Compact Vehicular Trajectory Encoding (extended version)

Markus Koegel* Wolfgang Kiess* Markus Kerper‡ Martin Mauve*
*Department of Computer Science, University of Düsseldorf, Germany
{ koegel, kiess, mauve }@cs.uni-duesseldorf.de
‡Volkswagen AG, Wolfsburg, Germany
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 position samples. Depending on the frequency and resolution of these samples, the resulting data may be very large, consuming significant resources for storage and transmission. 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 its task 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 with each other.

Depending on the context of the particular application, these information range 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], [3] or roadway condition monitoring [4] 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 [5]–[8]. When a given application needs to transmit such *polygonal chain* trajectory descriptions, a large amount of network resources may be occupied. At the same time, due to the principles of kinematics, polygonal descriptions of vehicular trajectories 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 will focus on trajectories as mappings of a progress variable onto a position, we will also briefly look at adding non-geometrical information to this mapping.

The core idea of our approach is to exploit the smooth and continuous nature of vehicular movements resulting from the principles of kinematics and that cubic splines are an excellent fit to describe such movements. We present an algorithm utilizing cubic splines to calculate compact descriptions of vehicular trajectories for configurable accuracy thresholds. We show that our assumptions hold and that our approach achieves significant compression rates even for very tight accuracy bounds through an evaluation based on real GPS measurements of vehicular trajectories.

This technical report paper is an extended version of [9].

The remainder of this paper is structured as follows. We describe the fundamentals of vehicular trajectories and potential approaches to encode them in Section II. We review related work in Section III before presenting and describing an efficient algorithm for the encoding of vehicular movement traces by spline interpolation in Section IV. Section V evaluates our approach on the basis of real-world GPS traces. 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 movements and the characteristics of movement measurements. On this basis, we 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 necessary for V2X applications is the use of *Global Navigation Satellite Systems (GNSS)*, such as GPS, GLONASS or the upcoming Galileo system [10]. GNSS position measurements, in general, do not return the receiver's true position. Disadvantageous satellite constellations and shadowing effects, for example, may cause measurements to be shifted by a temporarily stable bias. At the same time, signal propagation delays and time measurement inaccuracies due to signal multipath propagation,

scattering or reflection usually cause an error which is assumed to be Gaussian distributed in the literature [11].

However, the more information sources are being monitored, the higher is the achievable measurement accuracy. With the *Differential GPS* [12] technology, for instance, users benefit from a network of GPS receivers at well known positions that monitor the temporal development of systemic GPS measurement errors to calculate a correction code. In this way, the error can be reduced from more than ten meters to less than one meter [13]–[15].

Position measurements can only be taken at a specific frequency[1] and thus have a snapshot character. Then, *movement measurements* are merely chronologically sequenced static measurements that can be interpreted as a progression of an object's position over time. The result is a *polygonal chain* with a corresponding timestamp attached to each vertex. The accuracy of such movement measurements thus heavily depends on the snapshot frequency, especially at increasing velocities.

B. Vehicular Trajectory Volume Reduction

In vehicle dynamics [16], the movement of vehicles is modeled on the basis of the principles of kinematics, covering detailed calculations of longitudinal and lateral acceleration forces, among others. Vehicle trajectories in these models are directly influenced by the moving object's speed, acceleration and steering angle. These components can be described as smooth functions, because their changing rates are continuous. This implies that vehicle trajectories themselves can also be expressed as smooth[2] functions $s \in C^2$, mapping a progress variable to a geographical position[3]:

$$s : \mathbb{R} \longrightarrow \mathbb{R}^2, \qquad s(t) = (x, y) \tag{1}$$

Due to its weight and the mass inertia, for example, a vehicle is not able to perform an instant turnaround at a high speed but has to adjust its path and velocity continuously. With respect to the above mentioned position measurements, this continuity translates to redundancy: there is a *pattern* (based on kinematics) underlying these measurements. We propose to take advantage of this property to compress the information about the vehicle's trajectory.

To this end, we need a data structure for the trajectory representation that exploits the identified characteristics and is as generic as possible, so that it is employable for a wide set of applications. Two main conditions have to be met by the data structure: 1) the original measurements need to be restorable, at least with a certain accuracy since they are the only knowledge that has been gathered directly. Anything else is, at best, a reasonably good interpolation between those measures. 2) Due to the afore-mentioned GPS measurement uncertainties, the positions provided by a GPS receiver represent the true locations overlayed with an error. For this reason, it is perfectly acceptable to use a data structure which approximates the measurements but does not have to retain them exactly. However, as different applications require different degrees of accuracy, the maximum error threshold for this must be configurable.

Following these conditions, we will discuss several methods for the interpolation of position measurement sequences and analyze their suitability for our problem.

a) Linear Interpolation: Long, close-to-linear progressions of measurements can indeed be linearly interpolated, using only the end points of a straight line. Also, given the number and an approximate even distribution of the measurements between the end points of the resulting linear interpolation, these can easily be reconstructed. An error parameter could also be implemented without problems. However, since there is no guaranteed even distribution of measurements due to a vehicle's acceleration and deceleration, a higher effort becomes necessary for the interpolation of the omitted measurements. Even more importantly, in curves and bends that occur frequently in real-world vehicle movements, a linear interpolation cannot work efficiently. Finally the original position measurements can not be retrieved directly from such a data structure.

b) Interpolation with Circular Arcs: Over limited distances with nearly-constant steering angle, vehicle movements can be approximated by circular arcs. Such an encoding is very efficient, because merely a center point, a radius and an angular interval need to be defined to describe such a movement. It is also very simple to verify the interpolation error, which is the difference between the calculated radius and the distance from the regarded point to the center point. However, again, there is no possibility to directly retrieve the original position measurement distribution along the arc.

c) Interpolation with Clothoids: Clothoids, sometimes referred to as *Euler* or *Cornu spirals* are curves with a linearly increasing curvature. Due to this property, they are widely used in the design of roads and railways, in particular in the design of transition curves between road segments like straight lines and circular arcs [17]–[19]. Although clothoids can be efficiently defined by the covered curvature range, rotation and scaling parameters, they exhibit the same disadvantages as lines and circular arcs: the original measurements cannot directly be retrieved, because either an equal distribution along the clothoid curve or more additional information about the original position measurement distribution would be necessary.

d) Spline Interpolation: In numerics, spline interpolation is employed to create a smooth concatenation of curves that pass through a sequence of data points, also referred to as *knots*. To this end, polynomials are defined between two succeeding knots under certain conditions, such as C^n continuity for $n+1$*-th degree polynomial splines* at the knots between two adjacent polynomial pieces. For several reasons, cubic splines are well-suited to our demands: first, C^2 continuity ensures that a realistic steering behavior can be modeled (see

[1] Current common devices provide frequencies of one to five Hertz.

[2] 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.

[3] We reduce the geographical position to a point in $\mathbb{R}^2$, as it is done in the commonly-used Universal Transverse Mercator (UTM) coordinate system, for example.

Section II-B). Second, cubic splines provide an especially low oscillation behavior due to their minimal curvature between two successive knots, resulting in a special smoothness of the final spline curve. Third, a spline definition includes the number of points that have been skipped between two remaining knots. Thus, these points can be reconstructed with only minimal effort. Finally, with this systematic reconstruction, an error threshold can easily be implemented.

Due to the drawbacks that have been identified for the other presented data structures and the properties of spline interpolation, we will focus on the interpolation of vehicle trajectories with cubic splines after a brief overview of related work.

III. Related Work

In the literature, there are several areas related to the processing and representation of vehicle trajectories. One of these covers the calculation and encoding of trajectories for autonomous vehicles based on exact map material. This is achieved using Bézier curves [20], Splines [21], [22] or clothoids [23]. Though these publications strengthen the argument that these structures are basically suited for the efficient expression of trajectories, they do not provide a solution to our problems of retrieving original position measurements or data reduction. In contrast, the basic assumptions of the work in this field include an exact knowledge of the road geometry and even partially equidistant points, which is contrary to the nature of the position measurements that we can expect.

In the field of computer-aided design, the authors of [24] describe an approach for sketching a polygonal chain with lines, circular arcs and clothoids. The authors use linear curvature interpolation with a line segment penalty to improve the smoothness of the resulting curve. This penalty, however, allows only little inference to the approximation error of the real position measurements. Finally, the authors do not present a way to retrieve the original measurements from the trajectory description. In [25], cubic spline interpolation of vehicle movements is employed for missile strafe planning. The report focuses on the generation of additional waypoints but does not cover the reduction of data. Thus, the intention of this work is the complete opposite of what we are aiming at but underlines our assumption that cubic splines are well-suited for the approximation of vehicular trajectories.

In [26], the authors discuss the compression of ASCII-coded NMEA-0183 log files on mobile devices. Thereby, they rather focus on the implementation challenges due to the mobile architecture framework than on sophisticated and context-aware compression methods. In [27], the authors present an approach to compress GPS data in the form of RINEX files containing ground tracking measurements which is far from the nature of movements that we focus on.

The works closest to our own discuss the storage and communication data load reduction for the tracking of mobile objects with central *Mobile Objects Databases (MOD)*. To this end, mobile units send position updates to the server hosting an MOD; these transmissions may be triggered by time or once a moving object recognizes that the MOD cannot estimate the mobile unit's position any longer without violating an upper uncertainty bound ϵ. In general, mobile objects can update their positions in an *offline* or *online* fashion. Offline approaches, such as [28]–[31] first collect trajectory data and compress them in a second, *post-factum* step. The most promising approaches perform heuristic or optimal line simplifications [32]–[34] or employ linear dead reckoning (LDR) mechanisms. On the contrary, online trajectory data reduction mechanisms aim at reducing the measured data in real-time as presented in [35]–[39]. Here, LDR mechanisms are widely used, because these are less complex than performing line simplification algorithms after each new taken position measurement. However, none of these works considers using a non-linear movement modeling or polyline simplification.

The only previous work on trajectory data reduction with non-linear functions has been presented in the context of spatio-temporal data base indexing. In [40], the authors motivate that a good representation for spatial vehicular trajectories are so-called minimax polynomials, that approximate an original function in such a way, that the maximum approximation error is minimal for the given approximation parameters. They propose the use of Chebyshev polynomials that have been shown to be a very good approximation of optimal minimax polynomials. The authors of [41] extend this work by adding the temporal dimension to trajectory descriptions. Both contributions, however, use the degree of the Chebyshev polynomial as input parameter and calculate the resulting approximation error after the calculation. They do not present an efficient way of constructing a Chebyshev polynomial representation for vehicular trajectories, for which the maximal approximation error does not exceed a previously set up upper bound ϵ.

IV. Trajectory Interpolation

In this section, we introduce our approach for the compact encoding of vehicular trajectories. We first discuss the basic interpolation of pure geometric information with cubic two-dimensional splines. To this end, we present a greedy algorithm that can find a locally optimal solution for the error-aware reduction of a spline knot sequence. Next to the geometric information, a number of applications from the V2X communications context require additional data to be reported as accomplished with FCD. We therefore present subsequently several ways to add non-geometric measurement data to the trajectory description and present an efficient way to encode spatial and temporal information sets together by cubic spline interpolation.

A. Basic Trajectory Interpolation

In a first step, we provide an algorithm to efficiently encode vehicular trajectories with cubic splines. We will describe such trajectories below as measurement sequences $(m_i)_{i\in\mathbb{N}_0}$, mapping an index i onto a *measurement tuple* m_i:

$$m:\ \mathbb{N}_0 \mapsto \mathbb{R}^2, \qquad i \mapsto m_i = (m_{i,0}, m_{i,1}) \tag{2}$$

Where i is used as an ordering criterion and the elements m_i contain the geodetic positions' degrees of longitude and latitude, respectively. As long as each measurement tuple consists of pure geodetic information, we will refer to it as *position*.

To represent these vehicle movements, we seek a two-dimensional[4] cubic spline that interpolates every taken position measurement. Given an accuracy bound, we compress the trajectory by removing elements from (m_i), so that each of the removed elements can later be reconstructed by the two-dimensional 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 geodetic position measurements (m_i) as defined in Statement (2) and an error threshold $\epsilon \geq 0$. What is the minimal subsequence (m_i') that interpolates (m_i) with ϵ as an upper bound 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 possible subsequence of (m_i) and then select (m_i') as the smallest subsequence with an interpolation error not exceeding ϵ. Given the length of the original measurement sequence $n = \|(m_i)\|$, there are 2^n possible mutually distinct subsequences of (m_i). Since the calculation of the interpolation error has a linear complexity in n, the overall complexity of this approach of finding a globally optimal solution lies within $\mathcal{O}(n \cdot 2^n)$. In the domain of vehicular communication, up to several hundred positions can be included within a single measurement sequence. This makes the described naive approach unfeasible. Instead, we propose to approximate the optimal solution to the stated problem. Following this premise, we now present an algorithm running in $\mathcal{O}(n^3)$ that finds such an approximate solution, referred to as $(\widetilde{m}_i')$.

Our algorithm uses a greedy iterative search to reduce (m_i) down to $(\widetilde{m}_i')$: and is presented in Listing 1: given a measurement sequence (m_i) and an error bound ϵ, the algorithm checks in each iteration for every remaining element in (m_i), what the highest resulting interpolation error will be, if this element is removed from the knot sequence (cf. lines 6-18) [5]. Then, the element with the smallest of these resulting interpolation errors is removed from (m_i) The algorithm terminates once no further element can be removed without violating the error bound ϵ and returns the reduced knot sequence $(\widetilde{m}_i')$.

To reconstruct the original knot sequence, we need to remember the original indices of the elements removed from (m_i). This causes an additional overhead, e.g. in the form of a bit field of length n, where each bit indicates whether or not the respective measurement has been kept. To achieve an overall reduction in size, enough position measurement tuples need to be removed from (m_i) to compensate this overhead. How many tuples have to be removed exactly depends on the binary representation size $2 \cdot s_n$ in byte for each position measurement tuple, the overhead $s_o = \lceil \frac{n}{8} \rceil$ in byte and the reduced knot sequence length $n' = \|(\widetilde{m}_i')\|$. Given these parameters, we can formally describe the requirement that the size of the reduced knot sequence and the overhead need to be smaller than the original sequence's size. This directly translates into *absolute* (Inequation (4)) and *relative* (Inequation (5)) upper bounds for the node reduction:

$$n \cdot 2 \cdot s_n > n' \cdot 2 \cdot s_n + s_o \tag{3}$$

$$\Leftrightarrow \quad n - n' > \frac{s_o}{2 \cdot s_n} \tag{4}$$

$$\Leftrightarrow \quad \frac{n'}{n} < 1 - \frac{s_o}{n \cdot 2 \cdot s_n} \tag{5}$$

We will refer to the *reduction fraction* as $\sigma = \frac{n'}{n}$.

Algorithm 1 Greedy Spline Reduction.

Require: sequence of position measurements $(m_i)_{i \in \mathbb{N}_0}$
Require: error tolerance $\epsilon \geq 0$
1: $(\widetilde{m}_i') \leftarrow (m_i)$
2: $I_{rem} \leftarrow \emptyset$
3: **repeat**
4: $\quad \epsilon_{min} \leftarrow \infty$
5: $\quad j_{min} \leftarrow -1$
6: $\quad$ **for all** j such that $m_j \in (\widetilde{m}_i')$ **do**
7: $\quad\quad$ **if** $i \in [1; \|(m_i)\| - 2]$ **then**
8: $\quad\quad\quad I_{rem} \leftarrow I_{rem} \cup \{j\}$
9: $\quad\quad\quad$ remove m_j from $(\widetilde{m}_i')$
10: $\quad\quad\quad \epsilon_j \leftarrow$ compare(interpolate$((\widetilde{m}_i'), I_{rem}), (m_i))$
11: $\quad\quad\quad$ **if** $\epsilon_j < \epsilon_{min}$ **then**
12: $\quad\quad\quad\quad j_{min} \leftarrow j$
13: $\quad\quad\quad\quad \epsilon_{min} \leftarrow \epsilon_j$
14: $\quad\quad\quad$ **end if**
15: $\quad\quad\quad I_{rem} \leftarrow I_{rem} \setminus \{j\}$
16: $\quad\quad\quad$ re-insert m_j into $(\widetilde{m}_i')$
17: $\quad\quad$ **end if**
18: $\quad$ **end for**
19: $\quad$ **if** $\epsilon_{j_{min}} \leq \epsilon$ **then**
20: $\quad\quad I_{rem} \leftarrow I_{rem} \cup \{j_{min}\}$
21: $\quad\quad$ remove $m_{j_{min}}$ from $(\widetilde{m}_i')$
22: $\quad$ **end if**
23: **until** no further knot could be removed
24: **return** reduced knot sequence $(\widetilde{m}_i')$

B. Adding Non-Geometric Data

The knowledge about vehicular movements is important for a number of applications, but often just form the basis that has to be extended to provide a more sophisticated service. A very popular candidate for additional data is the sequence of points in time, at which corresponding measurements have been taken. This allows, for instance, to derive a vehicle's speed on a particular road segment, to conclude on the road segment's

[4]For the interpolation, we use one spline per input dimension. Since we need to interpolate elements in $\mathbb{R}^2$, we refer to two-dimensional splines.

[5]In fact, the first and last element are not considered, (cf. line 7) because otherwise an interpolation would not be possible any more.

position within a fundamental diagram[6] or to monitor the feasible average speed for it as in [2], [3]. For other projects, such as [4], additional measurements of temperature, humidity and friction coefficients need to be combined and transmitted.

Basically, every additional information that is added to the trajectory description increases the dimensionality of the data collection by one, so that we can generalize our previous definition of a measurement sequence as:

$$m:\ \mathbb{N}_0 \mapsto \mathbb{R}^d, \qquad i \mapsto m_i = (m_{i,0}, \ldots, m_{i,d-1}) \tag{6}$$

To achieve a high overall encoding efficiency, every additional measurement sequence should be reduced in length, as far as possible, as we have proposed for the geometric data. Furthermore, as already mentioned, it appears reasonable to consider measurement tuples as the smallest entities for the reduction process, because a separate interpolation of measurement sequences increases the overhead due to additional index bit fields for each dimension or context.

The above-stated upper performance bounds for such measurement collections can as well be generalized: assuming a measurement tuple size of $d \cdot s_n$, and since the overhead s_o is independent of d, the inequations for the absolute (7) and relative (8) upper bound

$$n - n' > \frac{s_o}{d \cdot s_n} \tag{7}$$

$$\frac{n'}{n} < 1 - \frac{s_o}{n \cdot d \cdot s_n} \tag{8}$$

have to hold for a successful reduction. For a generic measurement tuple dimensionality d, we can thus define the general relative upper bound $\beta = 1 - \frac{s_o}{n \cdot d \cdot s_n}$ that must not be exceeded by the reduction fraction σ. This implies that the necessary effective reduction ratio of the measurement set decreased for an increasing dimensionality d of the measurement tuples.

Note that though we consider spline interpolation exceedingly useful for trajectory interpolation, this has not necessarily to be true for every possible measurement sequence. Instead, the reduction algorithm or data structure should fit the particular context. For example, friction parameters probably do not need to be interpolated by an extraordinary smooth function but a much simpler construct instead.

In the code listing, we did not emphasize the dimensionality of the input data, because it is irrelevant for the algorithm's asymptotic runtime complexity. Line 10 holds the interpolation and error determination of the measurement subset, which is the key operation in the algorithm. The interpolation of a single spline runs on the order of $\mathcal{O}(n)$. If the error determination for a reduction step for an additional parameter 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 context[7].

[6] A fundamental diagram correlates the traffic flux (vehicles per hour) and the traffic density (vehicles per kilometer) for traffic flow estimations.

[7] For example, we have summarized the two geometrical dimensions to a single context with ϵ being the respective error threshold.

Topology	Length	# Measurements	Frequency
City	8,896.56 m	2086	2.0 Hz
Highway #1	23,372.38 m	2021	2.0 Hz
Highway #2	14,883.04 m	1306	2.0 Hz

Table I
TOPOLOGY OVERVIEW.

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 onto a geographical location. Basically, the points in time[8] at which the respective measurements have been taken can be employed as values for such a progress variable. To fulfill the first condition set up in Section II-B, however, all measurements must have been taken at a strict regular basis, i. e. the temporal distance between each to successive measurements has to be constant. With the knowledge of this measurement frequency and the first measurement's timestamp, we can easily reconstruct all measurement timestamps in the sequence. If this requirement cannot be met, e. g. if location measurements have been lost before being logged or have not been logged at 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 measurements at strictly regular intervals and contemporary more sophisticated positioning systems, which are e. g. based on Kalman filters [42] and inertial navigation systems [43], can easily accomplish this task as well.

Due to the inclusion of the additional temporal information within the index bit field, this bit field does no longer count as an overhead but as payload, containing measurement information. In other words, the additional encoding of temporal data comes at the minimal cost of a time reference and a measurement frequency value, thus setting the relative upper bound $\beta = 1$. Thus, an effective data reduction begins with the first removed element from the measurement sequence.

V. EVALUATION

In the previous section, we have presented our ideas not only to encode and reduce geographical trajectory data but also to append additional information sets such as time to this encoding. In the following, we will evaluate our proposals on the basis of extensive GPS real-world measurements.

A. Data Acquisition

For our evaluation, we have collected vehicular GPS measurements in one city and two highway topologies. The driven

[8] Strictly speaking, the respective time offsets to a starting value need to be employed for this.

Topology	$n = 50$ $o = 13$	$n = 100$ $o = 25$	$n = 150$ $o = 38$	$n = 200$ $o = 50$	$n = 250$ $o = 63$
City	157	80	51	38	30
Highway #1	152	77	50	37	29
Highway #2	97	49	31	23	17
Buffer Size	416 B	816 B	1.19 KB	1.58 KB	1.97 KB

Table II
NUMBER OF INPUT FILES AND NECESSARY BUFFER SIZE (IN BYTE AND KILOBYTE) FOR VARYING WINDOW SIZES.

tracks had lengths of several kilometers and consisted of up to several thousand position measurements. A detailed overview is given in Table I.

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

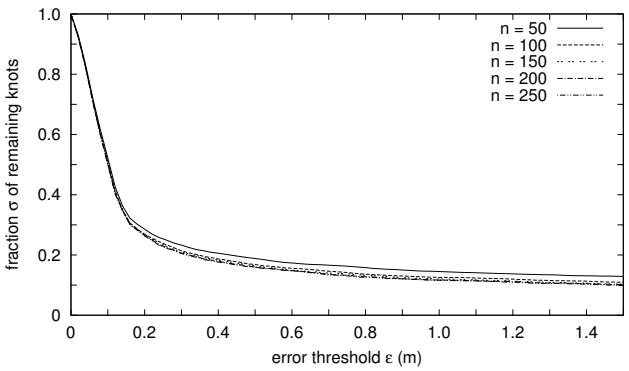
For real applications, static size buffers are an efficient technique to log and subsequently transmit position measurement traces. In our case, once the buffer is full, the data is passed to the reduction module based on Algorithm 1. The output is then transmitted to a collector and the logging starts over anew.

We have simulated this behavior using a sliding window approach for window sizes of 50 up to 250 measurements in steps of 50: starting with a window position at the respective measurement trace's beginning, we took all measurements within the window and copied them to measurement subsequences. In this way, various snapshots from during our test runs could be made that conform to the above-mentioned static size buffer's content. In Section IV, we have introduced such subsequences as $(m_i)_{i \in \mathbb{N}_0}$ and the window size thus directly translates into the original size of a measurement subsequence $n = \|(m_i)\|$. The window was then shifted by an offset $o = \frac{n}{4}$. Moving the window by only a fourth of its size allowed us to increase the number of regarded subsequences and to spread both advantageous and disadvantageous measurements onto multiple subsequences to make our evaluations as meaningful as possible.

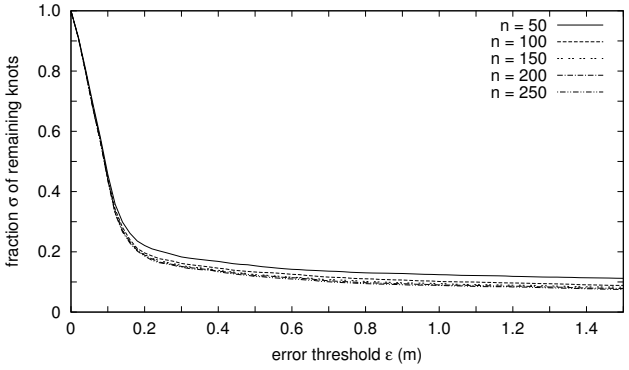
Table II gives an overview on the number of input files and necessary buffer sizes for varying window sizes. For the buffer size approximation, we assumed two 32-bit floating point position measurements per element in addition to a 32-bit integer reference timestamp for the whole measurement sequence and another 32-bit floating point measurement frequency value. The resulting low buffer sizes show that this approach is indeed feasible for the proposed window sizes.

B. Knot Sequence Reduction

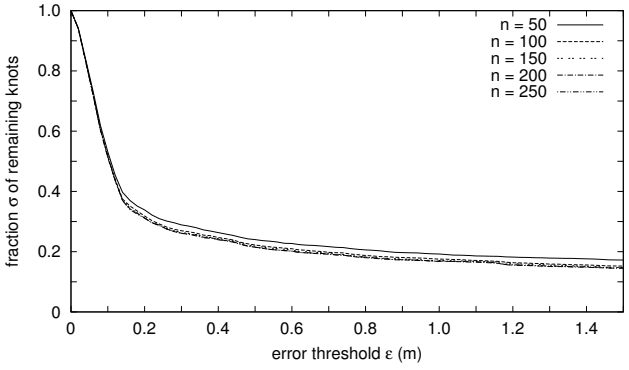
The most meaningful performance criterion for the reduction by means of spline interpolation is the fraction $\sigma = \frac{n'}{n}$ of remaining knots after the reduction has been applied. This is depicted in Figure 1 for all topologies. The figure shows the average fraction σ of remaining knots over an increasing error threshold ϵ for values in steps of 2 cm and up to 2 m in total.



(a) City topology.



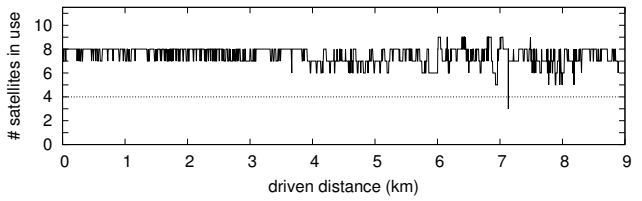
(b) Highway topology 1.



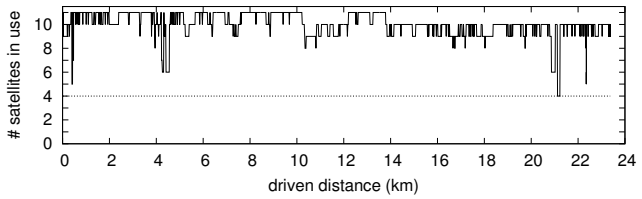
(c) Highway topology 2.

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

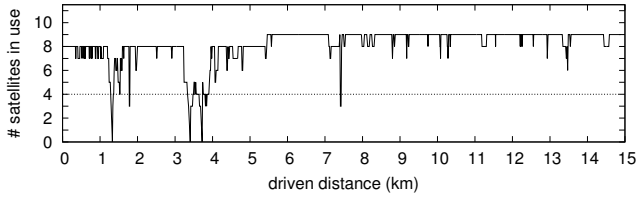
The first remarkable observation that can be made is that there is a steep declivity of σ for error thresholds up to approximately 20 cm. This results in an average knot sequence reduction of 70 % to 83 % at an interpolation error tolerance of only 20 cm and even a reduction of 83 % to 93 % for an error tolerance of 1.5 m. The steep declivity indicates a slight noise or 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 spline interpolation approach works very well. This observation clearly confirms our assumption that vehicular traces can be represented very



(a) City topology.



(b) Highway topology 1.



(c) Highway topology 2.

Figure 2. Number of satellites in use over driven distance.

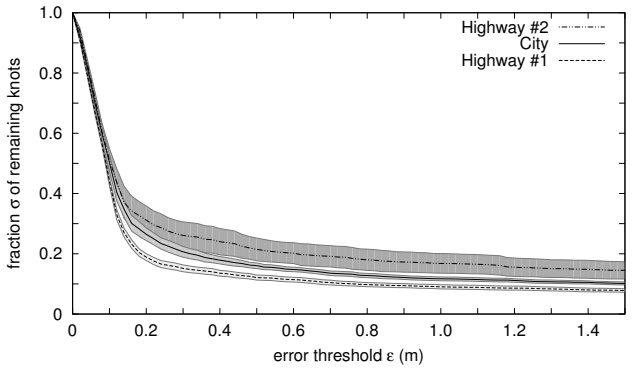


Figure 3. Knot reduction analysis for $n = 200$.

well by cubic splines.

Second, we see that the window size n has only a minor effect on the result of the reduction algorithm. Though the algorithm performs worst for each topology at $n = 50$, the results differ only marginally from the ones achieved with larger window sizes. It appears that a larger context bears a higher potential for reduction. Also, there is no distinct performance difference for window sizes greater than 150 elements visible in the results. For this reason, we will focus on a window size of $n = 200$ elements in our further analyses.

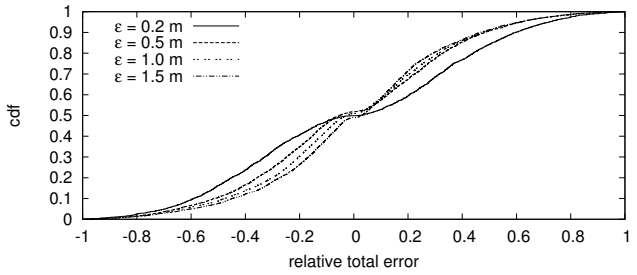
Finally, we notice that though the knot sequence reduction gives very good results for all topologies overall, the results for the second highway topology are not as good as for the other two. The reason for this is the setting of the second highway topology: roofings over parts of the highway heavily constrained the field of vision to the sky and thus impaired the GPS signal availability. This can be seen in Figure 2 that depicts the number of satellites in use over the driven distance for the three topologies. While there is a very good and stable satellite reception for the city and the first highway topology, the number of satellites for the second highway topology descends below four, which is the minimum number of satellites necessary for a reasonably exact GPS position determination. This threshold is visualized by dashed horizontal lines in the figures. This unstable reception behavior results in a higher dilution of precision (DOP), causing the measured positions to feature a high noise and only a poor smoothness and continuity. Due to this, a good reduction is only achievable at a higher error threshold for many subsequences in this topology. This can also be seen in Figure 3, showing the reduction results from Figure 1 for $n = 200$ with 95 % confidence interval corridors. In contrast to the other two topologies, the second highway topology has a very wide confidence interval which confirms a high variation of the achieved results.

C. Interpolation Error

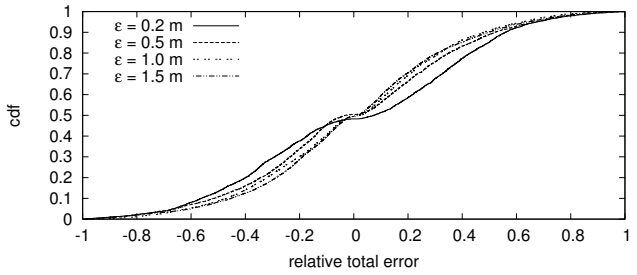
We have seen that even a relatively small interpolation error threshold allows for high knot amount reductions for measurement sequences. In the next step, we take a closer look on the nature and distribution of this interpolation error.

Figure 4 shows the cumulative distribution of the relative interpolation error of the three topologies for a selection of interpolation error thresholds. The relative interpolation error describes the ratio of the occurred interpolation error and the interpolation error threshold. A negative interpolation error occurs, if the original point lies on the left hand side of the interpolation, in the direction of progress. The results show for all topologies a symmetric interpolation error distribution, i. e. positive and negative interpolation errors are distributed alike. This means that there is no topology-specific error behavior. Second, we see that with an increasing error tolerance, the distribution's compactness grows in the area of 5-40 %. This is because the interpolation errors are not spread evenly, but that only few measurement points are interpolated at a close-to-maximum and close-to-minimum interpolation error. The latter circumstance results from a base noise and the above-mentioned discretization of position measurements.

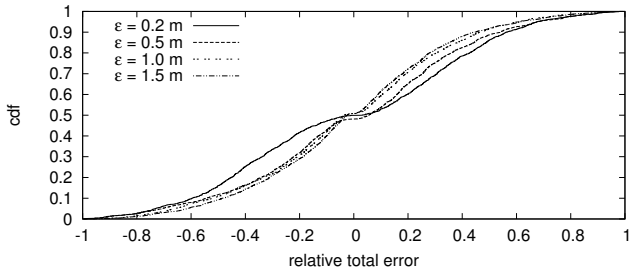
Though the total interpolation error distribution is a good indicator for the quality of the trajectory interpolation, it is not enough to make clear statements. A more thorough understanding can be provided by differentiating the components of the interpolation error, instead. Therefore, we will regard its longitudinal and lateral parts for our considerations as well: Figure 5 shows a polygonal chain as a dotted gray line and its spline approximation as a solid black curve. Additionally, the mentioned error components for the interpolated knot $\widetilde{m}'_j$ of the original position measurement m_j are depicted. While the longitudinal interpolation error describes the divergence



(a) City topology.



(b) Highway topology 1.



(c) Highway topology 2.

Figure 4. Total error distribution.

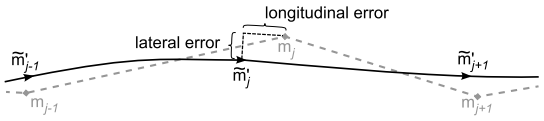
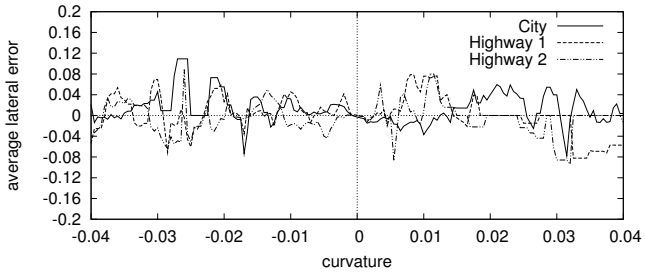


Figure 5. Longitudinal and lateral error components.

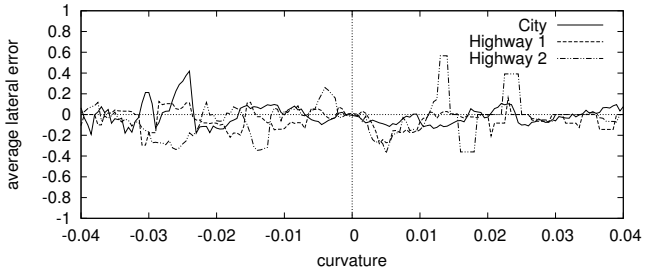
along the movement direction, the lateral error refers to the part perpendicular to it.

As mentioned in Section II-B, cubic splines provide an especially low oscillation behavior due to a minimal curvature between two successive knots. Therefore, it is surely possible that systemic errors in relation to the curvature of the spline interpolation occur, so that curves and bends are interpolated tighter than they originally are. In this case, a clear correlation between the curvature of the spline and the lateral error would exist. This is why especially the lateral error is a good metric for the accuracy for the spline interpolation of vehicle movements.

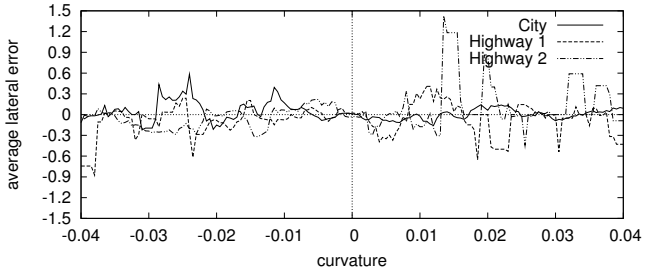
Figure 6 shows the lateral error in relation to the spline's curvature for all of our topologies. Since the curvature $\kappa = \frac{1}{R}$ is the inverse of the curve radius R, a high absolute curvature value translates into a tight curve or bend. The figure covers



(a) $\epsilon = 0.2\,m$.



(b) $\epsilon = 1.0\,m$.



(c) $\epsilon = 1.5\,m$.

Figure 6. Lateral error over curvature (summary).

curvatures of up to $|\kappa| = 0.04$, resulting in curve radii of up to 25 m, which corresponds to tight curves for the considered velocities. For all regarded interpolation error thresholds and topologies, 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 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, this task has been achieved by merely concatenating position measurements to *polygonal chains*. These chains contain a high degree of redundancy and are thus far from being optimal, especially when trajectory data should be transmitted over a channel with limited bandwidth. In this paper, we have proposed an encoding scheme based on cubic spline interpolation and have presented a greedy algorithm that filters out redundant position measurements in $\mathcal{O}(n^3)$ for a given error threshold ϵ. The omitted measurements can afterwards

be retrieved by means of cubic spline interpolation with ϵ denoting the guaranteed maximum interpolation error. We have applied our algorithm to a large number of real-world GPS measurements. Our results show that the amount of data that is necessary to encode a vehicular trajectory with an accuracy of 20 cm can be reduced by more than 80 %. The resulting interpolation error is not systemic, but it even appears that spline interpolation can reduce the lateral noise from GPS measurements instead, which will be subject to future work.

ACKNOWLEDGMENTS

We wish to thank Gian Perrone for his reliable work on the interpolation framework.

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 *ITS World Congress '02: Proceedings of the 9th World Congress and Excibition on Intelligent Transportation Systems and Services (ITS)*, Oct. 2002.

[3] J. Rybicki, B. Scheuermann, M. Koegel, and M. Mauve, "PeerTIS - A Peer-to-Peer Traffic Information System," in *VANET '09: Proceedings of the 6th ACM International Workshop on VehiculAr Inter-NETworking*, Sep. 2009, pp. 23–32.

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

[5] 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 International Federation of Automotive Engineering Societies*, Sep. 2008.

[6] W. Huber, M. Lädke, and R. Ogger, "Extended floating-car data for the acquisition of traffic information," in *ITS World Congress '99: Proceedings of the 6th World Congress and Excibition on Intelligent Transportation Systems and Services (ITS)*, Nov. 1999.

[7] M. E. Fouladvand and A. H. 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.

[8] R. Brüntrup, S. Edelkamp, S. Jabbar, , and B. Scholz, "Incremental map generation with gps traces," *itsc05*, Nov. 1999.

[9] M. Koegel, W. Kiess, M. Kerper, and M. Mauve, "Compact vehicular trajectory encoding," in *Submitted to VTC '11-Spring: Proceedings of the 73th IEEE Vehicular Technology Conference*, 2011.

[10] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more*, 1st ed. Springer Wien, 2007.

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

[12] J. Hurn, *Differential GPS Explained: An expose of the surprisingly simple principles behind today's most advanced positioning technology*. Sunnyvale, CA: Trimble Navigation Limited, 1993.

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

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

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

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

[17] K. G. Baass, "The use of clothoid templates in highway design," *Transportation Forum*, vol. 1, pp. 47–52, 1984.

[18] D. S. Meek and D. J. Walton, "Clothoid spline transition spirals," *Mathematics of Computation*, vol. 59, no. 199, pp. 117–133, July 1992.

[19] Forschungsgesellschaft für Straßen- und Verkehrswesen, *RAS-L – Richtlinien für die Anlage von Straßen – Teil: Linienführung*, 2nd ed. Der Verlag der Forschungsgesellschaft für Straßen- und Verkehrswesen, 1999.

[20] 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.

[21] 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.

[22] S. Kehl, "Querregelung eine Versuchsfahrzeugs entlang vorgegebener Bahnen," Ph.D. dissertation, University of Stuttgart, Institute for System Dynamics, Nov. 2007, in German.

[23] 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.

[24] J. McCrae and K. Singh, "Sketching piecewise clothoid curves," *Computers & Graphics*, vol. 33, no. 4, pp. 452–461, 2009.

[25] A. C. Jolly, "Trajectory generation by piecewise spline interpolation," US Army Research, Development and Engineering Laboratory: Missile Command, Redstone Arsenal, Alabama 35809, Tech. Rep. RG-76-56, 1976.

[26] R. Lever, A. Hinze, and G. Buchanan, "Compressing gps data on mobile devices," in *OTM Workshops (2)*, ser. Lecture Notes in Computer Science, vol. 4278. Springer, Nov. 2006, pp. 1944–1947.

[27] A. Villafranca, I. Mora, P. Ruiz, J. Portell, and E. García-Berro, "Optimizing gps data transmission using entropy coding compression," in *SDCCP '10: Proceedings of the 6th SPIE Conference on Satellite Data Compression, Communications, and Processing*, Aug. 2010.

[28] A. Leonhardi and K. Rothermel, "A comparison of protocols for updating location information," *Cluster Computing: The Journal of Networks, Software Tools and Applications*, vol. 4, no. 4, pp. 355–367, October 2001.

[29] N. Meratnia and R. A. de By, "Spatiotemporal compression techniques for moving point objects," in *EDBT '04: Proceedings of the 9th International Conference on Extending Database Technology*, Heraklion, Crete, March 2004.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] W. Chan and F. Chin, "Approximation of polygonal curves with minimum number of line segments," *Proceedings of the 3rd International Symposium on Algorithms and Computation, in: Lecture Notes in Computer Science*, vol. 650, pp. 378–387, 1992.

[35] G. Trajcevski, H. Cao, P. Scheuermann, O. Wolfson, and D. Vaccaro, "Online data reduction and the quality of history in moving objects databases," in *MobiDE '06: Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Chicago, IL, USA, June 2006.

[36] R. Lange, F. Dürr, and K. Rothermel, "Online trajectory data reduction using connection-preserving dead reckoning," in *MobiQuitous '08: Proceedings of the 5th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Dublin, Ireland, July 2008.

[37] N. Hönle, M. Gromann, D. Nicklas, and B. Mitschang, "Preprocessing position data of mobile objects," in *MDM '08: Proceedings of the 9th IEEE International Conference on Mobile Data Management*, Apr. 2008.

[38] 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.

[39] 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.

[40] 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, pp. 599–610.

[41] 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.

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

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