On the Spatio-Temporal Information Content and Arithmetic Coding of Discrete Trajectories

Markus Koegel and Martin Mauve

Department of Computer Science, University of Düsseldorf, Germany,
{koegel,mauve}@cs.uni-duesseldorf.de,
http://www.cn.uni-duesseldorf.de/

Summary. The trace of a moving object is commonly referred to as a trajectory. This paper considers the spatio-temporal information content of a discrete trajectory in relation to a movement prediction model for the object under consideration. The information content is the minimal amount of information necessary to reconstruct the trajectory, given the movement model. We show how the information content of arbitrary trajectories can be determined and use these findings to derive an approximative arithmetic coding scheme for trajectory information, reaching a level of compression that is close to the bound provided by its entropy. We then demonstrate the practical applicability of our ideas by using them to compress real-world vehicular trajectories, showing that this vastly improves upon the results provided by the best existing schemes.

1 Introduction

Gathering, storing and transmitting data on the movement of objects are common parts of many applications in the area of ubiquitous computing. These data, referred to as *trajectories*, basically comprise sequences of position and time measurements. Given that storage space and transmission capacity are valuable resources, in particular in the mobile domain, it is desirable to encode trajectories efficiently, e.g., by means of compression.

In general, compression methods seek to identify and remove redundancies from an input source; in this paper, we focus on redundancy within trajectories. This redundancy results from underlying principles of object mobility, such as kinematics or Newton's laws of motion. It is widely accepted that these principles cause mobility to be predictable to some degree; for example, several approaches have been proposed that use linear models for the compression of trajectories [1, 2], though non-linear models have also been discussed recently [3, 4].

However, no previous work has regarded the general upper bound for trajectory compression that is given by the *information content*, or *entropy*, of such a movement trace. In this paper, we therefore focus on the following question: given a prediction of how an object moves, how much information does a trajectory contain and what upper compression bound does this imply? Then, we use this knowledge to construct a compression scheme based on arithmetic coding that comes very close to reaching this bound.

Throughout this work we use the vehicular domain as an example to illustrate our findings and to prove its applicability to real world data. However, the ideas presented here can be used to analyze and compress any form of trajectory, provided that a prediction model for the respective mobility can be constructed.

In the remainder of this paper, we present related work on trajectory compression and probabilistic positioning in Section 2. We introduce our idea of information content for trajectories and how to measure it in Section 3. In Section 4, we discuss how to apply this idea to vehicular trajectories and briefly describe details of an arithmetic coder implementing our model. The formal model and the arithmetic coder are finally evaluated in Section 5.

2 Related Work

The *compression of movement measurements* is frequently discussed in the context of *Mobile Object Databases (MODs)*. MODs receive trajectory updates from arbitrary mobile units and can handle spatio-temporal search requests. For MODs, compression techniques have been proposed that either require an already completed data collection (*offline* approaches, such as [5, 6]) or that compress data on the fly (*online* approaches, such as [1, 2]). For both approaches, line simplification or linear dead reckoning algorithms have been used. The compression performance of both are upper-bounded by the optimal line simplification proposed in [7]. The authors of [3] describe trajectories using so-called *minimax* polynomials, approximating a given function, so that the maximum approximation error is minimized for the employed parameter values. Further compression techniques that focus on vehicular trajectories use cubic splines [4] and clothoids [8] as described in [9]; in general, these non-linear approaches attempt to model the smoothness of vehicular movements or roadways.

In robot navigation, *Probabilistic Positioning* is employed for self positioning, e. g., within office buildings [10, 11]: instead of precise positions, probabilities for the position on a discrete map are given. We work with a similar concept by defining a probability distribution over a limited region, but do not rely on previously known map material.

In [12], navigation decisions of pigeons are analyzed based on the stochastic complexity of trajectory sections by deriving the navigational confidence. The authors of [13] propose user movement estimation for cellular networks with Markov models. They determine state transition probabilities based on relative location frequencies and use these to derive compressed position update messages. Both approaches are special cases for information content measurements, but cannot directly be generalized to arbitrary movements. In this paper, we present a formal model that not only can be seen as a generalization of these approaches but can also be adapted to any other application area.

None of the existing approaches consider a general upper bound for trajectory compression that is given by the information content, or entropy, of such a movement trace. We will show in the following, that doing so will lead to significant improvements in the compression ratio of trajectories.

3 The Information Content of Trajectories

In this section, we will show what the information content of a trajectory is and how it can be measured. To this end, we will introduce a formal model for the entropy calculation of trajectories and discuss its components and parameters.

3.1 What is the Information Content of Trajectories?

Any object movement, e. g., the migration patterns of flocks, the movement of astronomical objects or the trajectories of road vehicles can be described by a formal model. In general, such models can be used for movement predictions of particular objects based on previous position measurements, exploiting the redundancy and predictability of movements. Since typically not all factors influencing the mobility of an object can be modeled accurately, the *actual* position of the object might differ from the prediction. This deviation is commonly referred to as *innovation*, i. e., the uncertainty of the prediction process.

In this paper, we investigate the information content of the innovations. Let us start introducing the necessary information theoretical concepts from [14]: given an ensemble $X = (x, \mathcal{A}_X, \mathcal{P}_X)$, with the outcome x of a random variable over the finite set $\mathcal{A}_X = \{a_1,\ldots,a_I\}$ and according probabilities $\mathcal{P}_X = \{p_1,\ldots,p_I\}$: $p_i = P(x = a_i), \forall 1 \leq i \leq I : p_i \geq 0$ and $\sum_{i=1}^{I} p_i = 1$. We refer to $\mathcal{A}_X$ as the discrete *alphabet* and to $\mathcal{P}_X$ as the ensemble's *probability distribution.* The *Shannon information content* (in *bits*) of x is defined as $h(x) = \log_2(1/P(x))$, where $P(x)$ is the probability of its occurrence. Then, the *entropy* $H(X)$ refers to the *average information content* of an outcome of X and is defined as $H(X) = \sum_{x \in \mathcal{A}_x} P(x)h(x) = \sum_{x \in \mathcal{A}_x} P(x) \log_2(1/P(x))$. In other words, the entropy is the average number of bits that is needed to represent an outcome x.

If we apply this definition to our previous discussion, we can identify the estimation innovation as the outcome of each estimation step: it represents the estimation uncertainty and bears the information that was missing when the prediction was made. So, if the innovations of all position estimation steps are regarded, we can derive the information content of a whole movement trace. On the other hand, the ensemble's alphabet and the probability distribution yield the entropy of the outcome, being the average information content.

3.2 How to determine the Information Content of Trajectories

Basically, each outcome x of an ensemble X is determined by the employed movement estimator. Therefore, we need to formalize all involved components: the *movement estimator* and the parts of X, namely the *alphabet* of possible values $\mathcal{A}_X$ and the set of corresponding probabilities $\mathcal{P}_X$.

The movement estimator is a function θ that determines a two-dimensional position based on an observation vector $m = (m_1, \ldots, m_{N-1})$ containing previously collected position measurements: $\theta : (\mathbb{R}^2)^{N-1} \rightarrow \mathbb{R}^2$, $\theta(m) = \hat{m}_N$. Then, the innovation i_N is the difference between the estimation and the actual measurement: $i_N = m_N - \hat{m}_N$, $i_N \in \mathbb{R}^2$. So, the innovation is a two-dimensional real vector itself and cannot directly be used for the outcome x, because $\mathbb{R}^2$ is uncountably infinite, i. e., neither countable nor bounded. To process the innovation into an outcome x, we need to overcome these two issues.

The innovation domain can be made countable by simple discretization: the real innovation vector is mapped onto a grid, with each grid node referring to a particular symbol in $\mathcal{A}_X$. The grid cell width and form are accuracy parameters, their choice is influenced by several aspects, e.g., the highest tolerable discretization error or the highest discretization error under which the movement model still produces reasonable results.

Once countable, the innovation domain can be bounded, while still keeping *all* reasonable innovations covered by $\mathcal{A}_X$. That is, all possible positions within reach in the time period since the last measurement need to be mappable to $\mathcal{A}_X$. Which positions can be reached depends, e. g., on the movement model, or measurement noise. The limitation of the innovation domain is important, because the resulting alphabet needs to be valid for the whole trajectory: if the limits are set too narrow, i. e., $\mathcal{A}_X$ misses reasonable innovations, such innovations could not be covered by the ensemble X. Contrariwise, too wide limits would include implausible innovations in $\mathcal{A}_X$ and thus would increase its entropy, which then could be significantly higher than the actual entropy of the trajectory.

Once the movement estimator and the alphabet are known, $\mathcal{P}_X$ is set up by assigning a probability to each symbol in $\mathcal{A}_X$. Like the alphabet, the probability distribution is crucial for the result of the entropy determination.

So, the entropy of an ensemble $X = (x, \mathcal{A}_X, \mathcal{P}_X)$ can be determined directly. To measure the information content of a trajectory, the deviations between the predicted positions and the actual position measurements are mapped to $\mathcal{A}_X$. Then, the information content of each measurement can be determined.

4 Exemplary Implementation of Information Measurement

We can now apply the necessary parts for the determination of a trajectory's information content to a specific use case and show how to implement these components for vehicular trajectories.

4.1 Movement Estimator

Consider position measurements $p = (p_x p_y) \in \mathbb{R}^2$; then, the velocity ($\mathbf{v}$) and acceleration ($\mathbf{a}$) vectors of a vehicle at the position p_i at the time t_i are given by: $\mathbf{v}_i = (p_i - p_{i-1})/(t_i - t_{i-1})$, $\mathbf{a}_i = (\mathbf{v}_i - \mathbf{v}_{i-1})/(t_i - t_{i-1})$. With these quantities, simple vehicular movement models can be set up as described in [15]: the first movement model only considers the last position and velocity vector:

$$\theta_{vel} = p_{i_{N-1}} + \mathbf{v}_{i_{N-1}} \mathit{\Delta} t \ . \tag{1}$$

The second model extends the first one by using the approximated acceleration:

$$\theta_{acc} = p_{i_{N-1}} + \mathbf{v}_{i_{N-1}} \mathit{\Delta} t + \frac{\mathbf{a}_{i_{N-1}}}{2} \mathit{\Delta} t^2 \ . \tag{2}$$

Obviously, more complex movement models, e. g., using sensor data fusion, are conceivable. We will show that these simple models already perform very well and thus defer the investigation of other movement models to future work.

4.2 The Discrete Alphabet $\mathcal{A}_X$

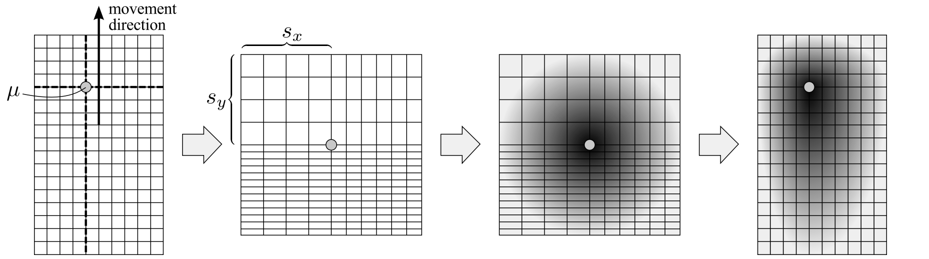
As described above, the innovation domain can be made countable and bounded by projecting each innovation to a grid of limited size. It is clear that the specific design of the grid depends on the application context; for example, in the vehicular domain, a uniform approximation error for any region of the grid is desirable, making regularly tessellated grids an interesting option. Additionally, the grid cell dimensions and density can be easily adjusted by a maximal discretization error ϵ. We will thus use regular square grids for our examination.

While setting up the grid cells is a simple task, the limitation of the scope of the grid is more challenging, because the grid needs to cover all reasonable (and only those!) measurement innovations. For the use case of vehicular movements, the grid boundaries strongly depend on the possible movements of a vehicle. Therefore, we use kinematic model to determine them.

For the dimensioning of the grid, we will refer to a logical—not necessarily geometrical—grid center, at which the grid will be aligned along the movement direction. We set this grid center to the estimated next position according to the non-accelerated movement model (1), disregarding both acceleration and steering. This enables us to dimension the grid with a simple line of kinematic arguments: the longitudinal dimension—i. e., the dimension along the movement direction—directly results from the highest possible deceleration $\mathrm{dec}_{\mathrm{max}}$ and acceleration $\mathrm{acc}_{\mathrm{max}}$ that could cause a position deviation from the predictand. We calculate the lateral dimension—i. e., the dimension perpendicular to the movement direction—by the maximal possible change in steering since the last measurement: with Coulomb's friction law and common friction values [15], we can determine the highest velocity $v_{max}(r)$ that can be driven in curves with radius r. The lateral dimension is then calculated directly based on $v_{max}(r)$.

However, the grid dimensions depend, next to kinematic influences, on the positioning noise level. In extreme cases, increased noise levels may cause innovations to lie outside the analytically deduced boundaries. For such situations, we add a single extra symbol to $\mathcal{A}_X$, representing outliers; this minimally increases the alphabet size and the entropy of the ensemble, though. Additionally, expected or current noise statistics, e. g., *dilution of precision (DOP)* values, could be regarded for the grid dimensioning. This would result in some kind of *guard zone* around the analytically set up grid that could reduce, but not completely avoid the probability of outliers. The setup of such a zone is nontrivial and beyond the scope of this paper and thus remains future work.

These approaches can be implemented complementary, depending on the expected or experienced noise level. Furthermore, filters can be employed to odd out or smooth implausible position measurements. Some filters, such as the (extended) Kalman filter, even provide probabilities or covariances of the quality of the estimate that could be employed in dimensioning the grid.



1. Separate the grid into quadrants, using μ.

2. Scale each quadrant of the grid to size $[s_x; s_y]$.

3. Calculate the probability p_i of each grid node.

4. Normalize probabilities, so $\sum_i p_i = 1$ applies.

Fig. 1. Skewing of the probability distribution and mapping it to the grid nodes.

4.3 The Probability Distribution $\mathcal{P}_X$

The probability distribution of the ensemble X assigns a probability p_i to each symbol $a_i \in \mathcal{A}_X$, as described in Section 3. We expect $\mathcal{P}_X$ to be reasonably close to the normal distribution which is commonly used in the context of noisy position measurements [16]. While it is unlikely that the innovations during an estimation process will be perfectly normal distributed, we consider this to be a good approximation. This assumption will be further evaluated in Section 5.

As explained above, we derive the alphabet from a two-dimensional grid, so we need to employ a *bivariate* normal distribution $\mathcal{N}(\mu, \Sigma)$, $\mu = (\mu_x, \mu_y)$, $\Sigma = (\ (\sigma_x^2, \rho\sigma_x\sigma_y)^T, (\rho\sigma_x\sigma_y, \sigma_y^2)^T)$ with the standard deviations σ_x, σ_y and the correlation coefficient ρ [17]. The distribution's mean is set according to the estimated next position: while the grid is aligned using the non-accelerated movement model (1), the mean μ of the probability distribution can be determined using other models, e. g., the accelerated movement model (2). We do not expect the dimensions of the innovation domain to correlate, so $\rho = 0$. However, the normal distribution is symmetric, but the discretization grid may not necessarily be. For simplicity, we propose a projection of the grid to implement a skewed probability distribution, which is schematically depicted in Figure 1: the mean μ separates the grid into four quadrants, each of which is scaled to the dimensions $[s_x; s_y]$, where s_z is the number of standard deviations that are supposed to cover the z axis of each quadrant. Due to the scaling, the standard deviation of the distribution can be set to $\sigma = 1$ and so, the probabilities for the grid nodes can be determined using the simplified density function $f(x, y) = \frac{1}{2\pi}\ \exp(-\frac{1}{2}\left(x^2 + y^2\right))$. Afterwards, the assigned probabilities need to be normalized to eliminate scaling effects, so that $\sum_{i=1}^{I} p_i = 1$ applies again.

Alternatively, asymmetric probability distributions could be employed, e. g., the log-normal distribution.

4.4 Exemplary Alphabets and their Entropies

We are now able to determine the entropy of particular ensembles, i. e., the expected information content per position measurement. Table 1 shows an

Table 1. Exemplary alphabet configurations and resulting entropies.

			entropy [bits]		
Δt	ϵ	# symbols	uniform	normal (3σ)	normal (4σ)
0.5 s	0.1 m	199	7.64	6.32	5.55
	0.5 m	16	4.00	2.38	1.77
	1.0 m	7	2.81	1.83	1.61
1.0 s	0.1 m	2761	11.43	10.25	9.47
	0.5 m	136	7.09	5.73	4.97
	1.0 m	41	5.36	3.86	3.20
2.0 s	0.1 m	41580	15.34	14.20	13.42
	0.5 m	1760	10.78	9.59	8.80
	1.0 m	476	8.89	7.65	6.86

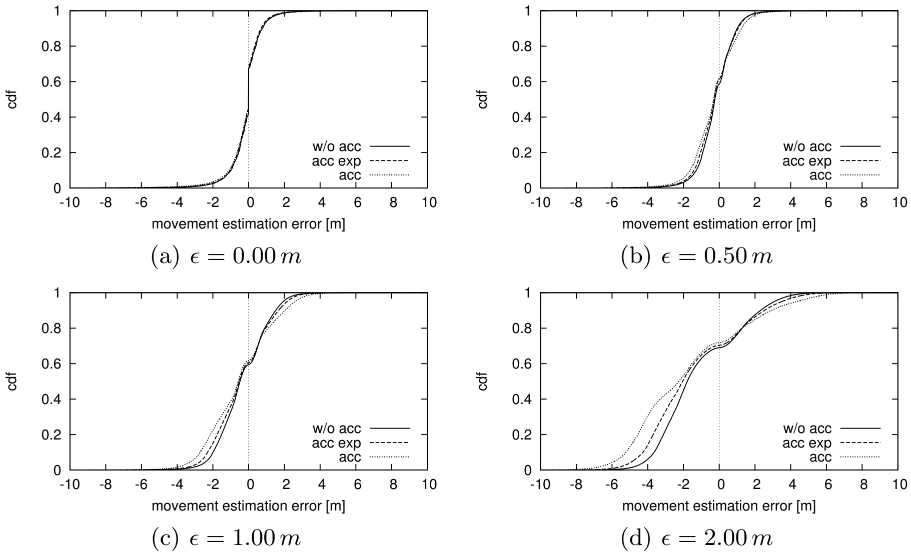
overview of exemplary entropies for varying measurement intervals and accuracy bounds. For the regular square grid setup, we assumed an acceleration interval $[\text{dec}_{\text{max}}; \text{acc}_{\text{max}}] = [-11; 8]\,\frac{m}{s^2}$. Also, we calculated the entropies for probability distributions with two different standard deviations: We chose $s_x = s_y = 3\sigma$ and $s_x = s_y = 4\sigma$, thus assuming that approximately 99.7 % and 99.99 % of all measurement innovations will lie within the grid, respectively. Please note that the entropy, as an expected value, solely depends on the used movement model θ and ensemble X and not on actual measurements.

We can see from the table that even for very high accuracy demands, the expected average information content per symbol is very low: while position measurements are normally provided by off-the-shelf GPS receivers as fixed-point numbers with six digital places and thus can be encoded with 57 bits, the expected average information content at an accuracy bound of 0.1 m always lies below 16 bits. This would even be true if the probability distribution of the alphabet was normal distributed. According to our model, the entropy becomes smaller with the measurement interval. This is reasonable, because the more information is provided by measurements, the lower is the missing information for a perfect estimation.

4.5 Model Implementation: An Arithmetic Coder

The estimation results presented in Table 1 encouraged us to build an arithmetic coder based on our formal model, since we could expect compression rates of more than 90 % even at tight accuracy bounds. For the implementation, we used the arithmetic coding project of Bob Carpenter [18], version 1.1 as a basis.

The only modification to our formal model lies in the handling of outlier measurements: Instead of adding an extra outlier symbol, we stop the arithmetic encoding once an innovation cannot be mapped onto $\mathcal{A}_X$. This is inevitable, because the mapping of an discretized innovation to a code symbol needs to be bijective; this is not fulfilled in case of outliers. Once an innovation cannot be mapped to a grid node, it would not be possible to retrieve a valid grid node in the decoding process. Therefore, in this case the symbol stream is terminated



(a) $\epsilon = 0.00\,m$

(b) $\epsilon = 0.50\,m$

(c) $\epsilon = 1.00\,m$

(d) $\epsilon = 2.00\,m$

Fig. 2. Movement estimation error analysis (urban movement).

with the *End Of Stream* symbol, the probability distribution and the estimator are reset and a new encoding begins. This is an undesirable situation, because at least one position needs to be transmitted uncoded; so, even with the mentioned *guard zone*, using a robust estimator is crucial for the compression performance.

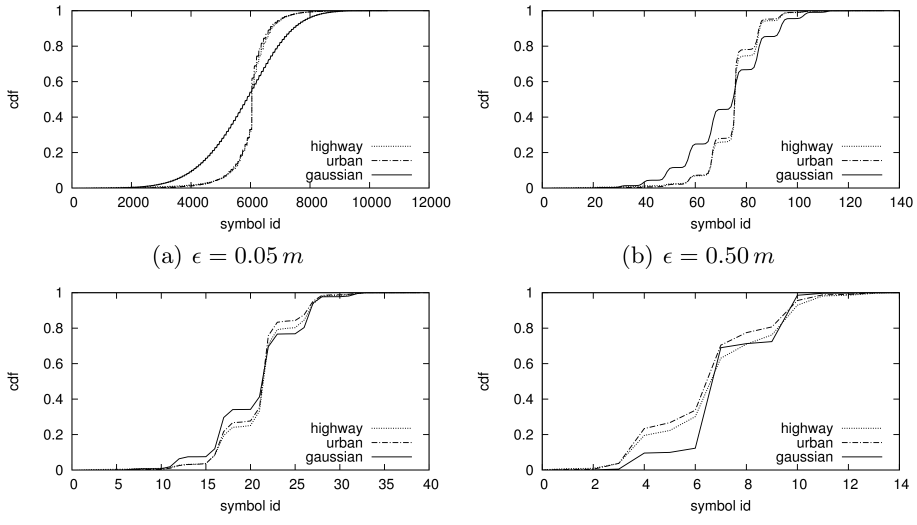
5 Evaluation

5.1 Movement Data

The results that we present here are based on real-world movement data from the *OpenStreetMap* project [19]. Those are available under the Creative Commons license BY-SA 2.0 [20]. We categorized each movement trace based on the highest object velocity v_{max} as *urban* ($8.3\,\frac{m}{s} < v_{max} < 17\,\frac{m}{s}$) or *highway* ($v_{max} \geq 17\,\frac{m}{s}$) movements. We then selected only those traces with 1 Hz measurement frequencies and more than 100 measurement points; in doing so, we retrieved 2263 and 4946 traces for the urban and highway pattern, respectively.

5.2 Movement Estimation

Since it is essential to use an accurate and robust movement model, we performed movement estimations with the non-accelerated and accelerated movement estimators. We were in particular interested in the influence of the discretization and therefore analyzed the movement estimation inaccuracies for varying discretization steps; the results are shown as cumulative distributions in Figure 2.



(a) $\epsilon = 0.05\,m$

(b) $\epsilon = 0.50\,m$

(c) $\epsilon = 1.00\,m$

(d) $\epsilon = 2.00\,m$

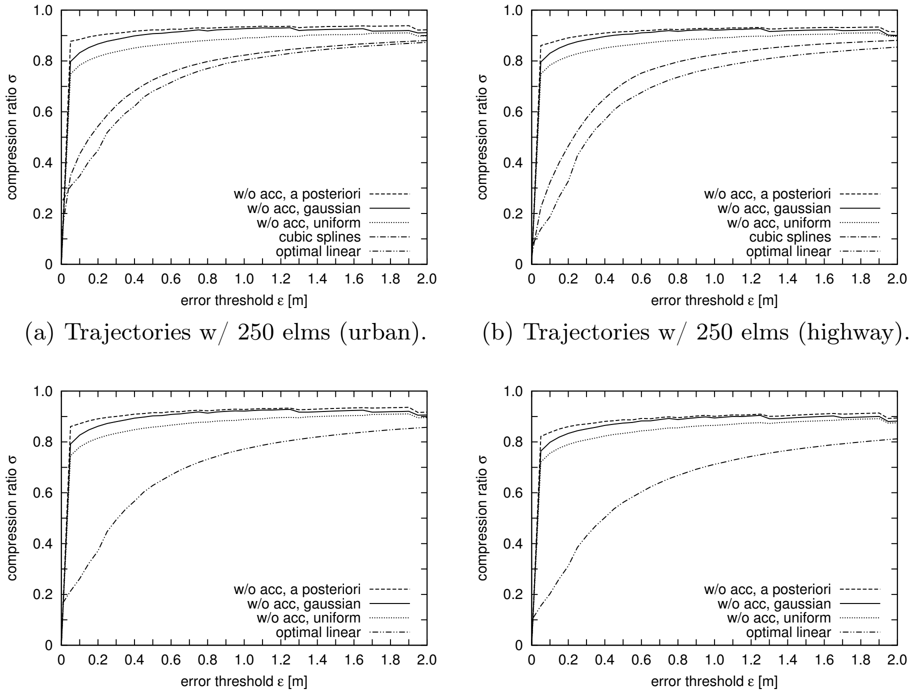
Fig. 3. Cumulative distribution analysis of the probability distribution $\mathcal{P}_X$.

Without discretization (i. e., $\epsilon = 0$), the error was below 2 m in 90-95% of the cases, with slightly worse results for the highway topologies, which we left out for lack of space. For increasing ϵ, the error distributions widen, which indicates that coarser discretizations lower the quality of the estimator's observation vector. This in turn has a direct influence on the information content and the compression performance.

Unexpectedly, the non-accelerated estimator (*w/o acc*) outperforms the accelerated variant (*acc*). The latter is more impaired by the discretization, since it derives acceleration values from distorted velocities. To reduce this effect, we amended the accelerated estimator by smoothing the acceleration values exponentially (*acc exp*). Though this really improves the situation, it does not provide the same robustness as the non-accelerated estimator as shown in Figure 2.

5.3 Probability Distribution

To evaluate the probability distribution, we compare the assumed (gaussian) and actual cumulative distribution functions for both topologies over the respective $\mathcal{A}_X$ in Figure 3. To this end, we serialized the two-dimensional distributions over the grid to one-dimensional distributions over the ordered symbol set to gain a better overview: we concatenated the symbols from cross-sections along the lateral grid axis, causing stepped curves, where each "step" in the curves refers to one such cross-section. The cdf graphs for the normal distribution resemble the empirically determined ones, though the distributions for both the urban and highway topologies are denser, especially for lower values of ϵ. This basically



(a) Trajectories w/ 250 elms (urban).

(b) Trajectories w/ 250 elms (highway).

(c) Uncut trajectories (urban).

(d) Uncut trajectories (highway).

Fig. 4. Compression ratio comparison.

confirms our assumption for $\mathcal{P}_X$, though it is also quite obvious that there is room for future research to improve the probability distribution and thereby the information content estimation and compression performance.

5.4 Compression

For the evaluation of the compression performance, we use the compression schemes based on optimal line simplification and cubic spline interpolation discussed in [9] as benchmarks for the work presented here.

The spline-based approach runs in $O(n^3)$ and was designed for relatively short trajectories of ∼250 elements; so, we selected typical trajectories of 1000-1300 positions and cut them to slices of 250 elements, each slice overlapping the previous one by 100 elements, and thus gained 318 and 327 shorter traces for the urban and highway topologies, respectively. In doing so, we avoid side effects and spread both advantageous and disadvantageous effects on multiple slices.

Both the line simplification and the arithmetic coding are capable of handling trajectories of several thousand measurements, so we additionally apply these to uncut traces in order to gain a broader basis for the analysis and to examine the ability of these approaches to handle data streams of variable length.

As an optimal benchmark for the arithmetic coder, we have performed compressions using *a posteriori* knowledge: we measured the empirical distributions of the code symbols for each trajectory and used these as stochastic models for the arithmetic coder. This is only a theoretical optimum, because for a productive operation, this distribution would have to be transmitted along with the code bit stream and would cause a serious overhead.

Figures 4(a) – 4(d) depict the average compression ratios against the discretization threshold ϵ. For all configurations, the same ranking of compression techniques is visible: the optimal line simplification algorithm performs worst, being outperformed by the approach driven by cubic splines (only for the trajectories with 250 elements in Figures 4(a) and 4(b)). The arithmetic coding approach performs best, especially for very tight accuracy bounds of $\epsilon < 1.0\,\mathrm{m}$ and even if a uniform probability distribution is used for $\mathcal{P}_X$ as reference. When a normal distribution is used, the compression ratios improve by another 20-30%. The results for using a posteriori knowledge are significantly better for $\epsilon < 0.5\,\mathrm{m}$, but thereafter are almost reached by the compression ratios with the normal distribution assumption. This underlines our findings from the probability distribution analysis that for growing ϵ, the impact of the probability distribution decreases. We only plot the compression results achieved with the non-accelerated movement estimator for a better overview. According to Section 5.2, it yielded slightly better compression ratios than the accelerated models.

An interesting effect are the visible drops of the compression ratios for very high ϵ that occur for all distributions and topologies alike. These originate from the discretization grid that is getting coarser for growing ϵ.

6 Conclusions

In this paper we determined the information content and entropy of trajectories with respect to a prediction model. Further, by using these findings we were able to specify a compression scheme based on arithmetic coding. We demonstrated the practical applicability of our ideas by using them to compress vehicular trajectories and applied this to a large number of heterogeneous real-world vehicular movement traces. The results from this evaluation show that our approach is far superior to the best existing compression scheme for vehicular trajectories.

There are two aspects of the work presented here that could be improved further. First, the prediction model we use is rather simple. It might make sense to investigate other models. After all, the more accurate the prediction is, the lower will be the information content of the trajectory in relation to that model. Second, we have used a normal distribution to assign probabilities to the alphabet. While this is an acceptable approximation it might pay off to investigate other ways to assign those probabilities.

Acknowledgments. The authors wish to thank Dennis Dobler for his work on the prototype of the arithmetic coder and the formal model. Also, the authors thank Bob Carpenter for the source code of his arithmetic coder and his support.

References

1. 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*, Mar. 2009, pp. 184–193.
2. N. Hönle, M. Großmann, 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.
3. J. Ni and C. V. Ravishankar, “Indexing Spatio-Temporal Trajectories with Efficient Polynomial Approximations,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 663–678, May 2007.
4. 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.
5. 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.
6. H. Cao, O. Wolfson, and G. Trajcevski, “Spatio-temporal data reduction with deterministic error bounds,” *VLDB Journal*, vol. 15, no. 3, pp. 211–228, Sept. 2006.
7. 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.
8. I. Baran, J. Lehtinen, and J. Popovic, “Sketching Clothoid Splines Using Shortest Paths,” *Computer Graphics Forum*, pp. 655–664, 2010.
9. M. Koegel, D. Baselt, M. Mauve, and B. Scheuermann, “A Comparison of Vehicular Trajectory Encoding Techniques,” in *MedHocNet ’11: Proceedings of the 10th Annual Mediterranean Ad Hoc Networking Workshop*, June 2011.
10. D. Fox, “Markov localization: A probabilistic framework for mobile robot localization and navigation,” Ph.D. dissertation, University of Bonn, Germany, 1998.
11. N. Roy, W. Burgard, D. Fox, and S. Thrun, “Coastal navigation – mobile robot navigation with uncertainty in dynamic environments,” in *ICRA ’99: Proceedings of the IEEE Int’l Conference on Robotics and Automation*, Aug. 1999, pp. 35–40.
12. S. Roberts, T. Guilford, I. Rezek, and D. Biro, “Positional entropy during pigeon homing i: application of bayesian latent state modelling.” *Journal of Theoretical Biology*, vol. 227, no. 1, pp. 39–50, 2004.
13. A. Bhattacharya and S. K. Das, “LeZi-update: an information-theoretic approach to track mobile users in PCS networks,” in *MobiCom ’99: Proceedings of the 5th Annual ACM/IEEE Int’l Conf. on Mobile Computing and Networking*, July 1999.
14. D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms.* New York, NY, USA: Cambridge University Press, 2002.
15. H. Kuchling, *Taschenbuch der Physik*, 17th ed. Fachbuchverlag Leipzig im Carl Hanser Verlag, Aug. 2007, in German language.
16. F. van Diggelen, “GPS Accuracy: Lies, Damn Lies and Statistics,” *GPS World*, vol. 9, no. 1, pp. 41–45, Nov. 1998.
17. N. Timm, *Applied multivariate analysis*, ser. Texts in statistics. Springer, 2002.
18. B. Capenter, “Arithcode project: Compression via arithmetic coding in java. version 1.1,” online resource, 2002, http://www.colloquial.com/ArithmeticCoding/.
19. “The OpenStreetMap Project,” online resource, http://www.openstreetmap.org/.
20. “Creative Commons BY-SA 2.0,” http://creativecommons.org/licenses/by-sa/2.0/.