Route Discovery from Mining Uncertain Trajectories

Hechen Liu[1], Ling-Yin Wei[2], Yu Zheng[3], Markus Schneider[1], Wen-Chih Peng[2]
[1] Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA
[2] National Chiao Tung University, Hsinchu, Taiwan
[3] Microsoft Research Asia, Beijing, China
{heliu, mschneid}@cise.ufl.edu, {lywei.cs95g, wcpeng}@nctu.edu.tw, yuzheng@microsoft.com

***Abstract*—Moving objects in the physical world usually generate many uncertain trajectories for some reasons such as the consideration of energy consumption, leaving the route passing two consecutive sampling points unknown. While such trajectories imply rich knowledge about the mobility of moving objects, they are less useful individually. This paper introduces an online trip planning system that mines collective knowledge (i.e., most possible routes between given locations) from massive uncertain trajectories following a paradigm of "*uncertain+uncertain→certain*". This system first builds a routable graph from uncertain trajectories, and then answers a user's online query (a sequence of point locations) by searching top-*k* routes on the graph. Two large-scale datasets consisting of "check-in" records from FourSquare and a trajectory dataset of taxis have been used to evaluate our system. As a result, our system provides a user with effective routes according to the user's query efficiently.**

Keywords-spatial trajectories, uncertain trajectories, moving objects, trip planning

I. INTRODUCTION

With the advances in location-acquisition technology (e.g., GPS services), the study of moving objects has been experiencing popularity [3], [13], [12]. A moving object such as a vehicle or a person can be represented by a sequence of locations with increment on time, or a *trajectory*. For example, a sequence of places of interest (POI) a traveler visits, the migration of birds, and the movement of hurricanes can all be represented by trajectories. Obtaining these trajectories will be useful in discovering knowledge from moving objects. However, trajectories are often generated at a low frequency due to the consideration of energy saving or other application features. For example, a traveler with a smart phone cannot take a geo-tagged photo every 10 seconds; a sensor tracking a hurricane cannot report the location every second. Assume that a traveler in New York City has visited 5 POIs including the Statue of Liberty, China Town, Times Square, Central Park, and the Metropolitan Museum of Art, but only taken photos at the Statue of Liberty and Central Park. The real path of this traveler is *uncertain*. However, another person who has also visited the same POIs may have photos at Times Square and the Metropolitan Museum of Art. Combining their uncertain trajectories, we can infer their real paths, i.e. "*uncertain + uncertain → certain*". The goal of this paper is to introduce an online system that enables route discovering through mining a large number of uncertain trajectories.

We collect the datasets of uncertainty trajectories from two sources, travelers' "check-ins" and taxi trajectories. In recent years, emerging social network websites with photo sharing and check-in functions make the trajectories of travelers available: a person with a smart phone can create a travel tip or capture a geo-tagged photo at any time and upload it to a social network such as Flickr or FourSquare. We have collected more than 425,000 check-ins of three months in New York City, and detected over 73,000 uncertain trajectories. We have also collected over 15,000 taxi trajectories in Beijing with the help of the GPS sensors embedded in taxis. A routable graph on top of the city area is built where the trip planning is performed. When a user inputs a sequence of query locations, the system will search all possible routes traversing them on the graph. The system will score these routes and report top-*k* optimal routes.

From the authors' knowledge, little existing work has been done to achieve our goal. The uncertainty of trajectories in a free 2D space has been studied intensively, as shown in [9], [6], [8] , [4] and [7]. Given the maximum speed of a moving object, the above approaches will show an uncertain region enclosing all possible locations while specific routes are not inferred. The approaches of mining GPS trajectories to provide route recommendation are discussed in [14], [10]. [10] infers fastest routes from historical trajectories on the road network with high sampling rates. [1] mines travelers' frequent trip patterns. [5] provides trip plans by mining from geo-tagged photos, where paths are detected by merging route fragments, however, part of a real route might be lost. [11] solves the uncertainty in a road network environment.

The contributions of this paper are,

- We design and implement a system to enable trip planning from mining uncertain trajectories.
- We apply our approach on check-in sequences of travelers as well as taxi trajectories. It shows that the system is effective in different applications.

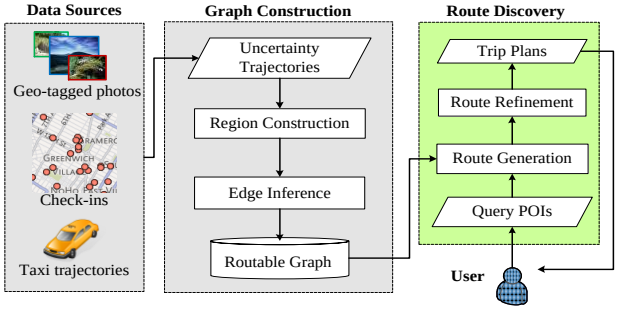
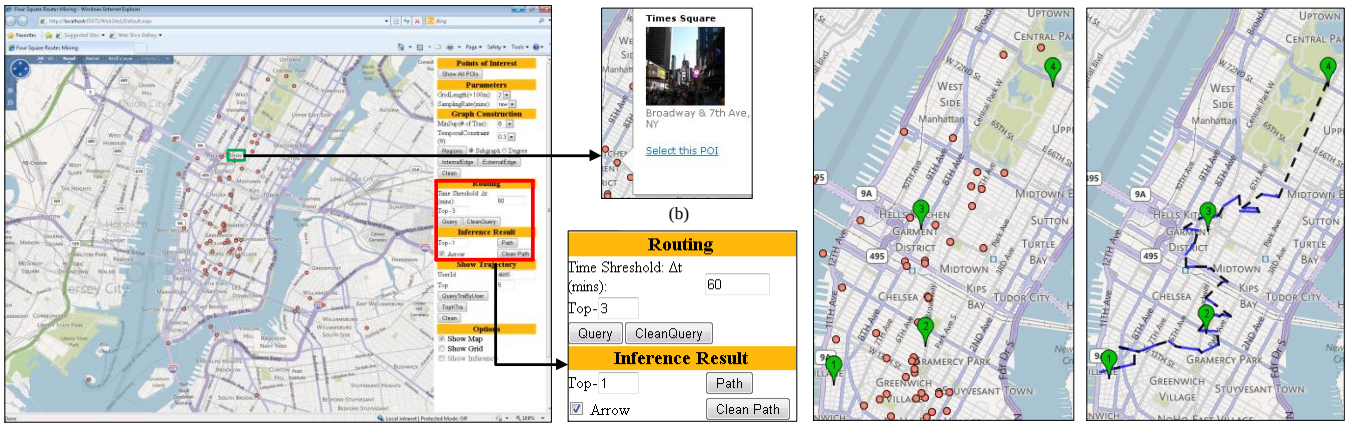


Figure 1. Overview of the system framework.



(a) (b) (c) (d) (e)

Figure 2. User interface of the trip planning system.

A. System Framework

The framework of the system is shown in Figure 1. The system consists of two components: graph construction and trip planning. The graph construction part is an offline module. Having a large dataset of raw geo-tagged photos and travelers' tips as input, we first group them by users and get a set of uncertainty trajectories. Then we partition the city's space into a set of disjoint grids, and index the trajectories. A set of connected grid cells will form a connected region. Edges connecting cells will be inferred. Each edge carries some important information, such as the moving direction, its support (number of trajectories traversed this edge) and the transition time. The regions and edges form a routable graph that will be stored. The second stage is performed online. When a user inputs a set of POIs and a time span, the system will find top-*k* rough routes on the basis of the routable graph. A rough route containing a list of grids will be first generated. In the end we refine the routes and generate the trip plan. We will show the detail of the algorithms in these two stages in Section II.

B. Demonstration of the System

The trip planning system is built as a website. Users could submit a query online and get the response quickly in less than one second. The user interface of the system is shown in Figure 2a. Here we apply the trip planning system to the area of the New York City.

The right side of the website shows the menu of the system (Figure 2c). Before submitting queries, parameters should be set properly. The meaning of these parameters will be introduced in Section II. A user could interact with the system by performing actions on the map. The red dots on the map represent top-100 points of interest (we have much more POIs in the database). When the user moves the mouse on a POI, the details will be displayed, including its name, address and a photo. The traveler can choose this POI as one of his/her destinations by click "Select this POI" (Figure 2b). A user can also arbitrarily choose a non-POI location on the map by right-clicking the position.

A user can select up to four POIs (Figure 2d), and click the "Query" button under "Routing" (Figure 2c). Within a second, the system will alert the traveler whether top-*k* routes are found.

The user can view each of the top-*k* paths. The top-1 path of the query is shown in Figure 2e. The blue segments show the real trajectory segments from the raw data. The black dash lines are the inferred lines from the routing algorithm.

II. Design and Implementation

A. Mining Uncertainty Trajectories

In this subsection, we introduce our *mining uncertainty trajectories* approach. TABLE I. shows the notations of the parameters that will be needed in the model, as we have mentioned in Section II.

TABLE I. Notations of Parameters

Notations	Description
gl	Grid length
sr	Sample rage: sampling from raw trajectories
θ	A temporal constraint in [0, 1]
C	Connection support: number of trajectories traversing a pair of spatial-close grids
Δt	Transition time of a sub-trajectory
k	The ranking of optimal routes

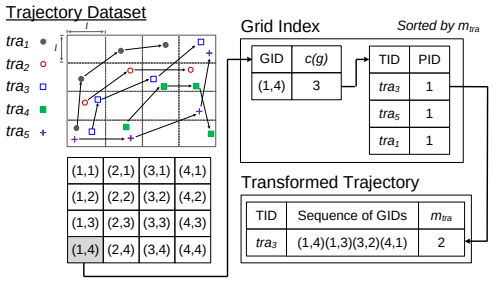


Figure 3. An example of the index structure.

We first divide a geographical area into a set of disjoint grid cells. A cell is a square and is denoted by (i, j) indicating the cell id. A trajectory is indexed by the order of the grid cells it traverses in Figure 3. $c(g)$ denotes the number of distinct trajectories traversing grid cell g. The trajectories in a grid cell are ordered by a variable m_{tra} in a descending order, where m_{tra} denotes the median of $c(g)$ of all cells traversed by a trajectory *tra*.

We define that two cells are *spatial-close* if their distance is less than one cell. Thus, a cell is spatial-close to 8 cells surrounding it. We say that two sub-trajectories (segments of trajectories) are *correlated*, if the following conditions hold: 1) the ratio of the difference of the transition times to the maximum transition time of the two sub-trajectories is less than a threshold θ, i.e. $\frac{|\Delta t_1 - \Delta t_2|}{\max\{\Delta t_1, \Delta t_2\}} \leq \theta$; 2) either the source grid cell of two sub-trajectories are spatial-close and they have the same sink grid cell (*Rule* 1), or they have the same source grid cell and their sink grid cells are spatial-close (*Rule* 2) (Figure 4). The value of θ shows the similarity between two sub-trajectories. The closer θ is to 0, the more similar two sub-trajectories are.

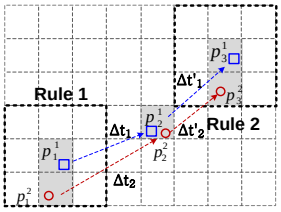


Figure 4. Correlated trajectories.

We define that two grid cells g and g' are *neighbors*, or *gNg'*, if 1) g and g' are spatial-close; 2) the connection support C of g and g' is greater than or equal to the given connection support C_0 specified by the user. A set of grids G is called a *region*, if for any grid cell $g \in G$, we can always find $g' \in G$, so that *gNg'* is satisfied.

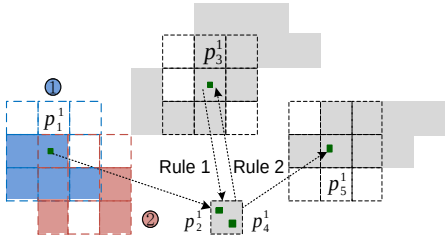


Figure 5. Region construction processes.

Then, we can obtain a set of connected regions (Figure 5). Edges carrying information such as the direction, the connection support and the transition time will be added. Internal edges will be added within a region, and external edges will be added between different regions (Figure 6a). If there are multiple edges between same regions, edges with connection support of 0 or longer transition times will be removed (Figure 6b). Then the routable graph is built.

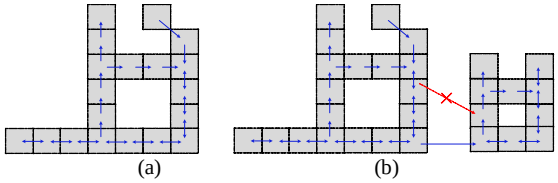


Figure 6. Edge inference (a) and remove redundant edges (b).

The last step is route generation. Given a sequence of query locations and a time span, we search qualified routes on the routable graph. We first map all query points to grid cells on the graph. If a query point is not in any grid, we map it to grids that are closest to it. It is possible that a query point is mapped to more than one grid cell (Figure 7a). We define a route score function which considers routes with higher connection support as "better routes". We find top-k local routes based on an A*-like algorithm between any two consecutive grids. Then we search top-k global routes by a branch-and-bound search approach (Figure 7b). When searching local routes between two grid cells which are located in different regions, we propose a "two-layer routing" algorithm. We determine the order of the regions to reduce the search space. By utilizing a lower bound of transition times between any two regions, we generate region sequences with respect to the given grid cells, and search by sequentially traversing these regions. In the end, we refine top-k routes by finding the segments from historical trajectories. We adopt linear regression on the point set in each grid to derive a segment, and concatenate the segments (Figure 7c).

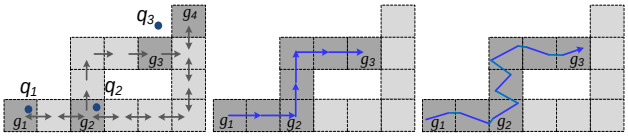


Figure 7. Query transformation (a); routing (b); route refinement(c).

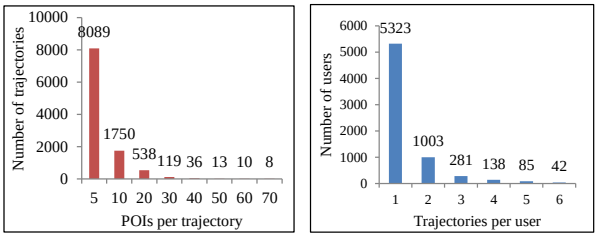
B. *Implementation*

To generate trajectories of travelers, we mine all users' tips data (a tip is a short message describing the user's experience at a POI, like "I am at the Apple store and it is so crowded!") in New York City from foursquare.com between May. 2008 and Jun. 2011. A trajectory is detected by a sequence of check-ins per day per user. The user should check in at least 3 times per day in order to form a trajectory. Thus, users who only check in randomly are considered as "noise" and are removed. We also collect taxi trajectories in Beijing with the help of GPS sensors embedded in each taxi. The raw data we collect are summarized in TABLE II.

TABLE II. SUMMARY OF DATASETS

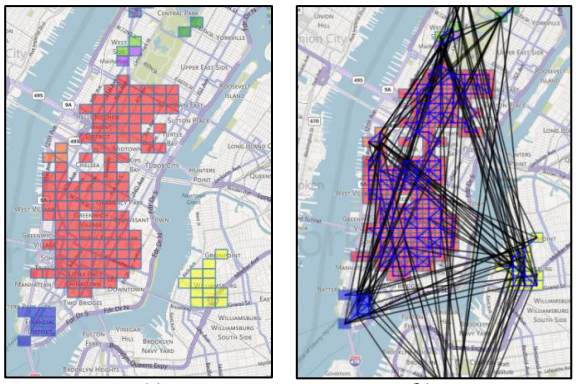
Data	FourSquare Data	Taxi Data
POIs	206,194	-
Users	49,023	3,531
Check-in sequences	425,558	2,989,165
Trajectories	73,088	15,098
Contributed users	10,337	3,531

A trajectory with more numbers of POIs potentially has more knowledge than a trajectory with less number of POIs. Thus the trajectories with more POIs are considered to have "good quality". Among all the uncertainty trajectories we detected, most of them (8089) have less than 5 POIs, 1750 trajectories contains 6 to 10 POIs (Figure 8a). A large number of travelers (5323) contribute only one trajectory (Figure 8b). Therefore, most of the data comes from a large number of different users, which reflects the real world. With the uncertain trajectory dataset, we build the routable graph (Figure 9a), with grid length 200 meters, θ=0.3, and Δt=1 hour on raw trajectories. Different colors represent different regions. The internal edges and external edges are shown in Figure 9b.



(a) (b)

Figure 8. Trajectory quality (a) and users' contributions (b).



(a) (b)

Figure 9. Routable graph construction on the area of NYC

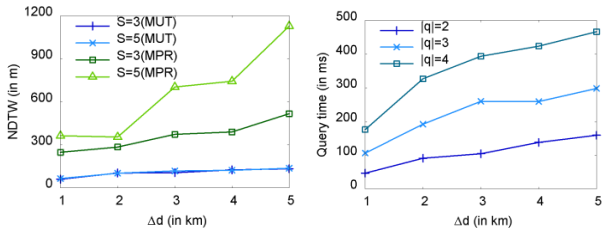
III. Experimental Evaluation

To measure the effect and efficiency of our approach, we apply it to large real datasets and study the performance. We choose the taxi trajectories in Beijing because: 1) A taxi trajectory contains a large number of points, thus we can set a raw taxi trajectory as a ground truth and sampled trajectories as uncertain trajectories; 2) taking taxis is also an important way when a traveler makes a trip to a new city. We choose the dataset which contains 15,098 raw trajectories from 3531 users. We first find the ground truth. Given 2 to 4 query locations, we select raw trajectories that have traverses these query locations and rank them. Trajectories which traverse more segments that are traversed more frequently will receive higher ranking. We choose the top-1 raw trajectory as the ground truth. We introduce a measurement called length-normalized dynamic time warping distance (NDTW) between two trajectories, which is modified from dynamic time warping distance (DTW),

$$NDTW(tra_1, tra_2) = \frac{DTW(tra_1, tra_2)}{length(tra_1)}$$

Obviously, lower NDTW indicates more precisely inferred routes. We compare the NDTW of our *mining uncertainty trajectories* approach (denoted by MUT) with the MPR approach in [2] on the inferred routes with sampling rates of 3 and 5 minutes respectively. The distance between two query points are determined by Δt, i.e., the transition time between the two query points. The experiment result (Figure 10a) shows that our approach will find more precise routes.

We also evaluate the query time (Figure 10b) when the numbers of query points are 2, 3 and 4 respectively.



(a) (b)

Figure 10. Effect (a) and effeciency (b) on route generation

IV. Conclustions and Future Work

In this paper, we introduce an online system for route discovering and trip planning from mining uncertainty trajectories. Users can input a number of points of interest (POI) and a time span, and the system will find top-*k* optimal routes traversing the specified POIs for this user. We mine from more than 73,000 travelers' trajectories in New York City and over 15,000 taxi trajectories in Beijing. We build a routable graph from the trajectories and search the routes on top of the routable graph. The system can answer users' queries effectively and efficiently. In the future, we plan to apply our system to more cities all over the world.

References

[1] Y. Arase, X. Xie, T. Hara, and S. Nishio. Mining people's trips from large scale geo-tagged photos. In *ACM MM*, pages 133-142, 2010.

[2] Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *ICDE*, pages 900-911, 2011.

[3] R. H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann Publishers, 2005.

[4] B. Kuijpers, B. Moelans, W. Othman, and A. Vaisman. Analyzing trajectories using uncertainty and background information. In *SSTD*, pages 135-152, 2009.

[5] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *ACM MM*, pages 143-152, 2010.

[6] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving object representations. In *SSD*, 1999

[7] R. Praing and M. Schneider. Modeling historical and future movements of spatio-temporal objects in moving objects databases. In *CIKM*, pages 183-192, 2007

[8] G. Trajcevski, A. Choudhary, O. Wolfson, L. Ye, and G. Li. Uncertain range queries for necklaces. In *MDM*, pages 199-208, 2010.

[9] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing Uncertainty in Moving Objects Databases. *ACM Trans. on Database Systems (TODS)*, 29:463-507, 2004

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

[11] K. Zheng, Y. Zheng, X. Xie, X. Zhou. Reducing Uncertainty of Low-Sampling-Rate Trajectories. In *ICDE* 2012 (Accepted).

[12] Y. Zheng, Y. Chen, X. Xie, W.-Y. Ma. GeoLife2.0: A Location-Based Social Networking Service. In *MDM* 2009.

[13] Y. Zheng, X. Xie, W.-Y. Ma, GeoLife: A Collaborative Social Networking Service among User, location and trajectory. In *IEEE Data Engineering Bulletin*. 33, 2, 2010, pp. 32-40.

[14] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791-800, 2009