Exploration of Ground Truth from Raw GPS Data

Huajian Mao
National University of Defense Technology
State Key Laboratory of High Performance Computing
Hunan 410073, China
huajianmao@nudt.edu.cn

Wuman Luo, Haoyu Tan, Lionel M. Ni
Hong Kong University of Science and Technology
Hong Kong
{luowuman, hytan, ni}@cse.ust.hk

Nong Xiao
National University of Defense Technology
State Key Laboratory of High Performance Computing
Hunan 410073, China
nongxiao@nudt.edu.cn

ABSTRACT

To enable smart transportation, a large volume of vehicular GPS trajectory data has been collected in the metropolitan-scale Shanghai Grid project. The collected raw GPS data, however, suffers from various errors. Thus, it is inappropriate to use the raw GPS dataset directly for many potential smart transportation applications. Map matching, a process to align the raw GPS data onto the corresponding road network, is a commonly used technique to calibrate the raw GPS data. In practice, however, there is no ground truth data to validate the calibrated GPS data. It is necessary and desirable to have ground truth data to evaluate the effectiveness of various map matching algorithms, especially in complex environments. In this paper, we propose truthFinder, an interactive map matching system for ground truth data exploration. It incorporates traditional map matching algorithms and human intelligence in a unified manner. The accuracy of truthFinder is guaranteed by the observation that a vehicular trajectory can be correctly identified by human-labeling with the help of a period of historical GPS dataset. To the best of our knowledge, truthFinder is the first interactive map matching system trying to explore the ground truth from historical GPS trajectory data. To measure the cost of human interactions, we design a cost model that classifies and quantifies user operations. Having the guaranteed accuracy, truthFinder is evaluated in terms of operation cost. The results show that truthFinder makes the cost of map matching process up to two orders of magnitude less than the pure human-labeling approach.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

System, Measurement

UrbComp'12, August 12, 2012. Beijing, China

Keywords

Ground truth, map matching, GPS data

1. INTRODUCTION

Smart transportation is expected to play an important role to meet the growing demand of various transportation-related services from citizens [16] and government officers [2], especially in modern cities. A fundamental requirement to smart transportation is to collect the dynamic vehicular location data to form the basis to build an effective traffic information system [4]. The collected large-scale vehicular dataset is subject to further analysis, such as traffic estimation [15], hot spot detection [9], driving pattern recognition [8], traffic mining [5], and similar routes discovery [3], before the goals of smart transportation can be achieved. Real-time vehicular data collection is the first step toward smart transportation. For example, in Shanghai Grid (SG) project [7], most of the public vehicles are equipped with a GPS and a GPRS wireless communication module. Each of these vehicles periodically sends GPS reports to a data center. In the current implementation of the SG project, a very large volume (about 3-4 million records per day) of vehicular GPS trajectory data has been collected with different techniques.

The collected raw GPS data unfortunately suffers from two major errors. First, due to the limitation of the GPS technology, the vehicle location coordinates are not necessary precise mainly due to environmental factors. Second, a vehicle's location trajectory is reported in discrete samples for cost concern of GPRS communication. Even worse, the reporting or sampling interval may be adjusted by the driver in the SG project. Thus, it is challenging to estimate a vehicle's location during the sampling interval. Consequently, it is inappropriate to use the raw GPS dataset directly, which may lead to inaccurate conclusions or decisions for the potential smart transportation applications.

Due to the problems introduced above, raw GPS data calibration or recovery is the next important step toward smart transportation. An intuitive approach to correct the raw GPS data is to align the data onto the corresponding road network to find out a sequence of road segments that a vehicle has traveled along. This process is usually referred to as *map matching* [13] [10]. Typically, a good map matching should possess the desirable property of high accuracy which is evaluated by a complete validation. In practice, however, there is no ground truth data to validate the calibrated GPS data. Although map matching has been studied for many years, there still exist several challenging problems due to

the lack of ground truth data. First, to validate the accuracy of a map matching algorithm, a ground truth path is required to compare with the output of the algorithm. Very few existing map matching algorithms provide a meaningful validation technique due to the aforementioned reason. Second, it is necessary and desirable to have ground truth data to tune and evaluate the effectiveness of various map matching algorithms. Since most map matching algorithms are heuristic, their accuracy is strongly related to the tuning and selection of various design parameters. However, the parameters should be tuned with the ground truth data. Wrong parameters will lead the algorithm inaccurate. Therefore, finding a complete trajectory ground truth is critical to the map matching research.

We observe that most of the ground truth path of the trajectories can be correctly identified by human-labeling on the historical raw GPS dataset. It is believed that human-labeled data can be almost 100% accurate and it is widely used to explore ground truth dataset to evaluate map matching algorithms [10] [17]. In general, a human labeling process involves both cognitive works (e.g., determining the road segment for a particular GPS report) and manual works (e.g., recording the sequence of road segment identifier). Since this process involves too much human intelligence and action, it is usually not feasible to apply pure human-labeling to large GPS datasets.

To solve this problem, in this paper, we propose truthFinder, an interactive map matching system for ground truth data exploration. The goal of truthFinder is to minimize the human involvement. Specifically, we try to let the user interact with the system as little as possible. Formally, the goal of truthFinder is defined as follow: *For a given trajectory T and a road network $G(V, E)$, we want to explore the ground truth path P with a small cost C in terms of operations.* For this purpose, there are several challenges. First, it is difficult to quantify the cost of human interaction. For this challenge, we propose a cost model for truthFinder to measure the efficiency of the method. Second, using the visualization of the trajectory and the digital map is not trivial. For example, the trajectory may contain the same road segment twice or more. We should avoid such overlapping in visualization and allow the user to select anyone of them. With this issue, we introduce several techniques (e.g., multi-layer presentation for showing trajectories and paths, and multi-color notation for the candidate roads) to make it convenient to explore the ground truth. Third, the existing map matching algorithms should be modified to be stable. As such, we propose an interactive map matching system, that is, taking the users interaction into account, the trajectory generated at the next round should be more accurate than the current one.

ThruthFinder incorporates traditional map matching algorithms and human intelligence in a unified manner. The accuracy of truthFinder is guaranteed by the observation that a vehicular trajectory can be correctly identified by human-labeling with the help of a period of historical GPS dataset. To measure the cost of human interactions, we design a cost model that classifies and quantifies user operations. Having the guaranteed accuracy, truthFinder is evaluated in terms of operation cost. The results show that truthFinder makes the cost of map matching process up to two orders of magnitude less than the pure human-labeling approach. To sum up, our contributions are as follows:

- We design a cost model that classifies and quantifies user interactions. Our model avoids absolute measurements of human behaviors. Instead, we define several operations with regard to our system and use the number of each operation in cost analysis.
- We propose the architecture and implementation issues of truthFinder in detail. We are arguably the first to offer an interactive map matching system. Our design can be easily generalized for similar purposes.
- We provide a method to explore the ground truth path data from raw GPS trajectory data while guarantying the accuracy for different situations at the same time. In this way, the issues of map matching algorithm validation can be overcome by using truthFinder.
- Our system is evaluated in terms of operation cost. The experimental results show that truthFinder significantly outperforms traditional method of exploring ground truth data from scratch.

The rest of this paper is organized as follows. Section 2 describes the prior related work in detail. Section 3 shows the system architecture design. Section 4 puts forward the cost model of our interactive map matching system for ground truth exploration. Section 5 gives the evaluation of our work based on our implemented prototype system. We conclude our paper and present the future directions in Section 6.

2. RELATED WORK

The truthFinder system shares its design and consideration with several recent efforts of data calibration work. We categorize the related works into two groups as the map matching algorithms and the methods of ground truth path exploration.

2.1 Map Matching

Map matching has been studied in many litterateurs [13] [10] [1] [11]. Different map matching algorithms have different strategies varying from those using simple search techniques to those using more advanced techniques. In [13], the authors present an in-depth literature review of map matching algorithms. Generally, the existing algorithms are classified into four classes: 1) *geometric analysis*, which makes use of the geometric information of the spatial road network data by considering only the shape of the links [6]; 2) *topological analysis*, which makes use of the geometry of the links as well as connectivity and contiguity of the links [14]; 3) *probabilistic map matching algorithms* [12] and 4) *advanced map matching algorithms*, which use more refined concepts such as a Kalmam Filter or a fuzzy logic model or a Hidden Markov Model [11].

2.2 Ground Truth Exploration

Generally, according to the dataset used in the evaluation of the aforementioned map matching works, ground truth path exploration methods can be classified into three classes:

Datasets collected by driving vehicles. The researchers of [1] [11] drive around the city, and periodically record the GPS positions together with the roads where they drive on. At the end of the travel, they will get a sequence of the raw GPS reports, along with the path they have passed. Each of the reports will be assigned with a road segment

to indicate where the vehicle is at the time it is reported. After the assignment, the GPS data contains both the reports information and the topological information. Then, the GPS reports are used as the input of map matching, while the paths are treated as the ground truth data. This approach is widely used because the GPS reports and the ground truth paths are well matched as the paths are constructed by the actual driving route. However, because this approach is highly time-consuming, it is not likely to collect a large such dataset in this way.

Human labeled datasets. This method has been used in [10] [17]. The researchers start with a set of raw GPS trajectories without any prior knowledge of the actual paths. Then they find the most likely road segment for each of the GPS records in the trajectory to represent the GPS record is reported from. After assigning all of the records, a path will be created. As the path is assessed by human intelligence for each of the records, the accuracy is guaranteed at a very high level (almost 100%). Therefore, the path is considered as a ground truth path. As it is easy to collect a large set of raw GPS trajectories and the corresponding road network, this method is capable of generating a large dataset of trajectories with ground truth paths. However, simply generating the ground truth path based on the raw GPS data is always expensive and inefficient.

Synthetic datasets. Some works[10] also generate ground truth data synthetically. They pick up a path from the road network, periodically select some points on the path, and introduce some errors with normal distribution to generate the synthetic data. Afterward, the paths generated are used as the ground truth data. This is presumably the most inexpensive way to generate a dataset containing both raw GPS trajectories and their ground truth paths. However, there exist differences between the synthetic and the real world dataset, e.g., example, the driving pattern, the reports sampling interval, the GPS position error distribution, and etc. Thus it is always not suitable to use only the synthetic dataset to evaluate the performance and accuracy of a map matching algorithm.

3. DESIGN OF TRUTHFINDER

Motivated by the issues of map matching and ground truth exploration, truthFinder is proposed to interactively match the raw GPS trajectories onto a road network with the help of both traditional map matching algorithm and human intelligence to explore the ground truth data efficiently. It should be noticed that the goal of truthFinder is different from that of the traditional map matching algorithms. For traditional map matching algorithm, they are always used as the tool of calibrating a large volume of data to the road network with a high accuracy and low latency. However, because of several reasons (e.g., the complexity road network, the outlier of the trajectory data), it is impossible for the map matching algorithm to keep its result consistently with high accuracy all the time in all of the situations. So it can not be used as a tool of exploring the ground truth data from the raw GPS data. While the truthFinder is an interactive system which has human effort involved, so it is observed that the explored data can be used as ground truth. However, as human label is involved in truthFinder, it is not possible to explore a very large volume of GPS data, for example trajectory data collected in two years. However, truthFinder can be used to explore as much ground truth data as possible, for example, ground truth of enough trajectories data which can cover whole of the Shanghai, for example, data collected in two days. In summary, map matching algorithm is used to calibrate a large volume of GPS data, while truthFinder is used to explore a complete dataset for other aims, like map matching algorithm validation and parameter tuning.

3.1 Design Issues

Generally, it is a good idea to explore the ground truth data by interactively matching the GPS reports which incorporates traditional map matching algorithms and human intelligence in a unified manner. This method not only keeps the accuracy to be almost 100%, but also has a low cost for ground truth exploration. However, to explore the ground truth from the raw GPS data by human-labeling is very challenging, especially in the environment where the road network is complex. There are mainly four challenging issues:

- To find the right position from mass of candidate roads effectively by the users is a challenge. There are always many roads around each of the GPS positions, especially in the environment where the road network is very complex. It is difficult for human to find the right road where the GPS record was reported from.
- To select the intermediate roads between two roads in a complex road network is difficult. After the roads are identified for two consequent GPS reports, it still costs human much effort to recognize the path between the two roads, especially for the situations where the two positions are far away.
- To correct all of the reports onto the right roads in one time is always impossible. For example, in the complex road network, roads may overlap, which may leads the user map the report onto a right position but wrong road. Then an unreasonable path may be explored.
- To explore ground truth data from a long trajectory (thousands of GPS reports are included) is very time costly. For example, if we use the trivial method like exploring ground truth data from scratch, the cost is always linear to the record number, which is always very large, like hundreds to thousands. It will cost the users a lot of operations to add the road segments and the mediate road segments between each adjacent pair of the GPS reports.

3.2 System Overview

Motivated by the previous discussed design issues, we design truthFinder with several considerations. The architecture of our proposed interactive system is shown in Fig. 1. It composes of four major components: *Recommendation Preparation*, *Information Presentation*, *Candidates Assessing and Tuning*, and *Ground Truth Data Exploring*.

Generally, the work flow of truthFinder can be summarized as follows: First, given a sequence of raw GPS reports, the recommendation preparation component generates a sequence of road segments (potential ground truth path) using a selected map matching algorithm, like STM [10], HMM [11], etc. Second, the information presentation component visualizes the path along with the original trajectory onto the digital map. If the user accepts the accuracy of the path,

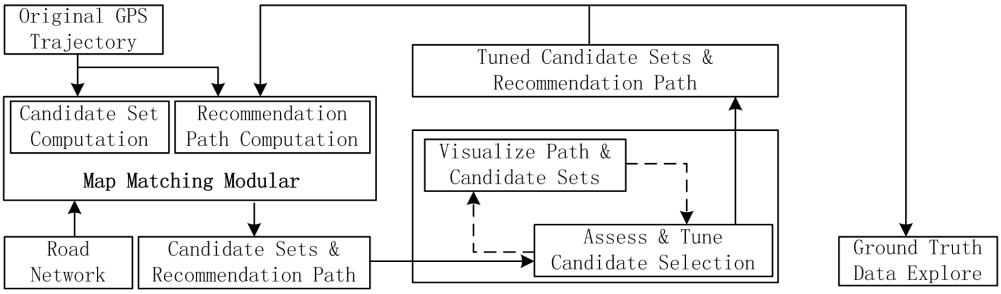

Figure 1: Overview of truthFinder system architecture

then system will jump to the last step. Third, if the accuracy is failed to achieve a high accuracy, the system asks the user to adjust the path by adding and removing a number of road segments. Fourth, based on the adjustment, the selected map matching algorithm takes the tuned path as the input and generates a new path that is supposed to be more accurate than the previous one. After a new recommended path is generated, truthFinder goes on its process from the step 2, and iteratively runs the step 2, 3 and 4 until an accurate enough path is found. Finally, truthFinder saves the path which is supposed to be accurate enough, and uses this path as ground truth for the original inputted trajectory. In the following sections, we present the details of our truthFinder one component by one component.

3.3 Recommendation Preparation

Based on the observation that human-labeled data can achieve a high accuracy, ground truth can be trivially explored by evaluating the possibility for each of the roads near the report position one by one for each of the records in the trajectory, and finding out the ground truth path by adding all of the most likely road segments to form a ground truth path. In this paper, we call this method *EPScratch*.

However, the cost of EPScratch is always linear to the record number, which is always very large, like hundreds to thousands. It will cost the users a lot of operations to add the road segments, which means it will take the users a very long time to explore the ground truth path from scratch. So how to significantly reduce the number of human operations needed to operate on the ground truth exploration is the key of the interactive map matching system. For this reason, truthFinder uses recommendation preparation component to generate a potential good path which is calculated by the traditional map matching algorithms to reduce the operations needed.

First, Recommendation Preparation component will generate a potential better path based on the original trajectory. Given a GPS trajectory T: $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_n$, truthFinder runs a traditional map matching algorithm basing on the related road network $G(V, E)$ to generate a recommendation path P: $e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_m$. As this component is iteratively called by truthFinder, both the original trajectory and the human tuned path can be the input of this component. Besides, truthFinder also retrieves the possible candidate roads $CandRoadSet_i = \{(Road_1, score_1), ..., (Road_j, score_j)\}$ for each of the GPS records r_i $(1 \leq i \leq n)$ on the trajectory to assist the user to select a better path. In the candidate road sets, each element consists of a road e_i and a score value $e_i.score$ which represents the possibility. After that, the recommendation preparation component will pack the trajectory T, the recommendation path P, and the candidate roads sets $CandRoadSet_i$ $(1 \leq i \leq n)$ together, and then sends them to the information presentation component.

3.4 Information Presentation

This component is used to visualize the information packed from the preparation component including: the trajectory T, the candidate road sets $CandRoad_i$ $(1 \leq i \leq n)$, and the path P. However, as there are always many candidate roads for each GPS position, especially in the environment where the road network is very complex, how to visualize the information from recommendation component is a big challenge. The challenge of visualization mainly comes from the mass of candidate roads, and the intermediate edges between two GPS positions. To overcome these challenges truthFinder uses the following techniques:

First, multi-layer is introduced in truthFinder. As we calculate the most likely path, the recommended path, we show them on the top layer. So that the user can assess the accuracy of the path easily. Also as there always exist wrong road segments in the recommended path, the top-level showing of the path makes user easier to find which roads should be removed from the path.

Second, multi-color notation is used for the candidate roads showing. Finding the most accurate road from the mass candidate road sets with a same color is challenged. The truthFinder system proposes a probability based color notation for the candidate road selection. The probability is calculated by the distance of the observed position to the road and the route between the consequent two candidate roads. The roads with a color representing higher probability will be more likely to be selected to add into the path.

Third, real time path showing is proposed. The user should assess their selection of the candidate roads both of the positions and the route between the two positions. And a good method to show the path between two selected candidate roads is needed. The truthFinder system designs a real time path showing technique. When a new candidate road is selected, truthFinder automatically calculates the intermediate roads to next selected position, and shows them with different colors. With our experience of using truthFinder, this makes the user convenient to select the most likely path.

The information representation component makes the ground truth data exploring from the historical GPS data much easier and more efficient with high confidence.

3.5 Assessing and Tuning

The existing map matching algorithms are always sensi-

tive to the map context and thereby the accuracy is not guaranteed for situations other than their experiment settings. To improve the accuracy and make the accuracy stable, human effort is needed for assessing whether the result is good enough to be exported as a ground truth data. When there are some of the GPS positions which can find a better candidate, the user should tune them and compose a better ground truth path. After tuning the positions, a new recommendation path will be created, which will be iteratively treated by the recommendation preparation component.

Generally, the user operations for tuning include: 1) deleting an unreasonable edge from the recommended path, 2) recognizing a better road segment and adding it into the path, 3) adding the intermediate edges between two GPS positions.

With the information preparation component, deleting the unreasonable edges from the recommended path is efficient. For example, as both the trajectory of raw GPS trajectory data and the recommended path are showed in truthFinder, users can compare them by their shape, potential route path, and other factors. The differences between the trajectory and the path can be easily found. Therefore, deleting unsuitable roads can be done by comparing between these candidate roads.

While for recognizing and adding operations, after a better candidate road is recognized, the user should add it into the recommended path, and consensually, adding the route path between two candidate roads to the recommended path. When these steps are finished, a better path would be found. However, errors may still exist in the recommend path. Users should iterate these steps until a ground truth data is found.

3.6 Ground Truth Data Exploring

After interactively visualized, assessed and tuned, a final recommended path will be generated. As the path is assessed by human effort, it is guaranteed to have a high accuracy, and reasonable to be treated as a ground truth data. Then the ground truth will be explored.

The truthFinder system explores the ground truth path information together with the original GPS data. Both the position and the road id where the position located on the path will be exported. For example, we use truthFinder to explore the ground truth path for the GPS trajectory T: $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_n$, suppose after map matching with truthFinder, a path P: $e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_m$ is found, where each position r_i $(1 \leq i \leq n)$ in T is mapped onto an edge e_{ij} at position r_{ij}, where $e_{ij} \in e_1, e_2, ..., e_m$. We not only explore the original information included in trajectory T, but also explore the longitude and latitude of r_{ij} together with the $e_{ij}.id$.

With this last phase, truthFinder generates a recommended path with high accuracy. To the best of our knowledge, it is the best method to explore the ground truth path data from the original GPS data with human intelligence involved. As the result is assessed with human intelligence, the explored dataset can be used as ground truth data. As such, it is widely used to generate ground truth dataset to evaluate map matching algorithms [10][17].

4. COST MODEL FOR TRUTHFINDER

To define the cost model, we first give some preliminaries, and then define a weighted cost model based on these preliminaries for the cost in each of the iterations. After that, we will give the cost model for the total cost of truthFinder in terms of operation per record.

Table 1: Main variables used in cost model.

Var	Description
r_i	the i_{th} position in the trajectory, $1 \leq i \leq n$
S_j	error positions set in the j_{th} iteration
N_j	number of error positions in set S_j
E_t	threshold of error positions in ground truth path
A_m	accuracy of the map matching algorithm
A_h	human ability to correct error position
w_d	cost of deleting an edge from the path
w_a	cost of adding a road segment into the path

4.1 Preliminaries

Viewing from the process of truthFinder, the phases involved include preparing a recommended path, visualizing the recommend data and interactively tuning them with human assessing, and finally exploring the ground truth. In these phases, human effort is mainly involved in the phase of tuning and assessing the path with several types of operations, like deleting an edge, adding a better road segment, etc. As different operations may have different cost (for example, deleting an edge from the recommended path always costs less than the operation of recognizing a better candidate road and adding it into the path), the cost of human aid is calculated from this phase in term of weighted operations.

To discuss our cost model conveniently, we give several definitions of the variables used in the cost model. Table 1 summarizes the main variables. Adding a road into a path needs to find and assess the suitable ones from a bundle of candidate roads. While deleting a road from a path can be done directly. Generally, w_a is always much bigger than w_d.

4.2 Cost Model

The cost model for truthFinder is composed by each of the iterations involved in the assessing and tuning phase. First, we give the cost of truthFinder in one of the iterations. Let's consider the cost of the j_{th} iteration. As we notated in Table 1, there are N_j error positions in the recommended path, and the positions are r_{jk}, where $r_{jk} \in S_j$. For each error position, user has to tune it to a better road by deleting it from the recommended path, and adding a better position, and the intermediate edges. So the cost for the error position r_{jk} in the j_{th} iteration is

$$C_{jk} = N_{jkd} * w_{jkd} + N_{jka} * w_{jka} \quad (1)$$

for each $1 \leq k \leq N_j$. So the total cost for the j_{th} iteration goes to

$$C_j = \sum_{k=1}^{N_j} (C_{jk}) \quad (2)$$

In this model for the j_{th} iteration, the cost is directly impacted by the number of operations and the corresponding cost weight. As such, to minimize the cost of adding and deleting operations, we should keep the weight of the operation cost at a low level. Empirically, the value of deleting a road from the ground truth candidate path is always stable. So the way to reduce the cost would be minimizing the weight of adding a road segment as low as possible.

Figure 2: Road network of Shanghai, China.

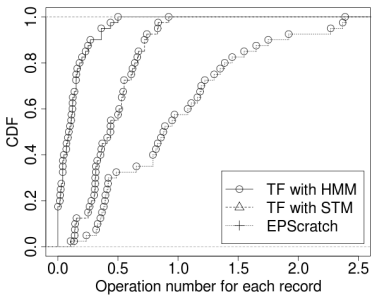

Figure 3: CDF of cost.

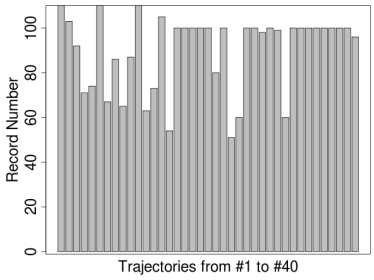

Figure 4: Record numbers of the trajectories.

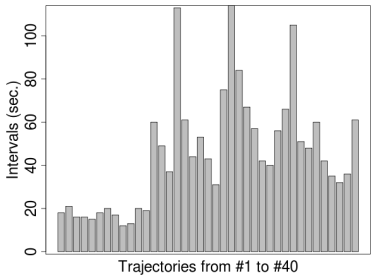

Figure 5: Average sampling intervals of the trajectories.

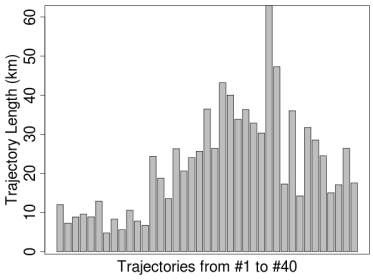

Figure 6: Lengths of the trajectories.

The weight of adding a road segment into the path, w_a, is always affected by the map context. In a complex road network, where roads are very dense, it always costs more to find a better road to be added into the ground truth path. So this operation of adding will cost more than that of adding a road into the truth path in a situation where only few roads are included in the spare region. As shown by the cost model, reducing the value of w_a makes the cost of ground truth exploration less, it is necessary to reduce the cost of adding a road into the ground truth path. For this reason, truthFinder selects the top k candidate roads for each report with the highest possibilities. This makes the adding a road by selecting it from the candidate road set at a low cost. However, one situation should be considered, that if the road segment in the ground truth path is not included in the selected top k candidate roads, then the user has to find a road from the original map, and add it into the path. This will make the cost very high as a penalty, which means the weight of the adding operation w_a very high.

As exploring a ground truth path from a given trajectory is done by iteratively assessing and tuning the recommended path, the total cost for truthFinder to explore the path should be iteratively added. In every iteration, the cost is calculated by Eq. (2), so the total cost for truthFinder is to add up the cost in every iteration. As we supposed, the accuracies of the algorithm and human are A_m and A_h, truthFinder has to iterative the assessing and tuning phase I iterations, where

$$I \leq \frac{\log(\frac{E_t}{1-A_m})}{log(1-A_h)} \tag{3}$$

Then, in average, for each of the reports, the total cost of exploring the ground truth data becomes the total cost of each iteration divided by the number of reports, which is

$$\bar{C} = \sum_{j=1}^{I} \sum_{k=1}^{N_j} (N_{jkd} * w_{jkd} + N_{jka} * w_{jka})/n \tag{4}$$

As the model shows, the average cost per report depends on the weighted cost in each iteration of the ground truth exploration process which we have discussed the previous section, and the iteration numbers. Relatively, the iteration number is decided by the accuracy of map matching algorithms and the ability of the user to correct the wrong selected roads. However, it is difficult to define a metric and fairly evaluate the ability. But empirically, the user can always reduce the iteration number at a very small value (like two to three iterations) for most of the case we did in the evaluation. Actually, it is an interesting and important work, we will study it in future work.

5. EVALUATION

The objective of the experiments are to evaluate the cost of truthFinder under our defined cost model, and find out how the map matching algorithms impact the cost of truthFinder. In this section, we present representative results for truthFinder. We first state the description of the experimental settings. After that, we give the results of our experiments including (i) the cost comparing to EPScratch, and (ii)the impact of map matching algorithms.

5.1 Experimental Settings

We present the experimental settings in this section, including the dataset we used and the road network of the city where we collected our dataset. The truthFinder system is deployed on an IBM server which has 4G memory and a

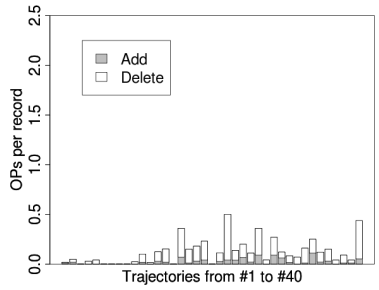

Figure 7: Cost of truthFinder using HMM.

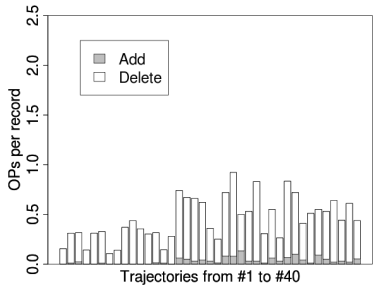

Figure 8: Cost of truthFinder using STM.

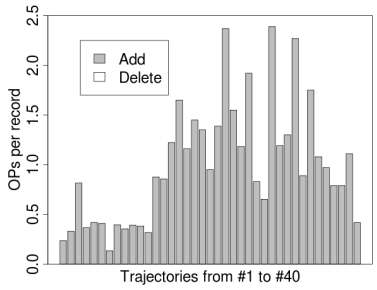

Figure 9: Cost of EPScratch.

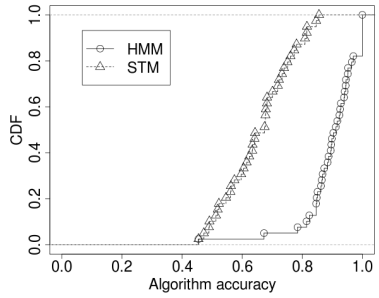

Figure 10: CDF of accuracy of the map matching algorithms.

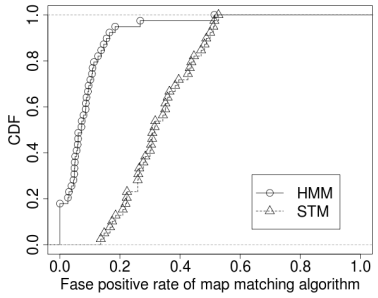

Figure 11: CDF of the false positive rate.

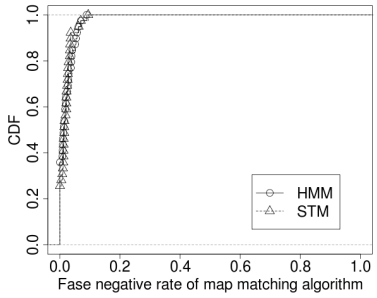

Figure 12: CDF of the false negative rate.

CPU of Intel Xeon with 8 processors (E5405 @ 2.00 GHz).

Road Network Since 2005, we have collected a large volume of GPS reports from Shanghai Grid. As our trajectory dataset was collected from Shanghai, we use the road network of Shanghai as shown in Fig. 2 in our evaluation experiments. There are 22180 vertexes and 65510 directed roads in this network. So for the road network $G(V, E)$ of Shanghai, $\#V = 22180$ and $\#E = 65510$.

Trajectory Dataset To explore the ground truth with truthFinder, we randomly selected a dataset with 40 trajectories in our experiments for comparing truthFinder to the method of EPScratch. The characteristics of the trajectories in the dataset are presented in the figures. Fig. 4 shows the record number for each trajectory in dataset. As it shows, the number varies from 50 to 120. We calculate trajectory length by adding the direct distance between every two adjacent reports. The distance ranges are showed in Fig. 6 (about 10km to about 60km). The sampling intervals of the reports in each trajectory are shown in Fig. 5.

5.2 Cost of truthFinder

In this section, we present the experiment result on the dataset to compare operation cost of truthFinder and EPScratch. We have done experiments on both the HMM [11] and STM [10] map matching algorithms in truthFinder for the recommendation path generating. We compare them in term of operation numbers, where operations of adding road and deleting road are separated.

As demonstrated by Fig. 7, truthFinder keeps the operation cost very low, which is about 0.2 operations for each record using HMM algorithm. While for the situation where truthFinder uses the STM algorithm, the cost is a little high, which come to 0.5 for every record (Fig. 8). The reason is that the accuracy of STM algorithm is a little lower than HMM algorithm in our dataset and map context. When the map matching algorithm has a high accuracy, truthFinder will significantly reduce the cost of human operations for matching the raw GPS trajectories to its ground truth path. Meanwhile from Fig. 3, we find that, about 98% of the ground truth path of the trajectories can be explored within 0.5 human operations for each record, and about 80% of them can be done within 0.2 operations.

While the method of EPScratch costs much higher than truthFinder, especially the adding roads operations. The cost for finding the ground truth for most of the trajectories are always very high, compared to our truthFinder based method. It is almost two orders of magnitude of that of truthFinder based on HMM map matching algorithm. The reason is that, not only the roads where the records are reported from should be added into the found-out ground truth path, but also the intermediate roads should be added. As discussed in Section 4.1, w_a is much larger than w_d. The total weight cost of EPScratch will be very large. The truthFinder system reduces the cost of map matching process up to two orders of magnitude less than the EPScratch approach. With this comparison, we are confirmed that, truthFinder will reduce the total cost significantly and can explore more trajectories than EPScratch.

5.3 Impact of Map Matching Algorithms

Next, we concern the impact of the traditional map matching algorithms for truthFinder. We first present the accuracy of the map matching algorithms using our dataset and the explored ground truth path. Then we present the impact

of the map matching algorithms.

We have implemented two map matching algorithms, including STM [10], and HMM [11]. We use each of the trajectories in the dataset as the input of each map matching algorithm, which will generate a path (a sequence of roads), as its output. Then we calculate their accuracy. We measure the accuracy of the map matching algorithms with the metric defined in Eq. (5), where $Set_i.ground$ is the set of road IDs in ground truth path, and $Set_i.alg$ is the set of road IDs in algorithm calculated path.

$$A_i = \frac{\#of(Set_i.ground \cap Set_i.alg)}{\#of(Set_i.ground \cup Set_i.alg)} \quad (5)$$

Fig. 10 shows the accuracy of different map matching algorithms in our road network with our dataset. From the result, we can find that, in our experiment environment, the accuracies of HMM algorithm, most of which are between 80% and 93%, are always higher than that of STM algorithm, whose values changes frequently and always are lower than 80%. Together with the results in the previous experiments, we are confirmed that the higher accuracy the map matching algorithm has, the less operations the truthFinder costs.

As demonstrated by Eq. (4), the cost of truthFinder depends on the number of adding and deleting a road operations, as well as the weight of these operations. The operation of adding a road is always caused by the situation that a correct road in the ground truth path not included in the recommended path (false negative), while the operation of deleting a road is caused by the reason that wrong roads are included in the recommended path (false positive). So we calculated the false negative (Fig. 12) and false positive (Fig. 11) rate for both of these two map matching algorithms. From the results, we can find that, for the false negative rates, HMM and STM algorithms share similar trends, so relatively, as demonstrated in figures of the system cost, the adding operations of truthFinder are also similar. However, the false positive rates of HMM are always less than 0.2 which is less than that of STM algorithm whose rate changes frequently ranging from 0.15 to 1. Together with results of cost, we can find that the higher the false positive rate is, the more the deleting are needed. Similarly, the higher the false negative rate is, the more the adding are needed.

6. CONCLUSION

In this paper, we propose truthFinder, a system of interactive map matching the collected raw GPS trajectory data. The truthFinder system characterizes itself with several unique features. It employs human intelligence to aid the map matching algorithms to explore ground truth data from raw GPS data. To measure the cost, we define a cost model and evaluate our prototype system with this model. The result shows that truthFinder significantly reduces the cost. The truthFinder system would be an efficient way to solve the validation issue map matching problem.

7. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments and kindly suggestions. This work is supported by the National Natural Science Foundation of China under Grant No. 60736013, Grant No. 61120106005, Grant No. 61025009 and Grant No. 60903040.

8. REFERENCES

[1] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proceedings of the VLDB'05*, pages 853–864. VLDB, 2005.

[2] Z. Chen, H. Shen, and X. Zhou. Discovering popular routes from trajectories. In *Proceedings of the ICDE'11*, pages 900–911. IEEE, 2011.

[3] Z. Chen, H. Shen, X. Zhou, Y. Zheng, and X. Xie. Searching trajectories by locations: An efficiency study. In *Proceedings of the SIGMOD'10*, pages 255–266. ACM, 2010.

[4] P. Cudre-Mauroux, E. Wu, and S. Madden. Trajstore: An adaptive storage system for very large trajectory data sets. In *Proceedings of the ICDE'10*, pages 109–120. IEEE, 2010.

[5] H. Gonzalez, J. Han, X. Li, M. Myslinska, and etc. Adaptive fastest path computation on a road network: A traffic mining approach. In *Proceedings of the VLDB'07*, pages 794–805. VLDB, 2007.

[6] J. Greenfeld. Matching gps observations to locations on a digital map. In *Proceedings of the 81th Annual Meeting of the Transportation Research Board (TRB'02)*, 2002.

[7] M. Li, M. Wu, Y. Li, J. Cao, L. Huang, Q. Deng, X. Lin, C. Jiang, W. Tong, Y. Gui, et al. Shanghaigrid: an information service grid. *Concurrency and Computation: Practice and Experience*, 18(1):111–135, 2006.

[8] Z. Li, M. Ji, J. Lee, and etc. Movemine: mining moving object databases. In *Proceedings of the SIGMOD'10*, pages 1203–1206. ACM, 2010.

[9] S. Liu, Y. Liu, L. Ni, J. Fan, and M. Li. Towards mobility-based clustering. In *Proceedings of the SIGKDD'10*, pages 919–928. ACM, 2010.

[10] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the GIS'09*, pages 352–361. ACM, 2009.

[11] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the GIS'09*, pages 336–343. ACM, 2009.

[12] W. Ochieng, M. Quddus, and R. Noland. Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 2(55), 2009.

[13] M. Quddus, W. Ochieng, and R. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

[14] M. Quddus, W. Ochieng, L. Zhao, and etc. A general map matching algorithm for transport telematics applications. *GPS solutions*, 7(3):157–167, 2003.

[15] K. Tufte, J. Li, D. Maier, and etc. Travel time estimation using niagarast and latte. In *Proceedings of the SIGMOD'07*, pages 1091–1093. ACM, 2007.

[16] M. Xie, L. Lakshmanan, and P. Wood. Comprec-trip: A composite recommendation system for travel planning. In *Proceedings of the ICDE'11*, pages 1352–1355. IEEE, 2011.

[17] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun. An interactive-voting based map matching algorithm. In *Proceedings of the MDM'10*, pages 43–52. IEEE, 2010.