COMPARISON OF INTERPOLATION TECHNIQUES FOR STATE ESTIMATION ON URBAN NETWORKS
L.J.J. Wismans\textsuperscript{1,2}, L.O. de Vries\textsuperscript{3} and E.C. van Berkum\textsuperscript{2}
\textsuperscript{1} DAT.Mobility, the Netherlands
\textsuperscript{2} University of Twente, the Netherlands
\textsuperscript{3} SWECO, the Netherlands

Abstract
State estimation is an important instrument for understanding the daily urban system and its spatial and temporal dynamics. With these insights we are able to better predict future traffic states and improve the demand-supply match. If state estimation is performed real-time it can be used for short term prediction and virtual patrolling. Most earlier research focuses on highways, while less is known about urban networks because of less available measurements and urban networks are more complex. Therefore we compared two promising slightly adapted but relatively simple, scalable and fast spatial interpolation methods, respectively a simplified form of Variance-Based Interpolation (VBI) and Learning-Database Interpolation (LDI), for an urban network using floating car data based on a micro simulation providing the ground truth. The performance of these methods was assessed depending on penetration rate compared with a reference situation of no interpolation. The results show that the VBI method performs reasonably well up to 9% coverage, but at higher penetration rates performs worse than the reference situation. The performance of LDI is much more promising, at low penetration rates it already shows large improvement and it continues to outperform the reference situation up to 40% of FCD coverage.
1. Introduction
Transport is a premise for economic growth. However, congestion problems and externalities related to traffic costs society billions of which substantial reductions can be realized by improving the match between demand and supply [1]. For this it is necessary to have a good understanding of the daily urban system described by the spatial and temporal dynamics of traffic states and underlying correlations [2,3,4,5]. Real time state estimation can for example be used to determine bottleneck locations, physical extent of queues and travel times, which is important information for ITS systems, in-car, road side as well as for traffic management centers [6].

For determining both the current and future network state, a distinction can be made between model based and data driven methods. The latter has the advantage of being able to run in real time and requires less to none calibration [6], yielding opportunities for direct traffic management. However, the big assumption in both the data driven and model based methods, is that the traffic information which provides a current network traffic state is assumed to be by definition correct, if the information provided is accurate and reliable. However, this is certainly not the case for simulation driven methods, but also the traffic information in general is subject to availability and reliability [7,8]. Literature therefore concludes that it is already difficult to estimate the current network traffic state from the traffic information provided and thus emphasizes on the importance of proper spatial and temporal interpolation methods [9]. For the traffic state estimation in literature the predominantly used estimation algorithms are based on the methodology of Kalman Filtering [10]. This mathematical method derives from a series of measurements (containing noise and inaccuracies) a more precise estimate of the unknown traffic flow variables. Also extensions and combinations of different methods are used, such as Extended Kalman Filtering (EKF) [11], the Unsended Kalman Filtering (UKF) [12] or Particle Filtering (PF) [13]. The literature also suggests less complex and already quite successful methods that are used for real-time traffic state estimation, as for example; Nudging [8], Learning Database Interpolation [14], Neural Networks [15] and Adaptive Smoothing [9].

Recently, the interest of traffic state estimation has shifted from the relative comfort of freeway segments, to more complex road networks [16]. To date, we lack a proper understanding of the daily urban system described by spatial and temporal dynamics in travel behavior and traffic conditions. Evidence-based planning is in desperate need of analytics and of relevant data to reveal caveats at a fine-grained scale, both in terms of space and time [17]. These networks contain, besides the upper network of mostly highways, also important primary and secondary urban arterials. As both a larger network and its urban nature (e.g. intersections and vehicles reaching their destination or departing) raises the traffic complexity, it provides for an additional challenge in network traffic state estimation. Furthermore, there are traditionally less measurements available compared to freeways.

The recent rapid advances in information technology have led to various cost-effective data collection systems which have enriched the sources of empirical data for the traffic state estimation problems. Such new data are usually collected from remote sensing techniques delivering floating car data. There is limited research available in which the performance of existing methods or new methods for urban networks are described. In [16] a model based approach is presented and tested, [17] presents methods for travel time and density estimation on an urban arterial, [18] presented a micro-state model and [19] a data expansion algorithm to estimate flow on non-measured segments, [20] and [21] present different method based on floating car trajectory data to estimate urban traffic congestion. In most cases new relatively complex methods are proposed compared to earlier developed methods for freeways and focus is on flows, congestion or travel times on routes, while it is of interest whether it is possible to use the same methods regardless of network type, which are fast, scalable and relatively easy to use (e.g. regarding parameter settings) and can provide a state estimation for all segments (i.e. can cope with missing data and data accuracy). Furthermore, it is always of interest to compare methods, to assess suitability for application or further development.

Therefore we evaluate the performance of two relative simple, which we slightly adapted, and quite successful interpolation methods earlier presented in [8] and [14], respectively a simplified form of Variance-Based Interpolation (VBI) and Learning-Database Interpolation (LDI) These methods are in previous research only assessed on either a freeway [14] or an urban-network without having access.
to the ground truth [8]. We will assess the performance for an urban setting using FCD to determine link velocities dependent of penetration rate and also show the influence in accuracy of the methods when the spatial or temporal resolution is changed. Additional reason for choosing these methods is that they can cope with the complexity of the urban network mentioned earlier and the needed low computation times, which means that they can be used for real-time purposes.

In the second Section we will further explain the two methods. Then the case, derivation of the ground truth and data handling is described in Section three. The assessment of the methods and results are described in the fourth Section. The last section describes the conclusions and further research.

2. Methods
2.1 Theoretical framework
The methods can best be described using a pheromone model as a basis for FCD. In an ant-based simulation, cars drop pheromones continuously throughout their journey, which evaporate after a certain time period. In microsimulation it is possible to assume that stronger pheromones are dropped whenever the car is braking, stopping, or turning. The basis of the pheromone model in macro simulation is the aggregated version of the microsimulation, in which the velocity of the car is directly linked to the pheromone output. When dividing up the network into road links (segments) the pheromones emitted on each link, within a certain time interval determine the mean velocity on that link. By inducing a cut, a source or sink can be introduced in the network. These act as a ghost links and represent the disappearing and appearing traffic, or can represent a junction. The situation on a two close by links might therefore be (very) different. This models the real world very closely, as in a city network, cars are appearing (starting a trip), cars are disappearing (arrived at destination) and cars are moving through space and time (actual travelling). In our case only aggregated emitted pheromones are required to yield the required velocity estimation.

In this research a microsimulation model (i.e. Paramics) is used to be able to construct the ground truth and to assess the methods. The creation of the pheromones is defined by the FCD generated by the model, in this case every 0.5 seconds. Appearing and disappearing of pheromones describes two aspects. First of all vehicles can start creating pheromones when departing and will stop creating pheromones when arriving at their destination. In our simulation model this can happen, like in reality, on every urban link at a random location, but in accordance with the defined Origin Destination matrix. Second the emitted pheromones evaporate. Intuitively the pheromones emitted on a certain timestamp t at a certain location x will become less reliable or redundant when time passes. Although it is possible to consider temporal interpolation (e.g. average speed of pheromones of previous time intervals on a certain link are used for the estimation of the average speed on that link for the current time interval) we assume that the pheromones evaporate at the end of every time interval of 60 seconds. After every 60 seconds therefore an empty pheromones network is laid out. The conductivity of the pheromones describes in this case how far the pheromones move after their creation and before their disappearance. Practically for this research it is related to the correlation of velocities expected between nearby links in the network and determines the size of the reference link database (RLD), which describes all adjacent links for each link in the network. The insights which are given for the pheromone movement to neighboring links are described by [22] using the theory regarding propagation of congestion. Shockwaves of congestion might move both upstream and downstream in the network therefore effecting directly adjacent links. Also, the congestion on a link might saturate the nearby junctions and therefore also effecting links connected to those junctions. For the relation to the link in opposite direction, the term rubbernecking is used, where the traffic slows down to look at the situation on the opposite lane. Although this can happen in reality, the simulation model does not take this into account, which means we only use the direct adjacent downstream and upstream links as a reference link database. As a result of the evaporation and conductivity we only assume spatial interpolation (i.e. average speeds on reference links in a certain time interval are used to estimate the average speed on the link for that same time interval)
2.2 Variance Based Interpolation

The Variance-Based Interpolation (VBI) method is a relative simple method inspired by [9] Adaptive Smoothing Method. It aims towards finding a weighting of reference links \( w(r) \), to retrieve the highest reliable estimation by using the variance in measurements of the reference links as an indicator for reliability. That is, the reported velocity on a link gets enriched by also including the mean velocity reported on adjacent links and thus report a more reliable final estimation. The indicator for this reliability is indeed linked to the variance of the sample, because the bigger the sample, the lower the variance. Therefore the variance it is a direct index of the sample’s reliability. Since the aim is to maximize the combined reliability, the goal is to minimize the combined variance; \( \theta(m) \) of a link \( m \), which is also the aim of the Adaptive Smoothing Method [9].

Through trial and error the parameters are set such that the RLD is for VBI limited to only upstream and downstream links and gives double weight to its own data. Which yields a maximum of four reference links per link in the RLD. The minimization problem proposed by [9] (p. 49) is expanded as to include also twice the link’s own velocity:

\[
\begin{align*}
\text{Minimize:} & \quad \tilde{\theta}(m) = \sum_r w(r)^2 \ast \theta(r) + 2(w(m))^2 \ast \theta(m) \\
\text{subject to:} & \quad \sum_r w(r) + 2w(m) = 1
\end{align*}
\]

Assumed is by [9] that in order to determine these reliability weights \( w(r) \) and \( w(m) \) the assumption is made that (i) the different data sources \( r \) and \( m \) bear no systematic errors, (ii) the variance \( \theta(m) \) and \( \theta(r) \) of the random errors is known, and (iii) the errors of the different sources are uncorrelated. By using Lagrange multipliers this minimization problem is solved [9] (p.50). The conclusion of this expanded problem remains the same; “the weights should be proportional to the inverse of the variance of the errors in the data sources”. The final velocity estimation (vector \( \tilde{x}(m) \)) is the result of taking the sum over the reported velocities of each link and their calculated weight, thus calculating the linear combination of velocities and their weights as coefficients. Any practical issues concerning this method, will be discussed later on.

2.3 Learning-Database Interpolation

The Learning-Database Interpolation (LDI) method, is a method designed by [14]. It is an evolved version of a (normal) spatial interpolation system for traffic conditions, which does not use the variance of the reported mean velocities. It aims at finding correlation between links in the network, to enrich the final estimation by assigning the proper weights to the reference links. [14] proposes the use of a system that includes estimation and learning agents, which are assigned to all the road links. Estimation agents renew the velocities for each road link, and learning agents renew their reference weight values for better estimation. All the data obtained, is stored in a central Learning-Database. The structure is presented in the next figure, which has been derived and simplified from [14] representation.

![Figure 1. Overview LDI method](image-url)
The main database requires input to break the deadlock. The data inputted at first, are the average velocities on the links. For clarity the time interval at which this velocity was measured is also added in the notation. These reported velocities are described by:

\[ x^t(m) = \text{average velocity on link } m \text{ at time interval } t \text{ [km/h]} \]

The Estimation agents (E) of the central database is then built up by storing this data. After each time step, a new line is added to the matrix, with the reported velocities at that time \((t+1)\). This database is therefore continuously expanding, as shown below:

The Learning part of the central database is stored in a vector, for which the reference road link matrix serves as basis. The vector consists only of the reference road links that are applicable (=1) for a particular link. Therefore, the road link itself is not used for weight determination, as it serves as the Estimation agent already. The Learning-Database for road link m is described as:

The nonzero weights (independent variables) depend on how many (predefined) reference road links a link has and can be modified. The exact weight of the nonzero weights, is determined by the interpolation system described in the next paragraph. Also, a new weight matrix is created after each iteration.

The output of the interpolation system on time \(t\), is both a vector of outputted link velocities; the so called estimation \((\hat{P}^t)\) and the weight matrix \((L^t)\) used to calculate that estimated velocity. These link velocities thus serve as the estimation \(\hat{x}(m)\), which is needed for the last phase of the Matlab model. The estimation \(\hat{P}^t(m)\) can be calculated by using matrix multiplication, if the weight matrix \((L^t)\) is known:

The weights \(L^t(m)\) of the links are determined, such that the mean velocity on the link \((m)\) is a linear combination of the velocities reported by the reference links, by using \(E(m)\), which is the reported velocity of the FCD, as an estimator of the outcome. The weight determination problem is described as:

\[
\text{Finding weights: } L^t(m), \text{ such that: } [E] \times [L^t(m)]^T = [E(m)]
\]

When the rank of \([L^t(m)]^T\) is less than the rank of \(E\), the solutions are not fixed, which is the case when the central database is smaller in size than the reference road links of a link. When the rank of \([L^t(m)]^T\) and \(E\) are equal, the equation can be solved perfectly. If the number of independent equations in \(E\) is larger than the rank of \([L^t(m)]^T\), then the above equation obviously cannot be solved. The solution is to use the (M)SE. That is, find weights such that the residuals of each equation is minimized, this also minimizes the MSE. The weight determination problem becomes;

\[
\text{Finding weights } L^t(m) \text{ such that } \sum_{i=1}^{160}([E] \times [L^t(i)])^T - [E(i)])^2 \text{ is minimized}
\]

It must be mentioned that [14] proposes to use linear, multivariate analysis and Gaussian Elimination to determine the regression coefficients which serve as the weights. For implementation, the Levenberg-Marquardt algorithm available in MATLAB is used to solve this nonlinear least-squares (nonlinear data-fitting) problem. This algorithm is also known as the damped least-squares (DLS). It is
generally slower than the Gauss-Newton approach of [14] but it is more robust and more practical for the purpose of this research. This yields the proper weights to calculate the estimation:

\[ [E] \times [L^T(m)]^T = P^T(m) \]

\[ \bar{x}(m) = P^T(m) = \text{average velocity on link } m \ [\text{km/h}] \]

2.4 Practical issues VBI and LDI

Both methods have some practical issues regarding feasibility, which need to be addressed. For example when the FCD sample size of a source \( N(s) \leq 1 \). When \( s = 0 \), there is no data for that link. The reported speed velocity is set at the \( v_{\text{max}}(s) \) with variance \( v_{\text{max}}(s)^2 \). When \( N(s) = 1 \), there is only one sample available, which yields (by definition); \( \theta(s) = 0 \). Thus again: \( \theta(s) = v_{\text{max}}(s)^2 \).

3. Case

To assess the VBI and LDI method for an urban setting a microsimulation model of a grid network of blocks, sized 600 by 600 meters, is build using Paramics simulation software. All 160 main links are of an equal length of 200 meters and consists of a mixture of road types (i.e. differences in speed limits (70, 50 and 30 km/h) and number of lanes (1 or 2 lanes)) and intersection types (give way and signal controlled) to mimic an urban environment. Modelled time period is one hour. The demand is chosen in such a way that serious congestion problems occur in the network (see Figure 2, showing the velocity per link in classes during the time interval 9 to 10 minutes), in which in some cases blocking back does appear. Route choice is modelled using default settings and for simplicity uni-modality is assumed (i.e. passenger cars only).

![Figure 2. Network and indication of ground truth velocities](image-url)
After running the simulation all FCD data of all vehicles is stored in a database and split in time intervals to calculate the ground truth average link velocities and variance for each link and each time interval. To assess the LDI and VBI methods depend on penetration rate, FCD data is randomly deleted. For all tag numbers associated with a unique car in the simulation, a random value between 0 and 1 is generated. If this number is bigger than the set FCD coverage variable, the tag number will be deleted from all the databases. This way the penetration rate is approximated. The result is a cleaned database, sorted per time interval, ready for further usage. As the deletion of FCD is randomized, each run might provide a different subset of the FCD. Especially in the lower percentages of FCD it is statistically more likely to draw a so called less representative sample from the population, thus affecting the final results. As the FCD coverage increases, this is less likely to occur. For the final result, at each FCD coverage % up until 50%, the model is run five times. Above 50%, the model is only run once every 5%.

To assess the results two measures are used, the MSE (Mean Square Error) and a relative MSE (rMSE), not to be confused with the Root Mean Square Error (RMSE). The MSE compares the velocity estimate $\hat{x}(m)$ of the LDI or VBI method with the ground truth velocity $\tilde{x}(m)$.

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (\hat{x}(m) - \tilde{x}(m))^2$$

The rMSE is used, because it is of interest to know how well the methods perform on links with low FCD penetration. First of all because on those links the interpolation techniques are the most needed and second of all, because on those links you would expect the largest differences between the estimates and the ground truth. The rMSE gives an indication of the $10^{th}$ percentile FCD on links, and their relative share towards the MSE. To determine the $10^{th}$ links with the lowest FCD penetration in a specific time interval, the number of unique car appearances on a link $m$ in that specific time interval is used for the ground truth $\tilde{N}(m)$ and LDI or VBI method $\tilde{N}(m)$ and the links with the $10^{th}$ lowest number of unique car appearances are stored in $F$.

$$rMSE(\cdot, 1) = \frac{1}{MSE} \sum_{i=1}^{m} (\hat{x}(m) - \tilde{x}(m))^2 \quad | m \in F$$

For comparison reasons the average velocities $E(m)$ based on the available FCD measurements (i.e. depend on penetration rate) for each specific link (i.e. no interpolation used) are used and called method REF.

4. Results
The VBI and LDI methods and REF method are compared on their performance on MSE and rMSE. Figure 3 shows the results depending on penetration rate.
The MSE of the LDI method performs at low FCD coverage approximately 50% better than without any interpolation (REF) and improves performance up to 75% at 10% FCD and until 40% FCD coverage remains the best performing method. Round the 10% FCD it already has a MSE of 115, which does not improve much until reaching 100% FCD at which the MSE is 83. It is also more robust as the spread of model results is less than VBI and the REF. Also in terms of the rMSE the LDI model is quite successful at estimating the velocity on the links with little to none FCD.

The MSE of the VBI method performs at low FCD coverage approximately 20% better than without any interpolation. However, this head start is lost at around 15% FCD after it performs considerably worse than the REF does. VBI does however do what it is supposed to do by minimizing the variance of the data, as the spread of the MSE is less than the REF throughout all coverage rates, as well as the rMSE is declining rapidly, indicating that the links with the least FCD are contributing relatively less towards the MSE.

The below scatter plot shows the difference between on one hand the VBI and LDI method and the difference compared to the GT on the other hand on link level, given an FCD penetration rate of 10%. The VBI results show a slight bias towards underestimation of the real network state. The LDI results show less bias towards underestimation. The results also show that, not surprisingly, the largest differences are found in the middle, representing links where queues start to form and the variance in velocity between vehicles are expected to be the highest (i.e. high velocity measurements in the beginning of the link and low velocity at the end of the links).

Figure 4 shows the results of a test in which we compared the performance of the VBI method when a time interval is chosen of 120 seconds instead of 60 seconds or link length of 600 meters instead of 200 meters. It is expected that the increase of aggregation has a positive effect on the estimation, as there is less chance that a link has no FCD, and also that a link has more FCD and thus the estimate has
less chance to be an outlier (of a very fast or slow car). As the number of unique cars on the network remains relatively the same, the positive effect is mostly related to the aggregation of outliers, and thus flattening the estimations. The MSE of the 120 second shows throughout all FCD coverage percentages up to 20% better MSE rates than with only 60 seconds data and spread in MSE is not visibly different. It does however come at the cost of twice the runtime, and twice the interval size which means that estimated velocities will be more flattened decreasing the value for virtual patrolling.

**Figure 5.** Performance depending on spatial and temporal resolution

In increase of aggregation on distance should also have a positive effect on the estimation, as again there is less chance that a link has no FCD, and also that a link has (on average) three times more FCD and thus the estimate is smoothened. The MSE of both the 600 meters and 200 meters follow the same trend. Only at the very low FCD rates, between 1% and 15% the 600 meters seems to perform up to 15% better in terms of the MSE. Also the spread is comparable. With a decrease of three times the number of links, this method is executed twice as fast as well (i.e. less links to estimate, however an increase in FCD to assess), however also in this case the value for virtual patrolling will be less.
5. Discussion and conclusions

The VBI, but especially the LDI method shows that at low penetration rates it is possible to deliver much better estimates of link velocities on urban networks than without using interpolation techniques. The advantage of these methods is that they are relatively simple, scalable and fast. However, the estimates are most weak at links where the traffic conditions are near saturation or spatial differences in velocities are potentially large. The results show that the relatively simple Variance-Based Interpolation (VBI) performs reasonably well up to 9% coverage, and performs worse at higher rates. Because there are also some practical issues related to this method regarding the assumptions made, it is not recommended to use this method in practice. First of all issues regarding the availability of the speed variance based on FCD data, second of all there are possible systematic errors using GPS and third there is a possibility that the errors of the different sources are correlated. The performance of Learning-Database Interpolation (LDI) is much more promising, although it is associated with higher calculation times, it already reaches a high potential in this research, at already 10% of FCD coverage and continues to outperform the reference situation up to 40% of FCD coverage. Spatial and temporal aggregation shows as expected an improvement of the estimates. However, more aggregation means less usefulness of the estimates for understanding the daily urban system and its spatial and temporal dynamics.

Further research will be needed by extending with more vehicle types, more traffic variables (e.g. flow and density, but also routes and origins and destinations) and more available data sources as well as testing in practice on more and larger networks. Additionally, other methods should be tested or combined for complete network state estimation.

References

7. Li, J., Xie, L., Lai, X. (2013). Route Reconstruction from Floating Car Data with Low SamplingRate Based on Feature Matching. Research Center of Intelligent Transportation System, Sun Yat-Sen University, China


