

# Spatial-Temporal Map Vehicle Trajectory Detection Using Dynamic Mode Decomposition and Res-UNet+ Neural Networks.

Tianya T. Zhang, *Student Member, IEEE*, Peter J. Jin, *Ph.D. Member, IEEE*

**Abstract**— This paper presents a machine-learning-enhanced longitudinal scanline method to extract vehicle trajectories from high-angle traffic cameras. The Dynamic Mode Decomposition (DMD) method is applied to extract vehicle strands by decomposing the Spatial-Temporal Map (STMap) into the sparse foreground and low-rank background. A deep neural network named Res-UNet+ was designed for the semantic segmentation task by adapting two prevalent deep learning architectures. The Res-UNet+ neural networks significantly improve the performance of the STMap-based vehicle detection, and the DMD model provides many interesting insights for understanding the evolution of underlying spatial-temporal structures preserved by STMap. The model outputs were compared with the previous image processing model and mainstream semantic segmentation deep neural networks. After a thorough evaluation, the model is proved to be accurate and robust against many challenging factors. Last but not least, this paper fundamentally addressed many quality issues found in NGSIM trajectory data. The cleaned high-quality trajectory data are published to support future theoretical and modeling research on traffic flow and microscopic vehicle control. This method is a reliable solution for video-based trajectory extraction and has wide applicability.

**Index Terms**—NGSIM, Spatial-Temporal Map, Artificial Intelligence, Traffic Detection

## I. INTRODUCTION

Video sensor has been used widely to extract vehicle trajectory data to support academic research, traffic operations, management, and design. One of the most impactful video-based trajectory datasets is the next generation simulation (NGSIM) trajectory dataset [1], which has significantly boosted the traffic flow and modeling research by revealing microscopic traffic characteristics. As highlighted by [2], although video-based trajectory data have greatly improved the types of models and the calibration/training of the models, there is still a substantial demand for high-quality, high-resolution trajectory data. Collecting useful trajectory data from traffic cameras with satisfactory accuracy is a very challenging task. The traditional trajectory extraction paradigm, which contains multi-stage algorithms, is error-prone to the varying weather, illumination, video quality, and other factors. Vehicle trajectory dataset often requires significant efforts for post-processing, such as cleaning and validation.

To solve the data quality issues in the NGSIM dataset and meet the needs of traffic flow and modeling research, this paper presented a machine learning enhanced scanline method to detect trajectory from Spatial-Temporal Map (STMap). The

Dynamic Mode Decomposition (DMD) is applied to analyze the STMap by decomposing it with different underlying structures. The DMD results were used to preprocess and prepare training data for a new deep neural network. Two existing convolutional neural network architectures were leveraged to build the Res-UNet+ model for the STMap segmentation task. This method has largely improved the scanline algorithm for vehicle detection and simplified high-fidelity trajectory data acquisition.

Furthermore, the paper also provides an efficient way to validate extracted vehicle trajectories by showing vehicle movements on the static STMap. The previous trajectory validation process relies on the direct approach by visualizing the detection and tracking results on raw video, or the indirect method by calibrating the microscopic model against both the raw and reconstructed trajectories and compare which of the two calibrated models is better. However, the spatial-temporal map allows identifying the error for the individual vehicle with minimal effort directly.

This paper is organized as follows. The related work section describes the state of the art of video analytics for vehicle trajectory extraction. The methodology section presents the proposed methods, including dynamic mode decomposition and Res-UNet+ convolutional network for STMap trajectory detection. Next, the experimental design section describes the dataset and model configurations, and the result analysis section analyzes the performance of the proposed models compared to baseline models. The final section concludes the paper and addresses future works.

## II. RELATED WORK

### A. High-Resolution Vehicle Trajectory Datasets

Three significant high-resolution vehicle trajectory datasets and their processing methods are reviewed in this section.

**NGSIM:** NGSIM trajectory dataset, which is a great asset to the transportation research community, are generated with traffic videos taken from high-rise buildings. NGSIM 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. The NGSIM dataset has been used to calibrate and evaluate traffic flow models as ground-truth data, demonstrate driving behavior or traffic phenomena, and conduct traffic-state estimation and prediction [3][4].

Submitted January 22<sup>nd</sup>, 2021. Revised May 31<sup>st</sup>, 2021. Accepted Oct 1<sup>st</sup>, 2021.

T. T. Zhang is a PhD candidate at the Civil and Environmental Engineering Department in Rutgers University, Piscataway, NJ 08854 USA (e-mail: tz140@soe.rutgers.edu).

P. J. Jin, is an Associate Professor at Civil and Environmental Engineering Department in Rutgers University, Piscataway, NJ 08854 USA (e-mail: peter.j.jin@rutgers.edu)

However, a growing amount of literature has uncovered the underlying systematic errors in the NGSIM dataset. Some research [5][6][7][8][9] examined the performance issue and proposed denoising methods based on statistic distributions, filtering & smoothing, traffic-informed constraints, and information theory [10].

**HighD & inD Dataset:** Krajewski et al. [11] published a HighD dataset that consists of 110,500 vehicles collected by Drones on German highways. The same group published an inD trajectory dataset of intersection road users containing pedestrians, bikers, vans, and others. The object detection algorithm used to generate HighD dataset is a U-Net semantic segmentation algorithm.

**pNEUMA:** In contrast to the highway vehicle trajectory dataset, Barmounakis and Geroliminis [12] presented a complete urban dataset collected from a swarm of drones named pNEUMA (New Era of Urban traffic Monitoring with Aerial footage). Their project was conducted with a commercial traffic platform [13]. The dataset covers a congested area with more than 100 km-lanes and around 100 intersections under the multi-modal traffic environment using ten drones. The raw pNEUMA dataset does not contain lane information, which requires users to apply an additional lane identification method.

#### B. Computer Vision Algorithms for Traffic Video Analysis

Traffic detection is a part of the object detection problem in computer vision. Great progress has been made in recent years with the rising of deep learning. Object detection involves not only recognizing the objects in targeted classes but precisely localizing each object. Table 1 is 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 model [20][21][22][23], feature-based model [24][25], and scanline methods [26][27].

Semantic segmentation is another computer vision task related to traffic video analytics, which predicts class labels at the pixel level for each image. The challenges lie in 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 in 2015 to process biomedical images [33]. The original vanilla U-Net has many variations with similar U-shape architecture, leading to a series of models as the U-Net model family.

Multi-Object Tracking (MOT) is vital for many applications in computer vision and has been extensively investigated. The object tracking methods can be classified into two types: online and offline tracking. The online tracking uses only current and previous frames. The long-term movements are embedded into a state-space for memorization [34-36]. The offline tracking is based on global optimization algorithms using a collection of time-series information of desired objects [37-39]. The recent deep learning approaches have gained tremendous momentum and successfully enhanced the performance for MOT, including Siamese Networks, Attention and Transformer, and Recurrent Neural Networks [40-43]. Some other practices consider the tracking issues from the mathematic formulation based on data association or machine learning models to extract trajectory features for clustering [44-46].

#### C. Scanline Method

The Scanline method stems from the Spatial-Temporal Slice (STS) structure used in computer vision literature. Early spatial-temporal slice methods were introduced to solve the structure from motion (SFM) problem for cameras on moving platforms [47][48]. Later, the STS method was also used for object and pedestrian detection [49][50]. In transportation research, the technique is named as scanline method, which 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, *Spatial-Temporal Map (STMap)* is obtained. On STMap, the horizontal axis shows the time progression, and the vertical axis contains distance information.

Two types of scanlines are used in traffic detection, lateral and longitudinal scanlines. The lateral scanline is a cross-section scanline across a lane, whereas the longitudinal scanline is along traffic direction. The lateral scanline method was intended primarily for traffic counting [51] and speed measurement [52]. The longitudinal scanline method was used for vehicle tracking [53][54] and detection [55]. However, most previous scanline methods were only used to estimate macroscopic parameters such as traffic count, headway, and spot speed.

A recent study [56] 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 STMap, the extraction of the pixel trajectories from 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

TABLE I  
VEHICLE DETECTION METHODOLOGIES AND ALGORITHMS

Categories	Typical Algorithms
Shape Based Method [13 ~15]	Deformable Part-Based Model Eigen Vehicle Model
Feature Based Methods [24,25]	SIFT/Surf Harr-like Optical Flow
Background Modeling [17 ~ 19]	Frame Difference Approximate Median Mixture of Gaussian
Deep Learning Model [20~23]	Fast/Faster R-CNN Mask R-CNN YOLO v1 ~ v5.
Scanline Method [26,27]	Lateral Scanline Longitudinal Scanline

removal, edge detection, and morphological image operations. Some of the image processing techniques, such as edge detection, are vulnerable to noises. LiDAR data were combined with the scanline method to facilitate the calibration process in a later paper [57].

#### D. Dynamic Mode Decomposition (DMD)

Dynamic Mode Decomposition is a data-driven analytic method that integrates Fourier transforms and singular value decomposition (SVD). DMD method was first introduced by Schmid [58] 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 [59], therefore reducing the dimensions of complex systems efficiently without losing accuracy [60]. 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.

$$X = \begin{bmatrix} | & \cdots & | \\ x_1 & \ddots & x_{m-1} \\ | & \cdots & | \end{bmatrix}, \quad X' = \begin{bmatrix} | & \cdots & | \\ x_2 & \ddots & x_m \\ | & \cdots & | \end{bmatrix} \quad (1)$$

where  $x_k$  ( $k = 1, \dots, m$ ) is a vector 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  $A$ :

$$X' = AX \quad (2)$$

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

$$\hat{A} = \underset{A}{\operatorname{argmin}} \|X' - AX\|_F^2 \quad (3)$$

where  $\hat{A}$  is an estimator of matrix operator  $A$ , which is computed by minimizing the Fresenius norm  $\|X' - AX\|_F^2$ .

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

$$\hat{A} \approx X'X^\dagger \quad (4)$$

Where  $X^\dagger$  is obtained by using Moore-Penrose pseudoinverse.

For the DMD algorithm, instead of solving the matrix operator  $A$  directly,  $A$  is often solved by the eigendecomposition of  $A$  after the proper orthogonal decomposition (POD).

**Step 1.** Factorization of matrix  $X$  using SVD.

$$X \approx U \Sigma V^* \quad (5)$$

Where  $U$  and  $V$  are column and row orthonormal bases of matrix  $X$ ,  $*$  denotes the complex conjugate transpose. Columns in  $U$  are orthonormal, such that  $U^*U = I$ ; Likewise,  $V^*V = I$ .

**Step 2.** Reduce the dimension of  $A$  and obtain of  $\tilde{A}$  by projecting  $A$  onto  $U_r$ :

$$\tilde{A} = U_r^* A U_r = U_r^* X' V_r \Sigma_r^{-1} \quad (6)$$

**Step 3.** Compute eigenvalue of  $\tilde{A}$ :

$$\tilde{A}W = W\Lambda_r \quad (7)$$

where the column of  $W$  are eigenvectors and  $\Lambda_r$  is the diagonal matrix of eigenvalues  $\lambda_j$ .

**Step 4.** The eigenvector of  $A$  can be reconstructed using

$$\Phi = X'V\Sigma^{-1}W \quad (8)$$

The above methods significantly reduce the complexities of the regression problem of estimating the full matrix  $A$  into calculating elements in the diagonal and sparse matrices. 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 [61]. The way to stack state vectors together into a large matrix and identify its coherent structure is also named Method-of-Snapshots by Sirovich [62].

### III. METHODOLOGY

#### A. STMap Generation

As illustrated in Figure 1, STMap is generated by stacking longitudinal scanlines ( $l_1, l_2, l_3, \dots, l_m$ ) 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.

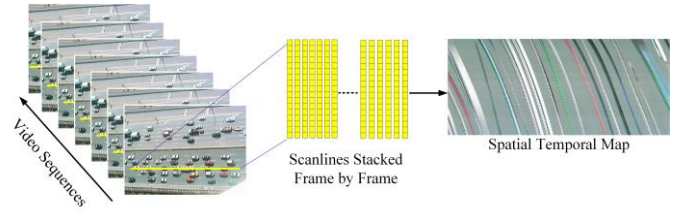


Figure 1 STMap Generation Using the Snapshot Vector

The color pixels moving simultaneously in the STMap represent a unique vehicle passing along the pre-defined scanline. Our purpose is to segment each vehicle strand from the STMap to detect trajectories.

#### B. Dynamic Mode Decomposition (DMD) for STMap Segmentation

The scanline pixel values at each frame can be considered the state of traffic dynamic at a particular timestamp. Traffic state of  $l_{x+1}$  at the time  $(x + 1)$  is assumed to relate to the previous traffic state of  $l_x$  by linear time-dependent operator  $A$ , which reflects the time evolution of scanline pixels.

$$l_{x+1} = A l_x \quad (9)$$

The STMap can be formulated as follows:

$$S = \begin{bmatrix} | & \cdots & | \\ l_1 & \ddots & l_{m-1} \\ | & \cdots & | \end{bmatrix}, \quad S' = \begin{bmatrix} | & \cdots & | \\ l_2 & \ddots & l_m \\ | & \cdots & | \end{bmatrix} \quad (10)$$

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

$$S' = AS \quad (11)$$

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  $\Phi$ .

$$A\Phi = \Phi\Lambda \quad (12)$$

The column of  $\Phi$  are eigenvectors comprising of the dominant mode  $\Phi_j$  and  $\Lambda$  is the diagonal matrix of eigenvalues

$\lambda_j$ . The STMap can be reconstructed using first  $k^{th}$  modes, where  $k \leq \min(n, m)$ .

$$STMap \approx \Phi BV = \begin{pmatrix} \phi_{11} & \dots & \phi_{1k} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \dots & \phi_{nk} \end{pmatrix} \begin{pmatrix} b_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & b_k \end{pmatrix} \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_k & \dots & \lambda_k^{m-1} \end{pmatrix} \quad (13)$$

where  $\Phi$  contains the dominant modes from the STMap, 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.

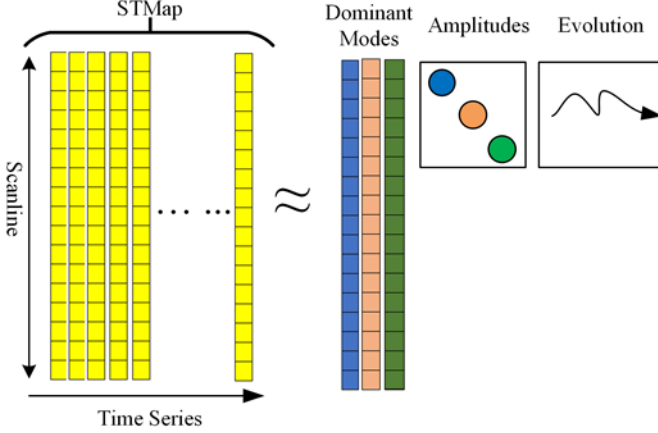


Figure 2 Decompose Spatial-Temporal Map into Dominant Modes

A scanline vector  $l_t$  at frame  $t \in 1, \dots, m$  can be estimated as follows:

$$\tilde{l}_t = \sum_{j=1}^k b_j \phi_j \lambda_j^{t-1} \quad (12)$$

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 scanline as follows.

$$\tilde{l}_1 = \sum_{j=1}^k b_j \phi_j \quad (13)$$

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

$$\tilde{B} = \underset{B}{\operatorname{argmin}} \|l_1 - \Phi B\| \quad (14)$$

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.

$$S_{DMD} = \text{background} + \text{foreground} = \sum_p b_p \phi_p \lambda_p^{t-1} + \sum_{j \neq p} b_j \phi_j \lambda_j^{t-1} \quad (15)$$

where  $|\lambda_p| = 1$ .  $t \in 1, \dots, m$  indicates the frame number.

As displayed in Figure 3, the background is time-independent and has a minimal DMD eigenvalue ( $\approx 10^{-6}$ ) compared with the foreground modes' eigenvalues. Figure 3 (D) plots the

oscillations of selected methods over time. The amplitude of the background mode with the lowest frequency indicates the least variations over time.

As shown in Figures 3 (A) and 3 (B), DMD's foreground detection results are not perfect for trajectory extraction on STMap. However, they are clear enough to preprocess and generate training data for deep learning models.

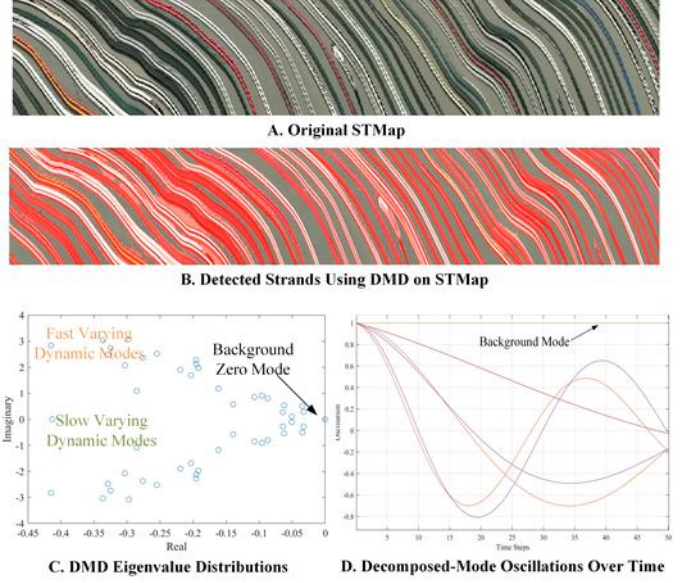


Figure 3 STMap and Its Dynamic Modes

### C. Res-UNet+ Model for STMap Segmentation

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 blocks replace the original encoders in the UNet model. The interconnections between encoding and decoding layers were designed to reduce the semantic gap. We added the intra-connections among different levels of decoding stages. Many segmentation studies [63-65] show that features learned from different scales often entail different information. Lower-level layers capture the boundary of objects, whereas higher-level layers explore the localization of the targeted objects. In the vanilla UNet architecture, there are only interconnections between the corresponding level of encoders and decoders. To enable lower-level decoder information to pass to higher-level decoders, we concatenate all the decoder layers to localize better and learn representations in the segmentation network. The multi-scale intra-connection and depth of each layer are shown in Figure 4, and the network is named Res-UNet+.

The encoder layer using two branch ResNet backbone, including upper branch and lower branch, is defined as the following:

$$H_l^i = \text{ReLU}(\text{id}(H^{i-1}) + F_1(H^{i-1})) \quad (16)$$

$$H_u^i = F_2(H_l^i) + F_1(H_l^i) \quad (17)$$

$$H^i = \text{ReLU}(H_u^i) \quad (18)$$

where  $H_l^i$  is lower branch output of  $i^{th}$  layer encoder.  $H_u^i$  is upper branch output of  $i^{th}$  layer encoder.  $H^i$  is the  $i^{th}$  layer output.  $F_1 = \text{BN}(\text{Conv3}(\text{ReLU}(\text{BN}(\text{Conv3}(x))))$ , and  $\text{Conv3}$



is 3 by 3 convolution operator.  $F_2 = BN(Conv3(x))$ , BN is Batch Normalization.

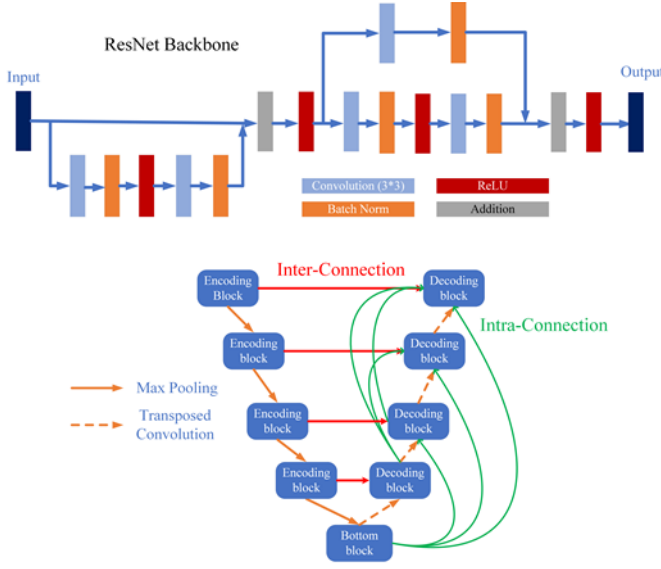


Figure 4 Proposed Res-UNet+ Topological Structure and Building Block

The decoder layer with intra-connections between different levels is described as the following:

$$L_{de}^i = \begin{cases} A^2([L_{en}^n, A_T(L_{bottom})]), & i = n \\ A^2([A(L_{en}^i), A_T(L_{de}^j)]_{j=i-1}^{j=n}), & i < n \end{cases} \quad (19)$$

where  $L_{de}^i$  is the  $i^{th}$  decoder layer output,  $L_{en}^i$  is the  $i^{th}$  encoder layer output,  $[\cdot]$  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

##### 1) Baseline Models in the Study

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

**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. In this paper, we 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.

**UNet:** The vanilla UNet 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 UNet 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.

**Res-UNet:** For the Res-UNet model, we did not add intra-connection to integrate the information from all encoder layers. Our Res-UNet architecture reformed the UNet model

architecture by replacing the original encoder layer with a two-branch ResNet block.

**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 [66].

**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 [67].

**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 [68].

##### 2) Dataset and Augmentation

Unlike other data labeling process, which is highly specialized and requires a lot of expertise and experience, 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. Furthermore, the patterns of vehicle strands compared to background pixels are easy to be partitioned.

In this study, the STMap training dataset was created using four 15-minute NGSIM I-80 videos. We obtained 20 STMaps from 20 lanes, which were then cropped into 1000 512\*512 images. Finally, we augmented the 1000 images to get more datasets by rescaling, shearing, and translating the generated 512\*512 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.

##### 3) 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 1060 (6 GB memory). The maximum epoch is 10, equivalent to a training time of around 5 hours. Once trained, the proposed model takes 2 minutes to process the STMap from a 15-minute video.

All training sets 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. Our goal is to correctly classify all pixels belonging to the vehicle strands from STMap and then extract the vehicle strands boundary.

##### 4) Trajectory Extraction

After the segmentation and obtaining the binary masks for all vehicle strands from STMap, the following step is to extract 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.

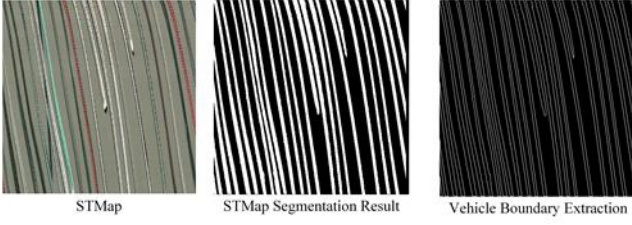


Figure 5 Extracting Vehicle Trajectory from STMap Segmentation

The pixel trajectories from STMap can then be converted into pixel movements on the original video. Then the video trajectory will be converted into NGSIM Local-y coordinates using the method in [56].

##### 5) 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 the percentage of correctly classified pixels to the total number of pixels for each class regarding the ground truth data.

$$Acc = \frac{TP}{(TP + FN)} \quad (20)$$

where TP is the count of true positives; FN is the count of false negatives. However, Acc is a primary metric and needs to use in junction with other metrics for a complete evaluation.

The Jaccard coefficient is used to measure the similarity between two sets A and B (also known as intersection over union or IoU) is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (21)$$

Jaccard coefficient is an overlap index that quantifies the agreement between two segmented image areas. The Jaccard coefficient can also be expressed in terms of true positives (TP), false positives (FP), and false negatives (FN) as:

$$J(A, B) = \frac{TP}{(TP + FP + FN)} \quad (22)$$

The third performance metric is BF Score, which computes the contour-matching score between model-predicted results and ground truth data. As shown in the following equation, the BF Score is defined as the harmonic mean between precision and recall values to decide whether the points on the boundary have been matched.

$$BF = 2 * \frac{precision * recall}{(recall + precision)} \quad (23)$$

The error measure for the trajectory detection results is the mean absolute error (MAE) of all trajectory points.

$$MAE_o = \frac{1}{N} \sum_{t=1}^N |\hat{y}_o(t) - y_o(t)| \quad (24)$$

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 trajectory and the estimated trajectory by averaging all distance discrepancies within the common time window. If the mean absolute error is below a pre-determined 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 I80-1 dataset, which was 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.

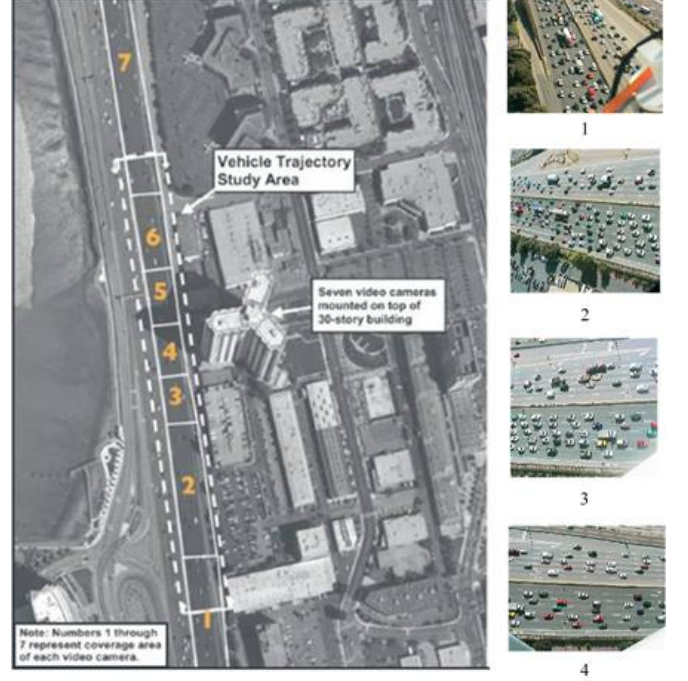


Figure 6 NGSIM I-80 Survey Area and 4 Study Cameras [1]

## VI. MODEL EVALUATION

In this section, we show the STMap trajectory detection results using proposed deep learning models. We will discuss both advantages and disadvantages of the proposed models for the STMap trajectory segmentation quantitatively and qualitatively. We will also compare the trajectory detected by the proposed model against the NGSIM trajectory.

### A. STMap Segmentation Evaluation

Figure 7 compares the proposed Res-UNet model and 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 of pixels. The mean accuracy is the average accuracy for each class. Mean IoU is the average IoU score across all classes. Finally, the weighted IoU is the IoU score weighted by the number of pixels for each class in the image. Using weighted metrics is to reduce the impact of imbalanced classes.

As shown in Figure 7, the UNet model family outperformed the ResNet model family in the segmentation task. 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 stage. 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+ has better performance than Res-UNet or UNet built with fewer connections. Res-UNet+ outperforms all the rest networks.

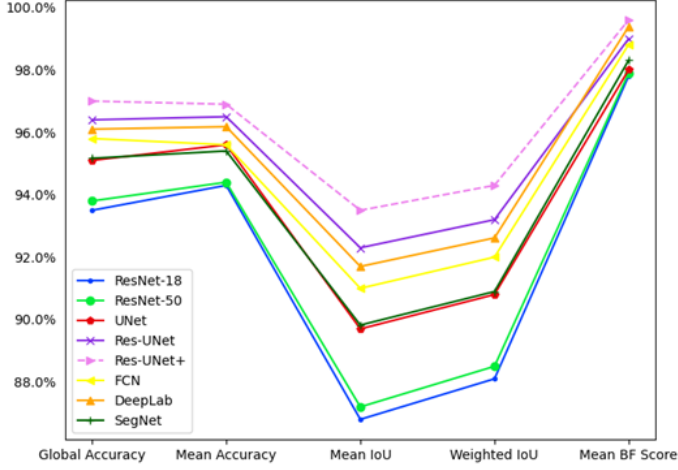


Figure 7 Model Performance Comparison with Baseline Models

Figure 8 performs a detailed comparison of the trajectory-level performance between the proposed model and the 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 major 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. Res-UNet+ is also more robust against vehicle shadows than other deep learning models tested (black cycles). Most of those models classify shadows as vehicle strands in difficult shadow situations, which will cause false detections. The false detected shadows stitch multiple vehicle strands together will result in missed detections. The proposed Res-UNet+ model was able to extract strands with fewer shadow fragments. Some small shadow pixels are detected, but those 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) as used to evaluate Type I and II errors.

$$TPR = TP / (TP + FN) \quad (26)$$

$$FPR = FP / (TP + FP) \quad (27)$$

The proposed model is compared with the prior HASDA model developed by the research team [56] in Table II.

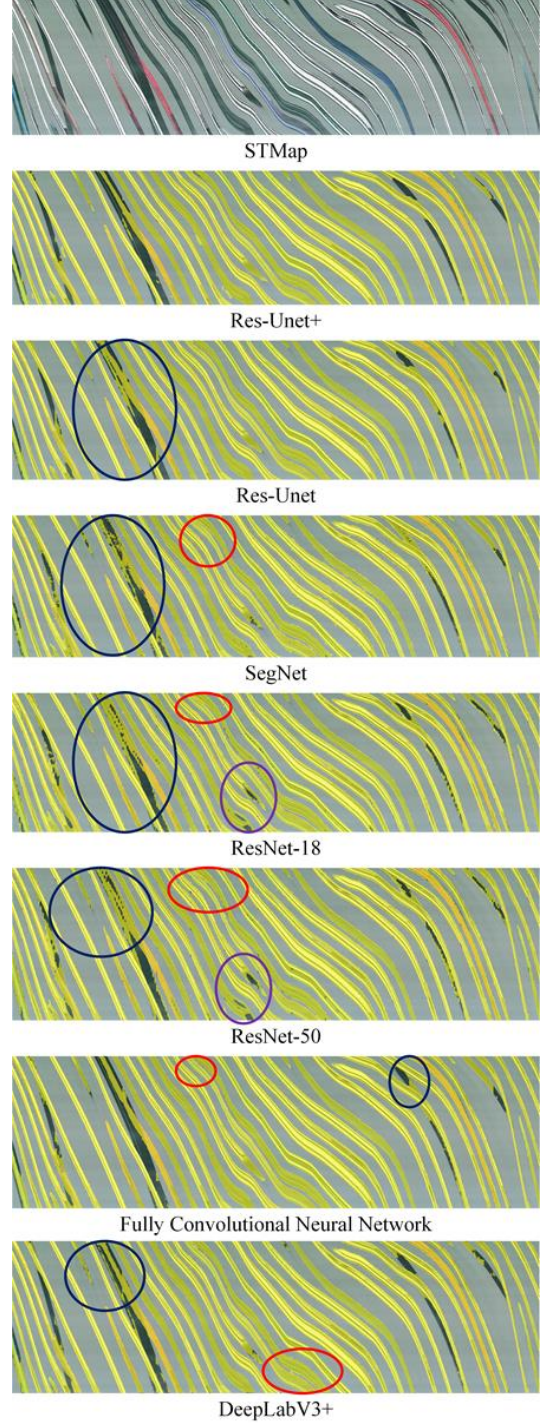


Figure 8 Segmentation Results Using Proposed and Baseline Models (Black Cycle: Shadows; Red Cycle: Overlapping; Purple Cycle: Holes)

TABLE II  
MODEL PERFORMANCE AT TRAJECTORY LEVEL

	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5
<b>Res-UNet+ Model</b>					
Positive Detect Rate	97.1%	98.9%	98.1%	98.0%	96.5%
False Detect Rate	3.2%	2.7%	3.2%	4.2%	3.0%
<b>HASDA Model</b>					
Positive Detect Rate	96.5%	96.9%	94.1%	89.58%	92.3%
False Detect Rate	5.8%	6.9%	2.4%	9.5%	3.5%

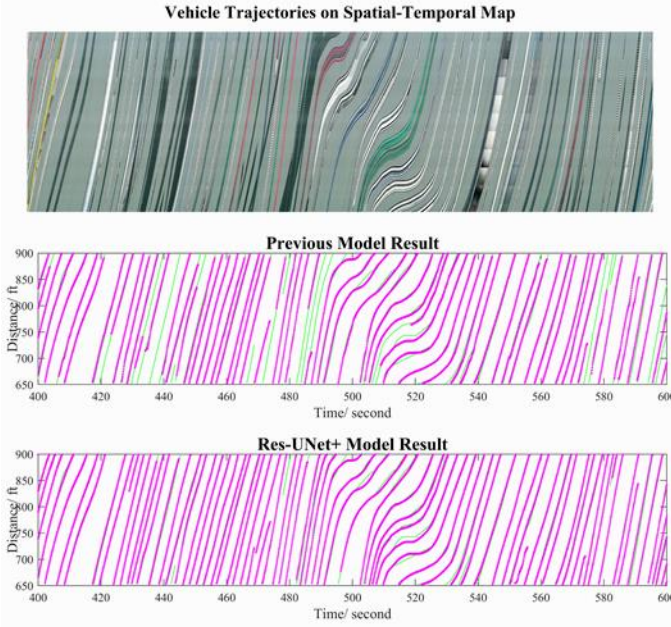


Figure 9 HASDA Model [56] and Res-UNet+ Model (Purple: Reconstructed Trajectories; Blue: NGSIM Trajectory)

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 the reference to provide a microscopic inspection of the trajectory detection results. The deep learning model has greater accuracy and completeness with its capability to capture different semantic characteristics levels. For the stop-and-go traffic, the deep-learning model detection results captured the oscillations very faithfully.

## 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 to study traffic flow characteristics or models. The previous filtering techniques relying on statistical filtering or smoothing can only improve trajectory data marginally. The proposed models were used to identify NGSIM data quality issues and then reconstructed the dataset completely. By processing the NGSIM videos from four cameras (1000 ft area), we thoroughly cleaned the datasets and achieved significant quality improvement.

As shown in Figure 10, the vehicle detection results and the NGSIM detection results are plotted in detail in 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 detected vehicle fronts by the proposed model. As shown in Figure 10, the NGSIM data show a significant drifting issue. At the same time, the proposed model detects vehicle fronts very 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 another NGSIM tracking error that 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.

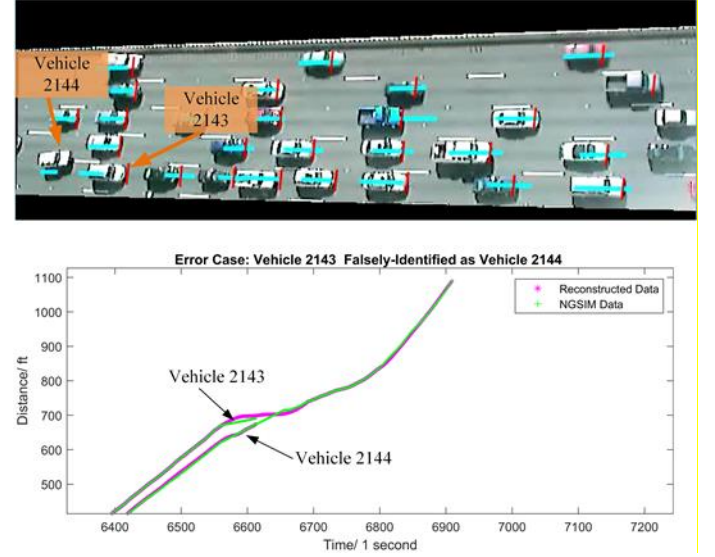


Figure 10 Video Validation of Reconstructed NGSIM data (Blue: NGSIM; Red: Detection results of the proposed model)

The second type of error in the NGSIM data is caused by the homography projection of raw video that assumes all objects are on the same ground plane. The simplified 2D plane homography assumption during the video transformation can lead to significant self-occlusion issues. Figure 11 shows another type of error in NGSIM datasets for large vehicles. Due to the false assumption 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), resulting in self-occlusion issues. The self-occlusion can lead to significant positioning errors for large vehicles, as illustrated by the white bus from the HOV lane in Figure 11. However, the proposed model is applied directly to the raw NGSIM video data and only extracts features on the ground and can track the large vehicle accurately.

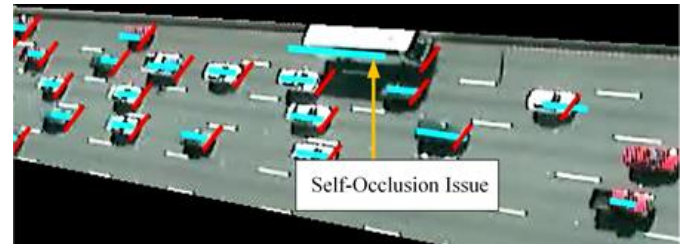


Figure 11 Vehicle Detection Error Caused by Self-Occlusion (Blue: NGSIM; Red: Detection results of the proposed model)

Figure 12 exemplifies typical errors in shockwaves in the NGSIM dataset caused by the unstable vehicle detection and tracking. We reversely plot the NGSIM trajectory data onto the STMap and found that the NGSIM data have issues at almost every shockwave condition. This may explain some of the calibration issues of microscopic car-following models based on NGSIM data [69]. The proposed model was able to generate trajectories much 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.



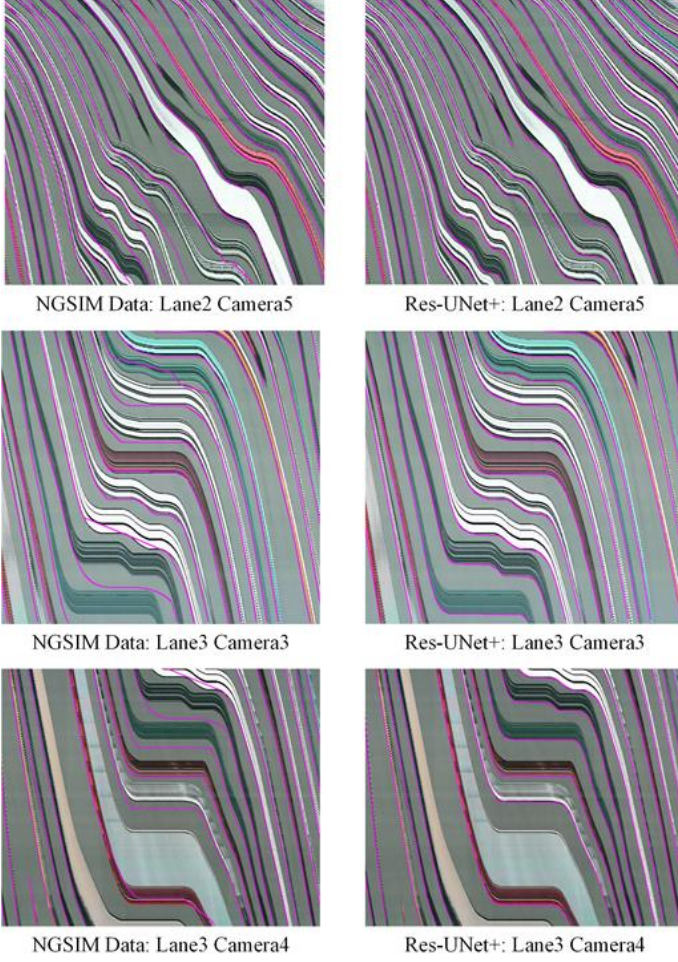


Figure 12 NGSIM Trajectories with Issues v.s. ResUNet+ Trajectories  
(Purple: Pixel Trajectory)

The original NGSIM video processing method is a multi-stage process that has led to some data quality issues. The proposed method can simplify the trajectory generation and validation process and improve data quality. All the validation videos can be found in [70]. The reconstructed NGSIM dataset is made openly available to the public. The code for the Res-UNet+ model and DMD model will also be publicly available.

### VIII. CONCLUSION

The proposed Res-UNet+ integrates two prevailing CNN structures to extract vehicle trajectories from the Spatial-Temporal Map (STMap). The STMap-based video analytics only analyzes the snapshot of a few scanlines from each video frame rather than the entire image, making it more computationally efficient than tracking-by-detection models that process the whole video frame by frame. The machine-learning-enhanced scanline methods addressed the issues of earlier image processing models (e.g., the HASDA model), especially against 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 parallelism for each scanline. The reliable trajectory results using STMap can fundamentally address the data quality issues caused by the limitations of the conventional tracking-by-detection paradigm. In response to the growing appeal for accurate trajectory datasets, this research reconstructed the NGSIM dataset using raw video data and solved nearly all error cases.

Future work of the research includes more extensive testing and sensitivity analysis under different traffic conditions (e.g., arterials, intersection, highway merging) and camera conditions (e.g., roadside CCTV cameras and Drones). In addition, the online version of the proposed algorithm will also be developed to support real-time intelligent transportation applications.

### IX. APPENDIX: NOTATION

$X, X'$ : Prior Snapshot Matrix, Posterior Snapshot Matrix.

$S, S'$ : Prior STMap, Posterior STMap.

$A, \hat{A}$  and  $\tilde{A}$ : Linear Transformation Matrix, Least Square Estimator, and Reduced-rank Linear Matrix.

$x_k$ : 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 Eigenvalue  $j$ .

$B, b$ : Diagonal Matrix of Amplitudes, Single amplitude.

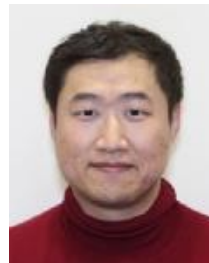
$\mathcal{V}$ , Vandermonde matrix.

### X. REFERENCE

- [1]. US Federal Highway Administration. Next-Generation Simulation Program (NGSIM). <http://ops.fhwa.dot.gov/trafficanalysisitools/ngsim.htm> 2006.
- [2]. Li, L. et al. (2020). Trajectory data-based traffic flow studies: A revisit. *Trans. Res. Part C: Emerg. Technol.*, 114, 225-240.
- [3]. He, Z., 2017. Research based on high-fidelity NGSIM vehicle trajectory datasets: A review. *Res. Gate*, pp.1-33.
- [4]. Ahn, S., et al., 2019. Traffic Flow Theory and Characteristics. *Centennial Papers*.
- [5]. Punzo, V., et al., 2011. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. *Trans. Res. Part C: Emerg. Technol.*, 19(6), pp.1243-1262.
- [6]. Zheng, Z. and Washington, S., 2012. On selecting an optimal wavelet for detecting singularities in traffic and vehicular data. *Trans. Res. Part C: Emerg. Technol.*, 25, pp.18-33.
- [7]. Punzo, V., Ciuffo, B. and Montanino, M., 2012. Can results of car-following model calibration based on trajectory data be trusted? *Trans. Res. Rec.*, 2315(1), pp.11-24.
- [8]. Montanino, M. and Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Trans. Res. Part B: Method.*, 80, pp.82-106.
- [9]. Venthuruthiyil SP, Chunchu M. Vehicle path reconstruction using Recursively Ensembled Low-pass filter (RELP) and

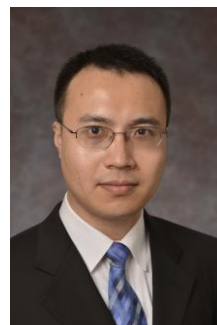
- adaptive tri-cubic kernel smoother. *Trans. Res. Part C: Emerg. Technol.*, 2020 Nov 1;120:102847.
- [10].Coifman, B. and Li, L., 2017. A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. *Trans. Res. Part B: Method.*, 105, pp.362-377.
- [11].Krajewski, R., et al., 2018, November. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st Int. Conf. on Intell. Trans. Syst. (ITSC)* (pp. 2118-2125). IEEE.
- [12].Barmounakis, E. and Geroliminis, N., 2020. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Trans. Res. part C: Emerg. Technol.*, 111, pp.50-71.
- [13].“Deep Traffic Video Analysis.” *DataFromSky*, 6 Jan. 2021, datafromsky.com/.
- [14].Felzenszwalb, P.F, et al., 2009. Object detection with discriminatively trained part-based models. *IEEE Trans. on pattern analysis and machine intelligence*, 32(9), pp.1627-1645.
- [15].H. Tehrani Niknejad, et al., "On-Road Multivehicle Tracking Using Deformable Object Model and Particle Filter With Improved Likelihood Estimation," in *IEEE Trans. on Intell. Trans. Syst.*, vol. 13, no. 2, pp. 748-758, June 2012, doi: 10.1109/TITS.2012.2187894.
- [16].Sun, Z., Bebis, G. and Miller, R., 2004. Object detection using feature subset selection. *Pattern Recognit.*, 37(11), pp.2165-2176.
- [17].Elgammal A., Harwood D., Davis L. (2000) Non-parametric Model for Background Subtraction. In: Vernon D. (eds) *Comput. Vision — ECCV 2000*. ECCV 2000. Lecture Notes in Comput. Science, vol 1843. Springer, Berlin, Heidelberg
- [18].Alawi, M. A., Khalifa, O. O., & Islam, M. R. (2013). Performance comparison of background estimation algorithms for detecting moving vehicle. *World Appl. Sci. J.*, 21, 109-114.
- [19].Shahbaz, A., Hariyono, J., & Jo, K. H. (2015, January). Evaluation of background subtraction algorithms for video surveillance. In 2015 21st Korea-Japan Joint Workshop on Frontiers of Comput. Vision (FCV) (pp. 1-4). IEEE.
- [20].Wu, X., Sahoo, D. and Hoi, SC, 2020. Recent advances in deep learning for object detection. *Neurocomputing*.
- [21].Girshick, R. (2015). Fast r-cnn. In *Proc. of the IEEE Int. Conf. on Comput. Vision* (pp. 1440-1448).
- [22].Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Syst.* (pp. 91-99).
- [23].He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proc. of the IEEE Int. Conf. on Comput. Vision* (pp. 2961-2969).
- [24].D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. of Comput. Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [25].Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In *Proc. of the 2001 IEEE Comput. society Conf. on Comput. vision and Pat. Recognit.*. CVPR 2001 2001 Dec 8 (Vol. 1, pp. I-I).
- [26].Ngo, Chong-Wah, et al. "Motion analysis and segmentation through spatio-temporal slices processing." *IEEE Trans. on Image Processing* 12, no. 3 (2003): 341-355.
- [27].Zheng, F., et al., 2010, January. Anchor shot detection with diverse style backgrounds based on spatial-temporal slice analysis. In *Int. Conf. on Multimedia Modeling* (pp. 676-682). Springer, Berlin, Heidelberg.
- [28].Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*. 2015 Nov 23.
- [29].Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems* 25. Curran Associates, Inc. 1097–1105.
- [30].Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [31].Szegedy, C et al., 2015. Going deeper with convolutions. In *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.* (pp. 1-9).
- [32].Kaiming He, et al., "Deep residual learning for image recognition", *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, pp. 770-778, 2016.
- [33].Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. on Medical image computing and Computer-assisted Intervention* (pp. 234-241). Springer, Cham
- [34].A.Dehghan, Y. Tian, P. H. S. Torr, and M. Shah. Target identity-aware network flow for online multiple target tracking. In 2015 IEEE CVPR, pages 1146–1154, 2015.
- [35].K. Fang, Y. Xiang, X. Li, and S. Savarese. Recurrent autoregressive networks for online multi-object tracking. In 2018 IEEE WACV, pages 466–475, 2018. 2
- [36].Xu Gao and Tingting Jiang. Osmo: Online specific models for occlusion in multiple object tracking under surveillance scene. In *Proc. of the 26th ACM Int. Conf. on Multimedia, MM '18*, page 201–210, New York, NY, USA, 2018.
- [37].Berclaz J, Fleuret F, Turetken E, Fua P. Multiple object tracking using k-shortest paths optimization. *IEEE PAML*. 2011 Feb 4;33(9):1806-19.
- [38].Dehghan A, Modiri Assari S, Shah M. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proc. of the IEEE CVPR 2015* (pp. 4091-4099).
- [39].Tang S, Andres B, Andriluka M, Schiele B. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE CVPR 2015* (pp. 5033-5041).
- [40].Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In 2017 IEEE ICCV, pages 4846–4855, 2017
- [41].B. Cuan, K. Idrissi, and C. Garcia. Deep siamese network for multiple object tracking. In 2018 IEEE 20th International Workshop MMSP, pages 1–6, Aug 2018.
- [42].Meinhardt T, Kirillov A, Leal-Taixe L, Feichtenhofer C. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*. 2021 Jan 7.
- [43].Anton Milan, S. Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Proceedings of AAAI'17*, page 4225–4232.
- [44].Trajectory-based Scene Understanding using Dirichlet Process Mixture Model, *IEEE Transactions on Cybernetics*, 2019.
- [45].Temporal unknown incremental clustering model for analysis of traffic surveillance videos, *IEEE Trans. Intell. Transp. Syst.*, 20(5):1762-1773, 2019. 2018.

- [46].Unsupervised tracking with the doubly stochastic Dirichlet process mixture model," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2594–2599, Sep. 2016.
- [47].Adelson, E.H. and Bergen, J.R., 1985. Spatiotemporal energy models for the perception of motion. *Josa a*, 2(2), pp.284-299.
- [48].Bolles, R.C., Baker, HH and Marimont, D.H., 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. J. of Comput. vision*, 1(1), pp.7-55.
- [49].Riquebourg, Y. and Boutheymy, P., 2000. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), pp.797-808.
- [50].Ran, Y., Chellappa, R. and Zheng, Q., 2006, August. Finding gait in space and time. In *18th Int. Conf. on Pattern Recognit. (ICPR'06)* (Vol. 4, pp. 586-589). IEEE.
- [51].Yue, Y., 2009. A traffic-flow parameters evaluation approach based on urban road video. *Int. J. Intell. Eng. Syst*, 2(1), pp.33-39.
- [52].Kadiķis, R. and Freivalds, K., 2013, December. Vehicle classification in video using virtual detection lines. In *Sixth Int. Conf. on Machine Vision (ICMV 2013)* (Vol. 9067, p. 90670Y). Int. Society for Optics and Photonics.
- [53].N. Jacobs, M. Dixon, S. Satkin and R. Pless, "Efficient tracking of many objects in structured environments," *2009 IEEE 12th Int. Conf. on Comput. Vision Workshops, ICCV Workshops*, Kyoto, 2009, pp. 1161-1168, doi: 10.1109/ICCVW.2009.5457477.
- [54].Dixon, M., Jacobs, N. and Pless, R., 2009, August. An efficient system for vehicle tracking in multi-camera networks. In *2009 Third ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC)* (pp. 1-8). IEEE.
- [55].Malinovsky, Y., Wu, Y.J. and Wang, Y., 2009. Video-based vehicle detection and tracking using spatiotemporal maps. *Trans. Res. Rec.*, 2121(1), pp.81-89.
- [56].Zhang, T. and Jin, P.J., 2019. A longitudinal scanline-based vehicle trajectory reconstruction method for high-angle traffic video. *Trans. Res. part C: Emerg. Technol.*, 103, pp.104-128.
- [57].Zhang, T. T. et al. (2020) 'Longitudinal-Scanline-Based Arterial Traffic Video Analytics with Coordinate Transformation Assisted by 3D Infrastructure Data', *Trans. Res. Rec.* doi: 10.1177/0361198120971257.
- [58].Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. *J. offluid mechanics*, 656, pp.5-28.
- [59].Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L. and Kutz, J.N., 2013. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041*.
- [60].Kutz, J.N., Brunton, S.L., Brunton, B.W. and Proctor, J.L., 2016. *Dynamic mode decomposition: data-driven modeling of complex Syst.*. Society for Industrial and Appl. Math..
- [61].Erichson NB, Brunton SL, Kutz JN. Compressed dynamic mode decomposition for background modeling. *J. of Real-Time Image Processing*. 2019 Oct 1;16(5):1479-92.
- [62].Lawrence Sirovich. Turbulence and the dynamics of coherent structures. I—III. *Quart. Appl. Math.*, 45(3):561–590, 1987.
- [63].Huang H, Lin L, Tong R, Hu H, Zhang Q, Iwamoto Y, Han X, Chen YW, Wu J. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. In *ICASSP 2020-2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* 2020 May 4 (pp. 1055-1059). IEEE.
- [64].Hasan SM, Linte CA. U-NetPlus: a modified encoder-decoder U-Net architecture for semantic and instance segmentation of surgical instrument. *arXiv preprint arXiv:1902.08994*. 2019 Feb 24.
- [65].Zhou Z, Siddiquee MM, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support 2018 Sep 20* (pp. 3-11). Springer, Cham.
- [66].E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, 1 April 2017, doi: 10.1109/TPAMI.2016.2572683.
- [67].Chen, L., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." *Computer Vision — ECCV 2018*, 833-851. Munic, Germany: ECCV, 2018.
- [68].Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), pp.2481-2495.
- [69].Jin PJ, Yang D, Ran B. Reducing the error accumulation in car-following models calibrated with vehicle trajectory data. in *IEEE Trans. on Intell. Trans. Syst.*. 2013 Aug 21;15(1):148-57.
- [70].[https://github.com/TeRyZh/Reconstruction-NGSIM-Trajectory-with-DMD-and-Res\\_Unet\\_plus](https://github.com/TeRyZh/Reconstruction-NGSIM-Trajectory-with-DMD-and-Res_Unet_plus)



**Tianya Zhang** received BS degree from Beijing Institute of Technology, China, and earned master's degree from Texas A&M University, College Station, Texas, US. Currently, he is a Ph.D. candidate in the Department of Civil and Environmental Engineering from Rutgers University.

He has been involved in several ITS projects related to Automated Traffic Signal Performance Measures (ATSPMs), Connected and Automated Vehicle (CAVs) and Smart Mobility Testing Ground. His research using computer vision and LiDAR for vehicle trajectory detection has been published in *Transportation Research Part C* and *Transportation Research Record*.



**Peter J. Jin** was an Associate Professor at Department of Civil and Environmental Engineering (CEE) at Rutgers, The State University of New Jersey. He received the BS degree in Automation at Tsinghua University, China. He received his MS and Ph.D. degrees in civil engineering from University of Wisconsin-Madison in 2007 and 2009 respectively. He worked at Center for Transportation Research, at the University of Texas at Austin as a postdoctoral fellow and research associate. He has more than 45 peer-reviewed journal publications and more than 70 conference papers. His research interests include transportation big data analytics, intelligent transportation systems, connected and automated vehicles (CAVs), and unmanned aerial vehicles (UAVs). He holds two patents in both UAVs and CAVs.