IPSJ Transactions on Mathematical Modeling and Its Applications Vol.12 No.1 43–51 (Mar. 2019)

Regular Paper

# Route graph: Joint Map-matching by Maximizing Posterior Probability

Hiroya Inakoshi[1,a) ] Junichi Shigezumi[1,b)] Tatsuya Asai[1,c)] Takuya Kida[2,d)] Hiroki Arimura[2,e)]

Received: August 24, 2018, Revised: October 17, 2018,
Accepted: November 8, 2018

**Abstract:** We propose a joint map-matching for estimating unobservable paths from GPS traces. Our method is the first to maximize the posterior probability of stochastic generative model, in which traces are emitted as vehicles drive the roads. We employed the EM algorithm to find the parameters of the generative model, as well as to evaluate the expectations of the latent variable, which is indeed the estimated unobservable path. The EM algorithm is reduced to the exploratory search of the *route graph*, which is the geometric graph that is most likely emitting the traces and corresponds to the parameters of the model. Due to this stochastic formulation, our method works well with the presence of sampling noises in the traces. We report that the residual degradation of the estimated paths was no more than 7.0% even when they are sampled at a rate as low as 40%.

**Keywords:** GPS, map-matching, trajectory analysis

## 1. Introduction

The ability to obtain spatio-temporal information is now commonplace as vehicles and smartphones are equipped with GPS devices. On the other hand, governments, dedicated private companies, and social communities have been providing and maintaining digital road maps (DRM). Among the various information services enabled by these technologies, the analyses of the flows of cars and people have enjoyed the most commercial success. For instance, analyzing traffic demands provides feedback for urban traffic design and the identification of typical routes improves the efficiency of distribution services [4], [10], [13], [20], [21], [25], [30].

GPS observations are collected from individual cars independently and asynchronously. Further more, they also contain observation noises, especially in urban areas with tall and large buildings. Before analyzing such irregular and unreliable GPS observations, a map-matching technique is commonly used to attach the observed trajectories on to a DRM. The authors of Ref. [5] surveyed the range of map-matching techniques, and those in Ref. [28] discussed recent developments and remaining problems.

Earlier proposals for on-line map-matching algorithms attach each observation to one of the neighboring road segments while considering the local connectivity of the segments [9], [14], [32]. Then, off-line map-matching algorithms were proposed, which consider the topological distances between trajectories and paths on a DRM [1], [2], [7], [12], [15], [23], [24], [29], [33]. Probabilistic map-matching algorithms have also been proposed for estimating the road links from which observations are made [3], [26], [27]. Due to the limited network bandwidth or the constraint on power consumption, map-matching low-sample trajectories has attracted recent interest. One advanced algorithm utilizes observations from other trajectories to map a trajectory onto a DRM [16], [17]. Another maps trajectories to the segments embedded in a DRM all at once by formalizing map-matching as an optimization problem [22].

Most of these preceding approaches, however, mainly focused on assigning trajectories to the routes that seem natural on a DRM. By contrast, less attention has been paid to estimating unobservable paths, which are unaccessible in practical situations. In addition, we must pay more attention to identifying major streams in the trajectories to provide useful insight for realizing applications such as demand analysis and urban design, as mentioned above. In this paper, we propose a joint map-matching method, which is formulated to maximize the posterior probability of a stochastic generative model. This model represents a process in which GPS devices on vehicles generate observations as they drive along the paths, which are actually unobservable. Using this stochastic model whose latent random variable represents an occurrence of a drive on a path, our method is able to directly estimate the unobservable paths from the observed trajectories. Our contributions are as follows: first, we present the process that generates GPS observations and formulate it as a stochastic generative model whose latent random variable represents the

1 Artificial Intelligence Laboratory, Fujitsu Laboratories Ltd., Kawasaki, Kanagawa 211–8588, Japan
2 Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido 060–0814, Japan
a) inakoshi.hiroya@jp.fujitsu.com
b) j.shigezumi@jp.fujitsu.com
c) asai.tatsuya@jp.fujitsu.com
d) kida@ist.hokudai.ac.jp
e) arim@ist.hokudai.ac.jp

occurrence of a drive on a path, and whose observed random variable represents the distance between the path and the trajectory. Second, we formulate an EM algorithm that maximizes the posterior probability of the generative model. Then, we show that the log-likelihood of the posterior probability should be reduced to an object function consisting of the residual of the trajectories from their maximizing paths and the description length of the DRM. Finally, we present our algorithm, which iteratively explores the subgraphs likely to emit the observations. The experimental results show that the residual degradation was within 7.0% even if we map-match trajectories sparsified at a rate of 40%.

The remainder of this paper is organized as follows. After providing a brief review of related works in Section 2, we propose the new map-matching problem and describe its key features along with a few preliminaries in Section 3. Next, we present our experimental results in Section 4. Finally, we conclude the paper and suggest future work in Section 5.

## 2. Related Works

On-line or local map-matching methods attach a newly observed GPS point to one of the neighboring links in the DRM. These methods use the coordinates, directions, and speeds localized to the current point to take into account the connectivities of these links [9], [14], [32]. By contrast, off-line or global map-matching methods consider the distance between a trajectory and a path in a DRM in the topological sense, from its origin to destination [7]. Alt et al. proposed a map-matching algorithm that utilizes Frechét distance [1], [2]. Algorithms with the relaxed Frechét distance have also been proposed [7], [12]. All these approaches simply map individual trajectories to the nearest paths in accordance with their own policy or distance function.

Due to the constraints on power consumption and transmission cost, trajectories are very sparse, and the above approaches do not always work well with low-sampled trajectories. To tackle these problems, probabilistic methods estimate the link from which the observation is made [24], [26], [33]. A multi-track map-matching method exploits the ensemble nature buried in the trajectories. It iteratively estimates the order of observations from different trajectories and the most likely segments from which they come [17]. A joint map-matching enumerates fixed-sized segments from a DRM and, using them as variables of the optimization problem, it discovers the paths to which the trajectories are assigned such that they seem to be as natural as routes [22]. This method, however, requires hyper parameters to balance the features such as the distance to the segments, stitching of segments, and regularity of the solution. Our method falls to this category that tackles the problem with low-sampled trajectories, which is not accessed well by single-track, especially on-line, map-matching methods. Therefore, we would concentrate on multi-track and off-line map-matching methods hereinafter.

Map-generation algorithms have been proposed for overall traffic analysis. Although governments, information companies, and social communities have manually developed DRMs, sometimes roads open or close either permanently or temporarily. To be adaptable, map-generation techniques maintain DRMs with less cost by automatically building them from a huge collection of observed GPS points. Some methods reconstruct DRMs through a series of dedicated procedures [11], [18], [19], and others do so based on Morse theory [31]. Their interests are mainly in building accurate DRMs, not in understanding the traffic flow. A method that consolidates trajectories to form a map does consider traffic flows to some extent [8], but it may not work on cases where the trajectories are sparsely sampled.

Many applications have been proposed for the analysis and prediction of traces. Some learn the repeated patterns of a car owner's history, e.g., commuting routes, the dropping off and picking up of family members, and visiting relatives or friends [13]. Turn prediction is another typical application for predicting which directions a car will take at an intersection, based on the route taken up to this point by learning others' traces [21]. These applications, however, are developed to predict particular purposes.

## 3. Proposed Method

### 3.1 Preliminaries

Let $i$, $j$ be non-negative integers and $k$, $n$, $p$, $q$, $N$, $K$ be natural numbers. We call $G = (V, E)$ a geometric graph, or simply a graph, where $V = \{(lon, lat) \mid lon, lat \in \mathbb{R}\}$ and $E = \{(u, v) \mid u, v \in V \text{ and } u \neq v\}$. Denoting a list as [ ] whose elements are ordered by $i$ or $j$, an element $\alpha$ in $\mathbb{P}$ is a trajectory of length $p$, where $\alpha = [\, \alpha^{(i)} \in \mathbb{R}^2 \mid i \leq p \,]$, and an element $\beta$ in $\mathbb{P}_G$ is a path in $G$ of length $q$, where $\beta = [\, \beta^{(j)} \in V \mid j \leq q$ where $(\beta^{(j-1)}, \beta^{(j)}) \in E$ if $j \geq 1 \,]$. Both trajectory and path in $G$ is also a polyline. Note that they may contain an element multiple times and that we have $\mathbb{P}_G \subset \mathbb{P}$. A path can be regarded as a sub-graph of $G$ and we denoted it as $G(\beta) = (V_\beta, E_\beta)$ where $V_\beta = \{\beta^{(j)} \mid j \leq q\}$ and $E_\beta = \{(\beta^{(j-1)}, \beta^{(j)}) \mid 1 \leq j \leq q\}$. To introduce binary set operators on graphs, we define an edge-induced graph of given $E$ as $G(E) = (V, E)$ where $V = \bigcup_{(u,v)\in E}\{u, v\}$. With this definition, given two graphs $G_i = (V_i, E_i)$ for $i \in \{1, 2\}$, we have $G(E_1) \circ G(E_2) = G(E_1 \circ E_2)$ where $\circ$ is a binary set operator.

**Definition 1.** *(Distance function): Let $d : \mathbb{P} \times \mathbb{P} \to \mathbb{R}$ be a distance function between polylines, where the following inequality and equalities hold for all $\alpha, \beta \in \mathbb{P}$:*

$$d(\alpha, \beta) \geq 0, \quad d(\alpha, \alpha) = 0, \quad d(\alpha, \beta) = d(\beta, \alpha).$$

**Definition 2.** *(Single-track map-matching) Let $G = (V, E)$ be a geometric graph and $d(\alpha, \beta)$ be a distance function between polylines. Given a trajectory $\alpha \in \mathbb{P}$, a single-track map-matching algorithm, or simply a map-matching algorithm, $\mathcal{M}_G : \mathbb{P} \to \mathbb{P}_G \times \mathbb{R}$ finds its minimizing path and its minimum distance $\hat{\beta}_G(\alpha) = \mathrm{argmin}_{\beta\in\mathbb{P}_G} d(\alpha, \beta)$ and $\hat{d}_G(\alpha) = \min_{\beta\in\mathbb{P}_G} d(\alpha, \beta)$, respectively.*

### 3.2 Stochastic Generative Model

Let $b(s) \in \mathbb{R}^2, 0 \leq s \leq q$ be a route, which is also described as a path $\beta \in \mathbb{P}_G$, such that:

$$b(s) = \begin{cases} v^{(j)} & \text{if } s = j, \\ (\lceil s \rceil - s)v^{(\lfloor s \rfloor)} + (s - \lfloor s \rfloor)v^{(\lceil s \rceil)} & \text{otherwise.} \end{cases} \quad (1)$$

Note that, without loss of generality, we attached the origin and destination of the route to the first and last vertices of the path, respectively.

The observations in a trajectory $\alpha \in \mathbb{P}$ are emitted on the route $b(s)$ at $s \in \{s^{(i)} \mid i = 0, \cdots, p\}$ such that $s^{(i)} < s^{(j)}$ for all $0 \le i < j \le p$. Additionally, assuming the first and last observations are made from the origin and destination of the route, respectively, we have $s^{(0)} = 0$ and $s^{(p)} = q$. Each observation has its observation noise $\epsilon_i \in \mathbb{R}^2$ and thus we have:

$$\beta^{(i)} = b(s^{(i)}) + \epsilon_i. \tag{2}$$

Also note that a trajectory has sampling noises that are induced by interpolating the finite number of observations comprising the trajectory.

In summary, although the routes are unobservable, the trajectories are observed and are emitted from one of the routes. We introduce the stochastic generative model with the observed and latent random variable $X, Z \in \mathbb{R}^K$, as follows:

**Latent variable** $Z$ is a 1-of-$K$ random variable whose realization is $z = (z_k)_{k=1}^K$, where $z_k \in \{0, 1\}$. There is a $k^*$ such that $z_k = 1$ if $k = k^*$ and $z_k = 0$ otherwise, which represents the occurrence of $k^*$-th route out of $K$ possible routes. The occurrence follows the prior probability distribution of $P(z) = \prod_{k=1}^K \pi_k{}^{z_k}$ such that $\sum_{k=1}^K \pi_k = 1$.

**Observed variable** $X$ is a random variable whose realization is $x = (x_k)_{k=1}^K$, which represents the distance between trajectory and path. This distance follows the probability distribution of $P(x|z) = \prod_{k=1}^K f(x_k|\sigma)^{z_k}$ where $f(x_k|\sigma) = \sigma \exp(-\sigma x_k)$ is the probability density function of exponential distribution. The parameter $\sigma$ is determined in accordance with the volume of the sampling noise.

### 3.3 Maximizing Posterior Probability

A generic EM algorithm maximizes the posterior probability of the stochastic model parameterized by $\theta$. With an initial $\theta^{\text{old}}$, it iterates the following E-step and M-step by replacing $\theta^{\text{old}}$ with $\theta^{\text{new}}$ until either $Q'$ or $\theta$ converges:

**E-step** updates the conditional probability $P(Z|X, \theta^{\text{old}})$, and

**M-step** finds the parameter $\theta^{\text{new}}$ that maximizes the log-likelihood of posterior probability:

$$Q'(\theta, \theta^{\text{old}}) = \sum_Z P(Z|X, \theta^{\text{old}}) \ln P(X, Z|\theta) + \ln P(\theta). \tag{3}$$

Given a collection of traces $T = \{\alpha_n \mid n = 1, \cdots, N\}$ and if let $x_n = (x_{nk})_{k=1}^K$ and $z_n = (z_{nk})_{k=1}^K$ be the independent realizations of the random variables $X$ and $Z$, respectively, we have the concrete E-step and M-step for the joint map-matching by applying the above probability distributions to the generic EM algorithm in a similar manner as that for a Gaussian mixture model [6]:

**E-step** evaluates the responsibility $\gamma(z_{nk})$ with the parameter $\pi^{\text{old}}$, and

**M-step** finds $\pi^{\text{new}}$ that maximizes the log-likelihood of posterior probability $Q'(\pi, \pi^{\text{old}})$ .

The responsibility and the log-likelihood are respectively defined as follows:

$$\gamma(z_{nk}) = \frac{\pi_k f(x_{nk}|\sigma)}{\sum_{k'=1}^K \pi_{k'} f(x_{nk}|\sigma)}, \tag{4}$$

$$Q'(\pi, \pi^{\text{old}}) = -\sum_{n=1}^N \sum_{k=1}^K \sigma\gamma(z_{nk}) x_{nk} + \sum_{k=1}^K z_k \ln \pi_k. \tag{5}$$

Although there are too many paths on the graph, it is practically sufficient to consider the paths that have shorter distances from each trajectory. This is feasible if we employ an algorithm [29] that can enumerate all the paths whose distances from the trajectory are within a certain threshold, such as $\sigma$. In extreme, considering just the minimizing path $\beta_{k^*} = \hat{\beta}_G(\alpha)$, we have $\gamma(z_{nk}) = 1$ if $k = k^*$ and $\gamma(z_{nk}) = 0$ otherwise. Assuming that the prior distribution is uniform, namely $\pi_k = 1/K$ for all $k$, the second term of Eq. (5) is straightforward and equal to $-K$. If we accept that $K$ is proportional to the description length of the geometric graph $G$, the joint map-matching is equivalent to the minimization problem below.

**Definition 3.** *(Route graph discovery) Let a hypothesis space of a graph be $\mathbb{G}$, a single-track map-matching be $\mathcal{M}_G$, and a collection of trajectories be $T$. A graph $G \in \mathbb{G}$ most likely emits the trajectories $T$ if it minimizes the following loss function:*

$$L(G; T) = \sum_{\alpha \in T} \hat{d}_G(\alpha) + \lambda \|G\|, \tag{6}$$

*where $\|G\|$ is the description length of the graph $G$, such as the total length of its edges, and $\lambda > 0$ is a hyper parameter.*

The first term is for the residual and the second term is for the regularization. This problem is equivalent to single-track map-matchings if $\lambda$ is zero. Otherwise, some edges are left unused so that $\|G\|$ decreases even though the distance $\hat{d}_G(\alpha)$ becomes longer for some trajectories.

### 3.4 Graph Exploration Algorithm

To minimize $L(G; T)$, we employ an exploratory search in the graph space, and obtain a decreasing series of graphs $G^{(t-1)} \supset G^{(t)}$ for $t = 1, 2, \cdots$ such that their losses also decreases. Let us denote the output of map-matching $\mathcal{M}_{G^{(t)}}$ as $\hat{\beta}_\alpha^{(t)} = \hat{\beta}_{G^{(t)}}(\alpha)$ and $\hat{d}_\alpha^{(t)} = \hat{d}_{G^{(t)}}(\alpha)$ for short.

Before presenting the important property that drives the exploration, we note that $\hat{d}_\alpha^{(t-1)} \le \hat{d}_\alpha^{(t)}$ always holds. This is trivial because if there were a path closer to $\alpha$ in $G^{(t)}$, it must be closer to $\alpha$ than the minimizing path in $G^{(t-1)}$ and this is contradictory. We do not care how the map-matching is implemented as long as it satisfies the inequality above.

The following theorem gives the condition to ensure that decreasing series of graphs decrease their losses.

**Theorem 1.** *Given a collection of trajectories $T$, and two graphs $G^{(t-1)}$ and $G^{(t)}$, $L(G^{(t)}; T) < L(G^{(t-1)}; T)$ holds iff the following inequality holds:*

$$\lambda \|\Delta^{(t)}\| > \sum_{\alpha \in T_{|\Delta^{(t)}}} \left\{ \hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)} \right\}, \tag{7}$$

*where $\Delta^{(t)} = G^{(t-1)} \setminus G^{(t)}$ and $T_{|\Delta^{(t)}}$ denotes the collection of trajectories whose minimizing paths run through $\Delta^{(t)}$.*

*Proof.* By evaluating the difference between losses of the two consecutive graphs in the series, we have the following:

$$L(G^{(t)}; T) - L(G^{(t-1)}; T)$$

$$= \left( \sum_{\alpha \in T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t)} + \sum_{\alpha \notin T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t)} + \lambda \| G^{(t)} \| \right)$$
$$- \left( \sum_{\alpha \in T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t-1)} + \sum_{\alpha \notin T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t-1)} + \lambda \| G^{(t-1)} \| \right)$$
$$= \sum_{\alpha \in T_{|\Delta^{(t)}}} \left( \hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)} \right) - \lambda \| \Delta^{(t)} \|.$$

Note that $\hat{d}_\alpha^{(t)} = \hat{d}_\alpha^{(t-1)}$ holds for $\alpha \notin T_{|\Delta^{(t)}}$ because both $\hat{\beta}_\alpha^{(t)}$ and $\hat{\beta}_\alpha^{(t-1)}$ are irrelevant with $\Delta^{(t)}$. It then follows that, Eq. (7) holds *iff* $L(G^{(t)}; T) < L(G^{(t-1)}; T)$ holds. □

**Algorithm 1** is the pseudo-code for the route graph discovery. Given a collection of trajectories $T$ and an initial graph $G^{(0)}$, e.g., a DRM, it finds the final graph that minimizes the loss (line 6). Note that $B$ maintains the minimizing path and the minimum distance for each trajectory $\alpha$ throughout every $t$-th stage (line 17). The main loop (lines 2–5) explores a series of subgraphs with decreasing losses as explained in **Theorem 1**. First, an edge $e$ is selected, for instance, in increasing order of the cardinality, which is the number of the minimizing paths running through it (line 3), and a new graph $G^{(t)}$ is obtained by *re-routing* with the edge $e$ disabled (line 4). We explain later what re-routing is, as well as why and how we select a single edge. Then, the graph $G^{(t)}$ is probabilistically accepted or rejected (line 4). Finally, the main loop either continues or breaks in accordance with the history of the obtained graphs (line 5).

Next, we explain how to obtain the new graph $G^{(t)}$ by finding $\Delta^{(t)}$. The re-routing technique serves this by map-matching with some edges of graph $G^{(t)}$ disabled (line 12). Note that trajectories not in $\Delta^{(t)}$ are irrelevant to $\Delta^{(t)}$ and that any edge $e$ in $\Delta^{(t)}$ satisfies the following inequality:

$$g \subseteq \Delta^{(t)} \subseteq \overline{\Delta}_g^{(t)}$$
$$\text{where } \begin{cases} g = G(\{e\}), \\ \overline{\Delta}_g^{(t)} = G^{(t-1)} \setminus \bigcup_{\alpha \notin T_{|g}} G(\hat{\beta}_\alpha^{(t-1)}), \end{cases}$$

and $T_{|g}$is the collection of trajectories whose minimizing paths run through $g$. Thus, the following strategy works: first conservatively select an edge $e$ from $G^{(t)}$ and optimistically initialize $\Delta^{(t)}$ with $\overline{\Delta}_g^{(t)}$ (line 8), as well as the cumulative differential residual $\epsilon$ with 0 (line 9). Then, as we re-route a trajectory in $T_{|g}$, $\Delta^{(t)}$ is subtracted by $G(\hat{\beta}_\alpha^{(t)})$, and $\epsilon$ is added by the differential residual before and after the re-routing (lines 12–14). In this implementation, we employed Zeheng et al.'s algorithm [33] for re-routing.

The procedure terminates as soon as it becomes obvious that Eq. (7) will never be satisfied (lines 15, 16). This is safe because of **Theorem 1** and, notably, this saves much computation by skipping unnecessary map-matchings. If no early termination has occurred, the procedure returns with the new reduced graph as $G^{(t)}$ (line 18). We describe four implementation issues in the following sections.

### 3.4.1 Meta-heuristic Optimization in Main Loop

The reason we probabilistically accept the graph at line 4 is to escape local minima. For simplicity, however, the current implementation always accepts the returned graph $G^{(t)}$. Other meta-heuristic algorithms are also applicable, in addition to this implementation.

**Algorithm 1** Building route graph

**Require:** trajectories $T$, minimizing paths $B$

1: **procedure** MAIN($G^{(0)}$)
2: **for all** $t = 1, 2, 3, \cdots$ **do**
3: select $e$ from edges in $G^{(t-1)}$
4: $G^{(t)} \leftarrow$ APPLY($G(\{e\}), G^{(t-1)}$) with some probability
5: break by history $\cdots, G^{(t-1)}, G^{(t)}$
6: report $G^{(t)}$
7: **function** APPLY($g, G^{(t-1)}$)
8: let $\Delta^{(t)}$ be subgraph only $T_{|g}$run
9: $\epsilon \leftarrow 0$
10: **for all** $\alpha \in T_{|g}$ **do**
11: $\hat{\beta}_\alpha^{(t-1)}, \hat{d}_\alpha^{(t-1)} \leftarrow B[\alpha]$
12: $\hat{\beta}_\alpha^{(t)}, \hat{d}_\alpha^{(t)} \leftarrow \mathcal{M}_{G \setminus g}(\alpha)$
13: $\Delta^{(t)} \leftarrow \Delta^{(t)} \setminus G(\hat{\beta}_\alpha^{(t)})$
14: $\epsilon \leftarrow \epsilon + (\hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)})$
15: **if** $\|\Delta^{(t)}\| < \epsilon$ **then**
16: **return** $G^{(t-1)}$
17: update $B[\alpha]$ with $\hat{\beta}_\alpha^{(t)}, \hat{d}_\alpha^{(t)}$ for $\alpha \in T_{|g}$
18: **return** $G^{(t-1)} \setminus \Delta^{(t)}$

### 3.4.2 Initial Graph

A DRM is one candidate of the initial graph $G^{(0)}$. A Delauney graph whose vertices are GPS observations is another. In the former case, **Algorithm 1** performs a joint map-matching. If it is certain that the GPS traces are from objects moving on a DRM, this is the reasonable option. We chose this option for the sake of experimentation in Section 4. In the latter case, **Algorithm 1** performs a map-generation. This is useful when no DRM is available, although GPS observations should be carefully sampled if the density differs from place to place over the two-dimensional space. This is because too many observations increase the computation whereas too few observations decrease the accuracy of the route graph.

### 3.4.3 Map-matching Algorithm

As mentioned at the beginning of Section 3.4, the route graph discovery employs a map-matching algorithm that follows **Definition 2**. The current implementation employed Zeheng et al. [33] because it is easy to implement. Simply introducing their algorithm, it performs the A* algorithm to find the shortest path between two vertices in the graph. Considering the combinations of vertices, each of which is one of the neighbors of consecutive observations within a window of size $w$, their algorithm finds the route that minimizes the cost function of the A* algorithm so long as $w$ is large enough. Thus, this algorithm satisfies the requirement of **Definition 2**. Notably, any map-matching algorithm could be used as long as it follows **Definition 2**.

### 3.4.4 Selection of Edges

There can be several priorities when selecting a disabled edge:

( 1 ) by the length of edge,
( 2 ) by the cardinality of edge,
( 3 ) by both the length and cardinality,
( 4 ) by the size of $\Delta^{(t)}$ for the edge, and
( 5 ) at random.

Except for the fifth option, these share the idea of first selecting an edge that is most unlikely to remain in the final graph. Although the fourth option is an exact greedy method, too much computation is needed because it requires roughly as many re-routings as the average cardinality times the number of edges in just one iteration. The first to the third options approximate the preference of edges without the eager computation of $\Delta^{(t)}$. Let us consider the case where a DRM is the initial graph. Intuitively, the length of the edge seems irrelevant to the likelihood that it will remain in the final graph. For this reason, the second one is the option we take because the first and the third employ the length of the edge. Note that, whether an edge is selected with or without replacement is another option. This implementation never replaces edges, but only selects an edge once because the reduced graph is always accepted at line 4.

## 4. Experiments

In this section, we examine whether our algorithm is able to estimate unobservable paths from sampled trajectories using the benchmark datasets. First, we explain the experimental configurations and then present the results.

We implemented the algorithm with Python and run it on an Ubuntu 16.04 box equipped with Intel Xeon E5-2623 v3 3.00 GHz and 256 GB memory. The process employed 16 cores managed by multiprocessing module that comes with Python such that each re-routing runs concurrently for computational efficiency.

### 4.1 Experimental configurations

We use the benchmark datasets of GPS traces, which are collected and shared by various research groups or volunteers. The DRM should be contemporary with the GPS traces, although we utilized the Open Street Map (OSM) of 2017. **Table 1** shows descriptions of the datasets and their corresponding DRMs. Note that they might differ from those in other reports because those datasets were preprocessed differently.

The goal of this experiments was to examine the accuracy of the algorithms in estimating unobservable paths from sparsified trajectories. These paths, however, are never available because the traces have been originally sampled. As such, we must make some assumptions to evaluate the accuracy even in the following settings:

- an unobservable path is a series of connected links in the DRM, indicating that a car drives on the roads,
- both the observation and sampling noises of the traces are sufficiently small,
- a certain algorithm, such as those using the Frechét distance [1], [29], can map a trace to its unobservable path if it contains sufficiently little noise.

The first setting is acceptable because the trajectories in the datasets we are using are all from those of cars or bikes. The third setting is also acceptable because map-matching is trivial in that unrealistic case. Even though the second setting depends on datasets, for the sake of experimentation, we decided to accept it.

Table 1 Popular GPS trace datasets.

| | GPS | | OSM | |
|---|---|---|---|---|
| name | #pts | #trace | #nodes | #edges |
| icdm | 2859950 | 4257 | 18716 | 35170 |
| bikely | 549920 | 3150 | 262699 | 540017 |
| chicago | 118360 | 889 | 46533 | 88942 |

In the experiments, we sampled observations in a trajectory-wise manner with variable rates, and compared the residuals of the following:
( 1 ) unsampled trajectory from DRM (lower bound),
( 2 ) unsampled trajectory from route graph (proposed),
( 3 ) sampled trajectory from DRM (upper bound).

The first situation gives the lower bounding residual, in that no algorithm can do better than this method, as we had accepted the three above assumptions. The second method is our proposed method. To evaluate how well the route graph represents the major streams in the GPS traces compared to the DRM, we evaluated the residual of the unsampled trajectory from the route graph. The third method evaluates the residual arising from the injection of sampling noises. This gives the upper bounding residual in the sense that no off-the-shelf single-track map-matching does worth than this.

### 4.2 Experimental results

**Figure 1** shows example results of a single-track and the proposed map-matching algorithms with variable sampling rates. Note that the unobservable path is identical to the result of single-track map-matching in the bottom picture. As the sampling rate increases, the results get similar with each other and they finally become almost identical. The estimated routes by the single-track map-matching cling to the traces because they minimize the Frechét distances. Especially with the lowest sampling rate, the route unnaturally comes and goes across the river. On the other hand, the estimated routes by the proposed method are less dependent on the sampling rates than those by the single-track map-matching. Furthermore, with the unsampled trace, the proposed method estimates the even natural route because of the regularization term as we will mention later.

**Figure 2** describes the residuals of the three methods with variable sampling rates. Note that the lower bound is constant because the first method is irrelevant with the sampling rates. We can see that the upper bound curve steeply increases as the sampling rate decreases. This is what we expected, as the lower is the sampling rate, the more the residual experiences injected sampling noises.

In contrast, the curve for the proposed method increases moderately. For instance, in the `icdm` and `chicago` datasets, the proposed method reduced the residual by more than 70% and 40%, respectively, for the upper bound at sampling rate of 40%. The reduction rate tends to increase when the dataset has a larger number of trajectories, which means that the proposed method leverages the residual using the other trajectories. Indeed, we can see that the `icdm` was able to decrease the sampling rate to 40% while the degradation of residual remained nearly constant at 7.0%.

The residual is slightly larger than the upper bound at a sampling rate of 100%, this is because our algorithm accepts a slight increase in the residuals to reduce the route graph. The behavior of the residual as well as the empirical loss is well understood in the regularization technique. The results may accordingly indi-

**IPSJ Transactions on Mathematical Modeling and Its Applications Vol.12 No.1 43–51 (Mar. 2019)**

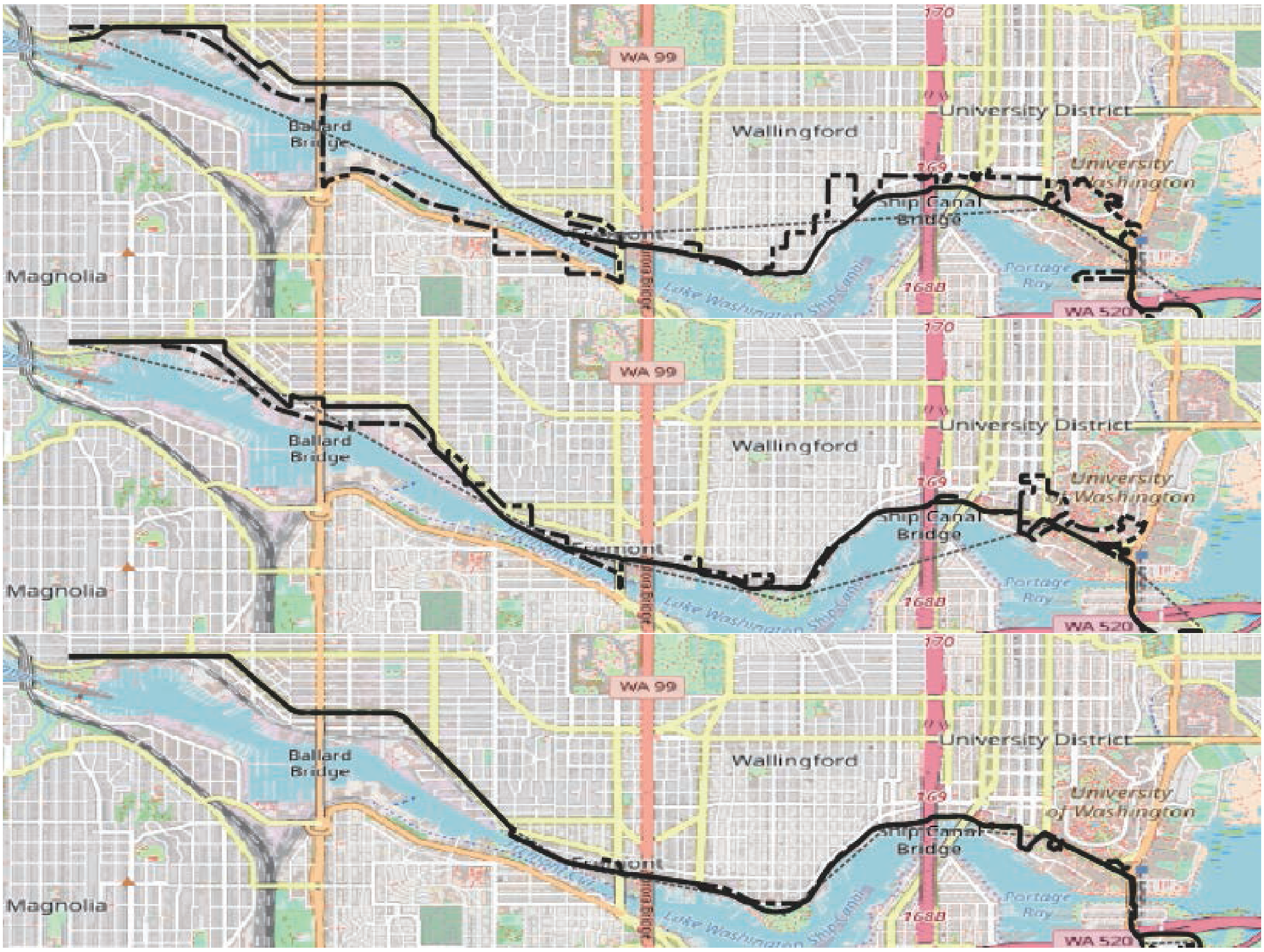


**Fig. 1** Estimated routes by single-track (chain), proposed (solid) map-matching, and sampled trace (dashed) from bikely with rate 20% (top), 40% (middle), and 100% (bottom).

cate that our algorithm may further exploit the observation and sampling noises.

**Figure 3** describes the description length of graph with variable sampling rates, which corresponds to the regularization term that we introduced to the loss function. In all three datasets, the description length monotonically and asymptotically increases as the sampling rate increases. By contrast, the residual monotonically and asymptotically decreases in Fig. 2, which shows that our algorithm reasonably favors these contradictory terms depending on the sampling rate.

### 4.3 Discussions

In this section, we discuss the differences between the existing methods and our method. Recall that we have been trying to estimate unobservable paths from traces with sample noise by proposing a new joint map-matching method. Therefore, we would discuss the relation between a joint map-matching [22] and our proposal, as well as the expected advantage of our method to the proceeding stochastic method [26].

Li et al. [22] formalized a joint map-matching as an optimization problem whose objective function contains residual, stitching, and regularization terms. Their method is essentially similar to our method as these two methods share two terms in their objective functions, although the proceeding method requires the stitching term to penalize the fragmentations introduced by its formalization. They differ, however, in terms of algorithms. The proceeding method requires three hyper-parameters while ours requires a single hyper-parameter which can be fixed as $\lambda = 1$ with a good reason. And we remarkably disclosed that the residual and regularization terms are obtained as we formalize the joint map-matching as a generative stochastic process. Concretely, the residual term is caused by the assumption that the distance between trace and unobservable path follows the exponential probability distribution, and that the regularization term is from the other assumption that the paths are equally likely.

Then, we compare our method to another stochastic method that employs HMM [26]. They has two major differences. One is that the latent variables of the HMM-based method correspond to the vertices from which the observations come, while those of our method correspond to the paths from which the traces come. The other is that the HMM-based method maximizes likelihood estimator (MLE), while ours maximizes a posterior (MAP). Thus, we suppose that the HMM-based method likely overfits without restricting the number of hidden states by manually tuning the complexity of the graph. On the other hand, our method automatically determines the complexity of the graph depending on the

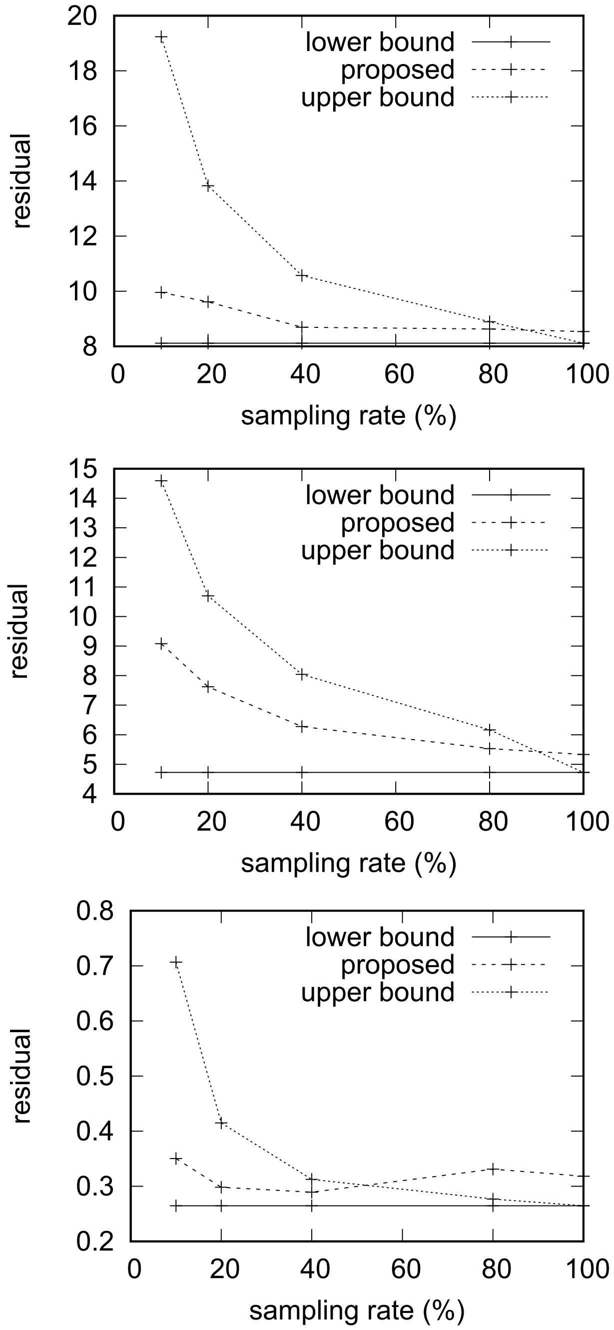


**Fig. 2** Residuals of icdm (top), bikely (middle), and chicago (bottom).

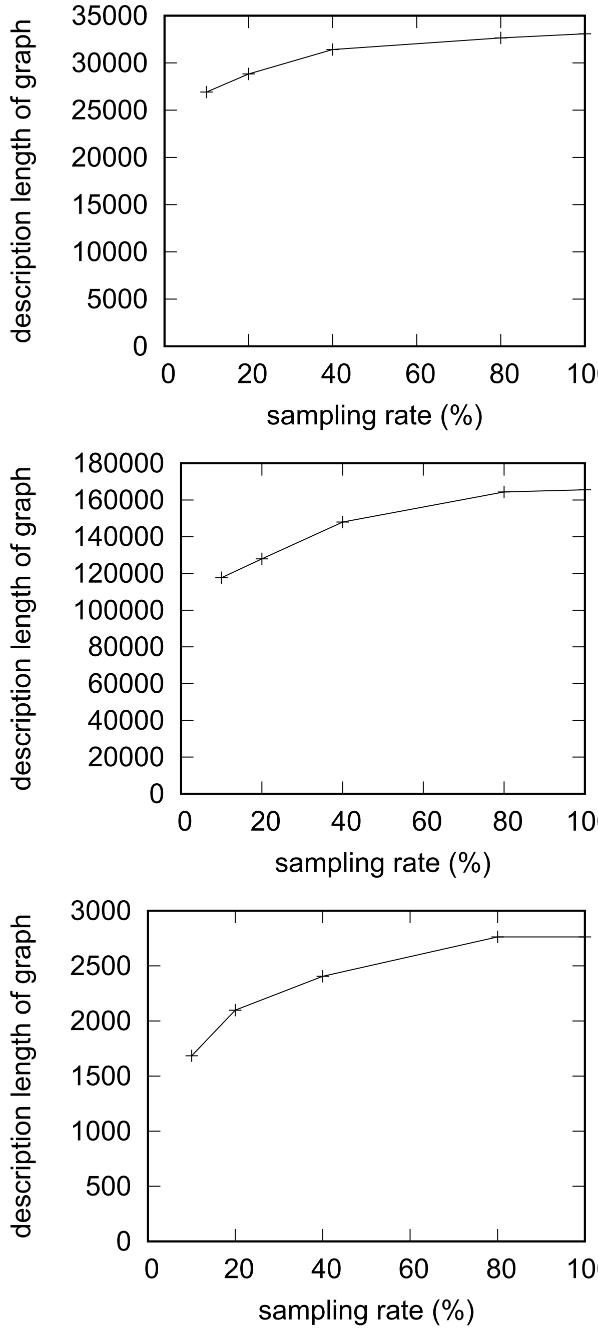


**Fig. 3** $||G||$ of icdm (top), bikely (middle), and chicago (bottom).

number of GPS observations. Because of these differences, we believe that the proposed method has advantages especially with a low sampling rate or less traces.

## 5. Conclusion

In this paper, we proposed a joint map-matching method based on the generative model for estimating unobservable paths by maximizing the posterior probability. Maximization is achieved by the EM algorithm whose object function consists of residual and regularization terms. We presented an iterative algorithm for exploring the route graph, which avoids as many map-matchings as possible by taking advantage of the proven property holding of the residual and the regularization terms. The experimental results showed that the residual degradations from the lower bound were no more than 7.0% when the sampling rate was reduced to 40%. This means that this algorithm reduces the volume of sampling noises and identifies the major streams in the trajectories.

In the future, we plan to continue this work in three directions: first, by realizing performance enhancements by further reducing the costly map-matching. One idea is to localize the re-routing to the disabled links without performing map-matching from the origin to destination of the trajectories. The other idea is to extend our algorithm to the incremental one that updates the route graph as trajectories arrive in sequence. The second direction is to develop more sophisticated formulations of the EM algorithm without considering the extreme case, or applying another generative model such as the Hidden Markov Model. The third and final direction is to apply our method to demand analysis, urban design, and other applications.

## References


[1] Alt, H., Efrat, A., Rote, G. and Wenk, C.: Matching planar maps, *Journal of Algorithms*, Vol.49, No.2, pp.262–283 (online), DOI: 10.1016/S0196-6774(03)00085-3 (2003).

[2] Alt, H. and Godau, M.: Computing the Fréchet distance between two polygonal curves, *International Journal of Computational Geometry & Applications*, Vol.5, No.1-2, pp.75–91 (online), DOI: 10.1142/S0218195995000064 (1995).

[3] Alvarez-Garcia, J., Ortega, J., Gonzalez-Abril, L. and Velasco, F.: Trip destination prediction based on past GPS log using a Hidden Markov Model, *Expert Systems with Applications*, Vol.37, No.12, pp.8166–8171 (online), DOI: 10.1016/j.eswa.2010.05.070 (2010).

[4] Benkert, M., Gudmundsson, J., Hübner, F. and Wolle, T.: Reporting flock patterns, *Computational Geometry*, Vol.41, No.3, pp.111–125 (online), DOI: 10.1016/j.comgeo.2007.10.003 (2008).

[5] Bernstein, D. and Kornhauser, A.: An introduction to map matching for personal navigation assistants, *The Transportation Research Board 77th Annual Meeting*, Washington, D.C. (1996).

[6] Bishop, C.M. and M., C.: *Pattern recognition and machine learning*, Springer (2006).

[7] Böhm, K., Pfoser, D., Salas, R. and Wenk, C.: On map-matching vehicle tracking data, *Proc. 31st International Conference on Very Large Data Bases*, pp.853–864, ACM (2005).

[8] Cao, L. and Krumm, J.: From GPS traces to a routable road map, *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, p.3, ACM Press (online), DOI: 10.1145/1653771.1653776 (2009).

[9] Chawathe, S.S.: Segment-Based Map Matching, *2007 IEEE Intelligent Vehicles Symposium*, pp.1190–1197, IEEE (online), DOI: 10.1109/IVS.2007.4290280 (2007).

[10] Chen, L., Lv, M. and Chen, G.: A system for destination and future route prediction based on trajectory mining, *Pervasive and Mobile Computing*, Vol.6, No.6, pp.657–676 (online), DOI: 10.1016/J.PMCJ.2010.08.004 (2010).

[11] Davies, J., Beresford, A. and Hopper, A.: Scalable, Distributed, Real-Time Map Generation, *IEEE Pervasive Computing*, Vol.5, No.4, pp.47–54 (online), DOI: 10.1109/MPRV.2006.83 (2006).

[12] Eiter, T. and Mannila, H.: Computing discrete Fréchet distance, *Notes*, Vol.94, p.64 (online), DOI: 10.1.1.90.937 (1994).

[13] Froehlich, J. and Krumm, J.: Route Prediction from Trip Observations (online), DOI: 10.4271/2008-01-0201 (2008).

[14] Greenfeld, J.: Matching GPS observations to locations on a digital map, *Transportation Research Board 81st Annual Meeting*, Washington D.C. (2002).

[15] Hsueh, Y.-L. and Chen, H.-C.: Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions, *Information Sciences*, Vol.433–434, pp.55–69 (online), DOI: 10.1016/J.INS.2017.12.031 (2018).

[16] Javanmard, A., Haridasan, M. and Zhang, L.: Multi-track Map Matching (online), available from ⟨http://arxiv.org/abs/1209.2759⟩ (2012).

[17] Javanmard, A., Haridasan, M. and Zhang, L.: Multi-track map matching, *Proc. 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, p.394, ACM Press (online), DOI: 10.1145/2424321.2424373 (2012).

[18] Karagiorgou, S. and Pfoser, D.: On vehicle tracking data-based road network generation, *Proc. 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, p.89, ACM Press (online), DOI: 10.1145/2424321.2424334 (2012).

[19] Karagiorgou, S., Pfoser, D. and Skoutas, D.: Segmentation-based road network construction, *Proc. 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*, pp.460–463, ACM Press (online), DOI: 10.1145/2525314.2525460 (2013).

[20] Krumm, J., Gruen, R. and Delling, D.: From destination prediction to route prediction, *Journal of Location Based Services*, Vol.7, No.2, pp.98–120 (online), DOI: 10.1080/17489725.2013.788228 (2013).

[21] Krumm, J. and John: Where will they turn: predicting turn proportions at intersections, *Personal and Ubiquitous Computing*, Vol.14, No.7, pp.591–599 (online), DOI: 10.1007/s00779-009-0248-1 (2010).

[22] Li, Y., Huang, Q., Kerber, M., Zhang, L. and Guibas, L.: Large-scale joint map matching of GPS traces, *Proc. 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*, pp.214–223, ACM Press (online), DOI: 10.1145/2525314.2525333 (2013).

[23] Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G. and Zhu, Y.: Mining large-scale, sparse GPS traces for map inference, *Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12*, p.669, ACM Press (online), DOI: 10.1145/2339530.2339637 (2012).

[24] Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W. and Huang, Y.: Map-matching for low-sampling-rate GPS trajectories, *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, p.352, ACM Press (online), DOI: 10.1145/1653771.1653820 (2009).

[25] Miklušċák, T., Gregor, M. and Janota, A.: Using Neural Networks for Route and Destination Prediction in Intelligent Transport Systems, pp.380–387, Springer (online), DOI: 10.1007/978-3-642-34050-5-43 (2012).

[26] Newson, P. and Krumm, J.: Hidden Markov map matching through noise and sparseness, *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, p.336, ACM Press (online), DOI: 10.1145/1653771.1653818 (2009).

[27] Pink, O. and Hummel, B.: A statistical approach to map matching using road network geometry, topology and vehicular motion constraints, *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp.862–867, IEEE (online), DOI: 10.1109/ITSC.2008.4732697 (2008).

[28] Quddus, M.A., Ochieng, W.Y., Zhao, L. and Noland, R.B.: A general map matching algorithm for transport telematics applications, *GPS Solutions*, Vol.7, No.3, pp.157–167 (online), DOI: 10.1007/s10291-003-0069-z (2003).

[29] Shigezumi, J., Asai, T., Morikawa, H. and Inakoshi, H.: A Fast Algorithm for Matching Planar Maps with Minimum Fréchet Distances, *Proc. 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data - BigSpatial '15*, pp.25–34, ACM Press (online), DOI: 10.1145/2835185.2835186 (2015).

[30] Tiwari, V.S., Arya, A. and Chaturvedi, S.: Route prediction using trip observations and map matching, *2013 3rd IEEE International Advance Computing Conference (IACC)*, pp.583–587, IEEE (online), DOI: 10.1109/IAdCC.2013.6514292 (2013).

[31] Wang, S., Wang, Y. and Li, Y.: Efficient map reconstruction and augmentation via topological methods, *Proc. 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, pp.1–10, ACM Press (online), DOI: 10.1145/2820783.2820833 (2015).

[32] Wenk, C., Salas, R. and Pfoser, D.: Addressing the Need for Map-Matching Speed: Localizing Globalb Curve-Matching Algorithms, *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, pp.379–388, IEEE (online), DOI: 10.1109/SSDBM.2006.11 (2006).

[33] Zheng, Y., Xie, X., Zhang, C., Lou, Y., Wang, W. and Huang, Y.: Map-Matching for Low-Sampling-Rate GPS Trajectories (2009).


**Hiroya Inakoshi** received his B.S. and M.E. degrees in Engineering from Kyoto University in 1993 and from Graduate School of Engineering in Kyoto University 1995, respectively. He is currently engaged in the research on data mining and machine learning with Fujitsu Laboratories Ltd. He is a member of IEICE and IPSJ.

**Junichi Shigezumi** received his M.S. and Dr. (Mathematics) degrees from Kyushu University in 2006 and 2009, respectively. He was engaged in research on spatio-temporal data mining with Fujitsu Laboratories Ltd. His current research interests include neuroscience and artificial intelligence. He is a member of IPSJ, JSIAM and ORSJ.

**Tatsuya Asai** received his M.E. degree from Nagoya University in 2001 and received Dr. (Information Science) degree from Kyushu University in 2004. He was engaged in research on XML database systems with Fujitsu Laboratories Ltd. from 2004. His current research interests include discovery science. He is a member of IPSJ and JSAI.

**Takuya Kida** received his B.S. degree in Physics, M.S. and Dr. (Information Science) degrees all from Kyushu University in 1997, 1999, and 2001, respectively. He was a full-time lecturer of Kyushu University Library from October 2001 to March 2004. He is currently an associate professor of the division of computer science, Graduate School of Information Science and Technology, Hokkaido University, since April 2004. His research interests include text algorithms and data structures, information retrieval, and data compression. He is a member of IEICE, IPSJ, and DBSJ.

**Hiroki Arimura** received his B.S degree in 1988 in Physics, M.S. and Dr.Sci. degrees in 1990 and 1994 in Information Systems from Kyushu University. From 1990 to 1996, he was at the Department of Artificial Intelligence in Kyushu Institute of Technology, and from 1996 to 2004, he was at the Department of Informatics in Kyushu University. Since 2006, he has been a professor of Graduate School of Information Science and Technology Hokkaido University, Sapporo, Japan. His research interests include data mining, computational learning theory, information retrieval, and algorithm design. He is a member of JSAI, IPSJ, and ACM.