A query optimization method based on the STIR-tree

Wubin Ma, Hongbin Huang*, Su Deng

Information system engineering key laboratory, National University of Defense Technology, Changsha, Hunan, China

*Corresponding author e-mail: hb_huang@nudt.edu.cn

Abstract. In this paper, the method is proposed to enable an query of dynamic resources. Specifically, first a index structure named the STIR-tree(Spatial Temporal Inverted-file R-tree) is proposed into resources locations and so enable an construction of dynamic resources. And a STIR-Tree based neighboring query method is proposed. At last ,experiment results show that the proposed algorithm is effective.

1. Introduction
The resources in IOT have a very strong dynamic, such as vehicles, crowds, planes and even tanks, armored vehicles, etc. Therefore, LBS need to further consider the query and use of dynamic resources. This paper studies a two-state separating model(TTSM) is established based on the characteristics of the trajectory data. According to this model, an index structure named STIR-tree (Spatial Temporal Inverted-file R-tree) based on dynamic resources is proposed, and the nearest-neighbor query algorithm based on STIR-tree is proposed to optimize the query algorithm. Finally, the experimental and structural analysis is given.

2. Related works
In the past ten years, the nearest-neighbor query has become the core of the spatial database and the spatio-temporal database, and this content has been constantly deepened with the development of the times. The most common type of the nearest-neighbor query is the KNN query of points, that is, finding k objects which are closest to point q from the dataset P. The kNN query of mobile objects is a hot topic in recent years, especially the kNN query of mobile trajectories. The current researches are mostly based on the Euclidean distance and cannot well adapt to the requirements of weak connections and frequent movement in the mobile environments. The literature [6] devised a structure for storing the partitioned data objects to represent the graphics model of the road networks and proposed a continuous kNN query algorithm suitable for mobile terminals. Papadias and Kolahdouzan et al. [7] studied the continuous k-nearest neighbor query based on location in spatial networks. Tao and Jensen et al. [8] respectively proposed the method of predicting path based on R-tree and B-tree. The literature [9] introduced the idea and method of transforming the trajectory of object in one-dimensional space into the point in two-dimensional space

3. STIR tree index construction and resource query
The form of the mobile trajectories in space is shown in figure 1. As the time changes, the moving points keep changing, and each trajectory can be represented by a TTSM. The sub-trajectory is mainly represented by the storage dynamic elements, the whole track generally only retains a static element. If the textual description changes, the sequence elements can be added behind the static elements.
Figure 1. The trajectory diagram based on the two-state separating model

According to the construction method of IR tree, the indexes is also constructed based on the trajectory of TTSM. Specially, the difference between this method and the traditional IR tree construction method is that the construction of trajectory index based on TTSM needs to consider the storage optimization and update of the dynamic elements and the problem of storage and update of text inverted index list.

Figure 2. STIR tree index construction

In the STIR tree, the dynamic elements exist at the level of the leaf node and the static information related with the dynamic elements exists in each level of the tree structure, the index structure is shown in figure 4. Separating the dynamic state from the static state can reduce the storage space of the index tree, improve the efficiency of constructing indexes and the query efficiency based on indexes. The update of indexes will be explained in detail later. In the layer of the non-leaf node named the storage area (MBR), the storage contents are that 1. Static information table, it contains the static information sequence tables of all tracks in its space. 2. inverted list, it contains the inverted sequence which is relative to the static information table. 3. The location of the regional space, it contains the location of the space’s range.

The inverted files refer to all the textual description related to the tracks, which are mainly the static textual descriptions and some dynamic descriptions about the trajectories. The construction of the inverted file is shown in table 1.

| Track      | Inverted File 1                      | Inverted File 2                      | Inverted File 3                      |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Tank       | <TR1 Static Inf.text4>, <TR1 Static Inf.text5> | <TR2 Static Inf.text6>, <TR3 Static Inf.text7> |                                            |
| Food       | <TR1 Static Inf.text4>, <TR1 Static Inf.text5> | <TR2 Static Inf.text6>, <TR3 Static Inf.text7> |                                            |

The vocabulary refers to the keywords that describe objects in the inverted files. Each element in the inverted index table (InveFile) contains two sub-elements. The first sub-element represents the
textual name of the vocabulary, the second one represented the times appeared. For example, "TR1.Static_inf.text,4" in the first row of table 1 represents that the vocabulary TR1.Static_inf.text appears four times in the text.

It is noted that the meanings of the inverted indexes between the non-leaf nodes and the leaf nodes are not the same. For the leaf nodes, the elements in the inverted indexes can directly refer to a trajectory, and the inverted index elements of the non-leaf nodes refer to the textual file contained in this area. The textual files contained in the area are generated during the construction of indexes. How to construct the indexes of STIR tree will be explained in detail below.

4. The description of querying problem

There are several types of querying methods based on the mobile trajectories.

Definition 1: The nearest-neighbor trajectory query of fixed points

The nearest-neighbor trajectory query of fixed points refers to that when the query point is fixed, query the trajectories of resources within a fixed time period that is closest and meets the requirement which is similar to the static information description.

Definition 2: The nearest-neighbor trajectory query of mobile points

The nearest-neighbor trajectory query of mobile points refers to that when the querying point is mobile, query the trajectories of resources within a fixed time period that is closest and meets the requirement which is similar to the static information description.

For example, there are two kinds of query points, ie, the mobile query point (mq) and the fixed query point (sq) in figure 2. Their query requirements are the same, are all seeking the trajectories of resources within a fixed time period that is closest and meets the requirement which is similar to the static information description. The query model established by reference [1-2], for a top-k query \( q \).

\( q = \{loc, text, dist, time, style, k\} \), \( q.loc \) and \( q.text \) respectively represent the location information of query object and the description of characteristic information, \( q.dist \) represents the radius of the spatial range required by the query, \( q.time \) represents the query time required. \( q.style \) is the query type, including fixed point and moving point. \( q.k \) represent the number of the query results to be returned.

5. The query optimization method

After establishing the index model, a method to query the system resource of location service based on mobile trajectories is proposed to solving with different query problems. Assuming that the user submits a query, the following text describes how to respond to such queries and how to find the result and return the result to user as quickly as possible.

Firstly, if fewer nodes are visited to find the result in the process of traversing the tree, the average traversal time will be shorter. Therefore, some rules to trim those non-leaf nodes which are not necessary to access need to be defined in the process of traversing the tree, this paper defines the following pruning strategies:

1) Pruning strategy for time mismatch.
2) Pruning strategy for keyword mismatch
3) Pruning strategy for far distance

The first one is the pruning strategy for time mismatch, whose process is that the leaf nodes store the time intervals in which all the child nodes exist, if the query request does not intersect with the time intervals of the nodes, all child nodes under the node are directly pruned out.

The second one is the pruning strategy for keyword mismatch which can refer to the literature and is not explained again.

The last one is the pruning strategy for far distance which need to be explained in detail. Since the query involves the distance between the point and the matrix and the distance between the line segment and the matrix (The point and the line segment refer to the location of the query requester, the matrix refers to the range of the sum of all child nodes’ locations stored at non-leaf nodes), a new method is needed to calculate the distance. Literature [3] studied the distance between line and rectangle and the calculational method was given. Since the system resources of location service in this paper also adopt MBR rectangular strategy, the proposed method in this paper can also be used to
calculate the distance of resource for the nearest-neighbor query of mobile points in this paper. The calculative process is as follows. If there are trajectory T and MBR which are shown in figure 7, determine whether T intersects with the MBR. If they are intersected, the distance between the trajectory and MBR is zero. If not, calculate the average of the shortest distances from these four vertices of MBR to the trajectory to measure the distance from MBR to the line. In addition, measuring the distance from the MBR to the trajectory also take into account the length of the trajectory, the average of the shortest distances from the endpoints of trajectory to MBR is the distance between the trajectory and MBR. Literature [4] also proposed a method to calculate the distance from the MBR to the trajectory. First determine whether they cross, if cross, the distance is zero. If not cross, then divide the space into four quadrants and the center of the quadrant coincides with the center of the MBR, as shown in figure 7, the endpoints of the trajectory are T(point1 and T(point2).

![Figure 3. The representation of the distance between the trajectory and the MBR](image)

Figure 3. The representation of the distance between the trajectory and the MBR

1. If T(point1 and T(point2 are belong to the same quadrant: (a) the minimum distance between the vertices of MBR within the trajectory quadrant and the trajectory, (b) the minimum distance between T(point1 and T, (c) the minimum distance between T(point1 and T.

2. If T(point1 and T(point2 which are adjacent are not belong to the same quadrant, assume T(point1 and T(point2 are respectively belong to quadrant X1 and quadrant X2. (a) The minimum distance between the vertices of MBR in X1 and T. (b) The minimum distance between the vertices of MBR in X2 and T. (c) The minimum distance from MBR. (d) the minimum distance to MBR.

3. If T(point1 and T(point2 which are not adjacent are not belong to the same quadrant, assume T(point1 and T(point2 are respectively belong to quadrant X1 and quadrant X2. The minimum distance from the vertices of MBR which are not in X1 and X2 to T.

The final distance from the trajectory to the MBR is the minimum of these minimum distance which are described above in this method. This method of calculating distance omits the calculation of some points in the MBR. Although the selection of the minimum effective distance reduces the accuracy of measurement, this method has a high computational efficiency, and that is the reason why this method was adopted by many scholars. Therefore, this paper has adopted this method.

The method of average distance is used to calculate the distance of the fixed points in this paper, the specific computational process is as follows. If point is in MBR, the distance is zero. If not, the distance from point to MBR is measured by the minimum distance from point to MBR, represented by \( \text{dist}(\text{MBR}, p) \).

Since the distance is measured by the measurement method based on the minimum distance, it is easily proved that when the MBR does not satisfy the distance requirement of users, all track nodes in the MBR will not satisfy the distance requirement of users. Therefore, all child nodes of the leaf nodes can be trimmed off, and the efficiency of querying the trajectory resource is greatly improved in that way.

6. Experimental analysis

This section uses the real datasets to experiment with the index structure based on STIR tree and the nearest-neighbor query algorithm. Comparing with the query process based on Bck-tree index structure proposed by reference [5], this method compares the time efficiency and the space efficiency.
from the aspect of the index construction and the track query. The dataset used in this paper is the data of the taxis in San Francisco, United States [6], the situation of data is shown in Table 2.

| The total number of tracks | The total number of units of tracks | The total number of entries | The total number of items |
|---------------------------|-----------------------------------|----------------------------|--------------------------|
| 1000                      | 112556                            | 131460                     | 6820                     |

The construction time of indexes is related to the number of trajectories according to the analysis of the construction complexity of indexes.

The construction time of indexes is mainly related to the number of trajectories, so different number of trajectories are used to analyze the construction time of indexes. At the meantime, in order to analyze the advantages and disadvantages of the resource trajectory index method based on the vector two-state model, different number of keywords and returned results are used to compare with the Bck-tree indexing method. The maximum number of nodes allowed for each node in the tree is set to 5, the other parameters are shown in Table 3.

| Parameters       | Values          |
|------------------|------------------|
| The number of results (k) | 1, 2, 4, 8, 16 |
| The number of keywords    | 6, 8, 10, 12, 14 |
| The number of tracks     | 200, 400, 600, 800, 1000 |

The impact of the parameter k on the query time.

After constructing the index tree, we use the T-KNN query method proposed in this paper to conduct the experiment. The detailed process of the experiment is as follows.

First, the experimental parameters are fixed, the number of traces is 1000, the number of querying keywords is 6. According to the change of the parameter k in Table 3, the impact of the time performance of different algorithm is analyzed, and the relationship between the number of results returned and the query time is shown in Table 4.

| The number of results returned (k) | 1 2 4 8 16 |
|-----------------------------------|-----------|
| T-KNN (ms)                        | 148 209 277 306 331 |
| TskSK (ms)                        | 182 241 322 348 373 |

The relationship between the execution time of the query algorithms and the parameter k is shown in figure 4.
According to Figure 4, the time performance of T-kNN method based on the historical trajectories proposed in this paper is better than that of the query method based on Bck-tree index structure proposed in literature [5]. Theoretically, because of the idea of the two-state separation, the comparison made by each query is only carried out in the dynamic information and the static information only provides a sequence index structure, and the comparison of static information caused by the frequent updated dynamic information is avoided in this way. Due to the calculation of the minimum distance and the improvement of pruning efficiency which are described in section 6.2.2, the querying time can be reduced.

7. Conclusion
In this paper, a nearest-neighbor query algorithm based on STIR-tree is proposed and the query algorithm is optimized. This paper queries the past or current location status of mobile object and the location status of the surrounding resource according to the historical trajectory data of mobile objects. Finally, an experimental analysis was conducted to verify the effectiveness of the method.

Acknowledgments
This work was financially supported by Hunan Natural Science Foundation 2018JJ3619 fund.

References
[1] Cao X, Cong G, Jensen C S. Retrieving Topk PrestigeBased Relevant Spatial Web Objects [C]. in: VLDB.2010.
[2] Yang Z, Zheng L, Li M, et al. Matching algorithm for plant protecting unmanned aerial vehicles and plant protecting jobs based on R-tree spatial index[J]. Transactions of the Chinese Society of Agricultural Engineering, 2017, 33(1): 92-98.
[3] D P, J Z, N M, et al. Query processing in spatial network databases [C]. in: Proceedings of the 29th VLDB Conference. Berlin: VLDB Endowment, 2003. 802-813.
[4] R K M, C S. Alternative solutions for continuous K-nearest neighbor queries in spatial network databases[J]. Geoinformatica, 2005,9 (4): 321-341.
[5] Y T, D P, J S. The TPR*-tree: an optimized spatio-temporal access method for predictive queries [C]. in: Proceeding of the 29th VLDB Conference. Berlin: VLDB Endowment, 2003. 790-801.
[6] Jensen C S, D L, Ooi B C. Query and update efficient B+-tree based indexing of moving objects [C]. in: Proceedings of the 30th VLDB Conference. Berlin: VLDB Endowment, 2004. 768-779.
[7] Kong X, Xu Z, Shen G, et al. Urban traffic congestion estimation and prediction based on floating car trajectory data[J]. Future Generation Computer Systems, 2016, 61: 97-107.
[8] Lalwani S, Dixit T, Malhotra G, et al. Optimal Enhancement of Location Aware Spatial Keyword Cover[J]. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), 2017, 6(6): pp: 73-77..
[9] Gotoh Y, Okubo C. A proposition of querying scheme with network Voronoi diagram in bichromatic reverse k-nearest neighbor[J]. International Journal of Pervasive Computing and Communications, 2017, 13(1): 62-75..
[10] Li J, Xia X, Liu X, et al. Probabilistic group nearest neighbor query optimization based on classification using ELM[J]. Neurocomputing, 2018, 277: 21-28...