# Trajectory Similarity Join in Spatial Networks

PVLDB, Vol.10(11), pages 1178–1189

Shuo Shang[1], Lisi Chen[2], Zhewei Wei[3], Christian S. Jensen[4], Kai Zheng[5], and Panos Kalnis[1]

[1]CEMSE Division, King Abdullah University of Science and Technology, Saudi Arabia
[2]Department of Computer Science, Hong Kong Baptist University, Hong Kong
[3]School of Information, Renmin University, China
[4]Department of Computer Science, Aalborg University, Denmark
[5]University of Electronic Science and Technology of China, China

## Motivation

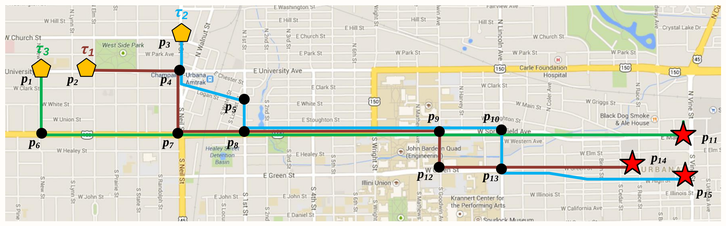
### Research Background and Application Scenarios

Trajectory similarity join (TS-Join) is fundamental functionality: given sets P and Q of trajectories and a similarity threshold $\theta$, the TS-Join returns all pairs of trajectories from P and Q with a similarity that exceeds $\theta$. The TS-Join may bring significant benefits to a range of applications, including trajectory near-duplicate detection, data cleaning, ridesharing recommendation, friend recommendation, frequent trajectory based routing, and traffic congestion prediction.

### Comparison to Existing Studies

| Studies | Space | Temporal matching | Parallel | Data |
|---|---|---|---|---|
| [1] | Euclidean | Sliding-window based | No | 50 K |
| [3] | Euclidean | Sliding-window based | No | 250 K |
| [2] | Euclidean | Time-threshold based | No | 150 K |
| [4] | Euclidean | Time-threshold based | No | 2 K |
| [5] | Euclidean | None (spatial join only) | No | 500 K |
| TS-Join | Network | Continuous matching | Yes | 10 M |

### An Example of the TS-Join




$\tau_1$ = < $p_2$, 09:37 >, < $p_4$, 09:40 >, < $p_7$, 09:48 >, < $p_8$, 09:51 >, < $p_9$, 09:57 >, < $p_{12}$, 10:02 >, < $p_{13}$, 10:05 >, < $p_{14}$, 10:07 >

$\tau_2$ = < $p_3$, 08:35 >, < $p_4$, 08:39> < $p_5$, 08:46 >, < $p_8$, 08:49 >, < $p_9$, 09:01 >, < $p_{10}$, 09:04 >, < $p_{13}$, 09:06 >, < $p_{15}$, 09:07 >

$\tau_3$ = < $p_1$, 09:32 >, < $p_6$, 09:43 >, < $p_7$, 09:48 >, < $p_8$, 09:51 >, < $p_9$, 09:59 >, < $p_{10}$, 10:03 > , < $p_{11}$, 10:13 >

● : sample point in a trajectory ⬟ : start point of a trajectory ★ : destination point of a trajectory

Here, $\tau_1$, $\tau_2$, and $\tau_3$ are trajectories, and $P = \{\tau_1\}$ and $Q = \{\tau_2, \tau_3\}$. In the example, $p_1$, $p_2$, ..., $p_{15}$ are timestamped sample points. Given a time interval (8:30, 10:30), existing sliding-window based trajectory similarity joins (e.g., [1, 3]) return trajectory pairs $(\tau_1, \tau_2)$, and $(\tau_1, \tau_3)$ because they are spatially close to each other. However, $\tau_1$ and $\tau_2$ have very different departure times, thus rendering a result such as this of little use in ridesharing and traffic congestion prediction. The TS-Join returns trajectory pair $(\tau_1, \tau_3)$ without the need for a query time interval.

## Similarity Functions and Problem Definition

### Similarity Functions

The spatial similarity $\mathrm{Sim_S}$, the temporal similarity $\mathrm{Sim_T}$, and the spatiotemporal similarity $\mathrm{Sim_{ST}}$ are defined as follows.

$$\mathrm{Sim_S}(\tau_1, \tau_2) = \frac{1}{|\tau_1|}\sum_{v_i \in \tau_1} e^{-d(v_i.p, \tau_2)} + \frac{1}{|\tau_2|}\sum_{v_j \in \tau_2} e^{-d(v_j.p, \tau_1)} \quad (1)$$

$$\mathrm{Sim_T}(\tau_1, \tau_2) = \frac{1}{|\tau_1|}\sum_{v_i \in \tau_1} e^{-d(v_i.t, \tau_2)} + \frac{1}{|\tau_2|}\sum_{v_j \in \tau_2} e^{-d(v_j.t, \tau_1)} \quad (2)$$

$$\mathrm{Sim_{ST}}(\tau_1, \tau_2) = \lambda \cdot \mathrm{Sim_S}(\tau_1, \tau_2) + (1 - \lambda) \cdot \mathrm{Sim_T}(\tau_1, \tau_2) \quad (3)$$

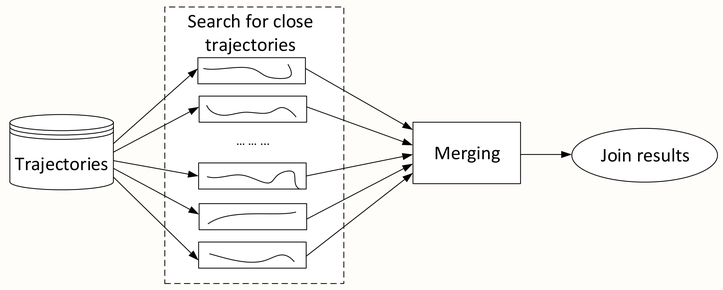
### Problem Definition

Given sets P and Q of trajectories and a threshold $\theta$, the trajectory similarity join (TS-Join) finds a set A of all trajectory pairs from the two sets whose spatiotemporal similarity exceeds $\theta$, i.e., $\forall(\tau_i, \tau_j) \in (P \times Q) \backslash A$ $(\mathrm{Sim_{ST}}(\tau_i, \tau_j) < \theta)$.

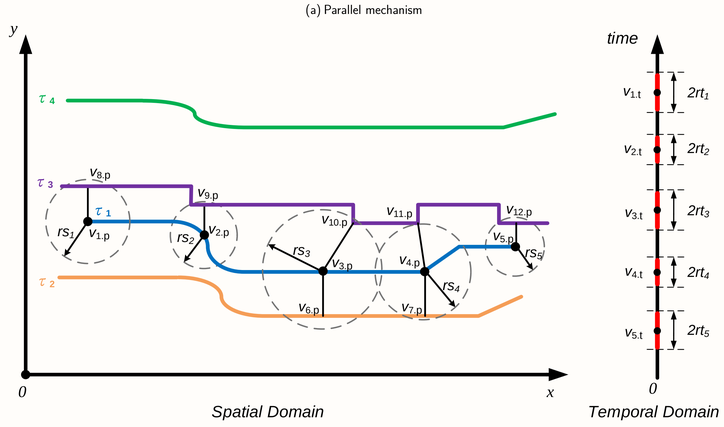
## Two-phase Algorithm

To process the TS-Join more efficiently, we develop a two-phase algorithm based on a divide-and-conquer strategy. (1) In the trajectory-search phase, for each trajectory $\tau \in P$, we explore the spatial and temporal domains concurrently and search for trajectories close to $\tau$. In the spatial domain, network expansion from each trajectory sample point is used to explore the spatial network, while in the temporal domain, we expand the search from each timestamp of $\tau$. An upper bound on the spatiotemporal similarity and a heuristic scheduling strategy are defined to enable pruning of the search space. The search process of different trajectories are independent, so the trajectory searches can be processed in parallel. (2) In the merging phase, we combine the computation results of all trajectories and find the solution to the TS-Join.

### An example of the two-phase algorithm




(a) Parallel mechanism



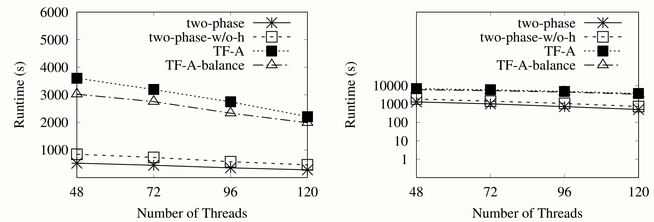

(b) Trajectory search

### Time Complexity

The time complexity of the trajectory search phase is $O(|P||P_\theta|)$. The time complexity for the merging phase is $O(|P||C|)$, where $|C|$ is the cardinality of candidate set for each trajectory. Since $C \subseteq P_\theta \subseteq P$, the time complexity of the two-phase algorithm is $O(|P||P_\theta|) + O(|P||C|) = O(|P||P_\theta|)$. If $\theta$ is sufficiently large, the time complexity is close to $O(|P|)$.

## Experiments

We evaluate the performance of the two-phase algorithm on the New York Road Network (NRN) with 10,000,000 trajectories. For the non-self join, the two-phase algorithm is able to process 10 M × 2 M trajectories with 120 threads in 255 seconds, while for the self join, the two-phase algorithm is able to process 10 M × 10 M trajectories with 120 threads in 540 seconds.




(c) Non-self join




(d) Self join

## Contributions

- We propose a novel network-based trajectory similarity join, called TS-Join, that takes into account both spatial and temporal similarity in a continuous manner, thus targeting applications such as trajectory near-duplicate detection, ridesharing recommendation, route planning, and traffic congestion prediction.
- The TS-Join uses new metrics to evaluate trajectory similarity in the spatial and temporal domains.
- We develop a two-phase algorithm with effective pruning and scheduling techniques that enables parallel TS-Join processing.
- We conduct extensive experiments on large trajectory sets to study the performance of the developed algorithms.

[1] P. Bakalov, M. Hadjieleftheriou, E. J. Keogh, and V. J. Tsotras. Efficient trajectory joins using symbolic representations. In *MDM*, pages 86–93, 2005.

[2] P. Bakalov and V. J. Tsotras. Continuous spatiotemporal trajectory joins. In *GSN*, pages 109–128, 2006.

[3] Y. Chen and J. M. Patel. Design and evaluation of trajectory join algorithms. In *ACM-GIS*, pages 266–275, 2009.

[4] H. Ding, G. Trajcevski, and P. Scheuermann. Efficient similarity join of large sets of moving object trajectories. In *TIME*, pages 79–87, 2008.

[5] N. Ta, G. Li, and J. Feng. Signature-based trajectory similarity join. *IEEE Trans. Knowl. Data Eng.*, online first:1–14, 2017.