, M. Treiber, and A. Kesting, "Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2088, pp. 90–101, Dec. 2008.

[2] A. Duret, C. Buisson, and N. Chiabaut, "Estimating individual speed-spacing relationship and assessing ability of Newell's car-following model to reproduce trajectories," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2088, no. 1, pp. 188–197, Jan. 2008.

[3] V. Punzo, M. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1243–1262, 2011.

[4] M. Montanino and V. Punzo, "Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns," *Transp. Res. B, Methodol.*, vol. 80, pp. 82–106, Oct. 2015.

[5] B. Coifman, L. Li, and W. Xiao, "Resurrecting the lost vehicle trajectories of Treiterer and Myers with new insights into a controversial hysteresis," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 20, pp. 25–38, Dec. 2018.

[6] V. Kovvali, V. Alexiadis, and L. Zhang, "Video-based vehicle trajectory data collection," in *Proc. 86th Annu. TRB Meeting*, 2007, pp. 1–4.

[7] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, no. 2, p. 284, Feb. 1985.

[8] Y. Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by exploiting spatio-temporal image slices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 797–808, Aug. 2000.

[9] T. Aagaard and J. Holm, "Digitization of wave run-up using video records," *J. Coastal Res.*, vol. 5, no. 3, pp. 547–551, 1989.

[10] R. T. Holland and R. A. Holman, "The statistical distribution of swash maxima on natural beaches," *J. Geophys. Res.*, vol. 98, no. 6, pp. 10271–10278, Jun. 1993.

[11] Z. Zhu, B. Yang, G. Xu, and D. Shi, "A real-time vision system for automatic traffic monitoring based on 2D spatio-temporal images," in *Proc. 3rd IEEE Workshop Appl. Comput. Vis.*, Sarasota, FL, USA, Oct. 1996, pp. 162–167.

[12] L. Anan and Y. Zhaoxuan, "Video vehicle detection algorithm through spatio-temporal slices processing," in *Proc. 2nd IEEE/ASME Int. Conf. Mechatronics Embedded Syst. Appl.*, Aug. 2006, pp. 1–5.

[13] R. Cho, "Estimating velocity fields on a freeway from low-resolution videos," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 463–469, Dec. 2006.

[14] Y. Malinovskiy, Y. Wu, and Y. Wang, "Video-based vehicle detection and tracking using spatiotemporal maps," *Transp. Res. Rec.*, vol. 2021, pp. 81–89, Oct. 2009.

[15] T. Zhang, "A longitudinal scanline based vehicle trajectory reconstruction method for high-angle traffic video," *Transp. Res. C, Emerg. Technol.*, vol. 103, pp. 104–128, Jun. 2019.

[16] V. Knoop, S. Hoogendorn, and H. Van Zuylen, "Processing traffic data collected by remote sensing," *Transp. Res. Rec.*, vol. 2129, pp. 55–61, Oct. 2009.

[17] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, "Motion analysis and segmentation through spatio-temporal slices processing," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 341–355, Mar. 2003.

[18] M. Kilicarslan and J. Y. Zheng, "Detecting walking pedestrians from leg motion in driving video," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 2924–2929.

[19] B. Coifman, D. Lyddy, and A. Skabardonis, "The Berkeley highway laboratory-building on the I-880 field experiment," in *Proc. IEEE Intell. Transp. Syst. Process.*, Oct. 2000, pp. 5–10.

[20] B. Coifman and M. Cassidy, "Vehicle reidentification and travel time measurement on congested freeways," *Transp. Res., A*, vol. 36, no. 10, pp. 899–917, 2002.

[21] H. Lee and B. Coifman, "Using LIDAR to validate the performance of vehicle classification stations," *J. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 355–369, Oct. 2015.

[22] B. Coifman, M. Wu, K. Redmill, and D. Thornton, "Collecting ambient vehicle trajectories from an instrumented probe vehicle: High quality data for microscopic traffic flow studies," *Transp. Res. C Emerg. Technol.*, vol. 72, pp. 254–271, Apr. 2016.

[23] Z. Tang *et al.*, "CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8797–8806.

[24] L. Wen *et al.*, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Understand.*, vol. 193, Apr. 2020, Art. no. 102907.

Benjamin Coifman holds a joint appointment in civil engineering and electrical engineering at The Ohio State University. He has been working to address deficiencies in traffic surveillance, traffic control, and traffic flow theory since 1995. He has been recognized by several awards, including the ITS America Award for The Best ITS Research, an NSF CAREER Award, and the TRB Greenshields Award.

Lizhe Li was born in Xi'an, Shaanxi, China, in 1989. He received the B.S. and M.S. degrees from the Department of Automation, Northwestern Polytechnical University, China, in 2013. He is currently pursuing the Ph.D. degree in electrical engineering with The Ohio State University. His research interests include image processing, vehicle tracking, and traffic dynamics.