Published in IET Intelligent Transport Systems
Received on 18th December 2009
Revised on 10th June 2010
doi: 10.1049/iet-its.2009.0149

Special Issue – selected papers from the 16th World Congress on ITS

ISSN 1751-956X

Estimation of the automatic vehicle identification based spatial travel time information collected in Stockholm

X. Ma[1] *H. Koutsopoulos*[2]

[1]*Centre for Traffic Research (CTR), Royal Institute of Technology, Teknikringen 72, KTH, Stockholm 10044, Sweden*
[2]*Traffic and Logistics, Department of Transportation Science, Royal Institute of Technology, Teknikringen 72, KTH, Stockholm 10044, Sweden*
E-mail: liang@kth.se

Abstract: To support the implementation of real-time traffic information systems in the Stockholm city area using automatic vehicle identification (AVI) data, a preliminary travel time analysis tool has been developed. The program can manage and analyse travel time measurements in a distributed database server where both online and historical traffic information are saved. Meanwhile, several existing travel time estimation algorithms are implemented in the travel time analysis program, and are evaluated using four months of AVI data collected in the urban streets and arterials of and near the Stockholm downtown area. The advantages and disadvantages of those algorithms are also analysed using the highly noisy travel time measurements collected under the urban context. In addition, the authors have also evaluated a common statistical median filtering approach and suggested some modifications for AVI data estimation. In general, all these algorithms have the potential to be applied for real daily travel time estimation and the statistical median filter with modifications has been suggested for historical travel time estimation in real application. Finally, the authors point out an essential problem in travel time estimation and suggest a direction that may have the potential to improve the online traffic information quality.

1 Introduction

As real-time information of traffic systems can benefit travellers in managing their trip and making optimum decisions, advanced traffic information and management systems (ATIMS) have been the kernel components of modern intelligent transportation system (ITS) technologies. Travel time is a key performance measure on road link and network, and can be simply defined as the time necessary to traverse a route between any two points within a road transportation network. In reality, travel times between any pair of origin and destination (OD) are subject to fluctuations because of stochastic variation of traffic characteristics and interactions between demand and supply on road networks. Thus, estimation and prediction of travel time become an essential part for any ATIMS system especially because the urban traffic control centre aims to affect traffic assignment through real-time information and guidance in order to decrease congestion delay on a road network.

Approaches to estimate traffic states and predict travel times on roads vary significantly because of the differences in traffic data collection methods and applied computational models. The data collection methods can be based on different technologies. Spot measurement using loop detectors or other fixed sensors is one of the most widely used methods in traffic data collection, and it can provide large amount of measurement data on flow and speed at certain locations. Travel times are, however, obtained indirectly from pure spot measurements, since analytical models have to be built to describe traffic flow propagation across road networks. Considerable research efforts [1–3] have been spent on traffic information estimation and prediction using datasets from fixed sensors. Although the methods have shown to be applicable on many freeway applications, prediction accuracy and validation under congested traffic conditions still need further study. In addition, the method has been limited to motorways where traffic dynamics models are relatively easier to be justified. When considering traffic flow of urban arterials, the interrupted nature makes travel time estimation and prediction much more challenging: travel time is not only related to road capacity and traffic flow itself but also involved with factors such as traffic signal timings.

With the current technical advances in ITS systems, a number of direct travel time data collection methods have been developed and started to be implemented, for example the GPS data system and automated vehicle identification (AVI) system. The obvious advantage of these methods is the convenience to obtain travel time data, which can be directly applied for traffic forecast and traffic information systems. In application of [4], GPS data have been combined with loop detector data to predict travel time on highways near the central city area of San Antonio in the USA. An advanced neural network was trained using experiment data to store the relationship between traffic flow characteristics captured by loop detectors and link travel time measured by the GPS system. Accurate link travel time prediction is reported by cross validation. However, the study is only based on relatively few travel time data from experiment. In practice, the GPS data collected from real network are scarce in most cities. On the other hand, travel time data can be continuously observed by AVI system on road networks, although implementation of such a system is expensive and requires large amount of work concerning installation and adjustment of both hardware and software. Technically, there are mainly two types of AVI systems: electronic tag-based and image processing (license matching)-based systems. Both kinds of systems have been mostly applied in ITS applications on motorways. Unlike travel times on motorways, measurements of travel times in urban areas using license-plate matching are highly fluctuating because of traffic congestion, effects of traffic signals, interactions with other objects and considerable noise. The data also usually include large number of outliers. For example, outliers are often generated by the fact that some vehicles may stop briefly between measuring stations for various reasons (e.g. take/drop passengers, shopping, loading/unloading etc.). Such outliers should be identified and properly treated.

2 AVI travel time estimation

2.1 Problem statement

As mentioned, the AVI system can help to obtain the timestamps that a vehicle is passing any observing station. In Fig. 1, we simply illustrate the basic travel time estimation problem. If a vehicle travels from station A to B with the entry time t_A and the exit time t_B, the travel time is thus derived as the elapsed time between them, that is

$$T_{AB}(t) = t_B - t_A \tag{1}$$

where $T_{AB}(t)$ is the travel time from A to B at certain time t. Since one objective of the AVI-based traffic information system is to reflect real-time traffic conditions and inform other travellers and help them choose a proper route or departure time, (1) represents, therefore, the travel time information at the entry time $t = t_A$. However, as it is obviously not possible to know the exact travel time unless a vehicle arrives, the real AVI system normally publishes the travel time estimated according to exits, that is, $t = t_B$,

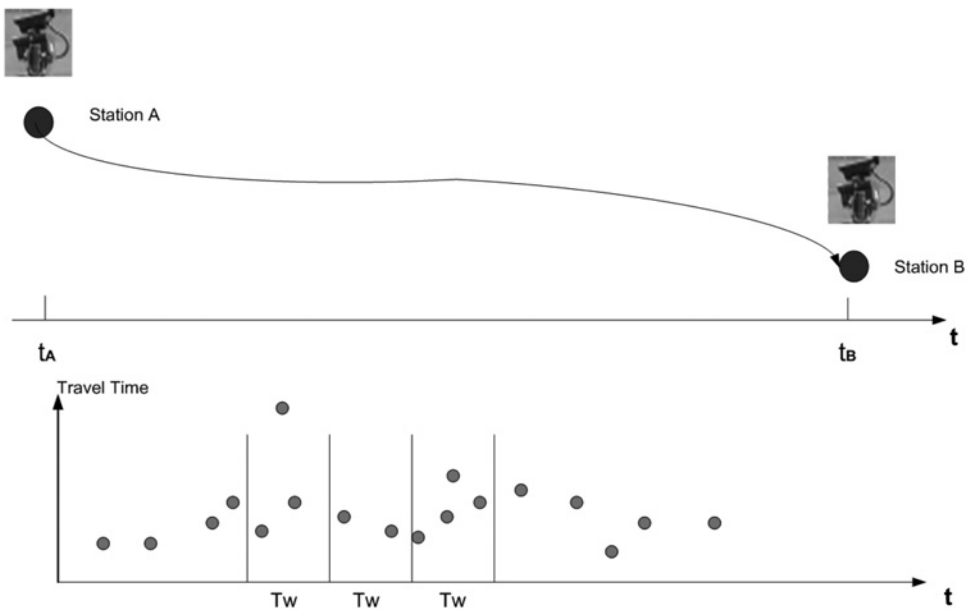

Figure 1 *Illustration of the AVI-based travel time estimation problem*

doi: 10.1049/iet-its.2009.0149

as current real-time information. This means there is usually a delay within the system, which equals the amount of the travel time itself. In the literature of travel time estimation, this delay has not been found to be treated explicitly. For a real AVI-based system, travel time measurements are obtained randomly whenever there are vehicle arrivals. However, a digital system requires reporting travel time every certain time interval. Thus, one of the central research questions is how to aggregate the travel time information that is measured according to vehicle arrivals into certain time interval, that is, T_W, especially when there are noises and errors presented in the system.

2.2 Existing approaches

In literature, a wide spectrum of methods has been developed for travel time estimation and predictions, especially prediction because of the fact that we illustrate in the previous section. However, we will focus on several existing algorithms that were developed to estimate link travel times using data of the AVI-based systems.

2.2.1 TransGuide and TranStar: One of the earliest methods on real-time travel time estimation appears in a commercial ATIMS system, called TransGuide [5], developed at San Antonio, TX in the late 1990s. Accordingly, the TransGuide system can collect online travel times using AVI tag readers, through which arrivals of vehicles with AVI tags installed can be detected, and a rolling average filtering algorithm is applied in the system to estimate online travel time per minute or every certain minutes based on previous measurements in a defined time window. The algorithm can be mathematically described as follows

$$T_{AB}(t) = \frac{\sum_i (t_{B,i} - t_{A,i})}{N(S_{AB}(t))}, \quad \text{where } i \in S_{AB}(t) \tag{2}$$

$$S_{AB}(t) \equiv \{k | t - T_W < t_{B,k} < t\} \cap \left\{ m \left| \left| \frac{t_{B,m} - t_{A,m} - T_{AB}(t - Tw)}{T_{AB}(t - Tw)} \right| < l \right. \right\} \tag{3}$$

where $T_{AB}(t)$ is the estimated travel time from A to B at time t and $t_{A,I}$ and $t_{B,i}$ are the time stamp that the vehicle i is detected at station A and B, respectively; $N(\cdot)$ is the number of vehicles of the set of $S_{AB}(t)$, which is definedby a dynamic set of vehicles whose arrival times are within the aggregate time interval and their travel times (from A to B) are within a percentage limitation (so that outliers can be removed), represented by l. (The definition is modified from the corresponding reference to make it mathematically clearer.) In the real operation of the TransGuide system, the aggregate time interval T_w is set to 2 min, and a threshold of $l = 20\%$ is used to determine outliers or errors, which means if any measured travel time deviates by more than 20% from the previously estimated travel time, it is deemed as an outlier and will be eliminated from the travel time estimation in that interval. TranStar is another existing algorithm developed by the Southwest Research Institute (SRI) for the auto-toll system in Houston. It takes a similar approach as the algorithm in the TransGuide system. The main difference is that travel time estimation is conducted each time the new travel time information is obtained because of an arrival of a vehicle with an electronic tag. l is also set to 20% but smaller aggregation windows of 30 s and 1 min are applied. Both TransGuide and TranStar use electronic-tag reading-based AVI systems. Such systems have an advantage in accurately identifying a passing vehicle but the travel time measurement frequency is statistically determined by the level of the market penetration of electronic tag. In principle, the higher the market penetration is, the better the estimation accuracy can be achieved. Thus, at the beginning of such system deployment, low sampling rate is also an issue that affects travel time estimation accuracy.

2.2.2 Dion algorithm: Owing to innate limitations of the TransGuide/TranStar algorithms for travel time estimation that we will discuss later, Dion and Rakha [6] proposed a more complicated exponential low-pass filtering algorithm based on the following factors:

- Expected average travel time and variability in the future time interval.
- Number of consecutive intervals without any readings since the last recorded trip time.
- Number of consecutive data points either above or below the validity range.
- Variability in travel times within an analysis interval.

Mathematically, the filtering algorithm can be represented as follows

$$T_{AB}(k) = \begin{cases} \exp((1-\alpha)\log(T_{AB}(k-1)) + \alpha \log(\tilde{T}_{AB}(k))), & n_{v,k} > 0 \\ T_{AB}(k-1), & n_{v,k} = 0 \end{cases} \tag{4}$$

$$\sigma^2_{AB}(k) = \begin{cases} (1-\alpha) \cdot \sigma^2_{AB}(k-1)) + \alpha \cdot \tilde{\sigma}^2_{AB}(k), & n_{v,k} > 1 \\ \sigma^2_{AB}(k-1), & n_{v,k} = 0, 1 \end{cases} \tag{5}$$

$$S_{AB}(k) \equiv \{h | t - T_W < t_{B,h} < t\} \cap \{m | T^{\min}_{AB}(k) < t_{B,m} - t_{A,m} < T^{\min}_{AB}(k)\} \tag{6}$$

$$T^{\min}_{AB}(k) = \exp(T_{AB}(k) - n_\sigma(k)\sigma_{AB}(k)) \tag{7}$$

$$T^{\max}_{AB}(k) = \exp(T_{AB}(k) + n_\sigma(k)\sigma_{AB}(k)) \tag{8}$$

$$\tilde{T}_{AB}(k) = \sum_i (t_{B,i} - t_{A,i})/n_v(k), \quad i \in S_{AB}(k-1) \tag{9}$$

IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 4, pp. 298–306

doi: 10.1049/iet-its.2009.0149

$$\tilde{\sigma}_{\mathrm{AB}}^2(k)=\begin{cases}0, & n_v(k)=0\\ (\log(t_{\mathrm{B},i}-t_{\mathrm{A},i})-\log \tilde{T}_{\mathrm{AB}}(k))^2, & n_v(k)=1\\ \sum_i (\log(t_{\mathrm{B},i}-t_{\mathrm{A},i})-\log \tilde{T}_{\mathrm{AB}}(k))^2/n_v(k), & n_v(k)>1\end{cases} \tag{10}$$

$$\alpha=1-(1-\beta)^{n_v(k)} \tag{11}$$

$$n_\sigma(k)=\lambda+\lambda[1-(1-\beta_\sigma)^{n_0(k)}] \tag{12}$$

Equations (4) and (5) describe how the estimated (smoothed) average travel time $T_{\mathrm{AB}}(k)$ and the variance of the logarithm of travel times $\sigma_{\mathrm{AB}}^2(k)$ are derived from the previous estimated values, $T_{\mathrm{AB}}(k-1)$ and $\sigma_{\mathrm{AB}}^2(k-1)$, and the expected average travel time and variance within the current sampling interval, $\tilde{T}_{\mathrm{AB}}(k)$ and $\tilde{\sigma}_{\mathrm{AB}}^2(k)$. Equations (6)–(8) define an adaptive window set for valid observation range. To reflect the fact that travel times are skewed towards larger values, a basic assumption here is that travel times statistically follow a log-normal distribution. Equations (9) and (10) describe the basic statistical method for computing the expected average travel time and its variance using valid observations within the current time interval. Equation (11) demonstrates how the smoothing factor is determined, that is, it depends on a decay factor β, and the number of valid observations within the sampling interval $n_v(k)$. The final equation shows that the boundaries to determine valid observations for each time interval expand outwards in case of zero valid observations in previous time intervals; $n_0(k)$ is the number of previous time intervals with no valid observations and it is mainly designed to deal with low sampling rate and λ is a factor set by users to determine how large boundaries are given in the normal case (e.g. 1.96 corresponding to 95% confidence intervals).

3 System and data analysis

3.1 AVI data in Stockholm

In Stockholm, license plate recognition-based AVI system was first deployed for the congestion charging purpose. At the same time, a similar image processing-based technology was evaluated for development of a real-time travel time data collection system. The data are planned to be used in the Stockholm Urban Traffic Control (UTC) centre, and to provide travellers with real-time travel time information on a number of major routes through the city, both via variable message signs (VMS) and through Internet (http://www.trafiken.nu). Recently, license plate matching-based AVI system has already started to be deployed as a main system to provide real-time travel time information for major arterials at the Stockholm downtown area. The system is planned to spread over most of the major arterials in the big Stockholm area. Before the formal system deployment, real-time travel times had been collected over 40 routes in the preliminary tests in Stockholm since 2006 for evaluation purpose. Based on the license number matching between two observation locations within the network, measurements are processed into a list of column variables: route ID, time stamp, coded license number, travel time, etc and saved in the data log files. Each file contains one-day raw travel time data for certain links. We have in total two month data to construct a database. Unlike travel times on motorways, measurements in city areas using license-plate matching-based AVI system are highly fluctuating because of varying levels of congestion, and moreover produce a considerable portion of noise, or errors. The main source of the noise is from vehicles that pass the starting point A of the measured stretch, park shortly for different reasons (e.g. shopping, loading/unloading etc.) and continue their journey passing the point B considerably later than they usually would. Obviously, these data should be treated by the real-time travel time estimation and prediction algorithm, but without separate information it may be hard to judge which measured vehicles are delayed because of traffic conditions, and which have parked in the meantime. Other sources of noise are buses stopping at bus stops, taxis picking up/letting off passengers etc.

3.2 Development of the travel time analysis platform

To be able to analyse obtained travel time data, a computing platform, TAO, is developed with a distributed MySQL database at the Linux server side and an analytical program running at the client end. In the database server, collected travel time information is stored in a database that the detailed information of the road network is also presented, that is, the nodes and links/streets. Similar as other scientific and engineering fields, the large amount of traffic observations obtained from various detectors are normally stored in the operational databases. In principle, such databases are designed for daily common operations, for example, indexing data, searching for particular records and advanced queries. They formulate online transaction processing (OLTP) systems. However, OLTP systems are not appropriate for online analytical processing (OLAP) purposes, in which large amount of multidimensional data need to be manipulated in an aggregated level. Therefore data warehouse, served as a main role of data analysis and decision making, has been introduced as a repository of summarised and subject-oriented information, aimed at supporting further data mining and knowledge discovery [7]. Data cube, defined by dimensions and facts, is the basic model for multidimensional data marts and warehouses. In our ongoing work with the AVI database, we have extended the daily AVI data transaction table into a data cube model in Fig. 2 so that historical travel time data can be combined with real-time data for advanced statistical prediction modelling. For example, we can consider different factors for daily travel time pattern such as weekday, weather etc. The data cube model provide a means to extend the previous table into efficient analysis forms. In addition, traffic data nowadays come from difference types of ITS sensors and Telematic systems

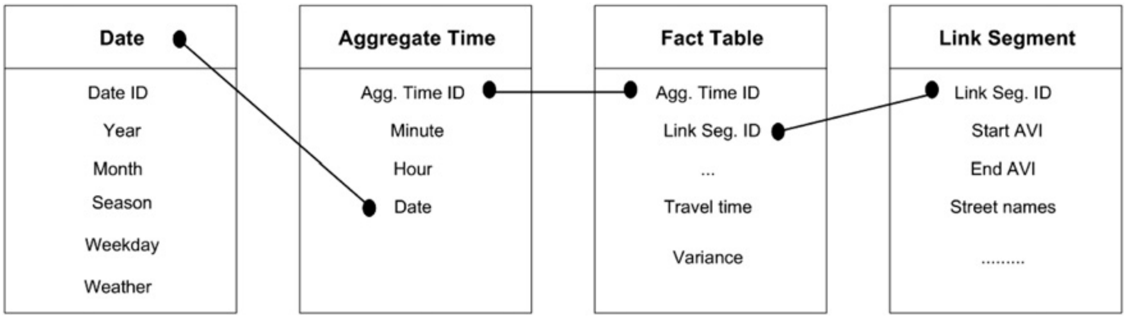

Figure 2 *Snow-flake scheme data cube model for the travel time data warehouse*

including AVI, loop detectors, road-side microwaves etc. Such a database design will provide the system with the potential ability to extend and use other sources of data.

After the collected travel time information is built into the historical database, different travel time estimation algorithms mentioned before are also implemented in the client program. The client tool is programmed using an object-oriented language, Python, and has mainly two functionalities: remote database maintenance and evaluation of the travel time estimation and prediction algorithms. The maintenance function makes it possible to append/reformat/delete records of the historical database. The algorithm implementation in the system is rather convenient so that it provides us a flexible environment to evaluate the performance of different methods and to simulate both offline and online travel time estimation and prediction cases. Recently, we have also integrated the Google Map into the graphical user interface (GUI) (Fig. 3) to formulate more intuitive database query interfaces so that users can handle travel time information in an easy manner. In addition, a web-based system has been developed for a natural extension of the current tool so that information can be accessed from the Internet. However, the system is still in an experiment mode and will provide first-hand experience for the final commercial system operated by 'Trafik Stockholm'.

3.3 Evaluation of the estimation approaches

In this section, we will mainly discuss about our evaluation of a number of existing algorithms such as TransGuide, TranStar and Dion algorithms introduced in the last section using the data that we collected in the urban context. In addition, the median travel time computed within each aggregated time interval is also used as an index to compare with other approaches as it is a simple and reliable approach in many signal processing applications [8]. Given the lack of out-of-sample validation data, justification of the estimation result is qualitative based on whether estimated travel time profile can capture the daily and local trends depicted by the majority of collected data (not including the outliers).

TranGuide and TranStar algorithms are previously criticised for their abilities to capture sudden changes in travel time data and their robustness in estimation. In our

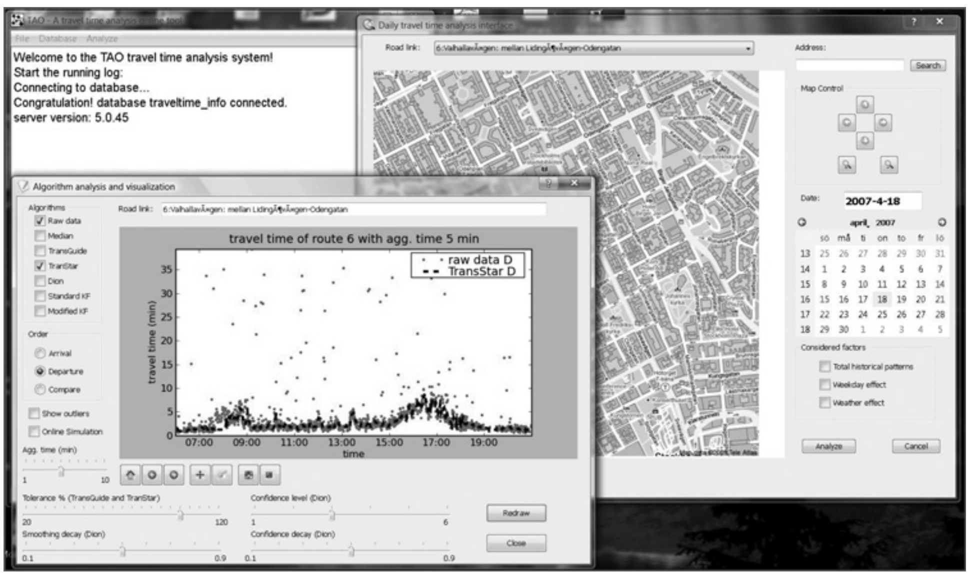

Figure 3 *Screenshot of the interface of the travel time analysis tool, TAO*

IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 4, pp. 298–306
doi: 10.1049/iet-its.2009.0149

numerical experiments, TransGuide algorithm does not produce proper estimation when travel time sequence includes sudden variation if tolerance level is only set to 20% as recommended in the implementation in the USA. This is illustrated by an estimation example of a common dataset (two periods of rush hours) in the first graph of Fig. 4. However, by raising the tolerance level to 100%, a realistic estimation can be achieved. In addition, by increasing the aggregation time to 5 min, we obtained a smoother estimation in the second graph of Fig. 4. Using the same dataset, we evaluate the TranStar algorithm with same parameter combination as in the TransGuide algorithm. It is shown that the TranStar algorithm has a similar problem as the TransGuide algorithm when the tolerance level is set to as low as 20%. In Fig. 4, increasing the tolerance level to 100% and aggregation time to 5 min gives a justifiable and smoother estimation, similar to the performance of the TransGuide algorithm. Nevertheless, the TranStar algorithm, which estimated travel time whenever there is an observation, is computationally more expensive than the TransGuide algorithm since the average value of vehicle arrivals or traffic flow rate is normally much higher than one vehicle per minute. So it is not practical as an online approach. To evaluate the robustness of both methods, we have tested the algorithms using many other data sequences and the results show that both algorithms can give reasonable estimation using a tolerance level of 100% and the aggregation time of 5 min. This is quite

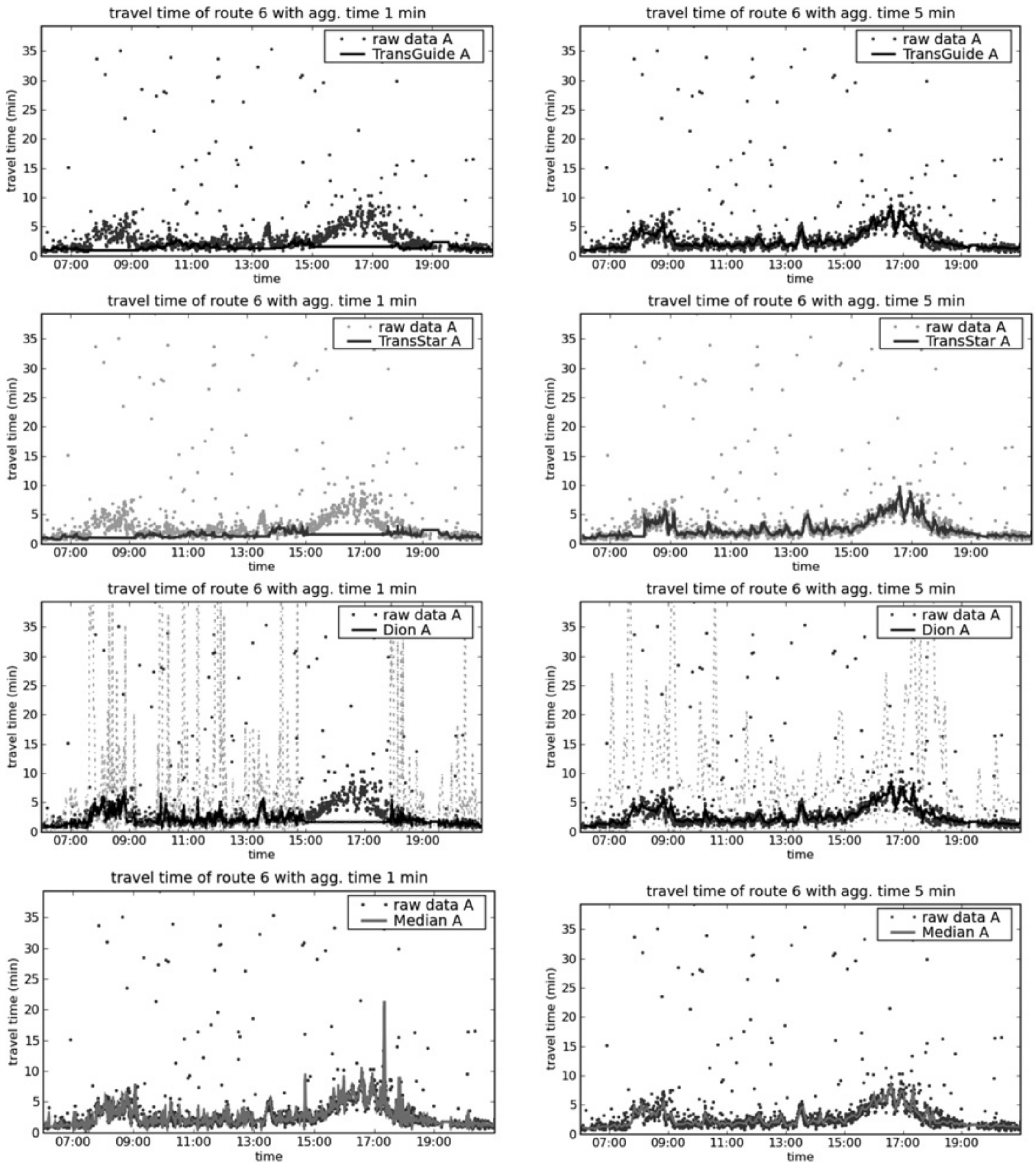

Figure 4 *Travel time estimation using the TransGuide, TranStar, Dion and Median algorithms with different aggregation times*

doi: 10.1049/iet-its.2009.0149

different from the parameters that were recommended in the original development of the TransGuide and TranStar algorithms, partially because there are a large amount of outliers involved in our AVI-based measurements on urban roads in Stockholm.

Besides the two algorithms mentioned above, we also evaluated the more complicated Dion algorithm and a simple approach using the median filter. Based on an assumption of log-normal distribution of travel times, the Dion algorithm applies an adaptive low-pass filter on both mean and variance of the estimated travel time data according to (4) and (5). One essential idea is to determine a valid window every certain time interval according to the travel time distribution, that is, when a data point in the current time interval is outside the boundaries (confidence intervals) determined in the previous time interval, it will be rejected as an outlier, and not used in the travel time estimation of the current interval. The approach seems to have a well-established analytical basis and, unsurprisingly, the method was shown more robust than TransGuide and TranStar for travel time estimation on freeways [6]. However, the general performance of the filter may suffer reduction in our case because of the high noise power in our travel time measurement on urban streets. In our experiment, the aggregation time definitely has an impact on the result

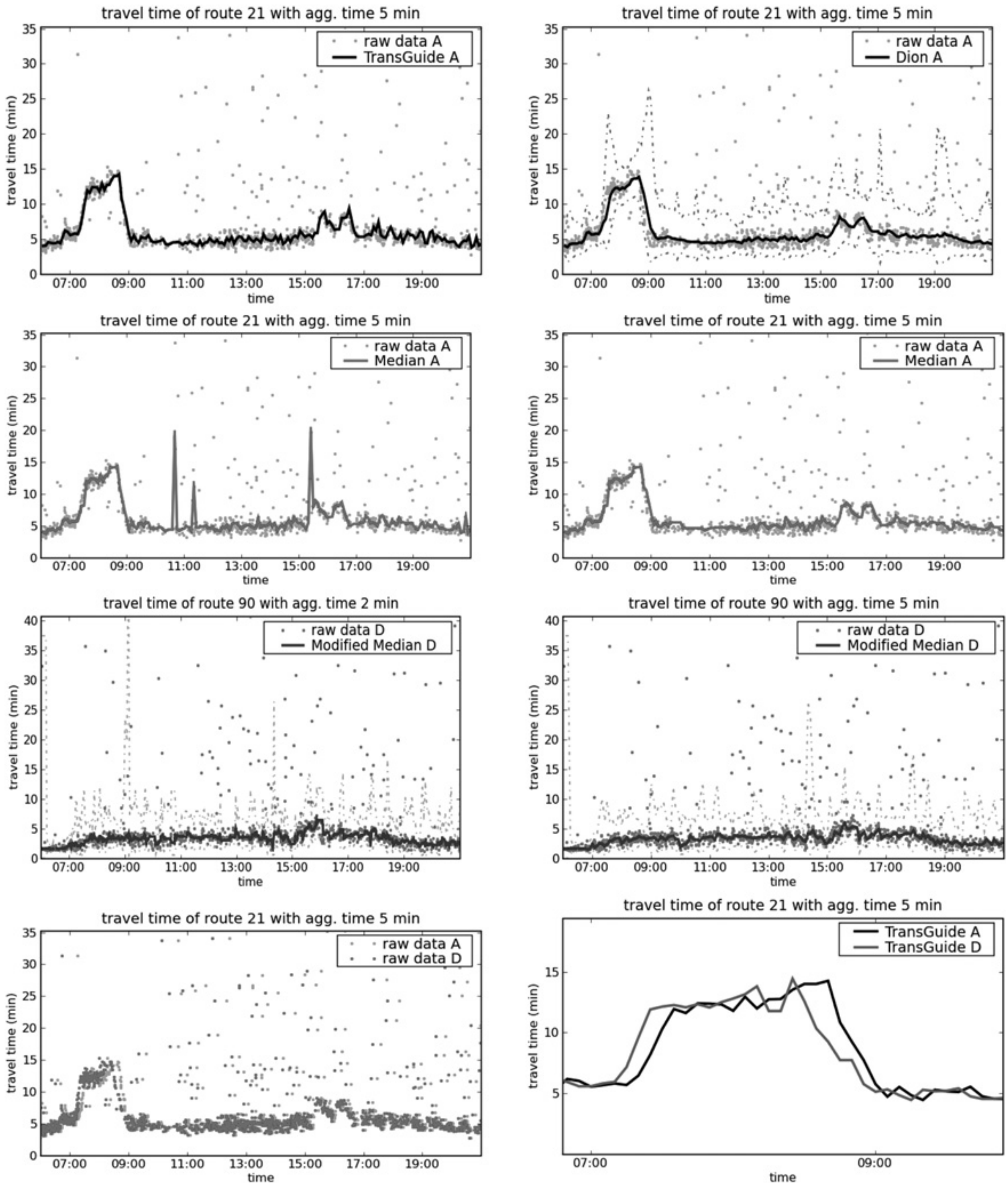

Figure 5 *Travel time estimation on another dataset and the estimation delay problem*

doi: 10.1049/iet-its.2009.0149

such as in Fig. 4, and similar to other approaches, 5 min is a proper time for information aggregation. The determination of other parameters, such as λ, β and β_σ, is rather tricky since they normally have a combined non-linear effect. For the dataset that we evaluated in Fig. 4, a parameter set of $\lambda = 4$, $\beta = 0.4$ and $\beta_\sigma = 0.5$ gives the result as in the graph. However, it seems very difficult to find a commonly 'best-fitted' parameter set for different roads without conducting a global search of the parameter space, although we find that $\lambda = 4$, $\beta = 0.1$ and $\beta_\sigma = 0.5$ will produce well-justified estimation result in all daily data sequences that we tested. On the other hand, we have also evaluated the relatively simple median filter approach. In the last two graphs of Fig. 4, the estimation result is generally good with an aggregation time of 5 min whereas small aggregation time of 1 min brings more fluctuations in the estimated sequence. In Fig. 5, we show the estimation result of travel times in another single day with the parameters exactly the same as those used to produce Fig. 4. Both TransGuide and Dion algorithms can reliably produce travel time estimation. One issue with the median filter is that it is easy to be corrupted by the outliers. Although it is possible to exclude outliers using historical travel time profiles, we have added a simple mechanism to increase the reliability of the method against the noise disruption, that is, we set a threshold of the minimal number of observations at each time interval; if the number cannot be fulfilled at certain time interval, the data from last intervals will be included to estimate the median until a maximal number of intervals are reached. The pseudo-code in Fig. 6 illustrates the improvement.

Besides, the median filtering algorithm has been further strengthened with the ability to estimate uncertainty of travel times in each aggregated interval based on a similar assumption as the Dion algorithm [6], which assumes travel times fulfilling the log-normal distribution

$$T_{\mathrm{AB}}(k) = \begin{cases} \exp(\mathrm{median}(TT\log_{\mathrm{AB}}(k))), & n_k \geq N_{\min} \\ T_{\mathrm{AB}}(k-1), & \text{else} \end{cases} \quad (13)$$

$$\mu^2_{\mathrm{AB}}(k) = \begin{cases} \sum_j \kappa(k,j)^2/(n_k - 1), & n_k \geq N_{\min} \\ \mu^2_{\mathrm{AB}}(k-1), & \text{else} \end{cases} \quad (14)$$

$$TT\log_{\mathrm{AB}}(k) = \{\log(\hat{T}_{\mathrm{AB}}(k,1)), \ldots, \log(\hat{T}_{\mathrm{AB}}(k,j)), \ldots, \log(\hat{T}_{\mathrm{AB}}(k,n_k))\}, \quad j \in S_{\mathrm{AB}}(k) \quad (15)$$

```
joint_vector=current_TT_sequence
for j = 1 to max_N_Intervals:
  joint_vector=joint_vector+TT_sequence[k-j]
  if length(joint_vector)>=min_N_Points:
    break
if length(joint_vector)<min_N_Points:
  TT_estimation[k]=TT_estimation[k-1]
else:
  TT_estimation[k]=compute_median(joint_vector)
```

Figure 6 *Pseudo-code for the improved median filtering of travel times*

$$\kappa(k,j) = \log(\hat{T}_{\mathrm{AB}}(k,j)) - \mathrm{median}(TT\log_{\mathrm{AB}}(k)), \quad j \in S_{\mathrm{AB}}(k) \quad (16)$$

$$S_{\mathrm{AB}}(k) \equiv \{m | T^{\min}_{\mathrm{AB}}(k) \leq t_{\mathrm{B},m} - t_{\mathrm{A},m} < T^{\max}_{\mathrm{AB}}(k)\} \quad (17)$$

$$T^{\min}_{\mathrm{AB}}(k) = \exp(T_{\mathrm{AB}}(k) - \lambda\mu_{\mathrm{AB}}(k)) \quad (18)$$

$$T^{\max}_{\mathrm{AB}}(k) = \exp(T_{\mathrm{AB}}(k) + \lambda(k)\cdot\mu_{\mathrm{AB}}(k)) \quad (19)$$

where $\hat{T}_{\mathrm{AB}}(k,j)$ is the travel time measurement at the interval k that is not outlier, and λ is the value corresponding to the confidence interval; the main difference here from the Dion algorithm is that median becomes the operator for estimation of travel time and involved with the computation of its uncertainty. The introduction of numeric boundaries at each interval helps exclude outlier data samples and therefore make the median filter less likely to be corrupted by noise. With all these modifications, the median filter becomes quite reliable and works for all test data that we have, for example, the data in Fig. 5, when the minimum number of the valid travel time observations is $N_{\min} = 6$, the maximal number of intervals to memorise is 2 and the parameter representing the confidence level is set to be 4.

4 Summary and discussion

In this paper, we have introduced our earlier effort of developing a travel time analysis platform based on a database server and an application client tool. Although the data analysis system is only initialised for estimation and prediction of the AVI travel time data, it can be extended to incorporate other types or sources of traffic data, for example, the traffic flow speed and flow rate data from AVI itself and loop detectors. Three existing estimation algorithms are implemented in the current analytical tool. We have also developed a Kalman filter-based approach [9] and are working on an online algorithm considering the system delay problem. The three existing algorithms were initially developed for travel time estimation on freeways, but where we apply is on urban streets and arterials and there is a big portion of noise involved in the data because of temporary or long-time parking.

From data analysis, we found that the parameters suggested for freeways are obviously not fitted in the urban context. However, using the data analysis tool, we are able to find parameters that make all three algorithms produce reasonable online travel time estimations, although our justification of acceptable estimation is qualitative and development of a quantitative measure is necessary for more concrete comparison of algorithm performance. In addition, a common statistical approach, the median filter, is also introduced as one potential travel time estimation algorithm. In particular, we show that the general median filter can have good performance by modifications such as parameterisation of the minimum number of observations in the time interval, introduction of smoothing memory and justification of outliers according to the estimation of uncertainty boundaries. Although all test algorithms work for travel time estimation, it is difficult to conclude which algorithm has the best fitness for the travel time observations without a global parameter search. Especially, methods like the Dion algorithm require extensive sensitivity tests on the coefficients. At the moment, the statistical median filter is recommended for establishing historical travel time databases since it is simpler than other approaches and equally robust. In addition, 5 min has been suggested as the aggregation time for storing historical patterns in real application.

One important issue that we want to discuss is the system estimation delay for travel time observations. As illustrated in Fig. 1 at the beginning of this paper, traffic information system will have travel time observations only if vehicle arrives. This means that the observed travel time is the travel time for the time instant at the vehicle departure at station A, and a delay as much as the travel time itself exists in the system. In the last two graphs of Fig. 5, we show the travel times according to their departures and arrivals. The left graph shows the raw data according to departure times (D) and arrival times (A). The right graph shows the estimations according to the two time points. Obviously the travel time estimation based on vehicle arrivals is delayed. This indicates a potential problem that the inaccurate travel time will be disseminated to public particularly when congestion builds up or dissipates off. Meanwhile, we have to consider this problem when determining the geometrical distribution of AVI stations. In general, if such a delay is considered, the algorithms mentioned before will be more appropriate for offline travel time estimation. To fulfill real-time travel time estimation, we have to, in fact, deal with online travel time prediction.

5 Acknowledgments

The authors would like to acknowledge the financial support of the spatial travel time prediction (STTP) project by the Traffic Administration Office of Stockholm (Trafikkontoret Stockholm Stad).

6 References

[1] NAM D.H., DREW D.R.: 'Traffic dynamics: method for estimating freeway travel times in real time from flow measurements', *J. Transp. Eng.*, 1998, **122**, pp. 185–191

[2] HOLT R.B., SMITH B.L., PARK B.: 'An investigation of travel time estimation based on point sensors'. Technical report Smart Travel Laboratory, Charlottesville, VA, August 2003

[3] ZHANG W.: 'Freeway travel time estimation based on spot speed measurement'. PhD thesis, Virginia Polytechnic Institute and State University, USA, 2006

[4] KISGYÖRGY L., RILETT L.R.: 'Travel time prediction by advanced neural network', *Period. Polytech. Ser. Civ. Eng.*, 2002, **46**, (1), pp. 15–32

[5] S.R.I.: 'Transguide model deployment report'. Technical report, Southwest Research Institute, San Antonio, TX 78228-0510, 2000

[6] DION F., RAKHA H.: 'Estimating dynamic roadway travel times using automatic vehicle identification data for low sampling rates', *Transp. Res. Part B*, 2006, **40**, (3), pp. 745–766

[7] HAN J., KAMBER M.: 'Data mining: concepts and techniques' (Morgan Kaufmann, 2001)

[8] YIN L., YANG R., GABBOUJ M., NEUVO Y.: 'Weighted median filters: a tutorial', *IEEE Trans. Circuit Syst. II*, 1996, **43**, (3), pp. 157–192

[9] MA X., KOUTSOPOULOS H.N.: 'A new approach for real-time travel time estimation using distorted automatic vehicle identification data'. IEEE Int. Conf. on Intelligent Transportation Systems, October 2008

doi: 10.1049/iet-its.2009.0149