Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

MAP STREAM: INITIALIZING WHAT-IF ANALYSES FOR REAL-TIME SYMBIOTIC TRAFFIC SIMULATIONS

Abhinav Sunderrajan

TUM CREATE Ltd
1 CREATE Way
138602, SINGAPORE

Heiko Aydt

TUM CREATE Ltd
1 CREATE Way
138602, SINGAPORE

Wentong Cai

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue 639798, SINGAPORE

Alois Knoll

Robotics and Embedded Systems Group
Department of Informatics
Technische Universität München
Boltzmannstraße 3
D-85748 Garching bei München, GERMANY

ABSTRACT

In the context of a city-scale symbiotic traffic simulation, real-time data about the location of many vehicles are obtained in the form of a continuous data-stream. In this paper, we present a scalable solution for performing map-matching using sliding-windows over a GPS data-stream onto a digital road network for initializing the what-if analysis process involved in symbiotic simulations. We focus on the optimizations performed to ensure that the latency associated with the map-matching process is low while maintaining a high degree of accuracy. Experimental results reveal the range in terms of sampling interval and noise for acceptable reliability and latency.

1 INTRODUCTION

Introduced in (Fujimoto et al. 2002), symbiotic simulation is a paradigm which is characterized by a mutually beneficial relationship between the physical environment and the simulation system. While the simulation system benefits from the continuous measurements provided by the physical system, the latter benefits from the near real-time decisions provided by the former. Symbiotic simulations involve a what-if analysis (WIA) process which is responsible for creating and evaluating several alternative scenarios through simulations. Executing these simulations as fast as possible is important in the context of a symbiotic simulation. Initializing a simulation run with the state of the physical system is potentially time consuming. In this paper we present and evaluate an appropriate initialization method for traffic simulations.

As motivated in (Aydt, Lees, and Knoll 2012), a city-scale symbiotic traffic-simulation involving tens of thousands of vehicles depends on incorporating real-time measurements provided by white-box and gray-box vehicles which share information regarding their geo-location, speed and origin-destination (white-box vehicles only). For the symbiotic simulation to be effective, it is essential to process the data, initialize and execute the simulation with minimum latency. A symbiotic traffic-simulation could be initialized by using the GPS location information provided by the vehicles. However due to inaccuracies associated with GPS data (Wang et al. 2011) suitable map-matching algorithms need to be employed. Map-matching refers to the process of aligning a sequence of GPS location data from a vehicle with a spatial road network to identify the road(s) the vehicle is traveling. Given the sheer number of vehicles on the entire road network of a city, the location of vehicles is obtained in the form of an unbounded, continuous data-stream. Hence traditional map-matching techniques which focus on matching over a batch of data are either unsuitable or need to be optimized. Given the streaming nature of the GPS probe data it is critical that the map-matching algorithm is not only reliable but also able to compute the results with minimal latency. In this paper we present and evaluate a scalable algorithm which can be used to map the position of several vehicles from a data-stream on to a digital road network using sliding windows. The solution also incorporates a set of optimizations in order to minimize the map-matching latency while maintaining a high degree of accuracy.

The organization of the rest of the paper is as follows. In Section 2 we introduce the map-matching problem and discuss related work. In Section 3 we present in detail our algorithm for map-matching over a data-stream while discussing the optimizations made for reducing the latency associated with map-matching. In Section 4 we present the results of our solution which compares two aspects of map matching namely GPS noise vs sampling interval vs reliability and GPS noise vs sampling interval vs latency. We present our conclusions in Section 5.

2 MAP MATCHING PROBLEM

Map-matching algorithms have been used to map noisy GPS data on to a digital road network for supporting functions such as navigation in Intelligent Transportation Systems (ITS). Detailed review of existing map-matching techniques can be found in (Quddus, Ochieng, and Noland 2007). Geometric map-matching algorithms (Jagadeesh, Srikanthan, and Zhang 2004) make use of the shape of the roads and (or) the trajectory of the vehicles on the roads. Topological map-matching (Yin and Wolfson 2004) algorithms consider the road connectivity and contiguity along with geometric features. Advanced techniques such as Kalman Filters (Obradovic, Lenz, and Schupfner 2006), have also been effectively used for map-matching.

Regardless of the approach of the approach they apply, all map-matching algorithms ultimately need to associate a GPS sample with a road. Due to inherent noise there is always some uncertainty with associating a GPS sample with a road. For example consider the case illustrated in Figure 1. The GPS sample z_1 could be associated with any of the three roads $\{AB, CD, EF\}$. Given a single geo-location, it is hard to identify the exact road segment a vehicle is on. Hence to make a guess on the probable road segment traversed, we need at least two signals from a single vehicle. Thus the second noisy signal z_2 and the associated roads $\{AB, CD\}$ are required.

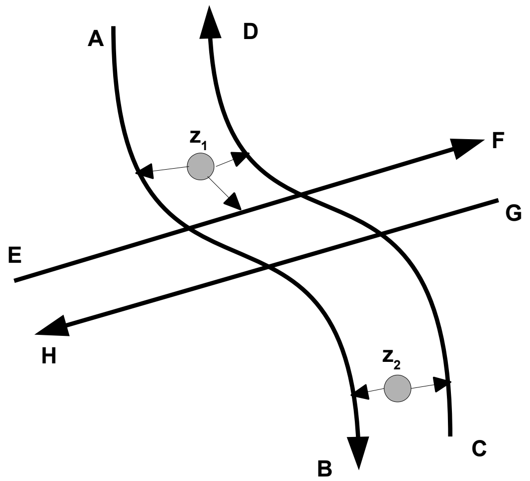

Figure 1: Example for uncertainty associated with map-matching.

We use a Hidden Markov Model(HMM) based algorithm for map matching. HMM is used to model a process with hidden states using the observed states. It is assumed that the hidden states form a Markov Chain.

An HMM is able to incorporate noisy data elegantly and has found applications in several domains (Rabiner 1989). Figure 2 shows the Trellis diagram of the HMM for map-matching. Given a road network $G(V,E)$, where V is a set of all vertices and E is a set of all edges, a hidden state of the HMM represents the actual position of a vehicle on an edge $e_t \in E$ at time t. We assume that the future location of a vehicle depends only on the current location thus making the vehicular movements a Markov process. $Z = (z_t|t = 1 \ldots n)$ denotes the entire trajectory of the vehicle from its origin to destination. The noisy GPS data point or the observed state of the HMM at time t is represented by $z_t \in Z$. Each z_t is a tuple containing latitude, longitude, vehicle-id and time-stamp information as fields. Each edge e_t is associated with a begin vertex $e_t.v_{begin}$ and an end vertex $e_t.v_{end}$. Finally for future reference, the term road segment refers to the sequence of edges constituting the shortest path from one vertex to another in the road network.

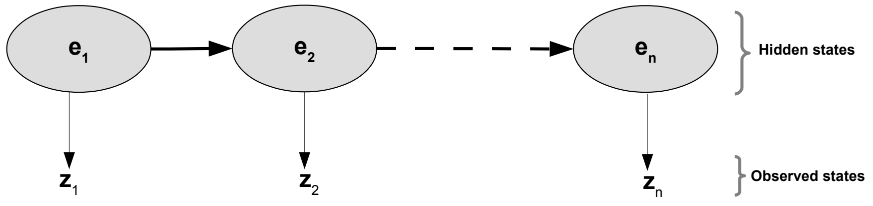

Figure 2: Trellis diagram for HMM.

The joint probability distribution of an HMM is given by Equation 1. The quantity $p(z_1|e_1) \times p(e_1)$ represents the initial distribution and is assumed to be known. The unknown terms namely the transition and emission probabilities need to be computed. Emission probabilities $p(z_t|e_t)$ represents the probability of a noisy GPS data point z_t being associated with an edge $e_t \in E$. Transition probabilities, $p(e_t|e_{t-1})$, represent the probability that a vehicle travels to edge e_t given that it is currently on e_{t-1}.

$$p(z_1, z_2, z_3 \ldots z_n, e_1, e_2, e_3 \ldots e_n) = p(z_1|e_1) \times p(e_1) \sum_{t=2}^{n} p(e_t|e_{t-1}) \times p(z_t|e_t) \tag{1}$$

HMM has been used for map-matching in various other works including (Newson and Krumm 2009), (Krumm, Letchner, and Horvitz 2007), (Goh et al. 2012). The work done in (Krumm, Letchner, and Horvitz 2007) and (Newson and Krumm 2009) (an enhancement of the former) treat map-matching as a batch problem and use the entire trajectory for computing the results. However we use the work done in (Newson and Krumm 2009) as the foundation for computing emission and transition probabilities. The work by (Goh et al. 2012) is close to our work since it focuses on keeping the latency associated with map-matching minimal while maintaining a high degree of accuracy for real time applications. For computing the transition probabilities, (Goh et al. 2012) train a Support Vector Machine classifier to classify incorrect and correct transitions from a given edge e_i to edge e_j. The associated scoring functions of the classifier make use of the velocity information of the vehicle. Further they use sliding-windows of variable length (with an upper bound) before emitting the optimal results. In our work we fix the length of the sliding window used to two. While the accuracy of the algorithm can be expected to be lower, the latency for computing the results are much smaller. Further we also incorporate optimizations to reduce the latency for map-matching. Experiments were also performed to determine the relative accuracy and time taken for map-matching under varying noise and sampling intervals δ.

3 MAP MATCHING OVER A DATA STREAM

3.1 Map matching with sliding windows of size two

For implementing a real-time, on-line map-matching algorithm, the GPS signals belonging to a single vehicle are thus added to a sliding length window of size 2. The Trellis diagram shown in Figure 2 thus reduces to computing only two states, i.e., e_{t-1} to e_t corresponding to z_{t-1} to z_t. Consider an example where two successive, noisy GPS signals z_1, z_2 could be associated with edges e_1 or e_2 and e_3 or e_4 respectively. The probability of the most likely route taken by the vehicle, p_{route} based on Equation 1 is given by the maximum of the possible joint probability distributions which are calculated as shown in Equations 2 to 4. As noted in Section 2, the unknown quantities namely the emission and transition probabilities need to be computed.

$$p_{route} = max\Bigg(p(z_1,z_2,e_1,e_3), p(z_1,z_2,e_1,e_4), p(z_1,z_2,e_2,e_3), p(z_1,z_2,e_2,e_4)\Bigg) \tag{2}$$

Expanding the terms we get,

$$\begin{aligned} p_{route} = max\Bigg(&(p(e_1) \times p(z_1|e_1) \times p(e_3|e_1) \times p(z_2|e_3)), (p(e_1) \times p(z_1|e_1) \times p(e_4|e_1) \times p(z_2|e_4)), \\ &(p(e_2) \times p(z_1|e_2) \times p(e_3|e_2) \times p(z_2|e_3)), (p(e_2) \times p(z_1|e_2) \times p(e_4|e_2) \times p(z_2|e_4))\Bigg) \end{aligned} \tag{3}$$

We now assume that $p(e_t)$, i.e., the probability of a vehicle being on any of the roads in the proximity of the GPS sample is the same. Hence the above formula reduces to

$$\begin{aligned} p_{route} = max\Bigg(&(p(z_1|e_1) \times p(e_3|e_1) \times p(z_2|e_3)), (p(z_1|e_1) \times p(e_4|e_1) \times p(z_2|e_4)), \\ &(p(z_1|e_2) \times p(e_3|e_2) \times p(z_2|e_3)), (p(z_1|e_2) \times p(e_4|e_2) \times p(z_2|e_4))\Bigg) \end{aligned} \tag{4}$$

Assuming a Gaussian noise (Newson and Krumm 2009) with standard deviation of σ_z meters, the emission probabilities for each of the edges e_t associated with a given GPS measurement z_t are computed as follows

$$p(z_t|e_t) = \frac{1}{\sigma_z\sqrt{2\pi}} \exp(-0.5(\frac{d_p(z_t,e_t)}{\sigma_z})^2) \tag{5}$$

where $d_p(z_t,e_t)$ represents the perpendicular distance from the GPS sample z_t to edge e_t.

For computing the transition probabilities $p(e_t|e_{t-1})$, we make use of the previous work in (Newson and Krumm 2009). We compute the absolute value of the great circle distance between successive GPS samples from a vehicle and the route length traversed along road network graph d_t as given by

$$d_t = \mid gc(z_{t-1},z_t) - routeLen(e_{t-1},e_t) \mid \tag{6}$$

The *routeLen* function measures the length of the shortest path from vertex $e_{t-1}.v_{end}$ to the vertex $e_t.v_{begin}$. The *gc* function computes the great circle distance between any two points on the earth using the haversine formula. Figure 3 shows the probability density of d_t at different sampling intervals. The density as noted in (Newson and Krumm 2009) fits an exponential distribution given by $p(d_t) = \frac{1}{\beta}e^{\frac{-d_t}{\beta}}$. The parameter β as suggested in (Newson and Krumm 2009) is estimated as given in Equation 7 where $D_\delta = (d_t|t = 1...n)$ represents a set of d_t for given a sampling frequency. The value for β has been determined for different sampling intervals in order to ensure better and consistent estimates (represented as β_δ). The concrete algorithm for a stream based map-matching using sliding window of size two is given in Algorithm 1.

$$\beta_\delta = \frac{1}{ln(2)} median(D_\delta) \tag{7}$$

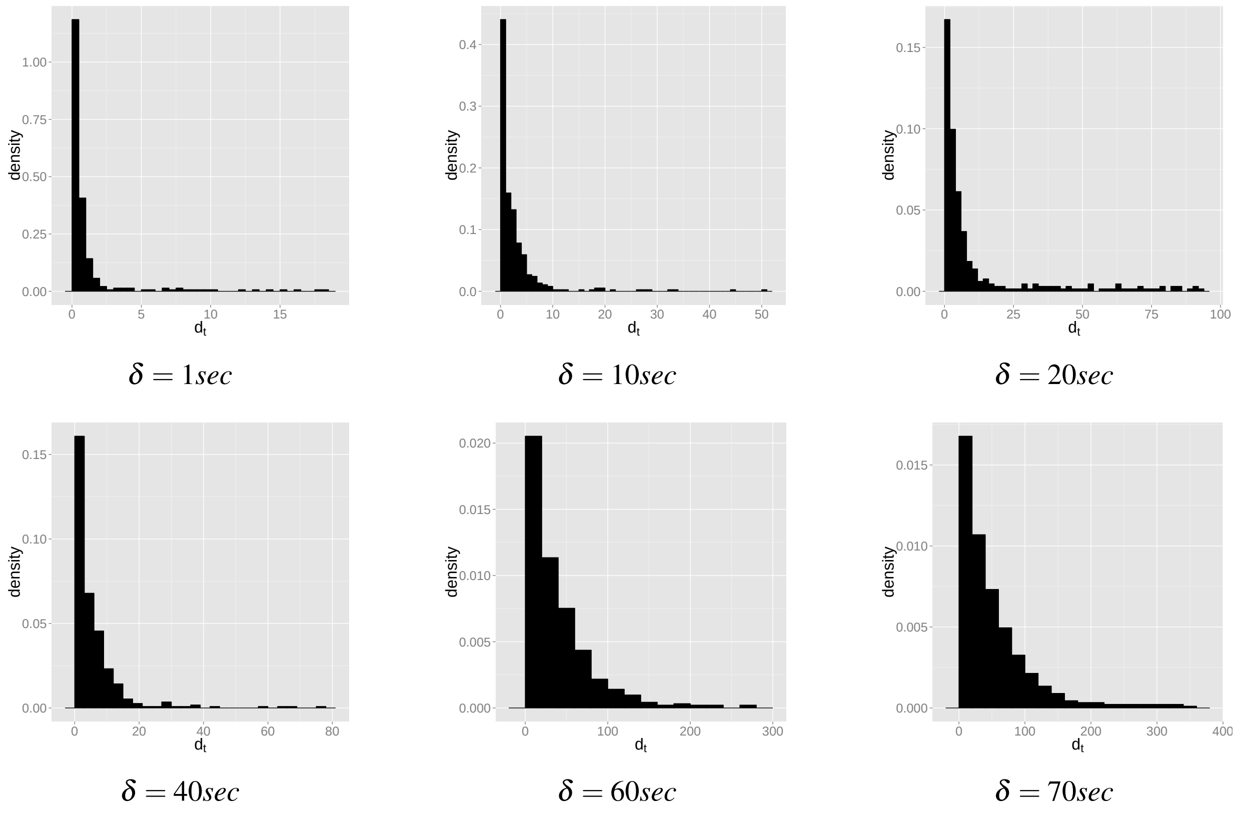

Figure 3: Probability density of d_t at different sampling intervals

3.2 Optimizations for latency reduction

For rapid initialization of the symbiotic traffic simulation, we need to ensure that the latency associated with the map-matching algorithm is as low as possible. In this section we describe the steps taken for reducing latency for computing the emission and transition probabilities by partitioning the road network using QuadTrees.

Figure 4 shows the road network of Singapore partitioned using QuadTrees (Finkel and Bentley 1974). Note that Figure 4 shows only the leaf nodes of the QuadTree. A QuadTree is a hierarchical data-structure with each node having exactly four children. QuadTrees are regularly used for partitioning two dimensional space for answering nearest neighbor queries efficiently. In comparison to a naive grid based partitioning, QuadTree partitioning takes into account the topology of the graph where denser regions are split into more regions in comparison to less dense regions. A node in the QuadTree is split into four equally sized children (or quadrants in the context of the 2 dimensional space) when the number of vertices in the node exceeds a predetermined limit. For the our road network containing roughly 44,000 vertices (and 88,000 edges) a node belonging to the QuadTree was split when the number of vertices associated with it exceeds 50.

Partitioning the road network helps us to reduce the time required in finding the edges in the proximity of a GPS sample by limiting the search space to only those edges contained within the leaf node of the QuadTree into which the GPS sample falls. Of the edges belonging to a partition, the closest eight edges are chosen for evaluation. Further, partitioning the road network helped us to speed up the process of evaluating the likely route alternatives for computing the transition probabilities for the HMM. For the shortest path computations used while estimating the transition probabilities, we used the arc-flag approach (Möhring et al. 2007) to minimize the number of edges evaluated by the Dijkstra's algorithm. During the initial

Algorithm 1 Map matching

Input:

- z_1, z_2 the two successive GPS samples from a single vehicle.
- $G(V,E)$ the road network graph.
- σ_z standard deviation of GPS noise in meters.
- β_δ the robust estimator for transition probabilities at the sampling interval δ .

Result: Route from most probable edge $e_1 \in E_1$ to most probable edge $e_2 \in E_2$. Where E_1 and E_2 represent the set of edges in the proximity of z_1 and z_2 respectively.

```
1:  $Mat_{rp} = \emptyset$
2:  $Mat_r = \emptyset$
3:  $E_1$ = Determine set of 8 closest edges to $z_1$
4:  $E_2$ = Determine set of 8 closest edges to $z_2$
5:  for $e_1 \in E_1$ do
6:      $p(e_1|z_1) = \frac{1}{\sigma_z\sqrt{2\pi}}\exp(-0.5(\frac{d_p(z_1,e_1)}{\sigma_z})^2)$
7:      for $e_2 \in E_2$ do
8:          $p(e_2|z_2) = \frac{1}{\sigma_z\sqrt{2\pi}}\exp(-0.5(\frac{d_p(z_2,e_2)}{\sigma_z})^2)$
9:          $d_t = \mid gc(z_1,z_2) - routeLen(e_1.v_{end}, e_2.v_{begin}) \mid$
10:         $p(e_2|e_1) = \frac{1}{\beta_\delta}e^{\frac{-d_t}{\beta_\delta}}$
11:         $Mat_{rp}[e_1][e_2] = p(e_1|z_1) \times p(e_2|z_2) \times p(e_2|e_1)$
12:         $Mat_r[e_1][e_2] = shortestPath(e_2 \leftarrow e_1)$
13: $rowIndex, colIndex = argmax(Mat_{rp})$
14: return $Mat_r[rowIndex][colIndex]$
```

preprocessing of the road network, we create an in-memory edge-partition(s) mapping. An edge is associated with a partition if it is part of a shortest path leading to the partition. While computing the shortest paths from a vertex belonging to one partition to another vertex belonging to a different partition, only those edges leading to the destination partition are considered. This results in much faster shortest path computations. As detailed in (Möhring et al. 2007), the arc-flag approach lends itself perfectly for graphs partitioned using QuadTrees.

4 EVALUATION

4.1 Synthetic GPS data stream

First and foremost a synthetic GPS data stream at different sampling intervals and with varying Gaussian noise was generated for performing the experimental analysis owing to the lack of vehicle trajectory data for Singapore. For generating a GPS stream at a given sampling frequency of δ seconds, we make use of the speed range information available for all edges constituting the road network. Given a speed range of $S_{i1}[\frac{m}{s}]$ to $S_{i2}[\frac{m}{s}]$ for an edge e_i, the distance traveled by the vehicle before emitting the next GPS location signal is taken to be $S_{i2} \times \delta$ meters. On reaching an edge e_j with different values for S_{j1} and S_{j2} the distance traveled is updated accordingly as $\delta \times S_{j2}$ meters. The steps are repeated for 50 randomly chosen routes at different sampling frequencies. Once the noise free GPS signals are created for 50 routes at different sampling frequencies, we add Gaussian noise of zero mean and varying standard deviation of σ_z meters at random angle θ between 0° and 360° from the original location. Figure 5 shows a GPS signal shifted from its original location of e_1 to z_1 due to noise.

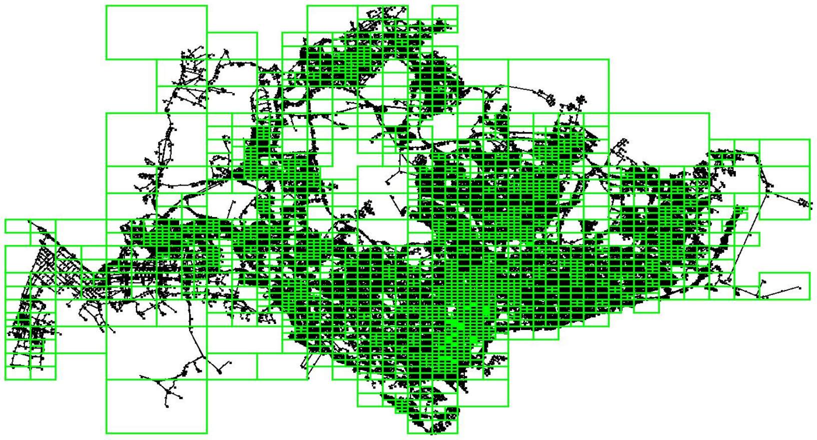
Figure 4: Road network of Singapore partitioned using QuadTrees

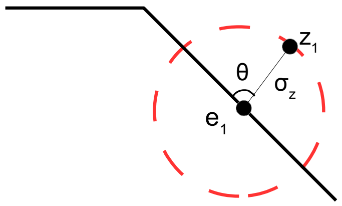

Figure 5: Shifted GPS position due to noise

4.2 Experimental results

In this section we describe the sequence of steps used to evaluate the efficacy of the map-matching algorithm discussed in Section 3. We are concerned with two aspects of performance, namely the reliability and the latency when identifying the correct road segment the vehicle(s) is driving on. For measuring reliability, consider a case where a vehicle emits n GPS signals from its origin to destination. The entire route then consists of $n-1$ segments. The chosen road segment thus represents the shortest path from the most probable edge e_{t-1} associated with the GPS data-point z_{t-1} to the most probable edge e_t associated with z_t. Hence reliability measures the percentage of correctly mapped segments along the entire route. Figure 6 shows the reliability of the algorithm under varying sampling intervals when the standard deviation of Gaussian noise increases. For the evaluation, the number of routes considered was 50 and the results were averaged over 10 repetitions.

Computational latency measures the time interval between the arrival of the second GPS signal (two successive GPS signals are necessary to map the correct road segment the vehicle is traversing upon) and the time at which the algorithm emits the result of the matched road segment irrespective of the result correctness. Figure 7 shows the average latency values for matching a single segment under varying noise and sampling intervals. The results are once again averaged over 50 routes and 10 repetitions for a given sampling frequency and noise.

4.3 Discussion of results

An example of the map-matching performed using the algorithm discussed in Section 3 is illustrated in Figure 8. The interval between successive samples is 20 seconds with the Gaussian noise having a standard deviation of 11 meters 0 mean. Computing the emission and transmission probabilities results in the algorithm mapping GPS samples A and B to the incorrect road segment (marked in red) instead of the

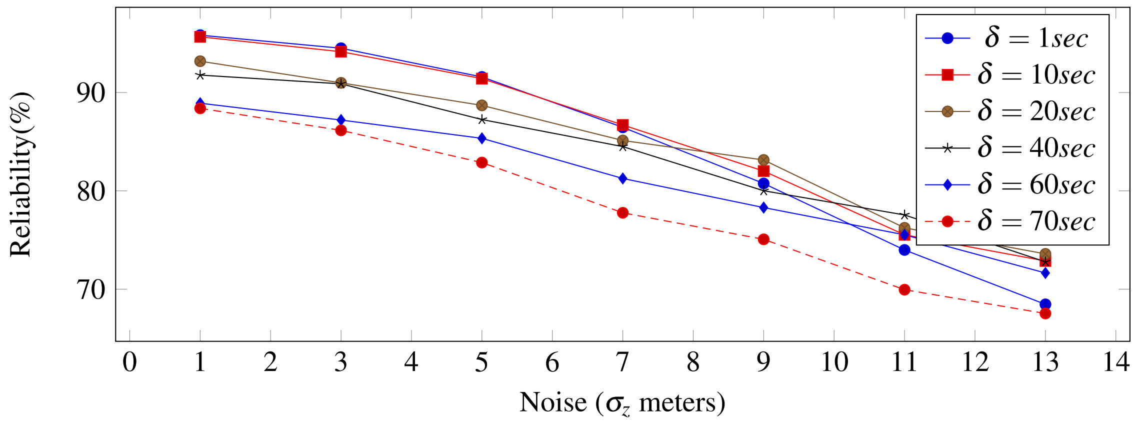

Figure 6: Noise vs sampling interval vs reliability

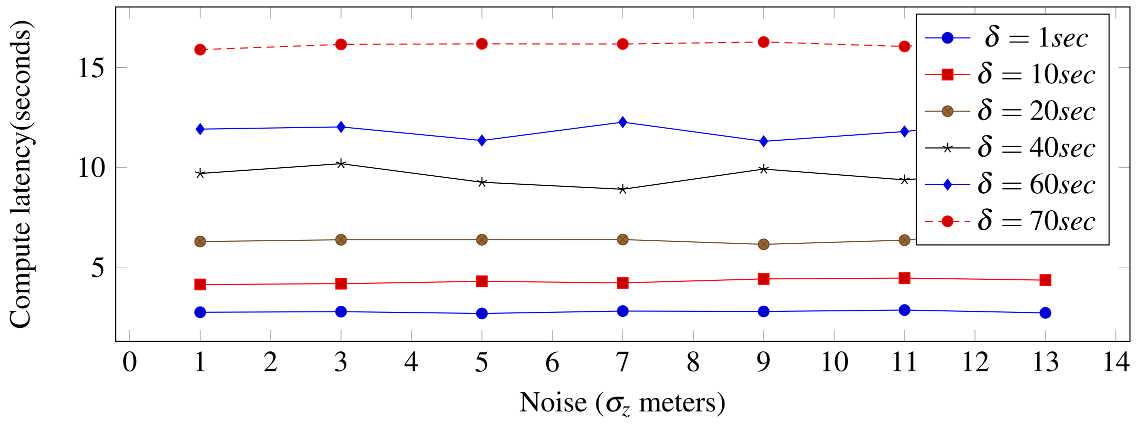

Figure 7: Noise vs sampling interval vs compute latency

correct one marked in green. The error is corrected once the next GPS sample C is received. Note that the emission (and transition) probabilities are computed once more for B along with point C to identify the correct road segment. Not retaining the previous results thus enables the algorithm to recover from wrongly matched road segments. Finally we have not considered errors where the GPS signal gets mapped to a wrong partition due to noise. We propose to correct this error by considering multiple partitions if the point lies close to the boundary of two or more partitions as a part of future work.

The second crucial aspect that needs discussion is that of computational latency. The results in Figure 7 indicate that the latency is noise invariant for a given sampling frequency. This is not surprising since the most probable segment is emitted without checking for result correctness. The latency increases as sampling frequency increases since the shortest path computations for transition probabilities are higher for vertices farther apart on the road network. Considering that we plan to implement a parallel version of the algorithm to match the position of several vehicles from a high velocity data-stream, it would be very useful to get a good idea about the latency. Given a scenario of 100,000 vehicles emitting GPS signals every second, despite the high degree of accuracy at low noise (Figure 6) it would be infeasible to implement the algorithm since the latency for map-matching exceeds 2 seconds resulting in 200,000 more records to be kept in memory even before we have the first segment for the vehicles mapped. For applications dealing with high velocity data-streams, large processing latencies could result in memory bottlenecks.

Considering a high degree of accuracy (even at reasonably high levels of noise) and less computational latency in comparison to the sampling intervals, GPS streams with $\delta = 10$ seconds to $\delta = 40$ seconds appear

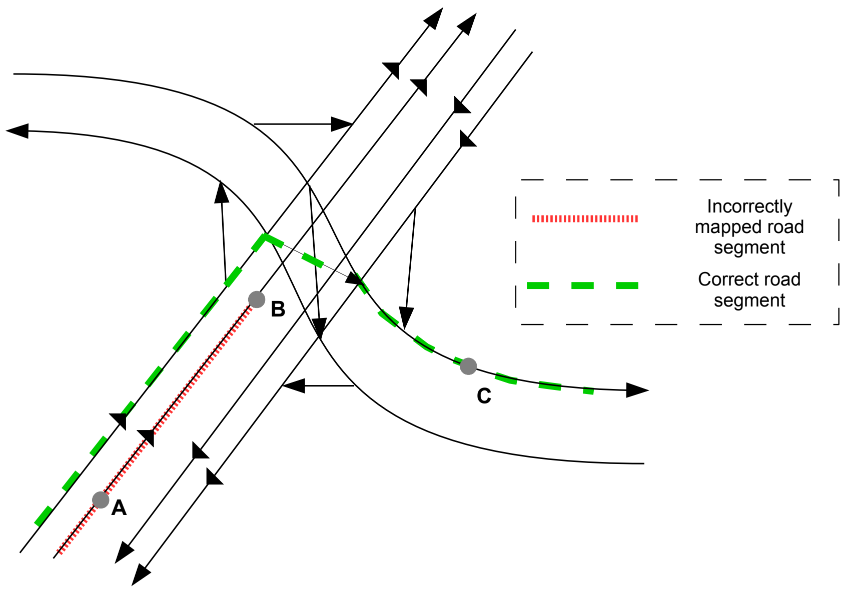

Figure 8: Source of errors while determining the path followed by the vehicle

to be the most effective. For $\delta < 10$ seconds, intermediate records can be filtered to achieve reasonable latency without compromising on accuracy.

5 CONCLUSIONS AND FUTURE WORK

In this paper we described and evaluated a map-matching algorithm which maps a GPS data-stream to a digital road network. We also implemented a couple of optimizations which reduce the latencies associated with computing the emission and transmission probabilities by using QuadTrees and a heuristic based on shortest path computations respectively. The results indicate that the reliability was in the range of $85\% - 92\%$ when δ was in the range of 10 seconds to 40 seconds and standard deviation of noise was in the range of 5 meters to 9 meters. The aforementioned noise range is based on the data-analysis of large scale GPS probe data in (Wang et al. 2011).

Though the experiments were performed and evaluated for the GPS data emitted by a single vehicle, we expect a parallel version of the algorithm to do the map-matching task over hundreds to thousands of vehicles to initialize a city-scale real-time symbiotic traffic simulation. As a part of future work, we plan to implement a parallel version of the presented algorithm using stream-processing engines such as STORM (Marz 2012). Finally from the perspective of a traffic simulation we plan to update the probability of vehicle taking an edge at a split on the road network (see Figure 9) based on historic information. The edge probabilities at splits can than be used to enhance the accuracy of the transition probabilities.

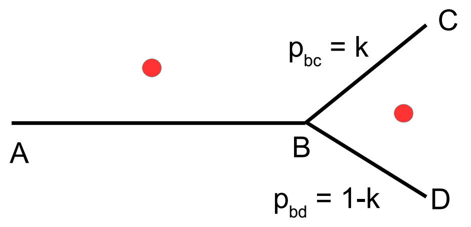

Figure 9: Update probability of vehicle taking an edge at split based on count.

ACKNOWLEDGMENTS

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

REFERENCES

Aydt, H., M. Lees, and A. Knoll. 2012. “Symbiotic simulation for future electro-mobility transportation systems”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Finkel, R. A., and J. L. Bentley. 1974. “Quad trees a data structure for retrieval on composite keys”. *Acta Informatica* 4 (1): 1–9.

Fujimoto, R., D. Lunceford, and E. Page. 2002. “Grand challenges for modeling and simulation”. Technical Report No. 350, Dagstuhl report, Schloss Dagstuhl. Seminar No 02351.

Goh, C. Y., J. Dauwels, N. Mitrovic, M. Asif, A. Oran, and P. Jaillet. 2012. “Online map-matching based on hidden markov model for real-time traffic sensing applications”. In *Proceedings of 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2012*, 776–781.

Jagadeesh, G., T. Srikanthan, and X. Zhang. 2004. “A map matching method for GPS based real-time vehicle location”. *Journal of Navigation* 57 (03): 429–440.

Krumm, J., J. Letchner, and E. Horvitz. 2007. “Map matching with travel time constraints”. In *(SAE) 2007 World Congress*. Detroit, MI USA.

Marz, Nathan 2012. “STORM: Distributed and fault-tolerant real-time computation”. http://storm.incubator.apache.org/.

Möhring, R. H., H. Schilling, B. Schütz, D. Wagner, and T. Willhalm. 2007. “Partitioning graphs to speedup Dijkstra’s algorithm”. *Journal of Experimental Algorithmics (JEA)*:2–8.

Newson, P., and J. Krumm. 2009. “Hidden Markov map matching through noise and sparseness”. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 336–343.

Obradovic, D., H. Lenz, and M. Schupfner. 2006. “Fusion of map and sensor data in a modern car navigation system”. *Journal of VLSI signal processing systems for signal, image and video technology*:111–122.

Quddus, M. A., W. Y. Ochieng, and R. B. Noland. 2007. “Current map-matching algorithms for transport applications: State-of-the art and future research directions”. *Transportation Research Part C: Emerging Technologies* 15 (5): 312–328.

Rabiner, L. 1989. “A tutorial on hidden Markov models and selected applications in speech recognition”. *Proceedings of the IEEE* 77 (2): 257–286.

Wang, Y., Y. Zhu, Z. He, Y. Yue, and Q. Li. 2011. “Challenges and opportunities in exploiting large-scale GPS probe data”. Technical Report No. 109, HP Laboratories.

Yin, H., and O. Wolfson. 2004. “A weight-based map matching method in moving objects databases”. In *Proceedings of 16th International Conference on Scientific and Statistical Database Management*, 437–438. IEEE Computer Society, Washington, DC, USA.

AUTHOR BIOGRAPHIES

ABHINAV SUNDERRAJAN is a Research Associate at TUM CREATE Ltd., a company dedicated to research in all aspects of electro-mobility. He received his Masters in Computing from the National University of Singapore. His email address is abhinav.sunderrajan@tum-create.edu.sg.

HEIKO AYDT received his Ph.D. degree in Computer Science from Nanyang Technological University (NTU), Singapore and his M.Sc. degree from the Royal Institute of Technology (KTH) in Stockholm. In 2006 he joined the Parallel and Distributed Computing Centre at NTU as Research Associate where he worked on several projects concerned with simulation-based optimisation and agent-based crowd simulation. In 2011, he joined TUM CREATE as Research Fellow and as of December 2012, he assumed the role of Principal Investigator for RP 5. His research at TUM CREATE is concerned with the analysis of the potential impact of electromobility on the infrastructure and the environment from a complex systems perspective. His current research interests are agent-based simulation, complex adaptive systems, symbiotic simulation, evolutionary computing and simulation-based optimisation. His email address is heiko.aydt@tum-create.edu.sg.

WENTONG CAI is a Professor in the School of Computer Engineering at Nanyang Technological University, Singapore. He is also the Director of the Parallel and Distributed Computing Centre. His expertise is mainly in the areas of Modeling and Simulation (particularly, modeling and simulation of large-scale complex systems, and system support for distributed simulation and virtual environments) and Parallel and Distributed Computing (particularly, Cloud, Grid and Cluster computing). His web page is http://www.ntu.edu.sg/home/aswtcai/ and his email address is aswtcai@ntu.edu.sg.

ALOIS KNOLL received his diploma (MSc) degree in Electrical/Communications Engineering from the University of Stuttgart and his PhD degree in Computer Science from the Technical University of Berlin. He served on the faculty of the computer science department of TU Berlin until 1993, when he qualified for teaching computer science at a university (habilitation). He then joined the Technical Faculty of the University of Bielefeld, where he was a full professor and the director of the research group Technical Informatics until 2001. Between May 2001 and April 2004 he was a member of the board of directors of the Fraunhofer-Institute for Autonomous Intelligent Systems. At AIS he was head of the research group ”Robotics Construction Kits”, dedicated to research and development in the area of educational robotics. Since autumn 2001 he has been a professor of Computer Science at the Computer Science Department of the Technische Universität München. He is also on the board of directors of the Central Institute of Medical Technology at TUM (IMETUM-Garching); between April 2004 and March 2006 he was Executive Director of the Institute of Computer Science at TUM. His research interests include cognitive, medical and sensor-based robotics, multi-agent systems, data fusion, adaptive systems and multimedia information retrieval. His email address is knoll@in.tum.de.