International Journal of Geo-Information

MDPI

Article

Map-Matching Error Identification in the Absence of Ground Truth

Subhrasankha Dey *, Martin Tomko and Stephan Winter

Department of Infrastructure Engineering, The University of Melbourne, Parkville, VIC 3010, Australia
* Correspondence: deys@student.unimelb.edu.au

Abstract: Map-matching of trajectory data has widespread applications in vehicle tracking, traffic flow analysis, route planning, and intelligent transportation systems. Map-matching algorithms snap a set of trajectory points observed by a satellite navigation system to the most likely route segments of a map. However, due to the unavoidable errors in the recorded trajectory points and the incomplete map data, map-matching algorithms may match points to incorrect segments, leading to map-matching errors. Identification of these map-matching errors in the absence of ground truth can only be achieved by visual inspection and reasoning. Thus, the identification of map-matching errors without ground truth is a time-consuming and mundane task. Although research has focused on improving map-matching algorithms, to our knowledge no attempts have been made to automatically classify and identify the residual map-matching errors. In this work, we propose the first method to automatically identify map-matching errors in the absence of ground truth, i.e., only using the recorded trajectory points and the map-matched route. We have evaluated our method on a public dataset and observed an average accuracy of 91% in automatically identifying map-matching errors, thus helping analysts to significantly reduce manual effort for map-matching quality assurance.

Keywords: trajectory data; map-matching; error identification; unsupervised classification

Citation: Dey, S.; Tomko, M.; Winter, S. Map-Matching Error Identification in the Absence of Ground Truth. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 538. https://doi.org/10.3390/ijgi11110538

Academic Editors: Hartwig H. Hochmair and Wolfgang Kainz

Received: 10 August 2022
Accepted: 25 October 2022
Published: 27 October 2022

1. Introduction

The map matching of trajectory data is essential for many intelligent transportation applications, such as vehicle tracking, traffic flow analysis, detection of transport modes, and route planning [1–6]. Trajectory data consist of geolocation points recorded on the basis of global navigation satellite systems (GNSS) by a tracking application on a vehicle or any other object, and map-matching is applied to trajectory data if the movement is along a known mobility network [7]. Map-matching algorithms determine the most likely segment of the mobility network for each GNSS point in the trajectory data [5,8,9].

Existing map-matching algorithms consider numerous approaches to handle map-matching errors [10,11]. However, they are still prone to errors due to many reasons, such as the inevitable occurrence of systematic GNSS errors, including blocked signals and multipath effects (measurement error) [4,12,13], low sample rate GNSS trajectories (sampling error) [10], limitations in the efficiency of online map-matching algorithms, incomplete map data [14], matching errors at junctions [11], and matching to wrong mobility networks [8,10–12].

The above-mentioned applications require either online or offline map-matching algorithms [6]. Online map-matching algorithms produce the most likely mobility network segment as soon as the GNSS point is recorded [15,16]. On the contrary, offline algorithms use an already recorded set of GNSS points to identify the path traversed by the moving object in hindsight [5,6]. This paper is concerned with offline map-matching. In this case, map-matching error identification would be straightforward in the presence of ground truth trajectory data (e.g., the mode and the route of the travel). However, the typical absence of ground-truth data makes it harder to identify and quantify map-matching errors.

Currently, offline map-matching is open to visual inspection, which enables a visual process of validation by human reasoning. Using this process for identifying map-matching errors is a laborious task when large numbers of trajectories are processed.

Thus, for this case of the absence of ground truth, we first define a **map-matching error**: *When a map-matched segment, incorporated in a map-matched route, reveals an unrealistic travel behavior, then matching this segment is considered to be an error.* Consequently, *map-matching errors can be quantified by counting the number of incorrectly (or unreasonably) matched segments*. With these two definitions at hand, we propose the following research question: *How can we automatically identify and quantify map matching errors when ground truth is not available?*

We present a scalable automatic error identification technique that quantifies the map-matching error by counting the number of incorrectly (or unreasonably) matched segments. The automatic error identification is implemented based on an graph-theoretic unsupervised learning approach such that no human intervention is required. The proposed methodology is implemented as a component within a Python-based open-source tool that also allows one to interactively analyze visually the map-matched route of a trajectory point set (Figure 1). Due to the lack of ground truth data, the visual analysis allows one to establish this ground truth, and to use this ground truth for a performance analysis of our novel technique.

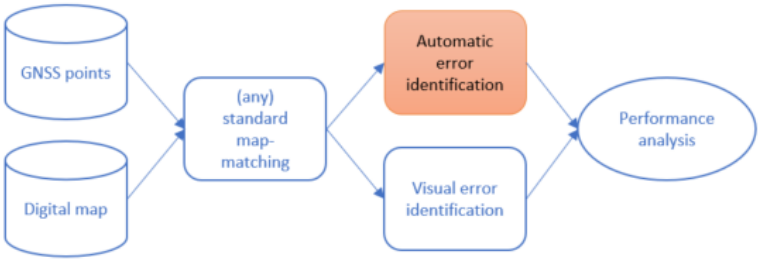

Figure 1. Proposed methodology for map-matching error identification in the absence of ground truth.

The implemented tool sketched in Figure 1 can function as a fully autonomous tool when the goal is to save human labor. However, a guided visual inspection to automatically identify map-matching errors can further improve the quality of the automatic process. The major contribution of this article, however, is the automatic algorithm for offline identifying errors in mapping matching. The algorithm is scalable because it does not require human intervention. Only for quality checks or performance measurements is a visual analytics component available.

The rest of the paper is structured as follows. Section 2 describes current map-matching techniques. Section 3 describes the relevant theory regarding errors in map-matching. Section 4 presents our methodology for identifying incorrectly mapped GNSS points. Section 5 describes the techniques and the experimental results to validate our methodology and measure its performance. Conclusions are provided in Section 6.

2. Literature Review

There are two main approaches for map matching in the literature: (i) geometry-based approaches and (ii) statistics-based approaches.

2.1. Geometry Based Map-Matching

Geometry-based approaches perform map-matching based on certain geometric properties (e.g., closeness) of the GNSS points and the shape of the road segments. Popular approaches are point-to-point map-matching (matching GNSS points to the nearest node of a road network) [17], point-to-curve map-matching (matching the nearest point along with a road segment) [18], curve-to-curve map-matching (matching a collection of points along with a road segment) [19], and Fréchet-distance-based matching [20] (the recorded GNSS points are matched onto the road segments of nearest Fréchet distance) [1,21–23].

The Fréchet-distance-based approach produces better results than the other geometric map-matching method [3]. However, the Fréchet-distance-based approach assumes that the complete trajectory is recorded before the matching process. Hence, this approach is unable to map match in real time [1]. Furthermore, geometry-based map-matching has limitations at road junctions and often produces incorrectly matched segments [10].

2.2. Statistic-Based Map-Matching

In statistic-based approaches, map-matching accuracy has been improved using different statistical models, e.g., the hidden Markov model (HMM) [5,10,13]. The purpose of the HMM is to match each location measurement with the most probable road segment by finding the most probable route in the road network [5,13]. The HMM takes into account the probabilities that govern the state measurements and the probabilities that govern the transitions between states (road segments) at each time [5,13]. The HMM calculates the emission probabilities by modeling the measurement noise and the transition probabilities by modeling the distance between the GNSS measurements and the probable route [5,7,9,24]. The Viterbi algorithm is used to calculate the best route through the HMM lattice [5,13,24,25]. Alternatively, using a bidirectional version of Dijkstra's algorithm leads to similar precision [9]. HMM-based map matching is also used in online map-matching [15,16,26]. Improvements in HMM-based map matching have been made using a segment-based hidden Markov model in denser road networks [13]. Another approach focuses on the junctions where the algorithm can map the wrong road segment [8].

Another major challenge, often unaddressed, is the unraveling of map matching and transport-mode detection [4,12]. According to the current literature, mode detection and map matching mainly consist of two steps [12]: (1) GNSS points are arranged according to multiple unimodal segments, and after that (2) map-matching and mode detection for each unimodal segment are done separately. When unimodal segments are not correctly identified by a mode-detection algorithm, the GNSS points are matched with a wrong segment of the mobility network [8,12]. Furthermore, mode segmentation does not typically consider travel logic; for example, the transition between two vehicle modes cannot occur without walking [4]. Mode detection algorithms perform poorly depending on the strength of the GNSS signal and the sparsity of the collected data [4,12]. Thus, GNSS points unlabeled with travel modes and induced with systematic error leads to erroneous map-matching [8].

In summary, all existing algorithms may incorrectly map some GNSS points onto a wrong road segment due to many factors, e.g., measurement errors in GNSS points, low sampling rates of GNSS points, limitations in computational resources for online map-matching, and complicated topology at road junctions [10]. The challenge of avoiding map-matching errors has not been systematically solved to date. Recent map-matching methods have achieved up to 92% precision for a high sample rate (under 1 min) and up to 82% precision for a low sample rate (one to two minutes) of recorded GNSS points, respectively [10]. Therefore, there remains the possibility of a residual map-matching error due to the stochastic approaches of existing methods [11]. To the best of our knowledge, no method has been proposed to identify these remaining map-matching errors. To overcome the above -entioned challenge in error identification in map-matched segments, we propose a reasoning-based approach to identify and quantify map-matching errors.

3. Concepts of Map-Matching Error Identification

In this section, we first discuss how the residual map-matching errors can be characterized (Section 3.1). This characterization allows then to formalize the quantitative error in a map-matched trajectory point series (Section 3.2) and to introduce an edit distance for this quantification (Section 3.3).

3.1. Qualitative Error in Map-Matching

Qualitative error in map-matching requires a specification of stationarity; e.g., is a car stopping at a red light interrupting its movement between two stationary activities? This requires consideration of the context of the movement, that is, the travel mode of the movement before and after a stop and the mode-specific thresholds of acceptable duration of these stops. Most frequently, these trajectories are sampled at regular time intervals, although sampling at regular distance intervals and irregular sampling is possible as well. Regular sampling strategies can also show gaps in their recordings. We have considered the following categories of unrealistic travel behavior to be reflected by qualitative errors in a map-matching algorithm:

- Infinite velocity.
- Unrealistic acceleration.
- Presence in multiple locations simultaneously.

Unrealistic travel behavior in a map-matched segment occurs due to three main reasons: (i) measurement error in collected GNSS points (since collected GNSS points are exposed to measurement error, true positions remain unknown, resulting in incorrectly matched segments in the road network [27]); (ii) complex topology and incomplete map information in the road network [11], leading to erroneous transition probability calculations in both statistics and distance-based approaches [5]; and (iii) mismatch of travel mode and road network type (e.g., map-matching of a car route on a public transport network) [12]. As a result of map-matching, in a map-matched route, there will be two types of map-matched segments: (i) correctly map-matched segments (reflects realistic travel behavior) and (ii) incorrectly map-matched segments (reflects unrealistic travel behavior).

3.2. Quantitative Error in Map-Matching

We propose quantifying qualitative mapping errors after a mapping algorithm produces the estimated route and map-matched segments. Let a *trajectory* T contain a number d of recorded GNSS data points, $d \geq 2$, such that the jth data point ($d \geq j \geq 1$) contains the tuple $< x_j, y_j, t_j >$, where x_j and y_j are the coordinates of the location of the jth point (e.g., longitude and latitude, respectively), and t_j is the timestamp of the location record. Let E be the set of edges and V be the set of vertices in the directed graph (digraph) of the road network $G = (V, E)$. Then:

$$E \subseteq \{\{p, q\} \mid p, q \in V \text{ and } p \neq q\} \tag{1}$$

Here, edges represent road segments on a map, a possible candidate segment of map-matching. We will refer to the term edge in the context of a graph and to a segment in the context of map data. The ground truth route of a trajectory data T is a sequence of connected road segments of the traveled route on the map. For this, we assume that the map is complete.

Let R be a sequence of the ground-truth map segments of the trajectory data T. Hence, R contains a sequence of connected road segments, i.e., map segments, which are expected to be matched correctly with GNSS points recorded on the same map segment. Let R have $n \geq 1$ edges, and let $V_R \subset V$ be a subset of vertices of G such that there exists an induced connected subgraph G_R whose vertex set is V_R and whose edge set $E_R = \{r_1, \cdots, r_n\} \subseteq E$ such that the vertex connectivity $\kappa(G_R) = 1$—i.e., G_R is 1-vertex-connected. The vertex cut or separating set of G_R contains at least one vertex whose removal renders G_R disconnected. Thus, the ground truth route $R = (r_1 \rightarrow r_2 \rightarrow \cdots r_n)$ is a *walk* in the digraph G_R.

Let the map-matched route $\hat{R}$ for the trajectory T consist of a set of $m \geq 1$ edges: $\{\hat{r}_1, \cdots, \hat{r}_m\} \subseteq E$. Thus, $\hat{R} = (\hat{r}_1 \rightarrow \hat{r}_2 \rightarrow \cdots \hat{r}_m)$. Map matching ensures that for a set of GNSS data points given by coordinate $\{(x_j, y_j)\}$ there exists a matching set of road segments $\{\hat{r}_i = (p, q)_i\}$ where $(p_i, q_i) \in V$ and $(p, q)_i \in E$. The set of location points has a subjective functional relationship with the set of map-matched segments such that multiple points can be matched with one and the same road segment.

For a correctly map-matched route, $\hat{R} = R$ will have a finite sequence of map-matched segments such that any ith$(i \geq 2)$ map-matched segment $\hat{r}_i$ (with vertices p_i and q_i) will have an incident edge $\hat{r}_{i-1}$ where:

$$q_{i-1} = p_i \tag{2}$$

A map-matching error occurs if:

$$\hat{R} \neq R \tag{3}$$

Equation (3) implies that given ground truth R, any difference of $\hat{R}$ can be identified and labeled as a map-matching error. Equation (3) further refers to the fact that $\hat{R} \neq R$ follows $q_{i-1} \neq p_i$ in a map-matching error.

3.3. Edit-Distance-Based Quantification

Edit distance is a string metric that measures the minimum number of operations required to transform one string into the other [28]. The identified map-matching errors can be quantified by calculating the edit distance between R and $\hat{R}$, i.e., by quantifying the single-segment edits (insertions, deletions, or substitutions) between the sequences in R and $\hat{R}$. We apply the Levenshtein distance, among various edit distances, because it caters for these three edit operations [28].

The Levenshtein distance between two *walks* R and $\hat{R}$ of lengths $|R|$ and $|\hat{R}|$, respectively, can be given by $\text{lev}(R, \hat{R})$, where

$$\text{lev}(R, \hat{R}) = \begin{cases} |R| & \text{if } |\hat{R}| = 0, \\ |\hat{R}| & \text{if } |R| = 0, \\ \text{lev}\left(\text{tail}(R), \text{tail}(\hat{R})\right) & \text{if } R[0] = \hat{R}[0] \\ 1 + \min \begin{cases} \text{lev}\left(\text{tail}(R), \hat{R}\right) \\ \text{lev}\left(R, \text{tail}(\hat{R})\right) \\ \text{lev}\left(\text{tail}(R), \text{tail}(\hat{R})\right) \end{cases} & \text{otherwise,} \end{cases} \tag{4}$$

where the tail of a walk is the *walk* shortened by its first edge. The first edge, or head element, of a walk R is $R[0]$. Thus, the Levenshtein distance evaluates the number of mismatches between ground truth and the map-matched route.

We categorize four common types of error in map-matching algorithms (Figures 2–5), all of which reveal unrealistic travel behavior in a map-matched segment. In the figures, the recorded GNSS points are shown as blue dots, and map-matched segments are shown in green lines. Each matched segment has a unique segment identification number in the travel sequence, shown in black. The identified common categories of map matching errors are the following:

1. *Cat-I error*: This type of error occurs when the incorrectly matched segments are unidirectional or hanging from the actual route (Figure 2). Referring to Equation (2), *Cat-I error* occurs in the ith map-matched segment $\hat{r}_i : (p_i, q_i)$ of $\hat{R}$ when the following conditions are true:

$$\begin{aligned} q_i &\neq p_{i+1} \\ p_i = p_{i+1} &= q_{i-1} \end{aligned} \tag{5}$$

   Such map-matched segments suggest the traveler needs to jump from the end of the segment (q_i) back to its start (p_i) in order to come back to the original route. An illustrative example of *Cat-I error* is shown in Figure 2.

2. *Cat-II error*: This type of error occurs when isolated lanes are matched with the GNSS points (Figure 3). *Cat-II error* occurs in the ith map-matched segment $\hat{r}_m : (p_m, q_m)$ of $\hat{R}$ when the following conditions are true:

$$\begin{aligned} q_i &\neq p_{i+1} \\ p_i &\neq q_{i-1} \\ \hat{r}_i &\notin R \end{aligned} \tag{6}$$

Such map-matched segments suggest that the traveler needs to have infinite velocity at that isolated segment. One illustrative example of a *Cat-II error* is shown in Figure 3. Note that the underlying movement behavior is perfectly legal and physically possible.

3. *Cat-III error*: This type of error occurs when there are discontinuities in the matched route due to sparseness of the recorded data points (Figure 4). *Cat-III error* occurs in the ith map-matched segment $\hat{r}_i : (p_i, q_i)$ of $\hat{R}$ when the following conditions are true:

$$\begin{aligned} q_i &\neq p_{i+1} \\ p_i &\neq q_{i-1} \\ \hat{r}_i &\in R \end{aligned} \tag{7}$$

Such map-matched segments suggest that the traveler needs to have infinite velocity while transiting from one segment to the very next disconnected segment. The illustration of *Cat-III error* is shown in Figure 4.

4. *Cat-IV error*: Sometimes, map matched segments are ambiguously selected by the algorithm due to measurement error in the recorded points (Figure 5). These are special cases of *Cat-II* and *Cat-III errors*:

$$\begin{aligned} q_i &\neq p_{i+1} \\ p_i &\neq q_{i-1} \end{aligned} \tag{8}$$

Equation (8) is independent of condition $\hat{r}_i \in R$ or $\hat{r}_i \notin R$. Such map-matched segments suggest the traveler needs to be in both the segments at the same time as if there is a simultaneous presence at two different lanes. This type of error is a *Cat-IV error*, as shown in Figure 5.

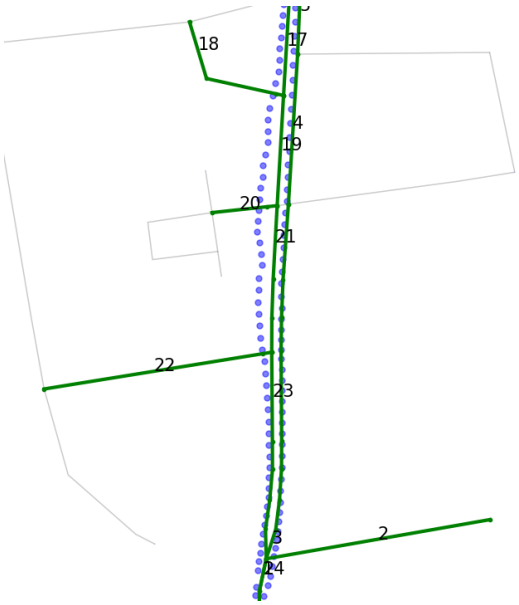

Figure 2. *Cat-I error*: hanging segments.

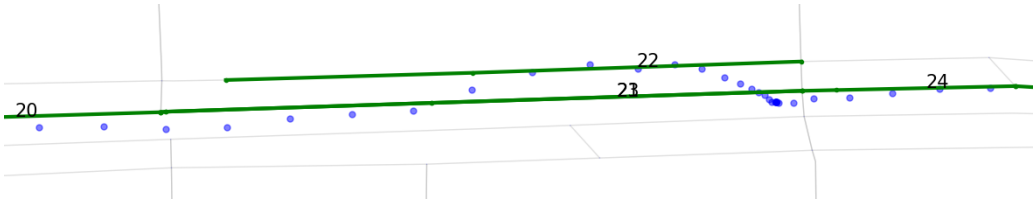

Figure 3. *Cat-II error*: isolated lane selection.

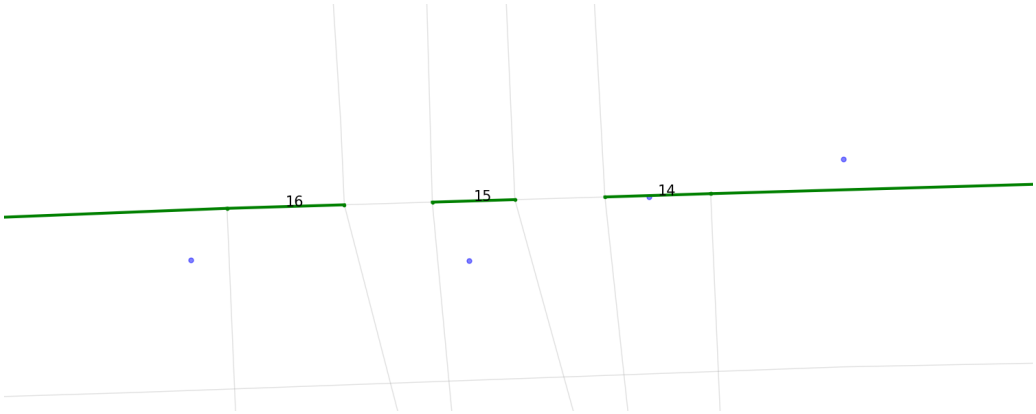

Figure 4. *Cat-III error*: discontinuity in the route.

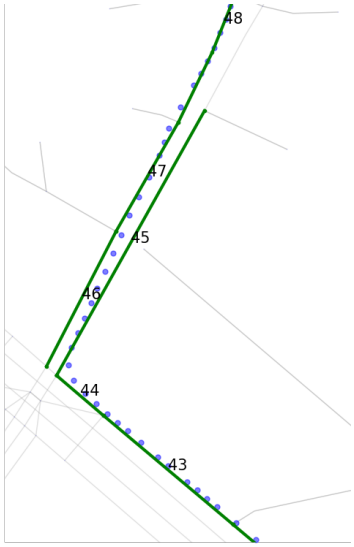

Figure 5. *Cat-IV error*: ambiguous lane selection.

However, in the absence of a ground truth map-matched route (R), map-matching errors (e.g., *Cat-I–IV errors*) can only be identified by visual analysis of $\hat{R}$ based on common-sense reasoning. With knowledge of R and $\hat{R}$, the Levenshtein distance can be calculated using Equation (4).

4. Computing the Map-Matching Error Estimation

We propose an autonomous two-step methodology to estimate whether a map-matched segment $\hat{r}_i$ is likely to be erroneous or not. We label each map-matched segment in $\hat{R}$ as either correctly matched or incorrectly matched. *Step 1*: estimating the effect of measurement noise in the recorded GNSS points on map-matched segments. *Step 2*: identifying unrealistic travel behavior due to unreasonable sequences in the map-matched route. Combining both *Step 1* and *Step 2*, we label each map-matched segment either as correctly

matched or incorrectly matched. We utilize the Levenshtein edit distance as a quantitative measure of this error.

4.1. Step 1: Identification of Map-Matched Segments Affected by Noisy GNSS Points

In Step 1, our objective is to detect whether a segment matched with the map is influenced by the measurement error at the GNSS points. All recorded GNSS points have some measurement error. For some points, the measurement error is within an acceptable quality for map-matching. The rest of the points are not within the granularity of map-matching. The threshold value for the acceptability of the induced measurement error is unknown for classifying the GNSS points. Hence, we developed a new unsupervised classification technique to classify the recorded GNSS points into two categories: (i) acceptable erroneous points and (ii) unacceptable erroneous points (or decidedly erroneous point). We perform a binary labeling of each GNSS point followed by classification. Then, we measure the likelihood of a segment to be influenced by each labeled GNSS point. Figure 6 presents a flow diagram to identify map-matching errors and the responsible erroneous GNSS points.

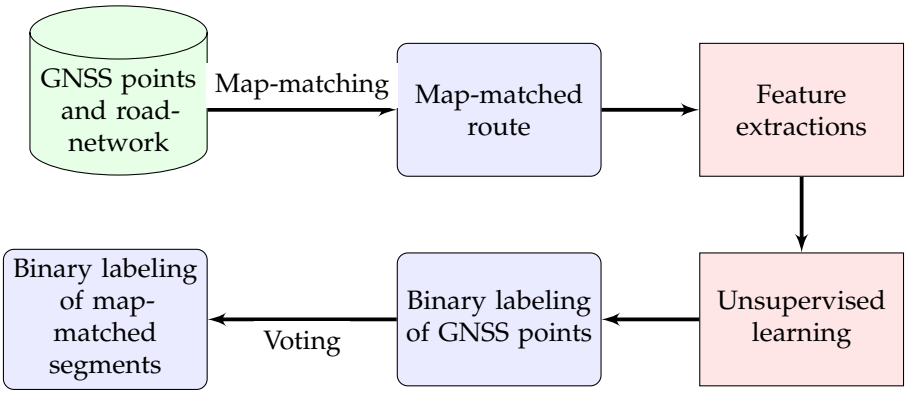

Figure 6. *Step 1* of the proposed methodology for map-matching error identification.

For the figure, a database of raw GNSS points $\{(x, y, t)\}$ and a road network graph G were used by a map-matching algorithm to generate the sequences of matched segments $\hat{R}$. We define the relevant features that are obtained using the recorded data point (x_j, y_j, t_j) and the corresponding map-matched segment $\hat{r}_j$. These feature values will be used for unsupervised binary classifications of GNSS points $\{(x, y, t)\}$ to filter out unacceptable erroneous points from less erroneous points. Each GNSS point has a corresponding map-matched segment, and there exists a surjection mapping between the set of GNSS points and the set of map-matched segments: multiple GNSS points may be matched to a single segment. We performed a binary labeling of each map-matched segment on the route depending on the statistical mode of the labeled GNSS points matched to that segment. Thus, we achieve *Step 1* of identifying incorrectly matched segments.

4.1.1. Feature Engineering

In the HMM-based model, the transition and emission probabilities of each map-matched segment depend on (i) the orthogonal distance of a GNSS point from the set of probable segments and (ii) the probability of measurement error in each GNSS point. Hence, the categorization of each GNSS point is achieved based on two measures (or features): (i) how far a GNSS point is away from the map-matched segment and (ii) how likely the GNSS point is to be corrupted. Features are individual measurable properties of an event being observed, here in the process of map-matching. Each feature is derived from a GNSS point (in T) and the corresponding mapped segment. Therefore, each feature of the jth point is a function of (p, q, c_j) where $c_j = (x_j, y_j)$. We have defined two features for the classifications discussed in the following subsections: (i) orthogonal distance and (ii) estimated noise.

Orthogonal Distance

We can write a straight line passing through the points p, q as: $\mathbf{f}_i = \mathbf{p} + K(\mathbf{p} - \mathbf{q})$, where bold means a vector and K is a constant. Then, the orthogonal distance of the jth GNSS point from $\mathbf{f}_i$ can be calculated as:

$$o_j = \frac{\mathbf{f}_i \cdot \mathbf{c}_j}{\|\mathbf{c}_j\|} \tag{9}$$

Estimated Noise

We use a Kalman filter [29] to estimate the noise of the recorded GNSS points induced by measurement error. Noise can be determined if we can estimate the true position of a recorded GNSS point. A typical Kalman filter estimates the next true position based on the current true position and the current estimated position, using two processes, a prediction process, and a correction process [29]. The prediction process at the jth timestamp is governed by the following equations.

$$\hat{\mathbf{X}}_j^- = \mathbf{A}_j \hat{\mathbf{X}}_{j-1} \tag{10}$$

$$\mathbf{P}_j^- = \mathbf{A}_j \mathbf{P}_{j-1} \mathbf{A}_j^\top + \mathbf{Q} \tag{11}$$

where $\mathbf{X}$ is the state vector; $\hat{\mathbf{X}}$ the estimated state vector; $\mathbf{P}_j$ the variance-co-variance matrix for jth state; $\mathbf{P}_0$ the initial variance-co-variance matrix; $\mathbf{Q}$ the process co-variance, i.e., the Gaussian noise in prediction $N(0, q^2)$, where q is the standard deviation of the process error [30]; and $\mathbf{A}_j$ is the time transition matrix. If $dt_j = t_j - t_{j-1}$, then:

$$\mathbf{X}_j = \begin{bmatrix} x_j \\ y_j \\ \dot{x}_j \\ \dot{y}_j \end{bmatrix} \tag{12}$$

$$\mathbf{A}_j = \begin{bmatrix} 1 & 0 & dt_j & 0 \\ 0 & 1 & 0 & dt_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$\mathbf{P}_j = \begin{bmatrix} {\sigma_x}^2 & \sigma_{xy} & \sigma_{x\dot{x}} & \sigma_{x\dot{y}} \\ \sigma_{yx} & {\sigma_y}^2 & \sigma_{y\dot{x}} & \sigma_{y\dot{y}} \\ \sigma_{\dot{x}x} & \sigma_{\dot{x}y} & {\sigma_{\dot{x}}}^2 & \sigma_{\dot{x}\dot{y}} \\ \sigma_{\dot{y}x} & \sigma_{\dot{y}y} & \sigma_{\dot{y}\dot{x}} & \sigma_{\dot{y}}^2 \end{bmatrix} \tag{14}$$

The correction process at the jth state is controlled by the following set of equations.

$$\mathbf{K}_j = \frac{\mathbf{P}_j^- \mathbf{H}^\top}{\mathbf{H}\mathbf{P}_j^- \mathbf{H}^\top + \mathbf{M}_e} \tag{15}$$

$$\mathbf{P}_j = (\mathbf{I} - \mathbf{K}_j \mathbf{H}) \mathbf{P}_j^- \tag{16}$$

$$\hat{\mathbf{X}}_j = \hat{\mathbf{X}}_j^- + \mathbf{K}_j [\mathbf{Z}_j - \mathbf{H}\hat{\mathbf{X}}_j^-] \tag{17}$$

where $\mathbf{K}_j$ is the Kalman gain matrix in the jth state, $\mathbf{H}$ the measurement matrix for the observation process, $\mathbf{Z}_j$ the observations at the jth state, and $\mathbf{M}_e$ the co-variance matrix of the measurement error, that is, the Gaussian noise in the measurement $N(0, \sigma^2)$, where σ is the standard deviation of the measurement error [14]. We will get the estimated position $\hat{c}_j = (\hat{x}_j, \hat{y}_j)'$ from $\hat{\mathbf{X}}_j$. Thus, we derived our fourth feature as the estimated noise in c_j as:

$$\hat{\eta}_j = \|\mathbf{c}_j - \hat{\mathbf{c}}_j\| \tag{18}$$

4.1.2. Unsupervised-Learning-Based Classification

Once we extracted the feature values of each GNSS point in the trajectory data T, we prepared a training dataset $\mathbf{U}_T$ for the classification of J points. $\mathbf{U}_T$ has 2 columns and J rows. Each jth row of the training dataset $\mathbf{U}_T$ has feature values $\{o_j, \hat{\eta}_j\}$ and is unlabeled (that is, we do not know the acceptable error in GNSS points). Hence, we used unsupervised learning to classify the GNSS points in $\mathbf{U}_T$ using a Gaussian mixture model [31]. Gaussian mixture models (GMMs) are probabilistic models, where each cluster corresponds to a normal multivariate probability distribution. The GMM clustering algorithms define soft boundaries. The aim of the proposed GMM is to perform binary clustering of GNSS points based on some features to separate unacceptable erroneous GNSS points from acceptable erroneous GNSS points. The cluster centers will reveal the binary labels of the recorded GNSS points, 1 for no acceptable erroneous points and 0 for acceptable erroneous points. We identified the GNSS points associated with each map-matched segment. A map-matched segment will then be assigned a label based on a voting-based approach: the labels with the highest majority win. Each segment in $\mathbf{U}_T$ were labeled either 0 (correctly matched) or 1 (incorrectly matched), based on the label of the majority of the GNSS points associated with the segment.

4.2. Step 2: Identification of Unrealistic Travel Behavior Due to Topological Irregularities

This step aims to estimate the ground truth R from $\hat{R}$. A correctly map-matched route (when $R = \hat{R}$) has the properties of a walk in a digraph G. Therefore, $\hat{R}$ cannot be disconnected, cannot contain multiple components, and cannot have a unidirectional segment (Equation (5)) or a unidirectional claw in G. In graph theory, a claw S_3 is a star with three edges. S_3 is a tree with one internal node and 2 leaves, i.e., a complete bipartite graph $S_{1,3}$. The presence of multiple components and/or unidirectional claws on a map-matched route $\hat{R}$ is addressed as topological irregularities in a travel route. Hence, in *Step 2*, we aim to detect whether a map-matched segment is causing unrealistic travel behavior due to the self-generated topological irregularities in the map-matched route. We propose labeling segments erroneously matched by the following methods:

4.2.1. Connected Components Analysis

A connected component is a maximally connected subgraph of a graph. We looked for the number of connected components in the map-matched route $\hat{R}$. In this way, we can determine any discontinuities in the map-matched route and the disconnected map-matched segments responsible for *Cat-II*, *Cat-III*, and *Cat-IV error*.

4.2.2. Claw Detection

In graph theory, a star with three edges is called a *claw*. Since unidirectional hanging segments are responsible for *Cat-I errors*, the ground truth R must be claw-free. After completing *Step 2*, each segment in $\mathbf{U}_T$ will be labeled 0 (correctly matched) or 1 (incorrectly matched). We create a new database $\mathbf{L}_T$ to store the output of *Step 2* after labeling.

5. Performance of the Automatic Map-Matching Error Identification

In this section, we discuss the validation of our method and its performance using GeoLife trajectory data [32]. Since GeoLife data are in principle multi-modal, but labeled with the transportation mode, we selected only the datasets labeled *drive/car* to not confuse separate map-matching challenges. Our experimental dataset T from GeoLife contains 23 datasets with in total 15,047 GNSS points. Additionally, we used the state-of-the-art HMM-based map matching [5], which produced 1296 map-matched segments.

5.1. Establishing Ground Truth from Visual Inspection

Validation and performance assessment of the automatic error identification requires ground truth data for the map-matched segments. Since ground truth of map-matched segment data is not available with large-trajectory datasets [14,32], we apply visual in-

spection of the map-matched segments, applying the criteria of reasonable sequencing described above.

For example, in Figure 7, part of the trajectory data of Dataset 4 is presented, where we can visually identify incorrectly map-matched segments from human reasoning. The discontinuities between Segments 8 and 9 suggest a Cat-III error, and those between Segments 5 and 6 suggest a Cat-IV error, according to Equations (7) and (8). A person would need infinite velocity to travel from Segment 5 to 6, and from Segment 8 to 9.

Once an error is visually identified, we label each erroneously map-matched segment $\hat{r}_e$. Thus, we create a ground truth dataset $\mathbf{G}_T$, where each map-matched segment of T is manually labeled as either 0 (if correctly matched) or 1 (if incorrectly matched).

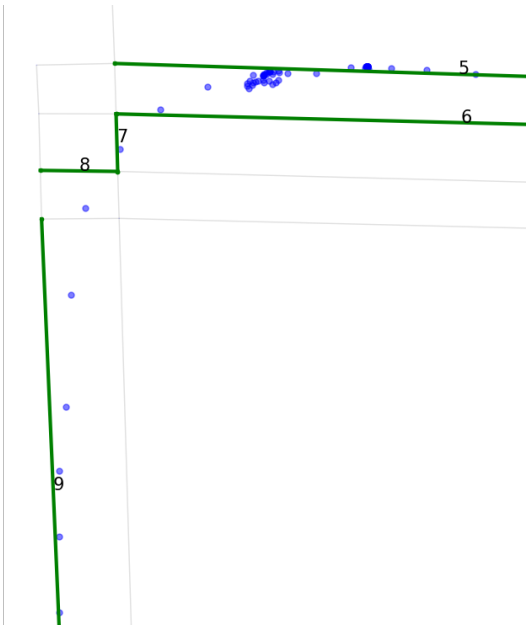

Figure 7. Multiple different types of error observed in Dataset 4.

To realize the visual inspection, we have developed an interactive, map-based tool (Figure 8). The recorded GNSS points are shown as red dots, and map-matched segments are shown as blue markers with popup numbers. The tool tracks user interactions such as in the figure, where the map-matched segment 6388 has been clicked as a ground truth error by a human inspector.

5.2. Using the Ground Truth for Validation and Performance Assessment

Our proposed method labels the map-matched segments as correctly or incorrectly map-matched, and creates $\mathbf{L}_T$. We compared $\mathbf{L}_T$ with the ground truth dataset $\mathbf{G}_T$ and performed an error analysis of our method. For this analysis, we applied the usual error measures:

1. True positive (*TP*): The proposed method correctly identifies an erroneous segment according to ground truth.
2. False positive (*FP*): The proposed method indicates an error when there is no error in the map-matching.
3. False negative (*FN*): The proposed method fails to indicate an observed error in the map-matching.
4. True negative (*TN*): The proposed method correctly labeled a segment as non-erroneous.

Say the ground truth route R has n segments and the map-matched route $\hat{R}$ has m segments, of those, m_C ($\leq m$) segments are correctly matched. Let the observed (ground truth) error in map-matching occur in m^{obs}_{Error} of segments. Then:

$$m^{obs}_{Error} = m - m_C = \text{lev}(R, \hat{R}) \tag{19}$$

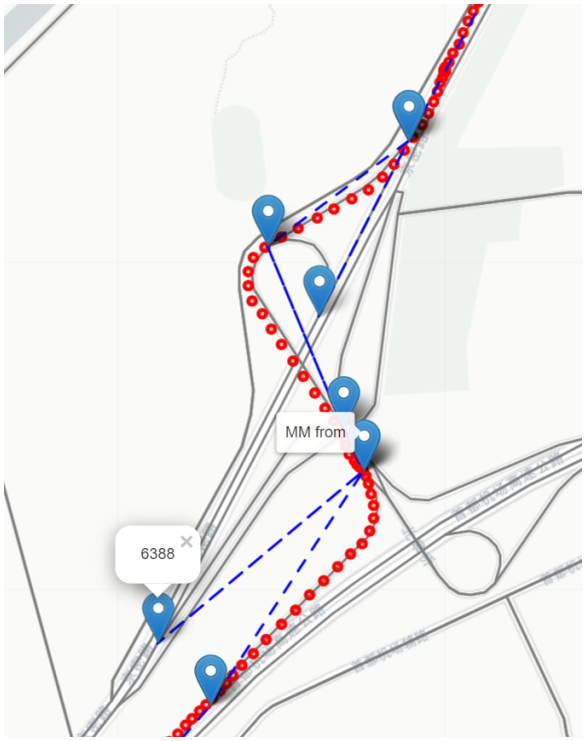

Figure 8. A screenshot of the visual inspection tool.

We have defined two error measures to evaluate the accuracy of a map-matching algorithm: (i) the error percentage of the map-matching algorithm ε_{MM} and (ii) the error percentage of the model outcome ε_{PM}. Let $\hat{m_C}$ be the number of correctly labeled segments. Then:

$$m_{Error}^{est} = m - \hat{m_C} \tag{20}$$

$$\varepsilon_{MM} = \frac{m_{Error}^{obs}}{m} \cdot 100\% \tag{21}$$

$$\varepsilon_{PM} = \frac{|m_{Error}^{obs} - \hat{m_C}|}{m_{Error}^{obs}} \cdot 100\% \tag{22}$$

In Figure 7, Dataset 4 has five map-matched segments; hence, $m = 5$. The *Cat-III error* between Segments 8 and 9, and the *Cat-IV error* between Segments 5 and 6 equate to $m_{Error}^{obs} = 2$. The automatic method identified the *Cat-III error* correctly, but the *Cat-IV error* went undetected. Therefore, $\hat{m_C} = 1$, with one true positive and one false positive. Thus,

$$\varepsilon_{MM} = \frac{2}{5} \cdot 100\% = 40\% \tag{23}$$

$$\varepsilon_{PM} = \frac{2-1}{2} \cdot 100\% = 50\% \tag{24}$$

We will conclude our validation by comparing sensitivity analysis (true positive rate) and specificity analysis (true negative rate).

5.3. Performance Results of Automatic Error Identification

Illustrations of identified errors are presented in Figures 9–12. All blue dots indicate recorded GNSS points. The incorrectly map-matched segments as estimation results are shown as red lines. The correctly estimated map-matched segments are shown with green lines.

- In Figure 9, a *Cat-I error* has been identified and highlighted by a red line with Segments 2, 22, 20, and 18.
- In Figure 10, a *Cat-II error* has been identified and highlighted by a red line with Segment 22.
- In Figure 11, a *Cat-III error* has been identified and highlighted by red lines with Segments 16 and 15 and a green line 14, as it was not detected by our method.
- In Figure 12, a *Cat-IV error* has been identified and highlighted by the red line with Segments 11 and 10.

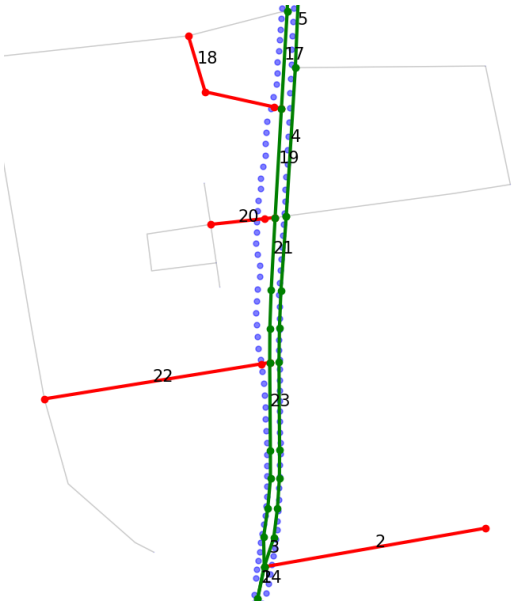

Figure 9. Model outcome for *Cat-I error*: identified hanging segments in Dataset 26.

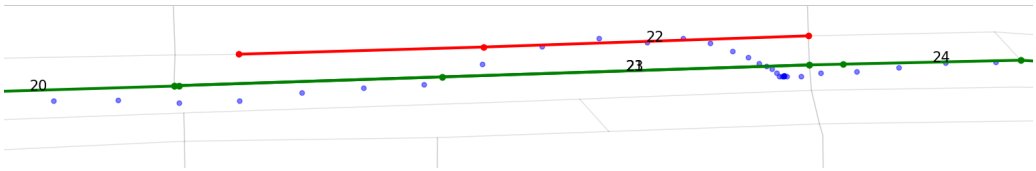

Figure 10. Model outcome for *Cat-II error*: identified isolated lane in Dataset 7.

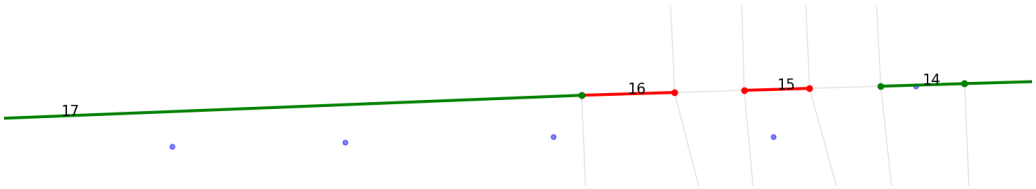

Figure 11. Model outcome for *Cat-III error*: discrete jumps in Dataset 12.

In this way, we have compared the 23 drive-only map-matched trajectories from GeoLife by counting the total ground truth segments (n), the observed errors in map-matching (m^{obs}_{Error}), correctly estimated errors ($\hat{m_C}$), false positives (incorrectly estimated errors), and false negatives (not estimated errors) in the proposed method. Then, we have calculated ε_{MM} and ε_{PM} based on Equations (21) and (22). The results are presented in Table 1, sorted according to the values of ε_{MM} in ascending order.

We have derived a two-dimensional matching matrix (Table 1) to visualise the performance of the proposed methodology. Each row of the matching matrix represents observed instances, that is, the ground truth label of the map-matched segments: (i) without error and (ii) with error. Each column of the matching matrix represents the estimated instances,

that is, the estimated labels of the map-matched segments using the proposed method: (i) segments estimated without error and (ii) segments estimated with error. Thus, a matching matrix evaluates a quantitative measure of mislabeled segments on the map using the proposed methodology. A matching matrix generated for the proposed methodology is presented in Table 2, where P and N in parentheses indicate positive and negative instances, respectively.

Figure 12. Model outcome for *Cat-IV error*: identified ambiguous lane in Dataset 6.

Table 1. Model performance validation with the GeoLife datasets 1–23.

DS	m	m^{est}_{Error}	m^{obs}_{Error}	$\hat{m_C}$ (TN)	FP	FN	TP	ε_{MM}	ε_{PM}
1	91	14	26	13	1	13	64	28.57	50.00
2	31	13	14	12	1	2	16	45.16	14.29
3	16	4	3	3	1	0	12	18.75	0.00
4	30	12	14	9	3	5	13	46.67	35.71
5	19	7	6	3	4	3	9	31.58	50.00
6	91	23	21	19	4	2	66	23.08	9.52
7	91	14	19	12	2	7	70	20.88	36.84
8	5	0	1	0	0	1	4	20.00	100.00
9	59	12	12	10	2	2	45	20.34	16.67
10	54	4	7	2	2	5	45	12.96	71.43
11	20	10	8	7	3	1	9	40.00	12.50
12	60	8	12	5	3	7	45	20.00	58.33
13	75	17	21	16	1	5	53	28.00	23.81
14	85	26	29	26	0	3	56	34.12	10.34
15	150	16	18	13	3	5	129	12.00	27.78
16	89	10	4	4	6	0	79	4.49	0.00
17	26	7	9	4	3	5	14	34.62	55.56
18	43	4	5	4	0	1	38	11.63	20.00
19	34	5	6	5	0	1	28	17.65	16.67
20	62	8	10	7	1	3	51	16.13	30.00
21	94	8	11	8	0	3	83	11.70	27.27
22	21	5	7	5	0	2	14	33.33	28.57
23	50	1	5	1	0	4	45	10.00	80.00

Table 2. Matching matrix generated using the proposed method.

		Estimated		
		Without Error (P)	**With Error (N)**	**Sensitivity (Recall)**
Observed	Without error (P)	$TP = 988$	$FN = 80$	0.93
	With error (N)	$FP = 40$	$TN = 188$	0.18
	Precision	0.96	0.30	

Our proposed method was evaluated based on the following scores, obtained after evaluating the proposed matching matrix: *AUC* (area under the curve)—*ROC* (receiver operating characteristic), true positive rate (*TPR*), true negative rate (*TNR*), positive predictive value (*PPV*), false omission rate (*FOR*), *F*1 score, and overall accuracy (*ACC*), as defined in Equations (25)–(31):

$$AUC = 0.87 \tag{25}$$

$$TPR = \frac{TP}{TP + FN} = 0.93 \tag{26}$$

$$TNR = \frac{TN}{TN + FP} = 0.82 \tag{27}$$

$$PPV = \frac{TP}{TP + FP} = 0.96 \tag{28}$$

$$FOR = \frac{FN}{FN + TN} = 0.30 \tag{29}$$

$$F1 = \frac{2TP}{2TP + FP + FN} = 0.94 \tag{30}$$

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} = 0.91 \tag{31}$$

6. Conclusions

Identifying errors in online and offline map-matching processes is important for related intelligent transportation applications, including driving behavior and traffic flow analyses, identifying changes in the street network, and planning of transport routes. In this paper, we have proposed a methodology to identify those map-matching errors in offline processes. We have discussed the theory of map-matching error identification. We then introduced and formalized a typology of four common types of map-matching errors arising from measurement errors and complex map topology. We proceeded to develop the theory to identify erroneous map-matched segments within map-matched trajectory data based on an unsupervised classification followed by a graph-theoretic approach to spot unrealistic travel behavior. Thus, our research question can be answered positively: We can automatically identify and quantify map matching errors when ground truth is not available. Beyond identification, we can also classify these errors using the proposed classification.

We have validated our methodology using real-world data collected in the GeoLife project. The validation results show that our method has a good accuracy score when map-matching errors are notably present: The method achieves an average accuracy of 91% in automatically identifying map-matching errors. This result indicates that our method can efficiently help the analysts to reduce human efforts in map-matching quality control by a large margin. By the choice of the unimodal trajectory subset of GeoLife, we evaluated data that should already work better with map-matching algorithms (i.e., present a hard baseline), and yet we demonstrated the presence and the ability of our method to identify of significant errors of current map-matching methods. The amount of potential errors that should be detectable using the proposed method is likely to be significantly higher for

multimodal trajectories, observed in multimodal mobility networks. The proposed method and error categorization method remain valid for these trajectories and errors.

As a limitation of our method, we detected erroneously map-matched segments without correction. Although our method contributes a theoretical and practical step forward towards map-matching error identification, further investigations into feature engineering and classification algorithms could overcome current limitations. In future work, it will be interesting to investigate the performance measures of our model on *Cat-I* to *Cat-IV errors* separately. Although the proposed method is applied with offline map-matching, can also be applied in online map-matching (e.g., using a real-time simulation-based approach with Geolife data).

Author Contributions: All research work in this paper, from conceptualization and methodology to validation, has been done by Subhrasankha Dey, under supervision of Martin Tomko and Stephan Winter; writing—original draft preparation, Subhrasankha Dey; review and editing, Martin Tomko and Stephan Winter. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by a University of Melbourne Postgraduate Scholarship.

Data Availability Statement: We have used the GeoLife dataset [32], which is an open, ubiquitous dataset, and the data were collected in the Geolife project (Microsoft Research Asia) from 182 users over a period of more than three years (from April 2007 to August 2012). We have cited relevant papers on this dataset in our article.

Acknowledgments: We are thankful to Anish Agarwal (Department of Civil Engineering, Indian Institute of Technology, Kharagpur, India) for his contributions to the literature review section.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bang, Y.; Kim, J.; Yu, K. An improved map-matching technique based on the Fréchet distance approach for pedestrian navigation services. *Sensors* **2016**, *16*, 1768. [CrossRef] [PubMed]
2. Dey, S.; Winter, S.; Tomko, M. Origin–Destination Flow Estimation from Link Count Data Only. *Sensors* **2020**, *20*, 5226. [CrossRef] [PubMed]
3. Brakatsoulas, S.; Pfoser, D.; Salas, R.; Wenk, C. On map-matching vehicle tracking data. In Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 30 August–2 September 2005; pp. 853–864.
4. Schuessler, N.; Axhausen, K.W. Processing raw data from global positioning systems without additional information. *Transp. Res. Rec.* **2009**, *2105*, 28–36. [CrossRef]
5. Newson, P.; Krumm, J. Hidden Markov map matching through noise and sparseness. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle WC, USA, 4–6 November 2009; pp. 336–343.
6. Roth, J. The Offline Map Matching Problem and its Efficient Solution. In Proceedings of the International Conference on Innovations for Community Services, Vienna, Austria, 27–29 June 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 23–38.
7. Raymond, R.; Morimura, T.; Osogami, T.; Hirosue, N. Map matching with hidden Markov model on sampled road network. In Proceedings of the 21st International Conference on Pattern Recognition, Tsukuba, Japan, 11–15 November 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2242–2245.
8. Qi, H.; Di, X.; Li, J. Map-matching algorithm based on the junction decision domain and the hidden Markov model. *PLoS ONE* **2019**, *14*, e0216476. [CrossRef] [PubMed]
9. Koller, H.; Widhalm, P.; Dragaschnig, M.; Graser, A. Fast hidden Markov model map-matching for sparse and noisy trajectories. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2557–2561.
10. Hsueh, Y.L.; Chen, H.C. Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions. *Inf. Sci.* **2018**, *433*, 55–69. [CrossRef]
11. Xu, Y.; Lu, C.; Kaw, S.; Wolf, D. *Improving Uber's Mapping Accuracy with CatchME*; Technical Report; Uber Engineering: San Francisco, CA, USA, 2019.
12. Chen, J.; Bierlaire, M. Probabilistic multimodal map matching with rich smartphone data. *J. Intell. Transp. Syst.* **2015**, *19*, 134–148. [CrossRef]
13. Cui, G.; Bian, W.; Wang, X. Hidden Markov map matching based on trajectory segmentation with heading homogeneity. *GeoInformatica* **2021**, *25*, 179–206. [CrossRef]
14. Dey, S.; Winter, S.; Goel, S.; Tomko, M. Identification of parking spaces from multi-modal trajectory data. *Trans. GIS* **2021**, *6*, 3088–3118. [CrossRef]

15. Goh, C.Y.; Dauwels, J.; Mitrovic, N.; Asif, M.T.; Oran, A.; Jaillet, P. Online map-matching based on hidden markov model for real-time traffic sensing applications. In Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 776–781.
16. Jagadeesh, G.R.; Srikanthan, T. Online map-matching of noisy and sparse location data with hidden Markov and route choice models. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2423–2434. [CrossRef]
17. Bernstein, D.; Kornhauser, A. *An Introduction to Map Matching for Personal Navigation Assistants*; Technical Report; New Jersey TIDE Center: Newark, NJ, USA, 1996.
18. White, C.E.; Bernstein, D.; Kornhauser, A.L. Some map matching algorithms for personal navigation assistants. *Transp. Res. Part C Emerg. Technol.* **2000**, *8*, 91–108. [CrossRef]
19. Phuyal, B.P. Method and use of aggregated dead reckoning sensor and GPS data for map matching. In Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002), Portland, Australia, 24–27 September 2002; pp. 430–437.
20. Eiter, T.; Mannila, H. *Computing Discrete Fréchet Distance*; Technical Report; Christian Doppler Laboratory for Expert Systems: Vienna, Austria, 1994.
21. Chen, D.; Driemel, A.; Guibas, L.J.; Nguyen, A.; Wenk, C. Approximate map matching with respect to the Fréchet distance. In Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments, San Francisco, CA, USA, 22 January 2011; Müller-Hannemann, M., Werneck, R.F., Eds.; SIAM: Philadelphia, PA, USA, 2011; pp. 75–83.
22. Alt, H.; Godau, M. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* **1995**, *5*, 75–91. [CrossRef]
23. Alt, H.; Efrat, A.; Rote, G.; Wenk, C. Matching planar maps. *J. Algorithms* **2003**, *49*, 262–283. [CrossRef]
24. Luo, A.; Chen, S.; Xv, B. Enhanced map-matching algorithm with a hidden Markov model for mobile phone positioning. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 327. [CrossRef]
25. Mohamed, R.; Aly, H.; Youssef, M. Accurate and efficient map matching for challenging environments. In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Dallas, TX, USA, 4–7 November 2014; pp. 401–404.
26. He, Z.C.; Xi-Wei, S.; Zhuang, L.J.; Nie, P.L. On-line map-matching framework for floating car data with low sampling rate in urban road networks. *IET Intell. Transp. Syst.* **2013**, *7*, 404–414. [CrossRef]
27. Ranacher, P.; Brunauer, R.; Trutschnig, W.; Van der Spek, S.; Reich, S. Why GPS makes distances bigger than they are. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 316–333. [CrossRef] [PubMed]
28. Navarro, G. A Guided Tour to Approximate String Matching. *ACM Comput. Surv.* **2001**, *33*, 31–88. [CrossRef]
29. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina: Chapel Hill, NC, USA, 1995.
30. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 4 October 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 153–158.
31. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
32. Zheng, Y.; Fu, H.; Xie, X.; Ma, W.Y.; Li, Q. *Geolife GPS Trajectory Dataset—User Guide*; Technical Report; Microsoft Research Asia: Beijing, China, 2011.