2010 13th International IEEE
Annual Conference on Intelligent Transportation Systems
Madeira Island, Portugal, September 19-22, 2010

Two Fast Implementations of the Adaptive Smoothing Method Used in Highway Traffic State Estimation

Thomas Schreiter, Hans van Lint, Martin Treiber and Serge Hoogendoorn

Abstract— **Freeway traffic state estimation is crucial for dynamic traffic management (DTM), Advanced Traveler Information Systems (ATIS) and highway performance analyses. Raw data collected by dual-loop detectors or GPS devices provide information about flow and speed at points in space and time. However, these observations are noisy and incomplete. The Adaptive Smoothing Method (ASM) estimates the traffic state between the observation points and reduces the noise inherent to observations. Current implementations of the ASM apply its model in a straight-forward manner, which leads to high computation times. In this paper, two new implementations are developed that drastically reduce the computation time while preserving the estimation quality.**

In the first implementation, the ASM is discretized to apply the cross-correlation. This is based on matrix operations, which are efficiently implemented and fast in execution. In the second implementation, the ASM is reformulated to apply the Fast Fourier Transform (FFT). The FFT, too, is based on fast matrix operations. These two new implementations are sequential programs, containing no loops. Experiments with a setup used in practical applications and real data show computation times of just a few seconds. These are computation time improvements of two orders of magnitude. The rapid computation of the traffic state makes the ASM with the proposed implementations applicable for real-time applications.

I. Introduction

Dynamic Traffic Management (DTM) and Advanced Traveler Information Systems (ATIS) require some form of the current traffic state as input. An accurate estimate of the current state of traffic is also the essential starting point of traffic predictors, which in turn are used in DTM and ATIS applications such as model-predictive control. Furthermore, traffic authorities need to evaluate the historic performance of their roads.

Usually, the traffic state cannot be directly measured everywhere but needs to be estimated from incomplete, noisy and local traffic data. Commonly, volumes or average vehicle speeds are measured at certain locations on the highway, for example by dual-loop detectors or GPS devices. To estimate the total traffic state from these point measurements, interpolation between sensors is necessary.

In the current state of practice, often simple (orthogonal) interpolation methods are used to estimate traffic quantities at locations and instants "between" observations. This is for example what happens in speed-based travel time estimation methods [1]. These simple methods assume that the behavior of traffic is always equal in all traffic conditions. In reality, however, the direction in which information travels through the network differs depending on traffic conditions: in free flow, information travels downstream, but in congested conditions, information travels upstream. Therefore, these simple methods exhibit significant bias [2].

One way to take the information direction into account is through the kinematic wave theory, or the LWR model [3], [4]. Model-driven approaches like Fastlane [5] and Renaissance [6] fuse the traffic state predicted by the model with the measurements taken by traffic sensors. Also higher-order models are used to describe the traffic behavior in more detail [7], [8]. A disadvantage of these approaches is that they have to be calibrated with a huge number of parameters.

Another kind of traffic estimators are data-driven approaches, which do not predict the traffic state, but only take the measurements into account. An important example of this approach is the Adaptive Smoothing Method (ASM) [9]. There, the measurements are smoothed in the space-time plane along the characteristic wave speed of the present traffic regime. The advantage of the ASM is that it is calibrated with only a few parameters.

There are several extensions of the ASM that expand the sources of raw data from loop detectors to floating-car data [10], [11]. The ASM is also part of a traffic state estimation framework, which additionally estimates the characteristic wave speeds [12] and reduces the speed bias inherent to loop detector data [13].

Current implementations of the ASM compute the traffic state of road stretches of a few kilometers length within a few minutes (on current state-of-the-art laptops). The computation time increases, however, if the traffic state of a large road stretch, of a long time period or with a fine-grained resolution has to be computed.

Therefore, in this paper, two fast implementations of the Adaptive Smoothing Method are proposed. The computation time to filter the traffic state is reduced, while the filter quality is preserved. One of these implementation uses the cross-correlation, which originates from signal processing. The conventional ASM implementation is reformulated to apply the cross-correlation, which significantly reduces the computation time in Matlab. The other of these implementations is based on the Fast Fourier Transform (FFT), which was also developed in signal theory.

This paper is structured as follows. The ASM and the two proposed implementations are presented in Section II. The implementations proposed are tested with data from

T. Schreiter, H. van Lint and S. Hoogendoorn are with the department Transport & Planning, Faculty of Civil Engineering, Delft University of Technology, 2600 GA Delft, The Netherlands, Email:{t.schreiter, j.w.c.vanlint, s.p.hoogendoorn}@tudelft.nl. M. Treiber is with the Institute for Transport & Economics, Technische Universität Dresden, Falkenbrunnen, Würzburger Str. 35, 01187 Dresden, Germany, Email: {treiber,kesting}@vwi.tu-dresden.de.

978-1-4244-7659-6/10/$26.00 ©2010 IEEE

Dutch freeways (Section III). The results show improvements of computation time up to two orders of magnitude (Section IV). The proposed implementations can therefore easily replace the conventional implementation in practical applications (Section V).

II. Methodology

The focus of this paper lies in a faster implementation of the ASM. Existing implementations compute the traffic state within several minutes in realistic setups. However, for recurring applications or in real-time environments, the computation time becomes a crucial factor. Therefore, two novel implementations are proposed. In this section, first the Adaptive Smoothing Method is briefly described; then, the new implementations are specified.

A. The Adaptive Smoothing Method

The Adaptive Smoothing Method was developed by Treiber and Helbing [9]. Since then, it has been generalized [10], [11] and used in various applications [14], [15].

The ASM takes as *input* speed values $v^{\text{raw}}(x,t)$, observed at locations $x \in X^{\text{raw}}$ at times $t \in T^{\text{raw}}$. A second spatio-temporal traffic variable is optional. Usually, the flow observed at the same points as the speed map is used, but also other macroscopic quantities such as traffic density can be used. In the remainder of the paper, the symbol Z refers to any macroscopic traffic quantity, whereas V denotes specifically the speed.

The *output* of the ASM is a continuous, estimated spatio-temporal variable z^{est}. In order to solve the ASM numerically however, the filtered map is discretized at locations X^{est} and times T^{est}. Usually, this underlying space-time grid is chosen to be equidistant with resolution Δx^{est} and Δt^{est}, respectively.

The calculation of the estimated map is based on kinematic wave theory. Depending on the underlying traffic regime, the characteristics of traffic travel with a certain wave speed over space and time. Traffic is to be assumed in one of two regimes. Each of these regimes has one typical wave speed, with which the characteristics travel. In free-flow, this wave speed is around $c^{\text{f}} = 80\,\text{km/h}$, in congestion around $c^{\text{c}} = -18\,\text{km/h}$. The exact values vary depending on the conditions. However, these values lead to reasonably good estimation results. To increase the estimation quality, these parameters can also be estimated directly from the observed data z^{raw} by the Wave Speed Estimator [12].

The observed map z^{raw} is nonlinearly transformed into a smooth, estimated map z^{est}, whose elements are a weighted sum of smoothed elements of both traffic regimes:

$$z^{\text{est}}(x,t) = w(x,t) \cdot z^{\text{c}}(x,t) + [1 - w(x,t)] \cdot z^{\text{f}}(x,t)\ . \quad (1)$$

The intermediate functions z^{c} and z^{f} represent the traffic in congested and in free-flow conditions, respectively. The weighting factor w depends on the underlying traffic regimes.

The congested function z^{c} is defined by

$$z^{\text{c}}(x,t) = \frac{1}{n^{\text{c}}(x,t)} \sum_{x_i} \sum_{t_j} \phi^{\text{c}}(x_i - x, t_j - t) \cdot z^{\text{raw}}(x_i, t_j) \quad (2)$$

with the normalization factor

$$n^{\text{c}}(x,t) = \sum_{x_i} \sum_{t_j} \phi^{\text{c}}(x_i - x, t_j - t)\ , \quad (3)$$

whereby the sums loop over the observations points $x_i \in X^{\text{raw}}$ and $t_j \in T^{\text{raw}}$.

The smoothing kernel

$$\phi^{\text{c}}(x,t) = \exp\left(-\frac{|x|}{\sigma} - \frac{\left|t - \frac{x}{c^{\text{c}}}\right|}{\zeta} \right) \quad (4)$$

is an exponential function with the spatial parameter σ and the temporal parameter ζ. The characteristic congested wave speed c^{c} influences the skew of the kernel.

The free-flow function z^{f} is similarly defined to z^{c} (2) with a normalization factor n^{f}, the free-flow smoothing kernel ϕ^{f} and the free-flow wave speed c^{f}.

The weighting factor w in (1) depends on the intermediate speed functions v^{c} and v^{f}:

$$w(x,t) = \frac{1}{2}\left[1 + \tanh\left(\frac{V_{\text{crit}} - v^*(x,t)}{\Delta V}\right)\right] \quad (5)$$

with critical speed V_{crit}, transition speed range ΔV and

$$v^*(x,t) = \min\left(v^{\text{c}}(x,t), v^{\text{f}}(x,t)\right)\ . \quad (6)$$

For details about the ASM refer to the original paper [9].

The result z^{est} is defined in continuous space-time. For numerical computations, the above equations are discretized, however.

Conventional algorithms implement the double sum of z^{c} (2) in a double loop. In scientific simulation tools such as Matlab, which are specialized in matrix operations, the execution of `for`-loops is particularly slow.

The Complexity: The most complex function is the calculation of z^{c} (2). The conventional implementation loops over space and time, where the number of filter points in these dimensions is $\frac{X}{\Delta x^{\text{est}}}$ and $\frac{T}{\Delta t^{\text{est}}}$, respectively, with X and T denoting the length of the resulting rectangle in space and time, respectively.

For practical reasons, not all observations are taken into account, because the values of the kernel ϕ^{c} (4) approach zero quickly. Therefore, only a certain space-time rectangle around the filter point (x,t) is relevant for the computation. Usually, this rectangle is chosen to be of length $2a\sigma$ in space and $2a\zeta$ in time, for a kernel width factor a. A value of $a = 5$ provides good results for an exponential kernel.

The number of these relevant observation points therefore depends on a and the average observation resolution Δx^{raw} and Δt^{raw}. To conclude, the complexity of the conventional method is

$$ASM \in \mathcal{O}\left(\frac{X}{\Delta x^{\text{est}}} \cdot \frac{T}{\Delta t^{\text{est}}} \cdot \frac{a^2}{\Delta x^{\text{raw}} \cdot \Delta t^{\text{raw}}}\right)\ . \quad (7)$$

(Since the focus of this paper is on the computation time dependent on the number of input and output data points, parameters that kept constant in this paper are omitted in the complexity classes (7), e.g. σ and ζ.)

B. Solving the ASM by the Cross-correlation

The crucial part of the ASM is the computation of the intermediate regime function z^{c} (2) (and z^{f}). This equation can be solved in a different way, relying on matrix arithmetics suitable for Matlab. For this purpose, the continuous equations of the ASM are discretized and solved with the two-dimensional cross-correlation $\circledast$, which is a fast operation. The cross-correlation is defined as

$$(A \circledast B)(m,n) = \sum_{\mu}\sum_{\nu} A(\mu,\nu)\cdot B(m+\mu, n+\nu)\ , \quad (8)$$

where A and B are matrices. This section presents the algorithm of the Cross-correlation implementation.

The most important equation of the ASM is the calculation of z^{c} (2). By applying a shift of indices with $\xi_i = x_i - x$ and $\tau_j = t_j - t$, the numerator of z^{c} can be expressed by a cross-correlation operation:

$$\sum_{\xi}\sum_{\tau} \phi^{\mathrm{c}}(\xi,\tau)\cdot z^{\mathrm{in}}(x+\xi, t+\tau) = (\phi^{\mathrm{c}} \circledast z^{\mathrm{in}})(x,t)\ . \quad (9)$$

This is a continuous function. To solve the ASM numerically, a discretized version is used, where these functions are sampled at equidistant points in space and time. For this purpose, the observation data z^{raw} are discretized at equidistant points with the spatial resolution Δx^{est} and the temporal resolution Δt^{est} to the matrix Z^{in}. The congested matrix then reads

$$Z^{\mathrm{c}}_{lk} = \frac{\sum_i \sum_j \Phi^{\mathrm{c}}_{ij}\cdot Z^{\mathrm{in}}_{l+i,k+j}}{\sum_i \sum_j \Phi^{\mathrm{c}}_{ij}\cdot M_{l+i,k+j}} = \frac{\left(\Phi^{\mathrm{c}} \circledast Z^{\mathrm{in}}\right)_{lk}}{\left(\Phi^{\mathrm{c}} \circledast M\right)_{lk}}\ , \quad (10)$$

with a discretized smoothing kernel Φ^{c} and a binary observation indication matrix M. The division of the matrices in this equation is elementwise. The denominator originates from the normalization function n^{c} (3), which is a weighted sum of the kernel at all observation points. In the discretized version, only elements of Φ^{c} that correspond to an observation should be summed up. Therefore, the observation indication matrix M is defined by

$$M_{lk} = \begin{cases} 1 & \text{if observation at } Z^{\mathrm{in}}_{lk} \text{ available} \\ 0 & \text{else} \end{cases}\ . \quad (11)$$

The following algorithm explains the steps of the cross-correlation in more detail.

1) The Algorithm:

a) Discretization of observed data: The observation data maps in the equations of the ASM are discretized at equidistant points with the spatial resolution Δx^{est} and the temporal resolution Δt^{est}. The observed quantity $z^{\mathrm{raw}}(x,t)$ is mapped to that discretized point in Z^{in} which is closest to (x,t). All remaining elements in Z^{in} are set to zero.

Example 1 (Discretization of observed data map): Let the following matrices be given by observations:

$$\text{observed speeds } V^{\mathrm{raw}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & f & 9 \end{bmatrix}\ , \quad (12)$$

$$\text{observation locations } X^{\mathrm{raw}} = \begin{bmatrix} 0 & 1000 & 2900 \end{bmatrix}\ , \quad (13)$$

$$\text{observation times } T^{\mathrm{raw}} = \begin{bmatrix} 0 & 60 & 120 \end{bmatrix}\ , \quad (14)$$

where f indicates a missing observation. Let the output resolution be $\Delta x^{\mathrm{est}} = 500$ and $\Delta t^{\mathrm{est}} = 30$. Then, the discretized variables read

$$X^{\mathrm{est}} = \begin{bmatrix} 0 & 500 & 1000 & \dots & 3000 \end{bmatrix}\ , \quad (15)$$

$$T^{\mathrm{est}} = \begin{bmatrix} 0 & 30 & 60 & 90 & 120 \end{bmatrix}\ , \quad (16)$$

$$V^{\mathrm{in}} = \begin{bmatrix} 1 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 5 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 9 \end{bmatrix}\ , \quad M = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}\ . \quad (17)$$

Note that a discretization error occurs in this example: The observations at location 2900 are relocated to the nearest sample point at 3000. In practice, these discretization errors are negligible, because the output resolution is usually chosen high, for example 100 m. (Note that with that resolution no discretization error would occur in this Example.)

b) Discretization of the kernel: Next, the kernel matrix ϕ^{c} (4) is discretized with the same resolution Δx^{est} and Δt^{est}, where the maximum is in the center of the matrix:

$$X^{\Phi} = \left[-a\cdot\sigma, \dots, -\Delta x^{\mathrm{est}}, 0, \Delta x^{\mathrm{est}}, \dots, a\cdot\sigma\right] \quad (18)$$

The temporal points T^{Φ} are defined in a similar way. The congestion kernel matrix is defined as

$$\Phi^{\mathrm{c}}_{ij} = \exp\left(-\frac{|X^{\Phi}_i|}{\sigma} - \frac{\left|T^{\Phi}_j - \frac{X^{\Phi}_i}{c^{\mathrm{c}}}\right|}{\zeta}\right) \quad (19)$$

The free flow kernel matrix Φ^{f} is defined similarly.

c) Apply smoothing kernels by Cross-correlation: Apply the cross-correlation $\circledast$ (10).

d) Weighting and summing: The result Z^{est} is computed by weighting (5) and summing (1) the intermediate regime matrices (10):

$$W = \frac{1}{2}\cdot\left(1 + \tanh\left[\frac{V_c - \min\{V^{\mathrm{c}}, V^{\mathrm{f}}\}}{\Delta V}\right]\right)\ , \quad (20)$$

$$Z^{\mathrm{est}} = W \bullet Z^{\mathrm{c}} + (1-W)\bullet Z^{\mathrm{f}}\ , \quad (21)$$

with elementwise minimum, elementwise $\tanh$ and elementwise multiplication $\bullet$. Z^{est} is the output of the cross-correlation implementation of the ASM.

2) The Complexity: The complexity is significantly defined by cross-correlation operation (10). In the following, the runtime complexity of numerator $\Phi^{\mathrm{c}} \circledast Z^{\mathrm{in}}$ is analyzed; similar results hold for the denominator.

The complexity of the cross-correlation of two matrices $A \in \mathbb{R}^{N_A \times M_A}$ and $B \in \mathbb{R}^{N_B \times M_B}$ is [16]

$$A \circledast B \in \mathcal{O}\left(N_A M_A \cdot N_B M_B\right) . \tag{22}$$

The size of the input matrix Z^{in} is linearly dependent on the road stretch length X and the filter grid resolution Δx^{est}:

$$|Z^{\mathrm{in}}| \in \mathcal{O}\left(X \cdot \frac{1}{\Delta x^{\mathrm{est}}} \cdot T \cdot \frac{1}{\Delta t^{\mathrm{est}}}\right) . \tag{23}$$

The size of the smoothing kernel matrix Φ^{c} is linearly dependent on the the kernel width factor a and the resolution Δx^{est}:

$$|\Phi^{\mathrm{c}}| \in \mathcal{O}\left(a \cdot \frac{1}{\Delta x^{\mathrm{est}}} \cdot a \cdot \frac{1}{\Delta t^{\mathrm{est}}}\right) . \tag{24}$$

The cross-correlation operation is therefore the product (22) of both complexities (23) and (24):

$$\Phi^{\mathrm{c}} \circledast Z^{\mathrm{in}} \in \mathcal{O}\left(a^2 \cdot X \cdot T \cdot \frac{1}{(\Delta x^{\mathrm{est}})^2 \cdot (\Delta t^{\mathrm{est}})^2}\right) . \tag{25}$$

The remaining operations in the cross-correlation implementation are multiplications, additions and other elementwise operations, which are less complex than the cross-correlation. Since cross-correlation implementation is a sequential algorithm, its complexity equals the complexity of the most complex operation (25).

The algorithm is linearly complex in the road stretch length X, which makes it easily scalable in practical applications. The quadratic complexity in the resolution Δx^{est} might be problematic, because a high resolution is needed to minimize discretization errors caused by the equidistant input grid of Z^{in}. The kernel width factor a is usually fixed, as described above. The argumentation in the temporal dimension is similar because of the symmetry of the algorithm.

The complexity of this cross-correlation implementation (25) is nearly equal to the one of the conventional implementation (7). The only difference is an exchange of the variables of the raw data resolution with variables of the filter data resolution. However, although there seems to be no improvement of the complexity class, there is a speedup in practical runtime, as it is presented below in Section IV.

C. Solving the ASM with the Fast Fourier Transform

A fast implementation of the cross-correlation already exists. However, an even faster computation is possible with the Fast Fourier Transform (FFT). The FFT implementation of the ASM and the computation complexity are presented in this subsection.

The FFT and the cross-correlation are both related to the convolution $*$, which is defined as

$$(A * B)_{mn} = \sum_{\mu} \sum_{\nu} A_{\mu\nu} \cdot B_{m-\mu, n-\nu} \tag{26}$$

of two matrices A and B. In the special case that one of the matrices is symmetric, i.e.

$$A_{mn} = A_{M-m+1, N-n+1} , \tag{27}$$

the cross-correlation $\circledast$ equals the convolution $*$. In all cross-correlation operations of Section II-B, the kernel matrices Φ^{c} or Φ^{f} are involved. These are indeed symmetric. Therefore, all cross-correlation operations of the ASM can be replaced by convolution operations.

The convolution law

$$A * B = \mathcal{F}^{-1}\left(\mathcal{F}(A) \bullet \mathcal{F}(B)\right) \tag{28}$$

connects the convolution $*$ and the Discrete Fourier Transform $\mathcal{F}$, where $\bullet$ denotes the elementwise multiplication of two (complex) matrices. The Discrete Fourier Transform

$$\left(\mathcal{F}(A)\right)_{mn} = \sum_{\nu=1}^{N} \left(\mathrm{e}^{-\frac{2\pi \mathrm{i} \cdot \nu}{N}} \cdot \sum_{\mu=1}^{M} \mathrm{e}^{-\frac{2\pi \mathrm{i} \cdot \mu}{M}} A_{\mu\nu} \right) \tag{29}$$

converts a two-dimensional discrete signal from its space-time domain into the frequency domain. This operation is inverted by the Inverse Discrete Fourier Transform (IDFT)

$$\left(\mathcal{F}^{-1}(A)\right)_{mn} = \frac{1}{NM} \sum_{\nu=1}^{N} \left(\mathrm{e}^{\frac{2\pi \mathrm{i} \cdot \nu}{N}} \sum_{\mu=1}^{M} \mathrm{e}^{\frac{2\pi \mathrm{i} \cdot \mu}{M}} A_{\mu\nu} \right) . \tag{30}$$

The convolution of two matrices can therefore be computed by converting these matrices into the frequency domain by the DFT, multiplying their results elementwise, and transforming this product back into the space-time domain by the IDFT. The Fast Fourier Transform (FFT) is a fast implementation of the DFT, provided that the input matrices are of the same size and that their number of elements in each dimension is a power of two. The efficiency of the FFT is widely used in other scientific field, for example in mathematics to speed up polynomial multiplications [16], in the digital image format JPEG [17] (instead of the FFT itself, the slightly different Discrete Cosine Transform (DCT) is used), but also in traffic engineering to efficiently determine the reliability of travel times [18].

1) The Algorithm: The computation of the ASM with the FFT is based in the Cross-correlation implementation of Section II-B. Every cross-correlation operation is replaced by the convolution law (28).

To make the matrices of same size and their length in each direction a power of two, they are padded with zeros. Furthermore, the size of the zero-padding area has to be sufficiently large to avoid overlapping: the data map has to be zero-padded in each dimension with at least the size (18) of the original kernel function (19). After the IFFT operation, the zeros padded are removed to restore the original size of matrices. This zero-padding procedure ensures that the FFT implementation leads to the exact same results as the cross-correlation implementation.

The remaining operations of the FFT implementation are equal to the Cross-correlation implementation.

2) The Complexity: The complexity of the FFT implementation differs from the Cross-correlation implementation only in the computation of the actual cross-correlation terms.

Let $A \in \mathbb{R}^{N\times M}$ and $B \in \mathbb{R}^{N\times M}$ be matrices of the same size. Then, the complexity of the FFT (29)

$$\mathcal{F}(A) \in \mathcal{O}\left(NM \log NM\right) \tag{31}$$

is less than quadratic in the number of elements [16]. The IFFT (30) is in the same complexity class, due to the similar definition. The elementwise multiplication in (28) is linear in the number of elements. Concluding, the cross-correlation can be solved with the FFT

$$A \circledast_{\text{FFT}} B \in \mathcal{O}\left(NM \log NM\right) \tag{32}$$

in less than quadratic time.

Note that the size of the kernel Φ^{c} must be the same as of the data map Z^{in}. The computation complexity of this equation is therefore

$$\Phi^{\text{c}} \circledast_{\text{FFT}} Z^{\text{in}} \in \mathcal{O}\left(\frac{XT}{\Delta x^{\text{est}}\Delta t^{\text{est}}} \log \frac{XT}{\Delta x^{\text{est}}\Delta t^{\text{est}}}\right). \tag{33}$$

The comparison of the complexity of the FFT implementation (33) with the Cross-correlation implementation (25) shows that the FFT implementation is less complex in the resolution; i.e., the higher the resolution, the faster the FFT implementation runs compared to the Cross-correlation implementation. (In fact, the log part in (33) is not noticeable in practice; the quadratic complexity of the resulution is therefore reduced to nearly linear.) In contrast, the FFT method is more complex in the road length X and the measurement time T, favoring the Cross-correlation implementation for very large data sets.

The complexity is independent in the (average) resolution of the observation points Δt^{raw} and Δx^{raw}. This allows for the combining of loop detector data with arbitrarily many floating car data (FCD) without the losing runtime.

III. Experimental Setup

To test the quality and runtime of the proposed implementations of the ASM against the conventional one, simulations are performed. In order to get useful runtime observations, all implementations are executed with the same parameter settings and run on an idle state-of-the-art laptop.

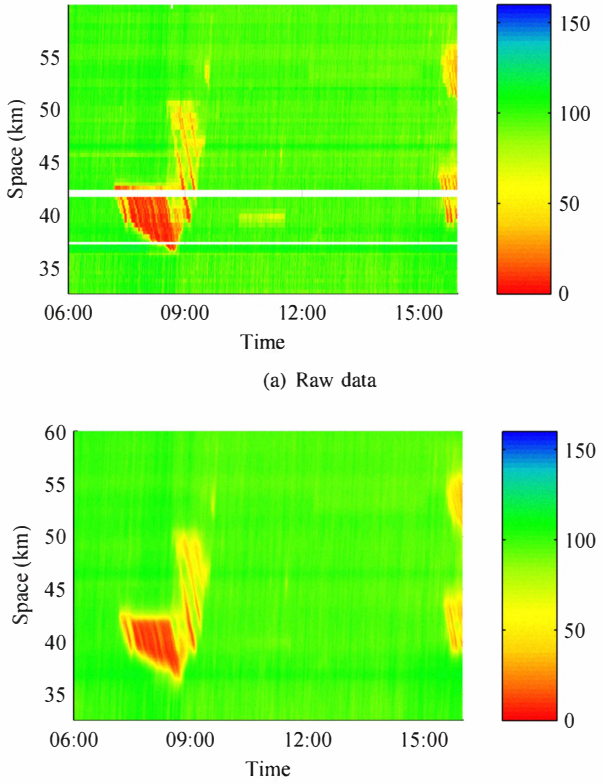
Three parameters are varied: the road length X, the measurement time T, and the resolution Δx^{est}. For every combination, simulation runs of data over 10 days are performed. The parameter settings are listed in Table I.

Raw data were gathered from dual-loop detectors from the Dutch A15 East between Kilometers 30 and 60 between 06.00 and 16.00 from October 1 to October 10, 2009. These observed speed and flow aggregated over 1 min at every 500 m on average.

The runtime is averaged arithmetically over these 10 observation days. The quality of the filter result is measured by the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) of the speed and flow maps of the Cross-correlation and the FFT implementation against the conventional ASM.

Parameter	Value(s)	Description
X	$\{5, 10, 20, 30\}$ km	road section length
T	$\{2, 4, 6, 8, 10\}$ h	measurement time length
Δx^{est}	$\{10, 25, 50, 100\}$ m	spatial resolution of results
Δt^{est}	30 s	temporal resolution of result
c^{c}	-18 km/h	congested wave speed
c^{f}	80 km/h	free-flow wave speed
ΔV	10 km/h	length of transition region
V_{crit}	70 km/h	critical speed
σ	500 m	spatial kernel length
ζ	60 s	temporal kernel length
a	5	size of kernel matrix

TABLE I
Simulation Parameter Settings



(a) Raw data

(b) Filtered data (by FFT implementation)

Fig. 1. Input and output speed data of the ASM

IV. Results and Discussion

An example of input and output data of the Adaptive Smoothing Method is shown in Figure 1. The induction loops of the A15 East near Rotterdam, The Netherlands, collected speed data over a length of $X = 30$ km and a measurement time of $T = 10$ h. Both traffic regimes, free flow (green) and congestion (yellow and red), are present. Note that some detectors do not provide data, leaving holes in the data map.

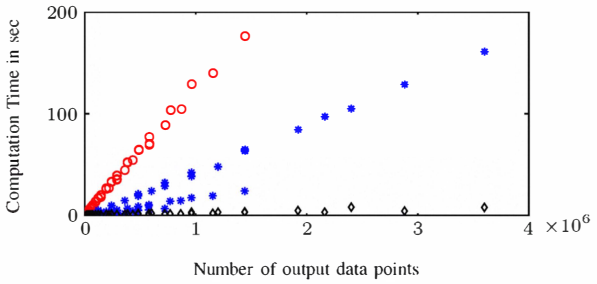


Fig. 2. Computation time against the number of filter points of the three implementations: conventional (red circles), proposed Cross-correlation (blue stars), proposed FFT (black diamonds)

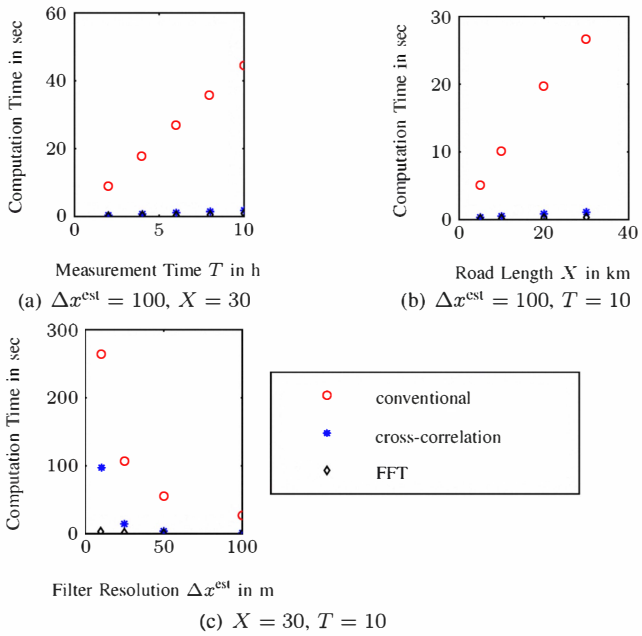


Fig. 3. Computation time against the varied variables (c.f. Table I)

(a) Δx^{est} = 10 m

	2 h	4 h	6 h	8 h	10 h
5 km	43.5	35.5	62.4	58.4	43.4
10 km	59.4	43.5	72.7	69.5	53.5
20 km	53.9	42.4	59.1	56.3	40.1
30 km	69.2	53.1	87.9	85.6	57.3

(b) Δx^{est} = 25 m

	2 h	4 h	6 h	8 h	10 h
5 km	43.4	32.2	43.9	40.3	28.8
10 km	46.7	36.7	47.7	43.3	33.3
20 km	64.3	48.0	76.5	73.5	58.4
30 km	62.5	48.2	75.5	72.0	59.4

(c) Δx^{est} = 50 m

	2 h	4 h	6 h	8 h	10 h
5 km	47.2	33.0	43.6	39.8	28.3
10 km	71.6	48.7	68.9	61.9	42.4
20 km	58.1	44.4	57.8	53.3	39.6
30 km	87.6	60.3	106.0	97.9	73.4

(d) Δx^{est} = 100 m

	2 h	4 h	6 h	8 h	10 h
5 km	48.8	35.0	44.4	39.8	26.7
10 km	60.3	41.9	56.0	50.5	36.0
20 km	86.6	56.8	81.5	72.5	49.8
30 km	88.2	59.4	84.0	75.0	52.2

TABLE II

SPEEDUP FACTOR OF THE FFT IMPLEMENTATION PROPOSED AGAINST THE CONVENTIONAL IMPLEMENTATION, AVERAGED OVER ALL 10 DAYS

	Speed	Flow
MAPE Cross-correlation	0.097 %	0.301 %
MAPE FFT	0.097 %	0.301 %
RMSE Cross-correlation	0.130 km/h	9.198 veh/h
RMSE FFT	0.130 km/h	9.198 veh/h

TABLE III

ERROR MEASUREMENTS OF THE IMPLEMENTATIONS PROPOSED AGAINST THE CONVENTIONAL ASM IMPLEMENTATION

The ASM smooths these observations, which results in a speed map. In this figure, the FFT implementation was used.

Figure 2 shows the relationship between computation time and the number of filter grid data points. Huge differences in computation time between the three implementations are apparent. The computation time of the conventional method increases sharply up to several minutes for data sets of practical size, whereas the Cross-correlation and especially the FFT run much faster. (In the case of the Cross-correlation, two lines seem to emerge. Its complexity is not solely dependent on the number of output data points; it is differently complex in the resolution Δx^{est} and in the road length X, as is shown in Section II-B.2).

Figure 3 shows the computation times in more detail. In each subplot, the computation time is plotted against one of the three variables varied from Table I, whereas the remaining two are fixed. (Note that the scale differs between the subplots.) In the largest data test with a road length of $X = 30\,\text{km}$, a measurement time $T = 10\,\text{h}$ and fine-grained resolution of $\Delta x^{\text{est}} = 10\,\text{m}$, the computation time difference is obvious (Figure 3(c)). The conventional implementation takes almost five minutes, whereas the Cross-correlation computes the result in less than two minutes. The FFT, however, is performing even faster, with a computation time of less than ten seconds.

Table II lists the speedup of the FFT implementation against the conventional ASM for all varying variables, averaged over the ten days. This speedup factor is at least one order of magnitude. The highest speedup gained, however, was more than a factor of 100. In the Cross-correlation implementation, speedup factors between 2.7 and 25 were observed.

In Table III, the errors between the two implementations proposed and the conventional implementation were calculated. The MAPEs of the speed and the flow maps for both

implementations is significantly less than half a percent. The RMSE of the flow map is around 10 $^{veh}/_{h}$, and of the speed map at less than 0.2 $^{km}/_{h}$. Further, both implementations proposed have the same error.

The filter of the two new presented implementations is therefore extremely close to the conventional implementation, leaving only very small errors. These small errors are induced by the discretization, which is necessary to apply the matrix applications. Other than that, the quality of the proposed implementations is equal to the conventional one.

V. Conclusion

This paper investigated alternative implementations of the Adaptive Smoothing Method (ASM) [9]. The ASM is a data-driven traffic state estimator, which filters macroscopic traffic observations like speed and flow by kinematic wave theory. The conventional implementation was compared with a new implementation based on the two-dimensional cross-correlation, and a new implementation based on the Fast Fourier Transform (FFT).

In these two implementations, the observations are not treated as single variables, but are gathered in a matrix representing a speed or a flow map. The cross-correlation operation and the FFT are applied to these matrices to solve the ASM. Very fast algorithms of these operations are available, which were incorporated into the ASM implementations proposed.

Experiments based on real data with realistic data sizes show a vast speedup of the computation. The gain in computation time reaches from a factor of twenty in the Cross-correlation up to factor of one hundred in the FFT implementation. The filter quality is nearly preserved with respect to the conventional implementation of the ASM, introducing only negligible discretization errors.

In addition, the proposed implementations do not contain any loops, which makes them easier to maintain and easier to port than the conventional implementation. This lack of loops is indeed a reason why the simulations performed in Matlab were so fast.

The fast computation time of a few seconds on a state-of-the-art laptop make the proposed implementations of the ASM usable for real-time applications which are based on the current traffic state. These include traffic state predictors, which in turn are fundamental tools for Dynamic Traffic Management and Advanced Traveler Information Systems.

Acknowledgment

This research work is sponsored under Research Grant "The MultiModal Port Traffic Centre" by the Port of Rotterdam Authority, Rijkswaterstaat Zuid-Holland and De Verkeersonderneming Rotterdam.

References

[1] J. van Lint and N. van der Zijpp, "Improving a Travel-Time Estimation Algorithm by Using Dual Loop Detectors," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855, pp. 41–48, 2003.

[2] C. Van Hinsbergen, F. Zuurbier, J. Van Lint, and H. Van Zuylen, "Using an LWR Model with a Cell Based Extended Kalman Filter to Estimate Travel Times," in *Third International Symposium of Transport Simulation*, no. 65, Surfers Paradise, QLD, Australia, 2008.

[3] M. Lighthill and G. Whitham, "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990)*, vol. 229, no. 1178, pp. 317–345, 1955.

[4] P. Richards, "Shock Waves on the Highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[5] J. Van Lint, S. Hoogendoorn, and M. Schreuder, "Fastlane: New Multiclass First-Order Traffic Flow Model," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2088, no. -1, pp. 177–187, 2008.

[6] Y. Wang, M. Papageorgiou, and A. Messmer, "RENAISSANCE–A Unified Macroscopic Model-Based Aapproach to Real-Time Freeway Network Traffic Surveillance," *Transportation Research Part C*, vol. 14, no. 3, pp. 190–212, 2006.

[7] H. Payne, "Models of Freeway Traffic and Control," *Mathematical Models of Public Systems*, vol. 1, no. 1, pp. 51–61, 1971.

[8] S. Hoogendoorn and P. Bovy, "State-of-the-art of Vehicular Traffic Flow Modelling," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, pp. 283–303, 2001.

[9] M. Treiber and D. Helbing, "Reconstructing the Spatio-Temporal Traffic Dynamics from Stationary Detector Data," *Cooper@tive Tr@nsport@tion Dyn@mics*, vol. 1, no. 3, pp. 3.1–3.21, 2002.

[10] J. Van Lint and S. P. Hoogendoorn, "A Robust and Efficient Method for Fusing Heterogeneous Data from Traffic Sensors on Freeways," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, pp. 1–17, nov 2009.

[11] M. Treiber, A. Kesting, and R. E. Wilson, "Reconstructing the traffic state by fusion of heterogenous data," *accepted for publication in Computer-Aided Civil and Infrastructure Engineering*, vol. preprint `physics/0900.4467`, 2010.

[12] T. Schreiter, H. van Lint, Y. Yuan, and S. Hoogendoorn, "Propagation Wave Speed Estimation of Freeway Traffic with Image Processing Tools," in *in proceeding of the Transportation Research Board, 89th Annual Meeting, Washington, D.C., DVD*, 2010.

[13] Y. Yuan, J. W. C. Van Lint, T. Schreiter, S. P. Hoogendoorn, and J. Vrancken, "Automatic Speed-Bias Correction with Flow-Density Relationships," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control; Chicago, IL, US*, 2010.

[14] J. Van Lint, "Empirical Evaluation of New Robust Travel Time Estimation Algorithms," in *Compendium of papers TRB 89th annual meeting*, 2010.

[15] A. Kesting and M. Treiber, "Calculating Travel Times from Reconstructed Spatiotemporal Traffic Data," in *Proceedings of the 4th International Symposium Networks for Mobility, Stuttgart*, 2008, iSBN: 978-3-921882-24-5.

[16] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT press, 2nd edition (September 1, 2001), 2001.

[17] G. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, 1992.

[18] M. Ng and S. T. Waller, "A computationally efficient methodology to characterize travel time reliability using the fast fourier transform," *Transportation Research Part B: Methodological*, vol. In Press, Corrected Proof, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6V99-4YMBVW1-1/2/74d810e4ba3e7bcc38feaaa56c24e822