


Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT)

Report**Author(s):**

Schüssler, Nadine; Axhausen, Kay W. 

Publication date:

2009-10

Permanent link:

<https://doi.org/10.3929/ethz-b-000019956>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

Arbeitsberichte Verkehrs- und Raumplanung 568

Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT)

Nadine Schuessler

Kay W. Axhausen

Working paper

Transport and Spatial Planning

568

October 2009

Working paper

568

Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT)

Nadine Schuessler

IVT

ETH Zurich

CH-8093 Zurich

phone: +41-44-633 30 85

fax: +41-44-633 10 57

schuessler@ivt.baug.ethz.ch

Kay W. Axhausen

IVT

ETH Zurich

CH-8093 Zurich

phone: +41-44-633 39 43

fax: +41-44-633 10 57

axhausen@ivt.baug.ethz.ch

October 2009

Abstract

In recent years, data from GPS-based surveys has become increasingly important since transport modellers benefit from more accurate and reliable information about times, geographic locations, and routes. At the same time, participants' burden is reduced substantially if the GPS data collection does not involve time-consuming questions to derive additional information. However, without respondent provided information, extensive data processing is required to derive results that can be used for analysis and model estimation. Since the first GPS survey (Wagner, 1997), one of the key postprocessing steps is the map-matching, i.e. the association of the GPS points to the links of a network, in order to establish the routes travelled by the survey participants.

The algorithm proposed in this paper is an adaptation of the algorithm proposed by Marchal *et al.* (2005). Like the algorithm by Marchal *et al.* (2005), it is designed to match large-scale GPS data sets on a high-resolution navigation network in an acceptable computation time. This paper describes the implementation of the algorithm after a short overview of the existing literature. Afterwards, the performance of the algorithm is evaluated both in terms of accuracy and computational efficiency for about 36,000 car trips from 2,434 persons living in and around Zurich, Switzerland, which are mapped on the Swiss Navteq network, a high resolution navigation network covering all regions of Switzerland and containing 408,636 nodes and 882,120 unidirectional links.

Keywords

GPS data, map-matching

Preferred citation style

Schuessler, N. and K.W. Axhausen (2009) Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT), *Working paper*, 568, institute for transport planning and system (IVT), ETH Zurich, Zurich.

1 Introduction

In recent years, data from GPS-based surveys has become increasingly important since transport modellers benefit from more accurate and reliable information about times, geographic locations, and routes. At the same time, participants' burden is reduced substantially if the GPS data collection does not involve time-consuming questions to derive additional information. However, without respondent provided information, extensive data processing is required to derive results that can be used for analysis and model estimation. Beginning with the first GPS survey (Wagner, 1997), one of the key postprocessing steps is the map-matching, i.e. the association of the GPS points to the links of a network, in order to establish the routes travelled by the survey participants.

State of the art map-matching algorithms have to be accurate as well as computationally efficient. The first algorithms focussed more on accuracy and consistency of the derived routes, since the survey samples were still rather small. Accordingly, most reviews of map-matching algorithms, (e.g. White *et al.*, 2000; Greenfeld, 2002; Quddus *et al.*, 2007) mean accuracy in terms of percentage of correctly identified links when they talk about the performance of the algorithm. However, with the increasing use of GPS devices in large-scale transport studies, the need for computational speed grows. This need is further amplified by the augmented use of high-resolution navigation networks, which are essential for an accurate identification of the chosen routes. Yet, only a few authors (Nielsen *et al.*, 2004; Marchal *et al.*, 2005) have addressed the issue of performance in the sense of computational efficiency so far.

The algorithm proposed in this paper is an adaptation of the algorithm proposed by Marchal *et al.* (2005). Like the algorithm by Marchal *et al.* (2005), it is designed to match large-scale GPS data sets on a high-resolution navigation network within acceptable computation time. This paper describes the implementation of the algorithm after a short overview of the existing literature. Afterwards, the performance of the algorithm is evaluated both in terms of accuracy and computational efficiency for about 36,000 car trips from 2,434 persons living in and around Zurich, Switzerland, which are mapped on the Swiss Navteq network, a high resolution navigation network covering all regions of Switzerland and containing 408,636 nodes and 882,120 unidirectional links. The paper closes with conclusions and an outlook on future work.

2 Related work

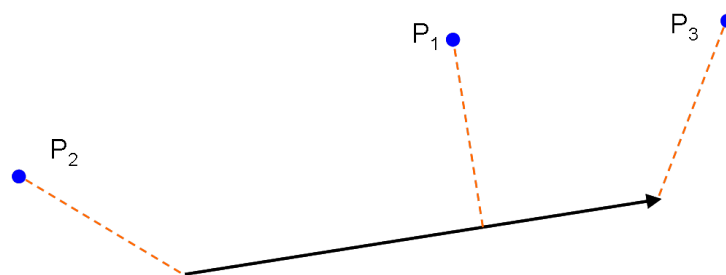
With regard to their underlying approach, map-matching procedures can best be classified into three categories:

- Geometric procedures
- Topological procedures, and
- Advanced procedures.

The most basic approaches are the geometric procedures, that only take into account the distance between the GPS points and certain network elements. Frequently used examples are the search for the nearest node or the nearest link. (White *et al.*, 2000) extended the nearest link search by comparing the heading of the GPS points with the heading of the links in question. And they also proposed the so-called curve to curve matching, where every two GPS points are connected by a line, i.e. curve and then the distance between this curve and the surrounding links is minimised. The main shortcoming of all geometric procedures is that they ignore the sequence of the GPS points over time as well as the connectivity of the network links. Therefore, it is possible that the derived route oscillates back and forth between links. In addition, they are very dependent on a correct network coding and are rather sensitive to outliers. Regarding computational efficiency, as Nielsen *et al.* (2004) demonstrated, only the nearest node search is in any way fast enough for the problem at hand. Yet, the error rate of the matching of nearly 3,000 trips from the Copenhagen area on an associated 300,000 link network was with 33% too high for any real life application.

In contrast to the geometric procedures, topological procedures not only account for the distance between the GPS points and the network elements, but also for the sequence or history of GPS points and the connectivity of network elements. Most procedures work in two steps. First, an the initial node or link is found using geometric approaches. Afterwards, the route is developed by choosing a link out of the set of candidate links. Usually this set consists of the last matched link and the links succeeding that link though some authors extend it for all links preceding the last matched link (Chung and Shalaby, 2005) or for the links succeeding the succeeding links (Greenfeld, 2002). For the choice of the most likely link out of the set of candidate links, different criteria can be employed. The most common one is the perpendicular distance between the GPS point and the link. The perpendicular distance equals the minimum of the euclidean distance between the GPS point and its orthogonal projection on the link, the euclidean to the start node and the euclidean distance to the end node. In the following, the distance measure which out of these three measures delivers the minimum value is called the relevant perpendicular distance. In Figure 1, the relevant perpendicular distance for GPS point

Figure 1: Derivation of the perpendicular distance



P_1 is the distance to its orthogonal projection on the link, for P_2 the distance to the start node and for P_3 the distance to the end node.

Other criteria are the heading of the GPS point compared to the one of the link (e.g. Greenfeld, 2002; Chung and Shalaby, 2005; Velaga *et al.*, 2009), the position of the point relative to the link, which is derived from the angle between the link and the line between the start node of the link and the GPS point (Quddus *et al.*, 2003) or if and at what angle the link and the line between the GPS point and its predecessor intersect (Greenfeld, 2002). If more than one criterion is used, they are usually weighted against each other with parameters determined using expert knowledge or a calibration procedure (e.g Velaga *et al.*, 2009). Additional features include treatment of outliers (Greenfeld, 2002) or a post-processing either in terms of a ex-post elimination of unlikely links based on the percentage of the link that is covered by the points (Chung and Shalaby, 2005) or a mode specific filling of gaps between links (Tsui and Shalaby, 2006). The topological approaches outperform the geometric ones in terms of computational speed as well as route accuracy. They are faster because for each GPS point, except the first one, only a very limited number of links has to be evaluated and they are more accurate because they take into account the whole sequence of points and are less sensitive to measurement errors and outliers. However, there is still room for improvement in both respects. In particular, there is no fall-back solution in case the initial link determination failed which could lead to a completely wrong route. Moreover, there is the issue of parallel streets, that are running closely next to each other. Once the algorithm chooses the wrong route, it is hardly able to correct such a mistake (Quddus *et al.*, 2003; Nielsen *et al.*, 2004).

To overcome these problems, in recent years more advanced approaches have been proposed. Advanced approaches do not only take into account the whole sequence of GPS points and the network topology, but also the fact that, due to errors in the GPS measurement as well as the network coding, the nearest link or node is not necessarily the right one. A lot of different procedures have been proposed of which a small selection is presented here.

A straight forward approach to account for GPS measurement errors is the construction of error or confidence regions around the GPS points (e.g. Doherty *et al.*, 2001; Ochieng *et al.*, 2004; Velaga *et al.*, 2009). The size of the error region should be derived from the error variances (Quddus *et al.*, 2007; Ochieng *et al.*, 2004). Then all links within this error region are evaluated based on distance, heading, connectivity to the previously matched link, and sometimes speed (Ochieng *et al.*, 2004). Quddus *et al.* (2006) extended Ochieng *et al.* (2004)'s error region approach by a fuzzy logic inference systems. The fuzzy rules consider different criteria such as distance, heading, speed, the quality of the position solution via the HDOP value, link connectivity and the position of the GPS point relative to the candidate link. Thereby, separate rules apply for the initial link search and the subsequent path development. Another example for an extension of the error region concept is the map-matching based on conditional random fields introduced by Liao *et al.* (2007). They use, however, only a reduced set of evaluation criteria. Only distance and connectivity are taken into account.

An approach without the use of error regions was proposed by Nielsen *et al.* (2004). Their algorithm resembles the Dijkstra algorithm for the single-source shortest path problem. The start node is determined in a not further described preprocessing step. Starting from there, the route is developed by adding the end nodes of all outgoing links of the current node to the set of nodes to be evaluated. The next node to be evaluated is then the node that could be reached in the shortest amount of time starting from the last node of the route so far. The score of each node is calculated based on the perpendicular distance between the GPS points and the links they are assigned to and the distance between the GPS points and the start node of the link they are assigned to. Yet, how the GPS points are assigned to the links cannot be derived from the paper even though this is a crucial aspect of any map-matching approach since a wrong assignment can lead to greatly biased scores. Another problem with this algorithm is that it cannot not guaranty to find the optimal solution, as the Dijkstra algorithm would, because the route development criterion differs from the scoring function. Moreover, even though Nielsen *et al.* (2004) claim that the algorithm is highly efficient, the risk of evaluating the whole network should not be neglected because the stop criterion is simply that the set of nodes to be evaluated should be empty.

However, the main disadvantage of all the approaches summarised hitherto is that after the evaluation of each GPS point only one link or route respectively remains. The next GPS point is then matched based on the assumption that the last GPS point was matched correctly. Accordingly, if the map-matching identifies just one link wrongly, the probability is quite high, that the remaining route will be wrong as well. Considering the frequency and magnitude of GPS measurement errors, especially at the beginning of a trip, this is not recommendable. Therefore, Pyo *et al.* (2001) introduced the use of the Multiple Hypothesis Technique (MHT) for map-matching. This means that, following the sequence of GPS points, several route candidates are kept in memory, developed and assigned a score. The best candidate is usually

only determined when the end of the GPS sequence is reached. Two different implementations of the MHT have been presented so far. While Pyo *et al.* (2001) employed error regions and extended the existing paths by the links within the error region, Marchal *et al.* (2005) adopt a topological search algorithm. The initial set of links is determined by searching the m nodes closest to the first GPS point and creating a single-link path for each of their incident links. Afterwards it is checked whether the current GPS point can be matched to the last link of the route or if a junction was reached. If a junction was reached, a new route candidate is created for each link succeeding the current link. The new link is added at the end of the route. Each of the new route candidates is scored and saved in the set of route candidates. Because this topological search inherently ensures for link connectivity, the scoring function is rather lean in summarising the perpendicular distances between the GPS points and the links the points are assigned to. In a later version of the algorithm, they extended the scoring function to account for the shape of the link by comparing the distance between two consecutive GPS points with the distance between their projections on said link. Compared to this, the list of criteria in Pyo *et al.* (2001)'s scoring function is rather extensive. They use distance, heading, and, as a measure of connectivity, the number of links necessary to get from the last matched link to the link in question. In both approaches it is necessary, to limit the number of candidates kept in memory to ensure computational feasibility. Marchal *et al.* (2005) simply define a maximum number of candidates. If the size of the candidate set exceeds this number after all candidates have been processed for the current GPS point, the route candidates with the lowest scores are cut. Pyo *et al.* (2001) do not define their pruning criteria on the absolute number of alternatives. They cut candidates based on the score of the candidate relative to the score of the best candidate so far. Accordingly, their probability to maintain a rather large set of candidates is high, especially in dense urban areas. This leads, combined with the need to establish an error region around each GPS record, to relatively high computational cost whereas Marchal *et al.* (2005) could show that their algorithm is accurate as well as fast enough for the map-matching of large amounts of GPS data on a high-resolution navigation network. Therefore, the algorithm presented by Marchal *et al.* (2005) is chosen as the basis for the algorithm proposed in this paper.

3 Implementation

The map-matching presented in this paper is the fifth step of a framework that derives mode specific route choice observations from GPS records. The first four steps, data filtering, detection of trips and activities, mode stage determination, and mode identification, have already been presented in Schüssler and Axhausen (forthcoming). All procedures are implemented in JAVA. While the first four steps did not employ any other information but the GPS points, the map-matching by nature requires the use of a network. In order to profit from existing infrastructure, the map-matching uses several elements of MATSim (MATSim-T, 2008). This includes the representation of the network as well as other helpful methods such as the calculation of distances between GPS points and certain network elements, or the search for all nodes within a certain radius.

The map-matching procedure itself consists of five consecutive steps:

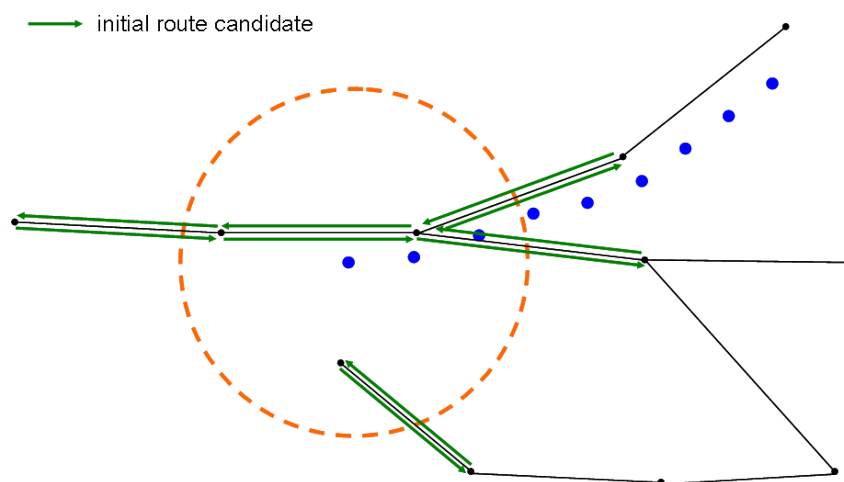
1. Trip segmentation
2. Determination of initial route candidates
3. Route development
4. Selection of the most likely candidate
5. Route Filtering
6. Treatment of the gaps between trip segments

The first four steps are based on the work of, Marchal *et al.* (2005). However, several aspects have been adapted to improve the accuracy of map-matching results.

First, the GPS points of each trip is subdivided into continuous segments. Therefore, spatial and temporal gaps in the sequence of GPS points are detected. If the time gap between two subsequent GPS points is longer than 120 seconds or the distance is larger than 500 metres, the map-matching would cannot deliver trustworthy results. The sequences of GPS points before and after each gap which contain at least 10 points are stored as separate trip segments. Shorter segments are neglected because they are likely to deliver unreliable outcomes because the GPS traces tend to be more noisy immediately after a recording gap. The steps two to four are then executed for each segment individually, before the segments are joined to complete trips in step five.

Second, the initial set of single-link route candidates is derived. As depicted in Figure 2, all nodes within the a radius of 750 metres around the first GPS point are found. Afterwards, a single-link route candidate is created for each link connected to at least one of these nodes. For

Figure 2: Derivation of the initial route candidates



each route candidate, the first GPS point is assigned to the one link of the route, the score is calculated and it is stored in the list of current route candidates. In case the number of route candidates found this way is lower than 25, the search radius is increased by 100 metres and the whole process is run again. This is repeated until there are at least 25 routes candidates.

The route development process is illustrated in Figure 3. For each GPS point, all route candidates remaining from the previous iteration are evaluated. If the route candidate contains only one link, it is first checked if the start node of this link was reached, i.e. the distance to the start node is the relevant perpendicular distance. In this case, the route candidate is discarded because apparently the GPS points are running in the opposite direction. Subsequently, it is examined if the point can be matched to the last link of the route candidate or if the end of this link was reached. Given that the GPS point can be assigned to the link, this assignment is stored and the score of the route candidate is recalculated. If the end of the link was reached, for each link succeeding the last link a copy of the route candidate is created, the succeeding link is appended to the route, the GPS point assigned to that link, and the score is recalculated. Then old route candidate is removed and the newly created route candidates are added to the set of route candidates. However, before a route candidate is added, it is verified, that the new route is a valid route candidate. A route is only valid if it is unique, i.e. the exact same sequence of links is not already included in the set of route candidates, if the new link does not head back to the start node of the last link, i.e. no u-turn, and if the route does not contain any link twice. Cycles in terms of using the same node twice are allowed since in one-way street systems with turn restrictions it is sometimes necessary, to pass the same crossing twice.

A crucial aspect of this procedure is the way how it is determined if the end of a link was reached or not. Even though this is an important issue for many map-matching procedures, only Ochieng *et al.* (2004) and Marchal *et al.* (2005) have discussed their way of doing this in

Figure 3: Flow chart of the route development for a trip segment

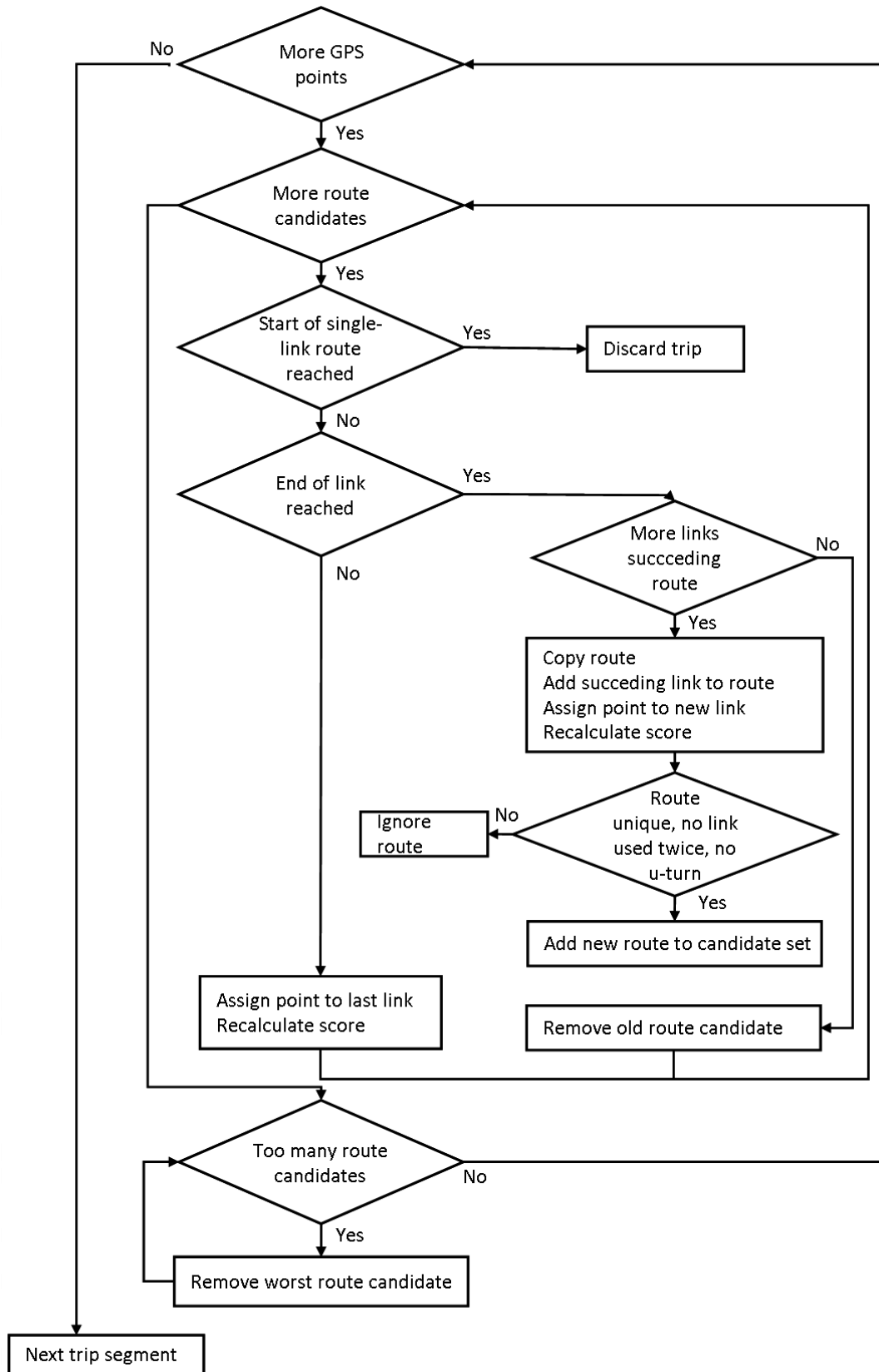
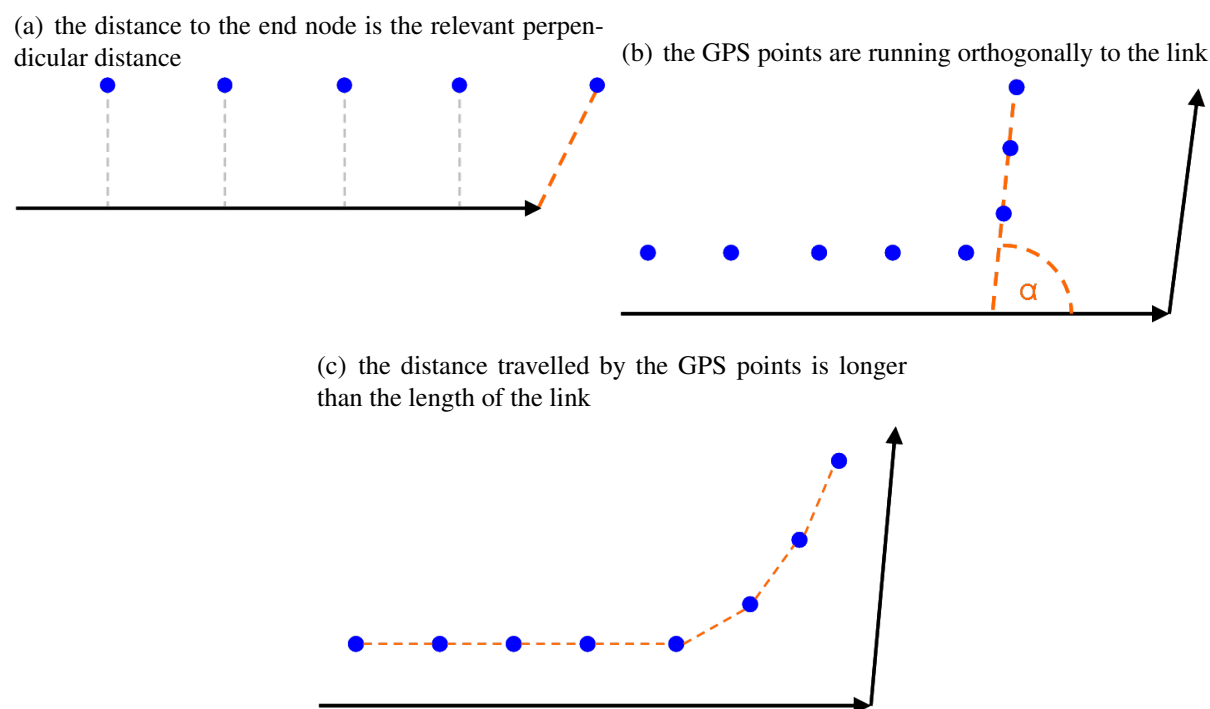


Figure 4: The end of a link is reached if...



more detail. Ochieng *et al.* (2004) use the relative position of the GPS point to the link as well as heading changes that exceed a certain threshold to as end of link criteria, whereas Marchal *et al.* (2005) compare the distance travelled by the GPS points with the length of the link. If the GPS points cover at least a certain percentage of the link, the end of the link is reached. As shown in Figure 4, the proposed algorithm uses a combination of these criteria. The GPS point has reached the end of the current link if the distance to the end node is the relevant perpendicular distance or if the GPS points are running in an orthogonal or opposite direction, i.e. at an angle bigger than 85° , to the link or if the distance travelled by the GPS points is longer than the length of the link.

When all route candidates have been processed for the current GPS point, the number of remaining route candidates is compared to the maximum number of route candidates N_{max} defined by the analyst. If there are too many route candidates, the ones with the worst score are removed until the size of the route candidate set equals N_{max} . Thereby, N_{max} has to be chosen carefully. On the one hand, a high N_{max} ensures that the route candidate set contains the actually chosen route even if it obtains lower scores at the beginning or in between. On the other hand, a low N_{max} improves computational performance. Several values for N_{max} have been tested in the development of the map-matching algorithm. Their implications are discussed in Section 4.

Another decisive factor for the success of a map-matching procedure is the way the score of each path candidate is calculated. As explained in Section 2, several criteria have been used so far. The most popular ones were the perpendicular distance between the GPS points and the links they were assigned to, the heading of two successive GPS points compared to the heading of the link, the speed of the GPS points, and the connectivity of the link to the preceding link. In the proposed algorithm the score is calculated as specified in Equation 1:

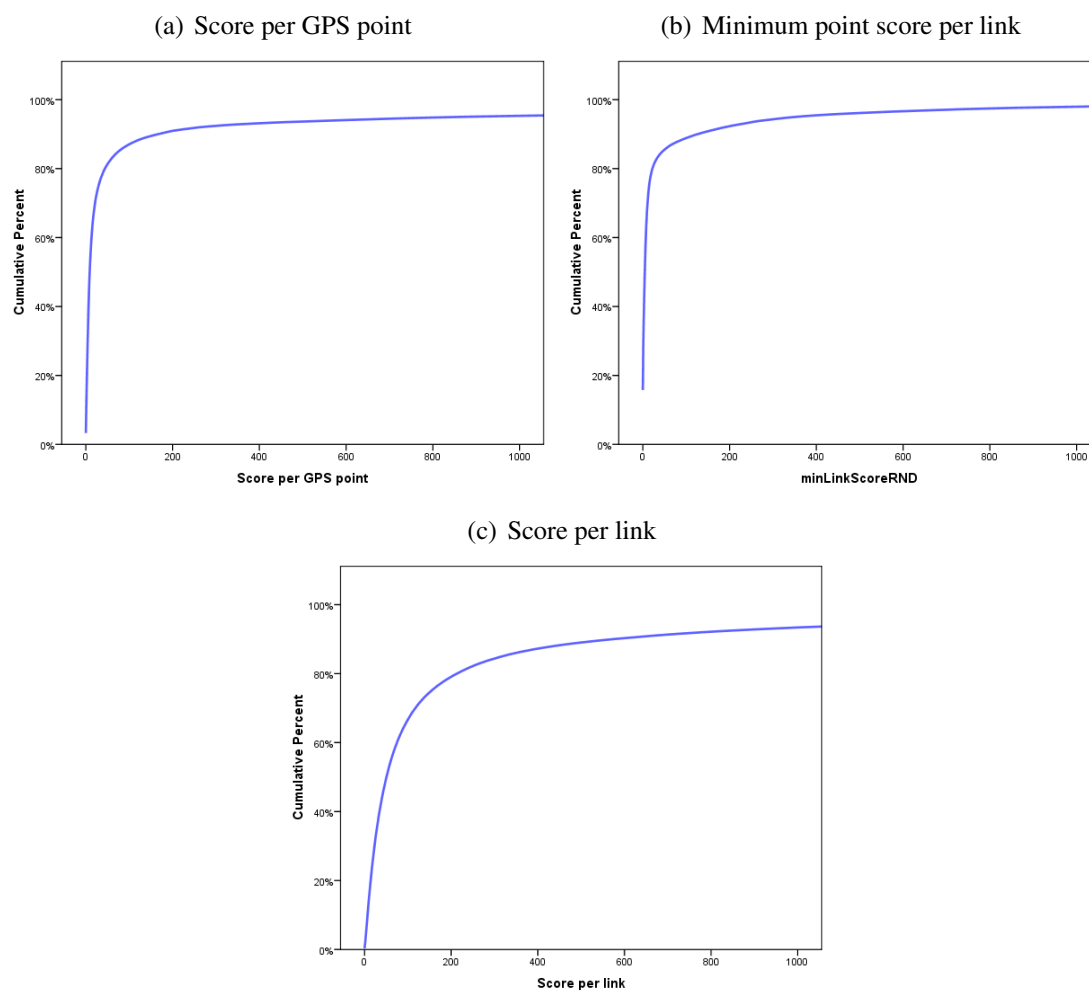
$$SC_{path} = \sum_P \sum_L (d(p_i, l_j) \delta_{ij} + (v(p_i) - v_{ff}(l_j))^2 \gamma_{ij}) \quad (1)$$

where $L\{l_1, l_2, \dots, l_t\}$ is the set of links composing the path, $P\{p_1, p_2, \dots, p_p\}$ the set of GPS points and δ_{ij} equals 1 if p_i is assigned to l_j , and = 0 otherwise. In addition, $v(p_i)$ is the GPS speed at point p_i , $v_{ff}(l_j)$ is the free-flow speed on link l_j and γ_{ij} equals 1 if $v(p_i) > v_{ff}(l_j)$ and 0 otherwise. The scoring function only considers the sum of the perpendicular distances between the GPS points and the links they are assigned to and a speed malus in case the speed of the GPS points exceeds the free-flow speed on the link in question. The malus equals the squared speed difference. This way, higher speed differences are punished much stronger. Other frequently employed criteria were not taken into account for different reasons. A score for connectivity, for example, was obsolete due to the design of the algorithm and the heading was found to be too unreliable.

In the end, after processing all GPS points of the trip segment, the route candidate with the lowest score is determined. Since it is most likely, that this route is the one actually travelled by the participant. However, before assigning the route to the trip segment, a final validity check is carried out. If the route contains only one link, it is not assigned to the trip segment and the trip segment itself is neglected in the fifth step, the joining of the trip segments to complete trips. A matching of GPS points to just one link can be terribly misleading, especially in directed networks. In such a network, a two-way road between two nodes is represented by two separate links, which would both obtain the same score. The selection of one of these single-link route would be totally arbitrary and might have bad effects on the subsequent joining of the trip segments.

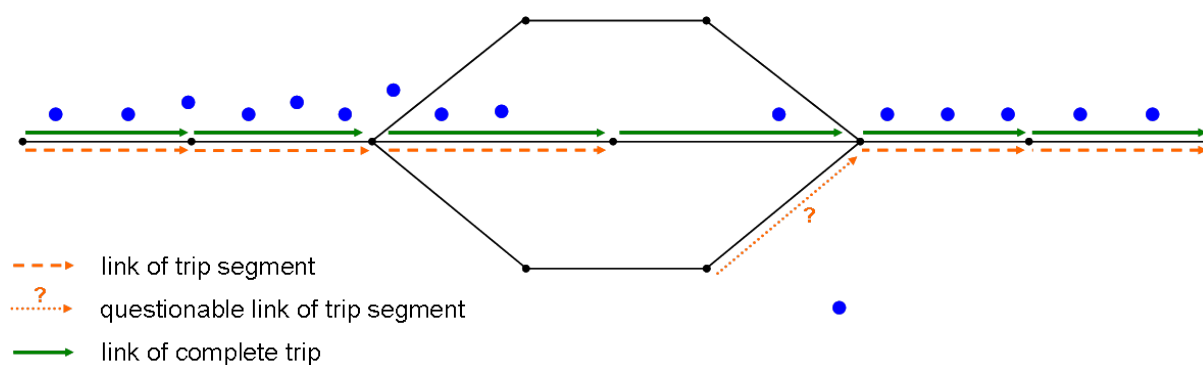
An important requirement for each map-matching algorithm to work properly is a correct, consistent and complete representation of the real network by the network used for the map-matching. Unfortunately, hardly any network currently available can guarantee this requirement. Especially missing network links lead to problems for the map-matching algorithm because the real route taken by the traveller can not be reproduced. Instead, either no route is found or a route that works its way around the missing link. This route does not reflect the actual behaviour and it is up to the researcher to decide how to deal with it. Since the applications using the results of this map-matching procedure require that the matched routes are reliably reflecting the actual route choice behaviour, it was decided to filter these routes based on the

Figure 5: Cumulative distributions of data quality measures



assumption that there are no systematic errors in the network coding but that missing links are randomly distributed throughout the entire network. The filtering mechanisms exploit the fact that routes which include a detour around a missing link are characterised by high score values. Thus, the filtering was implemented in two different ways. First, the average score per GPS point is evaluated. If it exceeds a value of 75, the route is removed. Second, the minimum point score for each link is determined. The point score is the contribution of GPS point to the score of the link it is assigned to. If the minimum point score for a link equals 100, the link is marked as *odd*. If a route contains more than 3 odd links, it is discarded. The filtering thresholds were derived from the distribution of the score per GPS point and the distribution of the minimum point score per link depicted in Figure 5 (a) and (b). It can be seen that nearly 85% of the GPS points have a score of less than 75 and nearly 90% of the links have a minimum point score of less than 100. Thus, routes with an average point score of higher than 75 or containing more than 3 links with a minimum point score of 100 are at least questionable and should be

Figure 6: Joining trip segments to complete trips



ignored in subsequent applications. This ensures a high quality and reliability of the remaining routes. However, the characteristics of the trips for which the routes are discarded are stored in a separate file to allow for a check of the representativeness of the resulting routes.

In order to obtain complete trips, the gaps between the trip segments have to be filled. Depending on the quality of the GPS points and the density of the network the length of the gaps and the number of links required to fill them can vary significantly. In addition, no information is available on where the participant actually travelled during the period of signal loss. Thus, a consistent assumption has to be made on how a gap between two subsequent trip segments should be filled. Because different assumptions will have different impacts on the resulting routes, the employed assumption should always be considered in later analysis. Two basic assumptions have been presented in the literature. Tsui and Shalaby (2006) claim that people try to avoid turning maneuvers as much as possible. Therefore, they use the heading of the links in the gap as main criterion to close the gap. Whereas others employ a shortest path search to fill gaps in route observations from GPS records (Pasquier *et al.*, 2008) or participant descriptions (Ramming, 2002).

The procedure applied here is illustrated in Figure 6. The procedure combines a shortest path search with a treatment of low quality map-matching results. Especially at the start and the end of a trip segment the map-matching can be less reliably due to sparse or noisy GPS points. The best measure to evaluate the reliability of the map-matching is the score assigned to the individual links. If it is higher than 50, the link is marked as *questionable*. As can be seen from the distribution of link scores in Figure 5 (c), this threshold is rather rigorous compared to the filtering threshold. It was intended since the threshold does not automatically lead to a deletion of the trip but instead to a treatment with care. If, however, all links of a trip segment are marked as questionable, something is probably not right with the entire segment and the whole trip is discarded. Otherwise, the shortest path search is extended to cover not only the last node of the segment preceding and the first node of the segment succeeding the gap but also

the start nodes of all questionable links after the last trustworthy link at the end of the segment preceding the gap and the end nodes of all questionable links before the last trustworthy link at the start of the segment succeeding the gap. After the shortest paths between all potential start and end nodes have been calculated, the shortest of these shortest paths is determined. Subsequently, two concluding quality tests are performed. First, all trips where at least one trip segment is entirely included in one of the shortest paths filling the gaps are discarded. They contain unrealistic circles and usually occur due to wrong map-matching around the entry and exit ramps of motorways. Second, trips which contain shortest paths that cannot be travelled in the time available according to the GPS points, are dropped as well. Finally, for the remaining trips, the links of the trip segments and the shortest paths are joined to form the route which was most likely taken by the participant and can now be used for further analysis and choice modelling.

4 Evaluation of the resulting routes and the computational efficiency

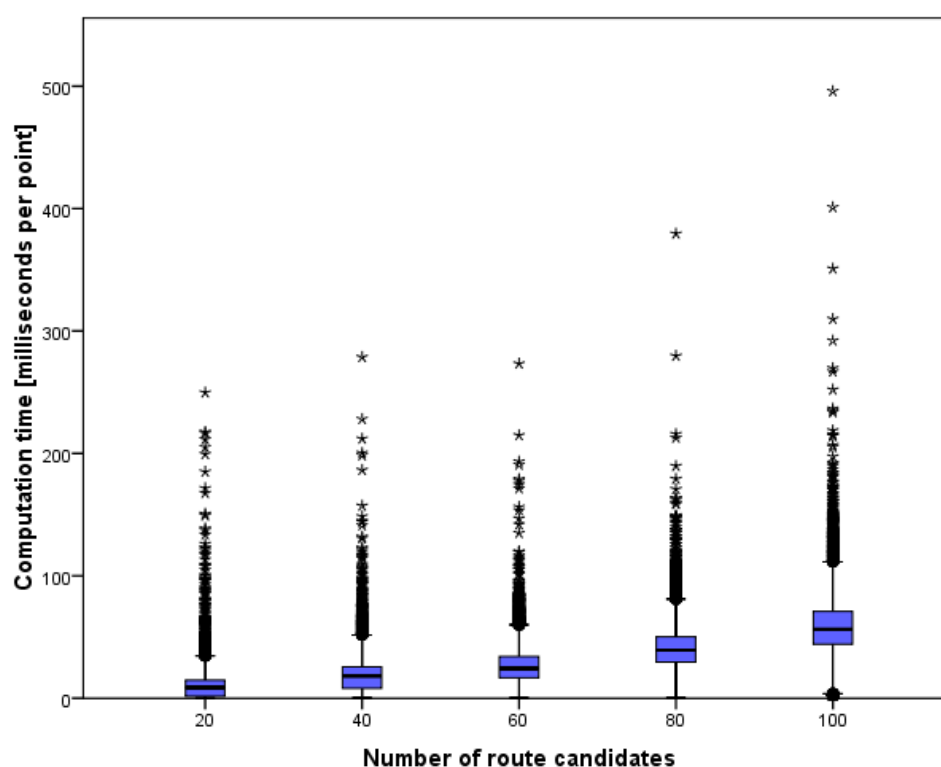
All tests described in this section were run on systems having two Dual-Core AMD Opteron Processors 2222 running at 3 GHz. Memory was connected through a front side bus clocked at 1,000 MHz. As the code was not multi-threaded, only one of the CPU cores was actually used by the tests. The Java 1.6 program runs on 1 CPU using 2 GB allocated memory. The test data set contained about 4.1 million GPS points recorded for 250 persons and 1776 person-days. Since the map-matching is embedded in a bigger framework for the processing of GPS raw data (Schüssler and Axhausen, forthcoming), the 4.1 million raw data points were first filtered, subdivided into trips and activities and assigned a mode. Subsequently, 3932 car stages comprising 2.4 million GPS points were matched to the Swiss Navteq network, a high-resolution navigation network covering all regions of Switzerland and containing 408,636 nodes and 882,120 unidirectional links. In order to evaluate the computation time of the map-matching alone, the computation time for the GPS processing without map-matching was subtracted from the total computation time of a test run.

One of the strongest drivers for the computational performance of the map-matching is the maximum number of candidate paths that is maintained. Therefore, 5 test runs with 20, 40, 60, 80 and 100 candidates respectively were executed. In the following, the results of these test runs are evaluated with respect to computation time, number of properly matched trips and the distribution of scores. The aim is to determine the optimal number of candidates that produces reliably results within a reasonable computation time.

Figure 7 depicts the distribution of computation time per GPS point depending on the maximum number of candidate paths for the map-matching itself, i.e. the steps 2-4 of the procedure described in section 3. As expected, the computation time increases linearly with the number of candidate paths. The total computation time for map-matching the sample of 250 OD pairs including all steps outlined in section 3 ranges between 393 minutes for 20 candidates and 3022 minutes for 100 candidates. This translates to a mean computation time per route between 6 and 46 seconds and a mean computation time per GPS point between 0.010 seconds and 0.075 seconds. Since the computation time increases substantially with the number of candidate paths this number should be kept as low as possible.

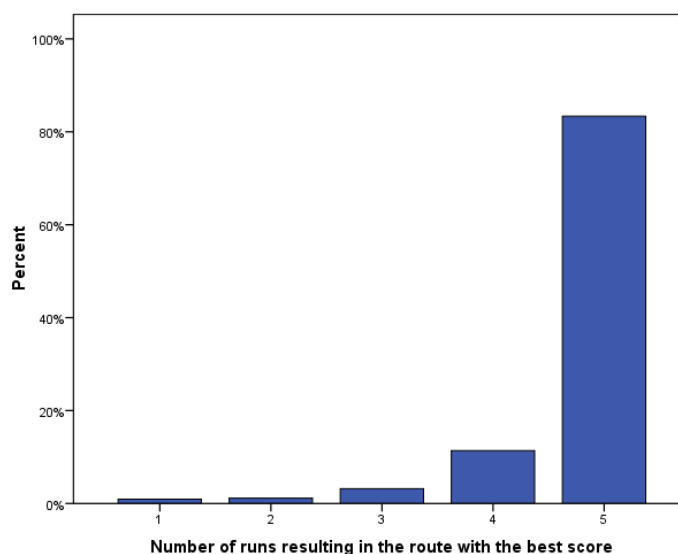
The number of routes that could be matched successfully varies only slightly with the number of candidates between 2065 and 2088 with the strongest increase between 20 and 40 candidates from 2065 to 2084 successfully matched routes. This means that 40 candidate paths would deliver the best trade-off between computation time and map-matching success. The more important result, however, is that compared to the total number of car routes which should have been matched, the number of successfully matched routes is extremely low. The question is

Figure 7: Distribution of computation time per GPS point depending on the maximum number of candidate paths



why so many routes could not have been matched. A first manual check of the data for a few persons revealed that most of the problems were network related. In more than 40% of the cases the map-matching was unsuccessful because links were missing in the network. Some of these links were used by several persons or repeatedly by the same person, making a network coding error very probable. Another 40% of the unsuccessful matchings occurred because part of the trip or the whole trip had been travelled off-network. The complete off-network trips were usually trips outside Switzerland. Since both error sources concern the quality and the range of the network and not the map-matching algorithm, solutions have to be found on the data side. In the meantime, the representativeness of the successfully matched routes can be derived from the characteristics of all trips stored in a separate file as described in the previous section. An issue that should be solved within the map-matching is the third most common reason for an unsuccessful map-matching: U-turns and even double u-turns, i.e. driving in circles. U-turn and circles account for a little bit less than 10% of the unsuccessful matchings. Since they are an aspect of real life transport behaviour, the map-matching algorithm should be able to handle them. This is, however, a future research issue. In addition, a more detailed analysis of the remaining unsuccessful map-matchings is necessary. For them no apparent error source could be detected in this preliminary analysis. They might reveal the potential for improvements.

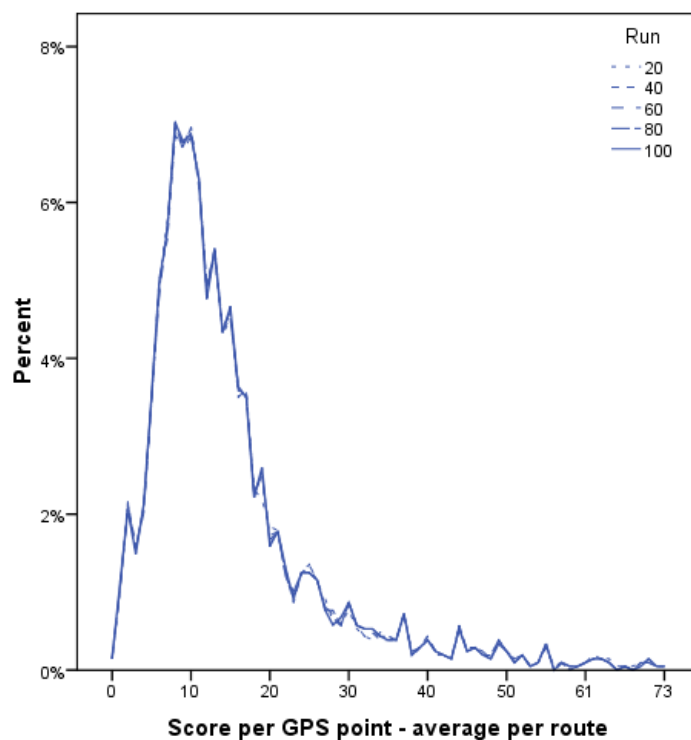
Figure 8: Number of runs with same output as least cost route



Continuing the comparison between the runs with different numbers of candidates paths, Figure 8 reveals that there are only few deviations in the route matchings between the different runs. 83% of the routes are identically matched in each run and for further 11% of the routes the matching is identical in 4 of 5 runs. This also leads to rather small variations in the average scores per route. The average score per route ranges between 6616 for 20 candidates and 6586 for 100 candidates. There is a trend to lower scores with an increasing number of candidates indicating a better matching of the routes. However, the differences are only marginal and the distributions of scores, as presented in Figure 9 for the scores per GPS point, are very similar for all runs. Figure 9 also reveals that for the majority of routes the average score per GPS point, which is mainly derived from the distance between the points and the route, is below 20 with an average of about 15 and a median of about 11. Considering that the accuracy of the GPS points under ideal conditions lies between 5 and 10 metres and that especially longer links are coded in a way that their position does only overlay at the beginning and the end with the position of the underlying road, this is a very good result for the map-matching.

Summarising the results discussed in this section with regard to the question of which would be the ideal maximum number of candidates path, one can say that a higher number of candidate paths does only slightly increase the quality of the results while leading to a substantial increase in computation time. Thus, a maximum number of candidates paths between 20 and 40 can be recommended. This corresponds with the findings by Marchal *et al.* (2005) who advocated a maximum number of candidate paths of 30 for their algorithm.

Figure 9: Score per GPS point - averaged per route



5 Conclusion and Outlook

The map-matching algorithm presented in this paper is able to match a large-scale GPS data set to a high-resolution navigation network in an acceptable computation time. Manual checks of the matched roads as well as the analysis in Section 4 suggest that the map-matching delivers reliable the routes actually taken by the participants if the underlying network is correct, consistent and complete. Trips for which a road used by the traveller is missing in the network representation or which take place (partly) off the network are filtered because for them no accurate representation of the actual travel behaviour is possible. These trips might, however, be used in the future to correct and complement the network.

Employing the Multiple Hypothesis Technique has several advantages. The most important one is that it makes the map-matching results more robust against erroneous map-matching due to noisy GPS paper or a simplified network coding where the shape of the link does not exactly follow the course of the actual road. Another advantage is that it results in more than one route candidate. These route candidates might allow to approximate the uncertainty connected with the outcomes of any map-matching algorithm. How this can be done is a future research issue. Initial ideas on how to explicitly model the likelihood of potential routes without employing a map-matching procedure has been presented by Bierlaire *et al.* (2009). This

interesting approach is however not yet ready for real size applications.

The most important future research issue is the treatment of u-turns in the GPS data. So far, trips that contain u-turns are also filtered by design. This is a shortcoming of the algorithm and should be resolved in the future. In the course of this work also those trips that have been filtered although there was no apparent reason should be analysed in order to improve the success rate of the algorithm. In the meantime, the representativeness of the successfully matched routes can be derived from the characteristics of all trips stored in a separate file as described in the previous section.

Although the computation time of the algorithm is already acceptable, there is still some room for improvement. A first step would be to decrease number of path candidates during the development of a path depending on the differences between the scores. In the beginning of the detection, a wide exploration and a large number of candidates is necessary to ensure that the right route is contained in the candidate set. But while the path development progresses the score differences between the right route and the other candidates increases considerable so that it might be sufficient to reduce the candidate set to 10-20 alternatives.

Another worthwhile investigation is to test the computational performance and the quality performance for different network resolutions. This algorithm was designed specifically for very high-resolution networks. The application to lower resolution networks could give some more valuable insight into the workings of the algorithm especially since it will also be used for the map-matching of public transport trips to the much sparser public transport network.

In terms of the embedding of the map-matching in the bigger framework for the processing of GPS data, the next step would be a feedback of the map-matching results to the mode detection. This would improve the distinction between car trips and particularly rail-based public transport trips. So far, especially the share of rail trips resulting from the mode detection is rather low and the current hypothesis is that some of the actual rail trips are misinterpreted as car trips. A feedback from the map-matching might help to determine these trips.

6 Acknowledgements

The authors would like to thank the Swiss National Science Foundation for the funding of this work and Stefan Muff (at that time Endoxon AG) for providing the GPS data.

References

- Bierlaire, M., J. Newman and J. Chen (2009) A method of probabilistic map distribution of path likelihood, paper presented at *the 9th Swiss Transport Research Conference*, Ascona, September 2009.
- Chung, E.-H. and A. Shalaby (2005) A trip bases reconstruction tool for GPS-based personal travel surveys, *Transportation Planning and Technology*, **28** (5) 381–401.
- Doherty, S. T., C. Noel, M. E. H. Lee-Gosselin, C. Sirois, M. Ueno and F. Theberge (2001) Moving beyond observed outcomes: Integrating Global Positioning Systems and interactive computer-based travel behaviour surveys, *Transportation Research E-Circular*, **C026**, 449–466.
- Greenfeld, J. S. (2002) Matching GPS observations to locations on a digital map, paper presented at *the 81st Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2002.
- Liao, L., D. Fox and H. Kautz (2007) Extracting places and activities from GPS traces using hierarchical conditional random fields, *International Journal of Robotics Research*, **26** (1) 119–134.
- Marchal, F., J. K. Hackney and K. W. Axhausen (2005) Efficient map matching of large Global Positioning System data sets: Tests on speed-monitoring experiment in Zürich, *Transportation Research Record*, **1935**, 93–100.
- MATSim-T (2008) Multi Agent Transportation Simulation Toolkit, webpage, <http://www.matsim.org>.
- Nielsen, O. A., C. Würtz and R. M. Jørgensen (2004) Improved map-matching algorithms for GPS-data - methodology and test on data from the AKTA roadpricing experiment in Copenhagen, paper presented at *the 19th European Conference for ESRI Users*, Copenhagen, November 2004.
- Ochieng, W. Y., M. A. Quddus and R. B. Noland (2004) Map matching in complex urban road networks, *Brazilian Journal of Cartography*, **55** (2) 1–18.
- Pasquier, M., U. Hofman, F. H. Mende, M. May, D. Hecker and C. Körner (2008) Modelling and prospects of the audience measurement for outdoor advertising based on data collection using GPS devices (electronic passive measurement system), paper presented at *8th International Conference on Survey Methods in Transport*, Annecy, May 2008.
- Pyo, J.-S., D.-H. Shin and T.-K. Sung (2001) Development of a map matching method using the multiple hypothesis technique, paper presented at *Intelligent Transportation Systems Conference (ITSC)*, Oakland, August 2001.

- Quddus, M. A., W. Y. Ochieng and R. B. Noland (2007) Current map matching algorithms for transport applications: State-of-the-art and future research directions, paper presented at *the 86th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2007.
- Quddus, M. A., W. Y. Ochieng, L. Zhao and R. B. Noland (2003) A general map matching algorithm for transport telematics applications, *GPS Solutions*, **7** (3) 157–167.
- Quddus, M. A., W. Y. Ochieng, L. Zhao and R. B. Noland (2006) A high accuracy fuzzy logic-based map matching algorithm for road transport, *Journal of Intelligent Transportation Systems*, **10** (3) 103–115.
- Ramming, M. S. (2002) Network knowledge and route choice, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge.
- Schüssler, N. and K. W. Axhausen (forthcoming) Processing GPS raw data without additional information, *Transportation Research Record*.
- Tsui, S. Y. A. and A. Shalaby (2006) An enhanced system for link and mode identification for GPS-based personal travel surveys, *Transportation Research Record*, **1972**, 38–45.
- Velaga, N. R., M. A. Quddus and A. L. Bristow (2009) Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems, *Transportation Research Part C: Emerging Technologies*, **17** (6) 672–683.
- Wagner, D. P. (1997) Lexington area travel data collection test: GPS for personal travel surveys, *Final Report*, Office of Highway Policy Information and Office of Technology Applications, Federal Highway Administration, Battelle Transport Division, Columbus, September 1997.
- White, C. E., D. Bernstein and A. L. Kornhauser (2000) Some map matching algorithms for personal navigation assistants, *Transportation Research Part C: Emerging Technologies*, **8** (1–6) 91–108.