An efficient approach to trajectory similarity range query based on Fréchet distance

Jiajie Ma[1], Amin Vahedian Khezerlou[2], Dr. Xun Zhou[2]

[1]High School Affiliated to Shanghai Jiao Tong University, Shanghai, China [2]University of Iowa, Department of Management Sciences, Iowa City, IA

Introduction

·**Trajectory**: A **function of time** that describes the path of a moving object. $\textit{i.e.}\ \boldsymbol{f(t) = (x, y), where\ (x, y) \in \mathbb{R}^2}$

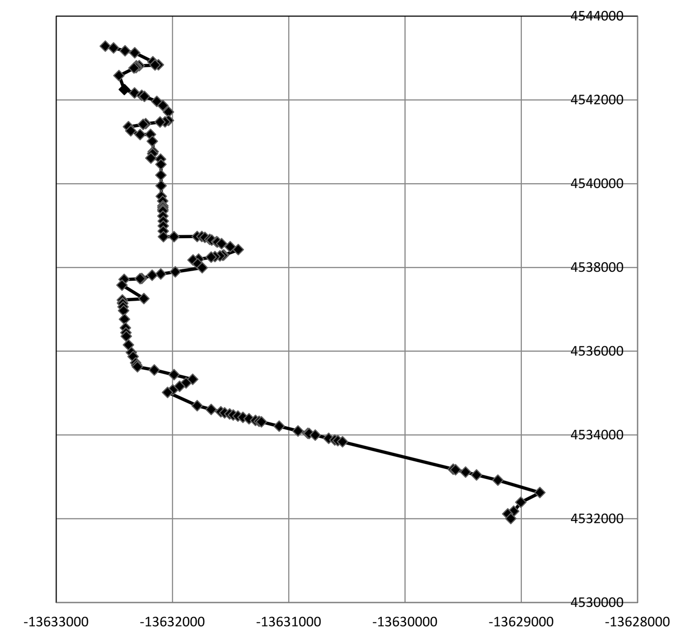


Figure 1. A polygonal trajectory in 2-D Euclidean Space

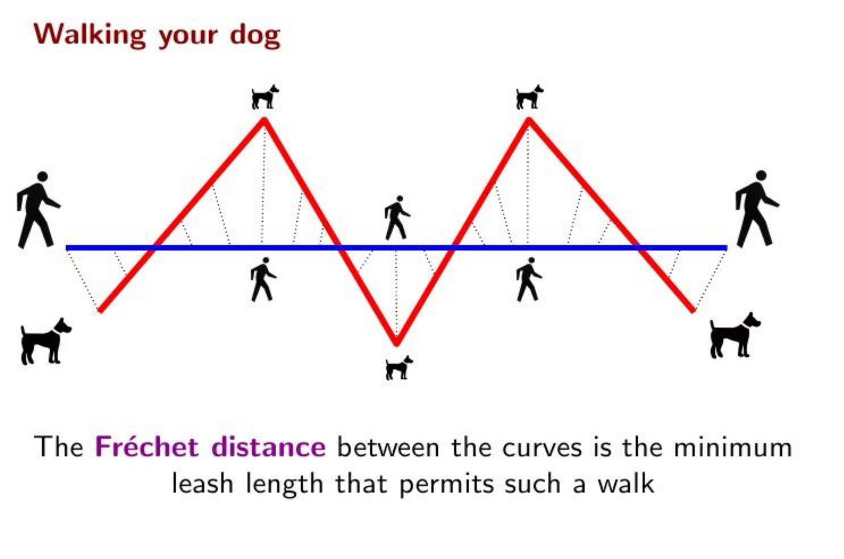


Figure 2. Intuitive definition of Fréchet distance
Retrieved from: https://www.slideshare.net/shripadthite/frechettalk

·**Fréchet distance**: "the **minimal length** of a leash that is necessary" for a man to walk his dog, "allowed to control their speed but not go backwards" (Alt & Godau, 1993, p. 76).

$$F(A,B) = \inf_{\alpha,\beta} \max_{i \in [1,n],\, i \in \mathbb{N}} \left\{ d\left(A(\alpha(i)), B(\beta(i))\right) \right\}$$

·**Significance & Motivation:**

This research will help advance analysis in these fields:

1) Trajectory database index and classification
2) Trajectory clustering and similarity identification
3) Movement pattern detection and prediction

·**Past works**: **1)** Algorithm for computing **Fréchet distance range query** between two polygonal curves using **dynamic programming: O(pq)** (Alt & Godau, 1993, p. 78).

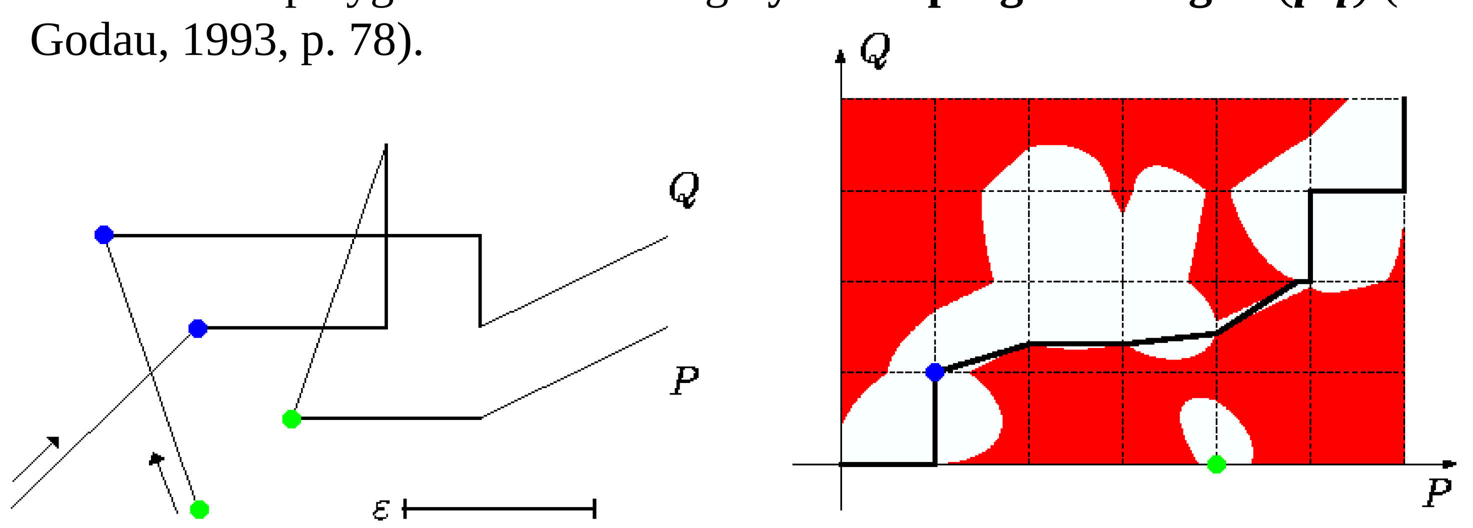
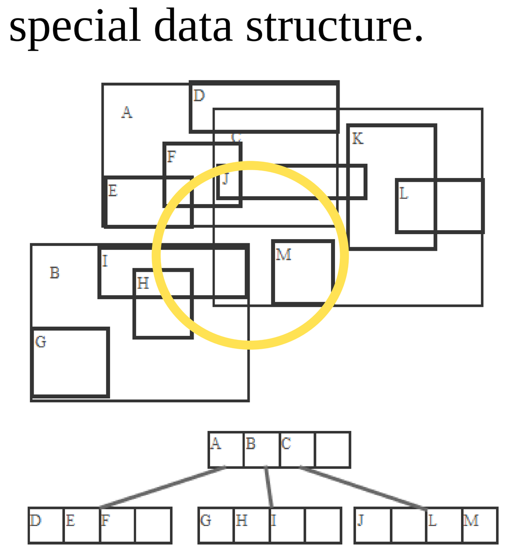


Figure 3. Free Space Diagram
Retrieved from: http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2002/StephanePelletier/

2) Filter-and-refine approach -> Trajectory index and retrieval: A query processing that searches for candidate trajectories efficiently with special data structure.



R-tree: ·nodes contain specific **spatial objects** (points, trajectories, etc.) or their **MBBs (Minimum Bounding Boxes).**

·**Range Search query** and **Nearest Neighbor Search** are performed efficiently (Guttman, 1984)

Figure 4. Two views of an R-tree example with a range search query
Retrieved from: http://xuyannus.blogspot.com/2014/05/

Objectives

Trajectory Similarity Query based on Fréchet distance:

Given: *A* ***query trajectory P*** *and* ***a large set of trajectory data*** X *and some* $\boldsymbol{\epsilon > 0}$ ***(query range)***

Find: *all the trajectories* $Q \in X$ *such that* $\boldsymbol{F(P,Q) \le \epsilon}$

Assumption: Trajectories are defined as **polygonal curves** in a **two-dimensional Euclidean Space**.

Methods

1.**Fréchet distance range query** between two polygonal curves was implemented by **depth-first search (DFS)** of a **directed acyclic graph**

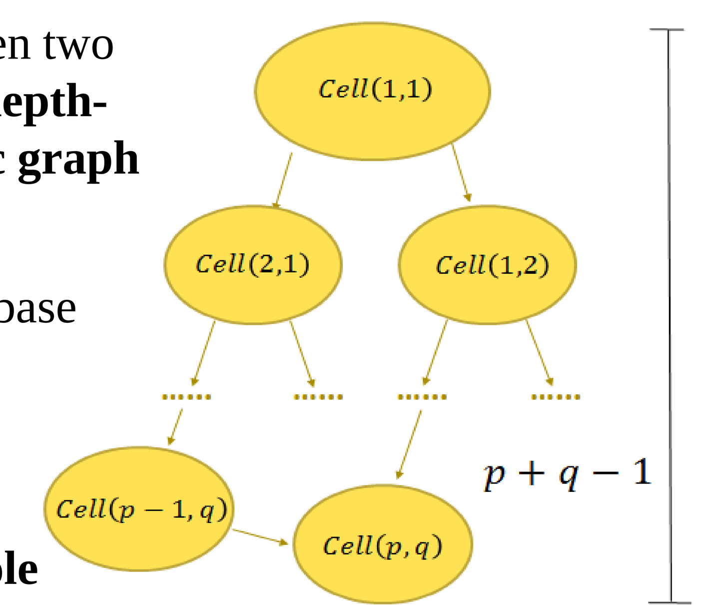


Figure 5. Directed acyclic graph view of depth-first search in Free space diagram

Baseline Approach:
Exhaustively searching the entire database with **DFS**

Point R-tree Method:

1. R-tree construction with **points (vertices)** of trajectories in the **whole database**
2. **Filter step**: Build an **MBB** around the **buffer** of the **query trajectory**: *to be a candidate, all the points of the trajectory should be in the box.*
3. **Refine step:** Apply **baseline approach** on all **candidate trajectories**

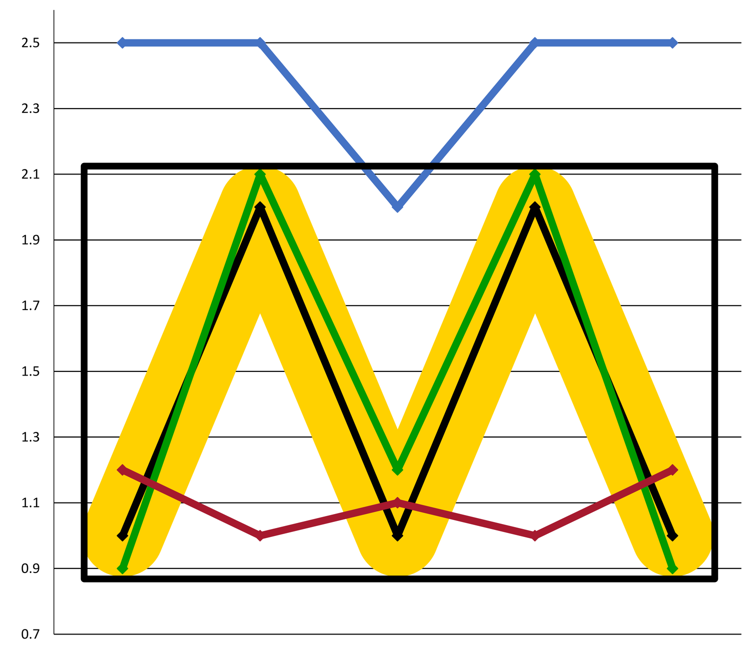


Figure 6. Point R-tree method filter step example

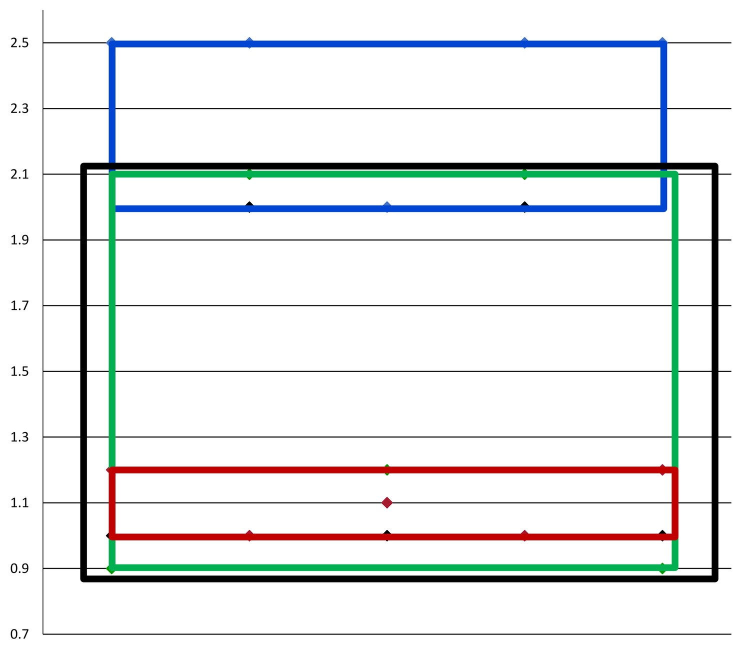


Figure 7. Trajectory R-tree method filter step example

Trajectory R-tree Method:

1. **R-tree** construction with **MBBs** of each trajectory in the **whole database**
2. **Filter step**: Build an **MMB** around the **buffer** of the **query trajectory**: *to be a candidate, the MMB of the trajectory must be in the box.*
3. **Refine step**: Apply **baseline approach** on all **candidate trajectories**

Results

Experiment setup:

·All algorithms were implemented in **C++** and **boost::geometry::index::rtree (1.6.2)** for R-tree construction.

·Programs were submitted to **University of Iowa Argon Cluster** for experiments.

Results Con'

·The dataset contains **20242 trajectories** in total and the **average length of all trajectories** is **31500.1**. (ACM GISCUP 2017)

·The **query range** is randomly selected **from 500 to 6000**, corresponding to **1% to 20%** of the average length of all trajectories.

·By default, **R-tree type** is chosen to be **Linear** and the **maximum entries in a node** is set to **20**

Fréchet distance range query time: 0.000583872 (s)

	Baseline	Point R-tree	Trajectory R-tree
R-tree construction time (s)	0	205.53	12.77
Average filter time (s)	0	0.9063052	0.00503462
Average number of candidates	20242	3691.27	3691.27
Average refine time (s)	11.01197187	2.202112018	2.281795463
Average query time (s)	11.01197187	3.11046	2.28845
Overall run time (s)	28644.8	6496.29	4640.08

Table 1. Run time comparison among all three methods

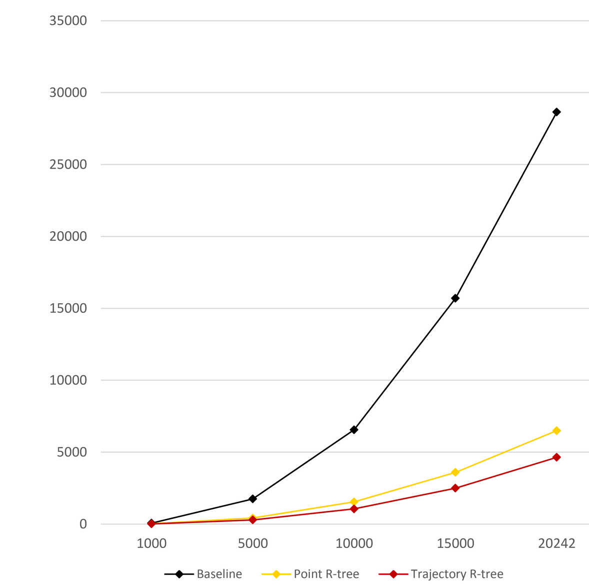


Figure 8. Overall run time of all three methods

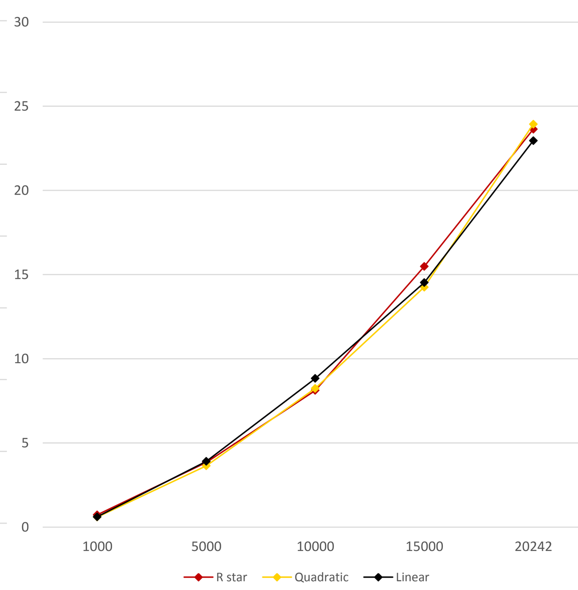


Figure 9. Overall run time of three R-tree types in Trajectory R-tree Method

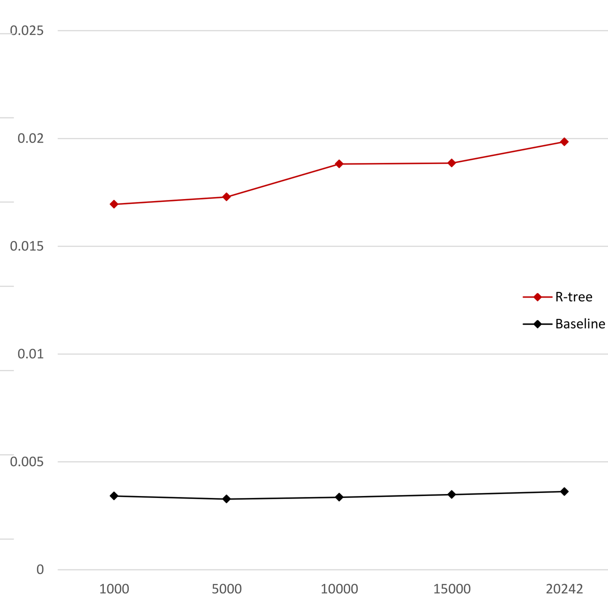


Figure 10. Ratio between the number of the answers and the candidates

Conclusion

·This research offers an efficient approach to trajectory similarity range query based on Fréchet distance in very large databases. With an effective spatiotemporal index, **trajectory R-tree method** significantly outperforms the baseline approach and is better than point R-tree method due to savings in R-tree construction time and filter time. Furthermore, it is expected that the advantage of this algorithm will be more remarkable in even **larger databases**.

Future works

·The adoption of **MBB** in **filter step** leaves relatively huge amount of **dead space** in **range query**, which evidently **increases the overall number of candidates**. Thus, the future direction can either change the **number of bounding boxes** around the query trajectory, or attempt **grid-based index**, which will both make the filter step theoretically more accurate.

Acknowledgement

Special thanks to all who made this project possible, including:
·Dr. Xun Zhou, Amin Vahedian Khezerlou
·SSTP and Belin-Blank Center

References

[1] Alt, H., & Godau, M. (1993). Computing the Fréchet Distance between two polygonal curves. Internat. J. Comput. Geom, 5(1), 75-91. https://doi.org/10.1142/S0218195995000064
[2] Guttman, (1984). R-trees: a dynamic index structure for spatial searching. Proceeding from SIGMOD '84: the 1984 ACM SIGMOD international conference on Management of data. New York, NY: ACM
[3] ACM GISCUP 2017. (2017). Trajectory dataset sample [dataset-sample.tgz]. Retrieved from http://sigspatial2017.sigspatial.org/giscup2017/download