2012 19th International Symposium on Temporal Representation and Reasoning

TVICS: An Efficient Traffic Video Information Converting System

Hang Yue

Mid-America Transportation Center
Civil Engineering Department
University of Nebraska-Lincoln
Lincoln, NE 68588, USA (until April 30th, 2012)
Email: yuehang366@gmail.com

Peter Z. Revesz

Computer Science & Engineering Department
University of Nebraska-Lincoln
Lincoln, NE 68588, USA
Email: revesz@unl.cse.edu

Abstract — **This paper presents a new system called TVICS that converts traffic video data into vehicular motion information in spatio-temporal databases. The TVICS system interpolates the vehicular trajectory data (time, location and velocity), which are extracted from video, and integrates them with spatial road information for the storage of dynamic transportation environments. TVICS can avoid the storage issues caused by traffic videos with their associated large data size. Moreover, users can manage and operate multiform and multidimensional traffic data in a spatio-temporal transportation environment. Experimental results show that TVICS has a high accuracy in transportation applications.**

Keywords – GIS; traffic video; spatio-temporal database; vehicular velocity interpolation; transportation

I. INTRODUCTION

In this paper we consider transportation applications. Some transportation-related applications, such as urban planning, require only static spatial databases or *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. In addition, 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, more interesting transportation applications need to consider the values of traffic parameters that vary continuously over time. Spatial database systems deal with these data sets in an inefficient way via discrete time points or intervals. Given traffic data's multiform and multidimensional nature, more efficient traffic data archiving is needed to add a temporal dimension to GIS-based transportation management systems.

Today, video cameras are widely used for traffic monitoring and data collection. The combination of space and time is a defining feature of digital video [3]. However, considering the large space and expensive cost in traffic video storage, traffic video data are usually saved into video segments, scenes, shots, or frames [4]. Moreover, in current video management systems the storage manner of traffic videos is defined on the physical level, which does not allow the expression of high-level spatio-temporal relationships among traffic data [5, 6]. Therefore, it is difficult for current video database systems to quickly scan traffic video data and find the desired transportation-related spatio-temporal query results.

We have looked at alternatives for storing video data. First, we can recognize that once the video cameras collect discrete vehicular trajectory data (some number of frames per second), and the trajectory data can be interpolated into continuous traffic data. Data model can describe the movement of vehicles with some functions of a temporal parameter *t* and spatial parameters. Second, such continuous traffic data can be conveniently stored in various types of spatio-temporal databases. For example, constraint databases can describe continuous spatio-temporal data in arbitrarily high-dimension [7]. At the same time, constraint databases allow various high-level query languages, including SQL and Datalog and their extensions [7].

The aim of this paper is to describe the development and features of an efficient *Traffic Video Information Converting System (TVICS)*. Our TVICS system can convert traffic video data into spatio-temporal transportation databases. The system allows the users to choose various data interpolation options. In addition, we integrate into the system not only standard SQL queries but also high-level queries that are specifically designed for transportation-related applications.

II. OVERVIEW OF THE TVICS SYSTEM

The overall design and data management plan of the TVICS system are illustrated in Fig. 1. The TVICS system can optimize traffic data completeness and offer high-level spatio-temporal queries of transportation data. The design and development of the TVICS system consists of the following four main parts:

- Data extraction: by the video-capture methods [8, 11], traffic video data extraction provides vehicular trajectory data (i.e. vehicular instantaneous location, time, and speed data points).

1530-1311/12 $26.00 © 2012 IEEE
DOI 10.1109/TIME.2012.9

- Data interpolation: discrete vehicular velocity points with different time intervals are interpolated into continuous instantaneous velocity by the linear and nonlinear data models.
- Data integration: both of the highway spatial data of GIS shape-files and the continuous vehicular trajectory data are transformed and integrated into spatio-temporal databases.
- Data retrieval: a high level traffic information query interface guides the users in performing spatio-temporal queries of the integrated dynamic transportation information.

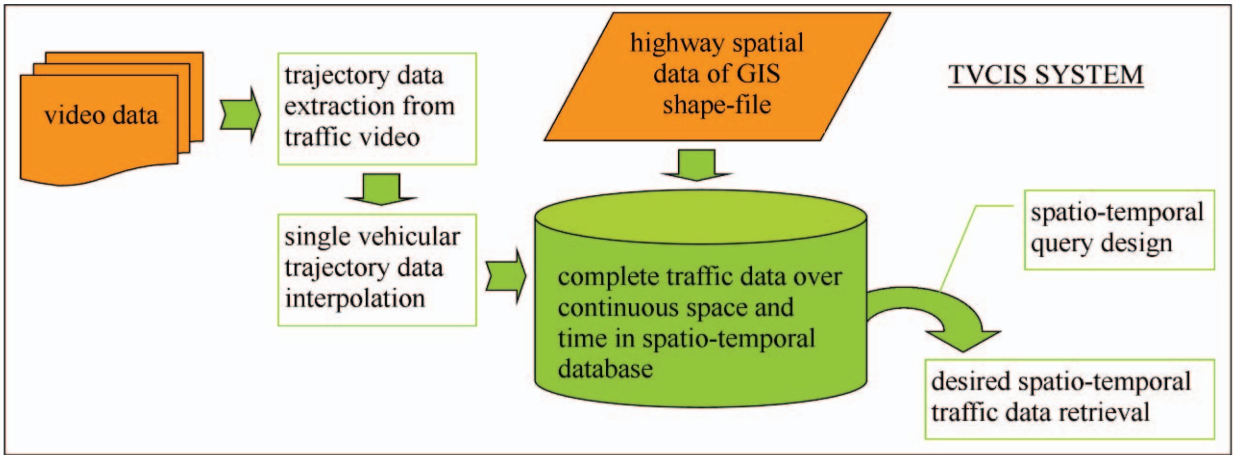

video data
trajectory data extraction from traffic video
highway spatial data of GIS shape-file
TVCIS SYSTEM
single vehicular trajectory data interpolation
complete traffic data over continuous space and time in spatio-temporal database
spatio-temporal query design
desired spatio-temporal traffic data retrieval

Figure 1. Overview of the TVICS system

III. DATA EXTRACTION

A. *Trajectory Data Extraction Steps*

Today, when high resolution cameras, good quality video-capture cards, and advanced video-capture-based approaches are available, it is increasingly cost-effective to extract accurate multiple-vehicular trajectory data. For instance, the advanced machine vision system in the *Next Generation Simulation (NGSIM)* program [8] can automatically extract vehicle trajectories from video data. The machine vision algorithms [9, 10] concerning vehicle detection and tracking were developed to obtain the comprehensive dataset about vehicle trajectory at 10 frames per second. The software *Vehicle Video-capture Data Collector (VEVID)* [11] can digitize full-motion video at a higher frame rate (up to 30 frames per second).

Wei et al. [11] described a general approach to extract vehicular trajectory data from video for traffic modeling. Fig. 2 shows the five steps about traffic data collection and extraction using the video-capture method.

- Step 1: to measure the distances between the reference points of the urban street.
- Step 2: to set up a camera in an elevated position above the urban street.
- Step 3: to digitize a segment of video into *Audio Video Interleave (AVI)* or Video for Windows with a user specified frame rate.
- Step 4: to put the AVI file and the distance information about the reference points in the software (VEVID or advanced machine vision system) to extract vehicular trajectory data.
- Step 5: to implement the vehicular trajectory data storage.

B. *Current Data Archiving Methods*

Traffic data collection technology has advanced faster than the technology of transportation databases into which data are properly archived and from which all data can be conveniently retrieved [12]. Some researches [8, 11] developed the approaches to extract vehicular trajectory data with small time intervals (e.g. 0.1 s) from video for traffic modeling. The studies [13, 14] used vehicular 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.

However, discrete vehicular trajectory data are stored in flat-files or relational databases, and these vehicular motion data are separated from highway spatial data in GIS. This separation causes a loss of information about the spatial relationships between the moving vehicles and the highways, and also leaves undefined the spatio-temporal relationships between the moving vehicles. Existing transportation databases fail to properly keep spatio-temporal relationships among traffic data and also fail to provide high-level spatio-temporal queries, such as queries that involve multiform and multidimensional traffic data. Take, for example, traffic spatio-temporal queries that aggregate traffic data by any time period, lane, and vehicle type or vehicular queries that track certain vehicles for a driver behavior analysis.

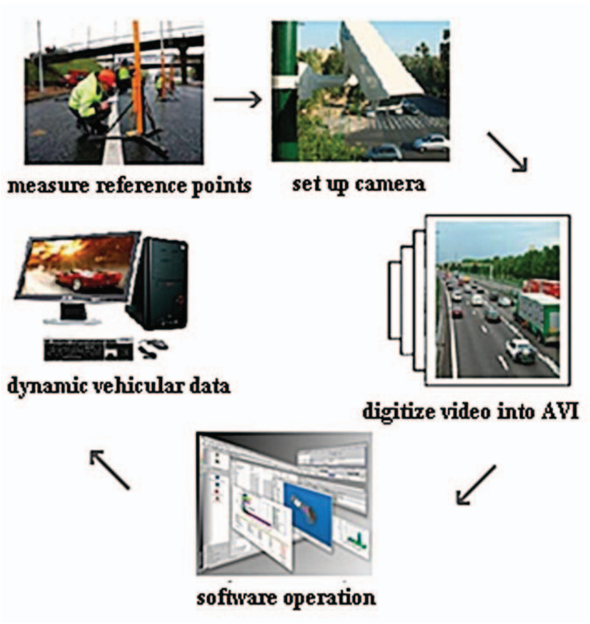


Figure 2. Traffic data collection and extraction using video-capture techniques.

Spatio-temporal databases can integrate a dynamic temporal effect with a description of spatial dimensions [15, 28]. However, in terms of tracking moving vehicles in a visual road network, the current studies [16, 17] mainly focused on the accuracy discussion about snapping discrete GPS points to a certain road segment. Vehicular data with high accuracy, extracted from video, were not discussed for the development of transportation systems, and need to be adapted for such a task.

IV. DATA INTERPROLATION

A. *Data Interpolation Methods*

In Yue et al. [18], the authors developed continuous time-mean speed estimation models for transportation applications in spatio-temporal databases, on the basis of the statistical interpolation methods. In order to achieve moving vehicles within dynamic transportation databases, cubic-splines are used to interpolate discrete vehicular velocity data into continuous instantaneous velocity.

Cubic-splines are typical numerical analysis methods for data interpolations. There are four types of cubic-splines, and they are the exact-slope spline, natural spline, zero-slope spline, and not-a-knot spline. The common function of cubic-spline $S(x)$ in [19] is:

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

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

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

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; and the coefficients on the cubic polynomials (a_i, b_i, $c_{i,}$ and d_i) are the weights of interpolating known data.

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

$$s'_i(x)=3a_i(x-x_i)^2+2b_i(x-x_i)+c_i \qquad (3)$$

$$s''_i(x)=6a_i(x-x_i)+2b_i \qquad (4)$$

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; that is (for $2 \le \mathrm{i} \le \mathrm{n}-1$):

$$S_i(x_i)=S_{i-1}(x_i) \qquad S'_i(x_i)=S'_{i-1}(x_i)$$

$$S''_i(x_i)=S''_{i-1}(x_i) \qquad h=x_i-x_{i-1}$$

The piecewise function $S(x)$ interpolates all discrete data points, $S(x_i)$ = y_i for $1 \le \mathrm{i} \le \mathrm{n}-1$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 ($1 \le \mathrm{i} \le \mathrm{n}-1$) are concluded below:

$$a_i=\frac{M_{i+1}-M_i}{6h} \qquad b_i=\frac{M_i}{2}$$

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

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 that is $s''_1(x_1)=s''_n(x_n)=0$. And the zero-slope spline has the zero slopes in x_1 and x_n, i.e. $s'_1(x_1)=s'_n(x_n)=0$.

B. *Data Sample*

From the NGSIM program, US 101 (Hollywood Freeway) in Los Angeles, CA is chosen as the test bed. This test bed involves GIS shape-files (a shape-file generally contains important spatial information and geometry features) and multiple vehicular trajectory data extracted from video (time, location, velocity, vehicle class, lane identification, etc.). The vehicle trajectory data was collected by the video cameras on the south-bound road between 7:50 am and 8:05 am on June 15, 2005. We used the trajectory data from the first video camera.

To achieve a tolerance (e) of ± 1.0 mph (i.e. ± 0.447 m/s), practical use is made of the knowledge that most velocity distributions have standard deviations (s) of approximately 5.0 mph. With 95% confidence, the formula $N \geq 1.96^2 s^2/e^2 = 96$ (1.96 is the 0.975 quartile of the standard normal distribution), so we randomly take 96 vehicles' trajectory data with a time interval 0.1 s from 1736 vehicles of NGSIM.

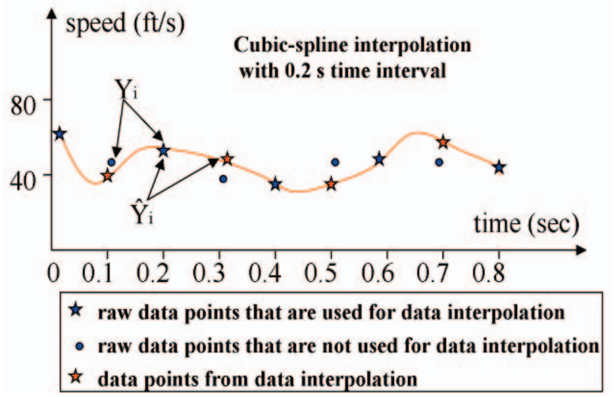


Figure 3. Cubic-spline interpolation.

C. *Velocity Interpolation Estimation*

Fig. 3 illustrates the accuracy estimation method about vehicular instantaneous speed via *Root Mean Square Error (RMSE)*. RMSE and its mean are defined in the following two equations:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n}} \tag{5}$$

$$\mu = \frac{\sum_{j=1}^{96} RMSE_j}{96} \tag{6}$$

where n is the number of the speed data points in the j^{th} vehicle; $\hat{Y}_i$ is the speed value from the cubic-spline interpolation models; Y_i is the raw speed point; and μ is the mean of RMSE about 96 vehicles' speed.

Our study gives the calculation results concerning four data interpolation models of individual vehicular velocity in Table 1, such as the piecewise-linear, not-a-knot spline, natural spline, and zero-slope spline. The comparison of data interpolation models does not include the exact-slope spline. The reason is that the acceleration values, i.e. the slopes of vehicular speed data, are usually unknown conditions, and they are calculated by speed and time values.

In terms of the estimation results of the three cubic-spline models, the error values in Table 1 show that the zero-slope spline is the best approach to interpolate individual vehicular velocity. The second option is the natural spline, and it has higher accuracy than the not-a-knot spline. This conclusion does not correspond to the contrast result of the three models in interpolating the function $y = xe^{-x}$ (the zero-slope spline is the worst interpolation model for this mathematical function interpolation) [19]. It means that vehicular velocity data have specific characteristics, and the vehicular velocity characteristics are different from the non-linear characteristics of general mathematical functions.

In comparison to the piecewise linear model, the three cubic splines are significantly better for vehicular speed interpolation in small time intervals, i.e. 0.2 s and 0.3 s. However, in a larger time interval (0.7 s), the three cubic splines lose their superiority. Even these splines are a little worse than the piecewise linear model in a 1.0 s' time interval. In addition, the piecewise linear model sometimes performs better than some of the more complex interpolation methods in earlier experiments on various spatio-temporal interpolation problems [20].

TABLE 1. INTERPOLATION MODEL ESTIMATION

Time interval (sec)	Piecewise-linear	Not-a-knot	Natural	Zero-slope
	μ (ft/s)	μ (ft/s)	μ (ft/s)	μ (ft/s)
t = 0.2	0.1937	0.1302	0.1294	0.1292
t = 0.3	0.3347	0.2493	0.2469	0.2461
t = 0.5	0.5895	0.5432	0.5324	0.5292
t = 0.7	1.0075	0.9882	0.9770	0.9725
t = 1.0	1.3559	1.4495	1.4335	1.4367

(Note that 1.0 ft/s = 0.3048 m/s)

V. DATA INTEGRATION

The integration of highway spatial data and vehicular trajectory data creates the dynamic transportation environment in spatio-temporal databases. The highway spatial data come from the transformation of GIS shape-files. The trajectory data transformation consists of two parts: the linear approximation of vehicular instantaneous velocity and the determination of instantaneous motion direction.

A. *Linear Approximation of Velocity*

The curves in computer software systems are not absolute non-linear; instead, they are some approximate linear data. By using the velocity points per 0.1 s from the cubic-spline data interpolation method, the piecewise-linear approximation algorithm [18, 28] automatically create the piecewise-linear functions with a high accuracy. A smaller error threshold (Ψ) in the piecewise-linear algorithm can produce more sub-functions for more accurate speed curve approximation. From the above data source in Section 4.2, the velocity data of one vehicle

(vehicle ID: 2531; vehicle class: auto; and total 300 velocity points in 30 s) are illustrated in Fig. 4.

With the 30 seconds' data sample and the error threshold Ψ = 0.05 as the input conditions, the piecewise-linear algorithm produces velocity approximations as shown in Fig. 5. The plot of the velocity approximation is the piecewise-linear function including 180 sub-functions.

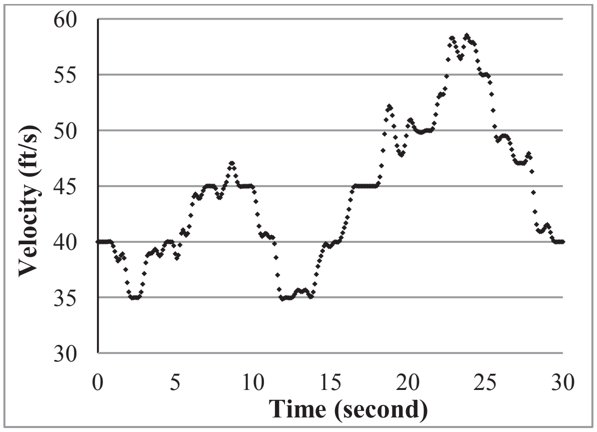


Figure 4. Individual vehicle velocity sample.

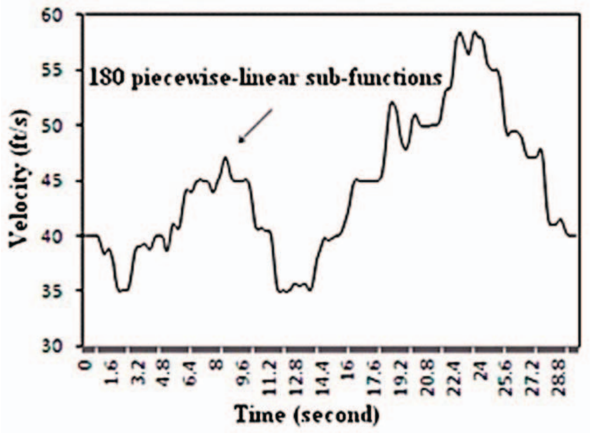


Figure 5. Velocity approximation (Ψ = 0.05).

Fig. 6 shows the accuracy estimation of every sub-function via RMSE. The RMSE has the same function as Eq. (5), but where n is the number of the data points around a sub-function of the piecewise-linear function; $\hat{Y}_i$ is the velocity calculated by the sub-function; and Y_i is the velocity of the data point. Most of the RMSE values are very close to 0 ft/s. Moreover, the RMSE values concerning just a few sub-functions are more than 0.02 ft/s. When using the error threshold Ψ = 0.5 as the input condition, the piecewise-linear algorithm creates 48 sub-functions with RMSE value range between 0.2 and 0.45 ft/s. By using Ψ = 0.05, the calculation results show that the RMSE values of 96 vehicles' velocity data are less than 0.11 ft/s (i.e. 0.335 m/s).

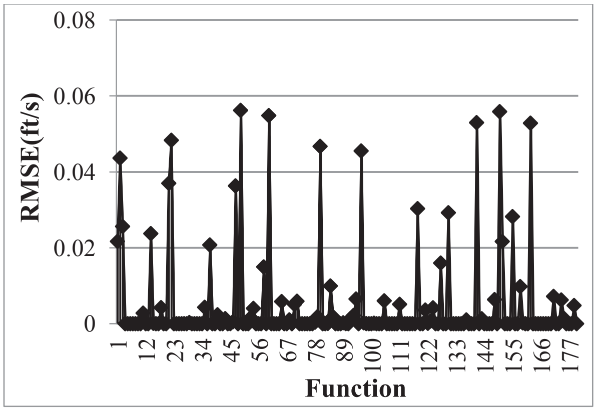


Figure 6. Velocity estimation (Ψ = 0.05).

B. *Vehicular Motion Direction*

On the basis of the above velocity interpolation methods, the velocity over continuous time can offer vehicular motion distance. In order to determine vehicular motion direction, the statistical linear regression is used to analyze the longitude and latitude of vehicular location points. For example, the first sub-function (time from 0 to 1.2 s) in Fig. 5 estimates the vehicular velocity as 39.61 ft/s, so the moving distance is 39.61 ft/s *1.2 s = 47.532 ft.

Fig. 7 gives the direction estimation of the vehicle during the short time interval (1.2 s). The closer the value of R^2, the coefficient of determination , is to 1.0, the better the linear regression fits the data. The value of R^2 is 0.9684, and the calculations about 96 vehicles' velocity sub-functions show that the range of R^2 values is between 0.9221 and 0.9865. Thus, the location linear regression for vehicular motion direction is an effectual estimation.

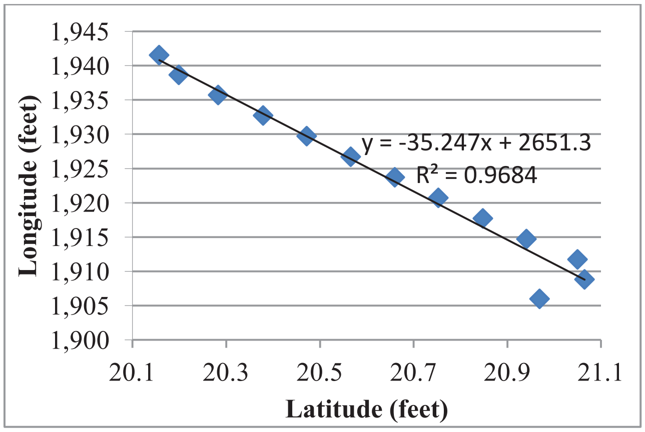


Figure 7. Location linear regression.

VI. Data Reterival

Spatio-temporal databases can offer not only distance-based static data operations as GISs do [21] but also dynamic or temporal operations. In the experimental case the test-bed spatial data represent the US 101 highway in TVICS, and the input constraint relations are:

- Car (id, x, y, t), which stores the multiple vehicular motion information at moving location (x, y) and time t. The velocities of the vehicles are described by piecewise-linear functions.
- Road (x, y), which records the static transportation network.

Below we give some example queries:

Query 6.1 Find the locations of car 1 at times 6.5 s, 8.5 s, 10.505 s, 12.51 s, and 14.49 s, respectively. This query is expressed in Datalog [7] as follows:

```
 Location (x, y): - Car (id, x, y, t), id = 1, t = 6.5.
 Location (x, y): - Car (id, x, y, t), id = 1, t = 8.5.
Location (x, y): - Car (id, x, y, t), id = 1, t = 10.505.
Location (x, y): - Car (id, x, y, t), id = 1, t = 12.51.
Location (x, y): - Car (id, x, y, t), id = 1, t = 14.49.
```

In Fig. 8 (a) the moving vehicles are shown in *the management of linear programming queries (MLPQ)* [30], and Fig. 8 (b) illustrates the result of the above query.

Query 6.2 Find the vehicular travel times for space-mean speed calculation when 12 cars pass the roadway segment (the location range of road segment is between 150 ft and 600 ft on the horizontal axis). Since the space-mean speed is computed as the length of roadway segment divided by the average time required to travel the segment [12, 22], Query 4.2 can be expressed in Datalog as follows:

```
 //query time when cars reach location 167.3 ft
 SpaceA (id, x1, t1):- Car (id, x1, y, 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).

 //query time when cars pass location 203 ft
 SpaceB (id, x3, t3):- Car (id, x3, y, 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 respectively 87.16 s and 122.17 s. The average travel time is (122.17 – 87.16)/12 = 2.9175 s, hence the space-mean speed is (230 – 167.3)/2.9175 = 21.491 ft/s (i.e. 6.55 m/s). The storage of space-mean speed is continuous in the TVICS system. Different results about space-mean speed and travel time can be retrieved from TVICS by changing the location values on the road segment.

Query 6.3 Find the spacing between car2 and car6 at time 10.5 s. The Datalog query is given below:

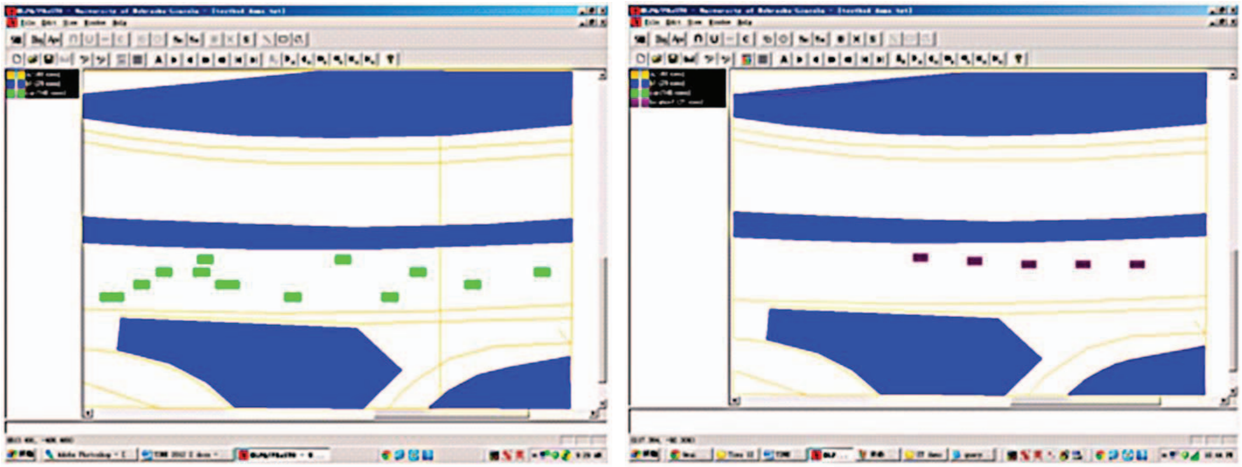
```
Spacings (t, s):- Car (2, x2, y, t), Car (6, x6, y, t),
 s + x6  – x2 = 0, t =10.5.
Spacing (t, min(s)):- Spacings (t, s).
```

The output result is 154.57 ft, and the different spacing values can be retrieved from databases by inputting different time values.

Query 6.4 Find the volume at location 610.45 ft with the time interval between 30 s and 50.9 s. The Datalog query is designed below:

```
Reach_line (id, x, t1):- Car (id, x, y, 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 inputting location x and time intervals. The TVICS outputs 5 as the above volume query result and car id numbers (including cars 6, 7, 9, 10, and 11).



(a) (b)

Figure 8. Dynamic transportation databases: (a) moving vehicles in MLPQ; (b) individual car tracking and query

VII. Discussion of Results

A. Data Completeness

Data completeness requires that database schema should include all information in the data source to meet the current and future demands of various data users. Traffic stream is observed at each spatial point within some distance interval over time, not just at one spatial point [22]. Fig. 9 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 [24-27] store just discrete traffic aggregate data, such as volume, density, headway, queue length, spacing etc. in relational databases. Aggregate data incompleteness in space and time causes the insufficient 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) [28]*.

TVICS, developed based on MLPQ, is the first traffic management system that can offer traffic aggregate data over continuous space and time. Complete traffic aggregate data are useful data sources for the description of traffic flow phenomena and for the calculation of various transportation engineering models. TVICS can be particularly advantageous in understanding highway flow breakdown, i.e. 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.

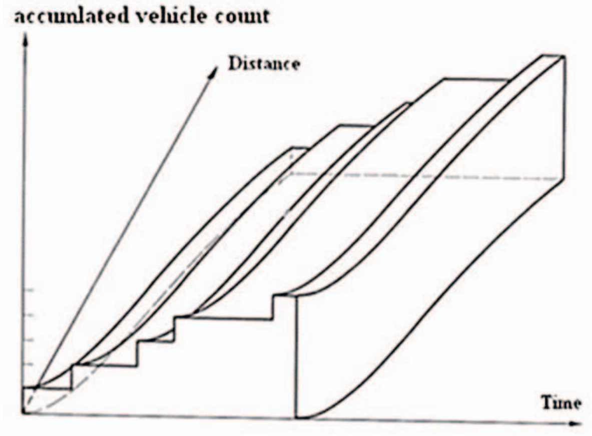


Figure 9. Traffic stream with a spatio-temporal pattern [23].

B. Data Redundancy

Discrete traffic aggregate data archives cause not only the loss of a large amount of aggregate traffic data, but also the increase of data redundancy in databases. Aggregate data are a typical data redundancy in databases, and it means that some data are stored for multiple times. Efficient data operations require data consistency and data synchronization in databases by minimizing or avoiding data redundancy [29].

In relational databases traffic data redundancy often causes data anomalies, data corruption, and data retrieval errors. For example, updating a certain volume value needs to change the values of other traffic parameters [12], 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.

By using new traffic data models, spatio-temporal databases just request the collection and storage of individual vehicular time, location, and instantaneous velocity. Traffic aggregate data can be retrieved from TVICS by database query designs. Therefore, TVICS provides traffic data archiving methods that can solve the above problems concerning traffic data redundancy.

VIII. Conclusions

Video cameras can easily collect traffic information, but storing the raw video data generally requires a huge storage space. The TVICS system is recommended to overcome the storage problem by converting traffic videos into a spatio-temporal database. Since TVICS 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 TVICS queries can search the complete continuous motion information of the moving vehicles.

As a further work, we plan to analyze different traffic situations that result in heavy traffic congestions or collisions [31]. The ultimate goal of traffic management systems is to improve the road conditions for vehicles and their drivers.

Acknowledment

The authors gratefully acknowledge Laurence R Rilett for his comments about interpolation model estimation and pointing out some related references.

Refrences

[1] H. J. Miller, “Potential Contributions of Spatial Analysis to Geographic Information Systems for Transformation”, *Geographical Analysis 31*, pp. 373-399, 1999.

[2] A. J. Butler and K. J. Dueker, “Implementing the enterprise GIS in transportation database design”, *URISA Journal*, Vol. 13, No. 1, 2001.

[3] Y. Gong, “Audio and Visual Content Summarization of a Video Program”, Chapter 10 in: *Handbook of Video Databases: Design and Applications*, B. Furht and O.

Marques (Eds.), 2003.

[4] H. D. Wactlar, M. G. Christel, Y. Gong and A. G. Hauptmann, "Lessons Learned from Building a Terabyte Digital Video Library", *IEEE Computer*, Vol. 32, pp.66-73, 1999.

[5] Y. A. Aslandogan and C. T. Yu, "Techniques and Systems for Image and Video Retrieval", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, pp.56-63, 1999.

[6] J. Agma, M. Traina and C. Traina Jr., "Similarity Search in Multimedia Databases", Chapter 29 in: *Handbook of Video Databases: Design and Applications,* B. Furht and O. Marques (Eds.), 2003.

[7] P. Kanellakis, G. Kuper, P. Z. Revesz, "Constraint Query Languages", *Journal of Computer and Systems Sciences*, Vol. 51, No. 1, pp. 26-52, 1995.

[8] Cambridge Systematics, Inc. Prepared for Federal Highway Administration. NGSIM U.S. 101 Data Analysis (7:50 a.m. to 8:05 a.m.), December 2005.

[9] A. Skabardonis and V. Alexiadis, "Traffic Data through the Berkeley Highway Laboratory", *Workshop on Traffic Modeling*, Sedona, AZ, Sept. 2005.

[10] Z. Kim, G. Gomes, R. Hranac and A. Skabardonis, "A Machine Vision System for Generating Vehicle Trajectories over Extended Freeway Segments", *the 12th World Congress on Intelligent Transportation Systems*, 2005.

[11] H. Wei, C. Feng, E. Meyer, and J. Lee, "Video-Capture-Based Approach to Extract Multiple Vehicular Trajectory Data for Traffic Modeling", *Journal of Transportation Engineering,* Vol. 131, No. 7, 2005.

[12] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering, 4th Edition,* Pearson Prentice Hall, 2011.

[13] R. H. Tao, H. Wei, Y. H. Wang, and V. P. Sisiopiku, "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. 2004.

[14] H. Wei, *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, 1999.

[15] C. X. Chen, *Data Models and Query Languages of Spatio-temporal Information*. Ph.D. Dissertation, University of California, Los Angeles, CA, 2001.

[16] H. Yin, O. Wolfson, "A Weight-Based Map Matching Method in Moving Objects Databases", the *16th International Conference on Scientific and Statistical Database Management*, 2004.

[17] J. Liu, O. Wolfson, and H. Yin, "Extracting Semantic Location from Outdoor Positioning Systems", *International Workshop on Managing Context Information and Semantics in Mobile Environments,* 2006.

[18] H. Yue, E. Jones, and P. Z. Revesz, "Local Polynomial Regression Models for Average Traffic Speed Estimation and Forecasting in Linear Constraint Databases", *17th International Symposium on Temporal Representation and Reasoning*, Paris, France, 2010.

[19] G. Recktenwald, *Numerical Methods with MATLAB: Implementation and Application*, Prentice Hall, 2000.

[20] L. Li and P. Z. Revesz, "Interpolation Methods for Spatiotemporal Geographic Data", Computers, Environment and Urban Systems, Vol. 28, No. 3, pp. 201-227, 2004.

[21] H. Yue and E. Jones, "Archiving Capability of Spatio-temporal Data in Different Highway Railroad Grade Crossing (HRGC) Databases", *Annual Intelligent Transportation System Conference*, Houston, USA, 2010.

[22] B. S. Kerner, *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory,* Springer, 2009.

[23] W. Leutzbach, *Introduction to the Theory of Traffic Flow*, Springer, Berlin, 1988.

[24] H. Al-Deek and A. Abd-Elrahman, "An Evaluation Plan for the Conceptual Design of the Florida Transportation Data Warehouse", University of Central Florida, Technical Report No. 16-50-706, 2002.

[25] J. Dahlgren, R. C. Garcia and S. Turner, *Completing the Circle: Using Archived Operation Data to Better Link Decision to Performance,* California Path Research Report No. UCB-ITS-PRR-2001-23, September 2001.

[26] H. X. Liu, R. He, Y. Tao and B. Ran, "*A Literature and Best Practices Scan: ITS Data Management and Archiving",* University of Wisconsin at Madison Technical Project No. 0092-02-11, May 2002.

[27] H. Yue and R. Yang, "Development of Intelligent Transportation Systems (ITS) and Plan of Integrated Information System", *Journal of Wuhan University of Technology*, Oct. 2005.

[28] H. Yue. "Advanced Traveler Information Inquiry, Archiving, and Decision Making System*", the 4th Chinese Oversea Student "ChunHui Cup" Entrepreneurship Competition*, Project Presentation, Dec. 2009.

[29] A. Schwinn and J. Schelp, "Data Integration Patterns", *Business Information Systems Conference*, Colorado Springs, USA, 2003.

[30] P. Z. Revesz, *Introduction to Database: From Biological to Spatio-Temporal*, Springer, New York, 2010.

[31] S. Anderson and P. Z. Revesz, "Efficient Max Count and Threshold Operators of Moving Objects", *Geoinformatica,* Vol. 13, No. 4, pp. 355-396, 2009.