Likelihood-based offline map matching of GPS recordings using global trace information

Peer-reviewed author version

KNAPEN, Luk; BELLEMANS, Tom; JANSSENS, Davy & WETS, Geert (2018) Likelihood-based offline map matching of GPS recordings using global trace information. In: TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES, 93, p. 13-35.

DOI: 10.1016/j.trc.2018.05.014
Handle: http://hdl.handle.net/1942/26216

Likelihood-based Offline Map Matching of GPS Recordings Using Global Trace Information

Luk Knapen[a,*], Tom Bellemans[a], Davy Janssens[a], Geert Wets[a]

[a]*Hasselt University, Transportation Research Institute (IMOB) Wetenschapspark 5 bus 6 3590 Diepenbeek, Belgium*

Abstract

In batch map matching the objective is to derive from a time series of position data the sequence of road segments visited by the traveler for posterior analysis. Taking into account the limited accuracy of both the map and the measurement devices several different movements over network links may have generated the observed measurements. The set of candidate solutions can be reduced by adding assumptions about the traveller's behavior (e.g. respecting speed limits, using shortest paths, etc). The set of feasible assumptions however, is constrained by the intended posterior analysis of the link sequences produced by map matching. This paper proposes a method that only uses the spatio-temporal information contained in the input data (GPS recordings) not reduced by any additional assumption.

The method partitions the trace of GPS recordings so that all recordings in a part are chronologically consecutive and match the same set of road segments. Each such trace part leads to a collection of partial routes that can be qualified by their likelihood to have generated the trace part. Since the trace parts are chronologically ordered, an acyclic directed graph can be used to find the best chain of partial routes. It is used to enumerate candidate solutions to the map matching problem.

Qualification based on behavioral assumptions is added in a separate later stage. Separating the stages helps to make the underlying assumptions explicit and adaptable to the purpose of the map matched results. The proposed technique is a multi-hypothesis technique (MHT) that does not discard any hypothesized path until the second stage.

A road network extracted from OpenStreetMap (OSM) is used. In order to validate the method, synthetic realistic GPS traces were generated from randomly generated routes for different combinations of device accuracy and recording period. Comparing the base truth to the map matched link sequences shows that the proposed technique achieves a state of the art accuracy level.

Keywords: GPS Traces, Map Matching, Transportation modeling, Big data analysis

1. Introduction

Map matching combines a road transport network description consisting of nodes and directed links with a time series of coordinate tuples that describes the movement of a traveler. A *trace* is a chronologically ordered sequence of all GPS recordings associated with

*Corresponding author: Tel.:+32-11-269-126

Email address: luk.knapen@uhasselt.be (Luk Knapen)

a movement. The purpose is to reconstruct the sequence of links crossed by the traveler in chronological order. In this section a short overview of existing map matching techniques and their respective fields of application is given in order to sketch the state of the art. Two main classes of map matchers are distinguished. The technique proposed in this paper is aimed at offline (batch) map matching of GPS traces.

1.1. Online Map Matching

Online near real time map matching processes coordinate pairs as soon as they come available and aim to determine the network link that is actually being traveled. Map matchers in this class are deployed in navigation aids. Their software operates on dedicated microprocessors and typically data sampling is in the order of 1 to 100 Hz. In many cases data from several sensors (odometer, gyroscope, accelerometer, etc) are available for data fusing along with GPS coordinates. Quddus et al. (2007) provide a comprehensive overview of online map matchers. Greenfeld (2002); Ochieng et al. (2010); Li et al. (2013); Abdallah et al. (2011) discuss the data fusion techniques and inference methods. The aim of online map matching is to determine the link on which the vehicle is moving and to calculate the position of the vehicle on the link as accurately as possible (e.g. for traffic signal influencing by buses (Quddus et al. (2007))). The latter is essential in ITS (Intelligent Transportation Systems) applications and in Advanced Driver Assistance Systems. Nowadays map matchers are based on the multi-hypothesis technique (MHT) about the position of the vehicle. Such methods are called Multi-Hypothesis Map Matching (MHMM) in Bonnifait et al. (2009). In many cases, MHT and sensor data fusing feed maximum likelihood Bayesian inference engines and often Kalman filtering is used. Recent online map matchers, starting with Greenfeld (2002), usually incorporate topology constraints.

Velaga et al. (2009) present a topological map matching (tMM) method and describe the technique used to learn the value for the required weight coefficients. Selecting the initial link using the first fix (GPS point) is based on weight functions for *proximity* (distance) and *heading* (compared to link direction). Two additional weight functions are used for link selection at a junction: *turn restrictions* and *link connectivity*. All weight functions can have negative values. The total weight score (TWS) is a linear combination of the applicable weight function values. The respective weight coefficients are determined using traces for which the link sequence is known. For all fixes near junctions, weight coefficients are generated randomly until a tuple is found that correctly identifies the chosen link. Those tuples are applied to all fixes and the portion of wrong fixes is determined for each case. Finally the tuple leading to the minimal error is determined by regression. Weight coefficient tuples for *rural*, *suburban* an *urban* environments are presented.

1.2. Offline Map Matching

Offline or batch map matchers aim to process previously recorded sequences of coordinate pairs in order to extract travel behavior information either for a single moving object (either person or vehicle) over a long period or for a large set of moving objects. GPS recordings are either *vehicle traces* produced by dedicated devices mounted in a vehicle or *person traces* recorded by smartphones carried by individuals. The aim is to determine the sequence of links used by the moving object. Schüssler and Axhausen (2009) state that map matching of person traces requires high resolution network information. Available data consist of time series of GPS recordings and in some cases from other sources (Bluetooth, Wifi and mobile phone related events). Large datasets are available and need to be processed efficiently. In Quddus et al. (2007); Schüssler and Axhausen (2009), map matching techniques

(both online and offline) are classified as (i) pure geometry based methods, (ii) topological methods, (iii) probabilistic methods and (iv) advanced procedures. Pure geometric methods are further classified by Quddus et al. (2007) as point to point matching (finding the nearest node or shape point), point to curve matching (finding the polyline to which the distance is minimal) and curve to curve matching (matching the vehicle trajectory against known roads). Those methods can deliver link sequences that represent non-connected walks in the network.

The technique proposed by Marchal et al. (2005) solves the problem of non-connected walks by adding topological constraints. It starts by determining which links are identified by the first few GPS recordings. Each of those constitutes the first link in a candidate path. When the next GPS coordinate pair is processed, for each route candidate being built, only the last link in the sequence and the links that can be reached from that link (*forward star*) are investigated when looking for links matched by the new coordinate pair. Since each route candidate shall consist of a linear sequence of links, route candidates are cloned and each clone is extended by exactly one member of the forward star. The route candidates then are assigned a score and in order to avoid huge sets of route candidates, only the N route candidates having the best scores are kept ($N = 30$). Scoring is done as follows. Each GPS point can match at most one link in each route candidate and each link in the route needs at least one GPS fix. The distance between the point and the link is a measure of quality of the selection (the lower the better). For each GPS fix, the distance between the recorded position and the matched link is computed. The sum of those values constitutes the score for a candidate link sequence. If there are too many candidates, the ones having the highest scores are discarded. The computational effort and memory requirements grow with N. Making N too small, can cause promising candidates to be removed prematurely and hence can decrease the average quality of the final candidates. Schüssler and Axhausen (2009) evaluate this technique by comparing the quality (score) of the best solutions found and the corresponding computational effort for several values for the maximal candidate set size N. The paper concludes that the value reported in Marchal et al. (2005) is a valid one; the average score per GPS point does not significantly decrease with the candidate set size for $N > 30$. It also reports that the processing time per point is between 10[ms] (for $N = 20$) and 75[ms] (for $N = 100$).

Zhou and Golledge (2006) use a similar procedure implemented in ArcGIS. GPS recordings are processed sequentially and a pool of candidate solutions is kept. In a preprocessing stage, they first replace clusters of GPS points by their centroid (*cluster reduction*) but also add interpolated GPS points when the distance between two consecutive points is larger than half of the minimum length for the links in the buffer defined by the two GPS points. Then a 2-norm (distance) and a rotation measure are used to determine the weight for each point in the preprocessed dataset. A set of candidate partial paths is kept and extended so that a connected walk results from the method. In the link selection phase, a Dempster belief function is used to determine the plausibility of the selected link. However, the authors do not explain what criteria were used.

Feng and Timmermans (2013) use a Bayesian Belief Network (BBN) to replace the ad hoc rules used in map matchers not making use of the multi-hypothesis technique (MHT), to select the next road segment candidate in a route. The input for the method consists of (i) PDOP (Positional Dilution Of Precision), (ii) the difference in direction between the road segment candidate on one hand and the line segment defined by the last two GPS points on the other hand, (iii) the distance from the GPS point to the line segment, (iv) the connectivity between road segments and (v) azimuth information. For a set of routes the

effectively used line segments have been recorded by the traveler. This dataset serves as the truth value which is used for training the BBN. While processing a new sequence of GPS recordings, the BBN is used to determine the probability for a candidate link to become the next one in the route. The link having the highest probability is selected. In this procedure, the topological constraint is not forced. Connectivity information is used as an input variable and the resulting sequence of selected road segments is not necessarily a connected one.

Chen et al. (2011) propose a probabilistic method to simultaneously detect the road segment sequence and the transportation modes used. The likelihood that a given multi-modal path in a network generates the observed sequence of smartphone data is estimated. The measurement equations establish the probability that a given path generates a given time series of measurements. The travel model consists of frequency distributions for the speed estimated for six different modes. The phone measurement model involves GPS coordinates, speed, acceleration and Bluetooth events.

Bierlaire et al. (2013) further elaborate the proposed probabilistic measurement model introduced by Chen et al. (2011) and show how to compute the integrals required for likelihood evaluation. The path is decomposed into arcs and integrals are evaluated over each arc and summed. The concept of *Domain of Data Relevance (DDR)* is used to limit the computational requirements; e.g. the difference between the arc direction and the reported heading (in points where the speed is sufficiently high) are used to discard candidate links. The procedure explicitly takes the map inaccuracy into account and rigorously elaborates the measurement equations and the traffic model. Network topology is taken into account during the path generation phase which is similar to the one used in Marchal et al. (2005) and in Schüssler and Axhausen (2009) but allows to look ahead over multiple links in order to relax the requirement that each link needs to be matched by at least one GPS recording.

The methods mentioned above process the GPS points in chronological order. For each point, they decide whether or not to accept a link as the next one in a candidate sequence based on scoring or rigorous stochastic likelihood calculations respectively. Each procedure keeps track of a limited set of candidate paths.

Brakatsoulas et al. (2005) propose three algorithms: (i) a greedy algorithm processing one point at a time using a distance and an angular criterion to select the next edge, (ii) a *recursive local look-ahead* method (inspecting up to 4 network links and GPS points ahead) and (iii) a *global method* that minimizes the Fréchet distance between curves. The latter method consists of the following steps. First the concept of *free space* is introduced. This is the set of points on two curves for which the distance is less than a given ϵ. Curves of finite length are defined by $[0,1] \rightarrow \mathcal{R}^2$ so that the free space is a subset of $[0,1]^2$. It is observed that if and only if a (monotone) continuous curve from (0,0) to (1,1) does exist in the free space, the (strong) Fréchet distance between the curves is less than ϵ. The free space concept then is extended to *free space surface* in order to compare a curve C to a graph (each edge combined with C generates a free space and those are combined into a *free space surface*). The sequence of GPS coordinates constitutes a piecewise linear curve. For a given ϵ, the free space surface for such curve and each path in the graph is computed. Finally the minimum value for ϵ for which a (monotonic) curve can be found in the free space surface, is determined by parametric search. This results in the *globally optimal* sequence of links (i.e. the one that delivers the minimum ϵ value). This method delivers topologically valid sequences and does not require each traversed link to be matched by a GPS point. The complexity of the method using weak Fréchet distance is $O(mn \cdot log(mn))$ where m is the number of vertices and edges and n is the number of GPS points. Processing time is not

given: the paper only states that the runtime for the global methods was much longer than the runtime for the incremental methods.

Wei et al. (2012) present a clear overview of the information and weight functions used in published maximal weight methods. A global method based on a hidden Markov model (HMM) and the Viterbi probability maximization is proposed. Weight functions are chosen to achieve both high accuracy and processing speed (ten thousand GPS records per second). In case the links matched by two consecutive GPS fixes are disconnected, shortest paths are inserted to connect each link matched by the first fix to each link matched by the second fix. The search for such paths is restricted by an upper bound for the path length derived from a maximal speed value and the observed travel time between the consecutive fixes.

Deka and Quddus (2015) present a global method for trip based offline map matching that minimizes the additional data requirements like route choice, mobility patterns etc. The method selects the candidate path that maximizes the sum of weight function values for each GPS point over the complete trip. In each step the last two fixes and their projections on the matched links are used in the weight computation.

In the method presented in this paper (i) a set of multiple initial matches is allowed, (ii) outliers are processed according to the GPS device accuracy, (iii) the concept of *likelihood* (as opposed to *weight*) is used, (iv) direction is not considered and (v) a graph is build that allows for enumeration of high likelihood candidate paths for posterior application of rules based on additional assumptions.

1.3. Contribution

This paper contributes as follows:

1. the proposed MHT does limit the size of candidate routes in advance but provides an acyclic digraph that allows for efficient enumeration of candidates

2. the GPS trace is subdivided into parts and likely global routes are found by combining maximum likelihood (MLH) partial routes corresponding to the trace parts

3. behavioral properties are not used by the candidates enumerator and can be used in a second stage for filtering (depending on the actual research objectives).

Conceptually map matching consists of two stages. The first stage builds a graph containing space and time information from which *all* candidate link sequences can be generated. This graph only contains space-time information extracted from the GPS trace and the road network. In the second stage, additional assumptions can be added to refine the link selection (e.g. traveller behavior assumptions). The focus in this paper is on the first stage.

For the convenience of the reader, a symbol table is provided in Appendix A and a list of abbreviations is provided in Appendix B. In order to avoid confusion between different graphs in the remainder of the text, symbols denoting graphs, vertices and edges will bear a superscript.

2. Application Domain - Map Matcher Design Decisions

The intended use of the map matching results influences the tool design. If the result of the map matching is used as input for other research, one shall be careful about the assumptions made by the map matching procedure. Let W denote the set of walks in the graph representing the road network (contiguous link sequences) output by the map

matching procedure. If subsequent research aims to verify a particular hypothesis H to hold for W, then the map matcher shall not use any assumptions that affect the hypothesis H verification. Example: when the final result serves to assess speeding behavior the link selection stage shall not include *speeding* related behavior assumptions (e.g. respecting maximum speed on road segments). A map matching procedure can consist of two stages so that (i) the first stage is based on information in the GPS trace and in the given map only and (ii) additional assumptions are concentrated in the second stage. This paper focuses on the first stage and hence does not make assumptions about the driver behavior.

This section briefly discusses the intended use of the map matcher and some of the design decisions emerging from the related requirements.

1. Results of map matching are used in research projects that focus on the analysis of revealed travel behavior. In particular, researchers aim to extract properties of routes revealed by GPS traces in order to support route choice set generation (i.e. the same purpose as the one mentioned in Bierlaire et al. (2013)). This leads to the requirement to efficiently derive the route in the road network that has the highest probability to have generated the time series of GPS recordings. Such route in general corresponds to a graph theoretical *walk* (repeated visits of edges and vertices allowed) and not necessarily to a *path* (no repeated visits allowed).

2. The map matched link sequences serves *route splitting* research reported in Knapen et al. (2014); Knapen (2015); Knapen et al. (2016) that aims to investigate how revealed routes can be decomposed into a minimum set of least cost subroutes. The size of such *minimum decompositions* (the *route complexity*) is expected to deliver relevant information to increase the quality of *route choice set generation* in travel behavior research. Accurate map matching is required because it heavily affects the complexity of the resulting link sequences (paths in a graph). On one hand, gap filling by means of shortest route segments in the case of missing recordings reduces the route complexity; hence it introduces bias and needs to be avoided. On the other hand, the requirement to have at least one GPS record for each link in unfeasible in practice because of the occurrence of very short links in junctions of multi-lane roads.

This paper proposes a map matching method based on likelihood maximization in which the requirement to have at least one GPS record for each link is relaxed.

3. Map Matching: Principle of Operation

The road network is modeled by a directed graph $G^T(V^T, E^T)$. The superscript T is used to identify the transportation graph. Each vertex $v \in V^T$ corresponds to a node in the network. Each edge $e \in E^T$ is associated to an ordered pair of nodes $\langle v_s, v_t \rangle$ and identifies the set of lanes of a particular road segment usable to move from the source v_s to the target v_t. A bidirectional road segment is represented by two edges.

The proposed method operates as follows. The trace is subdivided into contiguous subtraces. Each subtrace corresponds to a subgraph of G^T (network of links) that may have been used to generate the subtrace. Subgraphs for chronologically consecutive subtraces are constructed so that they are not disjoint and hence particular links or nodes are crossed while transiting from one subtrace to the next one. In general, multiple candidates do exist for each subtrace border crossing. Hence, for each subgraph multiple entry-exit combinations are available. Maximum likelihood (MLH) partial routes are determined for each entry-exit

combination in each subtrace. A new acyclic digraph is build in which each edge represents a MLH partial route. Each path in this graph represents a walk in the transportation graph G^T that consists of MLH partial walks. This graph allows to easily find the MLH walk in G^T for the given trace and also allows for enumeration of near maximum likelihood walks. Map matching a GPS trace proceeds in several steps.

1. In the first step, links *matched* by GPS recordings are selected for processing. The distance threshold R_M^I is used as a *selector* and no qualification or evaluation is applied yet. This is similar to what is done in other methods described in the literature. It corresponds to what is called *Domain of Data Relevance (DDR)* by Bierlaire et al. (2013). The purpose is to discard sufficiently improbable links. Details are explained in Section 3.4.

2. In the second step, the chronologically ordered sequence of GPS recordings is partitioned into contiguous subsequences so that each recording in a part matches the same set of links and so that the parts are maximal contiguous subsequences (see Figure 1). Each such part corresponds to a period in time (denoted by p) which is defined by the first and last recordings in the part. Potential link use is identified by a *link-period* pair $\langle l, p\rangle$. Details are explained in Section 3.7.

3. In the third step, the *link-period* pairs $\langle l, p\rangle$ are used as vertices to construct a graph G^U in which each vertex represents the assumption that a specific link l is used in a specific period p (not necessarily for the full duration of p). The subgraph for period p is denoted by G_p^U. Details are explained in Section 3.7. The links used in G_p^U constitute a (not necessarily connected) subgraph G_p^T of the transportation network.

   Each subgraph G_p^U contains some links that are in use at the start of period p (period entry links) and some links that in use at the end of p (period exit links). These sets may be disjoint or intersect. Each pair $\langle l_{en,p}, l_{ex,p}\rangle$ where $l_{en,p}$ is an *entry* link and $l_{ex,p}$ is an *exit* link is considered. If $l_{ex,p}$ can be reached from $l_{en,p}$, all possible trails in $G_p^{T,U}$ linking the entry to the exit are considered and the trail delivering the maximum likelihood to have generated the partial GPS trace for period p is retained for the pair $\langle l_{en,p}, l_{ex,p}\rangle$. Details are explained in sections 3.8, 3.10 and 3.9.

4. In the fourth step an acyclic directed graph is constructed by chaining the G_p^U graphs in chronological order. The *exit* links for G_{p-1}^U are connected to the *entry* links for G_p^U: details are explained in Section 3.11. The resulting graph G^B contains all the information that describes the possible road network link use in each period. Note that this graph is *layered*: for every $i > 0$ and $j > 0$ each path from a link-pair in G_{p-i}^U to a link-pair in G_{p+j}^U necessarily uses a subpath that connects a *entry* link to a *exit* link in G_p^U.

5. Finally, a maximum likelihood link-use trail that connects $l_{en,p-1}$ to $l_{ex,p}$ is found by considering each $l_{en,p}$ to transfer from period $p-1$ to period p. By repeatedly applying this, the maximum likelihood path linking en entry link in the first period to an exit link in the last period is found. Because the graph G^B is layered this is computationally feasible. Graph G^B is also used to enumerate sets of routes having sufficient likelihood: details are explained in Section 3.12.

In summary, the trace is partitioned. Each part corresponds to a time period. Each GPS record in a part matches the same set of links and these links constitute the subnetwork

that is crossed during the time period. A computationally feasible method is proposed to find the maximum likelihood *walks* linking the entries to the exits in the subnetwork. The subnetworks then are assembled by connecting exits to entries. This results in a layering of subnetworks for which the maximum likelihood crossing walks are known. A simple recursive algorithm then is used to fine the maximum likelihood walk for the observed trace.

3.1. GPS Accuracy, Failure Probability and Matching Radius

The accuracy (in meters) $\overline{a}$ of a GPS device as specified by the manufacturer is interpreted as the positional error that is not exceeded with a given probability $\overline{p}$ (e.g. $\overline{a} = 25[m]$ for $\overline{p} = 0.95$)).

The proposed method may fail if the GPS trace contains too many consecutive erroneous recordings. Given the accuracy threshold $\overline{a}$ and the associated probability $\overline{p}$, we derive that the probability to find N_e consecutive erroneous recordings is given by $(1 - \overline{p})^{N_e}$. Let p_a denote the acceptable probability to experience a matching failure due to GPS errors causing outlier recordings. The maximum number of consecutive erroneous recordings that the map matching procedure needs to be able to overcome then is given by $\overline{N}_e = \lceil \frac{\ln(p_a)}{\ln(1-\overline{p})}$ (where $\lceil x$ denotes the minimum integer value not smaller than x). For practical cases, $N_1 = \overline{N}_e + 1$ consecutive recordings can be assumed to contain at least one correct recording. The accepted failure probability p_a shall be near to zero and for the experiments $p_a \leq 1e-9$ was required. Then $\overline{N}_e = \lceil \frac{\ln 1e-9}{\ln 0.05} = \lceil \frac{-20.72}{-2.996} = 7$

For each recorded GPS location a circular area is used to find matched links. Radius R_M^I is used to find an initial set of links matched by the chronologically first N_1 GPS recordings. The matching radius is given by $R_M^I = \overline{a} + \overline{m}$ where

$\overline{a}$ is the device accuracy [m]
$\overline{m}$ is the map error [m]

In the remainder of the text the device and map accuracy values are combined in $\overline{A} = \overline{a} + \overline{m}$.

3.2. GPS Accuracy and Matching Probability

The error for the GPS device is assumed to have a normal distribution with zero mean and given standard deviation σ for both longitude and latitude: $e_{lon} = e_{lat} \sim Normal(0, \sigma)$. It is assumed that the error does not exceed a given value $\overline{a}$ with probability $\overline{p}$. Then the standard deviation follows from the inverse of the cumulative distribution function for the normal distribution : [1]

$$\sigma = \frac{\overline{a}}{\sqrt{2} \cdot erf^{-1}(2 \cdot \overline{p} - 1)} \tag{1}$$

The distance d between the true position and the measured one then is given by $d = \sqrt{e_{lon}^2 + e_{lat}^2}$ and has a Rayleigh distribution: $d \sim Rayleigh(\sigma)$. The probability that the error is larger than d is determined as follows. The CDF (cumulative distribution function) for the Rayleigh distribution is given by

$$F_R(x) = 1 - exp(-\frac{x^2}{2 \cdot \sigma^2}) \tag{2}$$

[1] expressions for the cumulative distribution function for the normal distribution are found in Weisstein (1999); Spiegel (1968) and others

and hence for observed distance d the probability for the measurement to have been generated from the true position is given by

$$q(d) = \text{Prob}(error > d) = 1 - F_R(d) = exp(-\frac{d^2}{2 \cdot \sigma^2}) \tag{3}$$

3.3. Link Match Likelihood

Let $\underline{\Delta} = \Delta(l, \langle x_g, y_g \rangle)$ be the minimum distance between the position specified by the GPS coordinate pair $\langle x_g, y_g \rangle$ and the geometry of link l. Then the probability that the position error is larger than $\underline{\Delta}$ (denoted by $q(\underline{\Delta})$) is taken as an estimate of the likelihood $\ell(\langle x_g, y_g \rangle | l)$ for $\langle x_g, y_g \rangle$ to have been generated from a position on the link. The likelihood value decreases with the distance between the location specified by the GPS recording and the link geometry. If the GPS coordinate is exactly on the link, the likelihood equals one.

Note that the likelihood value $q(\underline{\Delta})$ is an approximation and it is an overestimation. This can be seen as follows: let $x(z), y(z)$ with $z \in [0, 1]$ be the parametric specification of the geometry of the link; z is the developed relative distance measured along the road between the first vertex and $\langle x(z), y(z) \rangle$; sometimes z is called the *linear reference*. Then the likelihood for the GPS recording coordinate pair $\langle x_g, y_g \rangle$ to have been generated from link l is given by

$$\ell(\langle x_g, y_g \rangle | l) = \int_{z=0}^{z=1} q(\Delta(\langle x(z), y(z) \rangle, \langle x_g, y_g \rangle) \cdot p(z) \text{dz} \tag{4}$$

where $\Delta(\langle x(z), y(z) \rangle, \langle x_g, y_g \rangle)$ is the euclidean distance between $\langle x(z), y(z) \rangle$ and $\langle x_g, y_g \rangle$ and $p(z)$ is the probability density function for z. When the speed is assumed to be constant, $p(z)$ is constant. Then $p(z) = 1$ since $\int_{z=0}^{z=1} p(z) = 1$ and as a consequence

$$\ell(\langle x_g, y_g \rangle | l) = \int_{z=0}^{z=1} q(\Delta(\langle x(z), y(z) \rangle, \langle x_g, y_g \rangle) \cdot \text{dz} \tag{5}$$

Since $\underline{\Delta}$ is minimum

$$\forall z \in [0, 1] : \underline{\Delta} \leq \Delta(\langle x(z), y(z) \rangle, \langle x_g, y_g \rangle) \tag{6}$$

and as a consequence

$$\ell(\langle x_g, y_g \rangle | l) = \int_{z=0}^{z=1} q(\Delta(\langle x(z), y(z) \rangle, \langle x_g, y_g \rangle) \text{dz} \leq \int_{z=0}^{z=1} q(\underline{\Delta}) \text{dz} = q(\underline{\Delta}) \tag{7}$$

In the proposed method, the likelihood does not depend on the speed. Schüssler and Axhausen (2009) add a penalty term proportional to the square of the difference between the actual speed and the free-flow speed in the *weight* based scoring function. We deliberately refrain from making the likelihood dependent on the speed because part of the traces are produced by vehicles in congested traffic. The mentioned speed difference is not related to the positional measurement error (although the actual speed may be related).

3.4. Link Matching - Sub-network to Search

A link in the road network is *matched* by a GPS recording $\langle x, y, t\rangle$ if and only if the minimum distance between the point $\langle x, y\rangle$ and the link geometry is not larger than the matching radius R_M.

For each trace, the *complete* road network is searched for links matched using $R_M = R_M^I$ by the first N_1 GPS recordings (the N_1-head of the GPS trace) because at least one correct link match is required to start the algorithm. For GPS recording g_i with $i > N_1$, link matches are only searched for in the *subnetwork to search (SNTS)* which is a small subnetwork of G^T and which evolves as processing proceeds. In the latter cases, the matching radius $R_M = R_M^E$ (defined below) is used. The links matched by the N_1-head of the trace constitute the initial SNTS denoted by $G_0^{T,S}$.

GPS recordings are processed in chronological order. As soon as a GPS recording does not match the same set of links as its predecessor, a new SNTS needs to be established. In order to find the links matched by the i-th recording ($i > N_1$) that does not share the matched link set (MLS) with its predecessor, the j-th SNTS (denoted by $G_j^{T,S}$) is derived as follows. The set $\underline{L}$ of links matched by the N_1 most recent predecessors $\{g_k | k \in [i-N_1, i-1]\}$ is extended with all links that can be reached from at least one link in $\underline{L}$ when moving at speed $\overline{v}$. The value $\overline{v}$ is a global upper bound for the speed on the road (not the local speed limit). The maximum distance driven between two locations for which consecutive GPS recordings are generated is $d = R_M^I + \overline{v} \cdot \overline{\delta} = \overline{a} + \overline{m} + \overline{v} \cdot \overline{\delta}$ where

- $\overline{v}$ is the expected upper bound for the speed [m/sec]
- $\overline{\delta}$ is the expected upper bound of the recording period [sec] in case no data are lost (1.5 times the nominal value δ_n^s is considered to be plausible)

The SNTS is not necessarily a connected network. This holds for the initial and all consecutive SNTS. The procedure used to determine the SNTS ensures that each of the consecutively generated SNTS $G_j^{T,S}$ contains a subnetwork matched by at least one non-erroneous recording. It shows the importance of the $N_1 = \overline{N}_e + 1$ value which is determined by the quality of the dataset of GPS recordings to be processed.

For each GPS recording beyond the first N_1-th, link matching in the SNTS uses $R_M^E = d = R_M^I + \overline{v} \cdot \overline{d}$. This may seem counter-intuitive because too many links will be matched (selected). However, the matching function returns the minimum distance between the location determined by the GPS recording and the link geometry. If R_M^I were used for matching, the matched link sets for two consecutive GPS recordings may by disjoint because it is not required that each crossed link generates at least one GPS record. Then matched links need to be connected in some way. The proposed method uses a maximum likelihood technique. The distance returned by the matching function is used to determine the probability for each link-recording pair $\langle l, r\rangle$.

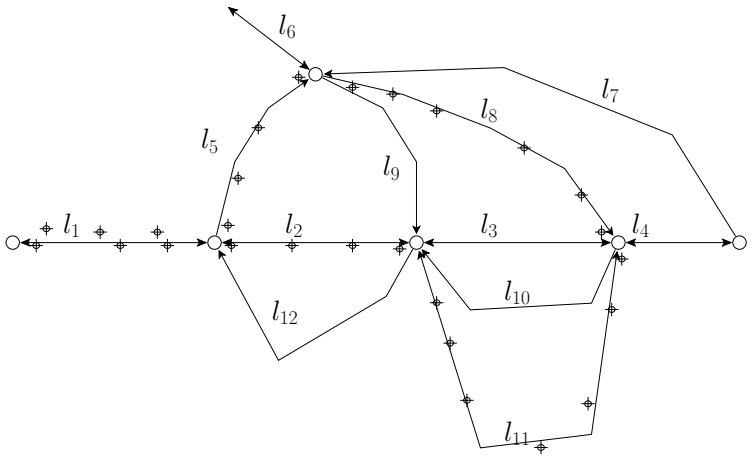
3.5. Matched Link Sets and GPS Trace Partitioning

Similar to other authors, we assume that the timestamp in the GPS records is correct. GPS recordings are processed in chronological order.

Definition 3.1 (trace)**.**

A trace is a chronologically ordered sequence of GPS recordings for a trip.

Definition 3.2 (subdivision of a trace)**.** *A* subdivision *of a trace $\mathcal{T}$ (ordered sequence of recordings) is a partition so that each part τ consists of a contiguous subsequence i.e. if $g_i, g_j \in \tau$ for $i < j$ then $(\forall k \mid i < k < j) : g_k \in \tau$ where g_i, g_j, g_k denote GPS recordings and $\tau \subseteq \mathcal{T}$ is a part in a subdivision of $\mathcal{T}$.*



Period	→ MLS
p_1	$\rightarrow\{l_1\}$
p_2	$\rightarrow\{l_1, l_2, l_5, l_{12}\}$
p_3	$\rightarrow\{l_1, l_2, l_5\}$
p_4	$\rightarrow\{l_5, l_6, l_7, l_8, l_9\}$
p_5	$\rightarrow\{l_7, l_8, l_9\}$
p_6	$\rightarrow\{l_8\}$
p_7	$\rightarrow\{l_4, l_8, l_{10}, l_{11}\}$
p_8	$\rightarrow\{l_{11}\}$
p_9	$\rightarrow\{l_2, l_3, l_9, l_{10}, l_{12}\}$
p_{10}	$\rightarrow\{\text{outlier}\}$
p_{11}	$\rightarrow\{l_2, l_{12}\}$
p_{12}	$\rightarrow\{l_2\}$
p_{13}	$\rightarrow\{l_1, l_2, l_5, l_{12}\}$
p_{14}	$\rightarrow\{l_1\}$

Figure 1: Mapping chronologically contiguous GPS subsequences to $\langle Period, MLS \rangle$ pairs. Each subsequence of GPS recordings maps to a CMLS-MP (Complete Matched Link Set for Maximal Period).

The given trace is subdivided as follows.

1. For each recording, the *matched link set (MLS)* is determined as follows: the SNTS is searched for links matched by the GPS recordings using the extended matching radius $R_M^E = d$. This is done because we need a likelihood value for each link that can have been used and not only for the ones that are within a distance from the GPS recording defined by the device accuracy. The resulting set is called the matched link set (MLS).

2. Multiple chronologically consecutive recordings may generate the same MLS (illustrated in Figure 1). This allows to partition the sequence of recordings into contiguous subsequences such that two chronologically consecutive recordings belong to the same part if and only if they share the MLS. Since the GPS sequence is chronologically ordered, each part corresponds with a time period and the time periods are disjoint. The partitioning is illustrated by the legend in Figure 1. The left side shows a part of the road network along with some locations determined by GPS recordings. The legend on the right side shows contiguous subsequences of the GPS trace and their mapping onto tuples consisting of a MLS and a period.

3. For each MLS a matrix ℓ is kept. The element $\ell[l, g]$ with $l \in MLS$ and $g \in \mathcal{T}$ is (an estimation of) the likelihood for g to have been generated from a position on link l.

Each subsequence (part) determined in item 2 is a *complete matched link set for a maximal period* (CMLS-MP). It is called *complete* since the link set contains all links matched by each GPS point in the subsequence. It is *maximal* since it cannot be extended in the time dimension (due to the construction rule). The k-th CMLS-MP is described by a tuple $\langle \langle t_k^f, t_k^l \rangle, MLS_k \rangle$ where MLS_k is the matched link set and t_k^f and t_k^l are the timestamps for the first and last GPS recording (the tuple $\langle t_k^f, t_k^l \rangle$ constitutes the period identifier shown in Figure 1).

Note that an outlier GPS recording may create a part containing the outlier recording as the only element. Assume three consecutive parts $\tau_{i-1}, \tau_i, \tau_{i+1}$ (corresponding to periods p_{i-1}, p_i, p_{i+1}) in the GPS trace where τ_i is generated by an erroneous (outlier) recording.

Then it is possible that the MLS_{i-1} associated with τ_{i-1} contains some links that have a node in common with links in the MLS_{i+1} for τ_{i+1} while none of the links in the MLS_i has any node in common with links in MLS_{i-1} or MLS_{i+1} respectively. This phenomenon is shown in Figure 2 (which is compatible with the example given in Figure 1). Links l_2, l_3, $l_9, l_{10}, l_{11}, l_{12}$ are *inherited* in period p_{10}.

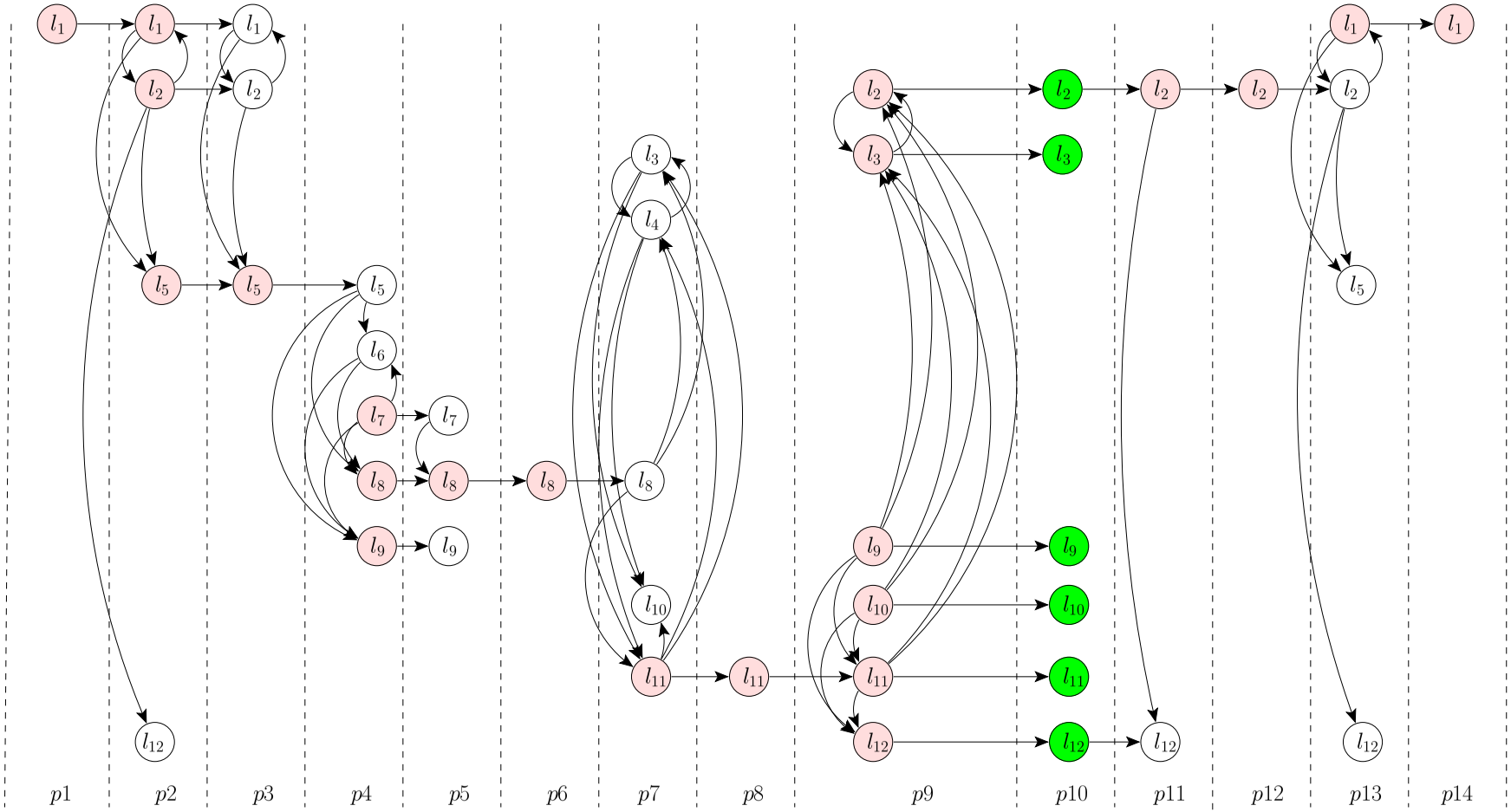


Figure 2: Sample ChronoLinkMatchGraph $G^U(V^U, E^U)$: p_i indicate periods, l_j denote links in the road network. The first *link use* in the walk corresponds to $\langle p_1, l_1 \rangle$ (initial vertex) and the last *link use* in the walk corresponds to $\langle p_{14}, l_1 \rangle$ (terminal vertex). In general, G^U contains multiple initial and terminal vertices. Red vertices as well as l_2 and l_{12} in p_{10} represent *periodExitLinks*: they are relevant for Figure 4. Green vertices are inherited (not matched in p_i but matched (at least) in p_{i-1}). Transfer from/to an inherited vertex within a period is not allowed. Link use period p_{10} was caused by an outlier GPS record: its matched link set (MLS) is empty; it contains *inherited links* (green) only. In general, a period may have non-empty MLS and ILS (not shown in order to keep the diagram simple). The route was a tour since l_1 appears *non-inherited* in the first and last periods.

3.6. Inheritance

Consider each period p_i and the corresponding matched link set MLS_i (which may be empty). In case the part τ_i for period p_i contains less than N_1 recordings, m preceding periods $p_{i-m}, \ldots, p_{i-1}$ is considered so that

$$m = \min_{j \in [0,i]} j \mid \Big(\sum_{x \in [i-j,i]} |\tau_x| \geq N_1 \Big) \vee \big(j = i\big) \tag{8}$$

where $|\tau_x|$ denotes the number of recordings in subtrace τ_x. This selects the minimum number m of preceding periods for which

$$\sum_{x \in [i-m,i]} |\tau_x| \geq N_1 \tag{9}$$

provided that sufficient recordings do exist. The inherited link set consists of all links contained in the selected predecessors link sets that are not contained in MLS_i. The inherited link set is specified by

$$ILS(p_i) = \Big(\bigcup_{x \in [i-m,i-1]} MLS_x \Big) \setminus MLS_i = \{y | (y \in \bigcup_{x \in [i-m,i-1]} MLS_x) \wedge (y \notin MLS_i)\} \tag{10}$$

3.7. Chronologically and Topologically Consistent Link Match Graph

While generating the CMLS-MP for period p_i, a tuple $\langle l_j, p_i \rangle$ is created for each link in the associated link set $MLS_i \cup ILS_i$. The $\langle l_j, p_i \rangle$ tuples for the matched and inherited links are used as vertices in a newly constructed digraph. Vertex $v_a = \langle l_{j_a}, p_{i_a} \rangle$ is connected to vertex $v_b = \langle l_{j_b}, p_{i_b} \rangle$ by a directed edge if

1. either $p_{i_b} = p_{i_a}$ (same period) and the target vertex in G^T of l_{j_a} is the source vertex of l_{j_b} (*topological* constraint): this is used to move to a topologically compatible road network link within period p_{i_a}

2. or $p_{i_b} = p_{i_a+1}$ (*chronological* constraint) and either $l_{j_a} = l_{j_b}$ or the target vertex of l_{j_a} in G^T is the source vertex of l_{j_b} (*topological* constraint): this allows to either move to a topologically compatible link or to stay on the same link during the transition to the next period p_{i_b}.

The resulting graph $G^U(V^U, E^U)$ is called the *ChronoLinkMatchGraph (CLMG)*. Figure 2 shows an example. Period p_3 in Figure 2 is caused by an outlier GPS recording matching only l_5 and represented by the red circle labeled D. Inherited links are represented by green circles; they appear in all periods that contain less than N_1 GPS recordings and not only in periods generated by outliers. This illustrates the problem described in Section 3.5.

Note that p_{i+1} denotes the immediate successor period of p_i. The resulting graph G^U has a *layered* structure since each path in the graph G^U is defined by a sequence of vertices $v_0, v_1, \ldots, v_k, v_{k+1}, \ldots$ for which equation 11 holds:

$$\begin{aligned} v_k = \langle l_x, p_m \rangle \Leftrightarrow v_{k+1} = \langle l_y, p_n \rangle \mid \\ \big((p_m = p_n) \wedge (\text{target}(l_x) = \text{source}(l_y))\big) \\ \vee \big((p_{m+1} = p_n) \wedge ((l_x = l_y) \vee (\text{target}(l_x) = \text{source}(l_y))\big) \end{aligned} \tag{11}$$

There is a one-to-one correspondence between *layers* and *periods*. The subgraph consisting of vertices $\langle \cdot, p \rangle$ is the ChronoLinkMatchGraph *layer* for period p denoted by G^U_p. Such subgraph in general is not acyclic. However, contracting G^U by replacing all vertices for a period p by a single vertex, leads to an acyclic graph (in particular a to linear sequence). This is easily observed in the sample CLMG in Figure 2.

3.8. Layers in the ChronoLinkMatchGraph

Following concepts are used.

Definition 3.3 (periodEntryLink)**.** *A link is a* periodEntryLink *for a given period if and only if it can be in use by the traveler at the start of the period.*

For the chronologically first period every matched link is a *periodEntryLink*; for consecutive periods, a link l is a *periodEntryLink* if and only if l or one of its topological predecessors was either matched or inherited in the preceding period.

Definition 3.4 (periodExitLink)**.** *A link is a* periodExitLink *for a given period if and only if it can be in use by the traveler at the end of the period.*

Every link matched in the last period is a *periodExitLink* for that period; for two consecutive periods, l is a *periodExitLink* of the chronologically first one if and only if l or a topological successor is matched or inherited in the succeeding period.

The corresponding link sets *periodEntryLink*s (denoted by L_p^{En}) and *periodExitLinks* (denoted by L_p^{Ex}) are determined for each period p.

Definition 3.5 (periodTransferLink)**.** *A link l_1 is a* periodTransferLink *for l_0 in period p_i if and only if $l_0 \in L_{p_i}^{Ex}$ and $l_1 \in L_{p_{i+1}}^{En}$ and either $l_1 = l_0$ or l_1 is directly reachable from l_0.*

Link l_1 is directly reachable from link l_0 if and only if one of it is one of the topological successors of l_0 (i.e. the target vertex of l_0 is the source vertex of l_1 in the digraph G^T). The set of *periodTransferLink*s for link l in period p is denoted by $L_{\langle l,p\rangle}^{Tx}$. The set of all *periodTransferLink*s for period p is denoted by $L_p^{Tx} = \bigcup_{l \in L_p^{Ex}} L_{\langle l,p\rangle}^{Tx}$.

The idea is to consider all pairs $\langle l_s, l_t\rangle \in L_{p_i}^{En} \times L_{p_i}^{Ex}$ for which $G_{p_i}^T$ contains a path leading from l_s to l_t. For each such pair, the path having the maximum likelihood (MLH) to have generated the subtrace τ_i associated with period p_i is determined as explained in Section 3.9. This leads to a $|L_{p_i}^{En}| \cdot |L_{p_i}^{Ex}|$ matrix of MLH values for period p_i.

3.9. Finding Maximum Likelihood Walks in a CLMG Layer

Consider the matched link set MLS_i for period p_i. Per construction $MLS_i \subseteq SNTS_i$ holds. The graph $G_i^T(V_i^T, E_i^T) \subseteq G^T$ can be considered as a time-space prism i.e. as the subgraph that may have been visited during period p_i taking into account (i) distances along the road and (ii) a global upper bound for the moving speed. G_i^T is not necessarily connected. $L_{p_i}^{En} \subseteq MLS_i \cup ILS_i = E_i^T$ and $L_{p_i}^{Ex} \subseteq MLS_i \cup ILS_i = E_i^T$. It is possible that $l_1 \in L_i^{Ex}$ is not reachable from $l_0 \in L_i^{En}$ in G_i^T.

According to Section 3.4 all links in MLS_i are matched by all recordings in τ_i (but MLS_i may be empty). According to Section 3.6 at least one edge in E_i^T is matched by a non-erroneous recording.

Figure 3 shows graphs used to determine the maximum likelihood assignment of GPS recordings to network links. The recordings in τ_i are processed in chronological order and each recording $g_k(\tau_i)$ needs to be assigned to an edge in G_i^T. The graph defined by the links $MLS_i \cup ILS_i = E_i^T \subseteq E^T$ in Figure 3a shows the part of the transportation network involved.

The graph G_i^A shown in Figure 3b represents the *GPS record assignment state*. A vertex in G_i^A represents a pair $\langle l_X, k\rangle$ where $l_X \in E_i^T$ is a link in the transportation network and k denotes the number of GPS recordings in τ_i that already have been assigned to the links in

E_i^T. In order to ease interpretation, the vertices have been arranged in a grid. The subgraphs in the columns except for the last one are isomorphic and each column corresponds to a particular k value. A subgraph in a column labeled $k = j$ corresponds to the state in which the first j recordings of τ_i already have been assigned.

Edges within a column (subnetwork for a particular value of k) correspond to transitions from a link e_x^T to a link e_y^T with $x \neq y$ without assigning recording $g_k(\tau_i)$ to e_x^T. An edge e_x^A in a row of the grid in Figure 3b that connects $\langle e_x^T, k\rangle$ to $\langle e_x^T, k+1\rangle$ corresponds to the assignment of $g_k(\tau_i)$ to the link e_x^T which means that $g_k(\tau_i)$ is assumed to have been generated from a position on e_x^T.

Each edge is labeled with a value $v = -\ln(\ell)$ where ℓ is the likelihood associated with the transition. All edge labels are non-negative and finite since $\ell \in (0, 1]$. The higher the likelihood, the lower the corresponding v value. The values are determined as follows.

Let $\mathrm{Prob}(e_x^T)$ be the probability that the moving object is positioned on edge e_x^T; it is based on the link length and the speed and is estimated by

$$\Delta s = \mathit{length}(e_x^T)/v \tag{12}$$

$$\mathrm{Prob}(e_x^T) = e^{-\Delta s/\Delta r} \tag{13}$$

where Δs is the expected duration to travel the link based on the value for the speed and Δr is the duration of the nominal recording period. This is based on the distribution for the time to wait for the first occurrence of a Poisson event. Because the Dijkstra (1959) algorithm requires the edge cost to be constant (independent of the head of the path being evaluated), a constant value v is assumed ($v = 16[m/s]$ for the experiments mentioned below) while Δr is given.

Let $\mathrm{Prob}(g_k(\tau_i), e_x^T)$ denote the conditional probability that $g_k(\tau_i)$ was generated from a position on e_x^T provided that the moving object is positioned on e_x^T.

The likelihood is estimated as follows:

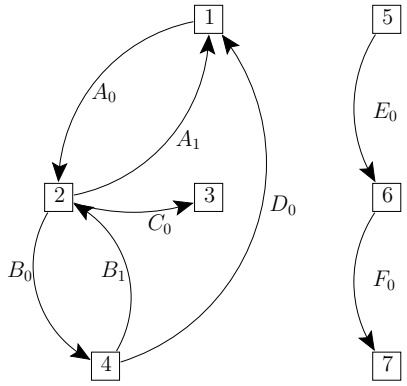
$$\ell = \begin{cases} \mathrm{Prob}(e_x^T) \cdot \mathrm{Prob}(g_k(\tau_i), e_x^T) & \text{for an edge } \langle\langle e_x^T, k\rangle, \langle e_x^T, k+1\rangle\rangle \\ 1 - \mathrm{Prob}(e_x^T) & \text{for an edge } \langle\langle e_x^T, k\rangle, \langle e_y^T, k\rangle\rangle \end{cases} \tag{14}$$

The edge labels in G_i^A are interpreted as edge traversal costs. In order to find the MLH path linking a periodEntryLink l_x^{En} to a periodExitLink l_y^{Ex} the least cost path in G_i^A from $\langle l_x^{En}, 0\rangle$ to $\langle l_y^{Ex}, |\tau_i|\rangle$ is determined using the Dijkstra (1959) algorithm. Note that the *shortest path* concept applies to minimization of unlikelihood and not to minimization of the distance along the road.

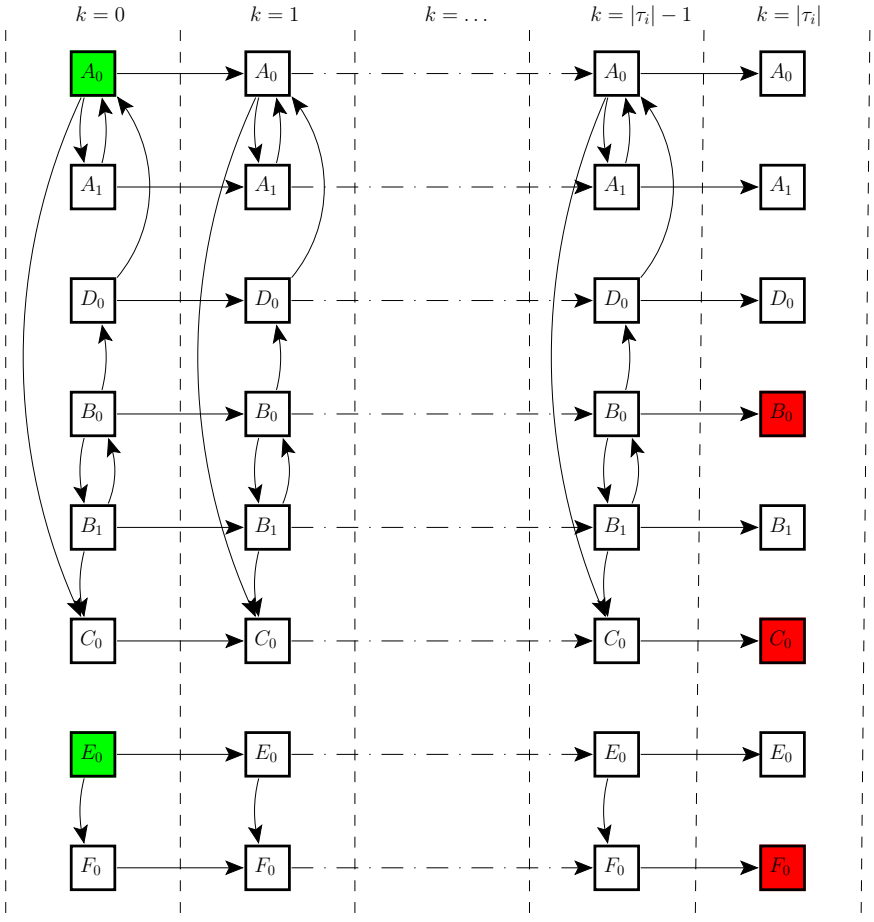
3.10. Using Layers in the CLMG to find Maximum Likelihood Walks

The link sequence in the transportation network that most likely generated the GPS trace is looked for. Several candidate links are matched by the head and tail recordings respectively resulting in candidate sets for the initial and terminal links in the link sequence looked for. MLH walks with given initial and terminal links are built in a piecewise manner as follows.

A MLH candidate walk (link sequence) in G^T is a concatenation of MLH subsequences corresponding to subtraces τ_i and hence to periods p_i. For each period p_i, every pair $\langle l_{p_i}^{En}, l_{p_i}^{Ex}\rangle \in L_{p_i}^{En} \times L_{p_i}^{Ex}$ is considered. If $l_{p_i}^{Ex}$ is reachable from $l_{p_i}^{En}$ in the graph constituted by the MLS and ILS for period p_i, the link sequence starting with $l_{p_i}^{En}$ and ending with $l_{p_i}^{Ex}$ having the largest probability to have generated the GPS subtrace τ_i is retained and denoted by $\overline{w}(l_{p_i}^{En}, l_{p_i}^{Ex})$. The corresponding probability is denoted by $\overline{\mathrm{Prob}}_{p_i}(l_{p_i}^{En}, l_{p_i}^{Ex})$.



(a) Part of the transportation graph associated with period p_i: it is defined by $MLS_i \cup ILS_i = G_i^T \subseteq G^T$. This graph is used to construct the graph in Figure 3b. Note that an edge in the transportation graph occurs in multiple vertices in Figure 3b.



(b) Graph G_i^A used to assign GPS recordings to transport network links for period p_i. A vertex labeled X in the column labeled $k = j$ represents a pair $\langle l_X, j \rangle$ where X is a transport network link shown in Figure 3a and j is the number of GPS recordings that already are assigned to a link in G_i^T (hence j is the offset in τ_i of the first GPS recording that shall be assigned to a link). A *green* colored vertex in the leftmost column belongs to L_i^{En}. A *red* colored vertex in the rightmost column belongs to L_i^{Ex}. A least cost path linking a green colored vertex (in column $k = 0$) to a red colored vertex (in column $k = |\tau_i|$) is to be found to determine a maximum likelihood (MLH) assignment.

Figure 3: Graph used to determine the maximum likelihood assignment of GPS recordings to links.

Let $\overline{\ell}_{tail}(l, p_i)$ denote the maximum likelihood for the head sequence of GPS recordings in $\mathcal{T}$ up to and including the recordings in period p_i to have been generated by a link sequence for which l is the tail. Then the maximum likelihood for a *periodExitLink* $l^{Ex}_{p_i}$ is computed by considering the maximum likelihood for each *periodEntryLink* and multiplying it with the likelihood for the walk extension that can be realized in period p_i. More formally:

$$\overline{\ell}_{tail}(l^{Ex}_{p_i}, p_i) = \begin{cases} \max\limits_{l \in L^{En}_{p_i}} \left(\overline{\text{Prob}}_{p_i}(l, l^{Ex}_{p_i})\right) & \text{if} \quad i = 0 \\ \max\limits_{\lambda, l \ \mid \ (\lambda \in L^{Ex}_{p_{i-1}}) \wedge (l \in L^{Tx}_{\lambda, p_{i-1}})} \left(\overline{\ell}_{tail}(\lambda, p_{i-1}) \cdot \overline{\text{Prob}}_{p_i}\left(l, l^{Ex}_{p_i}\right)\right) & \text{if} \quad > 0 \end{cases} \tag{15}$$

The first case in equation (15) holds because the *periodEntryLink*s in the first period do not have predecessors and their prior likelihood equals one. As a consequence, the MLH for each *periodExitLink* in period p_i can be computed either directly (for $i = 0$) or from the MLH for the *periodExitLink*s in period p_{i-1} (for $i > 0$).

By appropriate vertex contraction in CLMG layers as explained in Section 3.7 a directed acyclic graph (DAG) of *periodExitLink*s can be constructed and the MLH for each possible concatenation of link sequences for subtraces τ_i can be computed efficiently.

3.11. Maximum Likelihood Walk Generation

Finally, the *PeriodBoundaryLinkGraph* $G^B(V^B, E^B)$ is considered. An example is shown in Figure 4. A vertex $v^B \in V^B$ is a pair $\langle l, p \rangle$ where p is a period and l is an edge in the road network graph for which $(l \in L^{En}_p \wedge p = 0) \vee (l \in L^{Ex}_p)$ (the *periodEntryLink*s for the first period and the *periodExitLink*s for all periods). Vertex $v^B_0 = \langle l^T_0, p_0 \rangle$ is connected to $v^B_1 = \langle l^T_1, p_1 \rangle$ by an edge if and only if one of the following conditions is fulfilled:

1. Case $p_1 = p_0 = 0$: the vertices v^B_0 and v^B_1 belong to the same period $p = 0$ and a path from $v^B_0 = \langle l^T_0, p \rangle$ to $v^B_1 = \langle l^T_1, p \rangle$ was found in G^A_p (which means that the likelihood for a walk in G^T_0 linking l^T_0 to l^T_1 in period p is non-zero). Note that $L^{Ex}_0 \subseteq L^{En}_0$ since for $p = 0$ each link is a *periodEntryLink*.

2. Case $p_1 = p_0 + 1$: link l^T_0 has a *periodTransferLink* $l \in L^{En}_{p_1}$ that leads to $l^T_1 \in L^{Ex}_{p_1}$ in period p_1 (which means that a path was found in $G^A_{p_1}$ from l to l^T_1).

The *PeriodBoundaryLinkGraph* G^B is an acyclic digraph. An edge e^B_x is labeled with (i) the maximum likelihood value found using G^A_p which is a link additive quantity denoted by $w(e^B_x)$ and (ii) the corresponding MLH subwalk (to enable walk reconstruction). The value for a path $P_{x,y}$ in G^B connecting v^B_x to v^B_y is the sum of the corresponding edge likelihood values: $w(P_{x,y}) = \sum\limits_{e^B_z \in P_{x,y}} w(e^B_z)$. Let N denote the size of the partition discussed in Section 3.5. The graph G^B is used to find the path in G^B that delivers the largest value:

$$\overline{w} = \max_{x \in L^{En}_0, y \in L^{Ex}_{N-1}} w(P_{x,y}) \tag{16}$$

Since the PeriodBoundaryLinkGraph is an acyclic digraph, this can be achieved by means of an efficient recursive procedure. From this path in G^B, the maximum likelihood walk in G^T is easily derived.

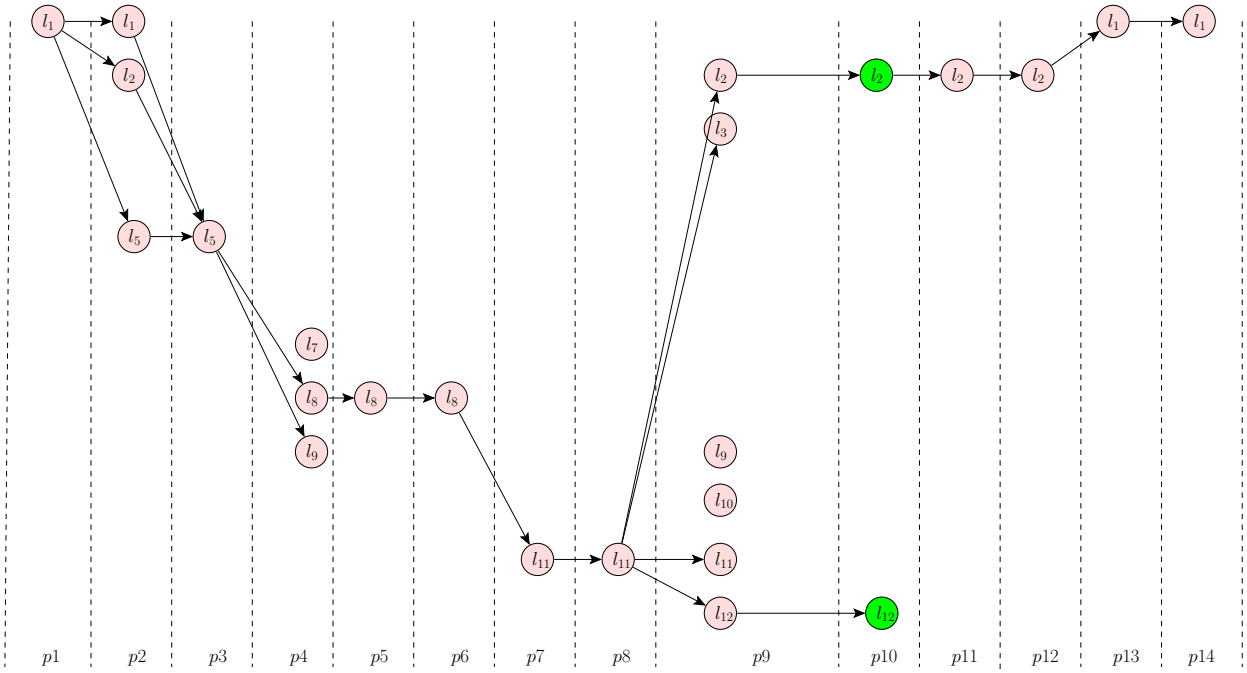


Figure 4: PeriodBoundaryLinkGraph G^B derived from the ChronoLinkMatchGraph G^U shown in Figure 2 by connecting the *periodExitLinks* for p_i to the reachable *periodExitLinks* in p_{i+1}. Each edge is labeled with a likelihood value.

3.12. Sufficient Likelihood Walk Generation

The set of paths leading to near maximal likelihood provides insight with respect to the uncertainty of the solution. The more paths have near maximal likelihood, the higher the uncertainty about which one was effectively used.

Let $\overline{\ell} = e^{\overline{w}}$ denote the maximum likelihood value found in Section 3.11. Once the maximum likelihood path is found, G^B can be used to deliver paths delivering a *sufficient* likelihood $f \cdot \overline{\ell}$ with $f \in (0, 1]$. Assume that the periods are numbered from 0 to $N_p - 1$. We look for a set of paths connecting any $v_x^B = \langle l_x^T, 0\rangle$ to any $v_x^B = \langle l_y^T, p_{N-1}\rangle$ in G^B delivering a predefined fraction of the maximum value $\ell(P_{x,y}) \geq f \cdot \overline{\ell}$ where $f \in (0, 1]$. Hence we look for paths for which $\ln(\ell(P_{x,y})) \geq \ln(f) + \ln(\overline{\ell})$ (note that $\ln(f) < 0$ and finite). This is done as follows:

1. Following concepts are used.

   Definition 3.6 (achievable value increase for a vertex pair)**.** *The achievable value increase aW_2 for a* specific vertex pair $\langle v_a, v_b\rangle$ *is the maximum value difference that can be realized by considering every path from v_a to v_b.*

   Let $\mathcal{P}(v_a, v_b)$ denote the set of all possible paths between v_a and v_b, then

   $$aW_2(v_a, v_b) = \max_{P \in \mathcal{P}(v_a, v_b)} w(P) \tag{17}$$

   Definition 3.7 (achievable value for a vertex)**.** *The achievable value aW_v for a* specific vertex v *is the maximum value that can be achieved by considering every path starting in an initial vertex and ending in v.*

   $$aW_v(v) = \max_{v_i \in \{\langle l,0\rangle \ \mid \ l \in L_0^{En}\}} aW_2(v_i, v) \tag{18}$$

Name	Value	Unit	Note
d_{min}	5 000	m	Lower bound for sampled trip length
d_{max}	50 000	m	Upper bound for sampled trip length

Table 1: Limits for the trip length used in the data generator.

Definition 3.8 (required value difference for a vertex)**.** *The required value* rW_2 *for a* specific vertex pair $\langle v_a, v_b \rangle$ *is the minimum value required in* v_a *so that there is a path* $P(v_a, v_b)$ *achieving a given value* W *in* v_b*.*

$$rW_2(W, v_a, v_b) = W - aW_2(v_a, v_b) \tag{19}$$

Definition 3.9 (required value for a vertex)**.** *The required value* rW_v *for a* specific vertex v *is the minimum value required in* v *in order to achieve a given value* W *in at least one terminal vertex* v_t *using a path containing* v*.*

$$rW_v(W, v) = \min_{v_t \in \langle L^{Ex}_{p_{N-1}}, N-1 \rangle} rW_2(W, v, v_t) \tag{20}$$

2. The overall maximum achievable value is computed as $\overline{w} = \max_{v \in \{\langle l, p_{N-1} \rangle \mid l \in L^{Ex}_{N-1}\}} aW_v(v)$.

3. The *sufficient value* then is given by $f \cdot \overline{w}$. This value is registered with each terminal vertex as the required value and the required value $rW_v(v)$ for every other vertex v is calculated recursively.

   Enumerating the paths delivering *sufficient value*, is done by successively starting in each initial vertex $v_i \in \{\langle l, 0 \rangle \mid l \in L^{En}_0\}$, recursively extending the path with a vertex v and calculating the achievable vertex pair value $aW_2(v_i, v)$. If for a given vertex v the achievable value is sufficient (i.e. $aW_2(v_i, v) \geq rW_v(W, v)$) then v is used to extend the path in the *PeriodBoundaryLinkGraph* G^B. Every time the recursive procedure reaches a terminal vertex, a *sufficient value walk* in the transportation network is found and reported.

4. Validation - Experimental Results

Collecting a large set of traces for which the base truth is known by trustworthy recording of link sequences is nearly impossible. Hence, the proposed technique was evaluated by means of synthetic traces so that for each trip both the link sequence and the associated GPS trace were available. Different parameter sets have been used to generate several cases for which the map matching accuracy was evaluated.

4.1. Link Sequence Generator

A set of trips on the Belgian road network has been created by means of a generator that takes as input the required approximate travel distance d and a value for the bearing β (direction). For each trip the uniformly distributed parameters $d \sim U(d_{min}, d_{max})$ (limits specified in Table 1) and $\beta \sim U(0, 2 \cdot \pi)$ are sampled.

A uniformly sampled random node is chosen as the start location. In each junction, all links leading to a previously unvisited node are considered for trip extension (this means that the trip will be a *path*). If at least one such node is found, the link with the smallest

Name	Value	Unit	Note
σ^s	1.0	sec	Standard deviation for the sample period
f_{min}	0.6	-	Fraction of the allowed speed that is used as a lower bound for the sampled target speed on a link
f_{max}	1.1	-	Fraction of the allowed speed that is used as a upper bound for the sampled target speed on a link (some speeding is considered)
a_{min}	-2.0	m/s^2	Lower bound for acceleration (upper bound for deceleration)
a_{max}	1.5	m/s^2	Upper bound for acceleration
c	50.0	-	Specifies a gamma distribution with mean equal to one which is used to modulate the standard deviation of the GPS error

Table 2: Parameters used for GPS trace generation.

absolute deviation between the direction defined by the endnodes and the specified bearing β is selected. That process is repeated recursively until the total length of the generated link sequence exceeds the given distance d.

Since the network is restricted to Belgium, it is finite and frequent backtracking may occur when a partial trip bounces to a border. This leads to complicated routes which ensures that the validation also covers cases that are more difficult than what realistically may be expected (see Figure 8).

4.2. GPS Trace Generator

For each synthetic trip a GPS trace is generated. The values used for the parameters are shown in Table 2. The generation process is straightforward. The moving object is assumed to start in the first node of the specified link sequence at a specified time t_0.

The nominal sample period δ_n^s is given. The actual duration δ_a^s between two consecutive recordings in a trace is stochastic $\delta_a^s \sim gamma(k, \theta)$ and is sampled for each pair. The expected value for the period equals $E[\delta_a^s] = \delta_n^s$. The standard deviation σ^s was estimated using available GPS traces. Then from the equations $\delta_n^s = k \cdot \theta$ and $(\sigma^s)^2 = k \cdot \theta^2$ holding for the gamma distribution, it follows that $\theta = (\sigma^s)^2/\delta_n^s$ and $k = (\delta^s/\sigma^s)^2$.

The moving speed is sampled for each recording. The *target* speed value $V(t, l)$ for the instantaneous speed at time t on link l is uniformly distributed $V(t, l) \sim U(v_{min}, v_{max}) = U(f_{min}, f_{max}) \cdot \overline{v_l}$ where $\overline{v_l}$ is the allowed speed on the link and $0 \leq f_{min} < f_{max}$. The *effective* speed value $v(t_i, l)$ is derived from the target value $V(t_i, l)$ and is constrained by the limiting values a_{min} and a_{max} for the acceleration. This requires to account for the remaining distance on the link and to look ahead at the speed limit on the next link. The speed equals zero in the first and last positions of the trip.

Let $\langle x_i, y_i, t_i \rangle$ denote the computed (true) position on a link and the associated time. A pseudo GPS record $\langle \tilde{x}_i, \tilde{y}_i, t_i \rangle$ is generated using

$$\tilde{x}_i = x_i + \epsilon_{x,i} \tag{21}$$

$$\tilde{y}_i = y_i + \epsilon_{y,i} \tag{22}$$

$$\epsilon_{y,i}, \epsilon_{y,i} \sim N(0, \sigma \cdot \alpha_i) \tag{23}$$

$$\alpha_i \sim gamma(c, \frac{1}{c}) \tag{24}$$

The position errors in both dimensions $\epsilon_{x,i}$ and $\epsilon_{y,i}$ obey a normal distribution with zero mean and given standard deviation. The generator allows to adjust the standard deviation

Quantity	Set of values	Unit	Note
σ	{10, 12, 15}	[m]	Standard deviation for the positional error, derived from the device accuracy specification
δ^s	{2,5,10,30}	[s]	Nominal length for the period between successive of GPS recordings

Table 3: Values for the standard deviation s of the device error and the nominal recording period δ used to generate synthetic traces. Each combination was used resulting in 12 validation experiments.

Number of edges in directed graph for network (Belgium)	= 1 239 002
Number of vertices in the network (Belgium)	= 584 795

(a) Sizes for the vertex (node) and edge (link) sets.

'Track', 'motorway', 'motorway_link', 'junction', 'trunk_link', 'residential', 'primary', 'secondary', 'track', 'mini_roundabout', 'tertiary', 'trunk', 'tertiary_link', 'driveway', 'secondary_link', 'primary_ link', 'platform', 'road'

(b) Link types selected from the OSM database.

Table 4: Properties of the road network used in the experiments. The network is extracted from OpenStreetMap.

using a factor α_i having a gamma distribution for which the mean equals one in order to simulate the variation in accuracy (as reported by the devices).

4.3. Evaluation by means of Synthetic Traces

Synthetic traces for several combinations of the standard deviation of the device accuracy $\sigma_{x,y}$ and the nominal sampling period δ_n^s have been generated using the values shown in Table 3. The other generation parameters were the same for each case and are given in Table 2.

4.4. Map Data

The road network was extracted from OpenStreetMap. Its properties have been summarized in Table 4. Since the synthetic trips and the corresponding traces were generated from this map, the map error is considered to equal zero in the validation process.

Map accuracy has been discussed in Ochieng et al. (2010) and in Bierlaire et al. (2013). Map errors cannot be ignored by the map matching process. In the case of effectively recorded traces, the map error and GPS device error are combined into a single value. The expected value for the positional error for roads in the map is added as a term in the definition of the matching radius R_M (see Section 3). This method is chosen although the map errors for points on a single polyline cannot be expected to be mutually independent.

Haklay et al. (2010) investigated the positional accuracy for OpenStreetMap roads in the Greater London area. 109 different roads having a total length of 328[km] were compared to their counterpart in ITN (Integrated Transport Network) maps for which it can be assumed that the error is below 1[m]. It is concluded that if 15 contributors are active in an area, the positional error for the road is well below 6[m]. In *complete areas* the average error is 9.57[m] with a standard deviation is 6.51[m]. In incomplete areas the average error is 11.72[m] and the standard deviation is 7.73[m]. Completeness is defined as '*a measure of the lack of data*' and examined for specific areas by Haklay (2010) using visual inspection

Name	CPU	nCores AU/PP	Memory	OS	postgres	postgis
A	Intel(R) Xeon(R) E5-2660 v4 @ 2.00GHz	28/56	128	Debian 8.10	9.4	2.1
B	Intel(R) Xeon(R) E5-2630 v4 @ 2.20GHz	8/40	128	Ubuntu SMP 4.4.0-67	9.5	2.2
C	Intel(R) Xeon(R) X5570 @ 2.93GHz	8/16	48	Ubuntu SMP 3.16.0-73	9.3	2.1
D	Intel(R) Xeon(R) E5440 @ 2.83GHz	6/8	32	Debian 8.10	9.4	2.1

Table 5: Properties for the machines used in the experiments. Under the heading *nCores*, AU/PP specifies the number of cores *available for use* by the map matcher (specified as an argument to the map matching program) vs. the *physically present* number of cores.

of maps and by comparing (by means of GIS) the total road length found in OSM and in reference maps respectively.

For the practical calculations we assume that Belgium constitutes a *complete area* and $\overline{m} = 10$[m] is used. From several non-authoritative website sources we derived that the accuracy threshold $\overline{a}$ at 95% can be assumed to be 20[m]. Hence $\overline{A} = \overline{a} + \overline{m} = 30[m]$ is used to determine the matching radius.

4.5. Computing Infrastructure Used

The map matcher is written in `Java 7`. The experiments ran on several machines; their properties are listed in Table 5.

4.6. Map Matching Quality

Table 6 summarizes accuracy values for the validation runs. For each of the 12 combinations of device accuracy (standard deviation for the device error $\sigma_{x,y}$) on one hand and recording period on the other hand, 1024 trips and their associated GPS traces were generated; this resulted in a set of 12 288 traces for validation. All average values are computed over the 1024 traces for the respective case.

The error is expressed as a function of *missed* and *extra* links.

L is the set of links in the given effective trip that generated the trace
L^+ is the set of *extra* links (links identified by the map matcher that were not in the effective trip link sequence)
L^- is the set of *missed* links (links not identified by the map matcher but present in the effective trip link sequence)

The following accuracy figures are computed for each trip for evaluation:

A_n^+ : accuracy based on the number of extra links
A_n^- : accuracy based on the number of missed links
A_d^+ : accuracy based on the total length (distance) of the extra links.
A_d^- : accuracy based on the total length (distance) of the missed links.

The following numbers were computed as average values over the respective sets and reported in Table 6 and Figure 5.

A_n : accuracy based on the number of correctly mapped links
A_d : accuracy based on the total length (distance) of the correctly mapped links.

The accuracy figures are defined by

$$A_n^+ = \max(0, 1 - |L^+|/|L|) \tag{25}$$
$$A_n^- = 1 - |L^-|/|L| \tag{26}$$
$$A_n = \sum_{\mathcal{T}_k \in C^{\mathcal{T}}(\sigma_{x,y}, \delta_n^s)} (A_n^+ + A_n^-)/2 \tag{27}$$
$$A_d^+ = \max(0, 1 - \sum_{l \in L^+} d(l) / \sum_{l \in L} d(l)) \tag{28}$$
$$A_d^- = 1 - \sum_{l \in L^-} d(l) / \sum_{l \in L} d(l) \tag{29}$$
$$A_d = \sum_{\mathcal{T}_k \in C^{\mathcal{T}}(\sigma_{x,y}, \delta_n^s)} (A_d^+ + A_d^-)/2 \tag{30}$$

where $C^{\mathcal{T}}(\sigma_{x,y}, \delta_n^s)$ denotes the collection of traces generated for a specific $\langle \sigma_{x,y}, \delta_n^s \rangle$ pair. where $d(l)$ denotes the developed link length (distance along the road). The maximum is taken in order to avoid negative accuracy values in cases of extremely short trips where $|L| \approx |L^+| \approx |L^-| \approx 1$. Table 6 shows

$\sigma_{x,y}$: the standard deviation used to generate measurement errors for both *easting* (x) and *northing* (y) coordinates expressed in meters. Due to the assumed normal distribution, the error is less than about twice the $\sigma_{x,y}$ for 95% of the recordings.

δ_n^s : nominal length in seconds of the (recording) period between consecutive GPS fixes

nLinks : average number of links per trip

A_n : average *link count based* map matching accuracy expressed as the fraction of the correctly matched links in the trip

distance : average trip length. The trip length is the sum of the lengths in meters measured along the road of all links in the trip. The lengths of the first and last links accounted for completely because the synthetic GPS traces always start and end in a network node

A_d : average *distance based* map matching accuracy expressed as the total length of the correctly matched links divided by the trip distance

The accuracy value based on distance (A_d) is slightly lower for the 2[s] recording period than for the 5[s] recording period. The effect is less prominent for the accuracy based on the number of correctly matched links (A_n). It may have been caused by incorrect selection of short links in the 2[s] case because, on average, more GPS recordings are available for each link. This may cause *overfitting* of short links to noisy GPS coordinates in the assignment stage described in Section 3.11.

Taken into account the average link length of 159[m] and the fact that at 36[km/h] 300[m] is traveled in 30[s], the accuracy for the 30[s] period is acceptable.

Figure 5 shows results for two extreme cases $\langle \sigma_{x,y} = 10, \delta_n^s = 2 \rangle$ and $\langle \sigma_{x,y} = 15, \delta_n^s = 30 \rangle$ respectively. Relative frequency histograms are shown in subfigures 5a and 5c respectively). Cumulative relative frequencies are shown in 5b and 5d. The x-axis shows the accuracy value. Note that in the case $\langle \sigma_{x,y} = 10, \delta_n^s = 2 \rangle$ 80% of the traces have an the accuracy

Std.Dev $\sigma_{x,y}$ [m]	Recording Period δ_n^s [s]	nLinks	A_n	distance [m]	A_d
10	2	140.0	0.980	25 865.8	0.989
10	5	136.2	0.980	25 166.1	0.994
10	10	127.4	0.977	25 859.9	0.993
10	30	139.8	0.947	25 709.8	0.977
12	2	133.4	0.979	25 434.1	0.987
12	5	128.5	0.980	25 633.8	0.993
12	10	125.6	0.973	25 517.4	0.992
12	30	153.2	0.942	25 503.7	0.973
15	2	146.2	0.974	24 326.2	0.984
15	5	132.3	0.974	24 927.9	0.992
15	10	129.3	0.971	25 596.9	0.991
15	30	127.5	0.941	25 783.1	0.975

Table 6: Accuracy for map matching of synthetic traces. For each $\langle \sigma_{x,y}, \delta_n^s \rangle$ case 1024 traces were processed. `nLinks` is the average number of links in the trips. A_n is the average fraction of correctly matched links. `distance` is the average trip length. A_d is the average fraction of the correctly matched distance.

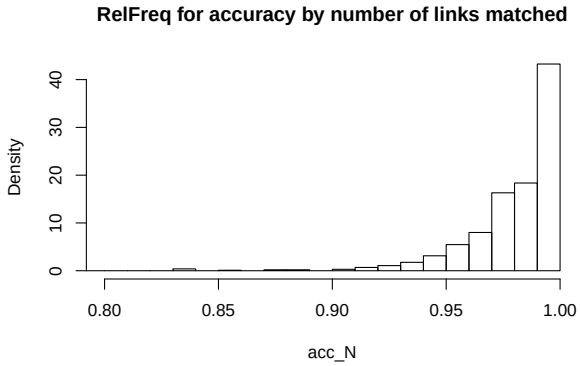
larger than 0.96 while for the case $\langle \sigma_{x,y} = 15, \delta_n^s = 30 \rangle$ 80% of the traces still have an accuracy larger than 0.90.

4.7. Samples

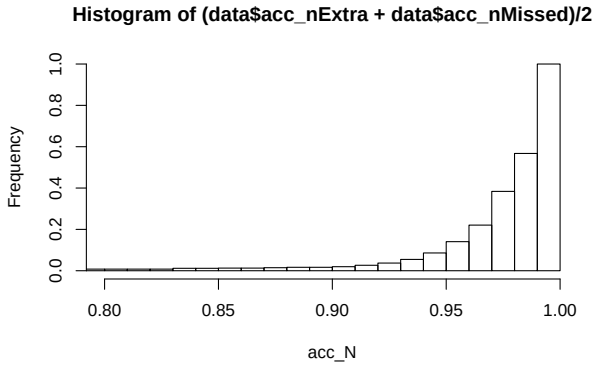
This section discusses some typical samples. Figure 6 shows part of a trip crossing a quite dense part of the network. The GPS trace belongs to the $\langle \sigma_{x,y} = 12[m], \delta_n^s = 30[s] \rangle$ case. Figure 6a shows the road network, the base truth generating trip and the link sequence resulting from map matching. In Figure 6b the road network is omitted to show the overlap of the base truth (blue) and the map matched sequence (red). They only differ in two locations (the quadrangles near the center of the map). The quality figures are:
$n = 337 \quad d = 34573.18[m] \quad A_n^- = 0.938 \quad A_n^+ = 0.973 \quad A_d^- = 0.970$ and $A_d^+ = 0.987$

Figure 7 shows a detail of a trip where the map matcher identified a plausible shortcut due to lack of sufficient GPS recordings. The GPS trace belongs to the $\langle \sigma_{x,y} = 15[m], \delta_n^s = 30[s] \rangle$ case. Figure 7a shows the road network and the base truth generating trip. In Figure 7b the road network is omitted. It shows the overlap of the base truth (blue) and the map matched sequence (red). They only differ by the missed roundabout. The quality figures are:
$n = 360 \quad d = 37211.29[m] \quad A_n^- = 0.906 \quad A_n^+ = 0.961 \quad A_d^- = 0.945$ and $A_d^+ = 0.966$

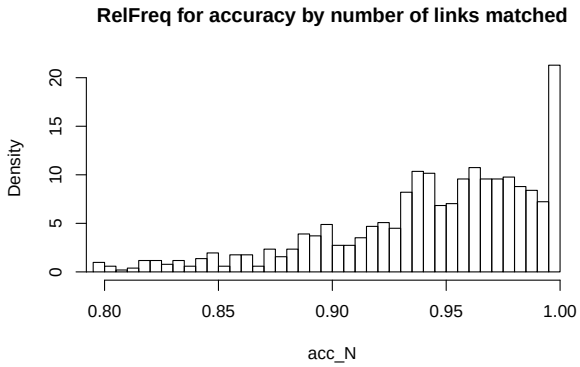
Figure 8 shows the trace for a trip that starts near the border of the map, heads to the north-east. The GPS trace belongs to the $\langle \sigma_{x,y} = 12[m], \delta_n^s = 10[s] \rangle$ case. Figure 8a shows the road network and the base truth generating trip. In Figure 8b the road network is omitted. It shows the overlap of the base truth (blue) and the map matched sequence (poorly visible due to complete overlap). The quality figures are:
$n = 356 \quad d = 41995.37[m] \quad A_n^- = 1.000 \quad A_n^+ = 1.000 \quad A_d^- = 1.000$ and $A_d^+ = 1.000$



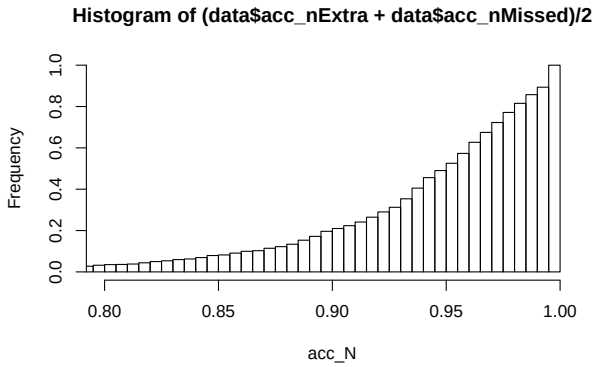
(a) Density: $\sigma_{x,y} = 10[m]$, $\delta_n^s = 2[s]$



(b) Distribution: $\sigma_{x,y} = 10[m]$, $\delta_n^s = 2[s]$

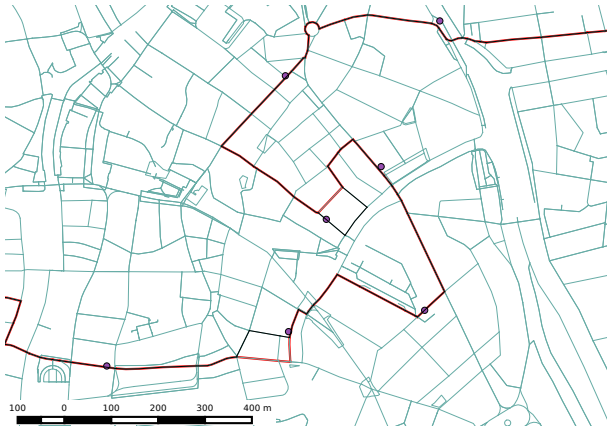


(c) Density: $\sigma_{x,y} = 15[m]$, $\delta_n^s = 30[s]$

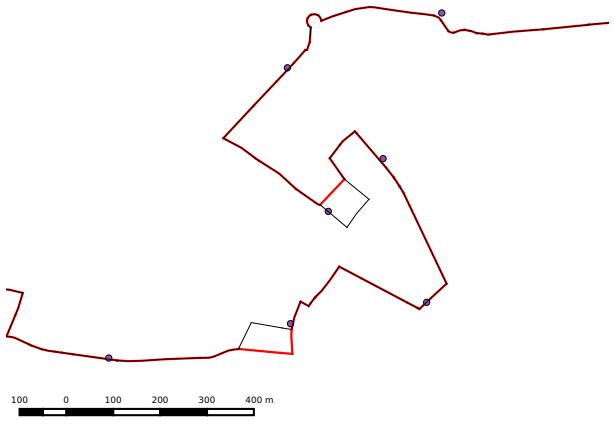


(d) Distribution: $\sigma_{x,y} = 15[m]$, $\delta_n^s = 30[s]$

Figure 5: Histograms (5a and 5c) and cumulative relative frequency diagrams (5b and 5d) for two extreme cases. The accuracy is shown on the x-axis (only the range [0.8,1.0] is shown since lower value do occur infrequently.

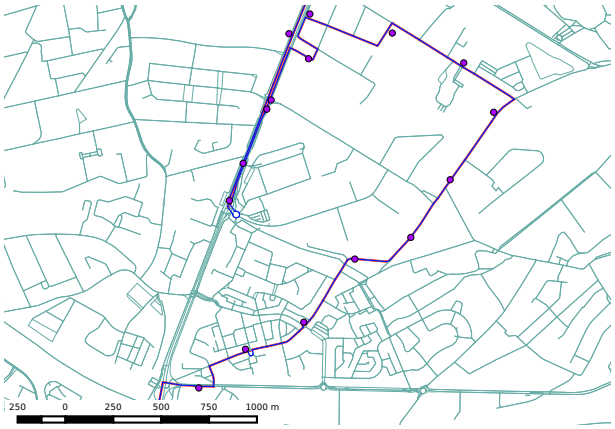


(a) Network links, given trip, matched links and GPS locations.



(b) Given trip, matched links and GPS locations.

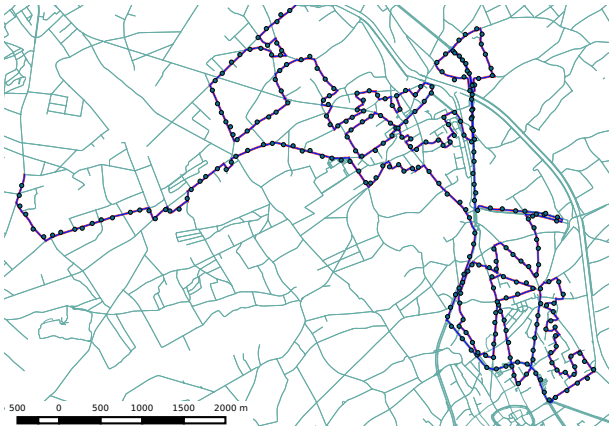
Figure 6: Trip for the case $\langle \sigma_{x,y} = 12[m], \delta_n^s = 30[s] \rangle$ crossing a dense part of the network. Two matching errors near the center. Road network (light blue), given trip (dark blue) and matched link sequence (red, mostly covered). Dots represent the recorded GPS locations.



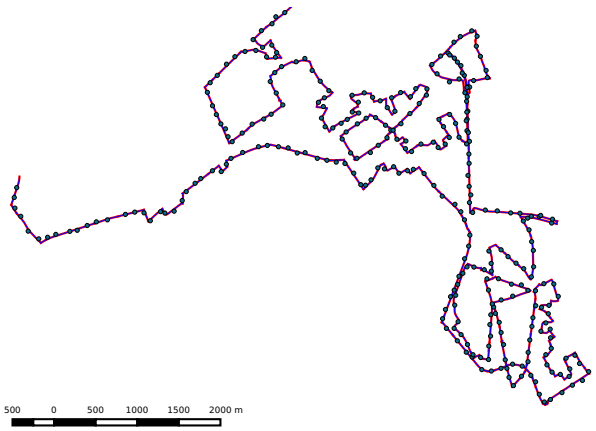
(a) Network links, given trip, matched links and GPS locations.

(b) Given trip, matched links and GPS locations.

Figure 7: Part of a trip for the case $\langle \sigma_{x,y} = 15[m], \delta_n^s = 30[s] \rangle$ crossing a dense part of the network. The roundabout near the lower left is missed due to lack of recordings. Road network (light blue), given trip (dark blue) and matched link sequence (red, mostly covered). Dots represent the recorded GPS locations.



(a) Network links, given trip, matched links and GPS locations.



(b) Given trip, matched links and GPS locations.

Figure 8: Curvy trip caused by recursive attempts by the trip generator to produce a sufficiently long trip length when the start location is near the border and the specified bearing points to the outside of the bounded network. The trip was matched without any error. The GPS trace belongs to the $\langle \sigma_{x,y} = 12[m], \delta_n^s = 10[s] \rangle$ case. Road network (light blue), given trip (dark blue) and matched link sequence (red, mostly covered). Dots represent the recorded GPS locations.

CPU	$\sigma_{x,y}$ [m]	Rec Prd [s]	# GPS Rec [-]	nLinks [-]	nG/nL [-]	Init [s]	Ext [s]	Total Time[s]	Time/GPS [ms]
D	10	2	956 192	143 368	6.67	232	23 856	24 088	25.2
D	10	5	366 044	139 485	2.62	228	8 312	8 540	23,3
C	10	10	187 819	130 475	1.44	321	3 592	3 913	20.8
A	10	30	62 331	143 195	0.44	221	27 946	28 167	451.9
A	12	2	940 238	136 611	6.88	216	6 157	6 373	6.8
B	12	5	372 585	131 588	2.83	205	3 580	3 785	10.2
C	12	10	185 522	128 639	1.44	323	3 642	3 965	21.4
A	12	30	61 979	156 880	0.40	222	38 191	38 413	619.8
B	15	2	900 573	149 718	6.02	204	9 449	9 653	10.7
B	15	5	362 544	135 467	2.68	208	3 568	3 776	10.4
C	15	10	185 824	132 421	1.40	359	3 883	4 242	22.8
C	15	30	62 535	130 582	0.48	334	73 299	73 633	1177.0

Table 7: Runtime in milliseconds per GPS record for map matching of synthetic GPS traces. For each case 1024 traces were processed. The measurement error is between two and three times the specified standard deviation.

4.8. Performance

Processing times for the validation runs are reported in Table 7. The allocation of runs to machines was determined by practical concerns and is somewhat unfortunate for easy interpretation of performance figures. Nevertheless conclusions can be drawn. Both stages of the map matcher software are designed for multi-threading. In the initial matching stage each `Java` thread uses its own `postgresql/postgis` process.

For the second stage, it was verified for all cases on all machines that the the allowed number of cores (see Table 5) was effectively used. Machines B and C were in use by other applications too; machines A and D had nearly no other workload. For each machine the runtime in milliseconds per GPS record seems to remain nearly constant for the recording periods up to 10 seconds. Processing time however heavily grows with the recording period and becomes too high. The matching radius R_M is proportional to the recording period (time-space prism) and hence the SNTS grows which causes more options to be evaluated.

Let nG denote the number of links in a trip and nG the number of GPS records in the corresponding trace. Values for nG/nL in Table 7 are average values computed over all links in the generated trips. $nG/nL \geq 1$ does not mean that for each link in the trip at least pseudo GPS record was generated.

For the cases applying to a recording period of up to 10 seconds, the total processing time is in the same order as the one reported in Schüssler and Axhausen (2009) i.e. 10[msec/point] for a 20 candidates set and 75[msec/point] for 100 candidates using single-threaded code.

5. Discussion - Future Research

5.1. MLH Estimation Heuristic

1. The proposed method is based on estimates for the likelihood for a GPS recording to have been generated from a particular link. In order to limit the computational effort the estimates are based on (i) the shortest distance between the link geometry and the recorded coordinates and (ii) a globally constant speed upper bound which enables a simple method to determine the SNTS and allows to keep the edge traversal

cost in the G^A graph constant so that the Dijkstra (1959) algorithm can be used as explained in Section 3.10. Notwithstanding the use of these heuristics, the method seems to produce accurate results. A more accurate estimate for the likelihood may increase the accuracy; on the other hand, it may also increase the computation time.

2. The simple heuristic poses a problem for short trips consisting of a single link. This is because information embedded in the chronology of recordings on a single link is ignored. In addition, the network representation for a bidirectional link has only one *geometry* specification shared by both directions; hence, each direction delivers the same likelihood value for a given GPS recording. As a consequence, they cannot be distinguished. The selection between them is solely based on the network topology and the link use chronology (as defined (i) by the ChronoLinkMatchGraph layers and (ii) by the technique used to find a MLH subwalk in the G^B graph).

5.2. Graph for Enumeration

It is not necessarily possible to enumerate all near MLH walks in decreasing likelihood order. This is not only due to the potentially large collection that can emerge from the G^B graph. A fundamental reason is the way we construct the walks. Some high likelihood walks may not be represented by the graph G^B since G^B only represents walks that are concatenations of MLH subwalks for a particular partition of the trace.

Properties of the completeness of the near MLH walk set (for a particular likelihood estimator) have not yet been proved nor disproved.

5.3. Route Candidates Set - Global evaluation

This section compares the proposed method to other procedures for batch processing of large sets of GPS recordings. The *PeriodBoundaryLinkGraph* G^B replaces the candidate sets maintained in the methods proposed by Marchal et al. (2005), Schüssler and Axhausen (2009) and Bierlaire et al. (2013). G^B is an acyclic digraph and contains all information required to enumerate a set of near MLH paths. Furthermore, that data structure allows for easy determination of the corresponding walk in G^T for every possible path in the *PeriodBoundaryLinkGraph* G^B without regeneration of any data. This results in the set of near MLH routes looked for.

The evaluation of these walks in G^T using particular (travel behaviour related) scoring functions requires a simple extension of the tool. This can be compared to the evaluation of the Fréchet distance for each path in Brakatsoulas et al. (2005). In both cases all information contained in the complete sequence of GPS points is used in the selection of each link transition in the walk. This technique avoids premature dropping of promising partial walks.

5.4. Online Map Matching

The proposed technique is aimed at *link sequence reconstruction* and as such not suited for realtime use where accurate positioning on the link is required. However, when one is not interested in the enumeration of near MLH solutions, completed graph layers can be discarded as soon as the MLH for each *periodEntryLink* is known for the *ChronoLinkMatchGraph* layer currently being processed. The technique is then usable for online operation but it needs to be extended with a facility to estimate the position on the last link of each intermediate candidate sequence.

5.5. Comparison to related Methods

The *U-turn* problem mentioned in Schüssler and Axhausen (2009) did not require any particular technique to be implemented in the proposed method. Weight accumulation based methods may suffer from *spikes* in the resulting link sequence. These are short visits of a link resulting in a U-turn. Spikes emerge because they increase the weight which is the quantity to be maximized. Such spikes however decrease the likelihood because likelihood values are smaller than one unless the GPS fix is exactly on the link geometry. Hence, they are avoided from first principles in the proposed method.

The proposed procedure and the one presented by Deka and Quddus (2015) share some properties. Both aim at processing batches of GPS recordings, compute a score for the complete path and evolve by keeping a list of next link candidates. The methods differ in following aspects: (i) likelihood is used for scoring instead of weight functions, (ii) *gap filling* is done by unlikelihood minimization in G^A as opposed to travel distance minimization in G^T, (iii) heavy weight resp. likelihood paths are kept in different ways before delivery for further evaluation and finally (iv) link geometry is handled differently (linear segments vs. OSM-based link geometries). The use of link geometries does not allow for easy heading verification unless the geometries are expanded to line segments or on-link positions are estimated.

5.6. Future Research - Extensions

1. Investigation of the number of near MLH sequences and of their properties is required. This includes estimating the effects on performance and accuracy of changes of the assumed global upper bound for the speed.

2. Investigation of the distribution for the results of behaviour related scoring functions to the set of near MLH walks.

3. Determination and evaluation of a heuristic likelihood estimate that takes timing into account at a finer level. The current technique derives the link sequence order from topology and chronology but the GPS times are not used to determine consecutive positions on a single link. Extending the method in this way may solve the problem with single link trips mentioned in Section 5.1 item 2 but it may also improve the estimate of the likelihood and hence the accuracy of the method.

4. Computational performance for cases where the recording period is large needs to be increased.

6. Conclusion

A new technique for map matching GPS traces is presented. Topological, geometric and chronological constraints are used to derive near maximum likelihood link sequences from the observed traces. The method does not require each link to be matched by at least one GPS record. Lack of link matches is not solved by bridging gaps using shortest paths but by minimizing unlikelihood using the available GPS trace. Validation by means of a set of 12 288 traces shows that the method delivers accurate results despite the use of a simple heuristic for likelihood estimation.

Apart from the coordinates and the order induced by the timestamps, no other information is required. The information available in the complete GPS trace is used to evaluate every candidate route. Partial route candidates are not dropped until the complete trace

has been processed. After processing the GPS trace, a set of near maximal likelihood routes can easily be enumerated and mutually compared using additional scoring functions that are based on parameters of travel behaviour.

The technique shows similar computational performance as state-of-the art map matching tools, based on the Multi-Hypothesis-Technique, that keep track of a limited number of candidate routes but is slower when the nominal period between successive GPS fixes grows.

Acknowledgment

The authors express their gratitude to Jan Vuurstaek, MSc and to the anonymous reviewers for useful comments and suggestions for improvement of this paper.

References

Abdallah, F., Nassreddine, G., Denoeux, T., 2011. A Multiple-Hypothesis Map-Matching Method Suitable for Weighted and Box-Shaped State Estimation for Localization. IEEE Transactions on Intelligent Transportation Systems 12, 1495–1510.

Bierlaire, M., Chen, J., Newman, J., 2013. A probabilistic map matching method for smartphone GPS data. Transportation Research Part C: Emerging Technologies 26, 78 – 98.

Bonnifait, P., Laneurit, J., Fouque, C., Dherbomez, G., 2009. Multi-hypothesis Map-Matching using Particle Filtering, in: 16th World Congress for ITS Systems and Services, HAL Id: hal-00445673, Stockholm, Sweden. pp. 1–8.

Brakatsoulas, S., Pfoser, D., Salas, R., Wenk, C., 2005. On Map-Matching Vehicle Tracking Data, in: Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005, Trondheim, Norway.

Chen, J., Bierlaire, M., Flötteröd, G., 2011. Probabilistic multi-modal map matching with rich smartphone data, in: STRC 2011.

Deka, L., Quddus, M., 2015. Trip-Based Weighted Trajectory Matching Algorithm for Sparse GPS Data, in: TRB 94th Annual Meeting Compendium of Papers, TRB (Transportation Research Board), Washington, D.C.

Dijkstra, E., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271.

Feng, T., Timmermans, H.J., 2013. Map Matching of GPS Data with Bayesian Belief Networks, in: Proceedings of the Eastern Asia Society for Transportation Studies, Eastern Asia Society for Transportation Studies, Taipei. p. 13.

Greenfeld, J., 2002. Matching GPS Observations to Locations on a Digital Map, in: TRB 2002 Annual Meeting, TRB (Transportation Research Board), Washington, D.C.. p. 13.

Haklay, M., 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. Environment and Planning B: Planning and Design 37, 682–703.

Haklay, M., Basiouka, S., Antoniou, V., Ather, A., 2010. How Many Volunteers Does it Take to Map an Area Well? The Validity of Linus' Law to Volunteered Geographic Information. Maney Publishing (c) The British Cartographic Society 47, 315–322.

Knapen, L., 2015. Refined tools for micro-modeling in transportation research. Doctoral Thesis. Hasselt University. Diepenbeek, Belgium.

Knapen, L., Bellemans, T., Janssens, D., Wets, G., 2014. Canonic Route Splitting. Procedia Computer Science 32, 309 – 316.

Knapen, L., Hartman, I.B.A., Schulz, D., Bellemans, T., Janssens, D., Wets, G., 2016. Determining structural route components from GPS traces. Transportation Research Part B: Methodological 90, 156 – 171.

Li, L., Quddus, M., Zhao, L., 2013. High accuracy tightly-coupled integrity monitoring algorithm for map-matching. Transportation Research Part C: Emerging Technologies 36, 13 – 26.

Marchal, F., Hackney, J., Axhausen, K.W., 2005. Efficient Map Matching of Large Global Positioning System Data Sets: Tests on Speed-Monitoring Experiment in Zuerich. Transportation Research Record: Journal of the Transportation Research Board 1935, 93–100.

Ochieng, W.Y., Quddus, M., Noland, R., 2010. Map-Matching in Complex Urban Road Networks. Revista da Sociedade Brasileira de Cartografia, Geodésia, Fotogrametria e Sensoriamento Remoto 55, 14.

Quddus, M.A., Ochieng, W.Y., 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, 312 – 328.

Schüssler, N., Axhausen, K.W., 2009. Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT). Working Paper 589. ETH Zürich. Zürich.

Spiegel, M., 1968. Mathematical Handbook of Formulas and Tables. Schaum Outline Series.

Velaga, N., Quddus, M., Bristow, A., 2009. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. Transportation Research Part C: Emerging Technologies 17, 672–683.

Wei, H., Wang, Y., Forman, G., Zhu, Y., Guan, H., 2012. Fast Viterbi Map Matching with Tunable Weight Functions, in: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA. pp. 613–616.

Weisstein, W., E., 1999. Normal Distribution.

Zhou, J., Golledge, R., 2006. A Three-step General Map Matching Method in the GIS Environment: Travel/Transportation Study Perspective. International Journal of Geographical Information System 8, 243–260.

Appendices

A. Symbols Used

Symbols for which the scope is limited to a small context (a particular sentence, paragraph or non-subdivided section) are not mentioned here.

Table 8: Symbols Used

Symbol	Meaning
$\langle x, y \rangle$	Ordered pair of elements x and y
$\overline{a}$	accuracy of a GPS device as specified by the manufacturer
δ_n^s	Nominal value for the GPS recording period duration.
$\overline{\delta}$	Expected upper bound for the recording period duration. This applies to in the case where no recordings are lost and hence reflects the error on the timing.
$\Delta(p_0, p_1)$	Euclidean distance between two points $p_0 = \langle x_0, y_0 \rangle$ and $p_1 = \langle x_1, y_1 \rangle$
$\Delta(p, l)$	Shortest (Euclidean) distance between a point $p = \langle x, y \rangle$ and the geometry of link l
G_i^A	Graph that represents the *GPS record assignment state* for a particular subtrace τ_i
G^B	*PeriodBoundaryLinkGraph*, acyclic digraph. It specifies the likelihood for a link to have been used at the end of period p_i as a function of similar likelihoods for the previous period p_{i-1}
G^U	Graph in which each vertex $\langle l, p \rangle$ represents the presence of the moving object on a particular network link l in period p, the *ChronoLinkMatchGraph*
G^T	Transportation graph representing the road network
$G_p^{T,S}$	$G_p^{T,S} \subseteq G^T$: Subnet to search (SNTS) for period p
G_p^T	$G_p^T \subseteq G^T$: Subnetwork constituted by the road network links used in period p
ℓ	Likelihood
$\mathcal{L}$	Log likelihood: $\mathcal{L}(x) = \ln(\ell(x))$
$\langle l, p \rangle$	Link-period pair
L_p^M	Matched link set (MLS) for period p
L_p^{En}	Set of *periodEntryLinks* for period p
L_p^{Ex}	Set of *periodExitLinks* for period p
L_p^{Tx}	Set of *periodTransferLinks* for period p
$\overline{N}_e$	Maximum number of consecutive erroneous recordings that needs to be supported by the map matcher without causing trouble
N_1	Minimum number of consecutive recordings that is assumed to contain at least one correct recording: $N_1 = \overline{N}_e + 1$
N_p	Size of the partition of the trace by determining CMLS-MP.
p	Period of time
p_i	i-th period of time in a partition of chronologically ordered GPS recordings (where each part is a contiguous subsequence τ_i)
P	Paths in a graph.
$\mathcal{P}$	Set of paths in a graph.

Continued on next page...

Table 8 – Continued

Symbol	Meaning
R_M^E	Radius for circular area used for *extended* link matching (selection) in order to determine the subnetwork to search (SNTS) for a GPS record *beyond* the head of the trace
R_M^I	Radius for circular area used for *initial* link matching (selection) in order to determine the subnetwork to search (SNTS) for a GPS record belonging to the head of the trace
$\sigma_{x,y}$	Standard deviation for the normally distributed error on the GPS x and y coordinates expressed in meters and derived from the stated device accuracy $\overline{a}$
$\mathcal{T}$	Trace, chronologically ordered set (sequence) of all GPS recordings for a movement
τ_i	i-th part (contiguous subsequence) of trace τ (i is the offset), corresponds to p_i
$\overline{v}$	Overall maximum value for speed
W	Set of walks (link sequences) output by the map matching procedure for a given trace of GPS recordings

B. Abbreviations Used

Table 9: Abbreviations Used

Acronym	Expansion
CMLS-MP	Complete Matched Link Set for Maximal Period
MLH	Maximum likelihood
MLS	Matched Link Set
SNTS	Sub Network To Search