Compact Vehicular Trajectory Encoding

Markus Koegel[*] Wolfgang Kiess[*] Markus Kerper[‡] Martin Mauve[*]
[*]Department of Computer Science, University of Düsseldorf, Germany [‡]Volkswagen AG, Wolfsburg, Germany
{ koegel, kiess, mauve }@cs.uni-duesseldorf.de markus.kerper@volkswagen.de

***Abstract*—Many applications in vehicular communications require the collection of vehicular position traces. So far this has been done by recording and transmitting unencoded or merely linearly filtered position samples. Depending on the sample frequency and resolution, the resulting data load may be very large, consuming significant storage and transmission resources. In this paper, we propose a method based on two-dimensional cubic spline interpolation that is able to reduce the amount of the measurement data significantly. Our approach allows for a configurable accuracy threshold and performs in $\mathcal{O}(n^3)$. We evaluate our approach with real vehicular GPS movement traces and show that it is able to reduce the volume of the measurement set by up to 80 % for an accuracy threshold of 20 centimeters.**

I. Introduction

In vehicular (V2X) communications, applications are designed to improve safety, efficiency and convenience of daily traffic. To this end, traffic participants run applications on their on-board computing units to collect and exchange information.

Depending on the context of the particular application, this information ranges from mere collections of location waypoints for map refinement [1] to detailed *Floating Car Data (FCD)*. The latter combines trajectory descriptions with corresponding timestamps or sensor measurements to provide up-to-date information about traffic conditions. Based on such detailed measurements, sophisticated applications like traffic information systems [2] or roadway condition monitoring [3] can be implemented. Despite the diversity of these applications, their functionality often depends on knowledge about the routes being driven by their particular users.

Up to now, the common way to encode trajectories is to concatenate waypoint coordinates to position sequences [4]–[6]. When such *polygonal chain* trajectory descriptions need to be transmitted, a large amount of network resources may be occupied. At the same time, due to the principles of kinematics, such polygonal descriptions contain a large amount of redundancy and thus bear a high potential for compression.

In this paper, we focus on the problem of the efficient encoding of vehicular trajectories and propose a lossy compression scheme for this kind of information. Our approach exploits the specific characteristics of vehicular mobility and allows for a configurable accuracy. Even though we focus on trajectories as mappings of a progress variable onto a position, we also briefly look at adding non-geometrical information to this mapping.

Our core idea is to exploit the smooth nature of vehicular movement due to the principles of kinematics and that cubic splines are an excellent fit to approximate them. We present an algorithm utilizing cubic splines to calculate compact vehicular trajectory descriptions for configurable accuracy thresholds. We show that our assumptions hold and that our approach achieves significant compression rates even for tight accuracy bounds by an evaluation based on real GPS vehicle trajectory measurements. An extended version of this paper including more detailed discussions and results can be found at [7].

In the remainder of this paper, we describe the fundamentals of vehicular trajectories and cubic splines to encode them in Section II. We review related work in Section III before describing an efficient encoding algorithm for vehicular movement traces by spline interpolation in Section IV. Section V evaluates our approach on the basis of extensive real-world GPS measurements. Finally we conclude this paper in Section VI with a summary and an outlook on future work.

II. Vehicular Trajectories

In this section, we discuss the nature of vehicle movement and the characteristics of movement measurements. We then summarize mathematical tools for the representation of vehicular trajectories and assess their suitability for our demands.

A. Positioning

A wide-spread technique for self-positioning is the use of *Global Navigation Satellite Systems (GNSS)*, such as GPS. GNSS position measurements, in general, do not return the receiver's true position. Several (e. g. shadowing) effects may cause temporarily stable biases, while e. g. signal multipath propagation or reflection usually causes an error which is assumed to be Gaussian distributed in the literature [8].

However, the more information sources are being monitored, the higher is the achievable measurement accuracy. For the *Differential GPS* technology, for instance, a network of GPS receivers monitors the systemic GPS error to calculate a correction code. In this way, the error can be reduced from more than ten meters to less than one meter [9]–[11].

Position measurements can only be taken at a specific frequency, and thus have a snapshot character. Then, *movement measurements* are merely chronologically sequenced static measurements that can be interpreted as a progression of a target's position over time. The result is a *polygonal chain*, whose accuracy heavily depends on the snapshot frequency.

B. Vehicular Trajectory Volume Reduction

Vehicle movement is usually modeled based on the principles of kinematics [12], covering e. g. detailed calculations of longitudinal and lateral accelerations by the target's speed, acceleration and steering angle. These parameters can be described by smooth functions, since their changing rates are continuous. This implies that vehicle trajectories can also be expressed by smooth[1] functions $s \in C^2$, mapping a progress variable to a geographical position: $s : \mathbb{R} \rightarrow \mathbb{R}^2, s(t) = (x, y)$ The smoothness translates to redundancy underlying the position measurements and we propose to utilize this to compress the information about a vehicle's trajectory.

We thus need a generic data structure for the trajectory representation that exploits the identified characteristics and can be employed for a wide set of applications. Still, this structure has to meet two conditions: 1) the original measurements need to be restorable, as they are the only directly gathered knowledge. Anything else is, at best, a reasonably good interpolation between those measurements. 2) Due to the afore-mentioned GNSS measurement uncertainties, the position measurements feature inaccuracies. For this reason, it is perfectly acceptable to approximate the measurements with the data structure instead of retaining them exactly. However, as different applications require different degrees of accuracy, the maximum error threshold for this must be configurable.

We have identified cubic splines to fit these conditions very well: in numerics, spline interpolation is employed to create a smooth concatenation of curves traversing a sequence of data points (*knots*). To this end, polynomials are defined between two successive knots under certain conditions, such as C^n continuity for *n+1-th degree splines* at the knots. For several reasons, *cubic* splines fit our demands: first, C^2 continuity ensures a realistic modeling of steering behavior. Second, cubic splines provide a minimal curvature between two successive knots, resulting in an especially low oscillation behavior and special smoothness of the final spline curve. Third, since a spline definition includes the number of skipped points between two remaining knots, these points can be reconstructed with minimal effort. Finally, with this systematic reconstruction, an error threshold can easily be implemented.

We will focus on the interpolation of vehicle trajectories with cubic splines after a brief overview of related work.

III. Related Work

For calculation of autonomous vehicle tracks, trajectories are modeled with Bézier curves [13], splines [14] or clothoids [15]. Though this underlines the suitability of such structures to approximate trajectories, these works base on the exact knowledge of the road geometry and do not allow conclusions for how to construct approximate representations, given only noisy position measurements.

Such measurements form the basis of object tracking with *Mobile Objects Databases (MOD)*: an MOD receives position updates from mobile units, the transmissions of which are triggered based on time intervals or assumed position estimation errors. For storage and communication load reduction, trajectory approximation mechanisms are applied during the update process. Mobile units can employ *offline* update mechanisms, such as [16], [17], thus separating the collection and compression of trajectory data. The most promising offline approaches perform heuristic or optimal line simplifications [18], [19] or utilize linear dead reckoning (LDR) mechanisms. Alternatively, *online* update mechanisms run in real-time (cf. [20], [21]) and mostly use LDR and simple line approximations. However, none of these works consider using non-linear trajectory approximations.

As far as we know, the only previous work on non-linear trajectory approximation was presented in the context of spatio-temporal data base indexing. In [22], the authors motivate that a good representation for spatial vehicular trajectories are so-called *minimax* polynomials, that approximate an original function, such that the maximum approximation error is minimal for the used parameter set. They suggest using Chebyshev polynomials that are very good approximations of optimal minimax polynomials. The authors of [23] extend this work by adding the temporal dimension to trajectory descriptions. Both contributions, however, use the Chebyshev polynomial degree as only input parameter and calculate the resulting approximation error after the approximation. They do not present an efficient way of constructing a Chebyshev polynomial representation for vehicular trajectories, given an upper error bound ϵ only.

IV. Trajectory Interpolation

We now discuss the interpolation of pure geometric data with two-dimensional cubic splines and present an algorithm that finds a locally optimal solution for the error-aware reduction of a spline knot sequence. Since sophisticated V2X applications report more data, we also show that non-geometric data can be easily and present a very efficient way to encode spatio-temporal information sets by spline interpolation.

A. Basic Trajectory Interpolation

First, we provide an algorithm to compactly encode vehicular trajectories with cubic splines. We will describe trajectories as measurement sequences $(m_i)_{i \in \mathbb{N}_0}$, $m : \mathbb{N}_0 \mapsto \mathbb{R}^2$, mapping an index i onto a *measurement tuple*, where each element m_i contains (for now) only a measurement's geodetic information. and is therefore referred to as *position*.

To encode vehicle movement, we approximate the input dimensions separately and thus seek a two-dimensional cubic spline that interpolates every taken position measurement. Given an accuracy bound ϵ, we compress the trajectory (m_i) by removing redundant elements that can later be reconstructed by the spline without violating the accuracy bound. In the following, we will refer to this subset as (m'_i).

We now can state the problem as follows: given a sequence of position measurements $(m_i)_{i \in \mathbb{N}_0}$ and $\epsilon \geq 0$. What is the minimal knot subsequence (m'_i), by which (m_i) can be interpolated with cubic splines and an upper bound ϵ for the interpolation error at every index $i \in [0, |(m_i)| - 1]$?

To find this globally optimal solution (m'_i), one would have to determine the interpolation error for every subsequence of (m_i) and then select the smallest one with an interpolation error not exceeding ϵ. Given the measurement sequence's length $n = |(m_i)|$, there are 2^n mutually distinct subsequences

[1] A function f is called *smooth* (i. e. $f \in C^n$), iff the first n derivatives $f', \ldots, f^{(n)}$ exist and are continuous.

of (m_i). Since the interpolation error calculation runs in $\mathcal{O}(n)$, the overall complexity of finding the globally optimal solution lies within $\mathcal{O}(n \cdot 2^n)$. In V2X communications, however, measurement sequences may easily feature $n > 100$, making the described naive approach unfeasible. Instead, we now present an algorithm running in $\mathcal{O}(n^3)$ that finds an approximation of this optimal solution, denoted by $(\widetilde{m}'_i)$.

Our algorithm uses a greedy iterative search to reduce (m_i) down to $(\widetilde{m}'_i)$: given a measurement sequence (m_i) and an error bound $\epsilon \geq 0$, the algorithm checks in each iteration for every but the first and last remaining element in (m_i), what the maximum resulting interpolation error will be, if this element is removed from the sequence. The element with the smallest maximum error is then removed from (m_i). The algorithm stops once no further element can be removed without violating the error bound ϵ and returns the reduced sequence $(\widetilde{m}'_i)$ with length $n' = |(\widetilde{m}'_i)|$. We denote the *reduction fraction* by $\sigma = \frac{n'}{n}$.

To reconstruct the knots removed from (m_i) at the correct positions, we need to remember their original indices, e. g. with a bit field of length n, where each bit indicates whether the respective measurement has been kept. This implies a small extra effort of only one bit per measurement.

B. Adding Non-Geometric Data

More sophisticated applications often extend the mere knowledge about vehicular movement by e. g. the sequence of points in time, at which the corresponding measurements have been taken. This allows, for instance, to conclude on the road segment's position within a fundamental diagram or to monitor the average speed for it as in [2]. For other projects, such as [3], additional measurements of temperature, humidity and friction coefficients need to be combined and transmitted.

Note that though we consider cubic splines exceedingly useful for trajectory interpolation in particular, the reduction technique should always fit the particular context. For example, friction parameters might be interpolated by a much simpler construct than cubic splines. Furthermore, the dimensionality d of the input data is irrelevant for the algorithm's asymptotic runtime complexity. As long as the error determination per reduction step does not exceed $\mathcal{O}(n)$, the algorithm's complexity remains in $\mathcal{O}(n^3)$, since a fixed dimensionality does not affect its asymptotic behavior. The only supplement that has to be added to the algorithm is a further error threshold for each new dimension or combination of new dimensions.

As already mentioned earlier, the remainder of this paper focuses on temporal information as non-geometric data due to its special interrelation with spatial information.

C. Exploiting the Space-Time Interrelation

We have stated that a vehicular trajectory can be expressed as a mapping of a progress variable to a position. Basically, the particular measurement's points in time can be employed as such a progress variable. However, to fulfill the first condition from Section II-B, all measurements need to be taken with a constant measurement frequency. With this frequency and the first measurement's timestamp, all measurement timestamps in the sequence can be reconstructed. Otherwise, e. g. if location measurements have been lost before being logged or have not been logged with a constant frequency, the original measurements or their original positions in time cannot be reconstructed and irregular large temporal gaps may occur. In this case, the measurements can still be interpolated, but the interpolation errors for the respective knots cannot be guaranteed to lie within the bounds any more. However, even current off-the-shelf GPS hardware provides exact measurement frequencies and more sophisticated positioning systems, using e. g. Kalman filters [24] or inertial navigation systems [25], can easily accomplish this task anyway.

Due to the additional temporal information contained within the index bit field, we have turned the only overhead of our approach into payload, containing measurement information. In other words, an effective data reduction begins with the first removed element from the measurement sequence.

V. Evaluation

A. Data Acquisition

For our evaluation, we performed extensive vehicular GPS measurements in one city (8.9 km, 2086 measurement points) and two highway (23.4/14.9 km, 2021/1306 measurement points) topologies with a measurement frequency of 2 Hz.

The GPS traces, stored in an NMEA-0183-like format, provided a coordinate precision of six decimal places. This discretization implies a maximum rounding error of ≈ 6 cm for a latitude of 52°, at which we performed our measurements.

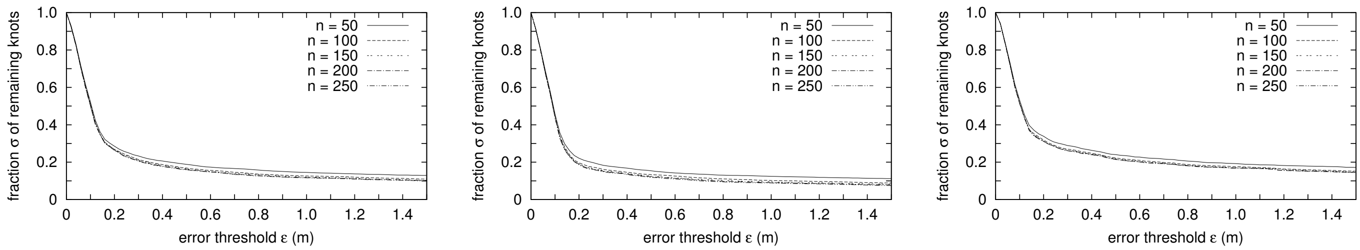
For real applications, static size buffers are an efficient technique to log and subsequently transmit position measurement traces. In our case, once the buffer is full, the data is passed to our algorithm. The output is then transmitted to a collector and the logging starts over anew.

We simulated the static size buffer using a sliding window approach: starting at the respective trace's beginning, we copied the window contents to according measurement subsequences. In Section IV, we have introduced such subsequences as $(m_i)_{i \in \mathbb{N}_0}$, so the window size directly translates into the original size of a measurement subsequence $n = |(m_i)|$. The window was then shifted by an offset $o = \frac{n}{4}$, thus increasing the number of regarded subsequences and avoiding boundary effects to make our evaluations as meaningful as possible.

We evaluated window sizes of 50-250 in steps of 50. If we assume two 32-bit floating point position measurements per element plus a 32-bit integer reference timestamp for the whole measurement sequence and another 32-bit floating point measurement frequency value, the resulting buffer sizes range from 416 byte ($n = 50$) to 1.97 KB ($n = 200$), showing that this approach is indeed feasible for the proposed window sizes.

B. Knot Sequence Reduction

The most meaningful performance criterion for our proposed algorithm is the fraction $\sigma = \frac{n'}{n}$ of remaining knots after the reduction. It is shown in Figure 1 for all topologies.



(a) City topology.

(b) Highway topology 1.

(c) Highway topology 2.

Figure 1. Knot reduction analysis for varying error thresholds and window sizes.

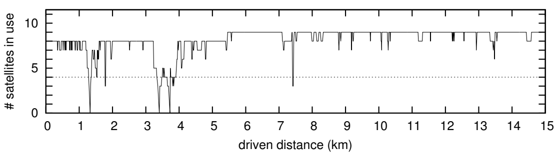


Figure 2. GNSS signal availability for highway topology 2.

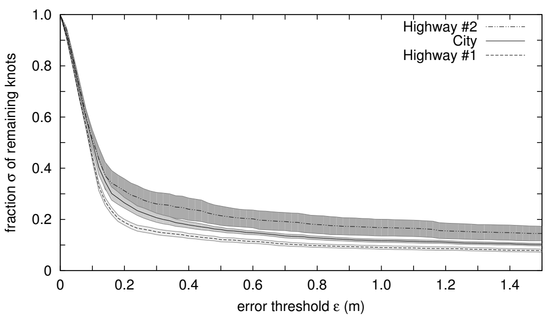


Figure 3. Knot reduction analysis for $n = 200$.

Each figure shows the average fraction σ over an increasing error threshold ϵ in steps of 2 cm and up to 1.5 m in total.

The first and most important visualized effects are steep declivities of σ for error thresholds up to approximately 20 cm. This results in an average knot sequence reduction of 70 % to 83 % at an interpolation error tolerance of only 20 cm and even a reduction of 83 % to 93 % for an error tolerance of 1.5 m. The steep declivity indicates an irregularity of the position measurements composed of the GPS noise and the coordinate discretization, keeping them from being perfectly smooth. Once the error threshold ϵ exceeds this noise level, our approach works very well, confirming our assumption that cubic splines are an excellent fit for vehicular trajectories.

Second, Figures 1(a) – 1(c) show that the window size has only a minor effect on σ. The results for $n = 50$ differ only marginally from the ones achieved with larger window sizes. Especially, there is no distinct difference for window sizes $n > 150$. We therefore fixed $n = 200$ for further evaluations.

Finally, the results for highway topology 2 are still very good but worse than the others. This is due to partial highway roofings that heavily impaired the GPS signal reception. Figure 2 depicts the number of used satellites over covered distance for this topology: the number of satellites is heavily unstable and even drops below four, which is the minimum number of satellites necessary for reasonably exact self-positioning. This caused a high positioning noise and a poor smoothness and continuity of the trajectory. For many subsequences in this topology, a good reduction is thus only achievable at a larger ϵ. This can also be seen in Figure 3, depicting our reduction results for $n = 200$ with 95 % confidence interval corridors. Solely the second highway topology has very wide confidence intervals which confirms a high variation of the achieved results.

C. Interpolation Error

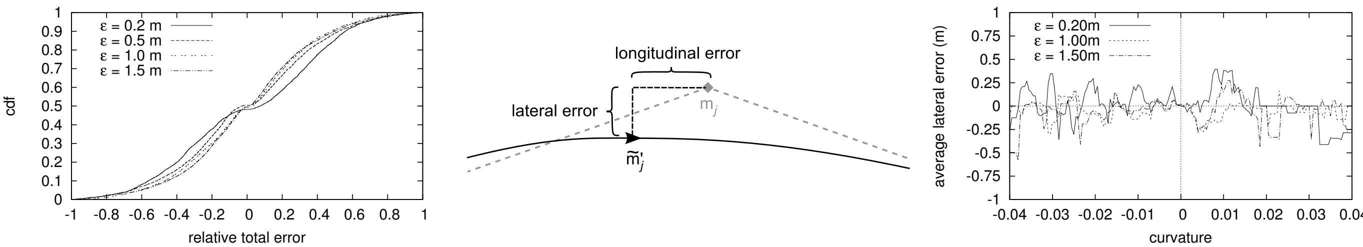
We have seen that even small interpolation error thresholds allow for high knot amount reductions. Next, we take a closer look on the nature and distribution of this interpolation error. Due to shortage of space, we use the first highway topology as an example, being representative for the other topologies.

Figure 4(a) shows the cumulative distribution of the relative interpolation error ρ (i. e. the ratio of the occurred interpolation error and ϵ.) for a selection of interpolation error thresholds. Negative interpolation errors occur, if the original point lies on the left hand side of the spline curve, in movement direction. The figure shows that positive and negative interpolation errors are distributed alike. This is also true for the other topologies, thus excluding topology-specific error behaviors. Second, we see that with increasing ϵ, the distribution's compactness grows around $0.005 \leq |\rho| \leq 0.4$. This is because the interpolation errors are not spread uniformly, but only few measurement points are interpolated with a close-to-maximum and close-to-minimum interpolation error. The latter results from a base noise and the discretization of position measurements.

The total interpolation error distribution is a good quality indicator for the trajectory interpolation. However, for a more thorough understanding, the interpolation error components need to be differentiated. We regard the longitudinal and lateral parts for our considerations: Figure 4(b) shows a gray polygonal chain and its black spline approximation as well as the error components for the interpolated knot $\widetilde{m}'_j$ of the original position measurement m_j. While the longitudinal interpolation error describes the divergence along the movement direction, the lateral error refers to the part perpendicular to it.

Since cubic splines provide a minimum curvature, systemic errors are possible to occur in relation to the spline's curvature. In this case, curves and bends would be interpolated tighter than they originally are and a clear correlation between the spline's curvature and the lateral error would exist.

To this end, Figure 4(c) depicts the lateral error in relation to the spline's curvature κ. Since κ is the inverse of the curve



(a) Total error distribution.
(b) Longitudinal and lateral error components.
(c) Lateral error analysis.

Figure 4. Error analysis using the representative example of the first highway topology.

radius, a high absolute curvature value translates into a tight curve or bend. The figure covers $\kappa \in [-0.04; 0.04]$, resulting in (for the considered velocities tight) curve radii of down to 25 m. However, no systemic errors are recognizable, but the errors seem to distribute arbitrarily, instead. This leads us to the assumption, that splines with their minimal oscillation may even reduce the lateral noise attached to GPS measurements. This is still subject to further investigations.

VI. Conclusions

In vehicular communications, a number of applications depend on the exchange of vehicular trajectory data. Up to now, mere position measurement concatenations (*polygonal chains*), that contain a high degree of redundancy and are thus far from being optimal, have been used for this. In this paper, we propose an encoding approach based on cubic spline interpolation and present a greedy algorithm that filters out redundant position measurements in $\mathcal{O}(n^3)$ for a adjustable error threshold ϵ. The omitted measurements can afterwards be retrieved by simple cubic spline interpolation with a guaranteed maximum interpolation error ϵ. We applied our algorithm to a large number of real-world GPS measurements and see that with an error tolerance of only 20 cm up to 80 % of the measurements can be removed in this way. The resulting error is not systemic, but it even appears that the spline interpolation can reduce the lateral noise from GPS measurements instead.

References

[1] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS Traces for Map Refinement," *Data Mining and Knowledge Discovery*, vol. 9, no. 1, pp. 59–87, July 2004.

[2] R.-P. Schäfer, K.-U. Thiessenhusen, E. Brockfeld, and P. Wagner, "A traffic information system by means of real-time floating-car data," in *Proceedings of the 9th World Congress and Excibition on Intelligent Transportation Systems and Services (ITS)*, Oct. 2002.

[3] J. Myllylä and Y. Pilli-Sihvola, "Floating car road weather monitoring," in *SIRWEC '02: Proceedings of the 11th International Road Weather Congress*, Jan. 2002.

[4] M. Fouladvand and A. Darooneh, "Statistical analysis of floating-car data: an empirical study," *The European Physical Journal B – Condensed Matter and Complex Systems*, vol. 47, no. 2, pp. 319–328, sep 2005.

[5] R. Brüntrup, S. Edelkamp, S. Jabbar, , and B. Scholz, "Incremental map generation with gps traces," in *ITSC '05: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2005, pp. 574–579.

[6] F. Kranke and H. Poppe, "Traffic Guard - Merging Sensor Data and C2I/C2C Information for proactive, Congestion avoiding Driver Assistance Systems," in *FISITA '08: World Automotive Congress of the Int'l Federation of Automotive Engineering Societies*, Sep. 2008.

[7] M. Koegel, W. Kiess, M. Kerper, and M. Mauve, "Compact Vehicular Trajectory Encoding (extended version)," Computer Science Department, Heinrich Heine University, Düsseldorf, Germany, Tech. Rep. TR-2010-002, Aug 2010.

[8] F. van Diggelen, "GPS Accuracy: Lies, Damn Lies and Statistics," *GPS World*, vol. 9, no. 1, pp. 41–45, November 1998.

[9] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications*. American Institute of Aeronautics & Astronomy, August 1996, vol. 2.

[10] L. S. Monteiro, T. Moore, and C. Hill, "What is the accuracy of DGPS?" *The Journal of Navigation*, vol. 58, pp. 207–225, 2005.

[11] U.S. National Oceanic and Atmosperic Administration: National Geodetic Survey, "GPS Accuracy Comparisons (online resource)," http://www.ngs.noaa.gov/FGCS/info/sans_SA/.

[12] T. D. Gillespie, *Fundamentals of Vehicle Dynamics*. SAE International, March 1992.

[13] J.-W. Choi and G. H. Elkaim, "Bézier curves for trajectory guidance," in *WCECS '08: Proceedings of the World Congress on Engineering and Computer Science*, Oct. 2008, pp. 625–630.

[14] C. Dever, B. Mettler, E. Feron, J. Popović, and M. Mcconley, "Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes," *Journal of Guidance, Control and Dynamics*, vol. 29, pp. 289–302, 2006.

[15] L. Labakhua, U. Nunes, R. Rodrigues, and F. S. Leite, "Smooth trajectory planning for fully automated passengers vehicles - spline and clothoid based methods and its simulation," Aug. 2006, pp. 89–96.

[16] A. Civilis, C. S. Jensen, and S. Pakalnis, "Techniques for efficient road-network-based tracking of moving objects," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 698–712, May 2005.

[17] H. Cao, O. Wolfson, and G. Trajcevski, "Spatio-temporal data reduction with deterministic error bounds," *The VLDB Journal*, vol. 15, no. 3, pp. 211–228, September 2006.

[18] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Canadian Cartographer*, vol. 10, no. 2, pp. 112–122, December 1973.

[19] H. Imai and M. Iri, "Computational-geometric methods for polygonal approximations of a curve," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 31–41, 1986.

[20] R. Lange, T. Farrell, F. Dürr, and K. Rothermel, "Remote real-time trajectory simplification," in *PerCom '09: Proceedings of the 7th IEEE International Conference on Pervasive Computing and Communications*, Galveston, TX, USA, march 2009, pp. 184–193.

[21] N. Hönle, M. Groddmann, S. Reimann, and B. Mitschang, "Usability analysis of compression algorithms for position data streams," in *GIS '10: Proceedings of the 18th ACM SIGSPATIAL international conference on Advances in Geographic Information Systems*, Nov. 2010.

[22] Y. Cai and R. Ng, "Indexing spatio-temporal trajectories with Chebyshev polynomials," in *SIGMOD '04: Proceedings of the ACM SIGMOD International Conference on Management of Data*, Jun. 2004.

[23] J. Ni and C. V. Ravishankar, "Indexing Spatio-Temporal Trajectories with Efficient Polynomial Approximations," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, pp. 663–678, May 2007.

[24] E. Brookner, *Tracking and Kalman Filtering Made Easy*. Wiley-Interscience, April 1998.

[25] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-696, August 2007.