Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Informatik I
Effiziente Algorithmen und wissensbasierte Systeme

Bachelor Thesis

An algorithm for map matching on incomplete road databases

Benedikt Budig

submitted on February 24, 2012

supervisors:
Prof. Dr. Alexander Wolff
Dr. Jan-Henrik Haunert

Zusammenfassung

Der Begriff *map matching* bezieht sich auf das Problem, zu einer gegebenen Folge von Positionspunkten (z.B. GPS-Punkten) auf einer Straßenkarte den zurückgelegten Weg zu rekonstruieren. In dieser Bachelorarbeit wird ein innovativer, selbst entwickelter Algorithmus vorgestellt, der *map matching* auch auf unvollständigem Kartenmaterial und für Bewegungen abseits der Straßen durchführen kann.

Zunächst werden in einem Überblick verwandte Arbeiten zum Thema *map matching* im Allgemeinen und für unvollständige Karten im Besonderen präsentiert. Anschließend wird das von Lou et al. im Jahr 2009 präsentierte System erläutert, auf dem der in dieser Arbeit vorgestellte Algorithmus basiert. Zu diesem System werden außerdem verschiedene Verbesserungsvorschläge eingeführt und getestet.

Auf dem verbesserten System setzen die Erweiterungen für die Verarbeitung von GPS-Trajektorien auf unvollständigem Kartenmaterial auf. Sie erlauben es, Bewegungen abseits des Kartenmaterials zuverlässig zu erkennen und korrekt darzustellen, was dem Basissystem gänzlich unmöglich ist.

Im Weiteren wird das neu eingeführte System in verschiedenen Versuchen auf die Qualität seiner Resultate getestet. Dafür werden unter anderem über tausend zufällig veränderte Straßenkarten herangezogen. Die Ergebnisse zeigen, dass der in dieser Arbeit eingeführte Algorithmus auch unter suboptimalen Bedingungen hervorragende Ergebnisse erzielt.

Nach einer formalen Laufzeitanalyse werden abschließend die neuen Erkenntnisse zusammengefasst und noch offene Probleme sowie zukünftige Erweiterungsmöglichkeiten benannt.

Contents

1 Introduction

In the last years, a proliferation of GPS-embedded devices has taken place: automotive navigation systems are widely available both as original factory equipment and as portable handheld devices, and GPS-enabled smartphones seem to be omnipresent. This leads on the one hand to an increasing demand of good corresponding software and on the other hand emerges a vast source for collected GPS data ready for examination.

Various applications, for example the analysis of traffic streams in large cities as described by Pang et al. (2011), take advantage of the latter. The features of navigation software, both installed in vehicles and running on smartphones or other handhelds, are already widely known. What is common with almost all the GPS-driven navigation and analysis approaches is the reliance on an accurate map matching algorithm.

The term map matching describes the problem of recognizing the roads a GPS receiver was carried along. The only information available for this are the GPS logs that have been recorded on the way and a digital map. In this thesis, we want to create a system that can deal with off-road movements as well as with travelling on roads.

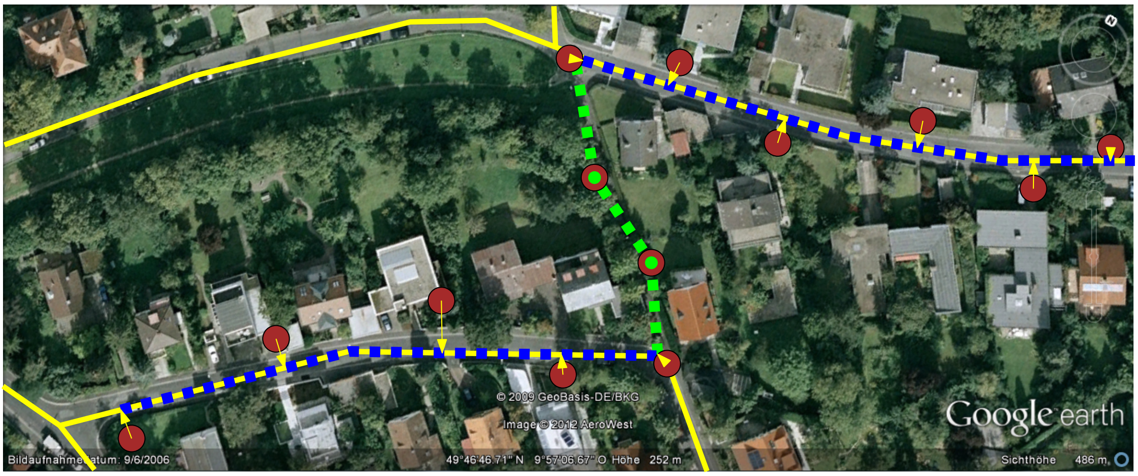

Figure 1.1: Map matching on incomplete road database.
Imagery ©2012 AeroWest, GeoBasis-DE/BKG, Google

In Figure 1.1, we can see an example of this. The brown points have been logged by a GPS receiver that was carried along by a pedestrian. For each of the points, our system determines the nearest road (yellow arrows) and recognizes it as taken by the walker (blue dotted path). However, the two brown/green GPS points can not be associated, because the small footpath taken is not covered by our map (yellow). In this case, our algorithm assumes them to be off-road points and connects them (green dotted path) with each other and with the on-road path (blue dotted).

Our system, as introduced in this thesis, is an innovative contribution for map matching on incomplete road databases. It provides advantages as well to navigation systems for bicyclists and walkers as to post-processing applications. One example for this is a collaborative map construction with a large quantity of involved users as mentioned by Pereira et al. (2009). In addition, our system can cope well with networks where neuralgic roads are missing (e.g. newly built bridges), producing still excellent matching results. This is beneficial for car navigation, too.

In the following, we first present some related work (Chapter 2), then revisit a map matching algorithm introduced by Lou et al. (2009) (Chapter 4) and propose improvements for it (Chapter 5). Afterwards, we introduce our off-road extensions to the system (Chapter 6) and evaluate it (Chapter 7). Last but not least, we draw a conclusion and point out future work (Chapter 8).

2 Related work

As demonstrated in the introduction, the map matching problem is of fundamental importance to several applications processing raw GPS data. Thus, it was the object of research in various papers in the past two decades. In this chapter, we will have an overview of the recent research papers. An excellent summary of prior related work (and its limitations) is available from Quddus et al. (2007).

2.1 Classification of map matching approaches

Many authors concerned with map matching differentiate between *on-line* and *off-line matching*.

The first term describes the problem of immediately finding a fitting map point for the latest captured GPS point. Consequently, only GPS points that have been collected before can be taken into account when searching for a match. Applied successively to each obtained GPS point, this approach is an *incremental* method for map matching. Algorithms following this principle have been introduced by Wenk et al. (2006), Chawathe (2007) and many others. Possible applications for on-line matching algorithms are navigation systems for cars and other vehicles. The current position of the GPS receiver has to be displayed on the internal map as fast as possible, and future GPS points are thereby naturally not available. Nevertheless, only a single GPS trajectory has to be analysed and the systems do not have to perform faster than real-time, which allows higher running times for on-line algorithms in comparison to the following class.

In contrast, with *off-line matching* it is possible to make use of the whole GPS trajectory when computing the matched path. The final result is obtained directly from one computation with the whole data available. Several recent papers by Lou et al. (2009), Marchal et al. (2005), Pereira et al. (2009) and a number of other authors focus on this topic applying various strategies. Such algorithms are often deployed in environments where a large quantity of trajectories is analysed, which is the reason why some authors call for fast running times. The results of their computation are of vital importance to traffic analysis and road planning tasks. For example, Lou et al. (2009) employed their algorithm to analyse the routes taken by taxis in Beijing.

Additionally, *statistical* methods using the Viterbi algorithm and Hidden Markov Models are available, like the one proposed by Newson and Krumm (2009). As Lou et al. (2009) state, such models are especially apt to deal with GPS measurement errors and can be combined advantageously with other approaches. However, it is difficult to modify these methods for operation on incomplete road databases.

Apart from these classes, the multitude of proposed systems differs in various internal concepts. Some approaches rely on special distance measures to determine the discrepancy between GPS trajectories and possible paths on the road network, for example, the work of Chen et al. (2011) makes use of the Fréchet distance. Lou et al. (2009) proposed a strategy using spatio-temporal analysis, which was improved regarding running times by Eisner et al. (2011). Their concept identifies a set of possible match points for every GPS log. Two consecutive candidate points are then connected with a shortest path. Finally, the most suitable ones will be chosen. Thereby, operation on low sampling rates is possible, too.

The recent approach of Dalumpines and Scott (2011) follows its own way, searching for a shortest path in an area engirding the GPS trajectory. Unlike Lou et al. (2009), they assume that start and end point of the complete trajectory are connected by a shortest path, not only two consecutive candidate points. However, this procedure disregards lots of given information, as the computation of the matched path is rather indistinct and relies heavily on high sampling rates and straightforward movement on the trajectory.

2.2 Algorithms for incomplete road databases

The topic of map matching on incomplete road databases, which demands algorithms that support parts of the trajectory not being matched to the road network, is given considerably less attention in the literature available.

Nevertheless, a small selection of research papers is at hand, featuring algorithms for incomplete networks based on some of the different map matching strategies described in Section 2.1.

Pyo et al. (2001) introduced a probabilistic approach, which uses a *multiple hypothesis technique*-based algorithm for map matching and is able to recognize off-road movements in probabilistic terms. When the road network is left, the system returns the logged GPS coordinates as matching output. However, the system is designed for on-line map matching and makes use of Dead Reckoning features as well as path prediction to improve matching quality in case of a blocked GPS signal. This is not suitable for the off-line matching system for incomplete road databases our thesis focuses on.

Pereira et al. (2009) discussed the same topic, proposing a by far more complex system based on the combination of two different algorithms: the global map matching algorithm by Marchal et al. (2005), which iteratively rates the quality of a set of candidate paths, and a newly introduced genetic algorithm called *GEMMA*. Due to excessive time consumption, *GEMMA* is only used to analyse areas where Marchal's algorithm produces inadequate results.

Although the approach by Pereira et al. (2009) supports incomplete road databases and is designed as a post-processing system, it has drawbacks that justify the introduction of other systems, as the one described in this thesis. The additional employment of *GEMMA* to overcome limitations in the algorithm by Marchal et al. (2005) comes with a heavy impact on the system's complexity, both in terms of running time and implementation. Though the combination leads to good matching results, the scalability of the system when analysing larger numbers of trajectories is uncertain.

3 Preliminaries

In this chapter, we will review important preliminaries and thereby give a more formal definition of the map matching problem. Several of the following definitions are loosely based on the definitions proposed by Lou et al. (2009) and are similar to those of other recent papers on the topic.

Definition 1 (*GPS point*)

> A **GPS point** p_{GPS} is a set of data acquired in a single position computation by a GPS receiver. It contains the determined latitude $p_{\text{GPS}}.lat$ and longitude $p_{\text{GPS}}.lon$, as well as additional data like the elevation and a timestamp.

In this thesis, the timestamp is implicitly used by ordering the GPS points consecutively. Furthermore, we apply the UTM projection to the GPS points, which are originally described by geographic latitude and longitude. As a result, we are able to compute Euclidean distances between GPS points later on.

In typical map matching cases, GPS points are collected by a moving vehicle or other objects equipped with a GPS receiver. These sensors may differ regarding their measurement noise and sampling rates. In order to track the way the receiver took, obviously more than one GPS point must be available, which leads to the next definition.

Definition 2 (*GPS trajectory*)

> A **GPS trajectory** is a sequence $T = \{p^1_{\text{GPS}}, p^2_{\text{GPS}}, ..., p^n_{\text{GPS}}\}$ of n GPS points in ascending order of their timestamps. GPS trajectories are used to represent the data collected by a GPS receiver logging GPS points over spatial and temporal distance.

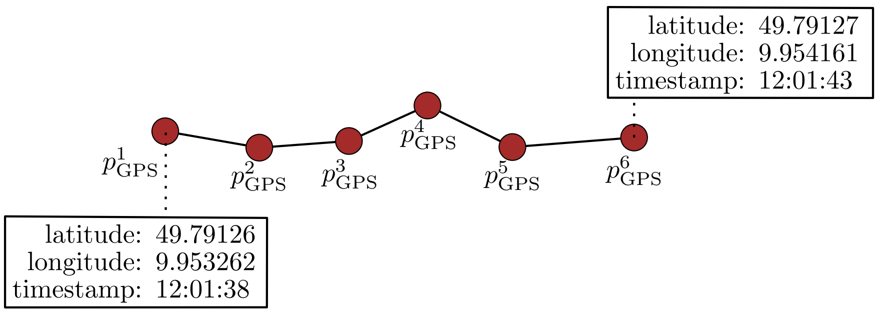

Figure 3.1: A short GPS trajectory. GPS data of start and end point listed.

The distance between two consecutive GPS points of a trajectory depends on the sampling rate of the GPS receiver in use and the speed of movement. Generally speaking, map matching algorithms tend to produce better results with trajectories of higher resolution. Nevertheless, approaches like those of Lou et al. (2009) can cope with sampling rates as low as one point every 5 minutes from a moving car—assuming that the driver prefers shortest paths. In the following, we speak of *low* sampling rates when the distances between the GPS points lie in the order of 100 meters, of *medium* sampling rates for distances in the order of 10 meters and of *high* sampling rates with spaces in the order of 1 meter.

Having defined sufficient terms to describe the measurement data that forms the input to map matching algorithms, it is now necessary to introduce their counterpart, the road network, in relation to which the captured paths shall be matched.

Definition 3 (***Road vertex***)

A **road vertex** v is a vertex of a polyline that represents a road. It is defined by its latitude $v.lat$ and its longitude $v.lon$. Additionally, it is associated with the road segment $v.seg$ on which it lies.

The road vertices can have degrees of 1 (in case of a dead end), of 2 (modelling a bend of the road) and any higher (at junctions).

Definition 4 (***Road segment***)

A **road segment** seg is a segment of a polyline representing a road. Is is defined as a directed edge starting in the road vertex $seg.start$ and ending in the road vertex $seg.end$. It furthermore holds its length $seg.length$, which is the length of the way from $seg.start$ to $seg.end$ travelling on the road segment.

Definition 5 (***Map point***)

A **map point** p_{map} is a point lying on a road segment. It is defined by its latitude $p_{\text{map}}.lat$ and its longitude $p_{\text{map}}.lon$. Additionally, it is associated with the road segment $p_{\text{map}}.seg$ on which it lies.

With definitions for vertices and corresponding edges at hand, we now define the road network:

Definition 6 (***Road network***)

The **road network** is a weighted, directed multigraph $RN = (V, E)$. The vertices in V are the road vertices, while the road segments as edges form the set E. Every edge is weighted by its length. The source material in use is called the **road database**.

This graph is typically constructed using existing cartographic material, which is available both commercially from services like Google Maps and Bing as well as freely from community-driven projects like Open Street Map. If available, additional information as speed limits and one-way roads can be taken into account, too.

Figure 3.2: Detail of a road network for the city of Würzburg

The term **map matching** thus describes the procedure of reconstructing the way a moving object took on a map using collected GPS data. In other words: with a given GPS trajectory and a given road network, the map matching system computes the path on the map that corresponds best with the trajectory. The question, which characteristics make a path *correspond* with the trajectory, is therefore of central importance to the design of map matching algorithms.

4 An existing map matching approach by Lou et al. (2009)

As this thesis mainly intends to propose new strategies for map matching on *incomplete databases*, an existing map matching algorithm for complete databases has been chosen as a sound base to work on. From the multitude of algorithms introduced by various authors in the past, we decided to follow the approach described by Lou et al. (2009), applying several modifications explained in Chapter 5. This approach seemed particularly appropriate to the author for three reasons.

First, it is solid regarding suboptimal conditions, e.g. missing samples and low sampling rates. Second, the implementation is rather straight-forward and clearly arranged. This ensures quick access to the multiple criteria featured, which can be used for fine tuning or enhanced with additional concepts. Third, Lou et al. (2009) provide structures that are particularly suitable to be augmented with extensions for matching on incomplete databases.

We detail the chosen approach in the following sections, starting with a short system overview. Several improvements to the algorithm described in this chapter will be presented in Chapter 5.

4.1 Overview

The implemented system consists of four major parts. At first, the input data, which consists of GPS logs and a geographic road map, is being parsed. Afterwards, for each GPS point a collection of possible candidate points on the road network is determined. These candidate points are analysed in terms of their quality to represent their associated GPS point. With that information, a candidate graph is constructed, which provides in a final step the path through the road map that corresponds best to the trajectory.

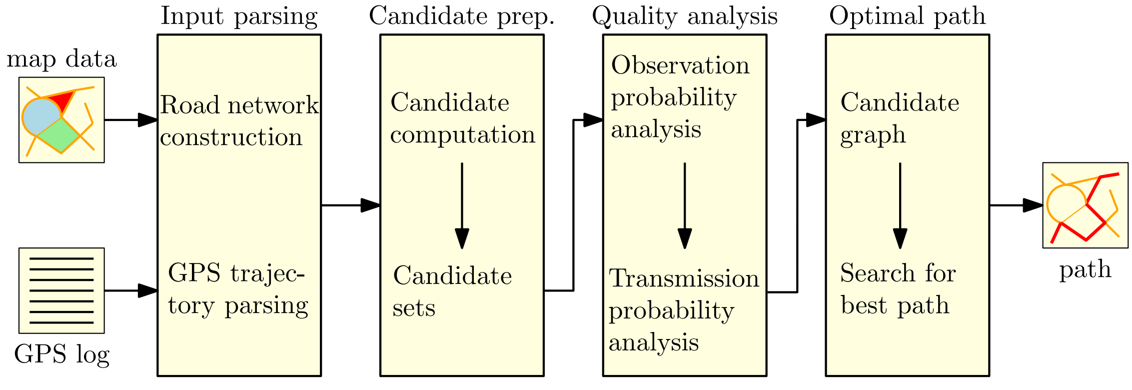

Figure 4.1: System overview

For the implementation, it is of advantage to analyse the quality of the candidate points directly when gathering the candidate sets. This has no effect on the system's overall performance and correctness, though.

4.2 Input parsing

In order to work properly, the system needs a road network as defined in Chapter 3 to operate on. An appropriate network can be constructed from various geographic material in different data formats. In our implementation[1], ESRI shape files containing data from the Open Street Map[2] project can be parsed, creating the network. Vertices of a polyline in the shape file are used as road vertices and the road segments connecting them as edges. These are weighted according to the Euclidean distance between start and end point. As we do not intend to limit our system to car traffic but are especially interested in pedestrians, too, we always insert edges leading in both directions and thus do not take driving restrictions on one-way roads into account.

To speed up computation later on, the edges of the road network are additionally stored in a Quadtree using their bounding boxes as envelops. This ensures a quick search for roads (and map points) close to a given GPS point when computing candidate points.

Aside from the road network, the system needs a GPS trajectory to process. As introduced in Chapter 3, a GPS trajectory consists of a sequence of single GPS points. There exists a multitude of data formats for GPS logs, and it is of little importance which is actually used. We implemented a parser for GPX files, a format that is based on XML and designed to hold various GPS data. Importing one data point after the other and creating a GPS point from each, we receive a GPS trajectory to work with.

In contrast to the GPS trajectory, the road network has not necessarily to be constructed from scratch every time the algorithm is run. As long as it is not out-of-date regarding the real road system and as long as the GPS trajectory lies inside the covered area, there is no need to rebuild the road network. Therefore, the runtime complexity of road network construction is not that relevant when requesting multiple matching computations.

As Lou et al. (2009) do not in detail explain their procedure for input parsing, it might be slightly different from the one presented here. Nevertheless, this question is more of technical than of conceptional relevance.

4.3 Candidate preparation

The next major component of the system is the candidate preparation. Finding the *best* fitting map point for a given GPS point requires first to find a set of *possible* candidates. Having used the term vaguely before in this thesis, it is now time for a proper definition.

Definition 7 (*Candidate point*)

> A **candidate point** c is a map point that is considered to be a possible matching candidate for a given GPS point p^i_{GPS}. It is associated with its corresponding GPS point $c.gps = p^i_{\text{GPS}}$. All candidate points to p^i_{GPS} form its set of candidates C^i.

For every GPS point p^i_{GPS} in the trajectory, an individual set of candidate points C^i has to be established. Appropriate elements for C^i must lie on road segments in the vicinity of p^i_{GPS}, taking from every segment the very point with the least Euclidean distance to p^i_{GPS} into the candidate set. These formal criteria have to be met by every chosen candidate point.

[1]Parts of the framework code were reused and originally written by Haunert and Sering (2011).
[2]See http://www.openstreetmap.org/ for the Open Street Map project.

To practically determine a set of candidate points to a given GPS point p_{GPS}, the system first gathers the road segments that lie within a given radius r around p_{GPS}. This procedure contains three steps:

1. Quadtree query returning near road segments
2. Line segment projection to find map point with minimal distance to p_{GPS} for each road segment
3. Verify that the obtained points lie inside the area limited by r

In the first step, the Quadtree that contains the edges of the road network is used. Queried with a square envelope of side length $2r$ enclosing p_{GPS}, the Quadtree returns all possibly intersecting road segments. The found segments are subject to a line segment projection in a second step. Using simple vector analysis, the nearest point in respect to p_{GPS} on each segment is determined. Of those points, each is finally examined whether it lies inside the area limited by the given radius r. The points that pass the three steps form the candidate set for p_{GPS}.

Technically speaking, some of the candidates points might actually not be vertices contained in the road network. This is especially true for all points on road segments that are *not* their start or end points. However, to be able to find shortest paths between candidate points, it is necessary that these map points are injected as road vertices into the road network graph. They are added together with edges representing the paths to start and end point of their road segment, thus splitting it into two segments.

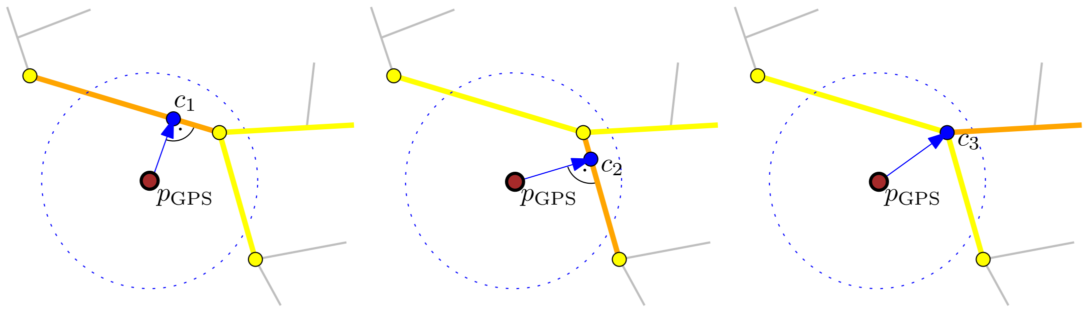

Figure 4.2: Candidate points

Figure 4.2 gives an example of candidate point computation. The target area limited by radius r around p_{GPS} is depicted as blue, dotted circle. The coloured lines illustrate road segments that contain points within range. On each of them, the point with the lowest distance to p_{GPS} is highlighted and considered a candidate. Note that the first two candidate points were originally not vertices of the road map graph. The grey elements are of no importance for candidate computation, as they do not contain points within the search radius.

The described procedure is repeated for every GPS point in the trajectory. If there are no candidate points found for a GPS point, for example in the case of an outlier due to a measurement error, it is no longer considered. For all other GPS points, a list of candidate points is available at the end of this step.

4.4 Candidate quality analysis

Now that we have determined a set of candidate points for each GPS point, a selection must be made to pick the best one in each case. As Lou et al. (2009) state, the problem now becomes to find a path on the road network that visits one candidate for every GPS point and in addition best matches the GPS trajectory. To achieve this, the candidate points themselves and possible connections between them are rated according to two quality criteria: observation probability and transmission probability.

Definition 8 (***Observation probability***)

The **observation probability** is the probability that a candidate point c matches a GPS point p_{GPS}. It is computed based on the relative positions of c and p_{GPS}. Assuming that GPS measurement errors can be described as a normal distribution, the observation probability $N(c_j^i)$ computes as

$$N(c_j^i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_j^i - \mu)^2}{2\sigma^2}}$$

where $x_j^i = dist(c_j^i, p_{\text{GPS}}^j)$ is the Euclidean distance between c_j^i and p_{GPS}^j, and with mean $\mu = 0$ and standard deviation $\sigma = 20\text{m}$ on an empirical basis.

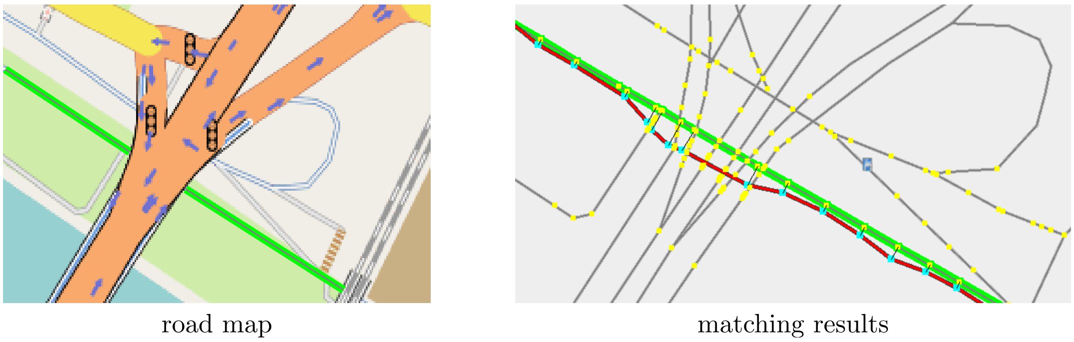

Figure 4.3: Overpass intersection. Of the large set of candidates (yellow dots), only the few beneath the bridges are appropriate to the trajectory (red) and the matching result (green).

At first glance, this measure might seem sufficient for picking the best candidate point in each set. However, there exist several situations in which observation probability alone can not guarantee the best candidate choice. Especially at parallel roads or overpass intersections, unsuitable candidate points might be located next to the examined GPS point (cf. Figure 4.3). Therefore, a second measure is necessary, which takes the chance of transition from the point before into account: the transmission probability.

Definition 9 (***Transmission probability***)

The **transmission probability** is defined as the probability that the shortest path from a given candidate point c_j^{i-1} to the next candidate point c_k^i is the correct path from p_{GPS}^{i-1} to p_{GPS}^i. Following Lou et al. (2009), the transmission probability $V(c_j^{i-1} \to c_k^i)$ is computed as

$$V(c_j^{i-1} \to c_k^i) = \frac{d_{(i-1)\to i}}{w_{(j,i-1)\to(k,i)}}$$

with $d_{(i-1)\to i} = dist(p_{\text{GPS}}^{i-1}, p_{\text{GPS}}^i)$ and $w_{(j,i-1)\to(k,i)}$ is the length of the shortest path between c_j^{i-1} and c_k^i on the road network.

In Chapter 5, we will discuss shortcomings of the definition above and propose improvements. Nevertheless, the consideration of transmission probabilities is important for the presented approach and its extensions. Compare to Figure 4.3 for an example where both observation and transmission probability are necessary.

Lou et al. (2009) also propose the **temporal analysis** of candidate points. With this concept, temporal data as the average moving speed and speed limits on the road network are taken into account. This feature has not been implemented in our approach due to relatively little gain in contrast to increased requirements on the cartographic material. Furthermore, the temporal analysis cannot be applied reasonably when travelling on roads not contained in the road network.

4.5 Finding an optimal path

After the evaluation of each candidate point in terms of observation and transmission probability, the results have to be processed. Particularly, a multitude of possible combinations of candidate points and the paths between them must be considered to obtain the final matching result.

Definition 10 (*Final matching result*)

The **final matching result** is the path through the road network best fitting the GPS trajectory with respect to the applied quality terms. It is defined as the path

$$c^1 \to c^2 \to \cdots \to c^n \text{ with } c^i \in C^i \text{ for } i = 1 \ldots n$$

so that its associated quality rating

$$N(c^1) \cdot V(c^1 \to c^2) \cdot N(c^2) \cdot V(c^2 \to c^3) \cdot \ldots \cdot N(c^{n-1}) \cdot V(c^{n-1} \to c^n) \cdot N(c^n)$$

is maximal.

For the purpose of computing this path, the last component of the system features a special graph, in which the candidate points and their quality ratings are represented: the candidate graph.

Definition 11 (*Candidate graph*)

A **candidate graph** is a directed acyclic graph (DAG) which represents all paths that are considered possible final matching results for the given GPS trajectory. While the determined candidate points serve as vertices to the candidate graph, its edges represent a link between two candidate points on the road network. The edges are weighted according to the transmission probability of their start and end vertices as well as the observation probability of their start vertex. The graph furthermore contains the vertices p_{start} and p_{end}, which feature edges to all candidates of the first and the last GPS point, respectively.

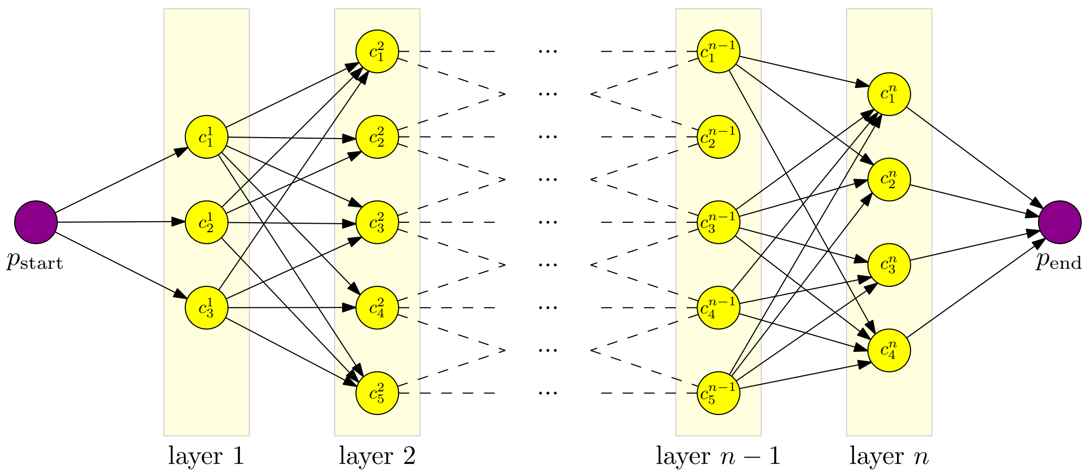

Figure 4.4: Candidate graph

To clarify this definition, Figure 4.4 shows a schematic depiction of a candidate graph. Note that each set of candidate points determined for a particular GPS point forms a layer in the candidate graph. Figure 4.5 shows an example of one of the candidate sets and its corresponding layer. As we can also clearly see in Figure 4.4, almost all vertices in two consecutive layers are connected with edges. Two vertices stay unconnected only if the road network lacks a path between the candidates they represent.

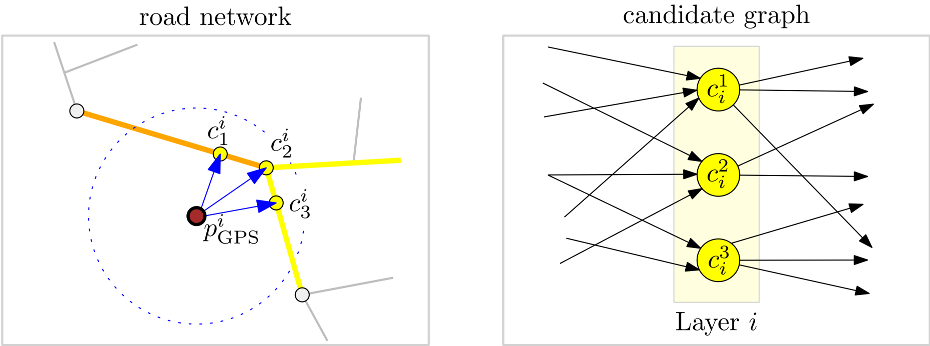

Figure 4.5: Candidate set and corresponding layer

Moreover, the paths from p_{start} to p_{end} through the candidate graph form the set of possible final matching results for the trajectory. However, the edges these paths consist of are not simply weighted with the product of the probabilities as proposed in Definition 10. Instead, we convert the product into a sum using logarithms, and achieve simplified circumstances for later computations.

In order to obtain the final matching result, the best rated path through the candidate graph has to be determined. This is possible under the assumption of optimal substructures using dynamic programming as described by Cormen et al. (2009): The final matching result can be considered a combination of smaller, optimal results for parts of the GPS trajectory. Note that the structure of a directed acyclic graph allows an efficient search for the longest (i.e. best rated) path through the graph.

From the final matching result, the path on the road network taken can easily be deduced, so that the matched set of road links is available to external systems for further processing or graphical representation.

5 Improvements of the basic algorithm

Now that we have considered the approach by Lou et al. (2009) and assured ourselves of its advantages, we want to discuss some of its shortcomings, too. Several problems arise out of the algorithm's original design for low sampling rates when applied to higher sampling rate trajectories. In addition to our solutions to those, we suggest ideas to improve the candidate quality rating.

The quality of map matching results will benefit from our modifications, both with and without using the extensions for incomplete road databases proposed in Chapter 6. In this chapter, we want to examine shortcomings and our suggested improvements to them consecutively.

5.1 Normalisation of transmission probability

As we have seen in Chapter 4, Lou et al. (2009) defined the probability that two successive candidate points c_j^{i-1} and c_k^i follow each other as

$$V(c_j^{i-1} \to c_k^i) = \frac{d_{(i-1)\to i}}{w_{(j,i-1)\to(k,i)}}.$$

The appeal of this definition clearly lies in its straightforwardness; however it has the drawback that its co-domain is not limited to the interval $[0, 1]$. As Figure 5.1 shows, in certain cases the road distance between two consecutive candidate points might be smaller than the Euclidean distance between their corresponding GPS points. This leads to a transmission probability greater than 1, which is generally an undesirable behaviour for a "probability" and problematic as its (potentially high) value enters directly into the overall quality computation, together with normalised measures.

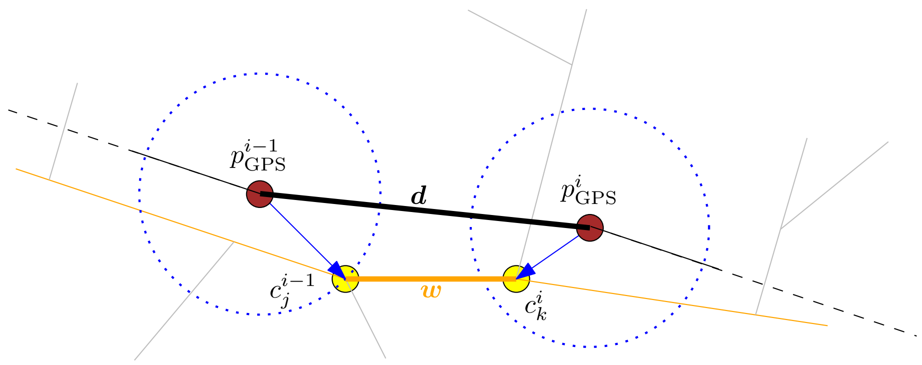

Figure 5.1: Distance w between two consecutive candidate points is shorter than distance d between the two corresponding GPS points.

The described effect is mainly caused by the matching radius r, which – in the worst case – allows the road distance w to be $2r$ shorter than the distance d between the GPS points. However, in the setting where Lou et al. (2009) used the algorithm, this drawback has little impact due to low sampling rates and the resulting long distances between the candidate and GPS points.

As we wish to apply the concept of transmission probability to a high sampling rate context as well, we have to erase this shortcoming. Therefore, the following modification to the definition of the probability is suggested.

Definition 12 (***Transmission probability: improved definition***)

The transmission probability $V(c_j^{i-1} \to c_k^i)$ is computed as

$$V(c_{i-1} \to c_i) = \frac{\min\{d_{(i-1)\to i}, w_{(j,i-1)\to(k,i)}\}}{\max\{d_{(i-1)\to i}, w_{(j,i-1)\to(k,i)}\}}$$

with $d_{(i-1)\to i} = dist(p^{i-1}_{\text{GPS}}, p^i_{\text{GPS}})$ and $w_{(j,i-1)\to(k,i)}$ is the length of shortest path between c_j^{i-1} and c_k^i on the road network.

The definition of the probability as the minimum of direct distance d and length of way w divided by the maximum of d and w guarantees probability values in the interval $[0, 1]$, thus solving the stated problems. In the formerly problematic cases, where w is shorter than d and the probability was greater than 1, the value is now smaller than 1 expressing a suboptimal matching quality. The concept of transmission probability itself remains justified and untouched.

5.2 Avoidance of loops in matched road path

Another problem we encountered is the occurrence of loops on the matched road path when using the map matching system with high sampling rate input data. With the term "loops", we refer to sections of the matched road path with the path leading to a candidate point and returning on the same road segment again to the next candidate point. This misbehaviour is mostly observed at scenarios where the GPS trajectory deviates from the road network and is caused by flaws in the conception of the transmission probability. Figure 5.2 shows an example for a (wrongly matched) loop.

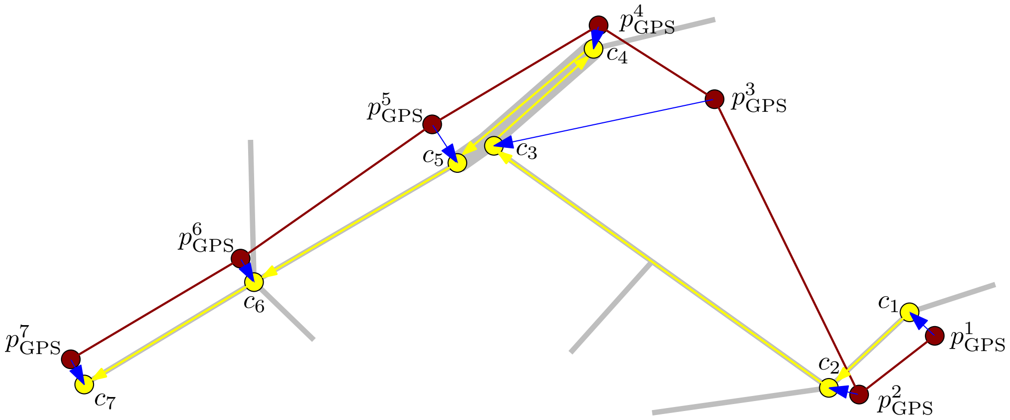

Figure 5.2: The algorithm by Lou et al. (2009) produces a loop on the road network. Note that the distance between c_2 and c_3 best fits the distance between p^2_{GPS} and p^3_{GPS}. The same is true for the next three pairs respectively. Nevertheless, the matching result is not satisfying, as $c_3 \to c_4 \to c_5$ form a loop not existent in the GPS trajectory.

As the appearance of such loops in the final matching result are highly undesirable, we have to modify the quality rating system again. While the proximity of candidate points is considered by the observation probability and the similarity of covered distances by the transmission probability, there is so far no mechanism to explicitly ensure correct directions of the matched road segments. Therefore, we introduce a third measure of quality for the candidate points:

Definition 13 (*Direction probability*)

The **direction probability** is based on the angular deviation between the line segment from a candidate point c_j^{i-1} to the next candidate point c_k^i and the line segment between the corresponding GPS points. It is normalised as

$$D(c_j^{i-1} \to c_k^i) = \frac{180° - \alpha}{180°}$$

where α is the angular deviation in degrees between the vector $c_j^{i-1} \to c_k^i$ and the vector $p_{\mathrm{GPS}}^{i-1} \to p_{\mathrm{GPS}}^i$. If one of the vectors has length 0, we define $D(c_j^{i-1} \to c_k^i) = 1$. The deviation is computed using basic vector analysis.

Figure 5.3 shows an example for the computation of the direction probability. The angular deviation amounts to 27°, which leads to a direction probability of $\frac{180°-27°}{180°} = 0.85$. The determined direction probability enters together with observation and transmission probability into the quality ratings represented in the candidate graph.

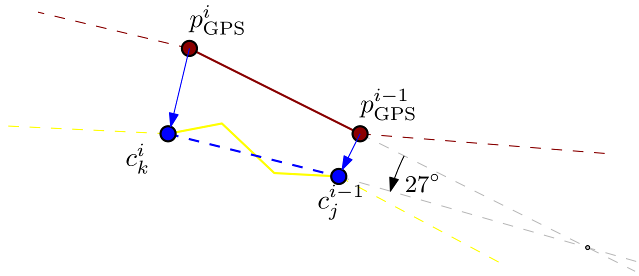

Figure 5.3: Angular deviation between GPS trajectory and matched road segment

Revisiting the case presented in Figure 5.2, now making use of the direction probability, we can clearly see in Figure 5.4 that the unwanted loops have disappeared and the final matching result is correct. It might be worth noting that, in the shown case, two GPS points share the same candidate point on the road network. In this case, $D(c_4 \to c_5) = 1$, which means the deviation is at its optimum.

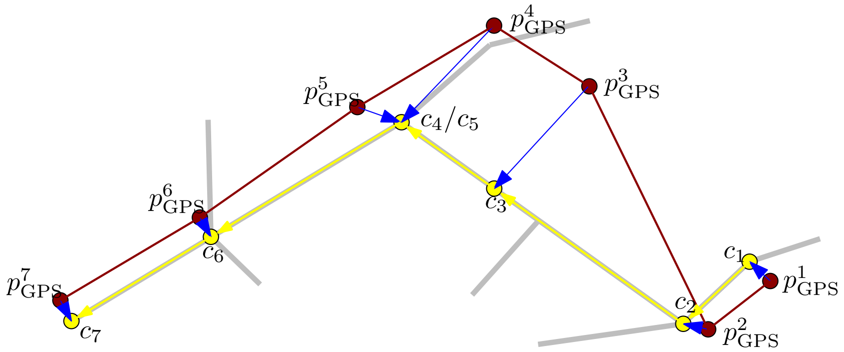

Figure 5.4: Same scenario as in Figure 5.2. The improved algorithm no longer produces a loop.

This might seem like a design flaw at first glance, but actually proves that the former loop has been penalised sufficiently. Better matching results are obtained in several cases by this behaviour.

In many situations, the direction probability is indeed sufficient to solve the loop problem. Nevertheless, there are worst case scenarios where the angular deviation as basic principle is not adequate, as we see in Figure 5.5.

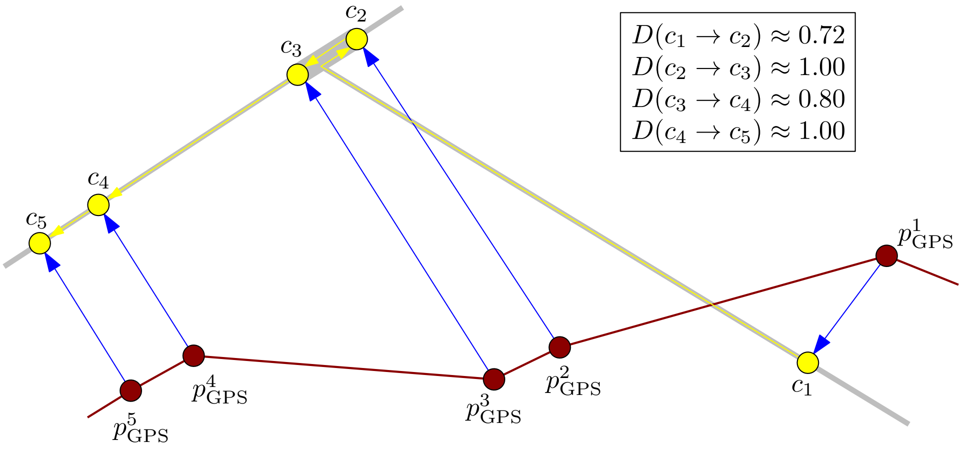

Figure 5.5: Direction probability does not in all cases prevent the occurrence of loops.

In the shown case, $D(c_1 \to c_2) \approx 0.72$, based on a angular deviation of about 50°, while $D(c_3 \to c_4) \approx 0.8$ and its deviation 35° respectively. For the remaining segments in the figure, direction probabilities are nearly optimal and for all matched segments, transmission probabilities are high. This combination leads to the illustrated, poor matching result.

As such worst case situations appear comparatively seldom, we have decided not to implement further improvements into the presented matching system. Nevertheless, an approach to solve this misbehaviour is proposed in Section 8.2 and is intended to be employed in future work. Anticipating the ideas of chapter 6, we note that the described problem is solved with the introduction of off-road points, too.

6 Extensions for incomplete road databases

6.1 Overview

In this chapter, we want to extend the described map matching system for operations on incomplete road databases. In doing so, we augment the current system of candidate quality rating and result matching accordingly. This means in particular that we take an approach different to Pereira et al. (2009), who realised a related system combining an algorithm proposed by Marchal et al. (2005) and a new genetic algorithm named *GEMMA*.

The motivation for introducing map matching on incomplete databases is numerous. The most obvious benefit—operation on a road network that is not fully mapped by cartographers—retreats into the background: at least in industrialised countries, complete data on the road networks is in most cases available. Instead, systems with "off-road" capability present their full potential on scenarios like those presented in Figures 6.1 to 6.3.

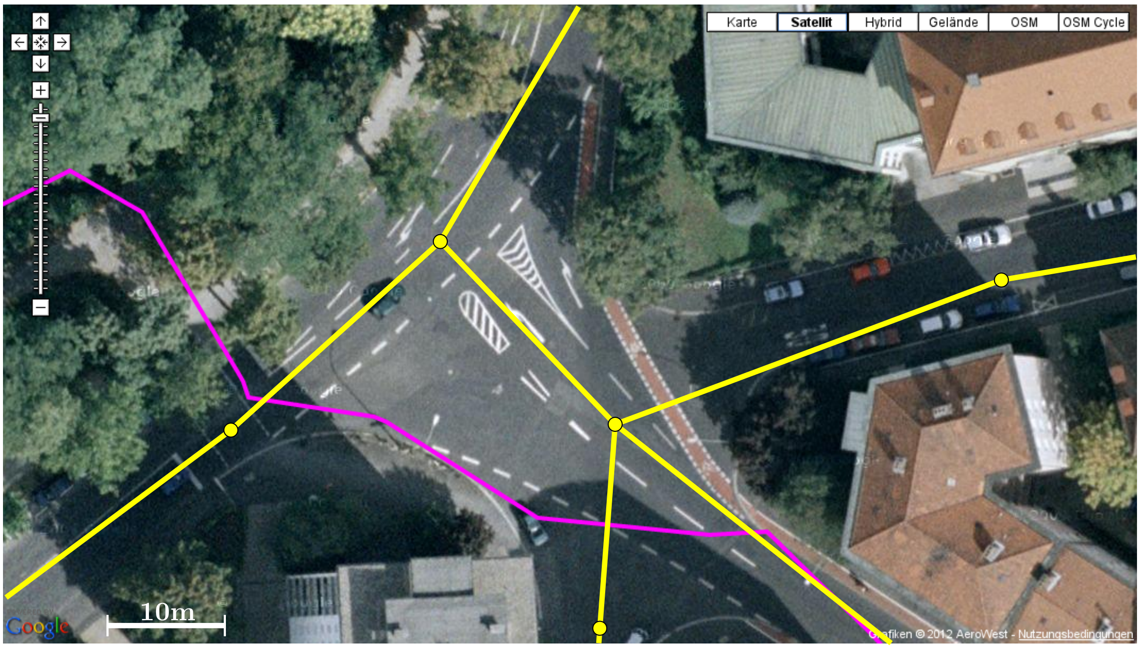

Figure 6.1: Crossing a large junction on foot, which is not satisfactorily modelled by edges of the road network.
Imagery ©2012 AeroWest, GeoBasis-DE/BKG, GeoEye, Google

In Figure 6.1, we see a trajectory (pink) recorded by a pedestrian. On the road network (yellow), the large junction displayed has to be represented by vertices and edges only, whereby information on the width of the junction is lost. As the pedestrian walks on the pavement and takes the shortest route through the junction, his path quite differs from the road segment on the network. Using an off-road compatible system, the path taken can be matched adequately, though.

Another informative case is given in 6.2: the trajectory crosses the urban highway moving through a new tunnel not yet included in the road network. Traditional map matching systems are entirely incapable of computing an acceptable matching result, while our off-road approach delivers an excellent result.

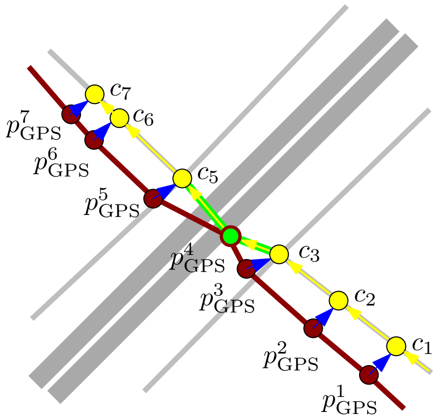

Figure 6.2: Missing segments on the road network bridged by off-road segments

Such scenarios appear frequently in practice, often caused by out-of-date cartographic material as well as the usage of temporary bypasses at road construction sites. Moreover, U-turns and forbidden turns that do not conform to the road network can be processed with much better results.

Last but not least, Figure 6.3 shows a trajectory that was recorded while actually moving off-road. Despite examining an urban area, walkers and bicyclists often move on trails and passages that are not part of the official road network. However, this rather fundamental problem can be solved elegantly with our off-road matching system as the figure shows.

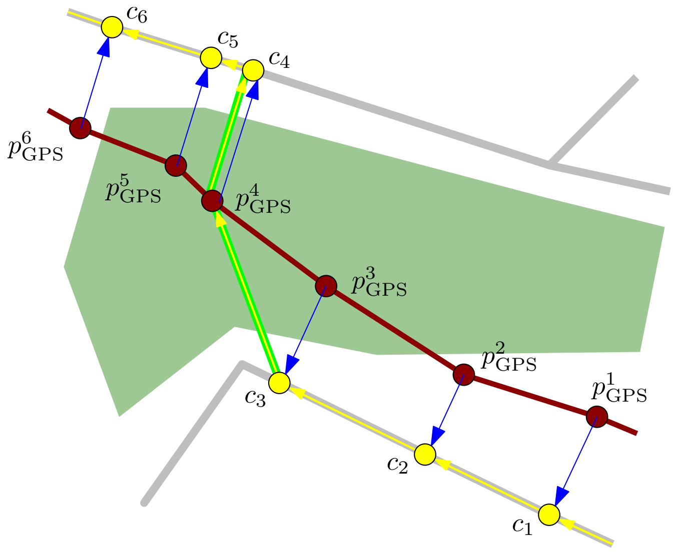

Figure 6.3: GPS trajectory features actual off-road passages, which are reflected by according off-road segments.

Now that we have briefly examined common use cases for off-road matching, we take a look at the actual implementation and conception of the proposed algorithm. As an underlying structure, the system described in Chapter 4 and improved in Chapter 5 is used.

6.2 Off-road points

To provide the desired off-road functionality, at first off-road points have to be injected into the road network. Thus, the road network is extended by one additional vertex for each GPS point in the trajectory. These new points are created as follows:

Definition 14 (*Off-road point*)

An **off-road point** p is a vertex injected into the road network, featuring the coordinates of its associated GPS point $p.gps$. It is considered a candidate point to $p.gps$.

With that, potential candidate points outside the original road network are available. In order to make them matchable, they need to be connected to other points. Therefore, additional road segments with the following definition are inserted.

Definition 15 (*Off-road segment*)

An **off-road segment** seg is a road segment with either $seg.start$, $seg.end$ or both being off-road points.
If $seg.end$ is an off-road point while $seg.start$ lies on the road network, seg is called **road exiting**. If $seg.start$ is an off-road point and $seg.end$ on the road map, seg is **road accessing**. In the remaining case, it is called a **true** off-road segment.

Strictly speaking, off-road points should be reachable from every map point in the network. However, this would mean a tremendous amount of additional edges needed, of which the great bulk is extremely unlikely to be ever used. Therefore, a more efficient concept has to be considered.

In the following approach, we assume that the path connecting two candidate points consists either exclusively of off-road segments or exclusively of road network segments. This assumption is supported by a closer examination of the structure of higher sampling rate trajectories, which shows that the distance between two consecutive GPS points is short enough for reasonably reducing potentially mixed on- and off-road paths to ones using exclusively one type of segments.

The gain of the proposed assumption is enormous. Instead of inserting edges from *every* vertex on the road network to each off-road point p^i, it is now sufficient to add edges from every *candidate point* associated to p^{i-1}_{GPS} and to every candidate point associated to p^{i+1}_{GPS}. In other words, the set S of off-road segments that has to be added for a GPS trajectory with length n is

$$S = \{s \mid \exists i \leq n : s.start \in C^{i-1} \wedge s.end = p^i\} \cup \{s \mid \exists i \leq n : s.start = p^i \wedge s.end \in C^{i+1}\}$$

where $p^i.gps = p^i_{\text{GPS}}$.

This concept guarantees good matching results while only taking a small, but crucial fraction of possible paths into account.

6.3 Quality analysis of off-road segments

In order to integrate the newly introduced off-road points and segments into the existing system, a quality rating mechanism for them has to be created. Basically, the same rating algorithm as for map points featured in the road network can be applied. Thus, the new off-road points are subject to the analysis of transmission and direction probability as described in Chapter 4 and 5.

The observation probability is not taken into account, as the off-road points and their GPS points share identical coordinates, which renders a corresponding examination useless. When matching outside the road network, the position of the logged GPS points is the only available information anyway.

Nevertheless, several changes have to be made for an operational off-road rating method. Due to the fact that off-road candidates exactly match their evoking GPS points, and off-road segments connect them with minimal Euclidean distance, the existing rating system would rate the off-road path along the trajectory as optimal matching result. That means, *not* matching to map points on the road network would always be favourable to the system.

To counteract such shortcomings, we have to introduce penalties for matching on off-road segments. In case of doubt, the matching system should stay on the roads of the road network. Therefore, we penalise all off-road segments with a certain value. To prevent the algorithm from jumping back and forth between road network and off-road segments, edges exiting and entering the road network are provided with an additional penalty.

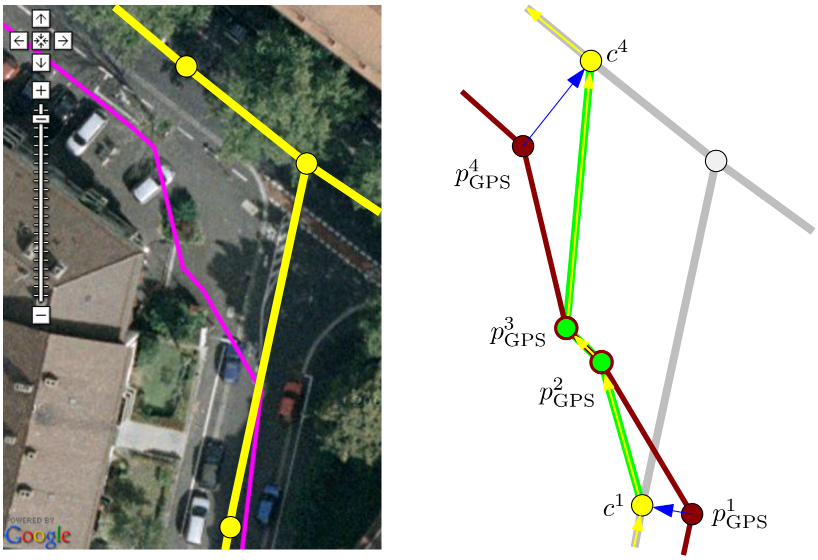

Figure 6.4: Off-road segments connecting c^1 to c^4. The mixed segments $c^1 \to p^2_{\mathrm{GPS}}$ and $p^3_{\mathrm{GPS}} \to c^4$ receive higher penalties than the true off-road segment $p^3_{\mathrm{GPS}} \to p^4_{\mathrm{GPS}}$.
Imagery ©2012 AeroWest, Google

The size of the proposed penalties somewhat relies on the sampling rate and especially on the accuracy of the GPS samples. In various tests, penalties with a value of $5 - 10\%$ in respect to the underlying rating results have proven appropriate. Moreover, the matching quality can be further improved by adjustment of the impact of direction probability.

6.4 Extended candidate graph

As the final matching result shall be obtained by shortest path search on the candidate graph, the proposed extensions must have impacts on its structure. As each GPS point is provided with an according off-road point as an additional candidate, these points have to appear in the candidate graph. Thus, to each layer in the graph another candidate point is added.

The off-road segments injected in the road network to provide the connection to and from the off-road points are represented by additional edges in the candidate graph. The assumptions in

Section 6.2 require each off-road point c^i_{GPS} to feature edges from every candidate c^{i-1}_j and to every candidate c^{i+1}_k. Their weights are assigned according to the quality rating rules of Section 6.3, that is, it additionally depends on whether or not the start and end points are off-road points.

Figure 6.5 shows the augmented candidate graph. The green/brown vertices illustrate the added off-road candidates, while their connections to the remaining candidate points are modelled by the brown edges. All other vertices and edges in the graph remain altogether unaffected by the off-road extensions.

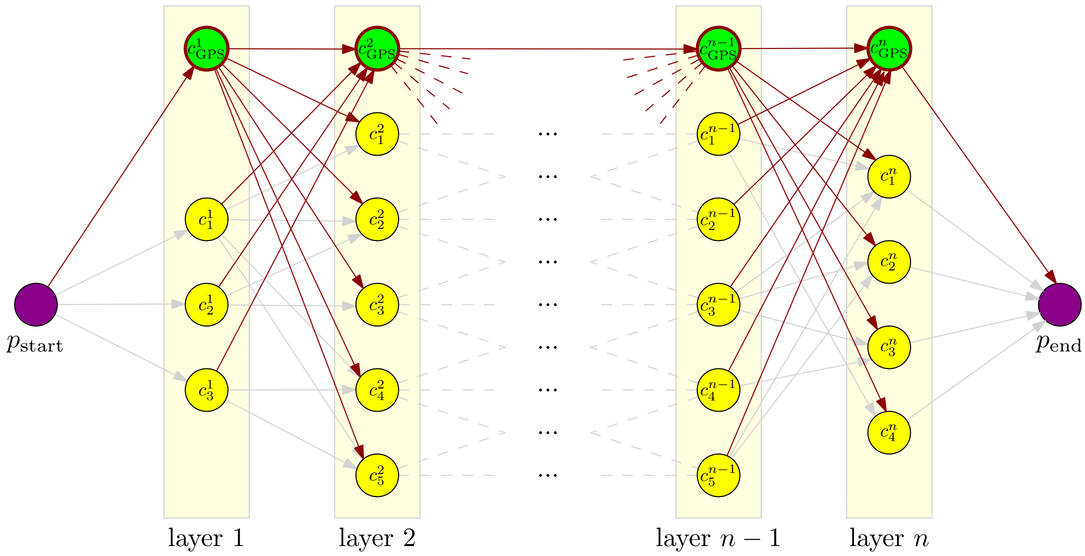

Figure 6.5: Candidate graph with off-road points and edges

As we can clearly see, the fundamental structure of the candidate graph has not been touched by the modifications: it is still acyclic, layered and directed. Therefore, the same techniques as in Chapter 4 can be used to obtain the shortest path through the graph, returning the final matching result. The optimal path found can now be expected to consist of road segments where possible and off-road segments where necessary.

7 Evaluation and running time analysis

Having presented our map matching system for incomplete databases in the chapter before, we want to examine and evaluate its performance in this chapter. Thereby, the quality of the returned optimal paths as well as the running time of the algorithm will be analysed. Both opportune and disadvantageous values of external parameters like sampling rate and measurement error level have to be tested on their impact to the matching quality.

In addition, we draw a short comparison to other approaches on incomplete databases. Taking a look at their algorithmic conception, we point out several differences to the approach proposed in this thesis.

7.1 Matching quality under different conditions

To examine the increase of matching capability that has been newly gained by deploying off-road extensions, we consider the scenario given in Figure 7.1. It compares two different strategies for handling "gaps" on the road network, that is, locations where the trajectory follows a path not existent in the (thus incomplete) road network.

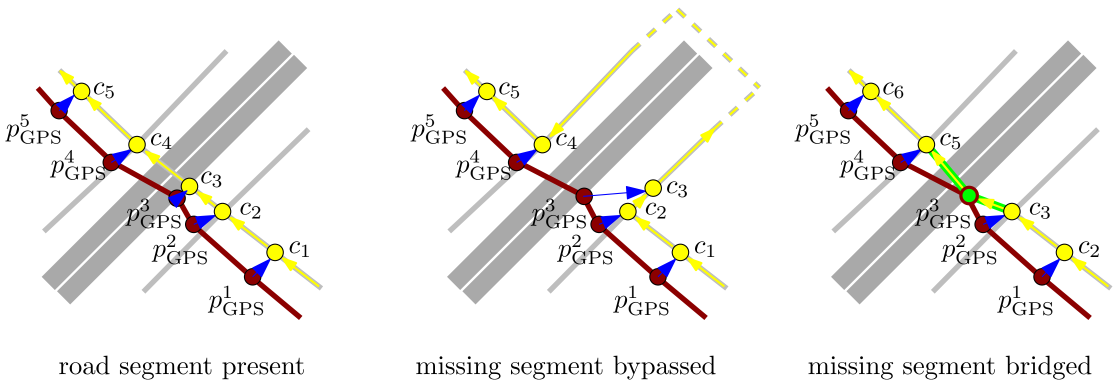

Figure 7.1: Strategies for handling gaps on the road network

On the left, the situation is shown on a *complete* road network. The system correctly matches the GPS trajectory to the adjacent road segments using the introduced quality ratings, irrespective of whether off-road extensions are available or not.

In the middle, the road segment connecting c_2 with c_4 is missing. Without off-road extensions available, the system tries to *bypass* this gap on the road network. This leads to detours of potentially immense length and thus to highly unsatisfactory matching results, as depicted in Figure 7.2.

On the right, the road segment is still missing, but the system makes use of its off-road extensions. Hence, the gap is *bridged* using the off-road candidate associated to p^3_{GPS} and the final matching result closely resembles the result obtained on the complete road network.

This example shows that the ability to operate on incomplete databases is especially valuable when neuralgic segments like bridges, tunnels etc. are missing in the road database. This is also the case when newly constructed roads are not yet available in the road network.

road segment present — missing segment bypassed — missing segment bridged

Figure 7.2: Implication of the different strategies on the matching result. The three scenarios correspond to the details shown in Figure 7.1.

In the tests hitherto presented, we assumed a high-sampling-rate GPS trajectory. However, the system produces considerably good results on medium sampling rates, too. As described in Chapter 6, we rely on the assumption that two candidate points are connected with a path either completely on-road or completely off-road. When sampling rates decrease and thus the distances between the GPS points and their candidates increase, this assumption is no longer true in the majority of cases. The following considerations are summed up in table 7.1.

Sampling rate	Increase of quality
low	no improvement
medium	better handling of missing neuralgic segments
high	off-road movement and missing segments fully supported

Table 7.1: Increase of matching quality when using off-road extension, ordered by sampling rate

In spite of the fact that Lou et al. (2009) designed the underlying system for low sampling rate trajectories, our extensions for incomplete road databases cannot improve the matching quality in such scenarios. At least as long as we consider urban areas, typical off-road movements cover only very small distances of a few meters, e.g. bridging two existent road segments. On low sampling rates with distances of several hundred meters between the GPS logs, such deviations can hardly be captured and do certainly not fulfil the necessary assumption named above. Nevertheless, the off-road extensions might produce useful results when applied on non-urban low sampling rate trajectories that describe longer off-road trips occasionally visiting mapped road segments on the way.

With medium sampling rate trajectories, the situation is much better. Though short deviations from the road network can often not be recognized due to the low resolution, missing road segments can be bridged very efficiently, producing considerably better matching results.

When high sampling rate trajectories are available, the system works best. As we have seen in several examples before, the system is able to detect and handle short deviations from the road network as well as missing segments with very good results.

7.2 Performance in comparison to the base algorithm

In the following section, we want to generalise the observations we made in Section 7.1. Therefore, we need a wider database of cases for examination. In order to generate incomplete road networks, a certain percentage of randomly chosen roads has been dropped from a given complete network. Afterwards, the different systems compute matching results for a given trajectory on the incomplete network, which we can compare.

Predictably, the basic algorithm by Lou et al. (2009) returns poor results as the integrity of the road network declines, that is, as considerable parts of the road network are missing. Our extended algorithm works much better and returns results that are very similar to those computed on the complete road network, as Figure 7.3 shows.

road network integrity 100%
basic system in use

red: trajectory
green: matched path

road network integrity 50%
basic system in use

red: trajectory
green: matched path

road network integrity 50%
off-road system in use

red: trajectory
green: matched path
blue: off-road segments

Figure 7.3: Comparison of performances on incomplete road networks

With this instrument of randomised testing at hand, a broad spectrum of possible scenarios and their impacts on the computed results can be examined efficiently. In the following test, both the basic algorithm by Lou et al. (2009) and our own extended system are on the trial. In 100 iterations for each scenario, a randomly trimmed road network has been generated, the algorithms have been run and their average deviations from the result on a complete network as well as from the trajectory have been recorded.

As the candidate points are limited to a small area around each GPS point anyway, only the different lengths of the matched paths are of value for a comparison. The results of this testing are shown in Table 7.2.

SYSTEM IN USE	NETWORK INTEGRITY	AVG. DEVIATION TO COMPL. NETWORK [m]	AVG. DEVIATION TO TRAJECTORY [m]	OFF-ROAD RATIO	NO PATH FOUND
off-road	100%	0	17.20	18%	0%
off-road	90%	2.14	14.98	26%	0%
off-road	80%	3.99	13.13	34%	0%
off-road	70%	5.01	12.12	42%	0%
off-road	50%	7.73	9.39	56%	0%
off-road	25%	13.69	3.43	79%	0%
basic	100%	0	18.72	0%	0%
basic	90%	986.58	1008.60	0%	15%
basic	80%	2402.71	2429.84	0%	31%
basic	70%	3543.12	3587.69	0%	58%
basic	50%	–	–	0%	97%
basic	25%	–	–	0%	100%

Table 7.2: Results of a randomised testing of the basic and extended algorithm. System in use, road network integrity, average deviation from result on complete network, average deviation from trajectory as well as off-road ratio and occurred errors are listed.

As we can clearly see, both systems deviate from their results on a complete road network in the different scenarios. However, our system for incomplete road databases computes results that converge with the given trajectory, whereas the basic system produces results that become increasingly detached from the original path. Nevertheless, our system still matches the trajectory to the road map wherever possible, as the off-road ratio shows.

In addition, the basic system produces frequently errors, not finding a path at all, as the integrity declines, because the road network is no longer connected – a condition that cannot be handled properly.

To draw a conclusion, the system proposed by Lou et al. (2009) is not at all capable of operation on incomplete road networks. Even if only 10% of the roads are missing, the algorithm returns results that are 100 times longer than its result on a complete network. Our algorithm handles missing roads as intended, bridging them with off-road segments and returning acceptable results even on highly incomplete road networks.

7.3 Running time analysis

To analyse the running time of our proposed system, we have to consider the complexity of its parts. All things considered, their complexity has not changed compared to the basic system by Lou et al. (2009).

As defined in Chapter 3, the number of GPS points in the processed trajectory is denoted as n, and let m denote the number of road segments in the network. Additionally, let N be the number of road vertices and the number of candidate points for each GPS point be limited by k. This can be justified by the fact that existing road networks do not feature infinite numbers of road segments.

During the construction of the candidate graph, $(n-1) \cdot k^2$ possible connections between candidate points have to be examined and at most the same number of edges have to be added. In doing so, the transmission probability has to be computed for every connection, which requires the shortest path between start and end candidate point. We use Dijkstra's algorithm to obtain the shortest path, which runs in $O(m + N \log N)$ time, and thus come to a total complexity of $O(n \cdot k^2 \cdot (m + N \log N))$.

The computation of the remaining probabilities and components is dominated by this bound: The observation probability can be obtained in constant time for each candidate point, resulting in a total running time in $O(n \cdot k)$ computing it for all candidates. The direction probability is computed in constant time for each of the edges in the candidate graph, too. That leaves us with a running time that lies in $O(n \cdot k^2)$ collectively. For the other components used for the construction of the candidate graph, similar estimates hold, so that they are dominated by $O(n \cdot k^2 \cdot (m + N \log N))$, too.

The second computational intensive component of the system is the search for the final matching result in the candidate graph. Deploying the *FindMatchedSequence* procedure proposed by Lou et al. (2009), we are able to find the longest path in the candidate graph within $O(n \cdot k^2)$, making use of the topological order of the graph.

Combining these bounds, the system has a time complexity of $O(n \cdot k^2 \cdot (m + N \log N))$. Lou et al. (2009) also propose to choose a small value for k, which holds in most practical scenarios, and brings the complexity near to $O(n \cdot (m + N \log N))$.

To conclude, the complexity of the system does not change with the introduction of our off-road extensions. Every layer in the candidate graph is extended with only one additional candidate point, which has no impact on the asymptotic running time of the candidate graph construction nor on that of the search for the final matching result.

8 Conclusion and future work

8.1 Conclusion

In this thesis, a new algorithm for map matching on incomplete databases was introduced. From the multitude of map matching systems proposed in the past, we decided to base our new system on the algorithm by Lou et al. (2009). In a first step, this algorithm was improved by several newly introduced modifications, which solved most of its important shortcomings. Afterwards, we modified the system with our off-road extensions, thus allowing operation on incomplete road databases.

In both cases, we augmented the basic system prudently, leaving its core structures intact. This has several benefits: first, the layout of the algorithm is still well-organized and extendible. Additionally, its asymptotic running time behaviour has not changed, although the functionality and the quality of the results has been considerably improved.

Having identified scenarios that profit from off-road capability, we tested our system in several ways. One of the tests involved more than 1000 randomly thinned out road networks. We finally concluded that our proposed system produces excellent results on high-sampling-rate input, and reasonably good results with medium-sampling-rate GPS trajectories in almost all scenarios.

8.2 Future work

In the future, our system could be further extended with the following ideas and become embedded in map generation projects as described by Pereira et al. (2009).

As already illustrated in Chapter 5, the introduced direction probability is not in all cases capable to guarantee correct matching results. Therefore, its concept might be refined in the future, examining the angle between incoming and outgoing paths in every candidate point in relation to the corresponding GPS point. This should solve the few remaining problems caused by wrong quality ratings.

Additionally, the observation probability could take the accuracy of the GPS receiver into account. Introduced in Chapter 4 on the base of a normal distribution, the standard deviation was fixed to 20m. Instead, information about the dilution of precision (DOP), as determined by many GPS receivers, could provide a individual distribution for every GPS point. This could improve the selection of candidate points.

Pereira et al. (2009) propose an interesting application for map matching on incomplete databases that leads beyond the mere visualisation of a travelled path. In their *YouTrace* project, off-road paths are collected and used for road database construction. Our algorithm could be deployed in such environments, too, especially because it has faster running times than their system.

Bibliography

Chawathe, S. S. (2007). Segment-based map matching. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*.

Chen, D., Driemel, A., Guibas, L. J., Nguyen, A., and Wenk, C. (2011). Approximate map matching with respect to the Fréchet distance. In *Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX 2011)*, pages 75–83. SIAM.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 3rd edition.

Dalumpines, R. and Scott, D. M. (2011). GIS-based map-matching: Development and demonstration of a postprocessing map-matching algorithm for transportation research. In *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography, pages 101–120. Springer, Berlin, Germany.

Eisner, J., Funke, S., Herbst, A., Spillner, A., and Storandt, S. (2011). Algorithms for matching and predicting trajectories. In *Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX 2011)*, pages 84–95. SIAM.

Haunert, J.-H. and Sering, L. (2011). Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of Information Visualization 2011)*, 17(12):2555–2562.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., and Huang, Y. (2009). Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09)*, pages 352–361, New York, NY, USA. ACM.

Marchal, F., Hackney, J., and Axhausen, K. W. (2005). Efficient map matching of large Global Positioning System data sets: Tests on speed-monitoring experiment in Zürich. *Transportation Research Record*, 1935:93–100.

Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09)*, pages 336–343, New York, NY, USA. ACM.

Pang, L. X., Chawla, S., Liu, W., and Zheng, Y. (2011). On mining anomalous patterns in road traffic streams. In *Advanced Data Mining and Applications – 7th International Conference (ADMA 2011)*, pages 237–251. Springer.

Pereira, F. C., Costa, H., and Pereira, N. M. (2009). An off-line map-matching algorithm for incomplete map databases. In *European Transport Research Review*, pages 107–124. Springer.

Pyo, J.-s., Shin, D.-h., and Sung, T.-k. (2001). Development of a map matching method using the multiple hypothesis technique. *IEEE Intelligent Transportation Systems Proceedings (ITSC 2001)*, pages 23–27.

Quddus, M. A., Ochieng, W. Y., and Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312 – 328.

Wenk, C., Salas, R., and Pfoser, D. (2006). Addressing the need for map-matching speed: Localizing global curve-matching algorithms. In *18th International Conference on Scientific and Statistical Database Management (SSDBM 2006): 379-388, Vienna, Austria.*, pages 379–388.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen Hilfsmittel und Quellen als die angegebenen benutzt habe. Weiterhin versichere ich, die Arbeit weder bisher noch gleichzeitig einer anderen Prüfungsbehörde vorgelegt zu haben.

Würzburg, den ____________________, ____

(Benedikt Budig)