

# A Comparison of Vehicular Trajectory Encoding Techniques

Markus Koegel\*   Daniel Baselt\*   Martin Mauve\*   Björn Scheuermann<sup>‡</sup>

\*Department of Computer Science, University of Düsseldorf, Germany  
{ koegel, baselt, mauve }@cs.uni-duesseldorf.de

<sup>‡</sup>Department of Computer Science, University of Würzburg, Germany  
scheuermann@informatik.uni-wuerzburg.de

**Abstract**—The transmission of vehicular trajectory information is one basic building block of car-to-car communication. Frequently, this information is transmitted as raw data, i.e., as a sequence of location measurements. In this paper we argue that, due to the laws of physics and the requirement to follow a road, vehicular mobility has very specific characteristics. Hence, vehicular trajectory information can be compressed very efficiently using domain-specific lossy compression schemes. We discuss and compare three promising approaches that can be used to this end: linear approximation, cubic splines, and clothoids.

## I. INTRODUCTION

Many applications in inter-vehicle communications require the exchange of trajectory information [1]–[3]. Typically this information is transmitted as a sequence of position measurements called *polygonal chains* or *polylines* [4]–[6]. Albeit the information given in this form is highly valuable, the transmission of such polygonal chains may occupy a large amount of network resources. Since vehicular movements obey kinematic principles and are therefore predictable to a certain extent, such descriptions contain redundancy and thus bear a high potential for data reduction.

In this work, we investigate *lossy* compression schemes for trajectory information: they allow for an error in the data once it is decompressed. Upper bounds for this error usually serve as input parameter for the compression schemes and influence their performance significantly. The main focus in research, so far, has been on the linear approximation of movement traces by the means of line simplification or linear dead reckoning (LDR) algorithms [7]–[10]. However, vehicular movements are approximately linear only over very short distances or on linear road segments; more generally, vehicles move along curves and are thus likely to be represented more efficiently by non-linear approximations. We have proposed this idea recently in [11] and have shown that two-dimensional cubic splines yield good approximations for vehicular movements.

In this paper, we extend our examination of geometrical methods for the approximation of vehicular movement traces and make three central contributions: we first introduce our spline-based approach, extend it, and complement our description by an overhead analysis. Second, we present a detailed comparison with two common line simplification algorithms often used in trajectory reduction; we point out their strengths and shortcomings. Finally, we discuss algorithms for clothoid-based polyline sketching, derive compression schemes from these, and analyze their performance and weaknesses. By

surveying and evaluating these approaches, we recapitulate the previous work in the area of geometric trajectory encoding methods and provide insights into their capabilities and limitations. In doing so, we present for each of the three approaches a basic version to clarify the fundamental idea, before we then discuss the most promising extension.

We introduce linear, cubic spline-based, and clothoid spline-based geometrical approaches for efficient trajectory encoding in Sections III–V, respectively. In particular, this includes the discussion of an extension to our own spline-based method in Section IV. We explain our methodology and present a compression evaluation based on real GPS movement measurements in Section VI, before we close this paper with a look on related work and a conclusion.

## II. PROBLEM STATEMENT AND NOTATION

In this paper, we focus on the encoding of vehicular trajectories by means of geometrical methods. We consider both linear and nonlinear approaches, and therefore need a clear problem statement that is applicable in both cases.

In the following, we will denote a position measurement trace as sequence  $\langle m_j \rangle_{j \in J}$ , mapping the elements of an (ordered) index set  $J$  to  $d$ -dimensional *measurement tuples*:  $m : J \rightarrow \mathbb{R}^d$ ,  $j \mapsto m_j = (a_0, \dots, a_{d-1})$ . The discussed techniques are more generally applicable, but for the sake of simplicity we focus on the two-dimensional case of geographic coordinate pairs here:  $m : J \rightarrow \mathbb{R}^2$ ,  $j \mapsto m_j = (x_j, y_j)$ . Furthermore, we assume that an accuracy bound  $\epsilon \geq 0$  is given.

Then, generally speaking, we aim to find a compact representation which allows us to reconstruct the elements from  $\langle m_j \rangle$  with a maximum error of  $\epsilon$ . In the following the representation we will look at is another point sequence  $\langle m'_j \rangle$ ;  $\langle m'_j \rangle$  need not necessarily be a subsequence of  $\langle m_j \rangle$ . To find  $\langle m'_j \rangle$ , we use different, purpose-tailored compression and decompression algorithms; let us denote such an ensemble by  $E(c, r, \epsilon)$ . Here,  $c$  denotes the compression mapping:  $c : (J \rightarrow \mathbb{R}^2) \rightarrow (J' \rightarrow \mathbb{R}^2)$ ,  $c(\langle m_j \rangle) = \langle m'_j \rangle$ ,  $|J'| \leq |J|$ . The corresponding decompression mapping is denoted by  $d^1$ :  $r : (J' \rightarrow \mathbb{R}^2) \rightarrow (\hat{J} \rightarrow \mathbb{R}^2)$ ,  $r(\langle m'_j \rangle) = \langle \hat{m}_j \rangle$ ,  $|J| = |\hat{J}|$ , so that the sequences  $\langle m_j \rangle$  and  $\langle \hat{m}_j \rangle$  do not differ by more than  $\epsilon$  element by element with a Euclidean distance metric.

In the following, we will present different versions for the compression and decompression mappings  $c$  and  $r$ .

<sup>1</sup>We slightly changed our notation from [11] and use  $\langle \hat{m}_j \rangle$  instead of  $\langle \tilde{m}_j \rangle$ .

### III. LINEAR MOVEMENT APPROXIMATION

In the next sections, we review a number of geometric trajectory encoding methods, which we compare in Section VI. We begin with common linear approaches and then increase the complexity by turning to non-linear methods employing cubic splines and finally clothoids.

#### A. The Douglas-Peucker Algorithm

The Douglas-Peucker algorithm [12] is a heuristic line simplification algorithm that implements a divide-and-conquer strategy: in a first step, the first and last node of a polygonal chain are taken as end points of a line. For all points in between, the shortest distances to this line are determined. If any distance exceeds  $\epsilon$ , the polygonal chain is separated with the point with the largest distance being both end and start point for the first and second new line segments, respectively. Then, the algorithm starts anew for each of these subchains. It terminates when no point-line distance exceeds  $\epsilon$ . Figures 1(a)–1(e) show a simple example.

As mentioned before, the Douglas-Peucker algorithm employs a heuristic, i. e., the result  $c(\langle m_j \rangle) = \langle m'_j \rangle$  is not neces-

sarily the global optimum. Its advantage, on the other hand, is its runtime complexity, which is  $\mathcal{O}(nk)$  with  $n = |J|, k = |J'|$  in a naive implementation, but can, for the two-dimensional case, be improved to  $\mathcal{O}(n \log n)$  [13].

#### B. Optimal Line Simplification

An algorithm that finds the minimal subsequence  $\langle m'_j \rangle$  in  $\mathcal{O}(n^2 \log n)$  time has been proposed in [14]: first, the algorithm creates a directed graph  $G(V, E)$ , where  $V$  includes all nodes from  $\langle m_j \rangle$ . An edge  $e = (i, j) \in E$ , iff  $i < j$  and the distances of  $m_{i+1}, \dots, m_{j-1}$  to the line segment between  $m_i$  and  $m_j$  are smaller than  $\epsilon$ . Formally:  $\forall k : i < k < j : d_{l_{i,j}}(m_k) \leq \epsilon$  with the distance metric  $d_{l_{i,j}}(\cdot)$  of a point to the line segment from  $i$  to  $j$ . Once this graph has been constructed, the minimal subsequence is determined by finding the shortest path from  $m_0$  to  $m_{n-1}$ . Obviously, the line simplifications found by this approach are upper bounds for all approaches based on linear movement approximation such as [7]–[10].

Figure 1(f) shows the minimal simplification for the example polygonal chain from Figure 1(a). It clearly differs from the result obtained with the Douglas-Peucker algorithm, and is, in this example, slightly more compact.

### IV. MOVEMENT APPROXIMATION WITH CUBIC SPLINES

Next, we describe the encoding of vehicular trajectories with cubic splines. We introduce our approach from [11] and discuss its overhead, before we turn towards some modifications.

#### A. Basic Approach

When looking at the specific characteristics of vehicular trajectories, in particular their smoothness and continuity, two-dimensional cubic splines appear to be an excellent fit: they ensure  $C^2$  continuity<sup>2</sup>, so their curvature  $\kappa = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}$  is continuous as well. Here,  $x(t)$  and  $y(t)$  denote the parameterized functions for the two dimensions at index  $t$ . This implies a special smoothness—in fact, cubic splines feature a minimal curvature and thus an especially low oscillation—that allows for realistic modeling of acceleration and steering. Moreover, since splines are defined by the index offsets of the given knots, which may be fragmentary, it is straight-forward to reconstruct the omitted knots.

So, the sought-after mapping  $r$  for the approximation of the original measurements would be a simple cubic spline interpolation. The more interesting question is: how can the compression mapping  $c$  that provides the minimal measurement subsequence be implemented?

Finding the minimal knot sequence  $\langle m'_j \rangle$  for which the interpolation  $\langle \hat{m}_j \rangle$  does not exceed the error bound  $\epsilon$  is non-trivial. Indeed, all possible  $2^n$  subsequences have to be considered for an optimal solution, because—different than for linear approaches—removing one knot has a direct impact on all knots, not only on those in the direct surrounding.

We propose a heuristic approach based on a greedy iterative search that finds a local optimum in  $\mathcal{O}(n^3)$  steps. First, the

<sup>2</sup>A smooth function is  $C^d$  continuous, iff its first  $d$  derivatives exist and are continuous.

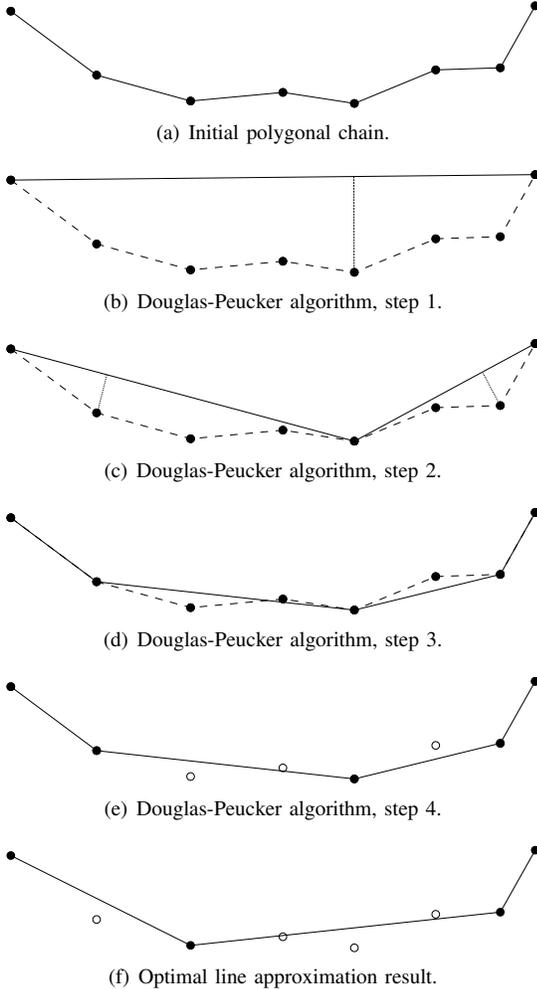


Figure 1. Examples for line approximation.

algorithm checks for every but the first and last knot what the highest resulting interpolation error would be if this particular knot were removed from the sequence. The knot with the smallest such error is then removed from the sequence, before the algorithm starts over. If all of the resulting interpolation errors exceed  $\epsilon$ , no further knot can be removed and a local optimum is reached.

As already mentioned, the indices of the omitted knots need to be remembered for the interpolation. We propose the usage of a simple bit field of length  $n = |J|$ , where each bit indicates whether or not the respective measurement has been kept. To achieve an overall reduction in size, enough tuples need to be removed from  $\langle m_j \rangle$  to compensate for this overhead: with the number of remaining tuples  $k = |J'|$ , the size of the binary representation of a tuple  $2 \cdot s$ , and the size of the bit field  $n$  (both sizes in bit), we can describe an absolute size threshold that needs to be under-run for the compression to be effective: the sum of the overhead and the size of the reduced tuple sequence needs to be smaller than the size of the original tuple sequence. More formally this is:

$$n + k \cdot 2 \cdot s < n \cdot 2 \cdot s \Leftrightarrow \frac{n}{2 \cdot s} < n - k \quad (1)$$

### B. Unseaming dimension contexts

We have previously shown in [11] that the above-described algorithm provides very good compression ratios for real GPS position traces. In its basic version, however, it has a drawback that may inhibit an even better compression performance: in the first step, we compute the interpolation errors for possibly-removed knots. For this, the absolute euclidean distance between an original knot and its interpolated counterpart is calculated, i. e., both dimensions are compressed simultaneously as a *context*. In doing so, the basic algorithm does not take into account that one of the dimensions may be more complex—and thus harder to compress—than the other; in fact, the compression result of the two-dimensional context can only be as good as the one of the more complex dimension.

To overcome this issue, we propose a modified version of our algorithm that compresses different dimensions separately. The first two steps of each loop cycle (determine the highest resulting interpolation errors for each knot and remove the knot that minimizes these errors) are applied to the dimensions sequentially: instead of removing a knot at the same index in both dimensions, one index for each dimension is determined independently. The dimensions are thus compressed separately, still regarding the threshold  $\epsilon$  for the total interpolation error. E. g., the index for the longitudinal (or  $x$ ) dimension is determined first, and the one for the lateral (or  $y$ ) dimension second. The changes performed on the first dimension influence the compression of the second one and therefore a changed order of dimensions is likely to affect the overall compression, albeit marginally. Note that it is now possible that, at some point, one dimension cannot be compressed any more, whereas the other one still can. The algorithm may then skip one dimension in the remaining cycles, until the compression of the second dimension has also reached a local optimum.

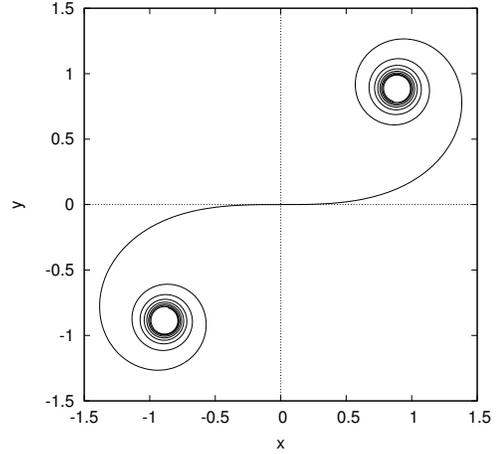


Figure 2. Unit clothoid.

The runtime complexity of our algorithm is not affected by this modification, since we merely multiply the quadratic effort of determining the interpolation error by the (fixed) number of dimensions:  $\mathcal{O}(d \cdot n^3) = \mathcal{O}(n^3)$ .

But even though this *context-loosened* compression may provide better results, it also causes a higher overhead in one respect: since the dimensions are compressed in a largely decoupled fashion, we now need to provide one bit field *per dimension* to remember the removed knot indices. We verify this by describing the absolute size threshold for the context-loosened compression mapping analogue to (1) as follows:

$$2 \cdot n + k \cdot 2 \cdot s < n \cdot 2 \cdot s \Leftrightarrow \frac{n}{s} < n - k \quad (2)$$

where  $k$  is now the mean dimension knot sequence length. The comparison to the basic algorithm according to (2) shows that the necessary number of knots that need to be removed from  $\langle m_j \rangle$  has indeed doubled.

### V. CLOTHOIDAL MOVEMENT INTERPOLATION

So far we have discussed approximations using lines and cubic splines. However, vehicular movements not *only* follow the principles of kinematics, but also the course of the roadway. Once it is known what kind of basic geometric elements a road is typically composed of, one could assume that any vehicular movements along such roadways can likewise be expressed or at least approximated by variations of these elements.

In the design and construction of roadways, linear segments, circular arcs, and clothoid segments—often referred to as *curve primitives*—are widely used to achieve optimal trafficability [15], [16]. Clothoids are special curves for which the arc length ( $L$ ) is inversely proportional to the curve radius ( $R$ ), and thus proportional to the curvature ( $\kappa$ ):  $L = A^2/R = \kappa/A^2$ , where  $A$  is a constant scaling factor. Figure 2 shows the double-ended *unit clothoid*, i. e.,  $A = 1$ .

These curve primitives reflect the nature of fundamental vehicular movements: first, for yaw rates equal to zero, i. e., the steering wheel is constantly at a neutral position, a vehicle obviously moves ahead on a straight line. With a uniformly

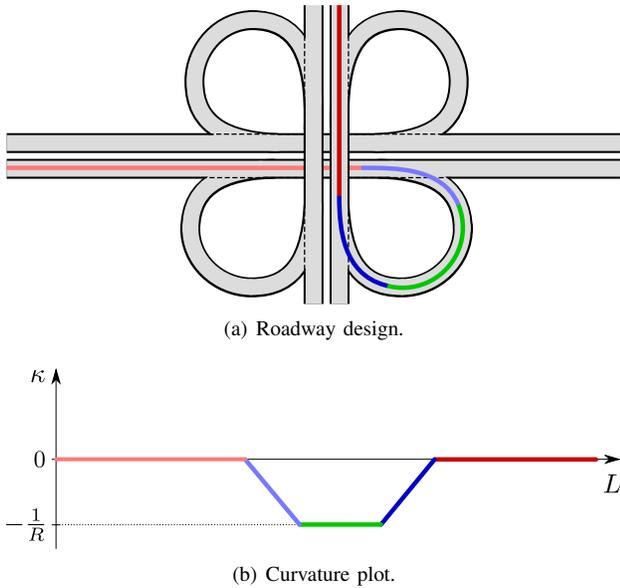


Figure 3. Roadway design using lines, circular arcs and clothoids.

varying yaw rate, i. e., a constant steering wheel angle other than the neutral position, the vehicle’s movement describes a circular arc with a radius determined by the steering angle. Finally, for the transition between linear and circular movements or between circular movements with different curve radii, vehicles move on clothoidal curve segments. The relation between road segments and the resulting steering behavior is shown in Figure 3: Figure 3(a) shows a part of a schematic, but typical highway interchange with linear (red), clothoidal (blue) and circular (green) road segments, and Figure 3(b) plots the corresponding curvature  $\kappa$  over distance  $L$ : the road curvature starts with zero for the line segment, linearly decreases<sup>3</sup> to  $\kappa = -\frac{1}{R}$  in the clothoidal section and then remains constant for the circular arc. Finally, the arc passes into a linear segment via a clothoid with a linear increase of the curvature up to zero.

Since the discussed curve primitives are uniquely defined by their curvature descriptions, a promising approach is to derive the sought-after composition on the basis of collected way point measurements ( $\langle m_j \rangle$ ) by a curvature analysis for this polygonal chain. We will next focus on two realizations of this idea from computer graphics and then work out how the compression and decompression mappings  $c$  and  $r$  from Section II can be implemented using these approaches.

#### A. Basic Clothoid Curve Sketching

In [17], McCrae and Singh present an approach for the modeling of piecewise clothoid curves (*clothoid splines*) to improve the *fairness* of polylines originating from free-hand drawings or other comparably noisy inputs. They propose to approximate the curvature over the arc length by means of linear regression to remove most of the noise in order to fair the input data. The linear regression will, however, rarely result in curvature approximations perfectly indicating

<sup>3</sup>Left turns feature a positive curvature, for right turns it is negative.

circular arcs or line segments; many regression line segments corresponding to noised arcs or lines will rather feature slopes close, but not equal to zero and will thus result in clothoids. To alleviate this, regression line segments with a slope beneath a threshold  $\text{thresh}_s$  are turned into parallels to the x-axis and those parallels with an offset to the x-axis beneath a threshold  $\text{thresh}_o$  are shifted onto the x-axis. After this step, the curve primitive parameters can easily be read from the curvature approximation. Using a smoothness criterion such as  $G^2$  continuity for the complete spline, the primitives are aligned, concatenated and scaled to fit the original polyline with a minimal distance. Two curves meet at a joint point with  $G^d$  ( $d$ -th order geometric) continuity, iff their parameterized functions meet at this joint point with  $C^d$  continuity [18].

This approach works well for short polylines with weak noise levels as they result from free-hand sketching. However, both higher noise levels and longer polylines affect the quality of the computed clothoid splines: first, the errors from the linear curvature regression are integrated twice during the computation of the primitive parameters and thus have a great impact on the final curve. This effect is amplified by errors resulting from the curvature estimation process: though relatively good statistical curvature estimation methods for discrete curves exist [19], curvature calculation for noisy discrete curves still implies at least small errors, especially when the average noise exceeds the distance between adjacent measurement points. Finally, the clothoid spline cannot be modified once the curvature estimation is complete, regardless of the fitting accuracy. Thus, the twice integrated curvature or regression inaccuracies accumulate and can result in high fitting errors that cannot be corrected; this is a critical point especially for long and noisy polylines.

Besides these issues, the basic approach does not provide a direct way to set up a threshold for the resulting fitting or measurement point approximation error as demanded in our problem statement. The problem of the incorrigibility of a clothoid spline is resolved by the second clothoid curve sketching approach that also provides a fitting error threshold parameter; we will discuss this approach in the following.

#### B. Clothoid Curve Sketching using Shortest Paths

The approach of Baran et al. comprises a complex seven-step algorithm to fair polylines using lines, arcs, and clothoids [20]: first, the algorithm *closes polylines* for which the distance between the first and last point falls below a threshold. Second, the algorithm splits the polyline at *corners*, i. e., positions with  $G^0$  continuity, and processes each isolated sub-polyline separately. Each polyline is then *resampled* to adapt the point distribution to the curvature situation, i. e., there should be a higher point density in regions with a high curvature. Afterwards, the algorithm performs a *curve fitting* on the resampled point sequence: for every subsequence of points, curve primitives are constructed regarding an error tolerance. So, instead of simplifying the curvature to find one specific sequence of curve primitives as for the basic approach, a huge number of these are created that could

possibly be used in the final spline. In the fifth step, a *directed, weighted graph* is constructed from the fitted primitives: for every curve primitive, a vertex is created; an edge is inserted between two vertices if the respective curve primitives can be interconnected. The edge weight indicates the quality of the transitions and depends on the fitting errors of the connected curve and transition primitives, and the curve primitive transition continuity ( $G^d$ : the greater  $d$ , the smaller the weight). Unlike the fourth step, the fitting errors caused from this step on are *not* ensured to stay within the error threshold; fitting errors due to new primitive transitions that exceed the threshold are merely penalized more severely to ensure that the primitives from the shortest path can definitely be connected. The *shortest path* through the graph is then calculated based on the edge weights and their transitions between the curve primitives are verified. Due to the cost function and depending on the choice of parameters, it is in fact possible that not only desired  $G^2$  transitions are used, but also  $G^1$  or  $G^0$  transitions, if these imply a lower fitting error. The found path contains the minimal sequence of curve primitives that approximate the given polyline. In the seventh step, these are finally *merged*.

This algorithm constitutes a more promising approach than the basic one, because it does not completely rely on a curvature approximation where errors accumulate. Instead, a sequence of curve primitives is found which takes the effective fitting error into account already during the calculation. However, despite this advantage this algorithm cannot *guarantee* compliance with the error threshold. Also, since our motivation is to find a compression mapping which is to be executed in vehicular on-board units with limited computational capabilities, the high complexity is a clear disadvantage. Furthermore, the algorithm has numerous parameters that influence performance and accuracy; the source code provided by the authors [21] defines more than 30, at least five parameters need to be chosen very carefully. Finally, both approaches suffer from the same problem as the linear approaches presented in Section III: without any further information about the measurement point distribution along the arc, a uniform distribution has to be assumed, which is unlikely to match reality. While the line simplification methods can directly take the uniform distribution assumption into account, this is not possible for the clothoid sketching methods. So, even if the accuracy threshold were definitely not exceeded, assuming a wrong point distribution would cause offset shifts along the spline’s arc, and would thus result in approximation errors.

### C. A Simple Clothoid Spline Compression Scheme

Although these clothoid sketching approaches originally serve the purpose to fair a noisy polygonal chain, and despite the discussed disadvantages, it is nevertheless possible to derive a simple compression scheme based on clothoid sketching. Basically, both approaches fulfill the formal requirements for the compression mapping  $c$ : given a polygonal chain  $\langle m_j \rangle$ , they compute a clothoid spline, i. e., a structure consisting of one or more  $G^1$  or  $G^2$  continuous sequences of parameterized curve primitives as compression result. In the following, we

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.

will only refer to  $G^1$  continuous curve sequences, because  $G^0 \supset G^1 \supset G^2$ . However, due to the severe shortcomings of the basic clothoid spline version, we merely consider the shortest-path approach for a compression scheme.

To set up a compression scheme, we need to find a compact description for such clothoid splines. Since all curve primitives are well-defined by the curvature progression, we will use it as one component of the spline description: every edge point of the curvature graph is encoded as a two-dimensional point; in case of discontinuities (e. g., for  $G^1$  clothoid splines), two points are assigned to one arc length. Thus, in general, each of these  $G^1$  continuous sequences can be uniquely identified by start and end positions as reference points and a curvature progression: by means of the curvature progression data, the curve primitives can be defined and concatenated, before being scaled, rotated and shifted using the start and end points. Then, for each transition with  $G^0$  continuity, an additional two-dimensional position needs to be stored, because in this case, adjoining curves’ orientations are ambiguous and need to be stored explicitly by intermediate points. Given this information, the whole curve primitive spline can be restored, regardless of the continuity degree or what primitives the spline contains in detail. For a simple encoding, we propose to store both the coordinates and curvature progression data elements as two-dimensional points. Without restraining the generality, let us assume that a spline is found consisting of  $u$  primitive subsequences with  $G^1$  continuity; furthermore, let the  $i$ -th subsequence consist of  $v_i$  primitives and let its curvature graph feature  $w_i$  discontinuities. Then, there need to be  $u+1$  necessary two-dimensional reference points and  $v_i+w_i+1$  points for the curvature graph description of the  $i$ -th subsequence. The overall compression result  $\langle m'_j \rangle$  thus consists of  $(u+1) + \sum_{i=1}^u (v_i+w_i+1)$  two-dimensional points.

The only missing information is the point distribution along the spline that approximate the original measurement tuples in  $\langle m_j \rangle$ . As mentioned before, an additional structure would be necessary, such as a mapping of the original measurements’ indices to the respective arc length. There are recent publications using such mappings, such as [22], but their combination with the clothoid sketching approaches in a data compression manner is non-trivial and out of the scope of this paper.

## VI. EVALUATION

In this section, we compare the discussed trajectory encoding methods and analyze their compression performances on the basis of extensive GPS real-world measurements.

### A. Data Acquisition and Parameters

Our evaluation is based on real-world vehicular GPS measurements from one city topology and two highway topologies. Each has a length of several kilometers and several thousand position measurements. A detailed overview is given in Table I. Our GPS measurements had a precision of six decimal places, which results in a discretization error of  $\approx 6$  cm for a latitude of  $52^\circ$ , at which we performed our measurements. For the calculations, we converted the traces to Cartesian coordinates.

In real applications, position measurements can efficiently be logged into static-sized buffers; once such a buffer is full, the content is compressed and transmitted, and the logging begins anew. To simulate such a behavior, we created static-sized subsequences from our position traces by means of a sliding window approach and used these as  $\langle m_j \rangle$ : starting at the beginning of a movement trace, we copied the window content to a subsequence, shifted the window forward by an offset and started over. We set the offset to one fourth of the window size to increase the number of considered subsequences and to spread advantageous or disadvantageous effects over multiple subsequences. We analyzed multiple window sizes and found that there was no significant difference for window sizes larger than 150. We therefore used a subsequence length of  $n = 200$  for our evaluation in this paper. We evaluate all approaches with error thresholds of 0–2 m in steps of 2 cm.

The linear and spline approaches have no parameters except for the error threshold  $\epsilon$ . In contrast, the shortest-path based clothoid sketching software *Cornucopia* [21], requires about 30 parameters that are preset with default values. Amongst others, we made the following parameter adjustments: the costs for  $G^0$  and  $G^1$  transitions were adjusted (from  $\infty$  to 101.0 and 51.0, respectively), because the algorithm failed for several subsequences and error thresholds due to too high default smoothness demands. Also, we increased the error cost (from 5.0 to 52.0), i.e., the amplification factor for fitting errors in the cost model, thus increasing fitting error penalties.

To validate the error threshold compliance of the clothoid sketching approach and thus to make sure that it generates valid compression results, we first analyze the effective fitting error over an increasing error threshold. We need to perform this check only for the clothoid sketching approach, because all other algorithms regard the error threshold as a direct stop criterion for the compression. The results of this evaluation are shown in Figure 4 as box plots; the points show the minimum and maximum values, the whisker ends mark the 0.02 and 0.98 percentiles, the box covers the percentiles from 0.25-0.75 and the band within the box marks the median. Dotted lines from the whisker ends to the points merely help to assign the points to the corresponding whiskers. The diagonal line marks the validation criterion: for the algorithm to produce valid compression results for a given error threshold, no part of the particular box plot may lie above the diagonal. However, the plots reveal an interesting behavior: the effective fitting errors are hardly affected by the value of the error threshold for

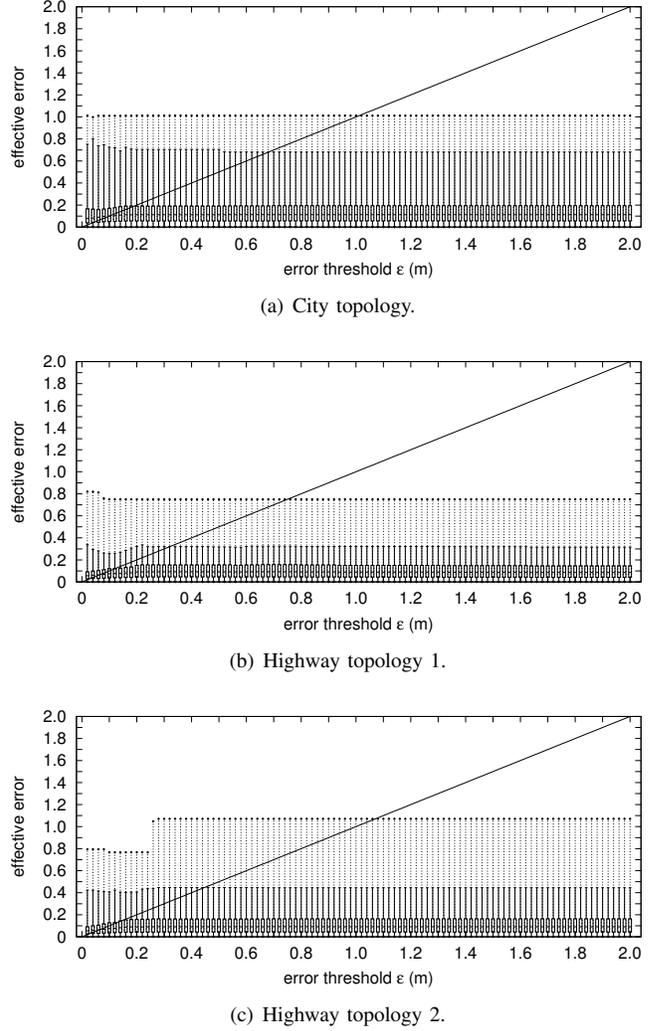


Figure 4. Effective clothoid spline fitting errors.

any topology at all. In fact, it seems that the algorithm quickly reaches an equilibrium for the trade-off between the number of curve primitives and the total fitting cost including error penalties. Thus, its results quickly stagnate. To exclude all invalid compression results from our evaluation, we will take only those error thresholds per topology into account that exceed all measured fitting errors. We will assume a compression ratio of 0% for the results under all other error threshold conditions. Essentially this means that if no compression is possible, then the uncompressed data are transmitted as a fallback solution.

### B. Compression Performance

We first take a closer look at the compression performance of the discussed approaches. To this end, we evaluate the mean compression ratio for each approach and topology, i.e., the mean fraction  $\sigma = 1 - \frac{|J'|}{|J|}$  over all subsequences. The results, visualized as mappings of the error threshold to the compression ratio possible under this condition, are shown in Figure 5. All approaches benefit remarkably from error thresholds of up to 30 cm, as steep increases of all curves in this area clearly show. For higher error thresholds, the curves

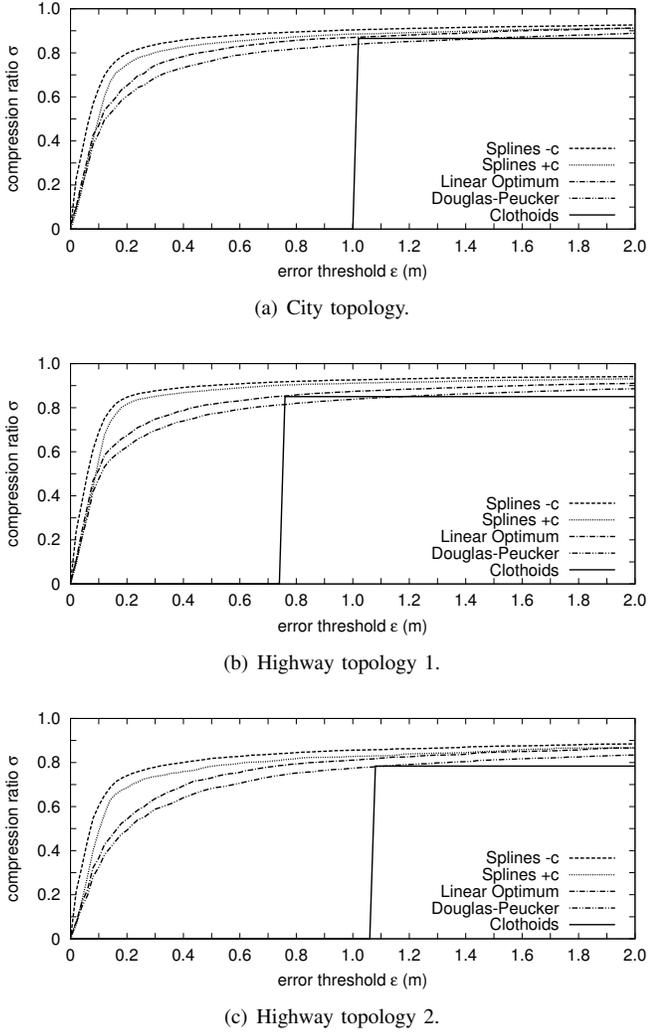


Figure 5. Compression analysis for varying error thresholds.

flatten and thresholds of more than 1 m have merely a minor impact on the compression ratio.

This pattern is basically the same for all topologies, only the slopes of the curves during the initial increase vary and the compression ratios converge to different values. Generally, all approaches perform best for the first highway topology and worst for the second. This can be attributed to the following causes: first, partial roofings over the roadway of the second highway topology as well as trees and buildings in the city topology impaired the reception of satellite signals for navigation, partially resulting in heavy positioning noise. For these settings, the error tolerance needed to be larger for the tested approaches to overcome the noise and approximate the underlying smooth movements. Second, vehicular movements in city scenarios appear to be not as well compressible as on highways. The reason for this trait might be that unsteady steering is usually filtered by on-board electronics at higher velocities, thus causing smoother trajectories.

Also, all topologies show similar compression performance rankings: generally, line simplification performs worst, which

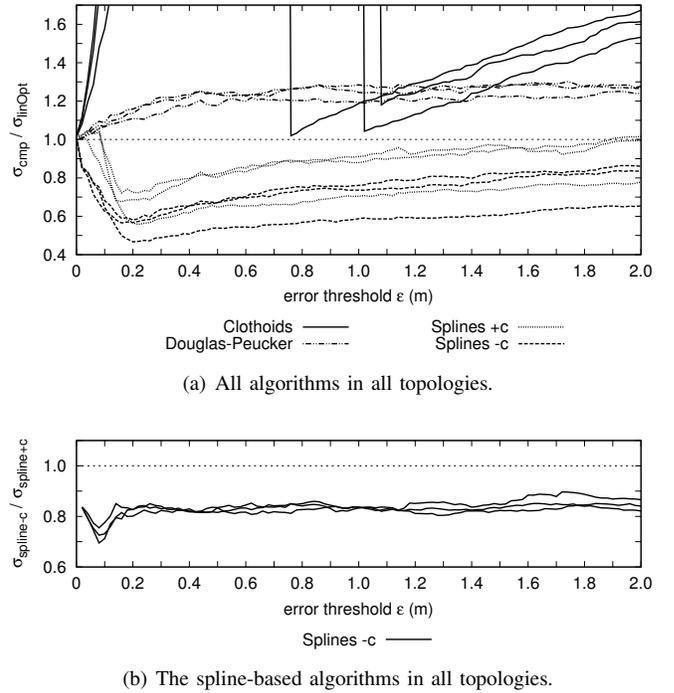


Figure 6. Relative compression analysis for varying error thresholds.

confirms our anticipation that linear methods do not represent vehicular movements as good as non-linear approaches can. The spline-based approaches perform significantly better, providing mean compression ratios of 75–85% for an error threshold of 20 cm with the *context-loosened* approach ('Spline -c') achieving clearly better compression ratios than the basic version. The compression results of the clothoid sketching method show that once no effective fitting error exceeds the threshold, the compression is comparable to those of the line simplification algorithms for every topology. Please note that for the clothoid-spline-based approaches, we can merely plot the results for  $\epsilon$  constraining the *fitting error*, not the *approximation error*; for the necessary encoding of the measurement distribution over the arc length, a significant additional overhead is anticipated, the exact amount of which depends on the employed encoding scheme. Both one-dimensional linear or spline-based approaches could be employed and thus the missing clothoid overhead is assumed to be half as large as the compression results ( $\langle m'_j \rangle$ ) of the respective approach.

Figure 6(a) shows the relative compression performances compared to the linearly optimal approach. It visualizes the ratios  $\frac{\sigma_{\text{cmp}}}{\sigma_{\text{linOpt}}}$  with  $\sigma_{\text{cmp}}$  taking over the values plotted in Figure 5, again for an increasing error threshold. The figure shows that the spline-based algorithms outperform the linearly optimal solution by up to 50% and more. They converge with increasing error threshold, though, because a decreasing number of remaining spline knots in  $\langle m'_j \rangle$  results in subsequences retaining only a fraction of their initial smoothness. Thus, the key feature on which spline-based compression is based deteriorates, while linear approaches are less affected.

Finally, the relative performance of the spline-based compression technique is plotted in Figure 6(b). This analysis shows that for all topologies and unaffected by the error threshold, unseaming the dimension context indeed enables a further performance gain of approximately 20 %.

## VII. RELATED WORK

We have already reviewed the most relevant previous work, i. e., vehicular movement approximation by linear approximation [7]–[10] and clothoids [17], [20] in the course this paper. We also compared the compression performances of the employed mechanisms. Besides that, Bézier curves [23], Splines [24] or clothoids [25] were previously used for modeling vehicular trajectories for autonomous vehicles using map material, but not based on position measurements. The only previous work on trajectory data reduction with non-linear functions has been presented in the context of spatiotemporal databases [26], [27]. There, Chebyshev polynomials are used as approximations of so-called *minimax polynomials*; their maximum approximation error is minimal for the given approximation parameters. However, both approaches use the polynomial degree as an input parameter and determine the approximation error *after* the calculation.

## VIII. CONCLUSIONS

In this work we investigate lossy compression schemes for the encoding of vehicular trajectories, using linear approximation, cubic splines, or clothoids. Linear approximation turns out to be the least computationally expensive form of lossy compression; it also provides reasonable compression ratios and does not rely on the specific mobility pattern of vehicles. Using cubic splines, on the other hand, extensively exploits these smooth vehicular movements. This approach provides higher compression ratios, in particular if only a small loss in accuracy can be tolerated. This advantage comes at the cost of a higher computational complexity. Finally, clothoid-based approaches prove their potential for very competitive compression ratios. However, the reliance of current clothoid-based approaches on a large parameter set and their very high complexity prevents their direct application to vehicular communication at this time. It is entirely possible, though, that future work on clothoid-based approaches could change this.

## ACKNOWLEDGMENTS

We thank Michael Singhof for his support in the evaluation and Ilya Baran for his support for the *Cornucopia* software.

## 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 *ITSWC '02: Proceedings of the 9th World Congress and Exhibition 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] 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*, Sept. 2008.
- [5] W. Huber, M. Lädke, and R. Ogger, "Extended floating-car data for the acquisition of traffic information," in *ITSWC '99: Proceedings of the 6th World Congress and Exhibition on Intelligent Transportation Systems and Services (ITS)*, Nov. 1999.
- [6] R. Brüntrup, S. Edelkamp, S. Jabbar, , and B. Scholz, "Incremental map generation with gps traces," *itsc05*, Nov. 1999.
- [7] 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, Oct. 2001.
- [8] A. Civilis, C. S. Jensen, and S. Pakalnis, "Techniques for efficient road-network-based tracking of moving objects," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 698–712, May 2005.
- [9] 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.
- [10] 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, Mar. 2009, pp. 184–193.
- [11] M. Koegel, W. Kiess, M. Kerper, and M. Mauve, "Compact Vehicular Trajectory Encoding," in *VTC '11-Spring: Proceedings of the 73rd IEEE Vehicular Technology Conference*, May 2011.
- [12] 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, Dec. 1973.
- [13] J. Hershberger and J. Snoeyink, "Speeding Up the Douglas-Peucker Line-Simplification Algorithm," in *SDH '92: Proceedings of the 5th International Symposium on Spatial Data Handling*, 1992, pp. 134–143.
- [14] 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.
- [15] K. G. Baass, "The use of clothoid templates in highway design," *Transportation Forum*, vol. 1, pp. 47–52, 1984.
- [16] D. S. Meek and D. J. Walton, "Clothoid spline transition spirals," *Mathematics of Computation*, vol. 59, no. 199, pp. 117–133, July 1992.
- [17] J. McCrae and K. Singh, "Sketching piecewise clothoid curves," *Computers & Graphics*, vol. 33, no. 4, pp. 452–461, 2009.
- [18] B. A. Barsky and A. D. DeRose, "Geometric Continuity of Parametric Curves," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-84-205, Oct. 1984.
- [19] E. Kalogerakis, P. Simari, D. Nowrouzezahrai, and K. Singh, "Robust statistical estimation of curvature on discretized surfaces," in *ESSGP '07: Proceedings of the Eurographics/ACM Siggraph Symposium on Geometry Processing*, 2007.
- [20] I. Baran, J. Lehtinen, and J. Popovic, "Sketching Clothoid Splines Using Shortest Paths," *Computer Graphics Forum*, pp. 655–664, 2010.
- [21] I. Baran, "Cornucopia: a clothoid sketching software," online resource, <http://code.google.com/p/cornucopia-lib/>.
- [22] L. Krol, "The reconstruction of vehicle trajectories with dynamic macroscopic data," Master's thesis, University of Twente, Enschede, Netherlands, Aug. 2009.
- [23] 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.
- [24] C. Dever, B. Mettler, E. Feron, J. Popović, and M. Mcconley, "Non-linear trajectory generation for autonomous vehicles via parameterized maneuver classes," *Journal of Guidance, Control and Dynamics*, vol. 29, pp. 289–302, 2006.
- [25] 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.
- [26] 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*, June 2004.
- [27] 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.