Discovering Corridors From GPS Trajectories

Nikolaos Zygouras
National and Kapodistrian University of Athens
nzygouras@di.uoa.gr

Dimitrios Gunopulos
National and Kapodistrian University of Athens
dg@di.uoa.gr

ABSTRACT

The increasing pervasiveness of GPS-enabled devices results in the collection of massive trajectories datasets. The vast amount of the generated location data is particularly difficult to be processed, interpreted and analyzed, due to its complexity. Nevertheless, in many cases a considerable number of moving objects share common paths and their whole trajectory can be decomposed as a sequence of such commonly accessed paths, referred as corridors. In this paper we formulate the problem of corridor discovery using GPS data that represent user trajectories. We initiate research for developing an algorithm to solve this problem efficiently and we present initial experimental results that demonstrate our approach.

ACM Reference format:
Nikolaos Zygouras and Dimitrios Gunopulos. 2017. Discovering Corridors From GPS Trajectories. In *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA, November 7–10, 2017,* 4 pages.
DOI: 10.1145/3139958.3139994

1 INTRODUCTION

In recent years the constantly increasing usage of ubiquitous computing devices that trace the moving objects' locations creates voluminous collections of trajectory data. A wide range of applications needs to access, use and process these data. For instance, in smart cities the traffic operators monitor buses' GPS positions [17] in order to detect traffic anomalies, while in zoological studies zoologists investigate animals' movements in order to detect their interactions with the ecosystem [5]. The massive amount of the unstructured trajectory data make it difficult to organize, process and understand the objects' movements (i.e. taxis, animals and pedestrians trips). Consequently, it becomes important to develop novel techniques and algorithms able to detect patterns in the objects' movement.

In this paper we describe a novel approach towards detecting frequent paths from a trajectory database. We refer to such frequently followed paths as "corridors". A corridor can be thought as a route that is commonly traversed by a considerable number of moving objects. Consider, for example, 5000 buses' trajectories at the city of Dublin, illustrated at the left part of Fig. 1. Our approach summarizes the provided trajectory dataset, detecting the set of the most frequently followed corridors. The right image of Fig. 1 illustrates the output of our approach showing the 50 most frequent corridors of buses' movements. It is particularly difficult for a human to understand what are the main flows of vehicles, just by plotting the trajectories on top of a map. Our approach is able to summarize the raw data providing as output to the user an abstract view of the main objects' movements, that highlight the moving patterns.

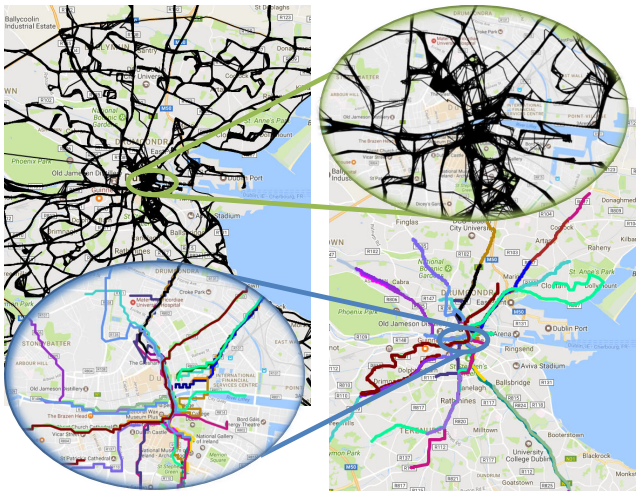

Figure 1: Buses' trips at Dublin (left) and the 50 most frequent corridors (right) detected by our approach.

We describe below our **desiderata** for learning corridors from a collection of trajectories: (*i*) *Objects moving in unconstrained space*: We assume that the moving objects can move in space without a road network constrain. (*ii*) *Mining massive trajectory datasets*: Moving objects generate large collections of movement data, thus a corridor learning technique should be capable to cope with large datasets of trajectories. (*iii*) *Discover a set of corridors:* Our technique focuses in discovering corridors that simplify the incomprehensible large collection of trajectories. This facilitates the understanding of the main mobility patterns.

It should also be mentioned that we either partition the trajectories into "trips" or we assume that we get them as "trips". A simple technique to get "trips" from a trajectory is to detect the points where the user stayed for a while and split there the trajectory [15].

The **contributions** of our approach are described below: (*i*) Summarizing a vast trajectories' database returning as output the set of corridors that represent the major movement patterns. (*ii*) Formally define the problem of learning corridors using the MDL principle; equating the corridors learning with the compression of the trajectories' database.

2 BACKGROUND AND RELATED WORK

Trajectory Pattern Mining. Research work in trajectory data mining can be distinguished into two main categories. The first one assumes that the objects are moving in a known, well structured road network, while the second assumes that such information does not

SIGSPATIAL'17, Los Angeles Area, CA, USA
 978-1-4503-5490-5/17/11…$15.00
DOI: 10.1145/3139958.3139994

exist or is not provided and the objects are moving in the *unconstrained space*.

Road Network: Chen et al. in [6] proposed a technique that computes a dictionary of pathlets, solving a constrained optimization problem. As pathlet they defined a fraction of the path. The pathlets were used to reconstruct the trajectories, compressing the trajectories' dataset. This work is the closest to our approach but assumes knowledge of an underlying map. Evans et al. in [7] studied the problem of summarizing a set of trajectories, that move on top of a road network, detecting k-Primary Corridors. They initially computed a track similarity matrix and then k-medoids clustering algorithm is applied to detect the k-PC.

Unconstrained Space: Lee et al. [11] proposed a trajectory clustering algorithm, named *TRACLUS* that discovers common subtrajectories from trajectories databases. *TRACLUS* contains two main steps: (i) partition a trajectory into a set of subtrajectories, adopting the MDL principle (ii) group the different subtrajectories together using a clustering algorithm that bares a lot of similarities with DBSCAN clustering algorithm. Lastly they returned a representative trajectory for each cluster.

Additionally the problem of summarizing trajectories in corridors has been investigated in [16]. In order to extract the corridors they segemented trajectories into subtrajectories using a mesh grid, then they grouped subtrajectories into clusters using an agglomerative clustering algorithm that considers their discrete Fréchet distance, creating clusters of similar movement. Finally the corridors were the sequences of the detected clusters with similar starting/ending locations. In [8] the authors transformed the trajectories into sequences of regions of interest and they found frequent patterns in these sequences considering the travel times. The authors in [4] proved that the problem of finding subtrajectories' clusters is NP-Complete.

Gudmundsson et al. [9] proposed a pipelined algorithm for clustering movement data, the algorithm splits trajectories in subtrajectories and provides labels for each subtrajectory according to its geometric property (i.e. sharp left turn). Then trajectories are transformed in sequences of these labels used to detect frequently occurring strings (motifs). Finally DBSCAN algorithm is applied in order to detect similar subrajectories. Another work that aims to identify paths followed by many tracked objects was presented in [2]. The authors used the Fréchet distance in order to calculate the distance between two subtrajectories. Then an *Apriori*-based algorithm is proposed in order to detect sets of subtrajectories that form a trajectory clique. Mamoulis et al. [12] transformed trajectories to sequences of regions and they detected periodic patterns in these sequences applying association rule mining.

3 PROBLEM DEFINITION

In this work we propose an efficient algorithm for detecting the main patterns of movement. The proposed framework receives as input a *trajectory Database* $\mathcal{D} = \{T_1, T_2, \ldots, T_N\}$ that contains N trajectories and detects a set of paths $\mathcal{H}$, named *corridors*, that are frequently followed by the moving objects of $\mathcal{D}$.

A *Trajectory* $T_i : c_1 c_2 \ldots c_{M_i}$, is a time ordered sequence of M_i coordinates of the i^{th} moving object. Originally the trajectories' coordinates lie on a two dimension Cartesian plane, $c \in \mathbb{R}^2$.

In this work we decided to discretize the coordinates' space in order to facilitate the exploration of hidden patterns in the objects' movements. We applied a *grid* of uniformly sized cells, mapping the coordinates to discrete grid cells.

Definition 3.1. (*Corridor*): A *corridor* $F \in \mathcal{H}$ is an *induced* trajectory that connects two coordinates through a *dense* path that was *frequently followed* by a considerable number of moving objects in $\mathcal{D}$. In more detail a corridor satisfies the following properties:

- *Induced*: there is not necessarily any other trajectory identical with F in $\mathcal{D}$, but is constructed capturing and aggregating the underlying information from the objects' movement.
- *Dense*: two consecutive grid cells of a corridor should be spatially close to each other .
- *Frequently followed*: a considerable number of trajectories in $\mathcal{D}$ contain subtrajectories that share the same movement, captured by F.

Consider for instance the 4 trajectories visualized in Fig. 2. It can be observed that even though they have different origins and destinations they contain subtrajectories that share a common movement behaviour at the marked area. Furthermore a corridor that describes the objects' movement at this area can be induced (dense black arrow in Fig. 2), aggregating the information from the subtrajectories of the marked area. Our proposed framework is able to detect a set of such corridors $\mathcal{H}$, given a collection of trajectories $\mathcal{D}$.

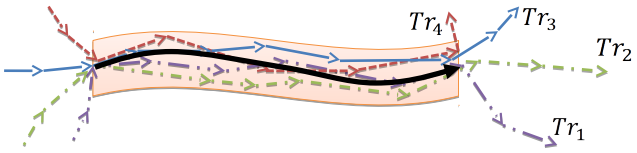

Figure 2: Example of 4 trajectories, that have different origins and destinations bus share a common path (corridor).

Given a trajectories database $\mathcal{D}$, a set of corridors $\mathcal{H}$ and a distance threshold θ_d we define the compressed dataset $\mathcal{D}|\mathcal{H}$, that is constructed replacing the subtrajectories of trajectories $T \in \mathcal{D}$ *covered* by a corridor $F \in \mathcal{H}$ with a pointer to F. A corridor F is said to *cover* a part or all the trajectory T if its similarity with any possible subtrajectory of T exceeds a threshold θ_d: $max(d(F, s)) \geq \theta_d \; \forall \, s \in subtrajectories(T)$. The similarity $d(\cdot, \cdot)$ could be any similarity measurement that measures the similarity between two trajectories. In this work we selected to use the LCSS.

Problem Definition: *Given a set of sparse trajectories $\mathcal{D}$ moving in the unconstrained space, assuming no time find the set of corridors $\mathcal{H}$ that minimizes the sum of $\mathcal{L}(\mathcal{H})$ and $\mathcal{L}(\mathcal{D}|\mathcal{H})$, where $L(\cdot)$ is the length of a data collection in bits.*

We adapt above the Minimum Description Length (MDL) principle [13] in order to formally define our problem, viewing learning as data compression.

4 LEARNING CORRIDORS

In this Section we give a preliminary description of our approach. We describe our architecture, for detecting a set of corridors, that can be decomposed into two main processing modules. More specifically in Section 4.1 we describe how to detect, from a given set of

trajectories, frequent sets of locations that are frequently visited by similar trajectories. In Section 4.2 we present how a set of corridors is detected aggregating the information from the similar parts of the given trajectories.

4.1 Discovering Frequent Sets of Locations

The first step towards detecting corridors is to decompose the space in different sets of locations frequently observed together in the objects movements. Call these sets of locations *frequent sets of locations* [1]. LDA is applied at the trajectories domain in order to detect K *frequent sets* of areas that share common traffic. Our LDA formulation is analogous to the typical NLP formulation where grid cells replace words, trajectories replace documents and frequent sets are analogous to topics. Our main intuition is that sets of neighboring grid cells could be grouped together into hidden frequent sets of locations and each trajectory can be modelled as a mixture of these frequent sets of locations.

4.2 Corridors Mining

4.2.1 Trajectories Segmentation. Below we describe how a trajectory is segmented into subtrajectories. Our approach creates a set of subtrajectories for each frequent set, that will be aggregated later to detect the set of corridors $\mathcal{H}$. Our method searches for intersecting cells between the trajectory and the cells of each *frequent set*. A subtrajectory for a frequent set k is generated from the cells of a trajectory T_i between the first and the last matched cells of T_i and the cells that are associated with frequent set k. If T_i contains more than θ_{gap} consecutive cells not matched, with the cells of the frequent set, then the trajectory is split further in 2 subtrajectories .

4.2.2 Grouping Subtrajectories. Here we describe how different subtrajectories are grouped together based on their similar movement along the grid cells of a frequent set. We applied the hierarchical clustering algorithm [10] to the set of subtrajectories of each frequent set, following a bottom-up approach. The algorithm stops merging clusters when the intra-cluster distances exceed a threshold θ_{cl}. We selected here to use Dynamic Time Warping (DTW) [3] in order to measure the distance between two trajectories. The distance between two cells is measured using the Manhattan distance. DTW is able to distinguish the different directions of movement and assign lower distance to the trajectories that have similar shape and ordering of locations. To speed up the computations we used the algorithm presented in [14], computing the DTW distance between two trajectories using the R-tree index, inserting in the R-tree smaller Minimum Bound Rectangles (MBRs) that surround the whole trajectory.

4.2.3 Corridors Induction. Given a cluster of trajectories, characterized by their similar movement across the cells of a particular frequent set; our objective is to induce a set of corridors. The objects' low GPS sampling rates complicates the corridors' detection, creating "uncertainty" on how objects moved among the GPS samples. Our approach transforms sparsely sampled trajectories (Fig. 3-i) in dense and informative trajectories (Fig. 3-ii) that describe in detail the objects' movement, using a set of similar trajectories (Fig. 3-iv).

In order to reduce the trajectories' uncertainty we follow a graph-based approach. A directed edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is

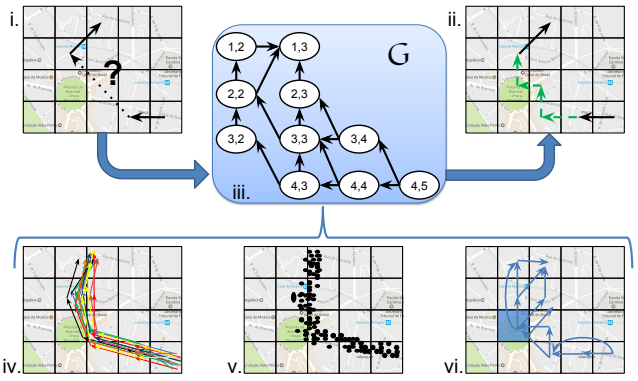

Figure 3: Example of a sparse uncertain movement (i) that is reconstructed to a detailed trajectory (ii), using the graph $\mathcal{G}$ (iii), generated from a bunch of sparse similar trajectories (iv), considering both the cells density (v) and the direction and the frequency of the cells' successors (vi).

constructed for *each* detected cluster. $\mathcal{G}$ comprises from a set of vertices $\mathcal{V}$ that correspond to the cells that trajectories accessed and a set of directed edges $\mathcal{E}$ that connect different adjacent grid cells. The weight w, that is assigned to each edge $e = (v_1, v_2)$, depicts the likelihood of moving from v_1 to v_2. The weights are mined from the cluster's subtrajectories. Higher weight is provided for the neighboring grid cells that have been visited more frequently. For instance a higher weight will be assigned to the most visited cells of Fig. 3-v. Also greater weight is assigned to the grid cells that have similar direction with the direction of the majority of the cluster's moving objects that departed from the under investigation cell v (Fig. 3-vi). Finally the most likely path that the object followed between two non adjacent consecutive grid cells is detected posing a shortest path query over the graph $\mathcal{G}$.

Finally a set of corridors C is extracted for each cluster. We complete each one of the cluster's subtrajectories, posing shortest path queries to $\mathcal{G}$ if two consecutive nodes of the subtrajectories are not adjacent. Each detailed path is inserted in the set of corridors C if the minimum distance from any of the already inserted corridors in C does not exceed the distance threshold θ_d.

5 EVALUATION

In this Section we present our initial experimental results that evaluate the performance of our approach. We used ten different synthetically generated benchmarks in our experiments. Each one of them contained different number of dense patterns DP and noisy patterns NP. Each dense pattern is followed by 15 trajectories, while each noisy pattern is followed by a single trajectory. The ten benchmarks were generated combining 5, 10, 20, 40 or 80 dense patterns with 0 or 200 noisy patterns. In the experiments presented bellow we denote as *COR* the experiments that do not contain noisy patters and as *COR-Noisy*, those experiments that contain noisy patterns.

Evaluation Metrics: In order to measure the quality of the proposed techniques we measure the following: (i) MDL: referring to the

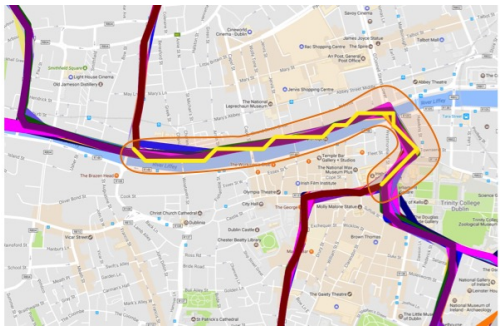

Figure 4: Buses moving at the upper bank of Liffey river in Dublin, and the detected corridor (highlighted yellow line).

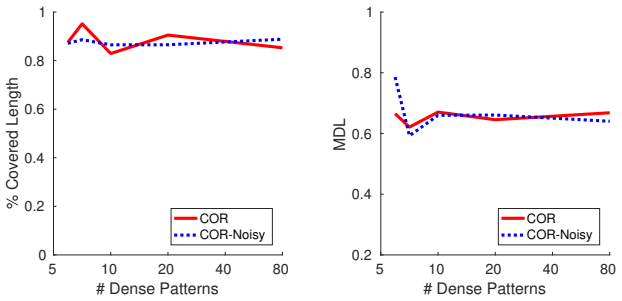

Figure 5: The *Coverage Length* by the detected corridors (left) and the corresponding *MDL scores* (right).

achieved compression, given by the following equation:

$$MDL = \frac{\mathcal{L}(\mathcal{H})\mathcal{L}(\mathcal{D}|\mathcal{H})}{\mathcal{L}(\mathcal{D})} \quad (1)$$

Obviously our objective is to find a set of corridors $\mathcal{H}$ that minimizes the *MDL score*. (ii) *Coverage Length*: represents the percentage of the synthetic patterns that is captured by the detected corridors.

Detecting Common Behavior: The movements of several buses at the city of Dublin are illustrated in Figure 4. The buses have different origins and destinations but they share a common movement behaviour, captured by the proposed technique. The detected pattern, at the upper bank of Liffey river, is followed by different bus lines transporting citizens from the west part of Dublin to the city centre.

Experimental Results: Here we describe the performance of the proposed technique on the synthetic datasets. Initially the left part of Fig. 5 illustrates the percentage of the patterns' length covered by the detected corridors for the datasets with and without noisy trajectory patterns, *COR* and *COR-Noisy* respectively. As we can see the detected corridors detect almost all the given dense patterns, irrespective of their number or the existence or not of noisy patterns. The *MDL score*s that our approach achieves are visualized at the right part of Fig. 5. We observe that the detected corridors capture the dense patterns resulting in low *MDL score*. Finally we can see that the *MDL score* is not affected by the existence of noise, except from the noisy benchmark that contained 5 dense patterns, where the amount of noisy trajectories (200) is much larger than that of the dense patterns (75).

6 CONCLUSION

In this work we proposed a pipelined approach for detecting a set of frequently accessed corridors from a vast collection of trajectories. Initially we applied a well known topic modelling technique to detect frequent sets of locations and then we derived frequent paths at these locations. Our initial experimental results demonstrate the ability of our approach to summarize a large collection of trajectories to a few number of frequently accessed paths.

ACKNOWLEDGMENTS

European Union Horizon2020 grant 688380 "VaVeL" and a Google 2017 Faculty Research Award.

REFERENCES

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, and others. 1996. Fast Discovery of Association Rules. *Advances in knowledge discovery and data mining* 12, 1 (1996), 307–328.

[2] Htoo Htet Aung, Long Guo, and Kian-Lee Tan. 2013. Mining sub-trajectory cliques to find frequent routes. In *International Symposium on Spatial and Temporal Databases*. Springer, 92–109.

[3] Donald J Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series.. In *KDD workshop*, Vol. 10. Seattle, WA, 359–370.

[4] Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler, and Jun Luo. 2008. Detecting commuting patterns by clustering subtrajectories. In *International Symposium on Algorithms and Computation*. Springer, 644–655.

[5] Clément Calenge, Stéphane Dray, and Manuela Royer-Carenzi. 2009. The concept of animals' trajectories from a data analysis perspective. *Ecological informatics* 4, 1 (2009), 34–41.

[6] Chen Chen, Hao Su, Qixing Huang, Lin Zhang, and Leonidas Guibas. 2013. Pathlet learning for compressing and planning trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 392–395.

[7] Michael R. Evans, Dev Oliver, Shashi Shekhar, and Francis Harvey. 2012. Summarizing trajectories into *k*-primary corridors: a summary of results. In *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems, SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012*. 454–457.

[8] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 330–339.

[9] Joachim Gudmundsson, Andreas Thom, and Jan Vahrenhold. 2012. Of motifs and goals: mining trajectory data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 129–138.

[10] L. Kaufman and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

[11] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. 2007. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 593–604.

[12] Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, and David W Cheung. 2004. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 236–245.

[13] Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica* 14, 5 (1978), 465–471.

[14] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn J. Keogh. 2003. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*. 216–225.

[15] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 29.

[16] Haohan Zhu, Jun Luo, Hang Yin, Xiaotao Zhou, Joshua Zhexue Huang, and F. Benjamin Zhan. 2010. Mining Trajectory Corridors Using Fréchet Distance and Meshing Grids. In *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I*. 228–237.

[17] Nikolaos Zygouras, Nikos Zacheilas, Vana Kalogeraki, Dermot Kinane, and Dimitrios Gunopulos. 2015. Insights on a Scalable and Dynamic Traffic Management System.. In *EDBT*. 653–664.