A General Feature-based Map Matching Framework with Trajectory Simplification

Yifang Yin
Interactive and Digital Media Institute, National University of Singapore, Singapore 119613
idmyiny@nus.edu.sg

Rajiv Ratn Shah
School of Computing, National University of Singapore, Singapore 117417
rajiv@comp.nus.edu.sg

Roger Zimmermann
School of Computing, National University of Singapore, Singapore 117417
rogerz@comp.nus.edu.sg

ABSTRACT

Accurate map matching has been a fundamental but challenging problem that has drawn great research attention in recent years. It aims to reduce the uncertainty in a trajectory by matching the GPS points to the road network on a digital map. Most existing work has focused on estimating the likelihood of a candidate path based on the GPS observations, while neglecting to model the probability of a route choice from the perspective of drivers. Here we propose a novel feature-based map matching algorithm that estimates the cost of a candidate path based on both GPS observations and human factors. To take human factors into consideration is very important especially when dealing with low sampling rate data where most of the movement details are lost. Additionally, we simultaneously analyze a subsequence of coherent GPS points by utilizing a new segment-based probabilistic map matching strategy, which is less susceptible to the noisiness of the positioning data. We have evaluated the proposed approach on a public large-scale GPS dataset, which consists of 100 trajectories distributed all over the world. The experimental results show that our method is robust to sparse data with large sampling intervals (*e.g.*, 60 s $\sim$ 300 s) and challenging track features (*e.g.*, u-turns and loops). Compared with two state-of-the-art map matching algorithms, our method substantially reduces the route mismatch error by 6.4% $\sim$ 32.3% and obtains the best map matching results in all the different combinations of sampling rates and challenging features.

CCS Concepts

•**Information systems** → **Global positioning systems;** *Location based services;* •**Mathematics of computing** → *Probabilistic algorithms;*

Keywords

GIS, worldwide GPS data; feature-based map matching; trajectory simplification

DOI: 10.1145/1235

1. INTRODUCTION

Given a vehicle track consisting of a sequence of GPS points, map matching algorithms aim to automatically determine the correct route where the driver has traveled on a digital map. The correction of the raw positioning data has been important for many downstream applications such as navigation and tracking systems [7, 17, 24, 25]. Recently, an increasing number of statistics-based map matching algorithms have been proposed to deal with the challenging GPS trajectories that pose difficulties in traditional geometry-based or topology-based methods, *e.g.*, data noise and sparsity. Among the advanced statistics-based algorithms, the Hidden Markov Model (HMM) is one of the most popular and widely used technique that models the road emission and transition probabilities based on the measurement noise and the road network layout [14]. It has been reported that the HMM-based map matching is highly effective when dealing with trajectories where the GPS sampling interval is less than 30 seconds. However, real positioning data are sometimes collected with a very low sampling rate, *e.g.*, more than five minutes [26], and thus pose great difficulties in the development of map matching algorithms.

To reduce the uncertainty in low sampling rate trajectories, hybrid methods have been proposed to estimate the transition probability between two road segments based on a fusion of multiple metrics [1, 2, 5]. For example, Aly and Youssef proposed to detect road semantics with multiple sensors and estimate the transition probability based on both the orientation difference and the skipped road semantics [1]. However, such algorithms mostly assume that the driver has traveled on the shortest path between two road segments which is not always true especially when dealing with low sampling rate data. To solve the above problem, Zheng *et al.* proposed to infer the possible routes based on the travel patterns derived from historical data [28]. Osogami and Raymond considered both the number of turns and the travel distance in the cost modeling of a candidate path [15]. However, the performance of such algorithms can be limited due to the requirement of sufficient historical GPS trajectories in the learning phase. Moreover, the original HMM-based map matching algorithm and its extensions mostly retrieve candidate road segments for every GPS point [1, 14, 15], leading to decreasing effectiveness for trajectories with the existence of large sensor noise.

We therefore present a novel feature-based framework for accurate map matching of challenging GPS trajectories. We first detect key GPS points to segment a trajectory into a list of subsequences. To reduce the method's sensitivity to

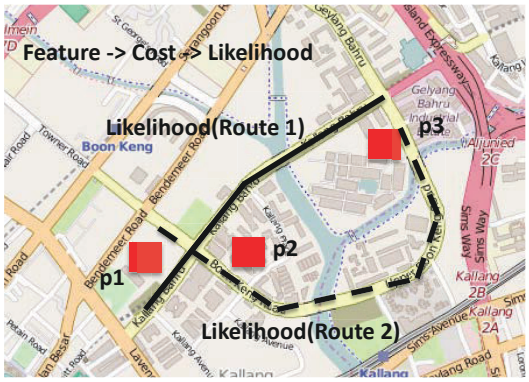


Figure 1: Illustration of the proposed feature-based route likelihood estimation.

data noise, we simultaneously consider all the GPS points in one segment to determine the most likely route taken by a user. Figure 1 illustrates a trajectory segment consisting of three GPS points. Instead of assuming users would always take the path with the shortest travel distance between two candidate roads, we model the cost of a route choice based on two types of features: (1) *trajectory-related features* to estimate the cost of a route from the perspective of GPS observations, and (2) *road-related features* to model the behavior cost of a route choice from the perspective of users. We next compute the likelihood of a candidate path based on its cost and determine the correct match as the path with the highest likelihood between two road segments. From a global perspective, we retrieve candidate road segments for each key GPS point, search for a local optimal path between any two neighboring candidate road segments, and compute the likelihood of a global candidate path by multiplying the likelihood values of the local optimal paths it contains. This problem can be solved efficiently by dynamic programming techniques. We have evaluated our proposed approach by comparing it with two state-of-the-art map matching algorithms. Experiments are conducted on real-world GPS data sampled with different intervals ranging from one minute to five minutes. We report the route mismatch fraction as the evaluation measure and the experimental results show that our method outperforms its competitors by a mismatch error reduction rate of 6.4% $\sim$ 32.3% on average. Here we summarize the contributions of this paper in the following four aspects:

- We present a novel feature-based map matching technique that models the cost of a candidate path with both trajectory-related features (*e.g.*, the distance to the closest GPS point) and road characteristics (*e.g.*, length and transitions).
- We consider more than one GPS point at a time by utilizing a new segment-based probabilistic map matching strategy that searches for shortest path between candidate roads of only the key points detected by trajectory simplification techniques.
- We perform extensive experiments on a large-scale real dataset consisting of trajectories with features (*e.g.*, u-turns and loops) that pose difficulties to map matching algorithms.
- We evaluate the proposed technique with varying sampling rates (1 min $\sim$ 5 min). The experimental results show that our method works consistently well and outperforms the state-of-the-art map matching algorithms.

The rest of the paper is organized is follows. We first report the important related work in Section 2 and present the system overview in Section 2. Next we introduce the technical details of the proposed feature-based map matching framework in Section 4. Finally, we evaluate the effectiveness of our proposed approach by comparing with the state-of-the-art map matching techniques in Section 5. Section 6 concludes and suggests future work.

2. RELATED WORK

Over the past decades, extensive research has been conducted on matching GPS points on a digital map. With simple road network information such as the locations and the shape of the roads, early map matching techniques can be generally classified into two categories: geometry-based matching and topology-based analysis. Geometry-based algorithms match a single GPS point [23] or a segment of GPS trajectories [3, 29] to the closest road arc based on geometric calculations. However, without considering the constraints induced by a map topology, these methods suffer from one significant drawback of being sensitive to measurement errors. On the other hand, topology-based map matching algorithms utilize not only the shapes, but also topological information such as connectivity and contiguity of the road network [6, 2]. Quddus *et al.* reported improved results by leveraging vehicle information of heading and speed in the topological analysis [18]. However, such algorithms are still vulnerable to sensor noise and unsuitable for highly erroneous and sparse positioning data [1].

To pursue improved matching accuracy, probabilistic map matching algorithms have been proposed in order to take advantage of statistical models such as Kalman Filter [16], particle filters [7, 10], and HMM [14, 4, 5]. Wang *et al.* proposed a novel statistics-based online map matching algorithm called Eddy with a solid error- and delay-bound analysis [22]. To work with mobile devices, Liu *et al.* presented a novel technique termed Passby which maintains high matching accuracies while working with the most simplified road network [11]. Situations that pose difficulties in map matching, *e.g.*, dealing with low sampling rate GPS tracks [13, 27] and matching to incomplete map data [21], have also been studied recently but still remain challenging problems. Newson and Krumm proposed an elegant HMM-based map matching algorithm for relatively noisy and sparse GPS trajectories [14]. However, experiments show that the accuracy decreases significantly when the sampling period grows larger than 30 seconds. Zheng *et al.* proposed a history-based route inference system which derives the travel pattern from historical data to reduce the uncertainty of GPS trajectories [28]. However, the inference process requires a large quantity of historical trajectories with good coverage and high density, which greatly limits the applicability of such algorithms.

Recently, a number of map matching algorithms started to utilize other sensors equipped on smartphones such as WiFi and cellular fingerprinting [20, 19]. Aly and Youssef proposed to detect road semantics (*e.g.*, speed bumps and tunnels) by leveraging smartphone's inertial sensors and pre-

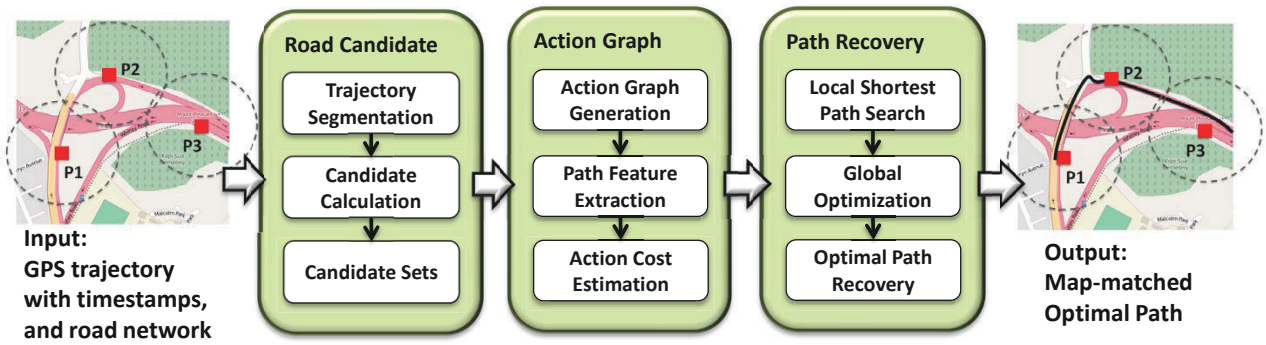


Figure 2: Overview of the proposed feature-based map matching system architecture

sented an improved HMM with a semantics-enriched digital map [1]. Furthermore, one interesting direction that emerged recently is to perform map matching with the assistance of driver behavior analysis. Drivers attempt to reach some destination while optimizing some trade-off between time, safety, and other factors which can be modeled by a list of path features such as road type and speed limit [30]. Osogami and Raymond proposed to integrate the number of turns in the transition probability calculation of a HMM in order to favor a more "natural" path in the decision making process [15]. Promising results have been reported on GPS points taken during a single trip in Seattle. However, the generality of such methods remains unclear without conducting extensive experimental studies on large-scale positioning data.

3. SYSTEM OVERVIEW

As an overview, we first provide formal definitions of the map matching problem using the proposed feature-based method. Next, we briefly introduce the functionality of each system component and leave the technical details to the next section.

3.1 Problem Statement

Given a GPS trajectory and a digital map, our goal is to find the most likely route that has been traveled by the user. Here we model a map as a simple directed graph where the nodes have assigned geo-coordinates on Earth and the edges represent linear road segments between two nodes:

Definition 1 *(Road Segment)*: A road segment e is a directed edge that is associated with an id $e.eid$, a length value $e.l$, a starting point $e.startp$, and an ending point $e.endp$. It represents a linear road segment between the two nodes $e.startp$ and $e.endp$.

Definition 2 *(Road Network)*: A road network is a directed graph $G(V, E)$, where E is a set of edges representing the road segments and V is the vertex set consisting of the starting and ending points of the road segments.

Definition 3 *(GPS Trajectory)*: A GPS trajectory is a sequence of GPS points $T = \{p_1, p_2, ..., p_n\}$. Each point p_i is associated with a geo-coordinate $< p_i.lat, p_i.lon >$ and a timestamp $p_i.t$.

To incorporate user behavior analysis, we also introduce the concept of *action* which describes the user behavior of traveling from one road segment to another. Note that the two road segments are required to be directly connected to form a legal action. The formal definition is given as below:

Definition 4 *(Action)*: An action a is a directed edge that is associated with a starting road segment $a.start$, an ending road segment $a.end$, the angle between the two road segments $a.angle$, and the cost of taking this action $a.cost$. It models a turning action from $a.start$ to $a.end$.

Definition 5 *(Action Graph)*: An action graph is a directed graph $G_A = (E, A)$, where the vertices E are the set of the road segments in $G(V, E)$, and the edges $A = \{a_1, a_2, ..., a_m\}$ are formed by the actions of traveling between two directly connected road segments.

Definition 6 *(Action Sequence)*: An action sequence AS is a path connecting two road segments in the action graph $G_A(E, A)$. Let $AS = \{\tilde{a}_1, \tilde{a}_2, ...\tilde{a}_{\tilde{m}}\}$ where $\tilde{a}_i \in A$, then for each i in range $[1, \tilde{m})$, we have $\tilde{a}_i.end = \tilde{a}_{i+1}.start$.

Definition 7 *(Path)*: A path P is a sequence of connected road segments. Given an action sequence AS, the corresponding path can be recovered by concatenating the road vertices traversed by AS.

Now we define the map matching problem as follows: Provided with a raw GPS trajectory T and a road network $G(V, E)$, generate the action graph $G_A(E, A)$ and estimate the cost and probability of taking each action in A. Find the most probable sequence of actions AS and recover the optimal path P accordingly.

3.2 Architecture Overview

The architecture of the proposed feature-based map matching system is illustrated in Figure 2. It consists of three major components: *Road Candidate Preparation*, *Action Graph Generation* and *Path Recovery*.

Road Candidate Preparation: Given a raw GPS trajectory and the corresponding road network information, this component retrieves possible candidate roads for a number of key GPS points detected from the trajectory. The key points are obtained by trajectory simplification while preserving the shape of the curve within a given tolerance [8, 12]. While previous statistics-based methods mostly retrieve candidate roads for every GPS point, we alternatively adopt a more effective segment-based map matching strategy that considers a subsequence of points at a time in the probabilistic modeling. Moreover, we incorporate user behavior analysis in finding the local optimal path between the candidate roads of two neighboring key GPS points, which significantly improves the map matching accuracy especially when dealing with low sampling rate positioning data.

Action Graph Generation: This component generates the action graph $G_A = (E, A)$ where the action sequence

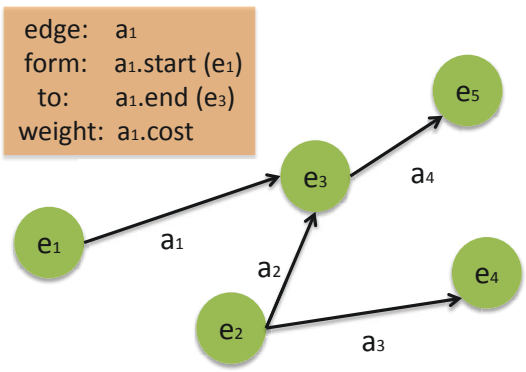


Figure 3: The generated action graph where nodes are road segments and edges are actions.

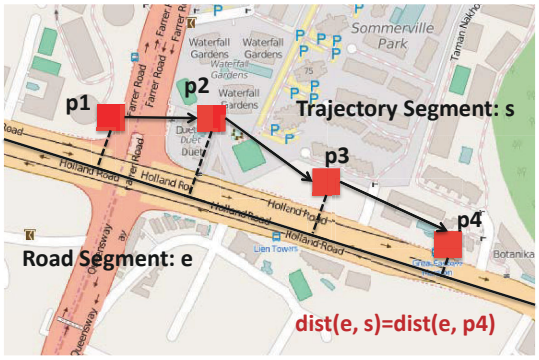


Figure 4: Illustration of the *trajectory-related feature* extraction.

that matches the real route can be detected by a simple shortest path search given any two neighboring candidate roads. As illustrated in Figure 3, the nodes in the action graph are formed by the road segments and the edges are formed by the actions A where each element a represents a directed edge from node $a.start$ to node $a.end$ with the edge weight to be $a.cost$. Please note that the cost here models not only the likelihood of taking one action given the observations of raw GPS data, but also the trade-off made by users (drivers) among a list of factors such as time, safety, and stress. Subsequently, we extract two types of path features based on GPS points and road characteristics, respectively, to estimate the cost values. Two basic road features (length and transitions) that are available from all digital maps are leveraged in this work. If provided with more semantic information such as road type and speed limit, user behavior cost can be better modeled by advanced feature fusion techniques [30]. Additionally, in areas where dense historical GPS trajectories are available, it is also possible to automatically infer road semantics, *e.g.*, road popularity, based on data-driven approaches [28].

Path Recovery: Based on the action graph generated, this component computes the shortest path and the corresponding cost between any two neighboring candidate roads in the retrieved candidate set. The local shortest paths are next concatenated at the starting and ending road segments to form a set of global candidate paths for the whole input trajectory. Thereafter, the probability of a global candidate path being the correct match is modeled based on the cost of the local shortest paths it contains. Finally, the candidate path with the highest probability score is returned as the predicted map matching result, which can be efficiently solved using dynamic programming. The results are next compared to the real route and the paths predicted by two state-of-the-art map matching algorithms. We visualize the results on a map interface and also report the route mismatch fraction which measures the matching accuracy.

4. FEATURE-BASED MAP MATCHING

In traditional map matching algorithms, the candidate path scoring mostly relies on the distance modeling between the input GPS observations and the road network database based on spatial and temporal constraints [13, 14]. In this work, we refer to the aforementioned aspect as trajectory-related path features and further propose a general framework that enables effective feature fusion with road characteristics in the decision making process. Our proposed map matching algorithm can effectively reduce the uncertainty of low sampling rate GPS data with possibly challenging patterns such as u-turns and loops. The technical details of the major system components are introduced as below.

4.1 Action Graph Generation

The *Action Graph Generation* is the core component of the proposed system. Recall that the action graph is created based on the road network. The nodes in the graph are road segments and an edge a describes the action of traveling from one road segment $a.start$ to a neighboring road segment $a.end$. The edge weight is set to $a.cost$. Next, we introduce how to extract path features and estimate action costs based on an input trajectory segment s.

4.1.1 Path Feature Extraction

As aforementioned, we extract two types of path features in order to estimate the traveling cost:

- *Trajectory-related features*: the distance between a road and a trajectory segment, which will next be used to estimate the cost of a road segment given the GPS observations.
- *Road-related features*: the road characteristics such as length and transitions, which will next be used to model the behavior cost of a route choice made by users.

Figure 4 illustrates the *trajectory-related feature* extraction with the input being a trajectory segment consisting of four GPS points. Intuitively, the road segments that are farther from the trajectory are less likely to be the correct match. Therefore, we formulate the distance between a road segment e and a trajectory segment s as

$$dist(e, s) = \min_{p \in s} dist(e, p) \quad (1)$$

where $p \in s$ denotes a GPS point p in trajectory segment s, and $dist(e, p)$ represents the distance between point p and road segment e, which is defined to be the great circle distance between point p and the point on road segment e which is the closest to p. For example in Figure 4, $dist(e, s) = dist(e, p_4)$ as point p_4 is the closest GPS measurement to road segment e.

On the other hand, Figure 5 illustrates the extraction of the two *road-related features* that are leveraged in this work.

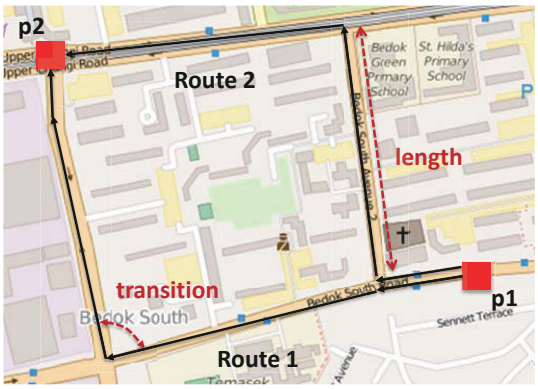


Figure 5: Illustration of the *road-related feature* extraction.

Due to concerns about the time, people usually favor shorter paths over longer ones. Moreover, the type and the number of road transitions also play an important role in users' route choices [15]. For example in Figure 5, even though *Route 2* is the shorter path between the two GPS points, it is more likely that people would choose *Route 1* due to safety concerns as this path contains only one 90-degree-turn while the former path contains two. Based on the above observations, we compute the length of a road segment and the transition angle between any two connected road segments. Each length feature is associated with a road segment e and denoted as $e.l$. Similarly, each transition angle is associated with an action a and denoted as $a.angle$. Next we introduce how to compute the action cost based on the extracted features.

4.1.2 Action Cost Estimation

Considering that the probability of users to turn on a road segment that is farther away from the GPS measurements is small, we model the cost of an action a based on the input trajectory segment s as

$$C_{traj} = \min\{dist(e, s), maxC_{traj}\} \tag{2}$$

where $e = a.end$ is the ending road segment of action a and $maxC_{traj}$ is a threshold that limits the maximum value of C_{traj}. We set the cost of the road segments that are far away from the input trajectory segment to a constant value $maxC_{traj}$. This is because that the trajectory segment s provides little information when the distance $dist(e, s)$ grows much larger than the GPS accuracy, *i.e.*, GPS measurements are only effective for the cost estimation of the nearby road segments. Therefore, we set $maxC_{traj} = 100$ meters in our experiments, which is reasonable according to the GPS measurement accuracy.

Based on *road-related features*, we estimate C_{len} of action a as the length of the ending road segment $a.end$ in meters

$$C_{len} = e.l \tag{3}$$

where $e = a.end$ and $e.l$ is the length attribute of road segment e. Thereafter, we model the cost of a transition C_{turn} as proposed by Osogami and Raymond [15]

$$C_{turn} = \begin{cases} 0 & |a.angle| < \pi/4 \\ 1 & \pi/4 \le |a.angle| < 3\pi/4 \\ 2 & 3\pi/4 \le |a.angle| \le \pi \end{cases} \tag{4}$$

where $a.angle$ is the transition angle between the two road segments $a.start$ and $a.end$. It is worth mentioning that currently we do not distinguish between left and right turns, therefore we have $0 \le a.angle \le \pi$. Later in the experiments we will see, the setting of the relative weights of angles shown in Eq. 4 works generally well on trajectories from different regions all over the world. Moreover, it is also possible to fine-tune the weights using machine learning techniques with the presence of large GPS data.

Finally, we fuse C_{traj}, C_{len} and C_{turn} into the overall cost of an action as

$$C = C_{traj} \cdot (C_{len} + \omega C_{turn}) \tag{5}$$

where ω is a balancing factor between road length and transition angle. The intuition of Eq. 5 is that, small C_{traj} values promote user actions of turning onto road segments that are closer to the GPS measurements, while small C_{len} and C_{turn} promote shorter and straighter routes which are more likely to be chosen by users. Thereafter, we compute cost C for every action a and set $a.cost = C$ accordingly. Next we present a segment-based map matching strategy that models the probability of a route choice based on the action cost.

4.2 Segment-based Probabilistic Modeling

Based on the action cost estimated by Eq. 5, we can retrieve the most likely action sequence that connects any two candidate road segments in the action graph by shortest path search. As an example, Figure 6 illustrates a trajectory consisting of six GPS points. The shortest-path-based strategy works well for relatively straight trajectory segments such as $\{p_1, p_2, p_3\}$ or $\{p_3, p_4, p_5, p_6\}$. However, it is highly difficult to recover the real path between p_1 and p_6 directly as *Route 2* is more likely to be mistakenly retrieved due to the small values of C_{len} and C_{turn}.

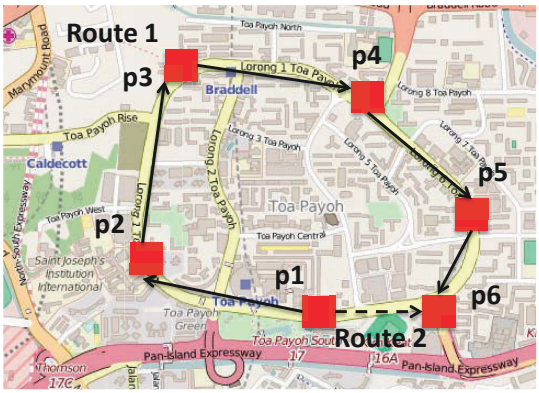


Figure 6: Problems caused by loops in GPS trajectories.

To solve the above problem, we first segment the input trajectory and obtain a list of key GPS points (*e.g.*, p_1, p_3, and p_6). Next, we retrieve possible candidate road segments for each of the key GPS points, compute the shortest path between any two neighboring candidates, and recover the real path by global optimization. The technical details are introduced as below.

4.2.1 Trajectory Segmentation

In our implementation, we obtain the list of key GPS points by applying the Douglas-Peucker algorithm [8]. Given

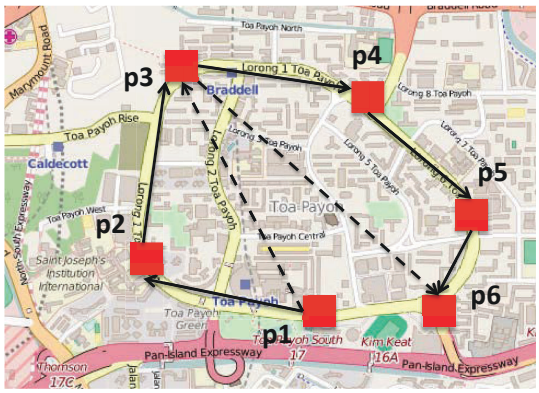


Figure 7: Simplifying a trajectory with the Douglas-Peucker algorithm.

a trajectory composed of line segments, this algorithm simplifies the trajectory by finding a similar curve with fewer points. For example in Figure 7, a trajectory consisting of six GPS points is approximated by curve $\{p_1, p_3, p_6\}$ after simplification. Subsequently, the trajectory is divided into two segments $\{p_1, p_2, p_3\}$ and $\{p_3, p_4, p_5, p_6\}$, with the key GPS points being p_1, p_3, and p_6.

The Douglas-Peucker algorithm initially keeps the first and the last points in the trajectory (*i.e.*, p_1 and p_6), and then finds the point that is the farthest from the line segment between the first and the last points (*i.e.*, p_3). This point will only be kept when its distance to the line segment is greater than a pre-defined threshold ϵ. If this point is kept, the algorithm will recursively process the segment from the first point to this point and the segment from the this point to the last point. Otherwise, any points other than the first and the last points can be discarded with the simplified curve being no worse than ϵ.

4.2.2 Path Recovery

For an input trajectory $T = \{p_1, p_2, ..., p_n\}$, we first detect the key GPS points, denoted as $\tilde{T} = \{\tilde{p}_1, \tilde{p}_2, ..., \tilde{p}_{\tilde{n}}\}$, based on the Douglas-Peucker algorithm. Next, we retrieve the nearest ten road segments and form the candidate set $E^i = \{e_1^i, e_2^i, ..., e_{10}^i\}$ for each key point $\tilde{p}_i$. We model the GPS noise with a Gaussian kernel and estimate the probability of candidate e_k^i being the correct match of key point $\tilde{p}_i$ as [14]

$$p(e_k^i) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{dist(e_k^i, \tilde{p}_i)^2}{2\sigma^2}} \tag{6}$$

where σ is the standard deviation of GPS measurements, $dist(e_k^i, \tilde{p}_i)$ represents the minimum great circle distance between candidate e_k^i and point $\tilde{p}_i$.

The probability of a route being the correct match of the real path between two neighboring key points $\tilde{p}_i$ and $\tilde{p}_{i+1}$ is calculated as follows. For any two candidate road segments e_f^i and e_t^{i+1} $(1 \le f, t \le 10,\ 1 \le i < \tilde{n})$, we obtain the optimal action sequence that is connecting e_f^i and e_t^{i+1}, denoted as $AS^*(e_f^i, e_t^{i+1})$, based on the shortest path search in the action graph. Subsequently, we compute the sum of the action cost in $AS^*(e_f^i, e_t^{i+1})$ as

$$C^*(e_f^i, e_t^{i+1}) = \sum_{a \in AS^*(e_f^i, e_t^{i+1})} a.cost \tag{7}$$

ALGORITHM 1: Feature-based Map Matching.

Input: trajectory T and road network $G(V, E)$
Output: the matched path P

Find a set of key GPS points $\tilde{T} = \{\tilde{p}_1, \tilde{p}_2, ..., \tilde{p}_{\tilde{n}}\}$ in T by Douglas-Peucker algorithm
for *each segment s between $\tilde{p}_i$ and $\tilde{p}_{i+1}$* **do**
  Extract path features based on s and $G(V, E)$
  Update action cost by Eq. 5
  Retrieve candidate road segments E^i and E^{i+1} for key points $\tilde{p}_i$ and $\tilde{p}_{i+1}$
  for $e_f^i \in E^i$, $e_t^{i+1} \in E^{i+1}$ **do**
    compute $p(e_f^i)$, $p(e_t^{i+1})$ by Eq. 6
    compute $p(e_f^i, e_t^{i+1})$ by Eq. 8
Initialize $f[e_k^1] = p(e_k^1)$, $k = 1, 2, ..., 10$
for $i = 2$ *to* $\tilde{n}$ **do**
  for $e_t^i \in E^i$ **do**
    for $e_f^{i-1} \in E^{i-1}$ **do**
      $l = f[e_f^{i-1}] \cdot p(e_f^{i-1}, e_t^i) \cdot p(e_t^i)$
      if $l > f[e_t^i]$ **then**
        $f[e_t^i] = l$
        $pre[e_t^i] = e_f^{i-1}$
$P = \arg\max_{e_{k_1}^1 \to e_{k_2}^2 \to \cdots \to e_{k_{\tilde{n}}}^{\tilde{n}}} f[e_{k_{\tilde{n}}}^{\tilde{n}}]$
return P

Osogami and Raymond [15] assumed that a route with total cost C matches the real path with a probability proportional to $\exp(-C)$. However, as the length of the trajectory segment between $\tilde{p}_i$ and $\tilde{p}_{i+1}$ has a great impact on the route cost $C^*(e_f^i, e_t^{i+1})$, we further normalize the cost by the great circle distance between $\tilde{p}_i$ and $\tilde{p}_{i+1}$, and the fraction of the number of GPS points in this segment over the total number of GPS points in trajectory T

$$p(e_f^i, e_t^{i+1}) = \exp\left(-\frac{N(\tilde{p}_i, \tilde{p}_{i+1})}{n \cdot dist(\tilde{p}_i, \tilde{p}_{i+1})} \cdot C^*(e_f^i, e_t^{i+1})\right) \tag{8}$$

where $N(\tilde{p}_i, \tilde{p}_{i+1})$ represents the number of GPS points in the segment between key points $\tilde{p}_i$ and $\tilde{p}_{i+1}$, n is the total number of GPS points in the input trajectory T, and $dist(\tilde{p}_i, \tilde{p}_{i+1})$ represents the great circle distance between points $\tilde{p}_i$ and $\tilde{p}_{i+1}$.

A global candidate path for the entire trajectory T goes through the candidate road segments of every key point $\tilde{p}_i$ in temporal order: $e_{k_1}^1 \to e_{k_2}^2 \to \cdots \to e_{k_{\tilde{n}}}^{\tilde{n}}$. We estimate the likelihood of a global candidate path by combining Eq. 6 and Eq. 8

$$\begin{aligned} &l(e_{k_1}^1 \to e_{k_2}^2 \to \cdots \to e_{k_{\tilde{n}}}^{\tilde{n}}) \\ &\quad = p(e_{k_1}^1) \cdot p(e_{k_1}^1, e_{k_2}^2) \cdot p(e_{k_2}^2) \cdots p(e_{k_{\tilde{n}}}^{\tilde{n}}) \end{aligned} \tag{9}$$

The candidate path with the highest likelihood is returned as the final map matching results, which can be efficiently solved by a dynamic programming technique.

$$P = \arg\max\ l(e_{k_1}^1 \to e_{k_2}^2 \to \cdots \to e_{k_{\tilde{n}}}^{\tilde{n}}) \tag{10}$$

Algorithm 1 outlines our feature-based map matching technique. As aforementioned, it detects a set of key GPS points and processes each segment in-between to calculate the likelihood values $p(e_f^i)$, $p(e_t^{i+1})$, and $p(e_f^i, e_t^{i+1})$. In terms of the global optimization, $f[e_k^i]$ records the highest likelihood

of a candidate path ending at road segment e_k^i. $pre[e_k^i]$ caches a candidate road segment of the previous key point $\tilde{p}_{i-1}$ from which the highest likelihood $f[e_k^i]$ is obtained. It is easy to see that the match of the ending point $\tilde{p}_{\tilde{n}}$ is $\arg\max_{e_{k_{\tilde{n}}}^{\tilde{n}}} f[e_{k_{\tilde{n}}}^{\tilde{n}}]$, and $pre[e_{k_{\tilde{n}}}^{\tilde{n}}]$ records the match of the previous point $\tilde{p}_{\tilde{n}-1}$. Therefore, the rest of the path can be reversely recovered from $pre[\,]$, and so on so forth.

5. EVALUATION

We implemented the proposed map matching algorithm and evaluated its effectiveness. The evaluation consists of two steps. The first part introduces the experimental setup and the details of the testing dataset. The second part verifies the effectiveness of our proposed approach and compares it to the state-of-the-art map matching methodologies.

5.1 Dataset and Experimental Setup

The dataset we used for evaluation is a large-scale real dataset which consists of 100 GPS tracks all over the world [9]. Each track is associated with a map and a correctly map-matched route. Moreover, the tracks are labeled with a selection of features that may pose difficulties to map matching algorithms including:

- *u-turns*: the vehicle turned 180° and reversed the direction of travel
- *hives*: large numbers of points packed in a small area
- *loops*: the vehicle was traveling in circles
- *gaps*: temporal gaps existing in the track
- *severe congruence issues*: situations where the map and the track are incongruent or dissimilar

The number of tracks that are labeled with each of the five features is reported in Table 1. Additionally, there are 19 tracks formed by high quality GPS data with no tags associated. The length of the tracks varies from 5 to 100 kilometers, and the dataset contains 247,251 points in total with a sampling rate of 1 Hz. For more details, please refer to the dataset paper [9].

Table 1: Number of tracks in the worldwide GPS dataset tagged with different features.

u-turns	hives	loops
25	3	24
gaps	congruence-issues	no tags
73	20	19

The sampling interval of real GPS data varies significantly from less than one minute to more than five minutes [26]. Therefore, it is important to evaluate the robustness of the proposed map matching algorithm when applied to low sampling rate GPS data. To achieve this goal, we generate five datasets by subsampling the original data with different sampling intervals of 60 s, 120 s, 180 s, 240 s, and 300 s, respectively. The average number of GPS points per track versus the sampling rate is illustrated in Figure 8. Moreover, we also report the average number of the key points per track detected based on the trajectory simplification technique as introduced in Section 4.2.1.

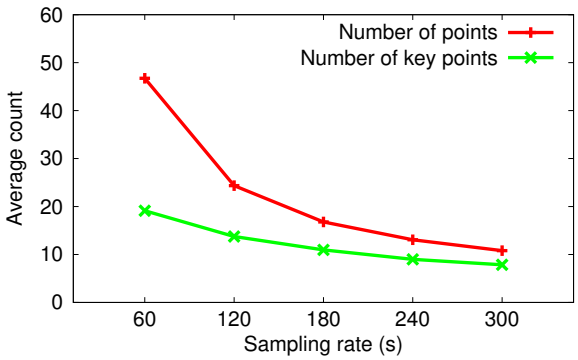


Figure 8: Average number of GPS points and segments before and after trajectory simplification.

From the results we can see, the shape of a trajectory can be described with much fewer points especially when the sampling rate is relatively high. For example, with a sampling interval of 60 s, the number of key GPS points detected, which is 19.12, is less than half of the total number of GPS points, which is 46.74, per track on average. The shape of a trajectory can be better preserved with more key points by setting the distance threshold ϵ required by the Douglas-Peucker algorithm to a smaller value. However, this is not necessary because our method works well as long as the segments between key points do not contain any loops. Thereby, we intuitively set the distance threshold $\epsilon = 0.001$ throughout the experiments, which obtains excellent map matching accuracies when comparing to other existing methods.

5.2 Map Matching Results

We evaluated the effectiveness of the proposed map matching algorithm. The matching accuracy is measured by the Route Mismatch Fraction (RMF) proposed by Newson and Krumm [14]. This measure computes the total length of the false positive road segments, denoted as d_+, and the total length of the false negative road segments, denoted as d_-. Let d_0 represent the total length of the real path. RMF quantifies the map matching error by the fraction of $(d_+ + d_-)/d_0$, so that a small RMF value indicates the map matching results are more similar to the real path. Next, we examine the map matching accuracy based on different parameter settings and compare our proposed method with the state-of-the-art techniques.

5.2.1 *Parameter Estimation*

The balancing factor ω in Eq. 5 is a key parameter in our model. It controls the weights of road length and transition angle in the action cost estimation. Here we examine the map matching results obtained by setting ω to different values. Figure 9 shows the average route mismatch fraction plotted against ω on trajectories sampled at different rates.

As can be seen, the best map matching results are obtained with ω setting to 100 or 150 at all the five GPS sampling rates. The accuracy of the predicted path slightly decreases with smaller or larger ω values, but the variations are not significant within a range of ω settings (*e.g.*, $50 < \omega < 300$). Additionally, the variation trend of the route mismatch fraction against ω is similar in the five groups with different sampling rates, which indicates that our method is

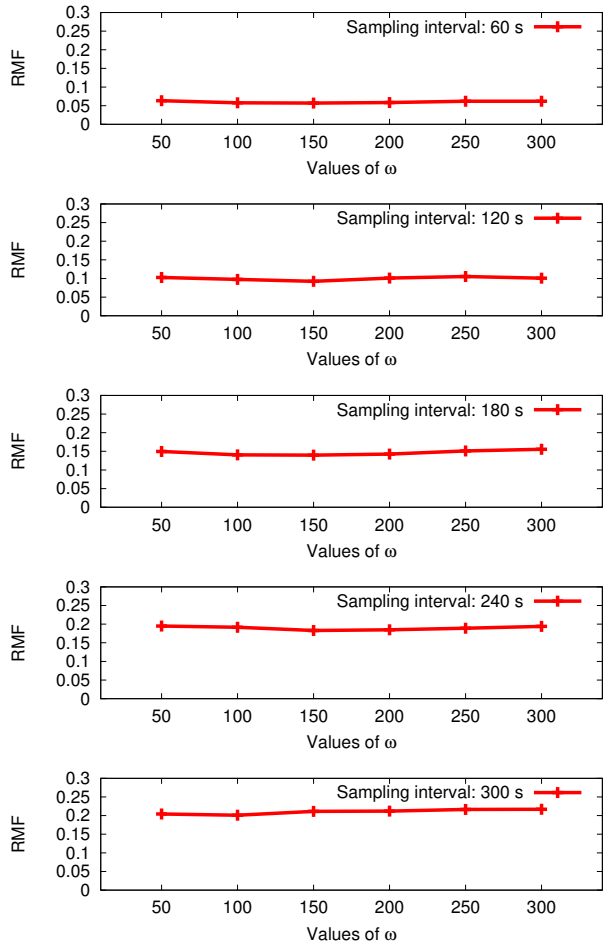


Figure 9: Route mismatch fraction plot based on different combinations of parameter ω and sampling intervals.

robust to trajectories with various sampling intervals or even temporal gaps.

Generally speaking, improved map matching results can be obtained with a range of ω values. But if large historical GPS data are available, such parameters can be better tuned by machine learning techniques such as the maximum entropy inverse reinforcement learning [30]. Regional factors can also be considered in the user behavior modeling based on training data with good geospatial coverage all over the world.

5.2.2 *Comparison with the State-of-the-art*

We evaluate the effectiveness of the proposed method using low sampling rate GPS data with various challenging features. We compare our method with the following two state-of-the-art algorithms and report route mismatch fraction based on different combinations of track features and sampling rates:

- **HMM-based Map Matching:** It leverages a Hidden Markov Model (HMM) to find the most likely sequence of road segments by conjunctively considering the measurement noise and the road network topology [14].
- **IRL-based Map Matching:** It extends the HMM-based map matching approach and estimates the transition probability between two road segments by a fusion of transition angle and travel distance, which is trained by Inverse Reinforcement Learning (IRL) [15].

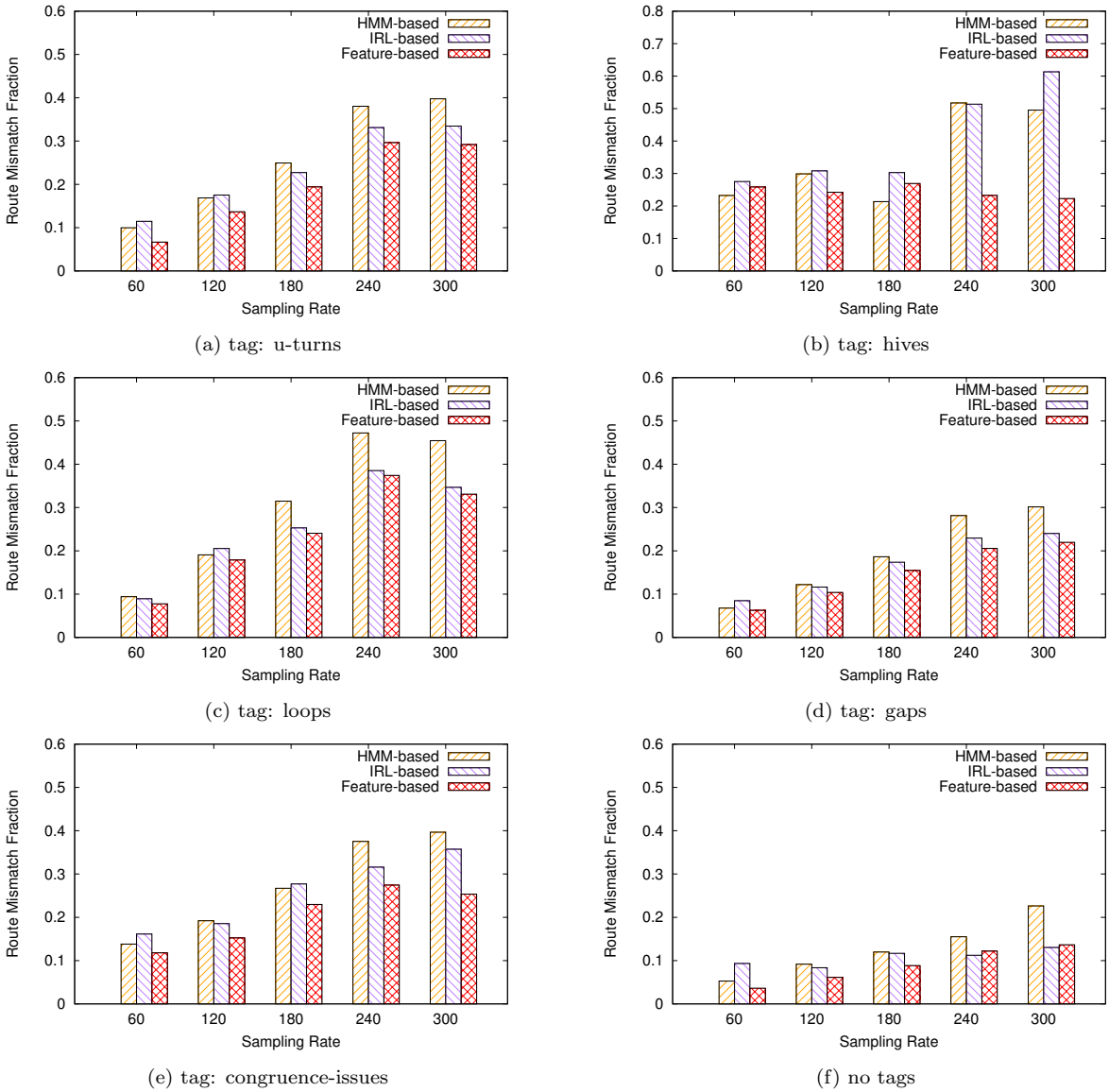
Throughout the experiments, the standard deviation of GPS measurements σ in Eq. 6 is set to 10 in all methods. The scaling factor β, which is used to estimate the transition probabilities in HMM-based and IRL-based methods, is also set to 10, but qualitative findings hold for a range of parameter settings [15]. The balancing factor ω in Eq. 5 is set to 100 in both the IRL-based method and our proposed feature-based method. For efficiency concerns, we only retrieve road segments that are within 200 meters of each GPS point to construct the candidate set in HMM-based and IRL-based methods. We conduct experiments on the worldwide testing dataset and report the average RMF over the 100 GPS trajectories in Table 2.

We compare the RMF based on five different sampling rates, and our proposed method obtain the best map matching accuracy in all cases. As illustrated in Table 2, our algorithm outperformed the HMM-based method by 12.0%, 18.0%, 20.7%, 26.6%, and 29.7% with sampling intervals 60 s, 120 s, 180 s, 240 s, and 300 s, respectively. Traditional HMM-based methods simply assume that drivers would always take the shortest path in terms of travel distance between two candidate road segments, which is not necessarily true especially for low sampling rate GPS data. With a growing sampling interval, the uncertainty in GPS trajectories is also increasing because most of the movement details are lost. Although the HMM-based method is able to handle data sparsity to some extent (*e.g.*, 30 seconds [14]), its effectiveness decreases dramatically as the sampling interval grows much larger. To obtain improved results, Osogami and Raymond proposed an IRL-based method based on the assumption that drivers would take the more "natural" path instead of the shortest path [15]. From Table 2 we can see that the IRL-based technique outperformed the HMM-based method for sampling periods larger than two minutes, but it only achieved similar or even worse results for the cases reported in the first two columns. It indicates that the IRL-based method is more susceptible to data noise as it tries to find a matching road segment for every GPS point. Our method, on the other hand, segments a trajectory and aims to find a matching route for every segment. Therefore, the robustness of the proposed feature-based method has been greatly improved by simultaneously considering all the points in one segment while performing map matching. Compared with the IRL-based method, our approach achieved improvements of 32.3%, 15.7%, 13.3%, 8.1%, and 6.4% in groups with different sampling intervals, respectively.

Next, we evaluate the effectiveness of our proposed approach for trajectories with different challenging features. The average map matching results are reported in Figure 10. Generally speaking, the qualitative findings are mostly the same as the results reported in Table 2. By segmenting an input trajectory into relatively straight subsequences, our method is able to obtain the highest accuracy even when applying to trajectories with *u-turns* and *loops*. For trajectories labeled by *hives* where a large volume of GPS points are packed in a small area, the IRL-based method performed much worse than the rest of the cases due to its sensitivity to data noise. On the other hand, Figure 10(f) shows

Table 2: Comparison of the average route mismatch fraction over 100 GPS trajectories.

Sampling Rate (s)	60	120	180	240	300
HMM-based	0.0657	0.1189	0.1773	0.2610	0.2856
IRL-based	0.0854	0.1157	0.1621	0.2085	0.2147
Feature-based	0.0578	0.0975	0.1406	0.1917	0.2009



(a) tag: u-turns

(b) tag: hives

(c) tag: loops

(d) tag: gaps

(e) tag: congruence-issues

(f) no tags

Figure 10: Route mismatch error comparison with different sampling rate.

the comparison when dealing with good quality data without any challenging features associated. Although the IRL-based method performed comparatively well or even slightly better than the proposed approach in some cases, it failed to maintain the good performance in other groups with decreasing sampling intervals. While in most of the cases, it is easy for humans to manually judge which is the correct route of a track on a map, there are situations where the correct matching is not clear even to us. Kubička *et al.* labeled such situations as *severe congruence issues* when the map and the track are incongruent or dissimilar [9]. Please note that although the hand-correction made by humans cannot be considered as the ground truth data for trajectories annotated with *congruence-issues*, the results shown in Figure 10(e) indicate that the route predicted by our proposed algorithm is more similar to human intuition. This is also important because it is reasonable to assume that a human mind is usually superior in matching GPS tracks on a digital map.

6. CONCLUSION AND FUTURE WORK

We have presented a novel feature-based map matching framework that models the likelihood of a candidate path based on both GPS observations and human factors. To improve the system robustness to data noise, we simultaneously process multiple GPS points at a time by segmenting the input trajectory into coherent subsequences. Road characteristics such as length and transition angles are incorporated as those factors have a major affect on a driver's route choice. We conduct extensive experiments on a challenging real-world dataset and the experimental results show that our proposed method obtains the state-of-the-art map matching accuracies.

In the future, we plan to explore the fusion of additional features for user behavior cost estimation. With a digital map that contains more semantic information such as road type and speed limit, improved results can be obtained by integrating these road characteristics in the action cost estimation. Furthermore, we will optimize the current system in terms of efficiency and provide accurate real-time map matching services in the future. The Douglas-Peucker algorithm can be replaced by more advanced online trajectory segmentation algorithms. Different influence factors on the system computational cost will be discussed and compared with the state-of-the-art map matching algorithms as well, in addition to the route mismatch fraction.

7. ACKNOWLEDGMENTS

This research has been supported in part by Singapore's Ministry of Education (MOE) Academic Research Fund Tier 1, grant number T1 251RES1415.

8. REFERENCES

[1] H. Aly and M. Youssef. semmatch: Road semantics-based accurate map matching for challenging positioning data. In *ACM SIGSPATIAL*, pages 5:1–5:10, 2015.

[2] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *VLDB*, pages 853–864, 2005.

[3] S. S. Chawathe. Segment-based map matching. In *IEEE Intelligent Vehicles Symposium*, pages 1190–1197, 2007.

[4] S. Fang and R. Zimmermann. EnAcq: Energy-efficient gps trajectory data acquisition based on improved map matching. In *ACM SIGSPATIAL*, pages 221–230, 2011.

[5] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet. Online map-matching based on hidden markov model for real-time traffic sensing applications. In *IEEE Intelligent Transportation Systems*, pages 776–781, 2012.

[6] J. S. Greenfeld. Matching gps observations to locations on a digital map. In *Transportation Research Board Annual Meeting*, 2002.

[7] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, pages 425–437, 2002.

[8] J. Hershberger and J. Snoeyink. Speeding up the douglas-peucker line-simplification algorithm. Technical report, 1992.

[9] M. Kubička, A. Cela, P. Moulin, H. Mounier, and S. I. Niculescu. Dataset for testing and training of map-matching algorithms. In *IEEE Intelligent Vehicles Symposium*, pages 1088–1093, 2015.

[10] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, pages 311–331, 2007.

[11] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding. Effective map-matching on the most simplified road network. In *ACM SIGSPATIAL*, pages 609–612, 2012.

[12] C. Long, R. C.-W. Wong, and H. V. Jagadish. Direction-preserving trajectory simplification. *VLDB Endowment*, pages 949–960, 2013.

[13] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate gps trajectories. In *ACM SIGSPATIAL*, pages 352–361, 2009.

[14] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *ACM SIGSPATIAL*, pages 336–343, 2009.

[15] T. Osogami and R. Raymond. Map matching with inverse reinforcement learning. In *IJCAI*, pages 2547–2553, 2013.

[16] O. Pink and B. Hummel. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In *Intelligent Transportation Systems*, pages 862–867, 2008.

[17] M. A. Quddus, R. B. Noland, and W. Y. Ochieng. A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems*, pages 103–115, 2006.

[18] M. A. Quddus, W. Y. Ochieng, L. Zhao, and R. B. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, pages 157–167, 2003.

[19] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, low-energy trajectory mapping for mobile devices. In *Networked Systems Design and Implementation*, pages 267–280, 2011.

[20] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Embedded Networked Sensor Systems*, pages 85–98, 2009.

[21] F. Torre, D. Pitchford, P. Brown, and L. Terveen. Matching gps traces to (possibly) incomplete map data: Bridging map building and map matching. In *ACM SIGSPATIAL*, pages 546–549, 2012.

[22] G. Wang and R. Zimmermann. Eddy: An error-bounded delay-bounded real-time map matching algorithm using hmm and online viterbi decoder. In *ACM SIGSPATIAL*, pages 33–42, 2014.

[23] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, pages 91–108, 2000.

[24] Y. Yin, B. Seo, and R. Zimmermann. Content vs. context: Visual and geographic information use in video landmark retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(3):39:1–39:21, 2015.

[25] Y. Yin, L. Zhang, and R. Zimmermann. Exploiting spatial relationship between scenes for hierarchical video geotagging. In *ACM ICMR*, pages 363–370, 2015.

[26] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: Driving directions based on taxi trajectories. In *ACM SIGSPATIAL*, pages 99–108, 2010.

[27] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Z. Sun. An interactive-voting based map matching algorithm. In *Mobile Data Management*, pages 43–52, 2010.

[28] K. Zheng, Y. Zheng, X. Xie, and X. Zhou. Reducing uncertainty of low-sampling-rate trajectories. In *IEEE ICDE*, pages 1144–1155, 2012.

[29] Y. Zheng. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology*, 6(3):29:1–29:41, 2015.

[30] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438, 2008.