Noname manuscript No.
(will be inserted by the editor)

A Detailed View on the Spatio-Temporal Information Content and the Arithmetic Coding of Discrete Trajectories

Markus Koegel · Matthias Radig · Erzen Hyko · Martin Mauve

Received: date / Accepted: date

Abstract 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 state-of-the art compression schemes for spatio-temporal data.

Keywords Spatio-Temporal Data · Information Content · Data Compression · Vehicular Movement

CR Subject Classification I.2.9 [**Artificial Intelligence**]: Robotics—*Kinematics and dynamics* · I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Motion* · H.1.1 [**Models and Principles**]: Systems and Information Theory—*Information theory* · E.4 [**Coding and Information Theory**]: Data compaction and compression

Department of Computer Science, University of Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany
E-mail:
{ koegel, mauve }@cs.uni-duesseldorf.de,
{ matthias.radig, erzen.hyko }@hhu.de
http://www.cn.uni-duesseldorf.de/

1 Introduction

Gathering, storing and transmitting data on the movement of objects are common parts of many applications in 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 [17, 11], though non-linear models have also been discussed recently [20, 14].

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. The contribution of this paper is the cautious consideration of the following question: given a prediction model for object movements, how much information does a trajectory contain with respect to this model 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 and evaluate the impact of the model parameters on the compression performance.

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.

This paper is an extended version of [15]. As novelties, it contains more details about the code model and its components. Also, it presents new findings on the symbol alphabet and the probability distribution and their impact on the arithmetic coding performance.

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. Both the model and the coder are evaluated in Section 5.

2 Related Work

In the literature, the *compression of movement measurements* is frequently discussed in the context of *Mobile Object Databases (MODs)*. MODs receive trajectory updates from 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, e. g. [8,6]) or that compress data on the fly (*online* approaches, e. g. [17,11]). For these approaches, line simplification and linear dead reckoning—both marking the current state of the art of trajectory compression—have been used. The compression performance of both are upper-bounded by optimal line simplification [12]. The authors of [5,20] approximate trajectories using so-called *minimax* polynomials, so that the maximum approximation error is minimized for the employed parameter values. Further compression techniques that focus on vehicular trajectories use cubic splines [14] and clothoids [19,3]; in general, these non-linear approaches attempt to model the smoothness of vehicular movements or roadways. [13] contains a detailed problem statement and a comparison of the above techniques.

In robot navigation, *Probabilistic Positioning* is often employed for self positioning, e. g., within office buildings [10,22]: instead of precise positions, position probabilities for a discrete map are given. We use a similar concept by defining a probability distribution over a limited region, but do not require any map material.

In [21], navigation decisions of pigeons are analyzed based on the stochastic complexity of trajectory sections by deriving the navigational confidence. The authors of [4] 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 of a movement trace. In this paper, we will show 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, such as the migration patterns of flocks, the movement of astronomical objects or the trajectories of road vehicles can each 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 begin with the necessary information theoretical concepts, for more details see [24, 18]: given a random variable X with a finite sample space $\mathcal{A}_X = \{a_1,\ldots,a_I\}$ and a *probability distribution* $\mathcal{P}_X = \{p_1,\ldots,p_I\}$: $p_i = P(x = a_i), \forall 1 \leq i \leq I : p_i > 0$ and $\sum_{i=1}^{I} p_i = 1$. In the following, we will also refer to $\mathcal{A}_X$ as the (discrete) *alphabet*. The *Shannon information content* (given in *bits*) of an event $x \in \mathcal{A}_X$ 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 necessary bits to represent an outcome $x \in \mathcal{A}_X$.

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 alphabet $\mathcal{A}_X$ and the probability distribution $\mathcal{P}_X$ 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 the random variable 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 means of 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 it is 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 most probable innovations for any valid trajectory need to be mappable on it: if the limits are set too narrow, i. e., $\mathcal{A}_X$ misses reasonable innovations, such innovations could not be covered by the random variable 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 a random variable X over the alphabet $\mathcal{A}_X$ and with a probability distribution $\mathcal{P}_X$ 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.

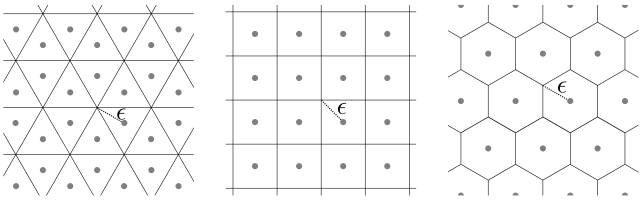
4 Exemplary Implementation of Information Measurement

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.

To this end, we state a number of assumptions, upon which we build our model: (1) We assume that the movement of vehicles are regular and can be expressed by the formulae from kinematics or Newton's laws of motion. (2) We expect that due to this regularity in movement, we can estimate a vehicle's future movement based on past position measurements and limit the area around this estimate containing all reasonable deviations. (3) We assume that, within this area, the positions closer to the estimate are more likely to match the vehicle's next position than those at the border and that the deviations from the estimate are regular as well, so that they can be *learned*.

4.1 Movement Estimator

As trajectory data, we 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



(a) Triangular. (b) Square. (c) Hexagonal.

Fig. 1 Regular tessellations for the discretization grid.

can be described by:

$$\mathbf{v}_i = \frac{p_i - p_{i-1}}{t_i - t_{i-1}} \ , \quad \mathbf{a}_i = \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{t_i - t_{i-1}}$$

With these quantities, some simple movement models can be set up as described in [16]: the first model only considers the last position and the velocity vector:

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

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

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

Obviously, more complex movement models, e. g., using sensor data fusion, are conceivable. However, for the context of the arithmetic coding that we will face later on, this means that all data that is used for the estimation needs to be transmitted to the remote receiver side, which increases the communication load. Therefore, we aim at a minimal data basis for the movement estimator and thus at simple movement models.

Moreover, 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. In this section, we will discuss possible configurations for the discretization grid; to this end, we will discuss several grid node alignments, how to determine reasonable grid dimensions and how to set up the grid frame.

4.2.1 Discretization Grid Node Alignment

It is clear that the specific grid design depends on the application context; in the vehicular domain, for example, a uniform approximation error for any region of the grid is desirable; this makes regularly tessellated grids—i. e., using regular triangles, squares and hexagons as shown in Figure 1—an interesting option. Also, the dimensions and the density of such grid cells can be easily adjusted by a maximum discretization error ϵ that directly influences the edge length of the polygons.

The use of different tessellations has several influences on the model performance: With increasing number of cell edges, both the cell area and the average discretization error increase as well; this leads to a smaller average discretization error of a triangular grid compared to a square or hexagonal tessellation. In turn, the smaller the average discretization error, the better the movement estimation is likely to work. And finally, the number of resulting cells is inversely proportional to the cell size; this means that the alphabet size will decrease with an increasing number of edges per cell, resulting in a smaller entropy of X.

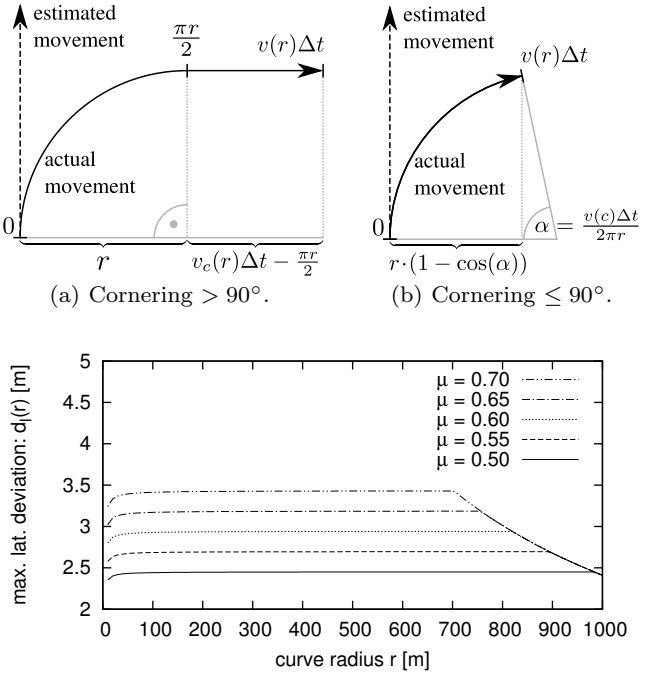
4.2.2 Discretization Grid Dimensions

While setting up the grid cells is a comparably straightforward task, the limitation of the grid scope 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. We therefore introduce in the following a kinematic model to determine these boundaries.

For the determination of the discretization grid dimensions, we 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. Given such a logical center, we can calculate the maximum spatial deviations that can be achieved under our model. We set up this range using a straight-forward line of simple kinematic arguments:

The longitudinal dimension of the grid—i. e., the dimension along the movement direction—directly results from the highest possible deceleration dec_{max} and acceleration acc_{max} that could cause a deviation from the predictand within one measurement interval. Then, according to (2), the longitudinal grid dimension interval relative to the logical grid center is $[-w^-_{lon}; w^+_{lon}]$ with $w^-_{lon} = \frac{|dec_{max}|}{2} \Delta t^2$, $w^+_{lon} = \frac{acc_{max}}{2} \Delta t^2$.

For the lateral grid dimensioning, we need to regard an extreme steering behavior to derive the highest achievable lateral deviation from the estimated position. To this end, we consider the vehicle to pass through a curve, with the vehicle's velocity and the radius of the curve being chosen to such an extent that the lateral deviation is maximized. This deviation is limited, however, by the vehicle's velocity and the ra-



(a) Cornering $> 90°$. (b) Cornering $\leq 90°$.

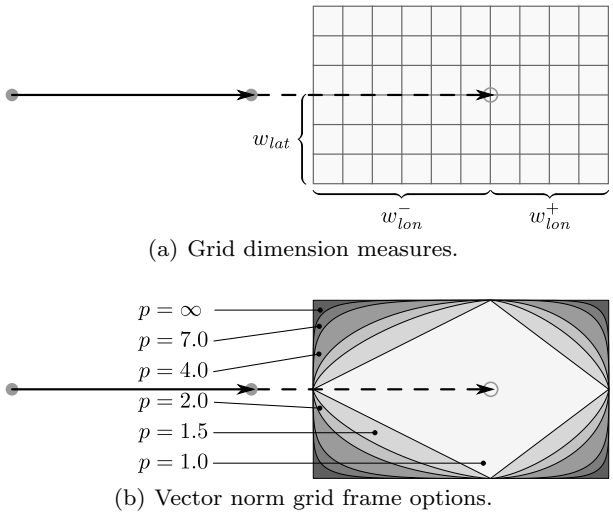
(c) Maximal lateral deviation over curve radii.

Fig. 2 Discretization grid dimension analysis.

dius of the curve: given a curve radius r, a vehicle's speed is upper-bounded by the *critical cornering speed* $v_c(r) = \sqrt{a_l \cdot r}$, where a_l refers to the highest possible lateral acceleration [23]. For the determination of a_l, we state according to Coulomb's friction law: $a_l \leq \mu_s \cdot g$, where μ_s is the static friction coefficient and $g \approx 9.81\,\frac{m}{s^2}$ is the earth's standard gravity acceleration. For the choice of μ_s, default reference values as in [16] can be used. With the critical cornering speed, we can calculate the maximum lateral deviation: at a cornering of more than 90° within the time interval Δt, the lateral deviation equals the sum of the curve radius and the distance that could be driven perpendicular to the assumed driving direction (cf. Figure 2(a)). Otherwise, the lateral deviation is merely the width of the curve that has been passed so far (cf. Figure 2(b)). Formally, the maximum lateral deviation can be expressed by the function $d_l : \mathbb{R}^{>0} \to \mathbb{R}$,

$$d_l(r) = \begin{cases} r + v_c(r)\Delta t - \frac{\pi r}{2} & v_c(r)\Delta t > \frac{\pi r}{2} \\ r \cdot (1 - \cos(\frac{v_c(r)\Delta t}{r})) & \text{else} \end{cases}$$

As depicted in Figure 2(c), the graph of d_l resembles a square root curve: after a rapid growth with an increasing curve radius of up to approximately $80\,m$, the curve stagnates and features merely a minor slope. The heavy drop of all curves results from an upper bound for $v_c(r)$ that we have set to $70\,\frac{m}{s}$. With a friction of $\mu = 0.55$ (tire on tar/asphalt [16]), we can derive from



(a) Grid dimension measures.

(b) Vector norm grid frame options.

Fig. 3 Discretization vector norm grid frame options.

this analytical model that the grid would need to be at least $2 \cdot 2.7\,m = 5.4\,m$ wide. Figure 3(a) shows an exemplary discretization grid with previous position measurements, the position estimate and the measures on the longitudinal and lateral dimension. In the following, we will refer to the lateral grid size as $2 \cdot w_{lat}$.

However, though these grid dimensions are analytically set up, they do not necessarily need to be optimal. Further influences such as increased positioning noise levels may cause innovations to lie outside the analytically deduced boundaries. A simple countermeasure for such situations would be to add a single extra symbol to $\mathcal{A}_X$, representing outliers, which only minimally increases the alphabet size and the entropy of the random variable X. Complementary, expected or current noise statistics, e. g., *dilution of precision (DOP)* values, could be regarded for the grid dimensioning: the grid dimensions could be simply increased by a certain size to set up a *guard zone* around the analytically determined grid dimensions, thus allowing for a higher noise level by increasing the alphabet. The setup of such a guard zone is nontrivial, however, and beyond the scope of this paper and thus remains future work.

4.2.3 Discretization Grid Frame

In Figure 3(a), we assume a rectangular grid shape. While this is concrete and easy to model, it does at the same time not really reflect a true analytic boundary for the measurement deviations from the predictand. For such a boundary, one would have to regard that due to Coulomb's law, vehicles cannot progress as far on the longitudinal dimension when cornering. If this is taken into account, the resulting grid frame would feature an elliptic form. We approximate such an elliptic form with

p-norms; a grid node with the metric offset $(\delta_{lon}, \delta_{lat})$ from the logical grid center (the predictand) is mapped to an alphabet symbol iff

$$({\delta'_{lon}}^{p} + {\delta'_{lat}}^{p})^{\frac{1}{p}} \leq 1$$

with

$$\delta'_{lon} = \begin{cases} \frac{\delta_{lon}}{w^{-}_{lon}} & \text{if } \delta_{lon} < 0 \\ \frac{\delta_{lon}}{w^{+}_{lon}} & \text{else} \end{cases} , \qquad \delta'_{lat} = \frac{\delta_{lat}}{w_{lat}} .$$

Figure 3(b) shows exemplary vector norm grid frames for $p \in \{1.0, 1.5, 2.0, 4.0, 7.0, \infty\}$. The ∞-norm results in a rectangular frame as shown in Figure 3(a).

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

The probability distribution of the random variable X assigns a probability p_i to each symbol $a_i \in \mathcal{A}_X$, as described in Section 3. Choosing the correct probability distribution that fits the actual nature of the innovations is non-trivial and needs to be regarded more closely. In the following sections, we therefore discuss several possible distributions that we will evaluate with our arithmetic coding scheme in Section 5.

4.3.1 Uniform Distribution

The simplest possible distribution is the *uniform* distribution, i. e.,

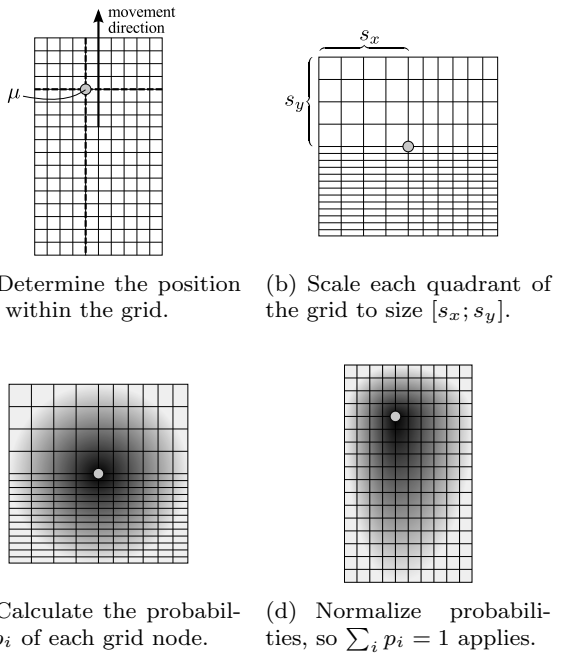
$$\forall 1 \leq i \leq I : p_i = \frac{1}{|\mathcal{A}_X|} .$$

Under the assumption that the employed movement estimator satisfies a reasonable degree of accuracy and therefore small deviations from the position estimate are more likely than large ones, this distribution is not what one would expect from $\mathcal{P}_X$ in reality. However, it is a good lower-bound benchmark that can be used to validate the performance of other probability distributions.

4.3.2 Normal Distribution

Since we expect predominant accurate results from the position estimators, we assume $\mathcal{P}_X$ to be reasonably close to the normal distribution which is commonly used in the context of noisy position measurements [9]. While it is unlikely that the innovations during an estimation process will be perfectly normal distributed, we consider this to be a good approximation.

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 [25]. 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).



(a) Determine the position of μ within the grid.

(b) Scale each quadrant of the grid to size $[s_x; s_y]$.

(c) Calculate the probability p_i of each grid node.

(d) Normalize probabilities, so $\sum_i p_i = 1$ applies.

Fig. 4 Skewing of the probability distribution and mapping it to the grid nodes.

We do not expect the dimensions of the innovation domain to correlate, so $\rho = 0$. However, the normal distribution is symmetric, which does not necessarily need to apply to the grid as well. Therefore, we need to find a mapping (skewing) of the probability distribution to the dimensions of the grid. To this end, we propose a projection of the grid, which is schematically depicted in Figure 4: the mean μ separates the grid into four quadrants (cf. Figure 4(a)), each of which is scaled to the dimensions $[s_x; s_y]$, where s_k is the number of standard deviations that are supposed to cover the k axis of each quadrant (cf. Figure 4(b)). 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

$$f(x, y) = \frac{1}{2\pi} \exp(-\frac{1}{2}\left(x^2 + y^2\right))$$

(cf. Figure 4(c)). Afterwards, the assigned probabilities need to be normalized to eliminate scaling effects, so that $\sum_{i=1}^{I} p_i = 1$ applies again (cf. Figure 4(d)).

Alternatively, asymmetric probability (e. g., log-normal) distributions could also be employed.

4.3.3 Trained Distributions

Under the assumption that vehicular movements, disregarding the terrain or surrounding, follow a general principle or pattern, we also suppose that there is a generally fitting distribution that can be found, or *learned*. Since the alphabet is discrete and finite, we attempt to obtain such a distribution from a collection of previously observed traces, referred to as the *training set*.

To this end, we map each measurement in the training set to its corresponding alphabet symbol using the non-accelerated movement estimator (1) and the grid as described in Section 4.2. For each symbol $a_i \in \mathcal{A}_X$, we can then determine the occurrence count n_i. The probability p_i can thus be written as

$$p_i = \frac{n_i}{\sum_{i=1}^{I} n_i} \ .$$

According to the definition in Section 3.1, all symbol probabilities need to be nonzero. We therefore define $n'_i := max\{n_i, 1\}$ and p'_i analogously to p_i, resulting in a close approximation of p_i that can be used for the determination of the information content and for the arithmetic coding of trajectories. If each symbol occurs at least once, $n'_i = n_i$ and thus $p'_i = p_i$.

4.3.4 Adaptive Distributions

In contrast predefined distributions, we also regard distributions that evolve over time. Such *adaptive distribution* models start with an initial setup, e. g., a uniform distribution, and evaluate the observed symbol occurrences to converge towards the actual distribution. Of course, more realistic distributions can also be used as initial setups. Common adaptive distribution models regard n-gram relations in the symbol stream, e. g., simply constructing a unigram probability distribution.

The advantage of this approach is that the resulting distribution is an optimal match for all previously, and at best upcoming symbol occurrences. Of course, this will only pay off for sufficiently long trajectories, i. e., if the ratio of trajectory length to the alphabet size satisfies a particular threshold value. Otherwise, the learned distribution reaches a representative version too late so that too few position measurements in the trajectory can benefit from the learned distribution to compensate for the learning phase, during which the distribution may be far from being an acceptable fit.

4.3.5 Contextual Distributions

Finally, we regard probability models that feature more than a single probability distribution and which we refer to as *contextual distributions*. For these, we assume that the actual distribution of measurement innovations correlate with the measurement vehicles' current movement parameters, such as velocity or acceleration. Thus, for each known context, there is one probability distribution that can be selected for use. In doing so, distinctive situations can be taken into account for the information content determination and for the arithmetic coding, such as halts, accelerations after halts or movements at constant velocities.

It is obvious that contextual distributions are actually no self-contained probability distributions but rather a way to combine multiple distributions into a single probability model and thus can be used as an extension instead of a stand-alone and exclusive alternative. Also, there can be different kinds of distributions employed for each context in order to assemble the optimal fit for each context.

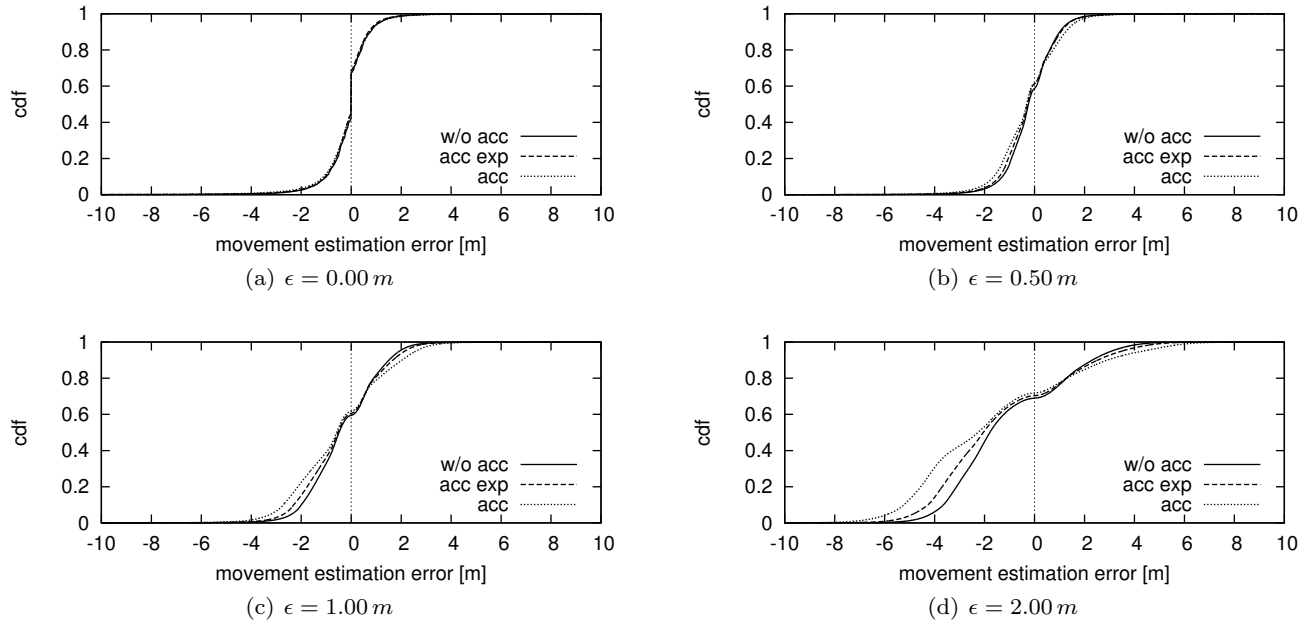
Table 1 Exemplary alphabet configurations and entropies.

			entropy [bits]		
Δt	ϵ	$\lvert\mathcal{A}_X\rvert$	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

4.4 Exemplary Alphabets and their Entropies

We can now determine the entropy of the random variable X, i. e., the expected information content per position measurement for a given alphabet and probability distribution. Table 1 shows exemplary entropies for varying measurement intervals and accuracy bounds. For the regular square grid setup we assumed an acceleration interval $[\mathrm{dec}_{\max}; \mathrm{acc}_{\max}] = [-11; 8]\,\frac{m}{s^2}$. 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 a predictand, solely depends on the used movement model θ, the alphabet $\mathcal{A}_X$ and the probability distribution $\mathcal{P}_X$ of the random variable 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 off-the-shelf GPS receivers provide position measurements as fixed-point



(a) $\epsilon = 0.00\,m$ (b) $\epsilon = 0.50\,m$

(c) $\epsilon = 1.00\,m$ (d) $\epsilon = 2.00\,m$

Fig. 5 Movement estimation error analysis for difference movement models (urban movement).

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 apply if the probability distribution of the alphabet is considered uniform. 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 [7], version 1.1 as a basis.

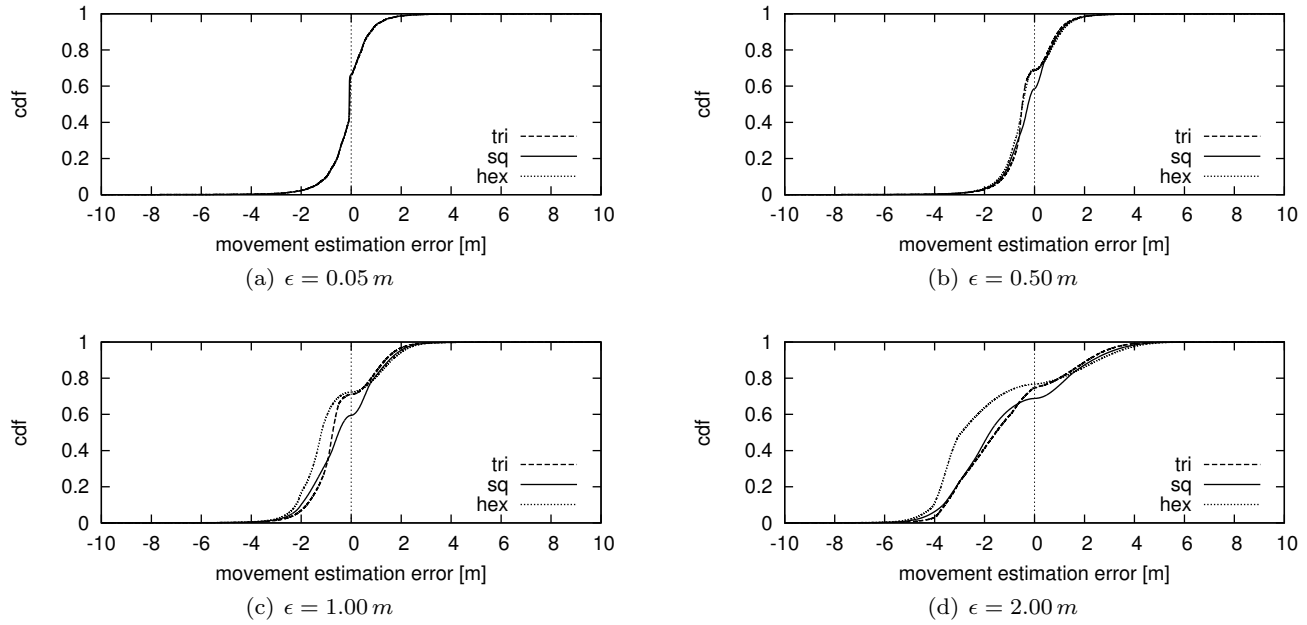
The only modification to our formal model lies in the handling of outlier measurements: Instead of adding an extra symbol to $\mathcal{A}_X$, the encoding stops upon an outlier measurement. This is inevitable, because the mapping of a discretized innovation to a symbol needs to be bijective; this is not fulfilled in case of outliers. Once an innovation cannot be mapped to a grid node, it is not possible to retrieve a valid grid node in the decoding process. Therefore, in this case the symbol stream is terminated with the *End Of Stream* symbol, $\mathcal{P}_X$ 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 and Methodology

We evaluate the presented arithmetic coding model on the basis of an extensive real-world movement data set from the *OpenStreetMap* project [2]. These data are available under the Creative Commons license BY-SA 2.0 [1]. To isolate the effects caused by road topologies, 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; this is a very common position measurement frequency that makes the selected database representative for a huge number of both off-the-shelf and high-accuracy positioning devices. We furthermore excluded traces with less than 100 measurement points to neglect side effects due to very short movement periods. In the vehicular domain, such trajectory lengths result mostly from positioning signal loss and thus from erroneous situations which we do not want to regard in this study. Finally, in doing so, we retrieved 2263 and 4946 traces for the urban and highway pattern, respectively.

For the evaluation of the trained probability distributions, we also split our trace collection into a *training*



(a) $\epsilon = 0.05\,m$ (b) $\epsilon = 0.50\,m$

(c) $\epsilon = 1.00\,m$ (d) $\epsilon = 2.00\,m$

Fig. 6 Movement estimation error analysis for non-acc. movements and different grid node alignments (urban movement).

set and two *test sets*. Since traces are classified as highway or urban based on the maximum velocity, highway traces still contain a sufficient amount of urban traffic and are thus a good choice for obtaining a trained distribution. Hence, we randomly selected 2000 highway traces to train a distribution and used the other 2946 highway traces and all urban traces as two test sets.

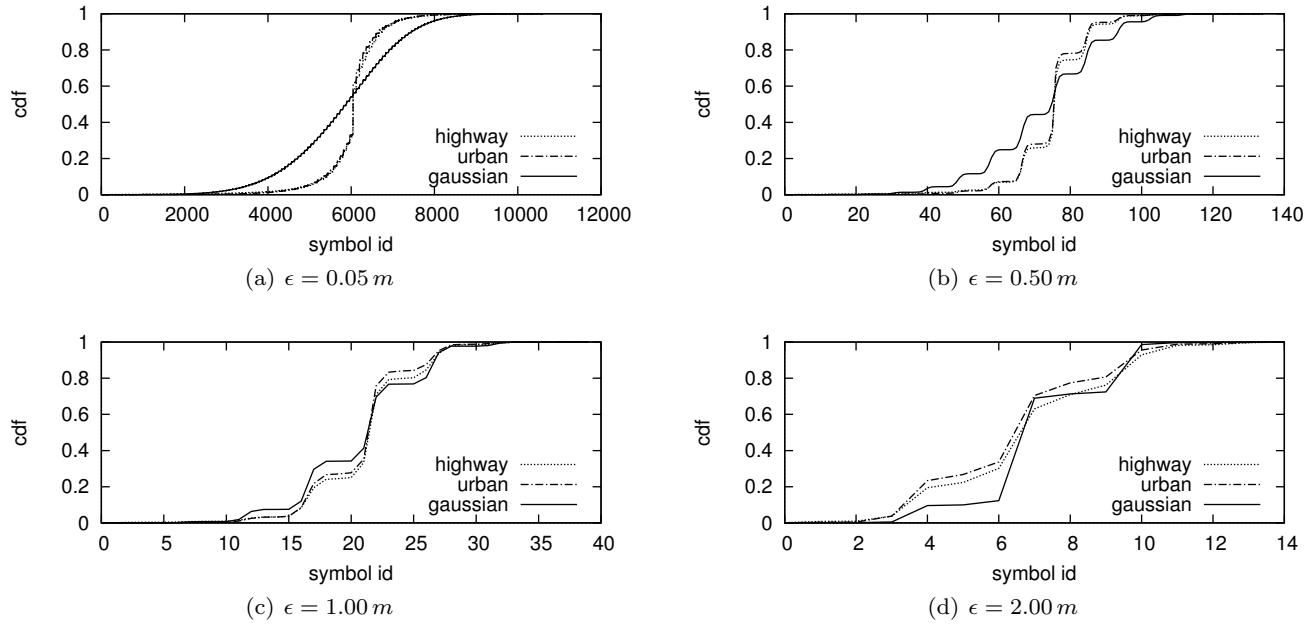
For the evaluation, we compressed every collected trace for a variety of approximation error bounds (ϵ) up to $2\,m$ and analyzed important effects, i. e., the accuracy of the movement estimator, the goodness of fit for the normal probability distribution and the compression performance. For the last-mentioned characteristic, we furthermore have a closer look on the influences of the grid node alignments, the grid frame shape, and the probability distributions with respect to a basic coding scheme configuration.

5.2 Movement Estimation and Discretization

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 5. 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 a coarser discretization lowers 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*), which is more impaired by the discretization, because it derives acceleration values from distorted velocities. To reduce this effect, we amended the accelerated estimator by smoothing the computed acceleration values exponentially (*acc exp*). This improves the situation, but does not provide the same robustness as the non-accelerated estimator (cf. Figure 5). In all of our tests, this directly resulted in lower compression ratios for the respective movement models. We therefore only regard the non-accelerated model in the remainder of our evaluation.

The discretization grid node alignment also influences the performance of the movement estimator as discussed in Section 4.2.1. Figure 6 shows the movement estimation error analysis for the non-accelerated movement estimator and the three discussed tessellation schemes. Though the estimation results are very close to each other, the estimation works slightly best with the triangular tessellation and worst with the hexagonal tessellation. However, with these two tessellation schemes the estimator exhibits more movement underestimations of up to one meter on the longitudinal axis. The reasons for this effect are not completely understood and are subject to further investigation.



(a) $\epsilon = 0.05\,m$ (b) $\epsilon = 0.50\,m$

(c) $\epsilon = 1.00\,m$ (d) $\epsilon = 2.00\,m$

Fig. 7 Cumulative distribution analysis of the probability distribution $\mathcal{P}_X$.

5.3 Gaussian Probability Distribution

To evaluate the goodness of fit for the assumed Gaussian probability distribution, we compare it to the actual *cumulative distribution functions (cdf)* for both topologies over the respective $\mathcal{A}_X$ in Figure 7. 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" 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 confirms our assumption that the Gaussian distribution is reasonable approximation for $\mathcal{P}_X$, though it is also quite obvious that there is room for improvements. We will evaluate the impact of the other distribution models in Section 5.4.4.

5.4 Compression

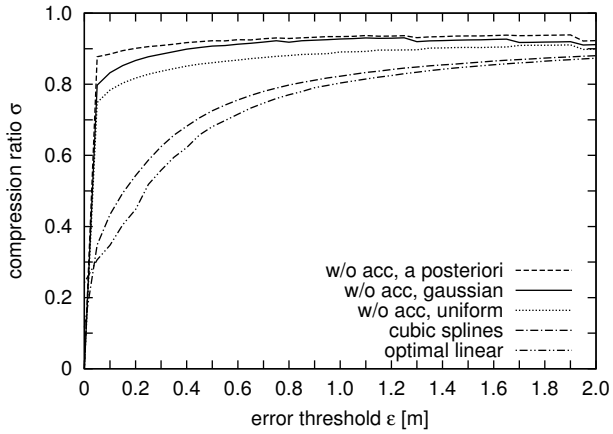
5.4.1 Basic Configuration and Benchmarks

At first, we want to evaluate the compression performance of our proposed arithmetic coding scheme in a basic configuration: the vehicle movement is estimated using the non-accelerated movement model, and we use a rectangular shaped discretization grid with a grid node alignment following a square grid cell tessellation. Additionally, we use the current state-of-the-art compression algorithms for spatio-temporal data as benchmark algorithms. As discussed in [13], these cover the compression schemes based on optimal line simplification and cubic spline interpolation.

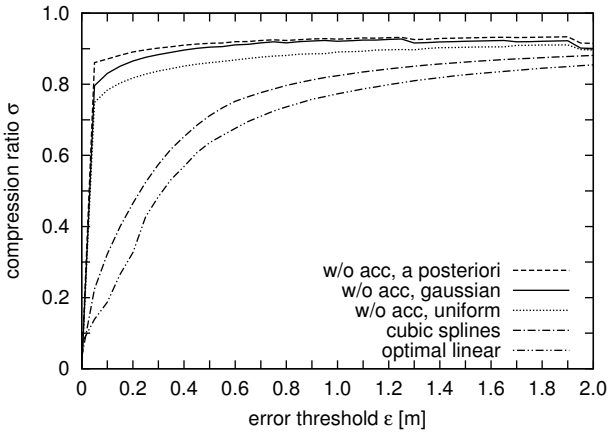
The spline-based approach runs in $O(n^3)$ and was designed for relatively short trajectories of ∼250 elements; so, we selected typical trajectories of 1000-1300 positions and cut them to slices of 250 elements, each slice overlapping the previous one by 100 elements, and thus gained 318 and 327 shorter 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.

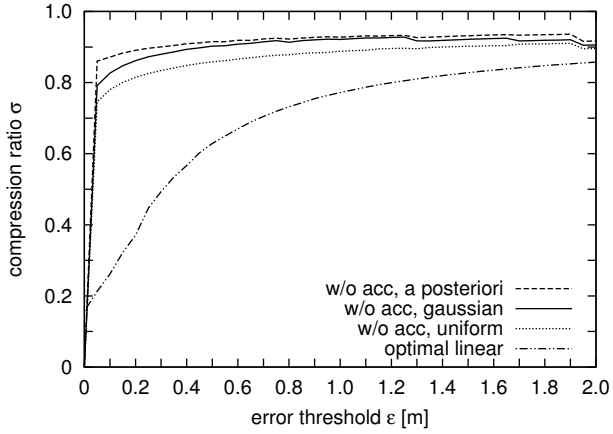
As an optimal probability distribution configuration for the arithmetic coder, we performed compressions using *a posteriori* knowledge: we measured the empirical distributions of the code symbols for each trajectory and used these as stochastic models for the arithmetic coder. This is only a theoretical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



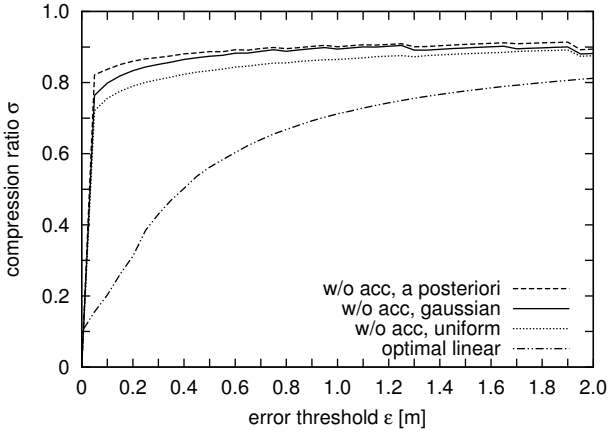
(a) Trajectories w/ 250 elms (urban).



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



(c) Uncut trajectories (urban).



(d) Uncut trajectories (highway).

Fig. 8 Basic compression ratio comparison.

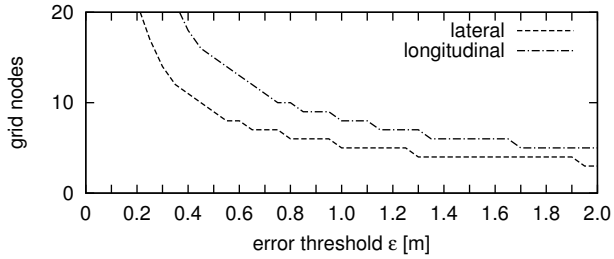
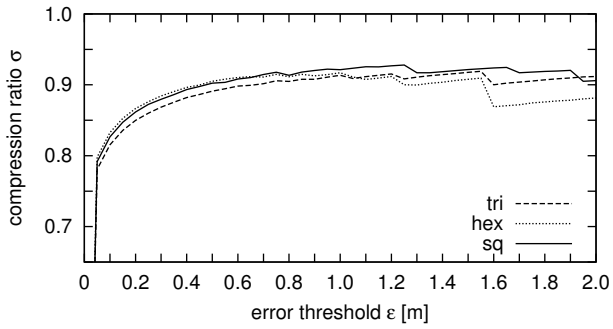


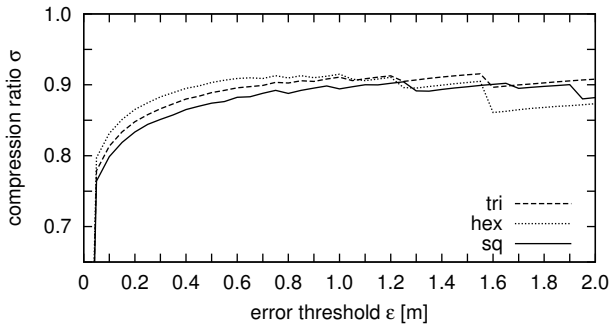
Fig. 9 Grid node counts over increasing error bound.

Figures 8(a) – 8(d) depict the average compression ratios against the error bound ϵ. For all configurations, the same ranking of compression techniques is visible: the optimal line simplification—representing the upper bound for the performance of current state-of-the-art movement compression—performs worst, being outperformed by the cubic spline approach (only for the trajectories with 250 elements in Figures 8(a) and 8(b)). The arithmetic coding 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. Because the basic arithmetic coding configuration outperforms the current state-of-the-art compression schemes, we will use it as benchmark for all enhanced code model configurations.

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. This can be seen in Figure 9 that depicts the number of discretization grid cells per dimension over increasing accuracy threshold values.

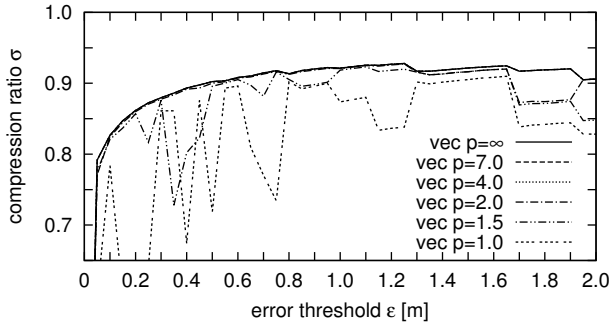


(a) Uncut trajectories (urban).

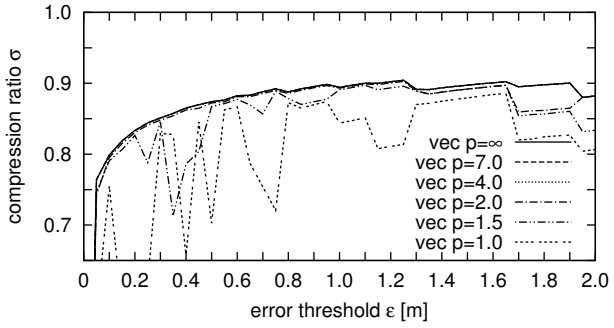


(b) Uncut trajectories (highway).

Fig. 10 Compression ratio comparison for different grid node alignments.



(a) Uncut trajectories (urban).



(b) Uncut trajectories (highway).

Fig. 11 Compression ratio comparison for different grid frames.

5.4.2 Impact of Discretization Grid Nodes Alignments

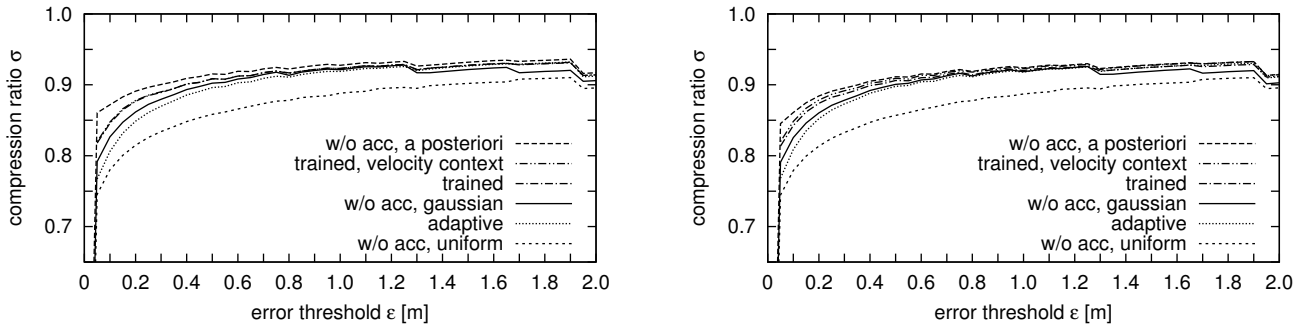
Figure 10 depicts the urban and highway compression performances for discretization grid node alignments with triangular, square and hexagonal grid cell tessellation. For the urban topologies, the hexagonal grid node alignment performs slightly best for low ϵ, while for $\epsilon > 1.0\,m$, its performance drops to the lowest of the three regarded tessellation models. The triangular grid node alignment achieves slightly worse results than the square tessellation, but in the end, it exceeds the compression ratio of the square tessellation. This can be explained with the results presented in Section 5.2: the triangular tessellation has a positive impact on the movement estimation, but on the other hand, it increases the alphabet size, causing a slightly worse compression performance. Also, the bad influence of the hexagonal tessellation on the movement estimator cannot be alleviated by the smaller alphabet.

For the highway scenario, the situation is different: due to a better movement predictability and the smallest alphabet, the coding with the hexagon grid performs significantly better than with the square grid. For the triangular tessellation, the higher grid node resolution causes also a better compression performance compared to the basic configuration benchmark.

However, based on this strongly different impact of the discretization node alignments, one cannot simply state a generally optimal choice of the discretization node alignments. However, we prefer the square grid tessellation for practical implementations due to its algorithmic simplicity and its good performance.

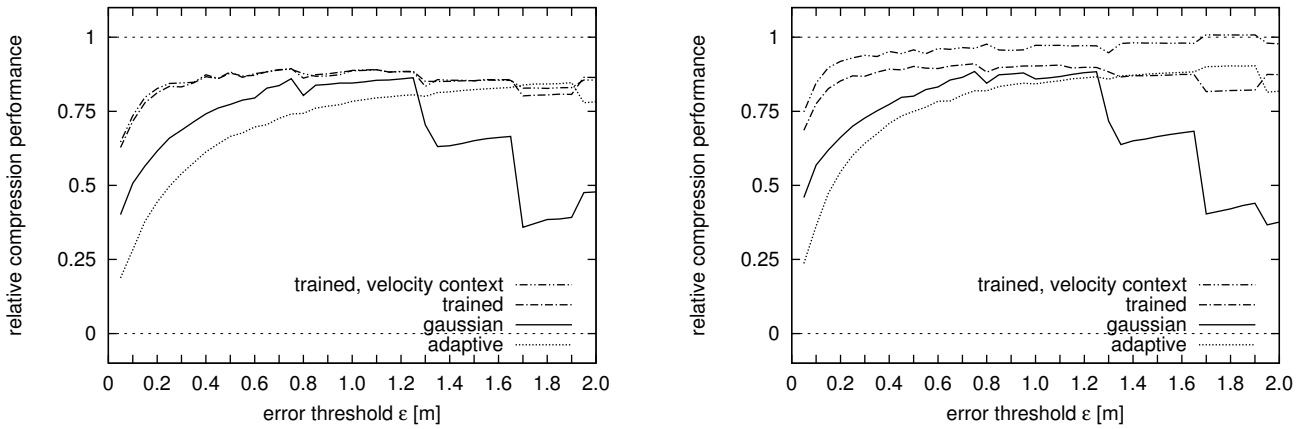
5.4.3 Impact of Discretization Grid Frame Shapes

Figure 11 depicts the compression ratios for different grid vector frame parameters. It can clearly be seen that the compression performance improves with increasing p, i. e., with a grid frame more and more similar to the rectangular shape of the basic configuration. Since a larger p causes a larger alphabet, this result is a good indicator that a more tolerant choice of the grid frame (and thus the alphabet) size is rather beneficial than disadvantageous. This is an interesting result, because it shows that a higher tolerance for outlier measurements is more important to the overall compression performance than an exact determination and dimensioning of the code symbol alphabet.



(a) Uncut trajectories (urban). (b) Uncut trajectories (highway).

Fig. 12 Compression ratio comparison for enhanced probability distribution models.



(a) Uncut trajectories (urban). (b) Uncut trajectories (highway).

Fig. 13 Compression performance comparison for enhanced probability distribution models, relative to the Gaussian arithmetic compression scheme within lower and upper compression bounds.

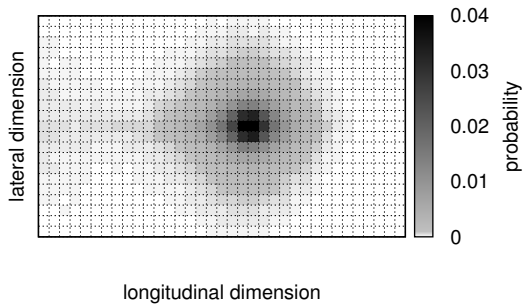
5.4.4 Impact of Probability Distributions

In this section we will analyze the compression performance of the trained, adaptive, and contextual distributions described in Section 4.3. As mentioned above, we created the symbol distribution based on a *training set* of 2000 highway traces and used the trained distributions for the arithmetic coding for all traces in the two *test sets*. For the contextual distributions, we used 10 individually trained distributions for different velocity ranges. We assigned each measurement in the training set to the distribution D_i with

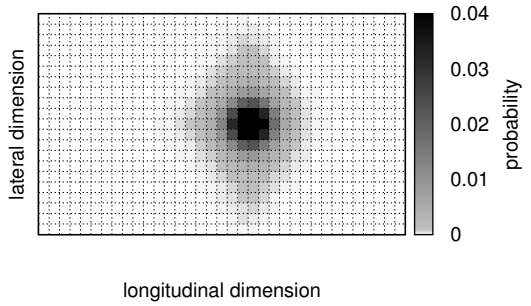
$$i = \min\left(\left\lfloor \frac{|\mathbf{v}|}{3\frac{\mathrm{m}}{\mathrm{s}}} \right\rfloor + 1,\ 10\right)$$

where $\mathbf{v}$ is the previously observed velocity. We then used the same formula to obtain the correct distribution during the arithmetic coding of the test data.

Figure 12 shows the compression results of the advanced distributions compared to the basic configuration. Though on the first sight, the compression results seem to be quite close to each other, it is worth to focus on the relative compression performance with respect to the interval between the upper and lower compression bound, given by the a posteriori and uniform distribution results; this is depicted in Figure 13: in this figure, a zero value means that a compression equals the achievement with a uniform distribution and a value of one means that a compression ratio as good as with a posteriori knowledge could be achieved. In general, the relative performance plots are very similar for the urban and highway traces; at first, it is clear to see that the trained distributions achieve the best compression results. For the highway traces, however, the trained distributions regarding velocity classes even exceed what we referred to as the upper performance limit. This is possible, because of the multitude of distributions that are selected depending on the current movement class and that are optimized views on the symbol distributions. Of course, if an a posteriori dis-



(a) Distribution for velocity class 2 (3-6 m/s).



(b) Distribution for velocity class 9 (24-27 m/s).

Fig. 14 Comparison of probability distributions at low and high speeds for $\epsilon = 0.2$.

tribution with velocity classes would have been used, this could not have been exceeded. The arithmetic coding using an adaptive distribution performs worst for $\epsilon < 1.3\,m$. The main reason for this is that the number of grid nodes for small ϵ is very large and thus it needs many samples to actually learn the actual symbol distribution; however, the trajectories are too short most of the time, so these necessary samples can not be collected timely. Of course, this situation improves for smaller alphabets, the probability distributions of which are easier to be learned. The relative compression performance again underlines that the Gaussian distribution is a reasonably good fit for $\mathcal{P}_X$; however, these plots also emphasize the heavy drop of compression performance for $\epsilon \geq 1.3\,m$ due to the coarseness of the discretization grid. The trained and adaptive distributions are naturally only marginally affected by this effect.

For highway topologies, the compression ratios are slightly better. This is most likely due to the more limited steering behavior at higher velocities, as switching between lanes becomes more prevalent. This is in fact shown by symbol distributions in Figure 14: longitudinal variations are lower than with smaller velocities, while the lateral variations nearly remain the same.

6 Conclusion

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 an arithmetic coding/compression scheme. 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 superior to the best existing compression scheme for vehicular trajectories.

Two open aspects from [15] are addressed in this paper: first, we analyzed both the impact of different discretization grid parameters, namely the grid node alignment and the shape of the grid frame. We found out that for the grid node alignment, no clear statement can be made regarding a benefit for the compression performance, but that more realistic and elliptic grid frames do not pay off but that a frame should increase fault tolerance instead. Second, we have investigated trained and adaptive symbol probability distributions. It showed that trained models, especially those providing distributions for particular velocity classes, improve the arithmetic coding performance significantly for all regarded topologies.

Acknowledgments

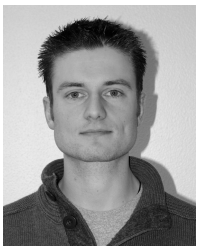
The authors wish to thank Dennis Dobler for his work on the prototype of the arithmetic coder and the formal model. Also, the authors thank Bob Carpenter for the source code of his arithmetic coder and his support.

References

1. Creative Commons BY-SA 2.0. http://creativecommons.org/licenses/by-sa/2.0/
2. The OpenStreetMap Project. online resource. http://www.openstreetmap.org/
3. Baran, I., Lehtinen, J., Popovic, J.: Sketching Clothoid Splines Using Shortest Paths. Computer Graphics Forum pp. 655–664 (2010)
4. Bhattacharya, A., Das, S.K.: 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 (1999)
5. Cai, Y., Ng, R.: Indexing spatio-temporal trajectories with Chebyshev polynomials. In: SIGMOD '04: Proceedings of the ACM SIGMOD International Conference on Management of Data (2004)
6. Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal data reduction with deterministic error bounds. VLDB Journal **15**(3), 211–228 (2006)

7. Capenter, B.: Arithcode project: Compression via arithmetic coding in java. version 1.1. online resource (2002). http://www.colloquial.com/ArithmeticCoding/
8. Civilis, A., Jensen, C.S., Pakalnis, S.: Techniques for efficient road-network-based tracking of moving objects. IEEE Transactions on Knowledge and Data Engineering **17**(5), 698–712 (2005)
9. van Diggelen, F.: GPS Accuracy: Lies, Damn Lies and Statistics. GPS World **9**(1), 41–45 (1998)
10. Fox, D.: Markov localization: A probabilistic framework for mobile robot localization and navigation. Ph.D. thesis, University of Bonn, Germany (1998)
11. Hönle, N., Großmann, M., Reimann, S., Mitschang, B.: 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 (2010)
12. Imai, H., Iri, M.: Computational-geometric methods for polygonal approximations of a curve. Computer Vision, Graphics, and Image Processing **36**(1), 31–41 (1986)
13. Koegel, M., Baselt, D., Mauve, M., Scheuermann, B.: A Comparison of Vehicular Trajectory Encoding Techniques. In: MedHocNet '11: Proceedings of the 10th Annual Mediterranean Ad Hoc Networking Workshop (2011)
14. Koegel, M., Kiess, W., Kerper, M., Mauve, M.: Compact Vehicular Trajectory Encoding. In: VTC '11-Spring: Proceedings of the 73rd IEEE Vehicular Technology Conference (2011)
15. Koegel, M., Mauve, M.: On the Spatio-Temporal Information Content and Arithmetic Coding of Discrete Trajectories. In: MobiQuitous '11: Proceedings of the 8th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking & Services (2011)
16. Kuchling, H.: Taschenbuch der Physik, 17 edn. Fachbuchverlag Leipzig im Carl Hanser Verlag (2007). In German language.
17. Lange, R., Farrell, T., Dürr, F., Rothermel, K.: Remote Real-Time Trajectory Simplification. In: PerCom '09: Proceedings of the 7th IEEE International Conference on Pervasive Computing and Communications, pp. 184–193 (2009)
18. MacKay, D.J.C.: Information Theory, Inference & Learning Algorithms. Cambridge University Press, New York, NY, USA (2002)
19. McCrae, J., Singh, K.: Sketching piecewise clothoid curves. Computers & Graphics **33**(4), 452–461 (2009)
20. Ni, J., Ravishankar, C.V.: Indexing Spatio-Temporal Trajectories with Efficient Polynomial Approximations. IEEE Transactions on Knowledge and Data Engineering **19**, 663–678 (2007)
21. Roberts, S., Guilford, T., Rezek, I., Biro, D.: Positional entropy during pigeon homing i: application of bayesian latent state modelling. Journal of Theoretical Biology **227**(1), 39–50 (2004)
22. Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal navigation – mobile robot navigation with uncertainty in dynamic environments. In: ICRA '99: Proceedings of the IEEE Int'l Conference on Robotics and Automation, pp. 35–40 (1999)
23. Schimmelpfennig, K.H., Hebing, N.: Geschwindigkeiten bei kreisförmiger Kurvenfahrt – Stabilitäts- und Sicherheitsgrenze. Der Verkehrsunfall **20**(5), 97–99 (1982). In German language.
24. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal **27**, 379–423 (1948)
25. Timm, N.: Applied multivariate analysis. Texts in statistics. Springer (2002)

About the Authors



Markus Koegel received the M. Sc. degree in Computer Science from the Heinrich Heine University, Düsseldorf, Germany, in 2008. After an internship at Mercedes-Benz RDNA in 2008, he became a Ph. D. student at the Heinrich Heine University. His current research interests include mobile and vehicular ad-hoc networks with a focus on data processing and compression, and information theory.

Matthias Radig received the B. Sc. degree in Computer Science from the Heinrich Heine University, Düsseldorf, Germany, in 2011. Currently, he is a M. Sc. student in Computer Science at the Heinrich Heine University. For his Master's studies, he is working on the adaptive and contextual probability distributions presented in this paper.

Erzen Hyko received the B. Sc. degree in Computer Science from the Heinrich Heine University, Düsseldorf, Germany, in 2012. The topic of his Bachelor's thesis was on the implementation of enhanced grid parameters presented in this paper. He was granted a scholarship from the Ministry of Innovation, Science, Research and Technology of the State of North Rhine Westphalia and currently, he is a M. Sc. student in Computer Science at the Heinrich Heine University.

Martin Mauve received the M. S. and Ph. D. degrees in Computer Science from the University of Mannheim, Germany, in 1997 and 2000 respectively. From 2000 to 2003, he was an Assistant Professor at the University of Mannheim. In 2003, he joined the Heinrich Heine University, Düsseldorf, Germany, as a Full Professor and Head of the research group for computer networks and communication systems. His research interests include distributed multimedia systems, multimedia transport protocols, mobile ad-hoc networks and inter-vehicle communication.