Finding Frequently Visited Paths: Dealing with the Uncertainty of Spatio-temporal Mobility Data

Mitra Baratchi , Nirvana Meratnia , Paul J.M. Havinga

Department of Computer Science, University of Twente
The Netherlands

{m.baratchi ,n.meratnia, p.j.m.havinga}@utwente.nl

Abstract—**With the ever-increasing advancements in sensor technology and localization systems, large amounts of spatio-temporal data can be collected from moving objects equipped with wireless sensor nodes. Analysis of such data provides the opportunity of extracting useful information about movement behaviour and interaction between moving objects. Inherent characteristics of wireless sensor nodes cause the data collected by them to have low or irregular frequency and often be erroneous. Existence of different levels of uncertainty in these data makes the procedure of finding movement patterns difficult and ambiguous. In this paper, we propose a hierarchical approach to find the frequently visited paths using location data of people carrying a custom designed mobile wireless sensor node. We hierarchically cluster trajectories and find their resemblance at the finest level while dealing with the uncertainties. The performance evaluation results show that compared with previous schemes, our method performs better in presence of ambiguity and sources of data uncertainty.**

I. INTRODUCTION

Integration of location acquisition technologies, such as GPS with data communication networks, such as wireless sensor networks has led to possibility of collecting a huge volume of time-stamped location data form humans, animals, and vehicles. Analysis of such spatio-temporal data can answer various questions about mobility patterns of individuals and groups, such as periodic patterns, swarm patterns, and movement interactions [1]. One of such questions is finding frequently visited paths traversed by individuals. Being able to answer this question is of importance for many real world applications such as urban planning [2], crowd sourcing, trip planning for tourists or cars [3], advertising (e.g. planning billboards), and animal behaviour analysis (effects of roads on animals) [4].

One approach towards finding frequently visited paths, is to compare trajectories as a whole and find groups (clusters) with the larger number of similar trajectories. This general comparison misses similar partitions shared by trajectories which generally belong to different groups. Therefore, effective partitioning methods should be used to partition trajectories before grouping them into similar clusters. This partitioning can be performed by the use of semantic information inferred from the regions of interest (e.g. stopping points).

On the other hand, when trajectories are partitioned and grouped based on the regions of interest the sub-trajectories in each group are already similar with respect to their origin and destination and potentially have common sub-paths. Therefore, finding their difference is difficult. Additionally, different sources of uncertainty make this procedure even more challenging. This is shown in Figure 1.

Fig. 1.Figures in left and middle show two possible paths between A and B. Figure in right shows three different trajectories between point A and B resulted by interpolation. Black and red trajectories, each follow one of these paths and the blue trajectory is not even classifiable by human eye.

When collecting spatio-temporal data by wireless sensor nodes, uncertainties [5] are introduced due to technical shortcomings . We categorise these uncertainties into (i) noise (a measurement which does not make sense considering the maximum speed and the distance from previous measurement), (ii) measurement error (minor deviation from the correct value), (iii) discrete sampling, and (iv) missing samples. Discrete nature of sampling is the source of uncertainty between two consequent samples. The level of uncertainty is affected by the frequency with which position samples are taken [5]. In real world applications there are also samples missing. This implies that data is unavailable because of hardware failure, and transmission error, among other reasons. Missing samples increase the level of uncertainty even more.

Representing trajectories from samples in presence of uncertainties is a challenge. One of the most popular methods in presenting trajectories is interpolating the two consecutive GPS measurements. Depending on the sampling rate, number of missing points, and their location, two similar trajectories may be presented completely differently, using interpolation (or two different trajectories may be presented similarly). Another alternative for trajectory representation is to use road network data. Other than for vehicles, which have road restricted movements, this approach is not very useful (for example for animals and pedestrians). Trajectory representation while dealing with the uncertainties can also be performed through correcting trajectories by the collective knowledge [6]. The collective knowledge in this case is the knowledge gained by considering mobility data of all trajectories on a specific path. As seen in Figure 2, the red points can better represent the frequently visited paths than the blue interpolating lines.

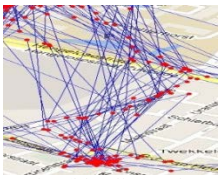

Fig. 2. Individual trajectories (blue lines) alone do not provide enough information to construct the route while their aggregation (red dots) can help reconstruct the route more accurately.

Our contributions in this paper are as follows: Firstly, we propose a hierarchical grid based clustering approach based on the semantic and geographical data to find the frequently visited paths by people to the finest level of similarity. To the best of our knowledge this is the first time such hierarchical approach is used to find frequently visited paths. Secondly, we use the concept of collective knowledge to deal with the uncertainty of trajectory representation when the level of uncertainty caused by missing samples and discrete sampling is increased.

II. Related work

Previous researches on analysing trajectories mainly apply data mining methods such as clustering on the collected mobility data. One of the first methods in trajectory clustering was proposed in [7], which suggested to use a probabilistic model-based approach. Trajectories have also been clustered using similarity measures [8, 9]. Some popular similarity measures which have been used to compare trajectories are Euclidian distance [10], LCSS (Least Common Subsequence) [11], DTW (Dynamic Time Warping) [12], ERP (Edit distance with Real Penalty) [13], EDR (Edit Distance on Real sequences) [9], and CATS (Clue Aware Trajectory Similarity) [14]. Without getting help from a complementary method such as sliding window, these similarity measures can only find the similarity between complete trajectories and will not give any information about the common sub-trajectories. To be able to find similar fractions of trajectories, Traculus [8] focuses on finding the common portions of trajectories (sub-trajectories) by first partitioning them based on the movement behavior and then clustering these trajectory partitions. Another view to finding frequent trajectories is by considering the semantic information such as regions of interest [15, 16] and finding frequent patterns in semantically defined trajectories.

While considerable attention has been so far paid to finding similar trajectories considering the entire trajectory, not much attention has been paid to finding (dis)similar sub-trajectories between regions of interest (e.g. between semantic areas). The semantically similar sub-trajectories may be composed of different paths. The uncertainty in such data can also cause the procedure of finding (dis)similarity between paths harder.

Recently, few methods have been proposed to deal with different notions of uncertainty in collected GPS data. In order to deal with the uncertainty caused by measurement error a constant uncertain area around the trajectory points (cylindrical or square) is considered [17, 18]. In another approach this problem was addressed by proposing a variant of fuzzy C-Means clustering algorithm [19].

Having daily trajectories of a person, we do not aim to find groups of complete trajectories which are similar, but instead we are interested to know what are the frequently used path between semantic places and *where* most of these trajectories are similar (e.g. what is the more visited road?). In terms of finding partitions which are shared by different trajectories, our work is more in-line with Traculus [8]. While Traculus does not take the uncertainty of trajectory representation into account, we have tried to deal with this notion of uncertainty through use of collective knowledge of trajectories.

III. Preliminary

The terms and notions that we use in this paper are as follows: A trajectory is a sequence of time-stamped sample points denoted by: $Tr^i = P^i_{t_1} \dots P^i_{t_n}$, where i is the trajectory ID and the indexes $(t_i \dots t_n)$ represent the time-stamps of samples. The length of trajectories are variable and there are missing samples due to different reasons (hardware failure, environmental conditions, etc.). A semantic place is a spatial region extracted from a complete trajectory where the person has spent more than a predefined amount of time (τ). A sub-trajectory $STr^i = P^i_{t_1} \dots P^i_{t_n}$ (the mobility data between two semantic places) is a fraction of a trajectory which has its first point in one semantic place (origin) and its last point is in another (destination). A path $PA^i = \{SP^1 \dots SP^k\}$ is a group of sub-paths (retrieved from the collective knowledge) that represent the realistic route from an origin to a destination. A sub-path $SP^i = \{g_i \dots g_n\}$ is a section of a path composed of a list of grid cells (g_i). Sub-paths can be considered as units of difference between paths.

Given a trajectory database denoted by D= $\{Tr^1, \dots, Tr^m\}$, where each trajectory represents a sequence of spatial points visited by a person during a day, we are interested in finding frequently traversed paths and sub-paths between semantic places.

IV. Methodology

A. Overview

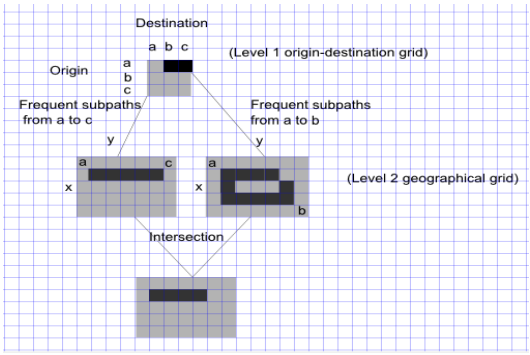

Fig. 3. Overview of our hierarchical approach

Grid based clustering methods quantize the object space into a finite number of grid cells on which all of the operations for clustering are performed [20]. We use a hierarchy of such grids as shown in Figure 3, to find frequently visited paths and

sub-paths. On a coarse grained scale, each grid cell represents common sub-trajectories based on their origin and destination. On a more fine grained scale, we find frequent paths and sub-paths between each pair of origin and destination. Finally, we can get the intersection of sub-paths from all pairs of origin and destination.

B. Level 1: Semantic grid

We can use semantic information of the semantic grid to first cluster sub-trajectories based on the conceptual information we can achieve from the regions of interest. Regions of interests are places where a person stays longer than a predefined threshold. We extract such regions by the method proposed in [21], in which each stay point is a place where the speed of the person is near zero. Then we will group sub-trajectories into clusters such that each cluster contains sub-trajectories between the same origin and same destination. For instance, a grid cell in the semantic grid can store all sub-trajectories from home (origin) to work place (destination). (Existence of a pattern mining layer on top of the semantic grid is also possible to find the frequent semantic trajectories [18]).

C. Level 2: geographical grid

In the next level, we cluster sub-trajectories to find frequently visited paths and sub-paths between each pair of origin and destination. The challenge to face here is that these sub-trajectories are already somewhat similar (as they have the same origin and destination). Therefore, it is necessary to first find the source of difference between them. Additionally, some sub-trajectories have missing points which make their correct representation difficult and consequently make them unclassifiable with respect to different paths traversed.

To address these challenges, we first aggregate all sub-trajectory points (from the same semantic grid cell) to find a connected neighbourhood between the origin and destination based on the common knowledge of sub-trajectories. Then, we find the source of difference between paths in such neighbourhood. Afterwards, we redefine and group sub-trajectories with respect to these units of difference. These procedures are better expressed in the following sections.

1) Step 1: Finding connected neighbourhoods

We form a geographical grid of size $\mathrm{M} \times \mathrm{M}$. Having the start time of all sub-trajectories synchronized, each grid cell denoted by g_i, $(1< i < \mathrm{M}^2)$ will hold the number of sub-trajectories which have a point in it, denoted by $c(g_i)$, along with the median of their time index, denoted by $m(g_i)$. With this median we can later keep the order of sub-paths.

Assuming that there is no backwards movement between the origin and destination, if we only filter the cells $\{g_i | c(g_i) >$ defined threshold$\}$, the layout of frequently visited paths between the origin and destination will become visible (in form of a connected neighborhood) based on the common knowledge of sub-trajectories. As seen in Figure 4, the paths that form a connected neighborhood may have some sub-paths in common and some different sub-paths. These sub-paths can be considered as the cause of (dis)similarity between two paths. Therefore, we need to break the connected neighbourhood to sub-paths to be able to use them to define frequent paths.

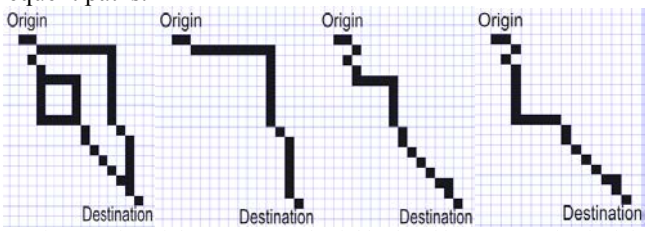

Fig. 4. Left to right: A connected neighborhood between a pair of origin and destination and 3 frequently visited paths which can be intuitively inferred from collective knowledge of all trajectories.

2) Step 2:Finding sub-paths in connected neighbourhoods

A frequently visited path can be represented as an ordered list of sub-paths. An idea to find sub-paths in the connected neighbourhood is to find breakpoints where a group of paths meet each other (converge) or where they separate from each other (diverge). Afterwards, the connected neighbourhood between these breakpoints can be defined as sub-paths.

The Algorithm **SPdefine** explains how we find sub-paths between the origin and destination:

Algorithm SPdefine
Input: start point S, threshold TH, a set of grid cells G={g|c(g)>0}
Output: A list of LSubPaths

```
1. Add S to a queue Q and to List of Breakpoints LBP
2. While Q is not empty
3.     Do
4.         SC← dequeue a cell from queue
5.     While SC is not visited
6.     Visit SC
7.     Add SC to TempSubPath
8.     NonAdjacentNeighbors← All non-visited non-adjacent neighbor of SC
9.     with C(SC)> TH
10.    Intersection=0
11.    While | NonAdjacentNeighbors |=1 & ~Intersection
12.        Visit NonAdjacentNeighbors(1)
13.        Add NonAdjacentNeighbors(1) to TempSubPath
14.        Intersection← Check if visited neighbors are in (LBP –
15.        TempSubPath (1))
16.    End
17.    If | NonAdjacentNeighbors |>1
18.        Add NonAdjacentNeighbors(1..n) to Q
19.        Add SC to the LBP
20.    End
21.
22.    Add TempSubPath to the list of LSubPaths
23. End
```

This algorithm starts traversing the grid from a cell which is selected as starting point (the cell which represents the origin) and follows the path on the grid by iteratively selecting neighbours (referred to as selected cell(s)(SC)) and moving forward. For moving forward, we define the concept of adjacent neighbours and non-adjacent neighbours as shown in Figure 5. The reason for choosing this definition is to be able to select more than one grid cell whenever necessary, for example when the width of a path is more than the width of a grid cell. Adjacent neighbours are neighbours of selected cell(s) which have a common edge and non-adjacent neighbours are those neighbours without a common edge. A group of adjacent neighbours should be considered as one

non-adjacent neighbour for the selected cell. Therefore, when we chose neighbours as the next round's selected cell(s), a group of adjacent neighbours might be chosen.

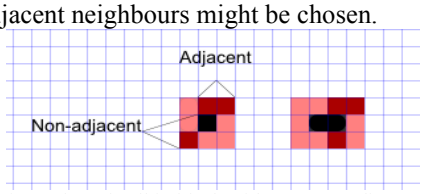

Fig. 5. left: a selected cell in black with two non-adjacent group of neighbouring cells (red) to move forward from. Right: two selected cells (black) and their two non-adjacent neighbours (red)

In each iteration, the algorithm extracts selected cell(s) from the queue, adds it to the start of a sub-path and checks its neighbours. When there is only one unvisited non-adjacent neighbour (N) (with $c(N) > TH$) and no breakpoints in the neighbours , this selected cell(s) will be added to the sub-path (lines 6-16). In case there are more than one adjacent cell to move forward from, this selected cell(s) will be added to the list of break points, the non-adjacent neighbours will be added to the queue, the current sub-path will be terminated and added to the list of sub-paths (lines 17-20). After a while, some sub-paths may have cells from both of their ends in the queue (a cell from their start and a cell from their end). In order to avoid traversing these sub-paths twice, when removing cells from the queue we will only select the cell(s), which are not already visited (lines 3-5). Finally, we will have a list of sub-paths in which each sub-path is defined by a list of cells denoted by $\{g_i | \, i \in M * M \}$ and one or two breakpoints (one on each end).

3) *Step 3: Ordering sub-paths in the tree of sub-paths*

After finding sub-paths in a connected neighbourhood between a pair of source and destination, we order them in a tree (we refer to it as tree of sub-paths). This ordering is done by matching the breakpoint in the beginning and end of each sub-path. As it can be seen in figure 6, the ordered sequence of sub-paths from the route to the leaf of the tree shows the frequent paths from origin to destination. In this figure, the frequent paths inferred from the tree of sub-paths are :$PA^1=\{SP^1, SP^2, SP^7\}$, $PA^2=\{SP^1, SP^3, SP^5, SP^6, SP^7\}$, $PA^3=\{SP^1, SP^3, SP^4, SP^6, SP^7\}$.

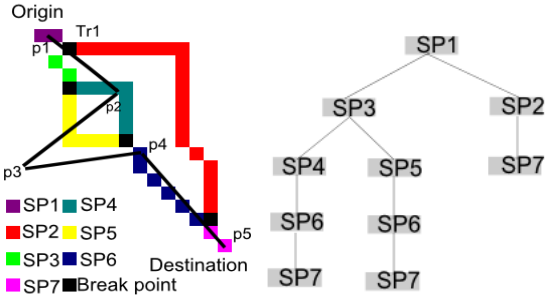

Fig. 6. Left: A connected neighborhood between a pair of origin and destination with 7 sub-paths. Right: the representative tree of sub-paths

4) *Step 4: Redefining trajectories based on sub-paths and clustering them based on the tree of sub-paths*

After fragmenting a neighbourhood to the frequently visited sub-paths and finding frequently visited paths using the tree of sub-paths, we redefine each sub-trajectory in terms of sub-paths. We start reading all the points on a sub-trajectory in order. If a point was in or near (in the neighbouring cells) a sub-path it will be replaced by that sub-path. This means that, existence of only one point in the sub-path indicates that this sub-trajectory passes the sub-path completely. Other points which are not on any sub-paths will remain intact (in case we are interested in outliers too). For instance, looking at Figure 6, the new representation of sub-trajectory STr^1 will be $\{SP^1, SP^4, P^3, SP^6, SP^7\}$.

If we consider that each path in the tree is the core of a cluster, then the final step is to assign the redefined sub-trajectories to the correct cluster. For the comparison we have chosen to use a similarity measure similar to LCSS [11], which ranks the matching parts between two time series based on their similarity. If $SP_{1\ldots m}$, and $SRT_{1\ldots n}$, denote the list of sub-paths from a path (cluster), and the sub-path list of a redefined sub-trajectory, then the similarity measure between the redefined sub-trajectory and a path (cluster) formed by the tree is:

$$SM = \begin{cases} 0 & if\ m = 0\ or\ n = 0 \\ SM\big(rest(SP), rest(SRT)\big) + 1 & if\ (ST_1 = SRT_1) \\ SM(Rest(SP), SRT) & otherwise \end{cases}$$

In this way, each sub-trajectory is compared to all clusters and assigned to one or some clusters based on the maximum similarity. The points that remain on the redefined sub-trajectory can be ignored as being noise.

During this procedure we can also score the sub-paths. The score of a sub-path SP_i denoted by $Score(SP_i)$ is the number of the sub-trajectories that have followed it. If the maximum similarity measure of a sub-trajectory and the paths (clusters) is owned by one path (cluster), then the sub-trajectory belongs to that path (cluster) and the score of all sub-paths on the path are incremented by one. If the maximum similarity measure of the sub-trajectory is shared by some paths (clusters), that sub-trajectory is uncertain between those paths (clusters). We can increment the score of the sub-paths forming those paths by 1/(number of paths with maximum similarity measure). Sub-trajectories that have equal similarity measure to all paths (potentially because they are only following the sub-paths on the start and end) and those with a considerable number of remaining points are outliers.

Let us consider the tree shown in Figure 6. A sub-trajectory, which is redefined as $\{SP^1, SP^5, SP^7\}$ has a similarity measure of 2 to PA^1 and PA^3 and similarity measure of 3 to PA^2. Therefore, it will be clustered with PA^2. This increases the score of each sub-paths on PA^2 by one. A sub-trajectory represented by $\{SP^1, SP^3, SP^7\}$ is similar to PA^2 and PA^3 with a similarity score of 3 while its similarity score to PA^1 is 2. Therefore, it will be considered as uncertain between PA^2 and PA^3, and cause the score of each sub-path on these paths to increment by ½ score.

If we consider having N number of trajectories and P clusters (frequent paths), the complexity of the abovementioned clustering approach will be $O(NP)$. For a density based clustering approach (e.g. Traculus [8]) the complexity will be $O(N^2)$. Therefore, when the number of

frequent paths (P) shared by a large number of trajectories are limited our approach performs more efficiently.

D. Level 3: Intersection

After all the sub-paths are scored with respect to the sub-trajectories that follow them for each cell of the semantic grid (each pair of source and destination), we can compare the sub-paths from one cell of the semantic grid to the sub-paths of other cells. In case the intersection of two sub-paths (in terms of the id of geographical grid cells) is not empty the intersected sub-path $SP_c = \{g_i | g_i \in SP_a \cap SP_b\}$ will be added to the list of sub-paths with a score equal to the score of two sub-paths ($Score(SP_c) = Score(SP_a) + Score(SP_b)$). By doing so, we will have a list of scored sub-paths, out of which the top sub-paths can be chosen as most frequently visited sub-paths.

V. EVALUATION

A. Experimental results

In this section we will present the result of the experiments we performed on the data that we have collected with a custom designed sensor node (Figure 7). Although, the sensor node is designed to sample location data derived from GPS every one minute, the collected data shows that the sampling frequency is very irregular.

Fig. 7. Image of the custom designed sensor node

Before analysis, we have formed regions of interest by extracting regions where the person has stayed longer than 30 minutes. Then we chose the group of sub-trajectories between two different staying points. The only previous clustering approach that addresses the problem of finding common sub-trajectories is Traculus [8] which first partitions trajectories based on the change in the behaviour of trajectory (e.g. direction). It then clusters the line segments using a line-based similarity measure. The behaviour of sub-trajectories that we formed between two semantic places are quite similar so we simply consider the sub-trajectory partitions as being the line segments achieved through interpolating consecutive measurements.

We compare clustering of 74 sub-trajectories between two semantic places shown in Figure 8-a. The distance between these two places has been traversed through two distinct paths shown in Figure 8-b-c. Some sub-trajectories have enough number of points to be assigned to path 1 or path 2 while some other have only points on the intersected sub-paths and cannot be clustered by human eye (see Figure 1). The goal is to distinguish between these two paths as two clusters, to find the number of sub-trajectories that have followed them and also to find their intersection as the most frequently visited sub-path. As shown in Figure 8-d, Traculus will only find one cluster. The reason is that due to closeness of two different paths there is small spatial difference between sub-trajectories of two different paths and the uncertain trajectories fill this gap. In addition, existence of missing points and spatial closeness causes that different trajectories look the same. The tree of sub-paths formed by our method, however, can find two frequent paths (clusters) between the source and destination with 4 sub-paths. Traculus represents the trajectories by getting an average of the cluster by a sweeping mechanism. We, however, represented the sub-paths by getting the average of points in each grid cell and ordered them by their median of timestamps. Using this mechanism the representation of sub-paths is closer to the realistic representation of them. Moreover, our method can make distinction between spatially close paths and uncertain sub-trajectories.

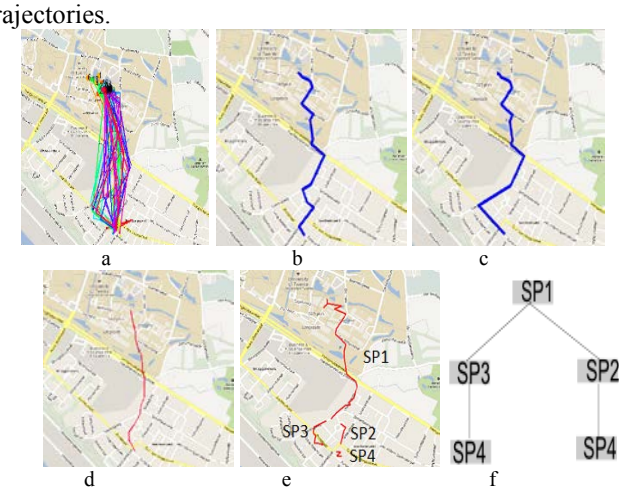

Fig 8 a) group of sub-trajectories, (b,c) two different paths common in one subpath b) path1 c) path2 d) representation of the only cluster found by Traculus e) representation of 4 sub-paths identified by our method f) the tree of sub-paths

TABLE I

NUMBER OF CLUSTERED TRAJECTORIES BEING IDENTIFIED USING THE TREE OF SUB-PATHS.

Clusters	Number of sub-trajectories assigned to the cluster with different grid sizes (GS)			Actual number of sub trajectories in clusters
	GS= 70*70	GS= 80*80	GS= 100*100	
Path 1	13	22	22	24
Path 2	12	35	37	38
Uncertain	49	17	15	12

TABLE II

SCORE OF SUB-PATHS

Sub-path	Score of sub-path with different grid sizes (GS)			Actual number of sub trajectories that have a point in the sub-path
	GS= 70*70	GS= 80*80	GS= 100*100	
Sub-path 1	74	74	74	74
Sub-path 2	40.5	30.5	29.5	24
Sub-path 3	37.5	42.5	44.5	38
Sub-path 4	70	74	74	74

We performed experiments with 3 different grid sizes. Table 1 and Table 2 show number of frequently visited paths and the score of sub-paths found respectively. In each table the last column shows the actual values which we measured

by analysing sub-trajectories visually. Table1, shows that the precision of the method increases as the grid size decreases. This is due to the fact that, the smaller the cell is, the easier it is to precisely represent the start and end of a sub-path. Therefore, the sub-trajectories that only have points near the start and end of a sub-path are better assigned to the right sub-path. Table 2, shows that the scores of sub-paths 2 and 3 are more than actual number of sub-trajectories that have a point on them. The reason is that, there are 12 unclassifiable trajectories between two paths which we chose to split their score between their sub-paths.

B. Different approaches and desirable properties

Table 3 compares different approaches in terms of their support for different desirable properties in finding frequent sub-paths between semantic places. Generally, methods that are interpolation based are sensitive to uncertainties caused by discrete sampling and missing points. Methods that do not rely on a framework for partitioning trajectories are not able to find their similar sections. Noisy measurements can be ignored by a non-metric system of comparison (not measuring the distance of points but counting the number of similar points) [10] or density based clustering. In this paper, we address the problem of finding common sub-trajectories by a hierarchical approach. We ignore noisy measurements by scoring the similarity of trajectories to frequent paths. We use collective knowledge of sub-trajectories to redefine them. By doing so, we deal with trajectory representation uncertainty caused by discrete sampling and the problem of missing sample. We also address measurement errors to some extend by defining adjacent neighbours to move forward with and assigning points on sub-paths if they are close to them. However, in case sub-paths are very close to each other, our method cannot well deal with uncertainty.

TABLE III

COMPARISON OF DIFFERENT TRAJECTORY CLUSTERING APPROACHES WITH RESPECT TO DIFFERENT DESIRABLE PROPERTIES

Methods	**Desirable properties**			
	Sub-trajectory based clustering	**Being prune to notions of uncertainty**		
		Noise	**Missing samples & discrete sampling**	**Measurement error**
Clustering by EDR [9]	No	Yes	No[1]	No
Traculus [8]	Yes	Yes	No	No
[19]	No	No	No	Yes
Our method	Yes	Yes	Yes	No

VI. CONCLUSION

In this paper we propose a hierarchical approach to find frequently visited paths and sub-paths traversed by moving objects by using both semantic information and geographical data. First, we use the semantic information (stopping points) to cluster trajectories into groups of semantically similar sub-trajectories. We then find frequently visited paths and sub-paths in each of these clusters. We use the notion of collective knowledge to deal with uncertainty of trajectory representation and show that compared with existing approaches we can more efficiently identify similar trajectories in presence of uncertainty.

[1] EDR has a mechanism to deal with gaps in data, but in case the paths are close to each other and have common sub-paths it will not be able to deal with the uncertainty of missing samples and discrete sampling

REFERENCES

[1] Z. Li, Et. al., "MoveMine: mining moving object databases," In Proc. international conference on Management of data Indiana, USA, 2010.

[2] Y. Zheng, Et. al., "Urban computing with taxicabs," In Proc. international conference on Ubiquitous computing, Beijing, China, 2011.

[3] J. Yuan, Et. al., "T-drive: driving directions based on taxi trajectories," In Proc. 18th SIGSPATIAL San Jose, California, 2010.

[4] P. K. Coe, Et. al., "Validation of elk resource selection models with spatially independent data," *The Journal of Wildlife Management,* vol. 75, pp. 159-170, 2011.

[5] D. Pfoser and C. S. Jensen, "Capturing the Uncertainty of Moving-Object Representations," In Proc. Symposium on Advances in Spatial Databases, 1999.

[6] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," In Proc. 18th ACM SIGKDD Beijing, China, 2012.

[7] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," In Proc. fifth ACM SIGKDD San Diego, California, United States, 1999.

[8] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," In Proc. ACM SIGMOD, Beijing, China, 2007.

[9] L. Chen, Et. al., "Robust and fast similarity search for moving object trajectories," In Proc. 2005 ACM SIGMOD Baltimore, Maryland, 2005.

[10] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," In Proc. 1994 ACM SIGMOD Minneapolis, United States, 1994.

[11] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Data Engineering, Proc.. 18th International Conference on*, 2002, pp. 673-684.

[12] Y. Byoung-Kee, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," *In Proc. 14th Int. Conf. Data Engineering*, 1998, pp. 201-208.

[13] L. Chen and R. Ng, "On the marriage of Lp-norms and edit distance," In Proc. Thirtieth int. conf. on Very large data bases Toronto, Canada, 2004.

[14] C.-C. Hung, W.-C. Peng, and W.-C. Lee, "Clustering and aggregating clues of trajectories for mining trajectory patterns and routes," *The VLDB Journal,* pp. 1-24, 2011/11/01 2011.

[15] V. Bogorny, B. Kuijpers, and L. O. Alvares, "ST‐DMQL: A Semantic Trajectory Data Mining Query Language," *International Journal of Geographical Information Science,* vol. 23, pp. 1245-1276, 2009/10/01 2009.

[16] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," In Proc. 13th ACM SIGKDD, San Jose, California, USA, 2007.

[17] G. Trajcevski, Et al., "Managing uncertainty in moving objects databases," *ACM Trans. Database Syst.,* vol. 29, pp. 463-507, 2004.

[18] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," In Proc. 13th ACM SIGKDD, San Jose, California, USA, 2007.

[19] N. Pelekis and E. al., "Clustering uncertain trajectories," *Knowledge and Information Systems,* vol. 28, pp. 117-147, 2011.

[20] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*: Elsevier, 2006.

[21] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," In Proc. ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008.