Searching Trajectories by Regions of Interest

Shubham mane[1], Shradha Sevada[2], Ajinkya Diwate[3] *Pro.Vaibhav Muddebihalkar[4]*

[1] Shubham Mane, Computer Department
Dr.D.Y Patil Institute of Technology, Maharashtra, India

[2] Shradha Sevada, Computer Department
Dr.D.Y Patil Institute of Technology, Maharashtra, India

[3]Ajinkya Diwate, Computer Department
Dr.D.Y Patil Institute of Technology, Maharashtra, India

[4]Pro.Vaibhav Muddebihalkar, Computer Department

Dr.D.Y.Patil Institute of Technology,Maharashtra,India

ABSTRACT

Search trajectory is used for increasing availability of Moving object tracking data. We propose and investigate a novel query type named trajectory search by regions of interest (TSR query).trajectory search by regions query Gives set of regions of interest provided through parameter and returns trajectory with highest spatial-density correlation to the query regions. This sort of question is helpful in numerous well known applications, for example, trip arranging and suggestion; furthermore, area based administrations all in all. TSR question handling three difficulties: how to show the spatial-density correlation relationship between query region and data Trajectory, how to effectively prune the search space, and how to effectively schedule multiple so-called query sources. The proliferation of mobile devices enables people to log their geographical positions and to trace historical movements, which have spawned various novel applications. An emerging one is trajectory sharing and searching. Different from the conventional similarity search over trajectory databases that uses a sample query trajectory for full matching according to shape or other criteria, new trajectory search applications demand to find trajectories that connect a few selected locations.

Keyword - *Trajectory search by regions, Spatial-density correlation, Spatial networks, Spatial databases*.

INTRODUCTION

Online Map Based Services:

The availability of GPS-equipped devices and online map-based services to catch their present area what's more, to share their directions by methods trajectories by means of services, for example, Bikely4, GPS-Way-points5, Share-My-Routes6, and Microsoft GeoLife7. Likewise, and more social networking sites, including Twitter8, Four square9, and Facebook10, bolster the sharing of trajectories. The accessibility of enormous trajectory data empowers novel portable applications. Such applications may use trajectory search, which discovers trajectory that are comparable in some particular sense to query. This sort of inquiry can profit well known administrations, for example, travel arranging and proposal, and location-based services in general. For instance, when arranging an excursion to various spots in a new city, a traveler may profit by the experience of past guests. Specifically, guests with comparative interests may have gone by close-by points of interest that the client may not know, in any case, might be occupied with. Or, then again others may have maintained a strategic distance from a particular street since it is upsetting, in spite of the fact that it might appear like a decent decision as far as separation. Such encounters are caught in trajectory shared by past guests. In existing examinations, all directions are dealt with the same, paying little mind to their frequencies of utilization. For instance, some less traveled trajectories might be new or quite recently less popularity since the district they are in is less traveled. Such trajectories may even now bear some significance with users.

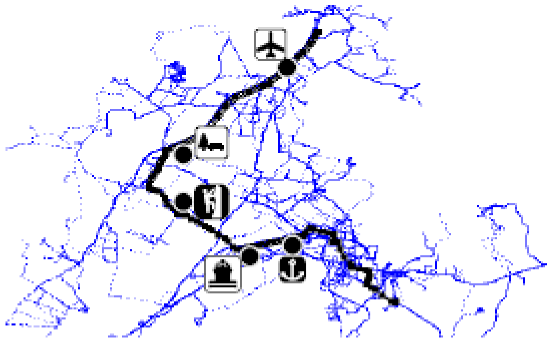


Fig1. Trajectory Search by Location

Background:
Trajectory search queries aim to find trajectories with the highest relevance to query arguments Trajectory similarity functions may contain spatial, temporal, textual, and density elements. The resulting queries are useful in many popular applications including travel planning, carpooling, friend recommendation in social networks, and location-based services in general. We classify the existing trajectory search queries into three categories according to their arguments. In the point-to trajectory category, the query argument is a single spatial point, and the query finds trajectories spatially close to the query point. Query to cover spatial and textual domains and propose the TkSK query, which retrieves the trajectories that are spatially close to the query point and also meet semantic requirements defined by the query.

Motivation:
This is the first study of region-based trajectory search in spatial networks that takes spatial-object density into account. Previous studies use spatial distance as the sole criterion when computing results. However, spatial distance in itself fails to fully capture the relationship between a trajectory and a set of regions.

Objective and Goal:

- To find trajectories by region of interest.
- To improve accuracy of finding trajectories.
- To develop a scheme for users that takes into account both the time and performance.

LITERATURE SURVEY

Sr. No	Paper name	Author name	Advantages and disadvantages	This paper refer to
[1]	Robust and fast similarity search for moving object trajectories	L. Chen, M. T. Ozsu, and V. Oria	1. An important consideration in similarity-based retrieval of moving object trajectories is the definition of a distance function. 2. The existing distance functions are usually sensitive to noise, shifts and scaling of data that commonly occur due to sensor failures, errors in detection techniques, disturbance signals, and different sampling rates. Cleaning data to eliminate these is not always possible. 3. In this paper, we introduce a novel	In order to improve the retrieval efficiency of EDR, we propose three pruning techniques and prove that they do not introduce false dismissals. We also propose different implementation methods of three pruning techniques and test their

			distance function, Edit Distance on Real sequence (EDR) which is robust against these data imperfections. 4. Analysis and comparison of EDR with other popular distance functions, such as Euclidean distance, Dynamic Time Warping (DTW), Edit distance with Real Penalty (ERP), and Longest Common Subsequence's (LCSS), indicate that EDR is more robust than Euclidean distance, DTW and ERP, and it is on average 50% more accurate than LCSS.	efficiency by extensive experimental studies. Most importantly, we show the three pruning methods can be combined to deliver superior retrieval efficiency.
[2]	Searching trajectories by locations: an efficiency study	Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie	1. Trajectory search has long been an attractive and challenging topic which blooms various interesting applications in spatial-temporal databases. 2. In this work, we study a new problem of searching trajectories by locations, in which context the query is only a small set of locations with or without an order specified, while the target is to find the k Best Connected Trajectories (k-BCT) from a database such that the k-BCT best connect the designated locations geographically. 3. Different from the conventional trajectory search that looks for similar trajectories w.r.t. shape or other criteria by using a sample query trajectory, we focus on the goodness of connection provided by a trajectory to the specified query locations. 4. This new query can benefit users in many novel applications such as trip planning.	We study a new problem of searching the k Best-Connected Trajectories from a database by using a set of locations with or without an order constraint. Since the number of query locations is typically small, it enables us to adopt a spatial method for answering a similarity search query. We start the study based on a simple IKNN algorithm and then analyze the efficiency of different variants.
[3]	Index-based most similar trajectory search	E. Frentzos, K. Gratsias, and Y. Theodoridis.	1. The problem of trajectory similarity in moving object databases is a relatively new topic in the spatial and spatiotemporal database literature. 2. Existing work focuses on the spatial notion of similarity ignoring the temporal dimension of trajectories and disregarding the presence of a general-purpose spatiotemporal index. 3. In this work, we address the issue of spatiotemporal trajectory similarity search by defining a	The proposed similarity metric efficiently retrieves spatiotemporally similar trajectories in cases where related work fails, while at the same time the proposed algorithm is shown to be efficient and highly scalable.

			similarity metric, proposing an efficient approximation method to reduce its calculation cost, and developing novel metrics and heuristics to support k-most-similar-trajectory search in spatiotemporal databases exploiting on existing R-tree-like structures that are already found there to support more traditional queries.	
[4]	Adaptive fastest path computation on a road network: A traffic mining approach	H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. Sondag.	1. Efficient fastest path computation in the presence of varying speed conditions on a large scale road network is an essential problem in modern navigation systems. 2. Factors affecting road speed, such as weather, time of day, and vehicle type, need to be considered in order to select fast routes that match current driving conditions. 3. Most existing systems compute fastest paths based on road Euclidean distance and a small set of predefined road speeds. However, “History is often the best teacher”. 4. Historical traffic data or driving patterns are often more useful than the simple Euclidean distance-based computation because people must have good reasons to choose these routes,	We developed an adaptive fastest path algorithm, that bases routing decision on driving and speed patterns mined from historical data. This is a radical departure from traditional algorithms that have focused only on speed and Euclidean distance considerations. The routes computed by our algorithm are not only fast given a set of driving conditions but also reflect observed driving preferences.
[5]	Ecomark: evaluating models of vehicular environmental impact	C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul	1. The reduction of greenhouse gas (GHG) emissions from transportation is essential for achieving politically agreed upon emissions reduction targets that aim to combat global climate change. 2. So-called eco-routing and eco-driving are able to substantially reduce GHG emissions caused by vehicular transportation. 3. To enable these, it is necessary to be able to reliably quantify the emissions of vehicles as they travel in a spatial network	We develop EcoMark that evaluates models of vehicular environmental impact. Eleven state-of-the-art impact models are categorized into instantaneous models and aggregated models. The models are compared and analyzed based on a substantial collection of 1 Hz GPS trajectories and a 3D spatial network. The empirical study suggests that the instantaneous models are appropriate for eco-driving, while the aggregated models are helpful for excoriating. The use of a 3D spatial network that records

				road grades benefits both eco-driving and eco-routing.
[6]	R-trees: a dynamic index structure for spatial searching	A. Guttman	1. In order to handle spatial data efficiently, as required in computer aided design and geo-data applications, a database system needs an index mechanism that help it retrieve data items quickly according to their spatial locations However, traditional indexing methods are not well suited to data objects of non-zero size located multidimensional spaces In this paper we describe a dynamic index structure called an R-tree winch meets this need, and give algorithms for searching and updating it. 2. We present the results of a series of tests which indicate that the structure performs well, and conclude that it is useful for current database systems m spatial applications	The R-tree structure has been shown to be useful for indexing spatial data objects that have non-zero size Nodes corresponding to disk pages of reasonable else (e g 1024 bytes) have values of A4 that produce good performance With smaller nodes the structure should also be effective as a mam memory index, CPU performance would be comparable but there would be no I/O cost
[7]	Exact indexing of dynamic time warping	E. Keogh	1. The problem of indexing time series has attracted much interest. Most algorithms used to index time series utilize the Euclidean distance or some variation thereof. 2. However, it has been forcefully shown that the Euclidean distance is a very brittle distance measure. 3. Dynamic time warping (DTW) is a much more robust distance measure for time series, allowing similar shapes to match even if they are out of phase in the time axis. 4. Because of this flexibility, DTW is widely used in science, medicine, industry and finance. Unfortunately, however, DTW does not obey the triangular inequality and thus has resisted attempts at exact indexing.	There are several directions in which this work may be extended. For example, we note that some algorithms for matching two and three-dimensional shapes are very close analogues of the DTW algorithm and thus may benefit from a similar lower bounding function.
[8]	Shapes based trajectory queries for moving objects	B. Lin and J. Su.	1. An interesting issue in moving objects databases is to find similar trajectories of moving objects. Previous work on this topic focuses on movement patterns of moving objects, rather than spatial shapes of their trajectories. 2. In this paper we propose a simple and effective way to compare spatial shapes of moving object trajectories.	An interesting issue in moving objects databases into find similar trajectories of moving objects. The similarity can be time sensitive or insensitive. In this paper we study the time independent similarity search

			3. We introduce a new distance function based on "one way distance" (OWD).	problem of moving object trajectories.
[9]	Effective map-matching on the most simplified road network	K. Liu, Y. Li, F. He, J. X u, and Z. Ding	1. The effectiveness of map-matching algorithms highly depends on the accuracy and correctness of underlying road networks. In practice, the storage capacity of certain hardware, 2. e.g. mobile devices and embedded systems, is sometimes insufficient to maintain a fat digital map for map matching. Unfortunately, most existing map-matching approaches consider little about this problem. 3. They only apply to environments with information-rich maps, but turns out to be unacceptable for map-matching on simplified road maps	We propose a new offline map-matching algorithm called Passby to match GPS data onto a simplified digital map. The experiment results demonstrate that our Passby algorithm achieves exciting effects compared to the incremental algorithm. Meanwhile, benefiting from the small size of map, simple index structures and heuristic filter strategy, Passby improves matching accuracy as well as efficiency.
[10]	User oriented trajectory search for trip recommendation	S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis	1. Trajectory sharing and searching have received significant attentions in recent years. 2. we propose and investigate a novel problem called User Oriented Trajectory Search (UOTS) for trip recommendation. 3. In contrast to conventional trajectory search by locations (spatial domain only), we consider both spatial and textual domains in the new UOTS query.	we proposed and investigated a novel User Oriented Trajectory Search (UOTS) for trip recommendation. Different from traditional trajectory search by locations (spatial similarity only), in the new UOTS query, both the spatial similarity and user-preference were taken into consideration.

SOFTWARE REQUIREMENT SPECIFICATION

User Classes and Characteristics

To design products that satisfy their target users, a deeper understanding is needed of their user characteristics and product properties in development related to unexpected problems that the user's faces every now and then while developing a project. The study will lead to an interaction model that provides an overview of the interaction between user characters and the classes. It discovers both positive and negative patterns in text documents as higher level features and deploys them over low-level features (terms). In proposed work is designed to implement above software requirement. To implement this design following software requirements and hardware requirements are used.

Software Requirements

- Operating System - Windows XP/7

- Programming Language - Java/J2EE
- Software Version - JDK 1.7 or above
- Tools - Eclipse
- Front End - JSP
- Database - Mysql

Hardware Requirements

- Processor - Pentium IV/Intel I3 core
- Speed - 1.1 GHz
- RAM - 512 MB (min)
- Hard Disk - 20GB
- Keyboard - Standard Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - LED Monitor

IMPLEMENTATION STATUS

To achieve better performance, a best-expansion search (BES) algorithm is proposed. First, we reuse an existing query source selection strategy to select a set of query sources from among the centers of the query regions. Second, we define new upper and lower bounds on the spatial-density correlation to prune the search space. Third, a heuristic search strategy based on priority ranking is developed to coordinate the use of multiple query sources. We maintain and make use of a dynamic priority ranking heap when processing the query. At each point in time, we expand the top-ranked query source until a new query source becomes top ranked. Compared to uniform-speed search, the best- expansion search algorithm has two major advantages: (i) it further prunes the search space for avoiding traversals of overlap regions; (ii) the effective heuristic search strategy focuses on trajectories more likely to be the solution and further improves query performance. Next, it is also possible that a traveler may specify a visiting sequence for intended regions (e.g., c1, c2, and c3 are the intended regions, and the visiting sequence is c1 ! c2 ! c3). The proposed USS and BES algorithms are further extended to process the TSR queries with a sequence efficiently.

COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

The most existing study of trajectory search is, we should set query parameter in sequence of location. Sometimes it will happen the place are not showing location. But may be a region of interest that contains several spatial objects like commercial District, dining area etc. whenever we planning to trip in unfamiliar city user fail to specify intended location exactly and may intended location instead.

Our proposed trajectory search region query takes a set of regions of interest as a parameter and returns the trajectory in the argument set with the highest spatial-density correlation to the query regions. This type of query is useful in many popular applications such as trip planning and recommendation, and location based services in general. TSR query processing faces three challenges: how to model the spatial-density correlation between query regions and data trajectories, how to effectively prune the search space, and how to effectively schedule multiple so-called query sources.

Disadvantages of Existing System

1. Fails to fully capture the relationship between a trajectory and a set of regions.
2. User fails to planning trip in unfamiliar city.
3. User fails to specify intended location exactly.

Advantages of proposed system:

1. This is the first study of region-based trajectory search in spatial networks that takes spatial-object density into account.
2. It further prunes the search space for avoiding traversals of overlap regions.
3. The effective heuristic search strategy focuses on trajectories more likely to be the solution and further improves query performance.

ALGORITHM FOR RELEVANT FEATURE DISCOVERY

- **Euclidean Distance Algorithm:**

1) Add source, destination and all regions in set S
   i.e. S = {s, r, d} where r = {r1, r2,..., rn}
2) for i=1 to m do
{
   Compute distance d(p, q);

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

   add distance d of p&q in set D
}
3) Get set D = {d1,d2,...,dn}
4) End

- **Best-Expansion Search:**

We proceed to introduce a heuristic scheduling strategy based on a priority ranking of the query sources, which is helpful to avoid devoting unnecessary search efforts to trajectories that are unlikely to be the optimal choice.

Data: graph $G(V,E,F,W)$, trajectory set T, query region set C

Result: trajectory τ with the maximum value of $C_{sd}(C,\tau)$

$LB \leftarrow 0$; $UB \leftarrow +\infty$; $T_f \leftarrow null$;
select query sources;
$\forall p_i \in E_c$ $(p.l \leftarrow 0)$;
$p \leftarrow E_c.top()$;
while *true* **do**
  $expand(p)$;
  for *each newly scanned trajectory* τ **do**
    if $\tau.scanned(p_i) = false$ **then**
      $\tau.scanned(p_i) \leftarrow true$;
      if τ *is fully scanned* **then**
        $T_f.add(\tau)$;
        compute $C_{sd}(C,\tau).ub$ and $C_{sd}(C,\tau).lb$;
        update UB and LB;
    if $(LB > UB) \vee (\forall p_i \in E_c(re_i \geq \epsilon + \frac{p.dist)}{2}))$ **then**
      for *each* $\tau \in T_f$ **do**
        if $C_{sd}(C,\tau).ub < LB$ **then**
          $T_f.remove(\tau)$;
    if $p \neq E_c.top()$ **then**
      $p \leftarrow E_c.top()$;
for *each trajectory* $\tau \in T_f$ **do**
  compute $C_{sd}(C,\tau)$;
  if $\max_{\tau \in T_r}\{C_{sd}(C,\tau)\} \geq \max_{\tau' \in T_u}\{C_{sd}(C,\tau').ub\}$ **then**
    return trajectory τ with $\max_{\tau \in T_r}\{C_{sd}(C,\tau)\}$;

SYSTEM ARCHITECTURE

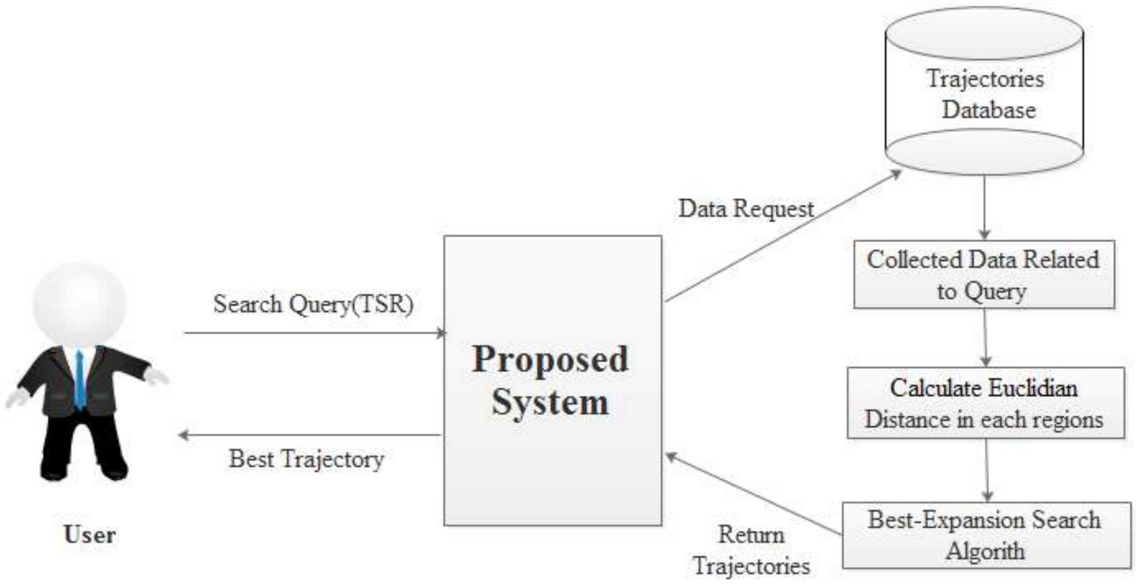


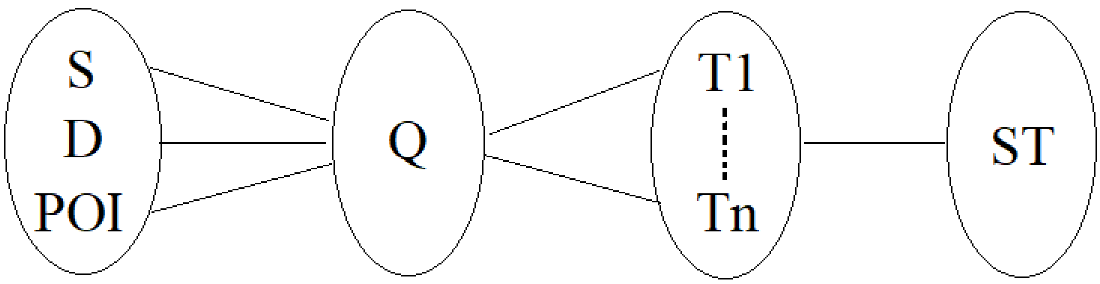
Figure 1: System Architecture

We proposed a novel trajectory search by regions of interest (TSR) query, which is useful in trip planning and recommendation in location-based services in general. We define new measures to evaluate the spatial-density correlation between a trajectory and a set of regions of interest. We develop a best-expansion search (BES) algorithm to compute the TSR query efficiently with the support of upper and lower bounds and heuristic query-source scheduling. We further extend the BES algorithm to scenarios where a sequence of query regions is given. We conduct extensive experiments with real and synthetic spatial data to investigate the performance of the proposed algorithms. In Existing System only find the trajectory with the highest spatial-density correlation to a sequence of query regions, means its search trajectories depends on only user interest, it not compare or find shortest path trajectories by finding distance between two regions by Euclidian distance algorithm. In proposed system we use BES algorithm and Euclidian distance algorithm which is helps to making user friendly application.

MATHEMATICAL MODULE

The following terms shows in detail working of project.

A] Mapping Diagram



Where,

S = User selected source location

D = User selected destination location
POI = User selected multiple point of interest
Q = Search query submitted by user
T1,...,Tn = Multiple trajectories found by consideting S,D and POI
ST = Shortest trajectory by considering user's POI

B] Set Theory

S={s, e, X, F, Y,Φ}

Where,

s = Start of the program.
1. Register with system.
2. Login with web page.

X = Input of the program.
X = {S, D, POI}
Where,
S = User selected source location
D = User selected destination location
POI = User selected multiple point of interest

F = Functions implemented to get the output
F = {ED, BES}
Where,
ED = Euclidian Distance Algorithm
BES = Best-Expansion Search Algorithm

Y = Output of the program.
Shortest trajectories given by system to user by considering user's POI and souce to destination route display on map with regions.

e = End of the program.

- **Space Complexity:**

The space complexity depends on Presentation and visualization of discovered patterns. More the storage of data more is the space complexity.

- **Time Complexity:**
  Check No. of patterns available in the datasets= n
  If (n>1) then retrieving of information can be time consuming. So the time complexity of this algorithm is O(n^n).

  Φ = Failures and Success conditions.
- **Failures:**
  Due to schilling attack recommendation lists can be altered by fake users**.**
- **Success:**
  Fake profile injection can be stopped and we can provide unbiased recommendation to user.

EXPERIMENTAL SET UP AND RESULT TABLE

1. Result Table

TSR Id	Time required to give output(in seconds)
1	2
2	3

3	2
4	2
5	4
6	3

Table 1: Time required executing TSR

Above table shows that the time required to execute each query in seconds. This table is shows that our system gives result in minimum time period as compare to other state of art systems.

2. Result Graph

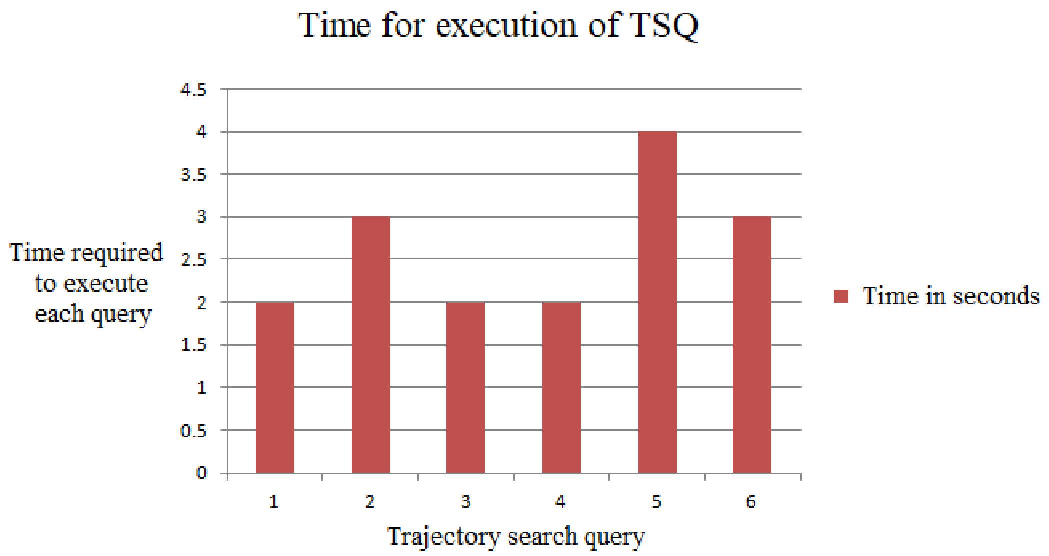


Figure 2: Result Graph of Time for Execution of Each Search Query

The above Figure 2 shows that the result graph of time (in seconds) for execution of each search of proposed system. In Figure 2, x-axis represents trajectory search query id and y-axis represents execution time of each query in seconds.

CONCLUSION

We propose and study a novel problem, namely trajectory search by regions of interest (TSR query), that finds the trajectory with the highest spatial-density correlation to a sequence of query regions. Compared to existing studies of trajectory search by locations, we take the concept of query region and the density of spatial objects into account. This type of query is useful in many popular applications such as trip planning and recommendation, and location based services in general. To compute the TSR query efficiently, we develop a best-expansion search algorithm that exploits upper and lower bounds to prune the search space and adopt a query source selection strategy, as well as a heuristic search strategy based on priority ranking to schedule multiple query sources. The performance of the TSR query was investigated through extensive experiments on both real and synthetic spatial data.

References

[1] L. Chen, M. T. Ozsu, and V. Oria. “Robust and fast similarity search for moving object trajectories”.
[2] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie. “Searching trajectories by locations: an efficiency study”.

[3] E. Frentzos, K. Gratsias, and Y. Theodoridis. “Index-based most similar trajectory search”.
[4] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. Sondag. “Adaptive fastest path computation on a road network: A traffic mining approach”.
[5] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul. “Ecomark: evaluating models of vehicular environmental impact”.
[6] Guttman. R-trees: “a dynamic index structure for spatial searching”.
[7] E. Keogh. “Exact indexing of dynamic time warping”.
[8] B. Lin and J. Su. “Shapes based trajectory queries for moving objects”.
[9] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding. “Effective map-matching on the most simplified road network”.
[10] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis. “User oriented trajectory search for trip recommendation”.