Geoinformatica (2016) 20:59–94
DOI 10.1007/s10707-015-0231-0

Spatio-temporal traffic video data archiving and retrieval system

Hang Yue[1] · **Laurence R. Rilett**[2,3] · **Peter Z. Revesz**[4]

Received: 24 September 2014 / Revised: 19 June 2015 /
Accepted: 23 July 2015 / Published online: 25 August 2015

Abstract This paper presents a transportation spatio-temporal system that efficiently converts traffic video data into vehicular motion information in spatio-temporal databases for a variety of transportation applications. The proposed transportation spatio-temporal system interpolates the vehicle trajectory data (i.e., time, location, and speed), which are extracted from video, and integrates them with spatial road information for storage of dynamic transportation environments. The proposed transportation spatio-temporal system can mitigate data storage and retrieval issues related to storing large amounts of traffic video. Moreover, users can manage and operate multiform and multidimensional traffic data in a spatio-temporal transportation environment. The proposed approach is demonstrated for typical transportation applications. The experimental results show that the proposed transportation spatio-temporal system has excellent potential for addressing issues related to storage of large amounts of traffic video data.

Keywords GIS · Spatio-temporal database · Vehicular speed interpolation · Cubic-spline · Local polynomial regression · Traffic video

✉ Hang Yue
yuehang366@gmail.com

Laurence R. Rilett
lrilett2@unl.edu

Peter Z. Revesz
revesz@cse.unl.edu

[1] Johns Hopkins Healthcare LLC, Glen Burnie, MD 21060, USA

[2] Nebraska Transportation Center, Lincoln, NE, USA

[3] Civil Engineering Department, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

[4] Computer Science & Engineering Department, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

1 Introduction

Some earlier transportation-related applications (e.g., urban planning) required only static spatial databases which are typically referred to as *geographic information systems (GISs)*. For instance, Miller [1] used a GIS for the evaluation of *traffic analysis zone (TAZ)* effects, the design of optimal zoning systems, and the derivation of better zonal distance measures. Also, various *intelligent transportation systems (ITS)* often use static GIS map databases for location referencing and frequently exchange spatial information with other map databases [2]. However, traffic data collection technology has advanced faster than the technology used for transportation databases [3]. For example, typically discrete vehicle trajectory data are stored in flat-files or relational databases and roadway spatial data are stored in standard GIS platforms. In essence, the two types of data are stored separately, which results in a loss of information about the spatial relationships between the moving vehicles and the roadways. The spatio-temporal relationships between the moving vehicles cannot be readily identified. A number of critical transportation applications need to consider explicitly traffic parameters that vary continuously over time. In these situations spatial database systems can be used, but they tend to deal with temporal data sets in an inefficient way through the use of discrete time points or intervals.

Today, video cameras are widely used for traffic monitoring and data collection. The combination of space and time information is a defining feature of digital video [4]. However, storing traffic video data requires large storage space, which can be quite expensive. Accordingly, traffic video data are saved into video segments, scenes, shots, or frames [5] in order to reduce storage costs. Also, it is very difficult to extract key spatio-temporal data, such as individual vehicle trajectories and traffic aggregate data, from traditional video storage media [6, 7]. Ideally, this would be done automatically, but this information is typically obtained using manual methods. Consequently, it is difficult for current video database systems to automatically scan traffic video data and identify important transportation-related spatio-temporal information. For example, a user may wish to know vehicle trajectories as a function of key events (e.g., railway gate activity at a highway-railway at-grade crossing). Such data queries are problematic for current video databases.

Many transportation applications require spatio-temporal databases [8, 9] for the data storage and retrieval of large sets of moving objects. Current spatio-temporal databases can integrate dynamic temporal effects with a description of spatial dimensions [10, 11]. However, in terms of tracking moving vehicles over a road network, current studies [12, 13] have focused only on translating discrete GPS points to a particular road segment. The use of vehicular data, which are extracted from traffic video, in standard spatio-temporal databases has not been explored in previous research studies.

2 Motivation and purpose

Given traffic data's multiform and multidimensional nature, it is hypothesized that a more efficient traffic data archiving approach is required to adequately address the temporal dimension for GIS-based transportation management systems. Also, it is hypothesized that by using a combined spatio-temporal database transportation engineers will be able to readily access and query multiform and multidimensional traffic data. A variety of traffic parameter values including time period, lane, and vehicle type can be retrieved from this type of database.

Moreover, information on individual vehicles, which are often required for driver behavior analyses, can be obtained.

This paper clarifies two main ideas for archiving video data. First, discrete vehicle trajectory data (i.e., video at some predefined number of frames per second) from video cameras are extracted from the video at set time periods and subsequently interpolated into "continuous" vehicle trajectory information. This translation from discrete to continuous is accomplished through a data model. This data model can be used to define the movement of vehicles as a function of time (t) and space (x, y) parameters. Second, the resulting continuous traffic data can be stored readily in existing spatio-temporal databases. The focus of this paper is the use of constraint databases, because constraint databases can describe continuous spatio-temporal data in user-defined high-dimensions. In addition, they allow various high-level query languages, such as SQL and Datalog, to be utilized. The end result is that the continuous trajectory information can be used to identify key traffic parameters of interest, including travel time over user-defined space and time intervals, space-mean speed, time-mean speed, volume, and density. These parameters are calculated from the stored vehicle trajectories and are not necessarily stored in spatio-temporal databases. Intuitively, this information retrieval approach will be more robust. For example, different users may wish to identify travel time at different spatial time frames (e.g., 5 min average, 60 min average) or density at different increments (e.g., vehs/200 ft or vehs/1000 ft) and this would be readily accomplished.

The aim of this paper is to describe the development and features of an efficient transportation spatio-temporal system. The proposed transportation spatio-temporal system can convert traffic video data into transportation spatio-temporal databases. The system allows the user to choose various data interpolation options. In addition, not only standard SQL queries, but also high-level queries can be specially designed and conducted for transportation applications.

3 Overview of the spatio-temporal traffic video data archiving and retrieval system

The overall design of spatio-temporal traffic video data archiving and retrieval system is illustrated in Fig. 1. It may be seen that the design and development of the transportation spatio-temporal system consists of the following five main parts:

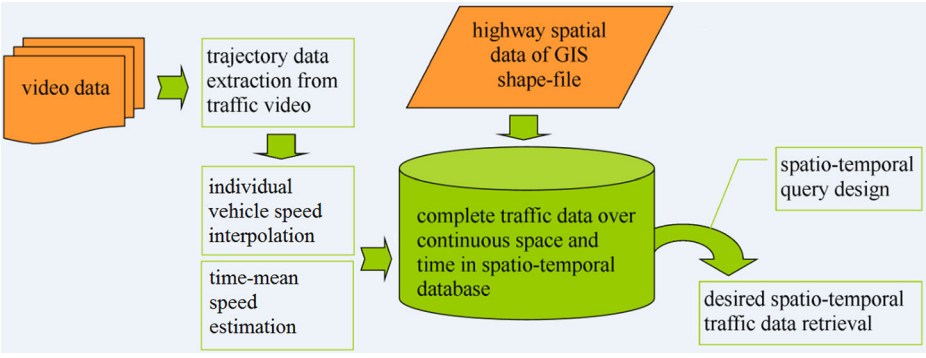

Fig. 1 Overview of the spatio-temporal traffic video archiving and retrieval system

Step 1 Vehicle Trajectory Data Extraction

Using state-of-the-art video-capture methods [14, 15], the traffic video data extraction step obtains vehicle trajectory data (i.e., instantaneous location, time, and speed data points for each individual vehicle) at discrete points of time for each vehicle in the video.

Step 2 Individual Vehicle Speed Interpolation

Linear and nonlinear data models are developed to translate the discrete speed values of each vehicle into continuous speed functions (i.e., continuous vehicle trajectory).

Step 3 Time-Mean Speed Estimation

An advanced statistical methodology is applied to estimate the continuous time-mean speed based on the discrete vehicle speeds. The discrete vehicle speeds are collected at a detector station on the highway segment while vehicles travel over a detector station.

Step 4 Data Transformation and Integration

Due to the use of linear constraint databases, continuous nonlinear data models, including individual vehicle data models and time-mean speed models, are required to transfer into the transportation spato-temporal system as a piecewise-linear form. Also, the highway spatial data of GIS shapefiles and the continuous vehicle trajectory data from the above individual vehicle data interpolation step are integrated into a spatio-temporal database.

Step 5 Traffic Information Retrieval

A high level traffic information query interface guides the users in performing spatio-temporal queries of the integrated dynamic transportation information.

4 Step 1 – Vehicle trajectory data extraction

In this step, vehicle trajectory data (i.e., instantaneous location, time, and speed data points for each individual vehicle) at discrete points of time are obtained from the video.

4.1 Basic steps

Because high resolution cameras, good quality video-capture cards, and advanced video-capture-based approaches are increasingly becoming available to transportation agencies, it has become cost-effective to extract accurate multiple-vehicle trajectory data from video. For instance, the advanced machine vision system used in the *Next Generation Simulation (NGSIM)* program [14] automatically extracted vehicle trajectories from highway traffic video data. The machine vision algorithms [16, 17] used for vehicle detection and tracking were developed to obtain a comprehensive dataset populated with individual vehicle trajectory at a rate of 10 frames per second. The software *Vehicle Video-Capture Data Collector (VEVID)* [15] can digitize full-motion video at an even higher frame rate of up to 30 frames per second.

Wei et al. [15] described a general approach for extracting vehicle trajectory data from video for traffic modeling. These research projects [14, 15] developed methods for extracting vehicle trajectory data over small time intervals (e.g., 0.1 s) from video for various traffic applications. Other research studies [18, 19] used vehicle trajectory data extracted from video

to calibrate transportation microscopic simulation, such as lane changing models, lane-choice models, car-following models, and lane-vehicle-allocation models.

Figure 2 shows the five steps associated with traffic data collection and extraction using the video-capture method:

Step I: Identify reference points along the roadway of interest, and measure and record the distances between reference points

Step II: Set up a camera in a position above the roadway of interest, and collect and store video related to the traffic phenomena of interest

Step III: Digitize critical video segments using *Audio Video Interleave (AVI)* or Video for Windows with a user-specified frame rate

Step IV: Input the AVI file and the distance information about the reference points into advanced machine vision software, such as VEVID, which can automatically extract the vehicle trajectory data

Step V: Store the vehicle trajectory data in the database

4.2 Quality control approach

In order to obtain each individual vehicle trajectory with a sufficient clarity, the NGSIM-VIDEO program [14, 20] requires high-resolution video images. The ideal resolution is 3 pixels per foot or higher, and the resolution must be at least 1 pixel per foot. Considering unstable situations during data collection (e.g., camera movement or wind disturbance), video stabilization is a pre-processing effort for data extraction. Video stabilization software [20] is used to crop of the image area or automatically select certain features with a zoom-in at a fixed location in successive video images. The data collector can determine the appropriate roadway segment in frame, and ensure a consistent view of the roadway area for data extraction over the whole time period.

Prior to the actual video extraction, video images are rectified on the basis of camera intrinsic parameters, boundary coordinates, rotation and translation matrices, and OpenCV (a software system of computer vision) functions. Compressed with the Xvid Codec (a software system of video compression) and a standard image compression rate (8000 Kb), the resulting videos are MPEG-4 standard AVI files [20]. Once the configuration file of the resulting videos is loaded and the database for archiving the vehicular trajectory information is connected, NGSIM-VIDEO would launch the vehicle trajectory data extraction. The configuration file in XML contains the AVI file names and locations, the camera's intrinsic parameters and transformation matrices, the direction polygon with directional points, and other input parameters.

Fig. 2 Traffic data collection and extraction using video-capture techniques

Some modes concerning the vehicle trajectory extraction are developed to support automatic detection and correction of trajectory data. These modes include visualization display, parameter adjustment, reverse, edit, truncate, and alert. The visualization display mode of the automatic tracking process can create a visible trajectory line that trails each vehicle and displays a box that defines a vehicle boundary for each vehicle and a vehicle ID inside each track box. Also, the camera boundary zones where vehicles are moving from one camera to the other can be outlined as occlusion zones. The parameter adjustment mode allows users to change the parameters to improve the detection and tracking quality of the vehicle trajectory.

When the distance covered by the study area is long, the task of tracking vehicles is usually divided into a forward tracking and a reverse tracking part. A reverse mode can reverse the tracking video images and make the vehicle appear to travel backward. If the user wants to change the direction polygon or is not satisfied with the vehicle length or width of the tracking box, then the edit mode is an alternative way to change those parameters. Moreover, the user can completely remove the trajectory of a vehicle from databases and re-track a particular vehicle in a certain image frame. The creation of an alert can direct users to manually correct trajectories, and the manual process can also track a vehicle that is not detected automatically. Hence, the above automatic and manual tracking processes of data extraction can guarantee the tracking accuracy of each vehicle in a clean way and avoid data extraction errors from missing detection and occlusion zones in video images with heavy traffic.

5 Step 2 – Individual vehicle speed interpolation

In this step, a methodology is developed to translate the discrete speed values for each individual vehicle obtained from the video into continuous speed functions. In other words, this step estimates a continuous vehicle trajectory for each vehicle obtained from the traffic video.

5.1 Proposed methodology

As discussed above, the first step is to identify discrete space and time points from each of the vehicles in the traffic video. The next step is to convert each vehicle's discrete trajectory information into a "continuous" estimate so that the speed, at any point in time and space, may be estimated. In this paper this is done using cubic-spline models where the data between the "missing" points are interpolated.

The use of cubic-splines for data interpolation has been used in a wide variety of applications [21–25]. The common function of cubic-spline $S(x)$ in [26] is:

$$S(x) = \begin{cases} s_1(x) & if \quad x_1 \le x \le x_2 \\ s_2(x) & if \quad x_2 \le x \le x_3 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ s_{n-1}(x) & if \quad x_{n-1} \le x \le x_n \end{cases} \tag{1}$$

where S_i is defined as a third degree polynomial below in Eq. (2):

$$s_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i \tag{2}$$

Where:

i=1, 2,…, n-1;
x_i is the interval value; and
a_i, b_i, c_i, and d_i are the coefficients in the i^{th} piece (i.e., the weights of interpolating known data).

The first and second derivatives of these n–1 equations ($1 \le i \le n-1$) are fundamental to the process, and they are:

$$s_i'(x) = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i \tag{3}$$

$$s_i''(x) = 6a_i(x-x_i) + 2b_i \tag{4}$$

The curve $S(x)$, the first derivative $S'(x)$, and the second derivative $S''(x)$ must be continuous across its entire interval $[x_1, x_n]$, and each sub-function must join at the data knots for $2 \le i \le n-1$:

$$S_i(x_i) = S_{i-1}(x_i) \quad S_i'(x_i) = S_{i-1}'(x_i) \quad S_i''(x_i) = S_{i-1}''(x_i)$$
$$h = x_i - x_{i-1}$$

The piecewise function $S(x)$ interpolates all discrete data points, $S(x_i) = y_i$ for $1 \le i \le n-1$ and $s_i(x_i) = y_i$ in every interval. When substituting $M_i = S_i''(x_i)$ and h into the above derivations, the results ($1 \le i \le n-1$) are concluded below:

$$a_i = \frac{M_{i+1} - M_i}{6h}$$
$$b_i = \frac{M_i}{2}$$
$$c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right)h$$
$$d_i = y_i$$

In general, there are four types of cubic-splines: exact-slope spline, natural spline, zero-slope spline, and not-a-knot spline. Given the slopes in x_1 and x_n are known, i.e., $s_1'(x_1)=k_1$ and $s_n'(x_n)=k_2$, the exact-slope spline is an optional approach to interpolate data. The not-a-knot spline does not specify any extra conditions at the end points, and this method requires that the third derivative of the spline $S'''(x)$ is continuous at x_2 and x_{n-1}. The natural spline has the known condition, i.e., $s_1''(x_1)=s_n''(x_n)=0$. Lastly, the zero-slope spline has zero slopes in x_1 and x_n, i.e., $s_1'(x_1)=s_n'(x_n)=0$.

5.2 Previous research on vehicle trajectory data

Many approaches have been developed for vehicle trajectory and movement analysis. Most trajectory analyses focus on data pattern visualization on map or location data aggregation in

grid cells or clusters by using a density or location-based method. For example, Guo et al. [27] developed a spatially constrained graph partitioning approach to establish topological relationships among trajectories. Liu et al. [28] interpolated, integrated, and calibrated GPS location data with a digital road network to identify vehicle traveling paths. In order to reduce the complexity of each trajectory route, the Douglas-Peucker algorithm [29] was created to simplify or generalize trajectory by removing points while retaining the general shape [30]. Lee et al. [31] partitioned each trajectory into sub-trajectories, and then classified or clustered vehicle trajectories using a density-based method. Also, Rinzivillo et al. [32] utilized different similarity measures at different cluster levels to progressively discover patterns.

Instead of simplifying, characterizing, grouping, or comparing trajectories for the extraction of data patterns, the proposed method interpolates each vehicle speed by using the piecewise-linear model and cubic-splines. The research problem is how to create continuous trajectory data to achieve a dynamic transportation information environment for traffic data archives and retrievals. Gindele et al. [33] used the Bézier curve method to obtain some intermediate values of vehicle trajectory for the driver behavior estimation. However, the Bézier curve fitting is a data approximation method, not a data interpolation method. Using a data approximation method, the control points lie close to the curve. That is to say, unlike a data interpolation method, the curve does not usually pass through all control points. Although Egerstedt and Martin [34] developed some smoothing splines for trajectories [34], these trajectories are from air traffic. Hence the data interpolation of discrete vehicle trajectory has not been well studied.

5.3 U.S. 101 example

U.S. 101, known as the Hollywood Freeway in Los Angeles CA, was chosen as the test bed for this study. This highway was part of the NGSIM program [14] and the discrete vehicle trajectory data are readily available. GIS shapefiles, which contain important spatial information and geometry features, were obtained from the NGSIM site. In addition, discrete vehicle trajectory data extracted from video, differentiated by time, location, speed, vehicle class, lane identification, etc. were available.

The vehicle trajectory data was collected by video cameras on June 15, 2005, and two data sets were created from the NGSIM data. The first, known as *Data Set 1 (D1)*, consists of the vehicles that were active during the 15-minute period between 7:50 and 8:05 am. There were 1993 vehicles and their associated 1,048,576 speed points in this data set. The second, known as *Data Set 2 (D2)*, consists of vehicles that were active during the 15-minute period between 8:05 and 8:20 am. There were 2017 vehicles and their associated 1,403,094 speed points in this data set. The combined data set has a total of 4010 separated vehicles, with an associated 2,451,670 speed points, and is known as *Data Set 3 (D3)* in this paper.

5.4 Experimental analysis

To measure the accuracy of the proposed approach, the *Root Mean Square Error (RMSE)* metric and the *Median Absolute Deviation* (MAD) metric, as shown in Eqs. (5) and (6) respectively, are used in this paper. It may be seen that the RMSE for the j^{th} vehicle captures the squared error between the observed and estimated

values for each discrete point along the trajectory. The smaller the RMSE or MAD value, the more accurate the data interpolation technique.

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{N_j}\left(Y_{ij}-\hat{Y}_{ij}\right)^2}{N_j}} \quad \forall\, j = 1\ to\ J \tag{5}$$

Where:

$RMSE_j$ is the Root Mean Square Error for the j^{th} vehicle trajectory
J is the number of vehicle trajectories being studied
N_j is the number of discrete speed data points for the j^{th} vehicle trajectory
Y_{ij} is the observed speed data points for the i^{th} speed data point of the j^{th} vehicle trajectory; and
$\hat{Y}_{ij}$ is the estimated speed value obtained from the linear or cubic-spline interpolation model for the i^{th} speed data point of the j^{th} vehicle trajectory.

$$MAD_j = Median\left(\left|\left(Y_{ij}-\hat{Y}_{ij}\right)-Meidan\left(Y_{kj}-\hat{Y}_{kj}\right)\right|\right) \quad \forall\, i = 1\ to\ N_j,\ k = 1\ to\ N_j,\ j = 1\ to\ J \tag{6}$$

Where:

MAD_j is the Median Absolute Deviation for the j^{th} vehicle trajectory
J is the number of vehicle trajectories being studied
N_j is the number of discrete speed data points for the j^{th} vehicle trajectory
Y_{ij} is the observed speed data points for the i^{th} speed data point of the j^{th} vehicle trajectory; and
$\hat{Y}_{ij}$ is the estimated speed value obtained from the linear or cubic-spline interpolation model for the i^{th} speed data point of the j^{th} vehicle trajectory.

Figure 3 illustrates this approach for a single vehicle trajectory. The individual observed data points (Y_i) are shown by blue stars and blue dots. The user identifies the time interval for interpolation. For this example, the time interval is 0.2 s and consequently only the data points indicated by the blue stars are used for data interpolation. The data points on the interpolated curve ($\hat{Y}_i$) are shown by orange stars and blue stars.

Equations (7) and (8) show the overall RMSE and MAD average for all vehicles in the first data set (i.e., J=1993) respectively. The average RMSE results for each of the four data

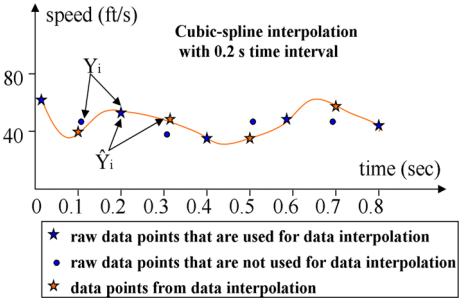

Fig. 3 Cubic-spline interpolation

interpolation models of individual vehicular speed as a function of time interval are shown in Table 1. Note that the exact-slope spline interpolation results are not shown. This is because this approach requires acceleration values (i.e., the slopes of vehicular speed data) and these are unknown. Note that the required accelerations may be estimated using the speed and time values.

$$\overline{RMSE} = \frac{\sum_{j=1}^{J} RMSE_j}{J} \tag{7}$$

Where:

$RMSE_j$ is the Root Mean Square Error for the jth vehicle trajectory
$\overline{RMSE}$ is the average Root Mean Square Error across all vehicles; and
J is the number of vehicle trajectories being studied.

$$\overline{MAD} = \frac{\sum_{j=1}^{J} MAD_j}{J} \tag{8}$$

Where:

MAD_j is the Median Absolute Deviation for the j^{th} vehicle trajectory
$\overline{MAD}$ is the average Median Absolute Deviation across all vehicles; and
J is the number of vehicle trajectories being studied.

The time interval among interpolated data set has a significant influence on the interpolation accuracy. Table 1 shows the results of the average RMSE for a similar analysis utilizing D1and D2, and Table 2 shows the results of the average MAD for a similar analysis utilizing D1 and D2. It can be seen that as the time interval increases, so too do both of the average RMSE and MAD. The results are expected because a greater time interval for interpolation implies a greater loss in information. It was found that all the approaches were more accurate for the

Table 1 Interpolation model estimation using RMSE

Time interval	Piecewise-linear (ft/s)			Not-a-knot (ft/s)			Natural (ft/s)			Zero-slope (ft/s)		
	D1	D2	D2−D1	D1	D2	D2−D1	D1	D2	D2−D1	D1	D2	D2−D1
t=0.2 s	0.21	0.15	−0.06	0.16	0.11	−0.05	0.16	0.11	−0.05	0.16	0.11	−0.05
t=0.3 s	0.30	0.24	−0.06	0.23	0.18	−0.05	0.23	0.18	−0.05	0.23	0.18	−0.05
t=0.4 s	0.43	0.33	−0.10	0.37	0.29	−0.08	0.37	0.29	−0.08	0.37	0.29	−0.08
t=0.5 s	0.55	0.44	−0.11	0.50	0.41	−0.09	0.50	0.40	−0.10	0.50	0.40	−0.10
t=0.6 s	0.71	0.56	−0.15	0.69	0.54	−0.15	0.68	0.54	−0.14	0.68	0.54	−0.14
t=0.7 s	0.85	0.66	−0.19	0.84	0.66	−0.18	0.83	0.66	−0.17	0.83	0.65	−0.18
t=0.8 s	0.98	0.75	−0.23	0.99	0.76	−0.23	0.98	0.76	−0.22	0.98	0.76	−0.22
t=0.9 s	1.09	0.84	−0.25	1.13	0.86	−0.27	1.11	0.86	−0.25	1.11	0.85	−0.26
t=1.0 s	1.19	0.92	−0.27	1.25	0.95	−0.30	1.23	0.94	−0.29	1.23	0.94	−0.29

1.0 ft/s=0.3048 m/s

Table 2 Interpolation model estimation using MAD

Time interval	Piecewise-linear (ft/s)			Not-a-knot (ft/s)			Natural (ft/s)			Zero-slope (ft/s)		
	D1	D2	D2−D1	D1	D2	D2−D1	D1	D2	D2−D1	D1	D2	D2−D1
t=0.2 s	0.03	0.02	−0.01	0.03	0.01	−0.02	0.03	0.01	−0.02	0.03	0.01	−0.02
t=0.3 s	0.06	0.03	−0.03	0.06	0.04	−0.02	0.06	0.04	−0.02	0.06	0.04	−0.02
t=0.4 s	0.12	0.06	−0.06	0.12	0.08	−0.04	0.12	0.08	−0.04	0.12	0.08	−0.04
t=0.5 s	0.16	0.09	−0.07	0.17	0.13	−0.04	0.17	0.13	−0.04	0.17	0.12	−0.05
t=0.6 s	0.23	0.15	−0.08	0.25	0.19	−0.06	0.24	0.18	−0.06	0.24	0.18	−0.06
t=0.7 s	0.29	0.19	−0.10	0.31	0.23	−0.08	0.30	0.23	−0.07	0.30	0.23	−0.07
t=0.8 s	0.34	0.23	−0.11	0.37	0.27	−0.10	0.36	0.27	−0.09	0.36	0.27	−0.09
t=0.9 s	0.39	0.27	−0.12	0.42	0.31	−0.11	0.42	0.31	−0.11	0.41	0.31	−0.10
t=1.0 s	0.43	0.31	−0.12	0.47	0.36	−0.11	0.46	0.35	−0.11	0.46	0.35	−0.11

1.0 ft/s=0.3048 m/s

trajectories in dataset D2 than dataset D1 as measured by the difference of the average RMSE values between D1 and D2 (i.e., D2−D1) in Table 1 and the difference of the average MAD values between D1 and D2 (i.e., D2−D1) in Table 2. For each interpolation model estimation with RMSE, the difference in accuracy (e.g., |D2−D1|) increases as the time interval rises when the time frame is larger than 0.2. However, for each interpolation model estimation with MAD, the difference in accuracy (e.g., |D2−D1|) may keep the same value as the time interval rises.

In terms of the interpolation model estimation with RMSE, it may be seen that the "best" interpolation method depends on the time interval. For time units of up to 0.7 s the zero-slope and natural approaches are the best. The "not-a-knot" approach is third, and the piecewise-linear approach is the worst. For small temporal intervals, the more complicated interpolation approaches work best. When the time frame is larger than 0.7 s, the piece-wise linear function is best or one of the best approaches. For this situation the simpler model performs very well. Given the same speed distribution profile, it is hypothesized that when the temporal interval becomes too large (e.g., greater than 0.7 s), the loss of information is such that the advantages of the more complex model are lost. In this case a simpler model would be preferred.

However, the RMSE has the high influence of outliers in data on performance evaluation, and the presence of outliers does not change the value of the MAD [35–38]. Due to the resistance to the outliers of the vehicular trajectory data, the interpolation model estimation with MAD in Table 2 shows that the piece-wise linear function is best or one of the best approaches, when the time frame is larger than 0.2 s. The zero-slope approach is second or one of the second best approaches, when the time frame is larger than 0.4 s. In addition, it has been found in previous research that the linear interpolation method for 2 days spatial data from a real estate data set also works better than the more sophisticated 2D spatial interpolation methods [8].

Figure 4a and b show the histograms of the observed speed distribution between 7:50 and 8:05 am (D1 data set) and between 8:05 and 8:20 am (D2 data set), respectively. There are considerably different data characteristics (such as mean, standard deviation, 95 % confidence interval, percentage range, peak percentage, data size, and so on) between the results for datasets D1 and D2. It may be seen that the vehicles in data set D2 have much lower speeds and are in more congested conditions. Intuitively, it is more difficult to interpolate data when

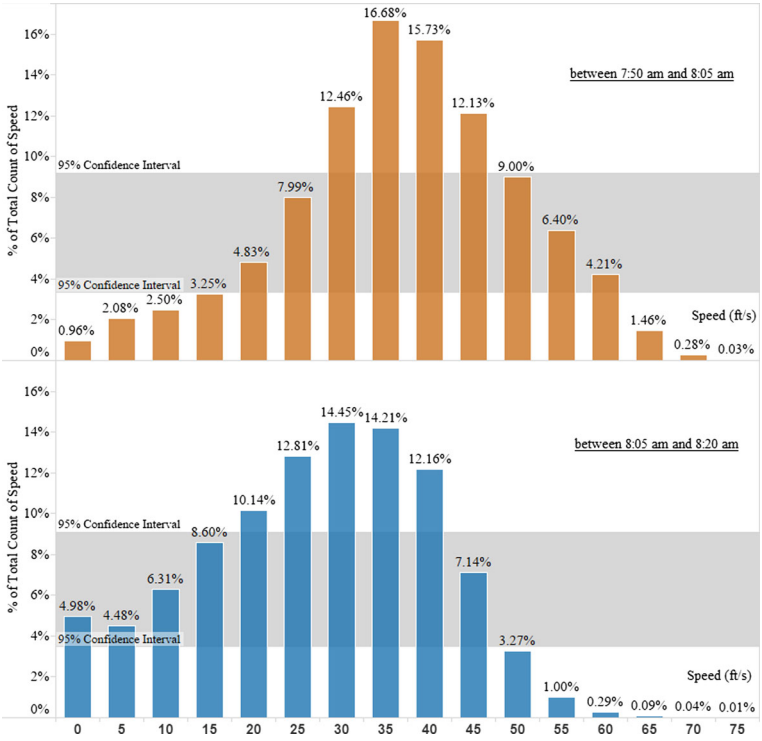

Fig. 4 **a** Speed profiles between 7:50 and 8:05 am, and **b** Speed profiles between 8:05 and 8:20 am

conditions are changing fast. In addition, the longer the aggregation interval, the more of a challenge it is to interpolate the data. The contrast in the histograms, as shown in Fig. 4, implies that the interpolation accuracy is related to the characteristics of a data set. It is hypothesized that the transition points would be site specific and would vary by time of day – that is, users would have to conduct preliminary analyses to identify the best approach for their particular application.

5.5 Sensitivity test

On the basis of the above data analysis, the zero-slope approach was identified as the best cubic-spline for individual vehicle speed interpolation. In order to test how robust the piecewise-linear and cubic-spline interpolation methods were in the face of data noise, the random function (*Randn*) in MATLAP was used to generate a scaled data noise for the piecewise-linear interpolation approach and the cubic-spline with zero-slope. Figure 5 illustrates 10 % scaled data noise on the speed of the vehicle (ID: 609) with a 1.0 s time interval.

Different percentages of data noise were added into the data set D3 by using Eq. (9) to make a cross table with three dimensions. The three dimensions for the average RMSE measure include time interval, percentage of data noise scaled, and interpolation method, as shown in Table 2.

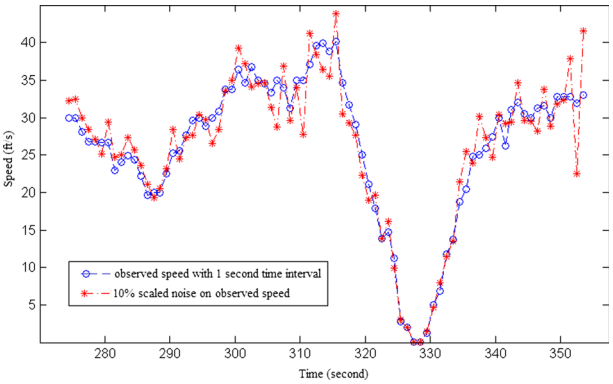

Fig. 5 Scaled data noise of individual vehicle speed

$$EY_{ij} = Y_{ij}\left(1 + ER_{ij}{*}EP\right) \tag{9}$$

Where:

Y_{ij} is the observed speed data points for the i^{th} speed data point of the j^{th} vehicle trajectory with a time interval t (t may be 0.2, 0.3, …, or 1.0 s)

ER_{ij} is the data noise that is normally distributed with mean 0 and variance 1 for the i^{th} speed data point of the j^{th} vehicle trajectory with a certain time interval

EY_{ij} is the adjusted speed value with data noise for the i^{th} speed data point of the j^{th} vehicle trajectory; and

EP is the percentage of data noise assigned into the entire data set.

In Table 3 it can be seen that the average RMSE and MAD increases with the percentage of data noise of each time interval. That result is expected because a greater percentage of data noise implies a greater loss in data accuracy. When 3 % of data noise is scaled into the data set, all average RMSE values are larger than 1.0 ft/s; when 6 % of data noise is scaled, all average RMSE values are larger than 2.0 ft/s; and when 9 % of data noise is scaled, all average RMSE values are larger than 3.0 ft/s. Similarly, when 3 % of data noise is scaled into the data set, all average MAD values are larger than 0.5 ft/s; when 5 % of data noise is scaled, all average MAD values are larger than 1.0 ft/s; and when 8 % of data noise is scaled, all average MAD values are larger than 1.5 ft/s. These results imply that the time interval increase has a detrimental influence on the average RMSE and MAD values with a small percentage of data noise scaled as compared to the larger percentage of data noise scaled. For example, the average RMSE value with 1 % percentage of data noise in the linear method is 0.42 and 1.2 ft/s for time intervals 0.2 and 1.0 s, respectively. In contrast, the average RMSE values for the 10 % data noise scenario were very similar at 3.48 and 3.54 ft/s for time periods 0.2 and 1.0 s, respectively.

In terms of vehicle speed with 1 % or more data noise, the average RMSE values of the zero slope method are not less than those of the piecewise-linear method. The only exception was the 1 % data noise scenario for time intervals 0.3, 0.4, and 0.5 s.

Table 3 Speed noise scaled with different time intervals and percentages using RMSE and MAD

Estimation Method		RMSE											MAD										
Time Interval	Method	Percentage											Percentage										
		0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
0.2 s	Line	0.2	0.42	0.73	1.07	1.41	1.75	2.1	2.44	2.78	3.13	3.48	0.03	0.24	0.43	0.63	0.83	1.03	1.23	1.43	1.63	1.83	2.04
	Zero slope	0.15	0.44	0.8	1.17	1.55	1.93	2.31	2.69	3.06	3.45	3.83	0.03	0.25	0.47	0.7	0.93	1.16	1.39	1.62	1.84	2.07	2.31
Difference		0.05	-0.02	-0.07	-0.1	-0.14	-0.18	-0.21	-0.25	-0.28	-0.32	-0.35	0	-0.01	-0.04	-0.07	-0.1	-0.13	-0.16	-0.19	-0.21	-0.24	-0.27
0.3 s	Line	0.29	0.46	0.75	1.07	1.4	1.73	2.06	2.4	2.73	3.06	3.41	0.06	0.26	0.44	0.64	0.83	1.02	1.22	1.42	1.62	1.82	2.02
	Zero slope	0.22	0.45	0.8	1.17	1.55	1.92	2.3	2.68	3.06	3.43	3.82	0.05	0.26	0.48	0.71	0.94	1.16	1.39	1.62	1.85	2.08	2.3
Difference		0.07	0.01	-0.05	-0.1	-0.15	-0.19	-0.24	-0.28	-0.33	-0.37	-0.41	0.01	0	-0.04	-0.07	-0.11	-0.14	-0.17	-0.2	-0.23	-0.26	-0.28
0.4 s	Line	0.41	0.54	0.8	1.1	1.42	1.74	2.06	2.4	2.73	3.07	3.39	0.11	0.29	0.47	0.66	0.85	1.04	1.23	1.43	1.63	1.83	2.02
	Zero slope	0.35	0.53	0.85	1.2	1.57	1.94	2.32	2.69	3.07	3.46	3.83	0.11	0.3	0.51	0.73	0.95	1.18	1.4	1.63	1.86	2.09	2.31
Difference		0.06	0.01	-0.05	-0.1	-0.15	-0.2	-0.26	-0.29	-0.34	-0.39	-0.44	0	-0.01	-0.04	-0.07	-0.1	-0.14	-0.17	-0.2	-0.23	-0.26	-0.29
0.5 s	Line	0.53	0.64	0.87	1.15	1.46	1.77	2.1	2.42	2.75	3.08	3.41	0.15	0.33	0.51	0.69	0.87	1.07	1.26	1.45	1.64	1.83	2.03
	Zero slope	0.48	0.62	0.91	1.25	1.61	1.97	2.35	2.71	3.09	3.47	3.84	0.16	0.34	0.54	0.76	0.97	1.2	1.42	1.64	1.87	2.09	2.32
Difference		0.05	0.02	-0.04	-0.1	-0.15	-0.2	-0.25	-0.29	-0.34	-0.39	-0.43	-0.01	-0.01	-0.03	-0.07	-0.1	-0.13	-0.16	-0.19	-0.23	-0.26	-0.29
0.6 s	Line	0.69	0.77	0.97	1.23	1.52	1.82	2.13	2.45	2.77	3.1	3.42	0.22	0.37	0.55	0.73	0.91	1.1	1.28	1.47	1.66	1.85	2.05
	Zero slope	0.66	0.77	1.02	1.33	1.67	2.02	2.38	2.75	3.12	3.49	3.87	0.23	0.39	0.59	0.8	1.01	1.23	1.45	1.67	1.87	2.12	2.34
Difference		0.03	0	-0.05	-0.1	-0.15	-0.2	-0.25	-0.3	-0.35	-0.39	-0.45	-0.01	-0.02	-0.04	-0.07	-0.1	-0.13	-0.17	-0.2	-0.21	-0.27	-0.29
0.7 s	Line	0.82	0.89	1.07	1.31	1.58	1.87	2.18	2.48	2.8	3.13	3.45	0.27	0.41	0.59	0.76	0.94	1.12	1.32	1.49	1.68	1.88	2.07
	Zero slope	0.8	0.9	1.12	1.41	1.73	2.08	2.44	2.79	3.16	3.53	3.9	0.29	0.43	0.63	0.83	1.04	1.26	1.48	1.69	1.91	2.15	2.36
Difference		0.02	-0.01	-0.05	-0.1	-0.15	-0.21	-0.26	-0.31	-0.36	-0.4	-0.45	-0.02	-0.02	-0.04	-0.07	-0.1	-0.14	-0.16	-0.2	-0.23	-0.27	-0.29
0.8 s	Line	0.94	1	1.17	1.39	1.65	1.93	2.23	2.54	2.84	3.16	3.47	0.32	0.45	0.62	0.8	0.97	1.15	1.34	1.52	1.71	1.9	2.08
	Zero slope	0.94	1.02	1.23	1.5	1.81	2.14	2.48	2.84	3.19	3.56	3.92	0.34	0.47	0.67	0.87	1.08	1.29	1.5	1.72	1.94	2.16	2.38
Difference		0	-0.02	-0.06	-0.11	-0.16	-0.21	-0.25	-0.3	-0.35	-0.4	-0.45	-0.02	-0.02	-0.05	-0.07	-0.11	-0.14	-0.16	-0.2	-0.23	-0.26	-0.3
0.9 s	Line	1.05	1.11	1.26	1.47	1.73	1.99	2.28	2.58	2.88	3.2	3.52	0.37	0.48	0.66	0.83	1.01	1.19	1.37	1.55	1.74	1.93	2.11
	Zero slope	1.07	1.14	1.33	1.58	1.89	2.2	2.54	2.89	3.24	3.61	3.97	0.39	0.52	0.71	0.91	1.12	1.33	1.53	1.75	1.97	2.19	2.41
Difference		-0.02	-0.03	-0.07	-0.11	-0.16	-0.21	-0.26	-0.31	-0.36	-0.41	-0.45	-0.02	-0.04	-0.05	-0.08	-0.11	-0.14	-0.16	-0.2	-0.23	-0.26	-0.3
1.0 s	Line	1.14	1.2	1.34	1.54	1.78	2.05	2.33	2.62	2.92	3.22	3.54	0.41	0.51	0.69	0.86	1.04	1.22	1.39	1.57	1.76	1.93	2.12
	Zero slope	1.18	1.24	1.42	1.66	1.95	2.27	2.59	2.93	3.28	3.62	3.99	0.44	0.55	0.75	0.95	1.15	1.36	1.56	1.77	1.99	2.2	2.42
Difference		-0.04	-0.04	-0.08	-0.12	-0.17	-0.22	-0.26	-0.31	-0.36	-0.4	-0.45	-0.03	-0.04	-0.06	-0.09	-0.11	-0.14	-0.17	-0.2	-0.23	-0.27	-0.3

None of the average MAD values of the zero slope method are less than those of the piecewise-linear method. The reason may be that the cubic-spline with a complicated formula is more sensitive to noise than the simpler linear model. Based on this sensitivity analysis, the piecewise-linear model for individual vehicle speed interpolation was chosen as the preferred method.

6 Step 3 – Time-mean speed estimation

On the basis of the discrete vehicle speeds collected at a highway detector station, advanced statistical models are applied for the evaluation of the continuous time-mean speed. In addition, the comparison of the advanced statistical models is discussed in this step. Finally, this step innovatively approximates data curves as a piecewise-linear form with high accuracy using an algorithm.

6.1 Introduction

Any individual vehicle at any time has a certain (parked or moving) location and a certain (zero or non-zero) speed. This implies that time-mean speed, as an aggregate data of individual vehicle speed values, is continuous over time and space. However,

traditional time-mean speed [3, 39] is the arithmetical average of the speed data of all vehicles observed crossing a location along a roadway over a specified time period.

Due to the collection of discrete speed data and the use of a spatio-temporal database, a continuous time-mean speed model can be developed to better meet transportation data requirements. The continuous time-mean speed model is a more useful method of storing data than traditional discrete time-mean speed data storage approaches because the data can be readily accessed in a form that is advantageous to the data requirements of multiple users. Considering the use of linear constraint databases, the continuous time-mean speed model is required to transfer into the transportation spatio-temporal system as a piecewise-linear form for data archives.

6.2 Model choice analysis

Local polynomial regression without data pre-classification has several advantages. The local polynomial models were minimax efficient for both interior and boundary points and were optimal in the minimax sense [40]. Local polynomial regression has better performance near the boundary of data points than the traditional kernel regression methodologies, such as the Nadaraya-Watson estimator [41, 42] and the Gasser-Müller estimator [43]. The Nadaraya-Watson estimator produces an undesirable bias, and the Gasser-Müller estimator must pay a price in variance to manipulate a random design model. Further, local polynomial regression with high curvature adapts well to the bias problems at boundaries [44, 45]. The Nadaraya-Watson and Gasser-Müller estimators converged more slowly at the boundary [46], although the convergence rate of the estimators was the same for boundary points and interior points. Ruppert and Wand [47] showed that a multivariate case had a similar result. Cheng et al. [48] found that no linear estimator could beat local polynomial models on the boundary in a minimax sense, and no other estimator could make a significant improvement.

In contrast to local models, global models (such as neural networks and time series) typically require offline training because they do not solely rely on data pattern recognition [49]. The local modeling can avoid the negative interference exhibited by the global models. Moreover, the local linear method is preferable to the local constant regression in traffic data analysis [49, 50]. In regards to local constant regression, Smith et al. [51] and Faouzi [52], respectively, implemented the k-nearest neighbor method and kernel estimator in transportation.

In addition, Gaussian process was viewed as an infinite-dimensional generalization of the multivariate normal distribution [53], and this approach was used for data estimation [54–57]. Comparing Gaussian process with nonparametric regression methods, Yakowitz and Szidarovszky [58] found that nonparametric regression methods were more robust and reliable for the data analysis and error estimation when the data was not produced from an intrinsic random function with the right variogram. Gaussian process was marginally better only if the sample data met the intrinsic random function hypothesis and had the “true” variogram family. The above analysis gives the reasons why local polynomial regression models are selected for the time-mean speed estimation in this paper.

6.3 Definition

There is a basic difference between a parametric approach and a nonparametric approach. The former assumes a pre-specified functional form for the density estimator, while the latter does not. The density estimation in nonparametric regression can effectively describe the overall pattern in a set of data. Suppose that in a sample of random pairs $(x_1, y_1),\ldots, (x_n, y_n)$, the response variable y_i is assumed to satisfy [50]:

$$y_i = m(x_i) + \upsilon^{1/2}(x_i)\varepsilon_i \tag{10}$$

Where:

m is a function to be estimated
υ is a variance function
ε_i is an independent random variable with zero mean and unit variance
x_i is a random variable having common density f; and i =1,…, n.

A local polynomial estimator $\hat{m}(x; p, h)$ [59–61] can be developed via "locally" fitting a p^{th} degree polynomial $\sum_{j=0}^{p}\beta_j(x_i-x)^j$ to (x_i, y_i) using weighted least squares. Bandwidth h is assumed to approach zero at a rate slower than $n-1$, that is:

$$\lim_{n\to\infty} h = 0 \quad \lim_{n\to\infty} nh = \infty$$

The function of the local polynomial estimator for the true function Y is shown below:

$$\hat{Y} = \hat{m}(x, \mathrm{p}, \mathrm{h}) = \mathrm{e}_1^{\mathrm{T}}\left(\mathrm{X}_{\mathrm{x}}^{\mathrm{T}}\mathrm{W}_{\mathrm{x}}\,\mathrm{X}_{\mathrm{x}}\right)^{-1}\mathrm{X}_{\mathrm{x}}^{\mathrm{T}}\mathrm{W}_{\mathrm{x}}\mathrm{y} = \mathrm{e}_1^{\mathrm{T}}\,\hat{\beta} = \hat{\beta}_0 \tag{11}$$

Where:

e_1 is a $(p+1)\times 1$ vector having 1 in the first entry and zero elsewhere

$y=(y_1,\cdots,y_n)^T$ is a vector of responses

$W_x=diag\{K_h(x_1-x),\cdots,K_h(x_n-x)\}$ is an $n\times n$ diagonal matrix of weights

$$X_x = \begin{bmatrix} 1 & x_1-x & \cdots & (x_1-x)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n-x & \cdots & (x_n-x)^p \end{bmatrix}$$

is an $n\times(p+1)$ design matrix, n is the number of observations

$$\hat{\beta} = \left(\hat{\beta}_0, \cdots, \hat{\beta}_p\right)^T$$

is able to minimize the locally weighted polynomial regression $\sum_{i=1}^{n}\left\{y_i - \sum_{j=0}^{p}\beta_j(x_i-x)^j\right\}^2 K_h(x_i-x)$; and

$K_h(\cdot)=K(\cdot/h)/h$ is a kernel function scaled by h (the kernel function is usually a unimodal symmetric probability with $\int K(x)dx=1$).

Figure 6 displays the important aspects of local polynomial regression theory. Let Y (i.e., the green curve) represent the true model, and let $\hat{Y}$ (i.e., the red curve) represent a local

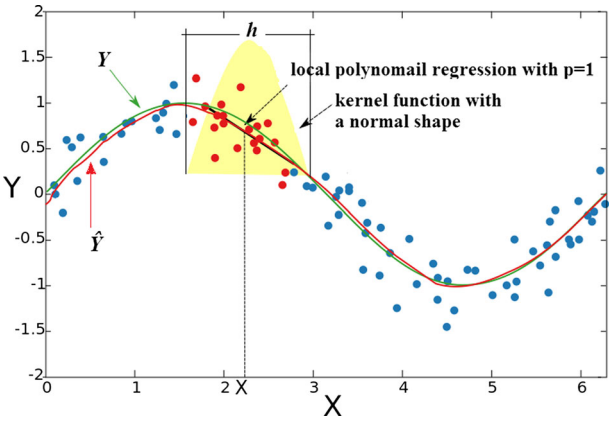

Fig. 6 Local polynomial regression model

polynomial regression line based on the observed points. The bandwidth h is a nonnegative number controlling the size of the local neighborhood. $K_h(x_i - x)$ is the weight assigned to y_i, and this weight depends on the height of the kernel function centered about the particular point x. The data closer to x carry more influence in the value of $m(x)$ for a general form of the regression function $m(x)$. There are some shape choices about kernel function [62], and these shapes may be Epanechnikov, Biweight, Triweight, Normal, Uniform, Triangular, etc.

6.4 Bandwidth selection

The bandwidth choice is particularly important to highlight the significant structure in a set of data. Jones et al. [63] executed a survey of several bandwidth selections for density estimation. The methods of bandwidth selections included the *Biased Cross-validation (BCV)* [64], *Least Squares Cross-validation (LSCV)* [65], *Rule-of-Thumb (ROT)*, *Solve-the-equation (STE)* [59, 66–69], and Smoothed Bootstrap [70]. Jones et al. [63] showed that the ROT had a small variance, but an unacceptably large mean; LSCV had a good mean, but too large a variance; BCV suffered from unstable performance; and that both STE and smoother bootstrap had a correctly centered distribution in mean and an acceptable variance.

Ruppert et al. [71] compared three plug-in bandwidth selection strategies [50], including ROT, STE, and *Direct Plug-in (DPI)* via data simulation and analysis. Also, they clarified the rules and calculation steps about these bandwidths. They found that both DPI and STE had acceptable performance. Most importantly, the DPI was less complicated because it required neither a root-finding procedure nor a minimization step. Based on previous research, the DPI approach was selected for calculating bandwidth in this paper.

6.5 Order choice

In terms of the order of polynomial fit for the asymptotic performance of $\hat{m}(\cdot; p, h)$, Fan et al. [50] showed that fitting polynomials of higher order led to a possible bias reduction and a

variance increase. They also showed that odd order fits were preferable to even order fits because they resulted in lower variance. Further, they identified that even order fits experienced a lower efficiency in terms of bias reduction, particularly in boundary regions and highly clustered design regions. In addition, they identified that higher order polynomial fits (i.e., greater than a cubic fit) required a very large sample to realize a significant improvement. Based on this previous research, this paper uses p=1 or p=3. A local cubic fit (when p=3) has more degrees of freedom for estimating a high curve region in a set of data than a local linear fit (when p=1), although a cubic fit has a higher requirement concerning its calculation and sample variability than a local linear fit does [62].

It is advantageous to keep both the local linear model and the local cubic model for data estimation. The local cubic model can provide an adequate fit to better capture sharp data curvatures. When the local cubic regression model tends to overfit the data or lack numerical stability for a given date set, the local linear model can be fit easily to the data. Moreover, the above DPI approach can find an appropriate bandwidth to control data overfitting, balance the variance and bias of a data set and minimize mean squared error.

6.6 U.S. 101 example

The values of vehicle speeds were collected at the detector station (717490) of the test bed (i.e., U.S. 101 Highway in Los Angeles, CA) from 4:30 to 6:30 pm on June 8, 2005 [14]. Figure 7 shows a schematic of the detector placement at a detector station located at a five-lane highway section used in this paper. Each station can record the time that each vehicle occupies the detector as the vehicle travels over it. The detector also counts the number of vehicles traveling over it. The occupancy and the flow rate of vehicles are then used along with the mean vehicle length to determine speed (speed = flow/occupancy/mean vehicle length). The data for occupancy, flow, and speed are aggregated and computed over a specified period of time and only the aggregated data are stored. The speed data at this station were averaged over five-minute periods. As shown in Fig. 8, the discrete speed data are estimated as a continuous time-mean speed curve using the local linear or cubic regression model.

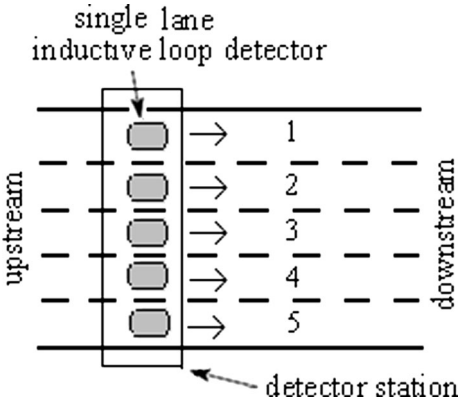

Fig. 7 Loop detector station

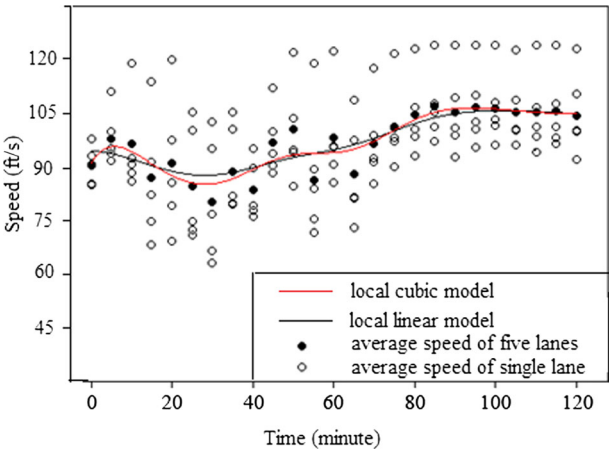

Fig. 8 Time-mean speed estimation using local polynomial models

6.7 Piecewise-linear approximation

An important innovation in this paper is the use of a piecewise linear approximation algorithm to approximate the data curves as a piecewise-linear form with high accuracy. The above continuous time-mean speed curves were divided into 361 data points. The latter discrete points were used as input for a piecewise-linear approximation [11], and these data points are referred as *speed-time pairs*. The piecewise-linear approximation compresses the discrete data points into a piecewise-linear function, which allows for data interpolation and faster queries. In the piecewise-linear approximation with data points (x_i, y_i), with $i=1,2,\ldots,n$, the relation between the piecewise-linear function $f(x_i)$ and y_i satisfies:

$$|f(x_i)-y_i|\leq\Psi \quad \text{for each}(x_i, y_i) \tag{12}$$

The maximum error threshold Ψ controls the maximum difference between the original data points and the piecewise-linear function. This means that the original data points are always within a narrow band with a width Ψ around the piecewise-linear function, as shown in Fig. 9.

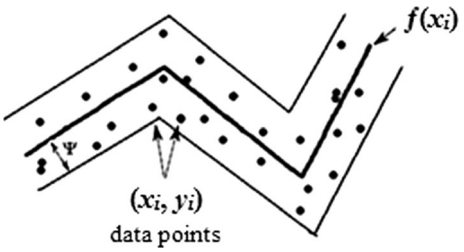

Fig. 9 Piecewise-linear approximation

6.8 Experimental analysis

The above speed data were used for the accuracy estimation of the local linear and cubic models. The bandwidth (h) was calculated by the above DPI approach (i.e., h=11.5). The *Mean Squared Error (MSE)*, RMSE, and *Mean Absolute Error (MAE)* [37] were used to measure the accuracy of the experimental data, and their definitions are listed below:

$$MSE = \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{n} \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{n}} \tag{14}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|}{n} \tag{15}$$

Where:

Y_i is the observed speed of the ith speed-time pair
$\hat{Y}_i$ is the estimated speed, calculated by the piecewise-linear sub-function for the ith speed-time pair; and
n is the number of the speed-time pairs (n=361 in this experiment data).

Table 4 provides the accuracy results of the linear and cubic models. It may be seen that the cubic model is more accurate than the linear model. Using the piecewise-linear algorithm with $\Psi = 0.05$, a previous study [72] transferred the above local linear and cubic speed curves in Fig. 9 into a piecewise-linear function with 14 sub-functions and a piecewise-linear function with 24 sub-functions, respectively. The RMSE was computed, using Eq. (14), for each sub-function, where n is the number of the speed-time pairs related to a certain sub-function, Y_i is the speed of the pairs, and $\hat{Y}_i$ is the speed calculated by the sub-function. The result of each sub-function RMSE was less than 0.04, indicating a very good model fit. The use of the local cubic model for data queries will be explored in more depth in Section 8 of this paper.

7 Step 4 – Data transformation and integration

As discussed previously, the focus of this paper is the transformation of continuous nonlinear traffic data models and the integration of highway spatial data and vehicle trajectory data. The

Table 4 Model estimation

Model	MSE	RMSE	MAE
Local linear model	0.0023	0.048	0.042
Local cubic model	0.0020	0.045	0.040

data transformation and integration consists of two parts: (1) the linear approximation of vehicular instantaneous speed and time-mean speed, and (2) the determination of instantaneous motion direction.

7.1 Speed linear approximation

By using the speed points per 0.1 s, the piecewise-linear approximation algorithm can automatically create the piecewise-linear functions. Note that if the time interval of the speed data is larger than 0.1 s, the data may be interpolated by the data interpolation methods in Section 5. A smaller error threshold (Ψ) in the piecewise-linear algorithm can produce more sub-functions for speed curve approximation with more accuracy. From the data set D3 in Section 5.3, the instantaneous speed data of five vehicles (vehicle ID is 2, 4, 5, 6, and 8) in 50 s are shown in Fig. 10. With the 50 s' speed data and the error threshold Ψ=0.05 as the input conditions, the piecewise-linear algorithm produces speed approximations as shown in Fig. 11. It may be seen that, visually at least, the approximation models closely follow the actual vehicle trajectories measured in the field.

Figures 12 and 13 show the average RMSE and MAD of the estimated sub-functions per vehicle in the entire data set D3. The average RMSE is the average Root Mean Square Error across all sub-functions, and the average MAD is the average Median Absolute Deviation across all sub-functions. The RMSE has the same format as Eqs. (5) and (7). Figure 12 shows that all average RMSE values are less than 0.015 ft/s. The MAD has the same format as Eqs. (6) and (8). Figure 13 shows that all average MAD values are less than 0.0095 ft/s.

7.2 Vehicular motion direction

According to the above speed interpolation methods, the speed over continuous time can be used to estimate vehicular distance. For example, the first sub-function of the vehicle (ID is 2

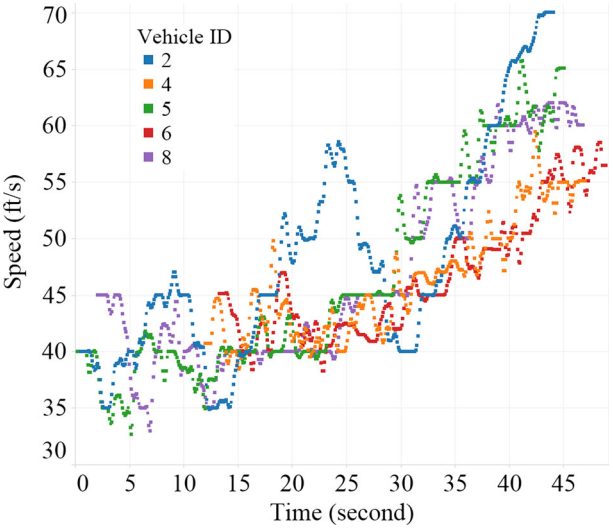

Fig. 10 Individual vehicle speed

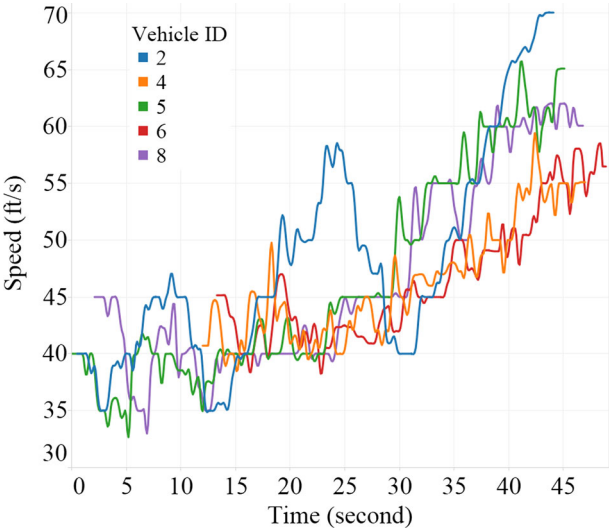

Fig. 11 Speed linear approximation

and time is from 0.5 to 1.1 s) in Fig. 11 estimates the vehicular instantaneous speed as 40.0 ft/s. The travel distance can be estimated as the product of the speed (i.e., 40.0 ft/s) and time step (i.e., 0.6 s) or 24.0 ft. In order to determine vehicular motion direction, the above piecewise-linear approximation is also used to analyze the longitude and latitude of vehicular location points. From the data set D3 the instantaneous location data of five vehicles (vehicle ID is 2, 4, 5, 6, and 8) are demonstrated in Fig. 14. With the location data and the error threshold $\Psi=0.05$ as the input conditions, the piecewise-linear algorithm generates continuous location linear approximations for the estimation of vehicle motion direction with a short time interval choice,

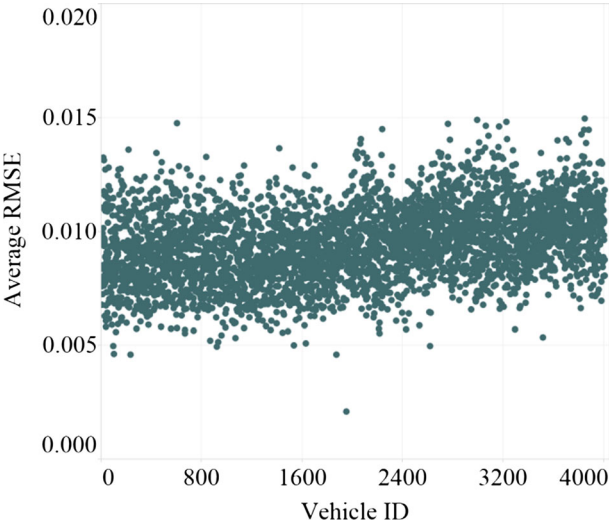

Fig. 12 Speed linear approximation estimation using RMSE

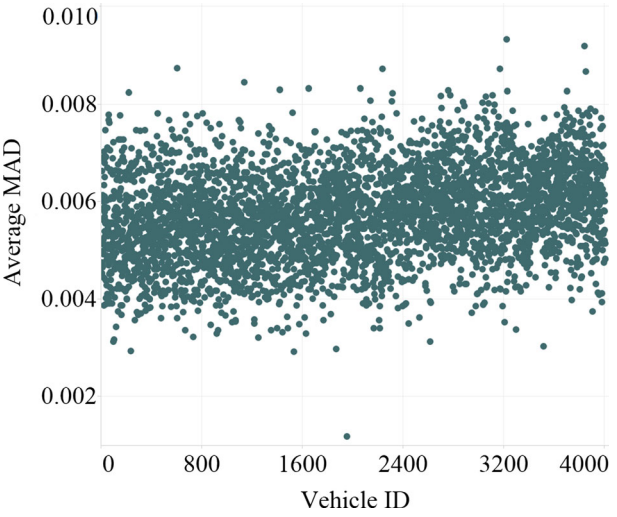

Fig. 13 Speed linear approximation estimation using MAD

as shown in Fig. 15. Certainly, a larger error threshold (Ψ) in the piecewise-linear algorithm would automatically produce less linear sub-functions of vehicle motion direction, but the entire accuracy of linear approximation would decline.

Figures 16 and 17 show the average RMSE and the average MAD of the estimated sub-functions per vehicle in the entire data set D3. Figure 16 shows that all average RMSE values are less than 0.034 ft for the motion direction estimation of the 4100 vehicles in D3. Figure 17 shows that all average MAD values are less than 0.018 ft for the motion direction estimation of the 4100 vehicles in D3.

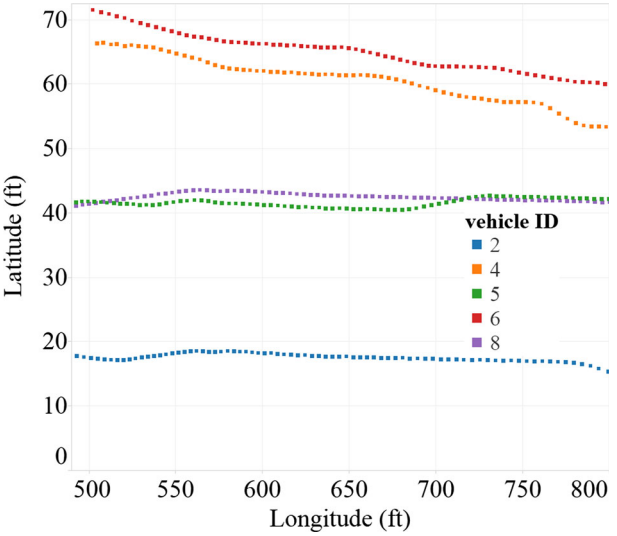

Fig. 14 Individual vehicle location sample

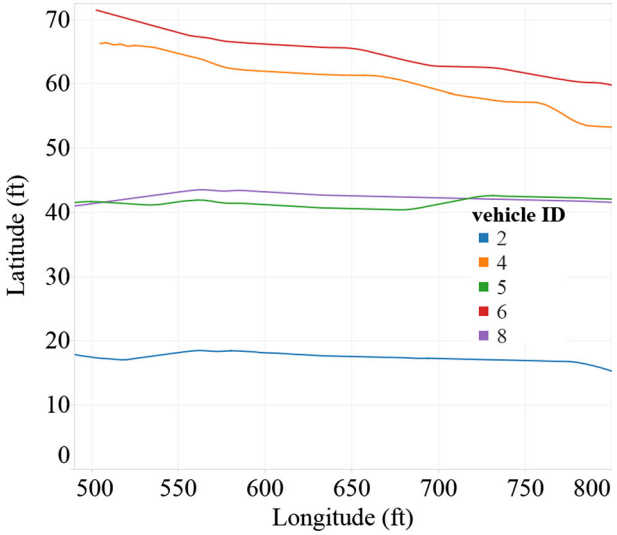

Fig. 15 Location linear approximation

8 Step 5 – Traffic information retrieval

Once the previous four steps are complete, the user can then retrieve traffic information related to the stored vehicle trajectories at any desired temporal and spatial interval. Spatio-temporal

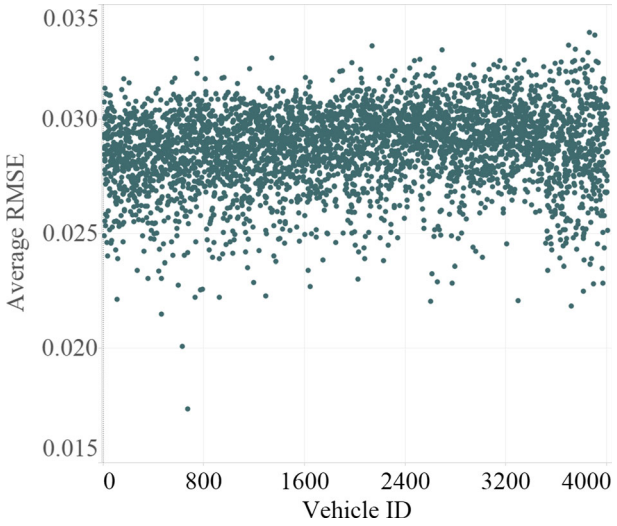

Fig. 16 Location linear approximation estimation using RMSE

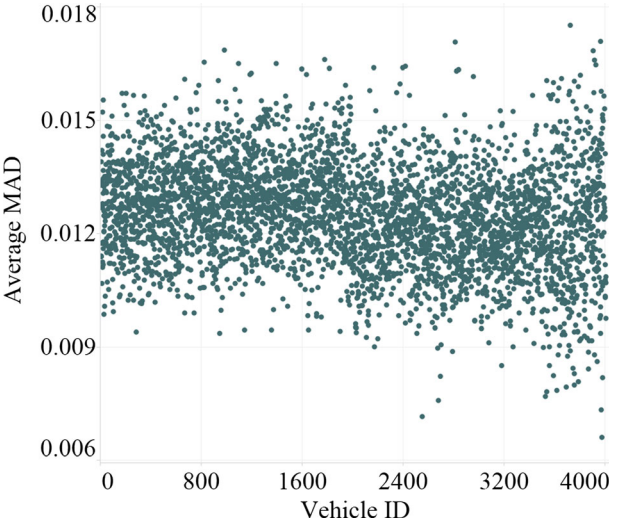

Fig. 17 Location linear approximation estimation using MAD

databases can offer not only distance-based static data operations, similar to GIS programs [73], but also dynamic and/or temporal operations as discussed below.

8.1 Constraint relation

To illustrate, consider the use of the test-bed spatial data represents the U.S. 101 highway in the transportation spatio-temporal system. The input constraint relations are:

Car (id, x, y, s, t), which stores the multiple vehicular motion information at moving location (x, y), time (t), and instantaneous vehicle speed (s). The instantaneous vehicle speeds are described by piecewise-linear functions. Take, for example, the dynamic vehicle in a constraint function form is given below:

$$Car(id, x, y, s, t) : id = 2, y \geq -269, y < -263.8, x - s*t < 136.14, x - s*t \geq 122.64, s = 0.2t + 35.2, t \geq 0, t < 3.$$
$$Car(id, x, y, s, t) : id = 2, y \geq -269, y < -263.8, x - s*t < 137.64, x - s*t \geq 124.14, s = -0.8t + 38.2, t \geq 3, t < 5.$$
$$Car(id, x, y, s, t) : id = 2, y \geq -269, y < -263.8, x - s*t < 136.64, x - s*t \geq 123.14, s = 0.4t + 32.2, t \geq 5, t < 10.$$
$$Car(id, x, y, s, t) : id = 2, y \geq -269, y < -263.8, x - s*t < 128.64, x - s*t \geq 115.24, s = -0.02t + 36.4, t \geq 10, t < 30.$$

Road (x, y), which records the static transportation network.
Take, for example, the static road in a constraint function form is given below:

$$Road(x, y) : -0.0214x - y = 301.79, x \geq 117.75, x < 325.34.$$
$$Road(x, y) : 0.015x - y = 313.646, x \geq 325.34, x < 489.93.$$
$$Road(x, y) : 0.0459x - y = 328.778, x \geq 489.93, x < 640.68.$$
$$Road(x, y) : x - 0.002y = 641.28, y \geq -299.37, y < -52.73.$$

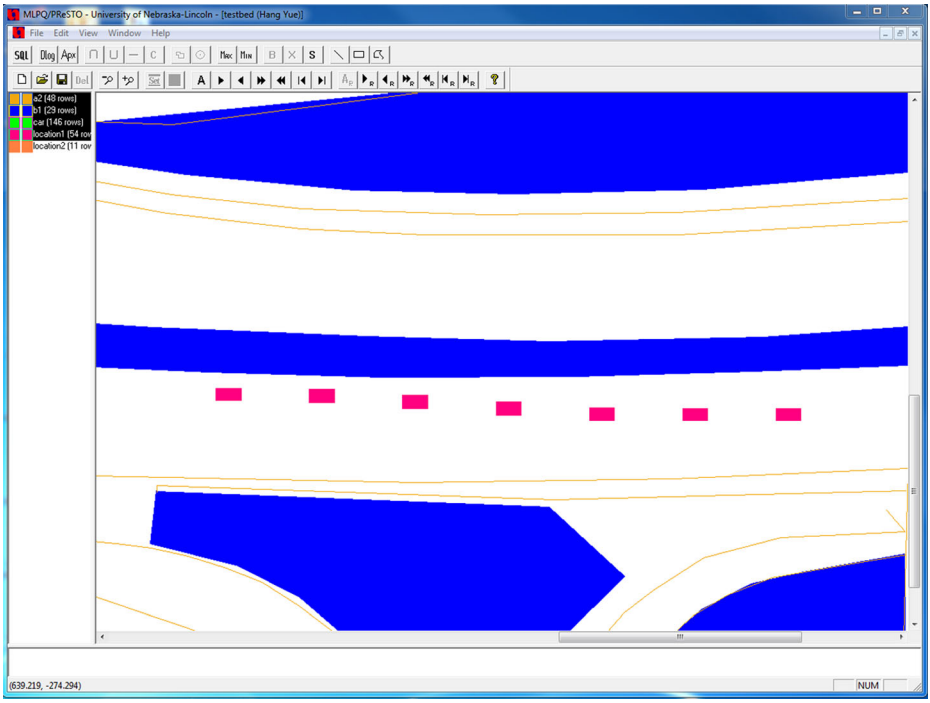

Fig. 18 Individual car tracking and query in the MLPQ system

8.2 Query case

Some example queries are provided below:

Query 1 Find the locations of car 1 at times 2.6, 4.55, 6.5, 8.5, 10.505, 12.51, and 14.49 s, respectively. This query is expressed in Datalog [74] as follows:

Location (x, y): Car (1, x, y, s, 2.6).
Location (x, y): Car (1, x, y, s, 4.55).
Location (x, y): Car (1, x, y, s, 6.5).
Location (x, y): Car (1, x, y, s, 8.5).
Location (x, y): Car (1, x, y, s, 10.505).
Location (x, y): Car (1, x, y, s, 12.51).
Location (x, y): Car (1, x, y, s, 14.49).

Figure 18 illustrates the result of Query 1 in the *MLPQ* (*Management of Linear Programming Queries*) system [9]. The spatial attributes of the roadway are shown by the yellow lines, the pink rectangles show the estimated trajectory of the given vehicle in terms of location and time (e.g., every .2 s), and the filled-in blue sections are road greenbelts.

Query 2 Find the spacing between cars 2 and 6, which travel along the horizontal direction, at time 5 s. The query is given below:

Spacing (sp): Car (2, x2, y, s, 5), Car (6, x6, y, s, 5), sp+x6−x2==0.

The output result of Query 2 is 154.57 ft. Different spacing values can be retrieved from the spatio-temporal database by inputting different time values.

Query 3 Find the volume at location 610.45 ft within the time interval 30 to 50.9 s. The query is designed below:

Reach_line (id, x, t1): Car (id, x, y, s, t1), x==610.45, t1≥30, t1< 50.9.
Reach_time (id, max (t1)): Reach_line (id, x, t1).
Car_time(id, t2): Reach_time (id, t2), t2≥30, t2< 50.9.
Volume (id): Car_time (id, t2).

The volume query results depend on the input location x and time intervals. The transportation spatio-temporal system outputs 5 as the above volume query result and car ID numbers (including cars 6, 7, 9, 10, and 11).

Query 4 Find the vehicular travel times for space-mean speed calculation when cars pass the roadway segment (the location range of road segment is between 150 and 600 ft on the horizontal axis).

Because the space-mean speed is computed as the length of the roadway segment divided by the average time required for traveling the segment [3, 75], Query 4 can be expressed as follows. First, the query time when all the cars reach the location 167.3 ft is:

SpaceA (id, x1, t1): Car (id, x1, y, s, t1), x1≥167.3.
TimeA1 (id, x2, min (t1)): SpaceA (id, x2, t1).
TimeA2 (id, x2, t2): TimeA1 (id, x2, t2), x2==167.3.
Sum_timeA (sum_min (t2)): TimeA2 (id, x2, t2).

Second, the query time when cars pass the location 203 ft is:

SpaceB (id, x3, t3): Car (id, x3, y, s, t3), x3≤203.
TimeB1 (id, x4, max (t3)): SpaceB (id, x4, t3).
TimeB2 (id, x4, t4): TimeB1 (id, x4, t4), x4==203.
Sum_timeB (sum_max (t4)): TimeB2 (id, x4, t4).

The output results are those times when the cars reach the road location 167.3 ft and pass the road location 203.0 ft. The output results also include the sum of these times, which are 87.16 and 122.17 s, respectively. Hence, the average travel time is (122.17–87.16)/12= 2.9175 s, and the space-mean speed is (230–167.3)/2.9175=21.491 ft/s (i.e., 6.55 m/s).

This example illustrates one of the main advantages of the proposed approach. Because the vehicle trajectory data is stored, the space-mean speed for any combination of distance and

time can be estimated accurately and quickly. This obviates the need to store specific combinations of space-mean speed information and interpolated aggregated data that are the hallmarks of traditional traffic data collection and storage systems.

Query 5 Find the time-mean speed at the loop detector station on the highway segment at times 10, 20, 40, 70, and 110 s, respectively. This query is expressed as follows:

Time-meanSpeed (ts): LocalCubicModel (id, ts, t), t==10.
Time-meanSpeed (ts): LocalCubicModel (id, ts, t), t==20.
Time-meanSpeed (ts): LocalCubicModel (id, ts, t), t==40.
Time-meanSpeed (ts): LocalCubicModel (id, ts, t), t==70.
Time-meanSpeed (ts): LocalCubicModel (id, ts, t), t==110.

The loop detector station is set up in the MLPQ system, as shown in Fig. 19. After cars travel over the loop detector, the MLPQ system gives the output results of the time-mean speed (i.e., 89.42 ft/s when t=10 s, 81.93 ft/s when t=20 s, 83.76 ft/s when t=40 s, 92.84 ft/s when t=70 s, and 101.99 ft/s when t=110 s). These results are evaluated using the local cubic model (see Section 6). Other results about time-mean speed can be retrieved by changing time at the loop detector station or location of the loop detector station on the road segment. Thus the MLPQ system can provide a time-mean speed at any time on any highway segment. Certainly, on the basis of Query 4 and Query 5, the MLPQ system can give the difference between time-mean speed and space-mean speed to meet traffic data requirement analysis.

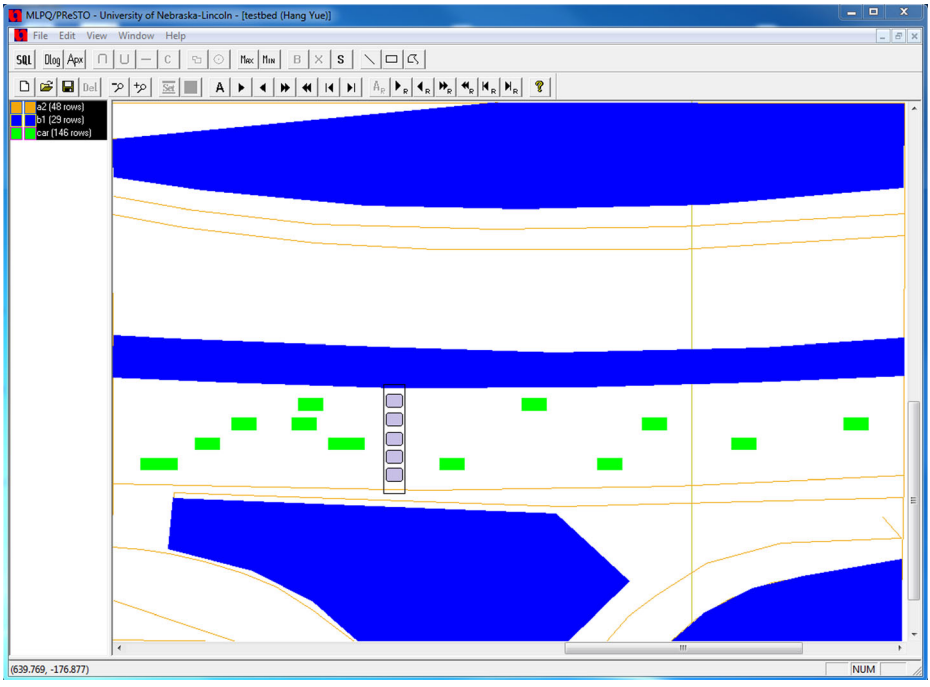

Fig. 19 Moving vehicles traveling over a detector station in the MLPQ system

9 Summary of advantages

This section discusses the advantages of traffic data completeness and data redundancy minimization in the proposed spatio-temporal archiving and retrieval system. Also, the discussion involves a comparison analysis of traffic motion information archived in different databases, and a contrast analysis of static discrete traffic data sources and the dynamic transportation information environment.

9.1 Data completeness

Data completeness requires that data sources in databases should cover all information (i.e., all data types and the complete information of each data type) to meet the current and future demands of various data users. In the approach proposed in this paper, the traffic stream is observed at multiple spatial points within some pre-specified distance intervals over time. This may be contrasted with current techniques, which utilize a single spatial point [75]. Figure 20 shows the traffic stream over continuous time and space as a set of steps. Each step represents the occurrence of an individual vehicle and the edge of each step represents the trajectory of the vehicle.

Existing transportation software systems [77–80] store discrete traffic aggregate data (such as volume, density, space-mean speed, time-mean speed, headway, queue length, spacing, and so on) in relational databases. Aggregate data incompleteness in space and time may lead to poor performance of traffic engineering models in transportation software systems. For example, due to the lack of volume over continuous time and space, not all travelers can gain desired travel time query information from volume-based travel time estimation models in *advanced traveler information systems (ATIS)* [11]. The ability of the user to define the space and time aggregation levels for traffic parameters for each application is an important advantage of the proposed approach.

The transportation spatio-temporal system [81], developed in this paper and based on MLPQ, has a number of advantages over traditional data storage systems currently being used in the transportation profession. This system can offer complete individual vehicle trajectory and traffic aggregate data over continuous space and time. Complete traffic data sources are useful for the description of traffic flow phenomena and for the calculation of various transportation engineering models. Such a spatio-temporal system can be particularly advantageous in understanding highway flow breakdown (e.g., incident detection), and dynamical traffic congestion because a detailed picture of traffic parameters over both time and space is better than these parameters in time alone.

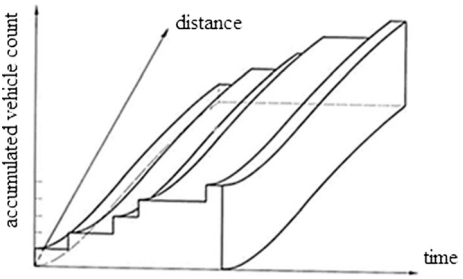

Fig. 20 Traffic stream with a spatio-temporal pattern [76]

Besides individual vehicular and traffic aggregate parameters, the transportation spatio-temporal system can be used to estimate a variety of traffic parameters over user-defined time and space conditions. For example, users could examine speed differences among moving cars at an intersection for any period of time, identify vehicles traveling above the speed limit during any period, and determine the number of trips during any period in the road network.

9.2 Data redundancy

Traditional discrete traffic aggregate data archiving systems result in the loss of a large amount of aggregate traffic data and, conversely, can increase data redundancy in databases. Efficient data operations require data consistency and data synchronization in databases by minimizing or avoiding data redundancy [82].

In relational databases, traffic data redundancy often causes data anomalies, data corruption, and data retrieval errors. For example, updating a certain volume value requires a change in the values of other traffic parameters [3], such as *average daily traffic (ADT)*, *average weekly traffic (AWT)*, *annual average daily traffic (AADT)*, *and annual average weekly traffic (AAWT)*. It is difficult for existing transportation management systems to keep data synchronization between volume values and the above four traffic parameters. The frequent operations of traffic data in databases easily cause data inconsistency or anomalies and data retrieval errors.

Due to the storage of individual vehicular time, location, and instantaneous speed data in the proposed spatio-temporal databases, the traffic parameters, at user-defined aggregation levels, can be retrieved readily using database query designs. Hence the proposed spatio-temporal system provides traffic data archiving methods that eliminate data redundancy.

9.3 Database comparison

The above analysis of data query retrieval, completeness, and redundancy demonstrate that dynamic spatio-temporal databases have significant advantages over video databases. For instance, the proposed spatio-temporal databases offer continuous spatial data and accurate location data (i.e., longitudinal and latitudinal data) similar to what standard GIS can do. From 15 min of traffic video (278 MB), the size of the vehicular data extracted is 83 MB, not including the corresponding GIS shapefiles. After integrating the vehicular data into the GIS shapefiles, the constraint data model of spatio-temporal databases requires only 1 MB data size for the archiving of dynamic transportation data.

Table 5 Characteristics of traffic motion information archived in different databases

Attributes of motion information	Spatio-temporal databases	GIS/spatial databases	Relational databases	Video databases
Time	Continuous	Discrete	Discrete	Tiny interval
Location	Continuous	Continuous	Discrete	Inaccurate
Data type	Complete	Limited	Limited	Very limited
Visualization	Dynamic	Static	Static	Dynamic
Data size	Small	Large	Large	Very large
Retrieval efficiency	No waiting time	Calculation needed	Calculation needed	Video playing time

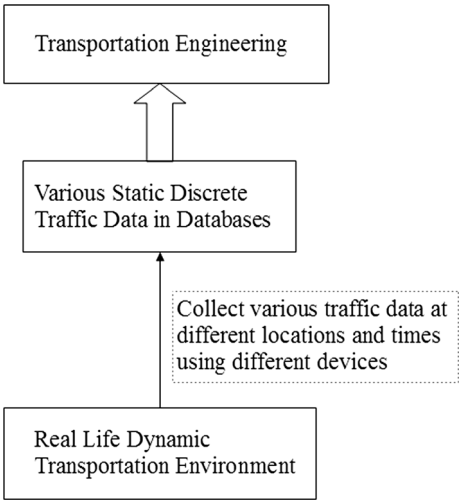

Fig. 21 Static traffic data sources

More importantly, the spatio-temporal data integration, which is a central feature of the proposed methodology, allows the aggregated traffic data to be retrieved in near real-time. To retrieve similar data from current systems (e.g., aggregated traffic data and archived video

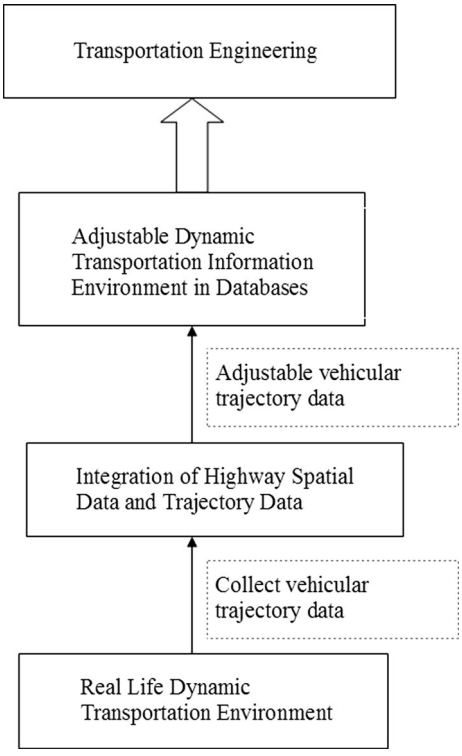

Fig. 22 Adjustable dynamic transportation information

data) would be extremely labor-intensive and time consuming – to the point that it is rarely done in practice. Moreover, due to the lack of map data in video frames or segments, video databases cannot provide exact longitudinal and latitudinal data. Table 5 summarizes different attributes of traffic motion information archived in different databases as follows:

9.4 Data operation

In contrast to static traffic data sources in existing transportation systems, as shown in Fig. 21, the proposed spatio-temporal system offers an adjustable dynamic transportation information environment, as shown in Fig. 22. It means that the data collection of individual vehicle trajectory would be more important than traffic aggregate data for data collection and storage for transportation applications. The integration of highway spatial data and vehicle trajectory data create the spatio-temporal logical relationships among the entire transportation motion data.

10 Concluding remarks

Video cameras can easily collect traffic information, but storing the raw video data generally requires a huge storage space. More importantly, it is difficult to retrieve the values of traffic parameters from video data for the calculations of transportation engineering models or the development of transportation software, not to mention traffic data operation or adjustability. The proposed transportation spatio-temporal system is designed to overcome the storage problem by converting traffic videos into a spatio-temporal database. Because the transportation spatio-temporal system was developed on top of the MLPQ system, it allows high-level Datalog and SQL queries, including specific predefined queries related to traffic management. The queries of the transportation spatio-temporal system can search the complete continuous motion information of the moving vehicles.

Further research may lead to the analysis of different traffic situations that result in heavy traffic congestion or collisions [83]. The ultimate goal of traffic management system development is to improve traffic and road conditions for drivers and their vehicles.

Acknowledgments The authors gratefully acknowledge Elizabeth G. Jones for her comments about Fig. 7 and pointing out some related references.

References

1. Miller HJ (1999) Potential contributions of spatial analysis to geographic information systems for transportation. Geogr Anal 31:373–399
2. Butler AJ, Dueker KJ (2001) Implementing the enterprise GIS in transportation database design. URISA J 13(1)
3. Roess RP, Prassas ES, McShane WR (2011) Traffic Engineering, 4th edn. Pearson Prentice Hall, Upper Saddle River
4. Gong Y (2003) Audio and visual content summarization of a video program, chapter 10. In: Furht B, Marques O (eds) Handbook of video databases: design and applications

5. Wactlar HD, Christel MG, Gong Y, Hauptmann AG (1999) Lessons learned from building a terabyte digital video library. IEEE Comput 32:66–73
6. Aslandogan YA, Yu CT (1999) Techniques and systems for image and video retrieval. IEEE Trans Knowl Data Eng 11:56–63
7. Agma J, Traina M, Traina C Jr. (2003) Similarity search in multimedia databases, chapter 29. In: Furht B, Marques O (eds) Handbook of video databases: design and applications, pp. 712–245
8. Li L, Revesz PZ (2004) Interpolation methods for spatiotemporal geographic data. Comput Environ Urban Syst 28(3):201–227
9. Revesz PZ (2010) Introduction to database: from biological to spatio-temporal. Springer, New York
10. Chen CX (2001) Data models and query languages of spatio-temporal information. Ph.D. Dissertation, University of California, Los Angeles, CA
11. Yue H (2009) Advanced traveler information inquiry, archiving, and decision making system, the 4th Chinese Oversea Student "Chun Hui Cup" Entrepreneurship Competition, Project Presentation
12. Yin H, Wolfson O (2004) A weight-based map matching method in moving objects databases, the 16th International Conference on Scientific and Statistical Database Management
13. Liu J, Wolfson O, Yin H (2006) Extracting semantic location from outdoor positioning systems. International Workshop on Managing Context Information and Semantics in Mobile Environments
14. Cambridge Systematics, Inc. (2005) NGSIM U.S. 101 data analysis (7:50 a.m. to 8:05 a.m.), Prepared for Federal Highway Administration
15. Wei H, Feng C, Meyer E, Lee J (2005) Video-capture-based approach to extract multiple vehicular trajectory data for traffic modeling. J Transp Eng 131(7):496–505
16. Skabardonis A, Alexiadis V (2005) Traffic data through the Berkeley highway laboratory. Workshop on Traffic Modeling, Sedona, AZ
17. Kim Z, Gomes G, Hranac R, Skabardonis A (2005) A machine vision system for generating vehicle trajectories over extended freeway segments, the 12th World Congress on Intelligent Transportation Systems
18. Tao RH, Wei H, Wang YH, Sisiopiku VP (2004) Modeling speed disturbance absorption following state-control action-expected chains: integrated car-following and lane-changing scenarios, the 83rd Annual Meeting of Transportation Research, Washington, D.C
19. Wei H (1999) Observed lane-choice and lane-changing behaviors on an urban street network using video-capture-based approach and suggested structures of their models. Ph.D. dissertation, Univ. of Kansas, KS
20. U.S. Department of Transportation, Federal Highway Administration (2007) NGSIM-VIDEO user's manual, Publication No. #FHWA-HOP-07-009
21. Patamanska G, Slavov N (2007) Using cubic spline interpolation to estimate vertical soil water profile. Bulg J Agric Sci 13:317–323
22. Boyko A, Pavlova V (1986) Restoration of soil moisture reserve profile using spline interpolation instrument. Tr VNISKHM 21:102–111
23. Thant A, Khaing A (2009) Application of cubic spline interpolation to walking patterns of biped robot. World Acad Sci Eng Technol 50:27–34
24. Su B, Tan J (2007) Sweeping surface generated by a class of generalized quasi-cubic interpolation spline. Int Conf Comput Sci (2):41–48
25. Dubau C (2011) Optimal property of the shape of aeolian blade profile using cubic splines. J Comput Anal Appl 13(1):254
26. Recktenwald G (2000) Numerical methods with MATLAB: implementation and application. Prentice Hall, Upper Saddle River
27. Guo D, Liu S, Jin H (2010) A graph-based approach to vehicle trajectory analysis. J Locat Based Serv 4: 183–199. doi:10.1080/17489725.2010.537449
28. Liu S, Liu C, Luo Q, Ni L, Krishnan R (2012) Calibrating large scale vehicle trajectory data. Living Analytics Research Center (LARC), Technical Report Series: LARC-TR-01-12
29. Douglas D, Peucker T (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Can Cartographer 10:112–122
30. Jeung H, Yiu ML, Zhou X, Jensen CS, Taoshen H (2008) Discovery of convoys in trajectory databases. VLDB Endowment 1:1068–1080
31. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ACM Press, pp 593–604
32. Rinzivillo S, Pedreschi D, Nanni M, Giannotti F, Andrienko N, Andrienko G (2008) Visually driven analysis of movement data by progressive clustering. Inf Vis 7:225–239
33. Gindele T, Brechtel S, Dillmann R (2010) A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments, the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal

34. Egerstedt M, Martin CF (2001) Optimal trajectory planning and smoothing splines. Automatica 37:1057–1064
35. Shcherbakov MV, Brebels A (2011) Outliers and anomalies detection based on neural networks forecast procedure. In: Proceedings of the 31st Annual International Symposium on Forecasting, ISF-2011, pp 21–22
36. Pham-GIA T, Hung TL (2001) The mean and median absolute deviation. Math Comput Model 34:921–936
37. Shcherbakov MV, Brebels A, Shcherbakova NL, Tyukov AP, Janovsky TA, Kamaev VA (2013) A survey of forecast error measures. World Appl Sci J 24 (Information Technologies in Modern Industry, Education & Society):171–176, ISSN 1818–4952
38. Rousseeuwa PJ, Crouxa C (1993) Alternatives to the median absolute deviation. J Am Stat Assoc 88(424): 1273–1283
39. Highway Capacity Manual (2000) Transportation Research Board; Bk&CD-Rom edition, ISBN-10: 9991332944, Chapter 7–3
40. Hastie T, Loader C (1993) Local regression: automatic kernel carpentry. Stat Sci 8(2):120–129
41. Nadaraya EA (1964) On estimating regression. Theory Probab Appl 10:186–190
42. Wathson GW, Leadbetter MR (1964) Hazard analysis I. Biometrika 51:175–184
43. Gasser T, Müller HG (1979) Kernel estimation of regression functions. In: Gasser T, Rosenblatt M (eds) Smoothing techniques for curve estimation. Springer, Heidelberg, pp 23–68
44. Silverman BW (1986) Density estimation for statistics and data analysis. Chapman and Hall, London
45. Fan J, Marron JS (1994) Fast implementations of nonparametric curve estimators. J Comput Graph Stat 3: 35–56
46. Fan J, Gijbels I (1992) Variable bandwidth and local linear regression smoothers. Ann Stat 20(4): 2008–2036
47. Ruppert D, Wand M (1994) Multivariate locally weighted least-squares regression. Ann Stat 22(3): 1346–1370
48. Cheng MY, Fan J, Marron J (1997) On automatic boundary corrections. Ann Stat 25(4):1691–1708
49. Sun H, Liu HX, Xiao H, He RR, Ran B (2003) Use of local linear regression model for short-term traffic forecasting. Transp Res Board 18(1836):59–71
50. Fan J, Gijbels I (1996) Local polynomial modeling and its applications. Chapman & Hall, London
51. Smith B, Williams B, Oswald K (1999) Parametric and nonparametric traffic volume forecasting. Transportation Research Record. CDROM
52. Faouzi E (1996) Nonparametric traffic flow prediction using kernel estimator. Proceedings of the 13th International Symposium on Transportation and Traffic Theory, Lyon, France, July 19, pp 24–26
53. Rasmussen C, Williams C (2006) Gaussian processes for machine learning. MIT Press, Cambridge, Massachusetts
54. Anjyo K, Lewis JP. RBF interpolation and Gaussian process regression through an RKHS formulation. J Math Ind 3 (2011A-6):63–71
55. Williams CKI, Rasmussen CE (1996) Gaussian processes for regression. In: Proceeding of the Neural Information Processing Systems, vol 8. MIT Press
56. Stein ML (1999) Interpolation of spatial data: some theory for Kriging. Springer, New York
57. Ranjan P, Haynes R, Karsten R (2011) A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. Technometrics 53(4):366–378
58. Yakowitz SJ, Szidarovszky F (1985) A comparison of Kriging with nonparametric regression methods. J Multivar Anal 16:21–53
59. Scott DW, Tapia RA, Thompson JR (1977) Kernel density estimation revisited. Nonlinear Anal Theory Methods Appl 1:339–372
60. Cleveland W (1979) Robust locally weighted regression and smoothing scatter plots. J Am Stat Assoc 74: 829–836
61. Fan J (1992) Design-adaptive nonparametric regression. J Am Stat Assoc 87:998–1004
62. Wand MP, Jones MC (1995) Kernel smoothing. Chapman and Hall, London
63. Jones MC, Marron JS, Sheather SJ (1996) A brief survey of bandwidth selection for density estimation. J Am Stat Assoc 91(433):401–407
64. Scott DW, Terrell GR (1987) Biased and unbiased cross-validation in density estimation. J Am Stat Assoc 82:1131–1146
65. Bowman AW (1984) An alternative method of cross-validation for the smoothing of density estimates. Biometrika 71:353–360
66. Sheather SJ (1986) An improved data-based algorithm for choosing the window width when estimating the density at a point. Comput Stat Data Anal 4:61–65
67. Park BU, Marron JS (1990) Comparison of data-driven bandwidth selectors. J Am Stat Assoc 85:66–72
68. Sheather SJ, Jones MC (1991) A reliable data-based bandwidth selection method for kernel density estimation. J R Stat Soc Ser B 53:683–690

69. Engel J, Herrmann E, Gasser T (1995) An iterative bandwidth selector for kernel estimation of densities and their derivative. J Nonparametric Statist 4:21–34
70. Faraway JJ, Jhun M (1990) Bootstrap choice of bandwidth for density estimation. J Am Stat Assoc 85:1119–1122
71. Ruppert D, Sheather SJ, Wand MP (1995) An effective bandwidth selector for local least squares regression. J Am Stat Assoc 90:1257–1270
72. Yue H, Jones E, Revesz PZ (2010) Local polynomial regression models for average traffic speed estimation and forecasting in linear constraint databases, 17th IEEE International Symposium on Temporal Representation and Reasoning, Paris, France, pp 154–161
73. Yue H, Jones E (2010) Archiving capability of spatio-temporal data in different Highway Railroad Grade Crossing (HRGC) databases, Annual Intelligent Transportation System Conference, Houston, USA
74. Kanellakis P, Kuper G, Revesz PZ (1995) Constraint query languages. J Comput Syst Sci 51(1): 26–52
75. Kerner BS (2009) Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory. Springer, Berlin
76. Leutzbach W (1988) Introduction to the theory of traffic flow. Springer, Berlin
77. Al-Deek H, Abd-Elrahman A (2002) An evaluation plan for the conceptual design of the Florida transportation data warehouse, University of Central Florida, Technical Report No. 16-50-706
78. Dahlgren J, Garcia RC, Turner S (2001) Completing the circle: using archived operation data to better link decision to performance, California Path Research Report No. UCB-ITS-PRR-2001-23
79. Liu HX, He R, Tao Y, Ran B (2002) A literature and best practices scan: its data management and archiving, University of Wisconsin at Madison Technical Project No. 0092-02-11
80. Yue H, Yang R (2005) Development of Intelligent Transportation Systems (ITS) and plan of integrated information system. Journal of Wuhan University of Technology 29(4):560–563
81. Yue H, Revesz PZ (2012) TVICS: an efficient traffic video information converting system, 19th IEEE International Symposium on Temporal Representation and Reasoning, Leicester, UK, pp 141–148
82. Schwinn A, Schelp J (2003) Data integration patterns. Business Information Systems Conference, Colorado Springs, USA
83. Anderson S, Revesz PZ (2009) Efficient max count and threshold operators of moving objects. Geoinformatica 13(4):355–396

Hang Yue obtained his M.S. degree in Transportation Engineering and Minors in Statistics and Geography from University of Nebraska—Lincoln in 2012 and M.S. degree in Computer Software Engineering from Zhejiang University in 2005. He served as a Data Analyst at Airsage Inc. in 2013, and a Data Scientist at Charter Global Inc. in 2014. He is currently a Data Analyst at Johns Hopkins HealthCare LLC. His research interests are machine learning, big data business intelligence, data warehouse, data visualization, geographic information systems (GIS), and spatio-temporal databases.

Laurence Rilett received his B.A.Sc. degree and his M.A.Sc. degree from the University of Waterloo and his Ph.D. degree from Queen's University. He is a Distinguished Professor of Civil Engineering at the University of Nebraska-Lincoln (UNL), the inaugural holder of the Keith W. Klaasmeyer Chair in Engineering and Technology, and serves as Director of the Nebraska Transportation Center.

Peter Revesz holds a Ph.D. degree in Computer Science from Brown University. He was a postdoctoral fellow at the University of Toronto before joining the University of Nebraska-Lincoln, where he is a professor in the Department of Computer Science and Engineering. Dr. Revesz is an expert in databases, data mining, big data analytics and bioinformatics. He is the author of Introduction to Databases: From Biological to Spatio-Temporal (Springer, 2010) and Introduction to Constraint Databases (Springer, 2002). Currently a Visiting Program Manager at the Air Force Office of Scientific Research, Revesz held previous visiting appointments at the IBM T. J. Watson Research Center, INRIA, the University of Hasselt, the Max Planck Institute for Computer Science, the University of Athens, and the U.S. Department of State, where he served as a Scientific Advisor in the Bureau of International Security and Nonproliferation. He is a recipient of an AAAS Science & Technology Policy Fellowship, a J. William Fulbright Scholarship, an Alexander von Humboldt Research Fellowship, a Jefferson Science Fellowship, a National Science Foundation CAREER award, and a "Faculty International Scholar of the Year" award by Phi Beta Delta, the Honor Society for International Scholars.