Abstract—This paper proposed the “Segmentation is traCK-ing” (SiCK) solution that extract vehicle movements with semantic segmentation for high-resolution trajectory reconstruction and validation. The dynamic mode decomposition (DMD) method extracts vehicle strands by decomposing the spatial-temporal map (STMap) into the sparse foreground and low-rank background. The Res-UNet+ neural networks was designed by adapting two prevalent deep learning architectures, which significantly improved the performance of the STMap-based vehicle detection and tracking. The DMD model provides many interesting insights for understanding the evolution of underlying spatial-temporal structures preserved by the STMap. The model outputs were compared with the previous image processing model and baseline semantic segmentation neural networks. After thorough evaluations, this solution is accurate and robust against many challenging factors. Finally, this work addresses data quality issues by correcting erroneous trajectories obtained from computer vision tools. Extracting high-fidelity vehicle trajectories for transportation scientific research is a systematic process. This new framework can significantly improve the accuracy and reliability of video-based trajectory data acquisition.

Index Terms—DMD structure, Res-UNet+ model, spatial-temporal map, quality assurance.

I. INTRODUCTION

VIDEO sensors have been widely used for collecting vehicle trajectory data to support academic research, traffic operations, management, and design. One of the most influential video-based trajectory datasets is the Next Generation SIMulation (NGSIM) trajectory dataset [1], which has significantly boosted traffic flow and modeling research by revealing microscopic traffic characteristics. As highlighted by a survey paper [2], although video-based trajectory datasets have greatly improved the simulations of driver behavior models and their calibration/training, there is still a substantial demand for high-quality, high-resolution trajectory data. Collecting useful trajectory data from traffic cameras with satisfactory accuracy is challenging. The traditional trajectory extraction approach, which contains multi-stage procedures, is error-prone. Previous practice for data quality control is very inefficient, involving manual processes to modify, add or delete tracked objects by hand. The signal-processing-based noise removal methods are unsatisfactory because they can only detect some incongruity points that deviate from the average driver’s behavior. However, they cannot address the root cause of detection and tracking failures. The scanline method efficiently validates extracted vehicle trajectories by showing vehicle movements on the static Spatial-temporal Map (STMap), allowing directly identifying the error for each vehicle.

Using high-angle traffic video for vehicle trajectory extraction is a topic of growing importance, given the continued accessibility of overhead cameras (e.g., drone-based or infrastructure-based) in the transportation area [3], [4], [5]. The NGSIM dataset has been used as the ground-truth data to calibrate and develop traffic flow models, demonstrate driving behavior or traffic phenomena, and conduct traffic-state estimation and prediction [6], [7]; however, many publications have uncovered underlying systematic errors in the NGSIM dataset. Several studies [8], [9], [10], [11], [12], [13] examined the performance issue and proposed denoising methods based on statistic distributions, filtering and smoothing, traffic-informed constraints, and information theory. NGSIM trajectory datasets were generated with traffic videos taken from high-rise buildings, which applied an appearance-based vehicle detection algorithm to extract line segments from images and match them to 3D vehicle models. The detected vehicles were tracked according to their appearance in the camera image. Then a laborious manual validation process is employed to modify the tracklets frame by frame.

The impact of NGSIM has been broad and far-reaching, making it the most used dataset in transportation research. As of July 2023, NGSIM data has been utilized in over 5300 publications by Google Scholar, spanning almost all premier venues in the traffic flow, ITS, and safety research. To solve the data quality issue in the NGSIM-like datasets and meet the needs of transportation research, this paper presents a quality control strategy, named as “Segmentation-is-traCK-ing” (SiCK), to reconstruct trajectories from STMaps. Dynamic mode decomposition (DMD) analyzes the STMap by decomposing it with different underlying structures. The DMD background subtraction results were used to prepare training data for a new deep neural network. The Res-UNet+ model is built on two convolutional neural networks with enhanced feature fusion. This method has substantially improved high-fidelity trajectory data acquisition and simplified quality assurance with an efficient tool.
II. RELATED WORK

A. Computer Vision Algorithms for Traffic Video Analysis

Traffic detection is part of the object detection problem in computer vision, and significant progress has been made in recent years with the rise of deep learning. Object detection involves recognizing objects in targeted classes and precisely localizing each object. Table 1 shows a summary of computer vision techniques related to traffic detection, which can be classified into five major categories, including shape-based methods [14], [15], [16], background/foreground modeling [17], [18], [19], deep learning models [20], [21], [22], [23], feature-based models [24], [25], and scanline methods [26], [27].

Semantic segmentation is another computer vision tool for traffic video analytics, which predicts class labels at the pixel level for each image. This task's challenges concern the requirements of pixel-level accuracy and multi-scale contextual information of the class labels [28]. Semantic segmentation has been used in many applications: self-driving cars, virtual and augmented reality, biomedical image segmentation, etc. Many segmentation models are built upon prevailing neural networks, such as AlexNet [29], VGG-16 [30], GoogLeNet [31], and ResNet [32]. U-Net was first proposed as a semantic segmentation method to process biomedical images [33]. Given the many similarities shared between STMap trajectory and the medical image segmentation task, the UNet framework is selected as our model architecture [34].

Multi-object tracking (MOT) is vital for many applications and has been extensively investigated. The object-tracking methods can be classified into online and offline tracking. Online tracking uses only current and previous frames, where long-term movements are embedded into a state space for memorization [35], [36], [37]. Offline tracking is based on global optimization algorithms using time-series information regarding desired objects [38], [39], [40]. Recent deep learning approaches have gained tremendous momentum and successfully enhanced the performance of MOT, including siamese networks, attention and transformer, and recurrent neural networks [41], [42], [43], [44]. Other studies tackled video understanding issues for event-based trajectory learning and classification through data association or clustering [45], [46], [47].

B. Scanline Method

The scanline method stems from the spatial-temporal slice (STS) structure used in computer vision literature. Early STS methods were introduced to solve the structure from motion (SFM) problem for cameras on moving platforms [48], [49]. Later, the STS method was also used for object and pedestrian detection [50], [51]. In transportation research, the scanline method is a set of pixels that can capture object movements on the user-selected roadway from the video image. After stacking scanline pixels together over continuous frames, a STMap is obtained. The horizontal axis of the STMAP shows the time progression, and the vertical axis contains distance information.

Two types of scanlines are used in traffic detection: lateral and longitudinal. The lateral scanline is a cross-section scanline across a lane, whereas the longitudinal scanline is drawn along the traffic direction. The lateral scanline method was primarily intended for traffic counting [52] and speed measurement [53], while the longitudinal scanline method was used for vehicle tracking [54], [55] and detection [56]; however, most previous scanline methods were only used to estimate macroscopic parameters such as traffic count, headway, and spot speed.

A recent study [57] developed a high-angle spatial-temporal diagram analysis (HASDA) model to generate high-resolution (0.1s) vehicle trajectories using the longitudinal scanline. The HASDA model includes three major steps: the generation of a STMap, the extraction of the pixel trajectories from the STMap, and the coordinate transformation of the pixel to physical distance. The HASDA model mainly relies on image processing techniques such as background subtraction, shadow removal, edge detection, and morphological operations. Those image processing techniques, such as edge detection, are vulnerable to noise. LiDAR data were combined with the scanline method to facilitate the calibration process in a later paper [58]. Another research applied scanline for arterial signal performance measurements [59].

C. Dynamic Mode Decomposition (DMD)

Dynamic Mode Decomposition is a data-driven analytic method that integrates Fourier transforms and singular value decomposition (SVD). The DMD method was first introduced to extract meaningful spatial-temporal coherent structures that dominate dynamic activities in fluid mechanics. DMD method conducts the eigen-decomposition of spatial-temporal coherent structures [60], therefore reducing the dimensions of complex systems efficiently without losing accuracy [61]. DMD methods have gained traction in many application areas, such as fluid dynamics, video processing, control, epidemiology, and financial models. The DMD algorithm seeks to find the best fit between the following two matrices.

\[
X = \begin{bmatrix}
| & \cdots & | \\
| x_1 & \cdots & x_{m-1} | \\
| & \cdots & |
\end{bmatrix}, \quad X' = \begin{bmatrix}
| & \cdots & | \\
| x_2 & \cdots & x_m | \\
| & \cdots & |
\end{bmatrix}
\]

(1)

where \(x_{k=1,...,m}\) is a vector that represents the dynamic system state at time interval \(k\), \(X\) matrix represents the prior states from intervals 1 to \((m - 1)\), and \(X'\) matrix represents the posterior states from intervals 2 to \(m\). Matrix \(X\) and \(X'\) are linked by a linear operator \(A\):

\[
X' = AX
\]

(2)

Our goal is to find the matrix \(A\) that represents the evolution of system states. A well-studied least square estimation problem is formulated.

\[
\hat{A} = \arg\min_A \|X' - AX\|_F^2
\]

(3)

where \(\hat{A}\) is an estimator of matrix operator \(A\), which is computed by minimizing the Fresenius norm \(\|X' - AX\|_F^2\).
Fig. 1. STMap generation after stacking scanline vector.

\( \hat{A} \) is governed by the least-square optimization

\[
\hat{A} \approx X'X^{-1}
\]

where \( X' \) is obtained by using the Moore-Penrose pseudoinverse.

Instead of solving the matrix operator \( A \) directly for the DMD algorithm, \( A \) is often solved by the eigendecomposition of \( A \) after the proper orthogonal decomposition (POD). The DMD method can be considered a robust principal component analysis (PCA) with high computational efficiency. The eigenvalues of matrix \( A \) can indicate the time evolution of dominant modes [62], [63].

III. METHODOLOGY

A. STMap Generation

As illustrated in Figure 1, the STMap is generated by stacking the longitudinal scanline \((l_1, l_2, l_3, \ldots, l_m)\) frame by frame to form a three-dimensional matrix \( S^{n \times m \times 3} \), where \( n \) denotes the number of pixels per scanline, \( m \) is the number of video frames, and 3 indicates the \( R - G - B \) channels. The color pixels moving simultaneously in the STMap represent a unique vehicle passing the scanline. We aim to segment each vehicle strand from the STMap to detect trajectories.

B. DMD for STMap Segmentation

In this section, we adopted the DMD method for STMap decomposition, allowing temporal information stored on static STMap to be processed with pattern recognition and segmentation. Previous DMD background subtraction converts the entire image into a state vector for background subtraction. While in this research, we first extract the vehicle movements with a spatial-temporal map and then apply mode decomposition to obtain vehicle trajectory. The scanline pixel values at each frame can be considered the state of traffic dynamics at a particular timestamp. Traffic state of \( I_{x+1} \) at the time \((x + 1)\) is assumed to relate to the previous traffic state of \( I_x \) by time-dependent linear operator \( A \), which reflects the time evolution of scanline pixels.

\[
I_{x+1} = A \ast I_x
\]

The STMap can be formulated as follows:

\[
S = \begin{bmatrix}
| & | & | \\
| l_1 & \ldots & l_{m-1} | \\
| \downarrow & \ldots & \downarrow |
\end{bmatrix}, \quad S' = \begin{bmatrix}
| & | & | \\
| l_2 & \ldots & l_m | \\
| \downarrow & \ldots & \downarrow |
\end{bmatrix}
\]

where \( S \) is the prior STMap, \( S' \) is the posterior STMap. \( S' \) has a one-frame shift from \( S \). The relationship between \( S \) and \( S' \) becomes

\[
S' = AS
\]

where the matrix \( A \) describes the time-differencing operation. The DMD mode that contains spatial information is an eigenvector of \( A \). And each DMD mode corresponds to an eigenvalue of \( A \). By finding the eigenvectors and eigenvalues of the matrix \( A \), we obtain the DMD mode \( \Phi \).

\[
A \Phi = \Phi \Lambda
\]

The columns of \( \Phi \) are eigenvectors comprising the dominant mode \( \phi_j \), and \( \Lambda \) is the diagonal matrix of eigenvalues \( \lambda_j \). The STMap can be reconstructed using first \( k \) modes, where \( k \leq \min(n, m) \).

\[
STMap \approx \Phi BV
\]

where \( \Phi \) contains the dominant modes from the STMap, and matrix \( B \) is the matrix of amplitudes. \( V \) is the Vandermonde matrix representing the time evolution of DMD modes. This function is illustrated in Figure 2.

A scanline vector \( l_t \) at frame \( t \in 1, \ldots, m \) can be estimated as follows:

\[
\tilde{l}_t = \sum_{j=1}^{k} b_j \phi_j \lambda_j^{t-1}
\]

where \( b_j \) is amplitude, \( \phi_j \) is each DMD mode, and \( \lambda_j^{t-1} \) is the time evolution of each mode.

Let \( t = 1 \), which represents the initial state of the scanline as follows.

\[
\tilde{l}_1 = \sum_{j=1}^{k} b_j \phi_j
\]

The matrix \( B \) can then be estimated as a least-square problem using the first scanline \( l_1 \) as an initial state.

\[
\tilde{B} = \arg \min_B \| l_1 - \Phi B \|
\]

Any DMD mode that does not change in time will have an eigenvalue \( \lambda_j = 1 \), which forms the background of the STMap.

In the STMap, the background pixels are highly correlated between neighboring columns, suggesting the low-rank structure within the STMap. Therefore, the DMD algorithm separates background and foreground by decomposing the STMap into low-rank (background) and sparse (foreground) components.

\[
S_{DMD} = \text{background} + \text{foreground}
\]

\[
= \sum_p b_p \phi_p \lambda_p^{t-1} + \sum_{j \neq p} b_j \phi_j \lambda_j^{t-1}
\]

where \( \lambda_p \) and \( \lambda_j \) are the eigenvalues corresponding to the background and foreground modes, respectively.
to higher-level decoders, we concatenate all the decoder layers. To enable lower-level decoder information to pass connections between the corresponding level of encoders and objects. In the vanilla UNet architecture, there are only inter-higher-level layers explore the localization of the targeted objects. Lower-level layers capture the boundary of objects, whereas learned from different scales often entail different information. Many segmentation studies [64], [65], [66] show that features in intra-connections among different levels of decoding stages. were designed to reduce the semantic gap. We added the interconnections between encoding and decoding layers blocks replace the original encoders in the UNet model. The Res-UNet model uses the ResNet block as the backbone, and further increases its performance by modifying the decoding layers. In the encoding process, the ResNet line models are considered as follows:

$$L^{i}_{de} = \begin{cases} A^{2} \left( \left[ L^{n}_{en} , A_{T} \left( L_{bottom}^{i} \right) \right] \right) , & i = n \\ A^{2} \left( \left[ A \left( L^{n}_{en} \right) , A_{T} \left( L^{i}_{de} \right) \right]_{j=i-1} \right) , & i < n \end{cases}$$

where $L^{de}_{i}$ is the $i^{th}$ decoder layer output, $L^{en}_{i}$ is the $i^{th}$ encoder layer output, $A_{T}(\cdot)$ represents the depth concatenation operation, $A(\cdot)$, $A_{T}(\cdot)$, and $A^{2}(\cdot)$ represents respectively convolution, transposed convolution, and twice operation of convolution, followed by the ReLU activation, $L_{bottom}$ is the bottom bridge layer output.

IV. MODEL IMPLEMENTATION

A. Baseline Models

Mainstream image semantic segmentation models as baseline models are considered as follows:

1) ResNet-18/ResNet-50: The building block for ResNet includes the main branch that contains convolution, batch normalization, and ReLU layer consecutively, as well as a residual connection that bypasses the main stem to allow the gradients to flow more easily. This paper tested the 18-layer...
and the 50-layer ResNet architecture as reference models. The left branch of Figure 4 illustrates a similar ResNet structure used in the proposed model.

2) U-Net: The vanilla U-Net model with encoding and decoding stages is also used as a reference model. The skip connection is comprised of two sets of convolutions and ReLU layers. The vanilla U-Net has a U-Shaped structure similar to Figure 4 but with a more straightforward left branch and without the intra-connected decoding layers defined in the proposed model.

3) Res-U-Net: For the Res-U-Net model, we did not add intra-connection to integrate the information from all encoder layers. Our Res-U-Net architecture reformed the U-Net model architecture by replacing the original encoder layer with a two-branch ResNet block.

4) Fully Convolutional Network (FCN): The FCN model is an end-to-end encoder-decoder semantic segmentation neural network. The encoder-decoder architecture was inherited by almost all subsequent segmentation models [67].

5) DeepLabv3+: DeepLab model was also built on the encoding-decoding framework, adopting the Xception model and the depthwise separable convolution achieve a faster and stronger encoder-decoder network [68].

6) SegNet: Another pixel-wise segmentation neural network is SegNet, which uses 13 convolutional layers topologically similar to VGG16 as the encoder. Their decoder layer uses pooling indices computed in the corresponding encoder layer to perform non-linear upsampling [69].

B. Dataset and Augmentation

Unlike other data labeling process, which is highly specialized and requires a lot of manual efforts, a single person will suffice to complete the task of labeling hundreds of STMap using the mentioned DMD methods. This is one of the advantages of using the STMap method, as there is no need to collect vehicle images from all possible scales, parts, angles, colors, or shapes.

The STMap training dataset was created using four 15-minute NGSIM I-80 videos in this study. First, we obtained 20 STMaps from 20 lanes, then cropped them into 1000 512x512 images. Finally, we augmented the 1000 images to get more datasets by rescaling, shearing, and translating the generated 512x512 images. Since the vehicle strands in the STMap extend from the top left to the bottom right, we do not need to use the rotation transformation in the data augmentation process.

C. Implementation Details

We programmed the proposed algorithms with image processing and Deep Learning Toolbox in MATLAB. The pre-trained model parameters were retained as the initialization values. The stochastic gradient descent with momentum is used to optimize the proposed network, and the batch size is set to 3 images each time due to the GPU memory constraints, and the learning rate is initialized as 0.05. All the models were trained on GeForce GTX 3070 (8 GB memory), which can be implemented with a laptop. The training time is around 5 hours for the model to converge. Once trained, the proposed model takes 2 minutes to process the STMap from a 15-minute video. The labeled datasets were shuffled and partitioned into 60% training, 20% testing, and 20% validation. Two classes are defined in the pixel classification layer, vehicle strands and background.

D. Trajectory Extraction

After the segmentation and obtaining of the binary masks for all vehicle strands from STMap, the following step extracts the vehicle pixel trajectory. The vehicle trajectory is acquired using the lower boundary of each vehicle strand. In MATLAB, it’s just a one-line function bwperim(), making the method super-efficient, shown in Figure 5.

The pixel trajectories from STMap can then be converted into pixel movements on the original video. Then the vehicle trajectory will be converted into NGSIM Local-y coordinates using the method in [57].

E. Performance Metrics

Three main performance metrics were used to quantitatively assess the segmentation model performance, accuracy, Jaccard coefficient, and BF Score (Boundary F1 Score). Accuracy (Acc) is calculated as Equation (18):

\[ Acc = \frac{TP}{TP + FN} \tag{18} \]

where \( TP \) is the count of true positives; \( FN \) is the count of false negatives. The Jaccard coefficient measures the similarity between sets A and B (intersection over union or IoU). The Jaccard coefficient can also be expressed in terms of TPs, FPs, and FNs as Equation (19):

\[ J(A, B) = \frac{TP}{TP + FP + FN} \tag{19} \]

The third performance metric is BF Score, which computes with the following Equation (20)

\[ BF = 2 \cdot \frac{precision \cdot recall}{recall + precision} \tag{20} \]

The error measure for the trajectory detection results is all trajectory points’ mean absolute error (MAE) defined in Equation (21).

\[ MAE_o = \frac{1}{N} \sum_{t=1}^{N} |\hat{y}_o(t) - y_o(t)| \tag{21} \]

where \( o \) is the trajectory index, \( y_o(t) \) and \( \hat{y}_o(t) \) are the actual and model-estimated local-Y location at time \( t \), respectively, \( MAE_o \) is the Mean Absolute Error between the ground truth.
and estimated trajectory by averaging all distance discrepancies within the common time window. If the mean absolute error is below a predetermined threshold (15 ft in this study), we will consider the detected trajectory as a true positive. Otherwise, it will be considered a false-positive result.

V. EXPERIMENTAL DESIGN

The video data used in this study was from the NGSIM I80 dataset, recorded from 4:00 p.m. to 4:15 p.m. on April 13, 2005, on I-80 in Emeryville, California. The direction of traffic flow recorded was northbound. Each camera watched vehicles passing through the study area from the roof of a 30-story building adjacent to the freeway. Five lanes from four cameras were used in the study, including a high-occupancy vehicle (HOV) lane, as shown in Figure 6.

VI. MODEL EVALUATION

This section shows the STMap trajectory detection results using proposed deep learning models. It discusses both advantages and disadvantages of the proposed models for the STMap trajectory segmentation.

A. STMap Segmentation Evaluation

Figure 7 compares the proposed Res-UNet model with the baseline models using selected performance metrics on the testing dataset. Global accuracy is the number of correctly classified pixels over the total number. The mean accuracy is the average accuracy for each class. Using weighted IoU metrics to reduce the impact of imbalanced classes.

As shown in Figure 7, the UNet model family outperformed the ResNet model family. The vanilla UNet produced better results with only 70 layers than ResNet-50 with 206 layers, because of the interconnection between the encoding and decoding stages. A fully convolutional Neural Network (FCN) also yielded desirable segmentation results. Deeper ResNet-50 outperforms the less deep ResNet-18 model within the same model family. Intra-connections between decoding layers are efficient, as Res-UNet+ performs better than Res-UNet or UNet built with fewer connections. Res-UNet+ outperforms all the baseline networks.

Figure 8 compares the pixel-level performance between the proposed and reference models by inspecting the vehicle segmentation results (in yellow stripes) from a representative STMap section impacted by shadows and occlusions. The STMap shows four prominent shadow stripes created by vehicles from neighboring lanes. All tested models can distinguish the vehicle strands from static noises caused by lane markers.

The proposed Res-UNet+ model produces the cleanest boundaries for detected vehicle strands and is more robust against vehicle shadows than other deep learning models (black cycles). Most models classify shadows as vehicle strands in difficult shadow situations, which will cause missed detections by stitching multiple vehicle strands together. The proposed Res-UNet+ model was able to extract strands with fewer shadow fragments. Some small shadow pixels are detected but can be easily filtered out since they are isolated with a small number of pixels. In the oscillation scenario, the ResNet model family creates holes within vehicle strands or overlaps with nearby vehicle stands (red cycles). The Res-UNet+ model generates the best overall segmentation results.

B. Trajectory Level Evaluation

This section further evaluates the trajectory-level performance. Three detecting cases are considered, including true positive (TP), false positive (FP), and false-negative (FN). True positive rate (TPR) and false-positive rate (FPR) are used to evaluate Type I and II errors.

\[
TPR = \frac{TPs}{TPs + FNs} \quad (22)
\]

\[
FPR = \frac{FPs}{TPs + FPs} \quad (23)
\]

The proposed model is compared with the prior HASDA model developed by the research team in Table I.
Figure 9 shows that the deep learning model is more robust than the previous HASDA model under the influence of shadows. NGSIM trajectories are used as a reference to provide a microscopic inspection of the trajectory detection results. The deep learning model has greater accuracy and completeness and can capture different semantic characteristic levels. The deep-learning model detection results faithfully captured the oscillations for the stop-and-go traffic.

VII. NGSIM DATA RECONSTRUCTION

Despite its importance and frequent use in literature, NGSIM data have critical quality issues; therefore, data cleaning becomes a prerequisite before using NGSIM data. The previous techniques relying on statistical or smoothing can only marginally improve trajectory data. The proposed solution was used to identify NGSIM data quality issues and then completely reconstruct the dataset. We thoroughly cleaned the datasets and significantly enhanced them by processing the NGSIM videos from four cameras (1,000 ft area).

As shown in Figure 10, the STMap vehicle detection results and the NGSIM detection results are plotted on the raw NGSIM I-80 video. The blue lines are plotted based on the local-y and vehicle length data from NGSIM I-80 trajectory data. The red bars are the vehicle front bumpers detected by the proposed model. As shown in Figure 10, the NGSIM data have significant drifting issues, but the proposed model detects
vehicle fronts close to their actual positions in the video. The drifting problem of NGSIM data occurs more frequently when vehicles are joining or leaving a congested platoon. Figure 10 also shows a different NGSIM tracking error, where vehicle 2144 was misidentified as vehicle 2143 after vehicle 2144 changed from lane 5 to lane 4. This type of error cannot be corrected with smoothing or filtering. The fixed vehicle trajectories for 2143 and 2144 in our reconstructed NGSIM data are shown in the lower subplot of Figure 10.

The second type of error in the NGSIM data is caused by the homography projection of raw video, which assumes that all objects are on the same ground plane. The simplified 2D-plane homography assumption can lead to significant self-occlusion issues during the video transformation. For example, figure 11 shows an image projection error in NGSIM datasets for large vehicles. Due to the false premise of the 2D homography projection that all objects are on the same ground, the NGSIM data often capture the off-ground features of large vehicles (e.g., trucks and buses) that obstruct their own front bumper positions. This self-occlusion can lead to significant positioning errors for large vehicles, as illustrated by the white bus from the HOV lane in Figure 11. The proposed model only extracts features on the ground and can accurately track the large vehicle.

Figure 12 exemplifies typical shockwave errors in the NGSIM dataset caused by unstable vehicle detection and tracking. We reversed plotted the NGSIM trajectory data onto the STMap. It is found that the NGSIM data have issues at almost every stop-and-go condition, which may explain some calibration issues of microscopic car-following models based on NGSIM data [70]. The proposed model was able to generate trajectories that are more consistent with the vehicle strands in STMap. Results from the proposed model and the raw NGSIM datasets are displayed side-by-side in Figure 12.

The original NGSIM video processing method is a multi-stage process leading to systematic data quality issues. The proposed method can simplify the trajectory post-processing and validation process and improve data quality. All the validation data can be found in [71]. The remaining NGSIM videos will be reconstructed in future, and an error-free dataset will be available to the public. Lessons learned from cleaning NGSIM data are transferable to 3rd or 4th generation trajectory datasets.

VIII. CONCLUSION

This paper aims to enhance the efficiency of the data validation process by introducing a segmentation-is-tracking strategy. This approach helps identify and reconstruct faulty trajectory data, ultimately improving the overall accuracy and reliability of the extracted trajectories. The STMap-based video analytics only analyzes a few scanlines’ snapshots rather than the entire image, making it more computationally efficient than tracking-by-detection models that process the whole video frame by frame. The enhanced scanline methods addressed the issues of earlier image processing models (e.g., the HASDA model), especially concerning static noises, moving shadows, and vehicle occlusions. The data-labeling process is semi-automated by using the DMD method.

The proposed Res-Unet+ model is evaluated with the NGSIM I-80 dataset and performs better than several baseline
deep learning models. Running the proposed model is also computationally efficient, which can be deployed in parallel for each scanline. The reliable trajectory results using STMap can fundamentally address the data quality issues caused by the limitations of the conventional multi-processing steps.

The SiCK framework is model-agnostic that can incorporate any foundational segmentation model. Future direction related to this research includes applying zero-shot image segmentation using prompt-based foundational segmentation models [72]. Preliminary trajectory extraction results are shown in the following Figure 13. The segmentation visual prompts could be auto-generated through foreground/foreground subtraction algorithms.

**APPENDIX**

**NOTATIONS**

- $X, X'$: Prior Snapshot Matrix, Posterior Snapshot Matrix.
- $S, S'$: Prior STMap, Posterior STMap.
- $A, \tilde{A}$, and $\hat{A}$: Linear Transformation Matrix, Least Square Estimator, and Reduced-rank Linear Matrix.
- $s_k$: One snapshot at column $k$.
- $l_k$: Scanline of STMap at column $k$.
- $\Phi, \phi_j$: Matrix of DMD modes, and single DMD mode $j$.
- $\Lambda, \lambda_j$: Matrix of eigenvalues, and Single Eigen-value $j$.
- $B, b$: Diagonal Matrix of Amplitudes, Single amplitude.
- $\Psi$: Vandermonde matrix.

**REFERENCES**

[1] Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data, U.S. Dept. Transp. Federal Highway Admin., Washington, DC, USA, 2016. Accessed: Jan. 9, 2021. [Online]. Available: https://data.transportation.gov, doi: 10.21949/1504477.

[2] L. Li, R. Jiang, Z. He, X. Chen, and X. Zhou, “Trajectory data-based traffic flow studies: A revisit,” Transp. Res. C, Emerg. Technol., vol. 114, pp. 225–240, May 2020.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[4] V. N. Kolmogorov and R. Zabih, “Multi-scale context aggregation by dilated convolutions,” 2015, arXiv:1511.07122.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2961–2969.

[6] A. Shahbaz, J. Hariyono, and K.-H. Jo, “Evaluation of background subtraction algorithms for video surveillance,” in Proc. 21st Korea-Japan Joint Workshop Frontiers Comput. Vis. (FCV), Jan. 2015, pp. 1–4.

[7] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Dec. 2001, pp. I–I.

[8] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[10] A. Elgammal et al., “Non-parametric model for background subtraction,” in Computer Vision—ECCV 2000 (Lecture Notes in Computer Science), vol. 1843, D. Vernon, Eds. Berlin, Germany: Springer, 2000.

[11] H. T. Niknejad, A. Takeuchi, S. Mita, and D. McAllester, “On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 2, pp. 748–758, Jun. 2012, doi: 10.1109/TITS.2012.2187894.

[12] Z. Sun, G. Bobis, and R. Miller, “Object detection using feature subset selection,” Pattern Recognit., vol. 37, no. 11, pp. 2155–2176, Nov. 2004.

[13] A. Shahbaz, J. Hariyono, and K.-H. Jo, “Evaluation of background subtraction for 511 traffic cameras with U-shaped dual attention inception assist. Intervent. Proc. Int. Conf. Med. Image Comput. Comput.-image segmentation,” in Proc. Neurocomputing Joint Workshop Frontiers Comput. Vis. (FCV), Jan. 2015, pp. 31–66, 2014.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2961–2969.

[15] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, Nov. 2004.

[16] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Dec. 2001, pp. I–I.

[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] F. Zheng et al., “Anchor shot detection with diverse style backgrounds based on spatial–temporal slice analysis,” in Proc. Int. Conf. Multimedia Modeling, Berlin, Germany: Springer, 2010, pp. 676–682.

[19] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” 2015, arXiv:1511.07122.
