Resource Index Construction based on Two-state Separating Model

Wubin Ma, Gan Cai-zhao, 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. The development of IOT(Internet of Things) make everything take into the internet. The query of these things in internet is different because they have different characters, including dynamic and static resources . In this paper, the method is proposed to enable an query of dynamic resources. Specifically, first a model of the trajectory of dynamic resources, called the trajectory two-state separating model(TTSM) is constructed. Experiment results show that the proposed algorithm is effective.

1. Introduction
With the development of IOT (Internet of Things), objects can be accessed into the network for users to query and use. The traditional LBS (location-based service) mainly consider the query and use of the static resources, such as Baidu map, Google map and so on.

According to the model, an index structure named STIR-tree (Spatial Temporal Inverted-file R-tree) based on dynamic resources is proposed, the experimental and structural analysis is given.

2. Problem description
First of all, the typical problems of querying mobile objects at home and abroad are given in Table 1.

It can be seen from these questions that the current research on the query of mobile object resources mainly focuses on several types. The first type is to query the location of the mobile object, where it is located, where it passes and where the track is. The second type is to query what mobile resources exist around the mobile resource, what related resources around them, where their positions are and whether they will meet in a certain period of time. The establishment of trajectory model directly affects the query efficiency of these problems.
Table 1. Moving object and related query question

| Moving Object            | Question                                                                 |
|--------------------------|--------------------------------------------------------------------------|
| Person: Politicians,     | When and where did Obama meet Dalai?                                    |
| terrorists, criminals    | Give the mobile track of Abe on a certain day of the year               |
| animals                  | Determine the trajectories of migratory birds, wildlife and so on        |
|                          | Where are the tigers now?                                                |
|                          | Have the movement patterns of the wild herds changed in the past ten years?|
| planes                   | Are there two planes flying to each other (about to crash)?             |
|                          | Which planes that are closest to this plane can cooperate?              |
|                          | Where was the plane at some point in the past?                          |
|                          | Where does the plane cross the airspace? What weapons and equipment around can be used?|
| Cars: taxis, trucks      | Which taxi is nearest to the visitor?                                   |
|                          | Which route does the truck usually travel?                              |
| Military vehicles:       | Which UAV is the closest to a UAV at some point?                        |
| Rockets, missiles,       | What are the speed and direction of the rocket?                         |
| tanks, submarines,       | What is the movement of the warship? What weapons and equipment around can be used?|
| drones and so on         |                                                                          |

The simplest space representation of the trajectories is shown in Figure 1. The x and y coordinates make up the space plane together. The y coordinates represent the time dimension. When the time coordinate is fixed, that is, at a fixed time, every mobile trajectory point has a position in the space plane. With the increase of the time coordinate, the space position can also change constantly. The top dashed line indicates that the cube omits the upward extension. However, the establishment of the trajectory model for a specific environment is complex. A two-state separating model based on the object features is proposed below.

Figure 1. The trajectories of the spatial moving points

3. The trajectory two-state separating model

Combined with the practical application of this paper, the TTSM that the mobile resources have in the system will be proposed below.

The establishment of the trajectory model need to consider the following aspects:

Because of the difference of the environment where the mobile objects are, just like the road traffic network studied in this paper, for example, the cars in the city; However, there may be situations where there is no traffic network, such as tanks in the battlefield. The location service system need to manage these two moving objects in different environments together, so the versatility of the model needs to be considered when modeling the trajectories.

(1) Since the states of mobile objects may change over time, a large amount of the state information may be generated in this process. If the state information is stored in the database, the massive amounts of data that cannot be processed will be brought to the database. Therefore, when modeling trajectory, the optimization problems of trajectory storage should be fully considered in order to use the smallest storage space to store the most trajectories.

(2) In the location service systems, mobile objects may contain a large amount of information described by text and the textual information may change over time. Modeling the trajectories must consider how to store the information described by the text and how to update and maintain them.
Modeling the trajectories must take into account the space complexity of indexing later and the time complexity of querying indexes. Modeling the trajectories must have a trade-off, neither can it be exhaustive, nor can it miss any critical information.

Therefore, this paper proposes the mobile object trajectory model in the location service system based on the above three considerations.

The trajectory two-state separating model: The vector trajectory two-state separating model is represented by a two-element relationship pattern $TR = \{\text{Static}_{\text{Inf}}, \text{Dynamic}_{\text{Inf}}\}$ in which Static $\text{Inf}$ stores the static information sequence of the mobile objects $\text{Static}_{\text{Inf}} = (\text{sta}_{\text{id}}, \text{object}_{\text{id}}, \text{name}, \text{style}, \text{text})$. $\text{sta}_{\text{id}}$ represents the id of the static information sequence, $\text{object}_{\text{id}}$ represents the number of the object in the sequence, $\text{style}$ represents the category of network where the object is, $\text{text}$ represents the textual description of the object. In general, it does not change with the changes of the trajectory points and need not to update. However, the description may change in some cases. This paper uses the value-added sequence to store the static information, that is, Static $\text{Inf}$ is about to expand into the dynamic collection value-added sequence mode. When the value of $\text{text}$ changes, a new static information tuple will be added to the original sequence. Dynamic $\text{Inf}$ stores a units sequence, $\text{Dynamic}_{\text{Inf}} = \{p_1, p_2, \ldots\}$, in which each dynamic element represents a sub-trajectory, represented by a six-tuple model: $p_i = (\text{sta}_{\text{id}}, \pos, \text{function}, \text{time}, \text{route}, \text{speed})$. $\text{sta}_{\text{id}}$ represents the number of the static message sequence. $\pos$ represents the current location of the dynamic element. $\text{function}$ represents the function of the object’s trajectories in a certain period of time, which is related to the mobile characteristics of each trajectory and the network type of the static information. $\text{time}$ represents the effective time period of the trajectory model, $\text{time} = (\text{starttime}, \text{endtime})$. $\text{route}$ represents a line space object, when the type of the route is the road network, it describes the geometric properties of the movement of objects in the road network which means the route of moving. $\text{speed}$ represents the vector moving speed of objects.

This model uses an event-based approach. When the dynamic information changes, the six-tuple is updated and the new dynamic elements generated are stored in the dynamic information. Since $\text{text}$ itself contains a large amount of data and the frequency of updating is generally small, so it is put into the static information tuples. If it is put into the dynamic information, the textual description properties will generate a lot of unnecessary redundant storage as the dynamic information is frequently updated.

The graphical representation of the trajectory model is shown in Figure 2.

![Figure 2. The block representation of the trajectory](image)

What needs to be clarified here is that the textual storage of the two-state separating model takes the form of a text index table in order to further reduce the storage space. When constructing the indexes of resources, the construction of text mainly uses inverted indexes, the number is used to represent the text and a special index table is used to record the relationship between the number and the text. The storage method of this model will not only save a lot of storage space, but also can save the querying time. This is because the index construction and traversal query can also be completed by the way of digital code matching and the matching complexity of the digital code is obviously lower than the text matching.
Table 2. The separating algorithm introducing the TTSM

**Algorithm 2**: Split(node, newnode)

Input: tree node, new node  
Output: null

1: If node is not the left node then  
2: Use the method in this article 4.1, group the children of node, assumed to be A, B;  
3: Point the pointers of node and newnode to A and B, respectively;  
4: update static information sequence, dynamic information sequence and inverted list of node and newnode;  
5: If the father of node (fath_node) is not full  
6: Point the sub-node of fath_node to newnode;  
7: update static information sequence, dynamic information sequence and inverted list of fath_node;  
8: Else add newnode into fath_node, call the splitting function;  
9: update static information sequence, dynamic information sequence and inverted list of fath_node;  
10: EndIf;  
11: EndIf;  
12: If node is a leaf node then  
13: According to the method in this article 4.1, combined with the vectorical information of space tracks, group the space elements contained by node, assumed to be A, B;  
14: Point the pointers of node and newnode to A, B;  
15: update static information sequence, dynamic information sequence and inverted list of node and newnode;  
16: If the father of node is not full  
17: point the sub-node of fath_node to newnode;  
18: update static information sequence, dynamic information sequence and inverted list of fath_node;  
19: Else add newnode into fath_node, call Split(fath_node, newnode)  
20: update static information sequence, dynamic information sequence and inverted list of fath_node;  
21: EndIf;  
22: EndIf;

### 4. 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 Bek-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 3.
Table 3. The situation of the selected moving dataset

| 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 relationship between the construction time of indexes and the number of trajectories is shown in figure 3.

![Figure 3](image_url)

**Figure 3.** The relationship between the construction time and the number of trajectories. The construction time of indexes is related to the number of trajectories according to the analysis of the construction complexity of indexes.

iment also verifies the theoretical correctness of the reference’s the deduction of time complexity.

5. The research status

For the general textual and numeric data, researchers have proposed a variety of index structures, including Indexed Sequential Access Method (ISAM), index structures based on tree (such as B tree, B+ tree, B* tree, R tree, etc.) and index structure based on hash table (such as external static hash table, dynamic hash table, scalable hash table and linear hash table, etc.). The current high-dimensional index is mainly evolved from four kinds of indexing structure [7] which are KD-Tree, B-Tree, Hashing and Space Filling Curve. Among them, KDB-Tree [8] evolved from KD-Tree and R-Tree [9] evolved from B-Tree are widely used. KDB-Tree is a combination of KD-Tree and B-Tree, KD-Tree is derived from the extension of high-dimensional data by binary search tree, but its fan-out is too small, resulting in the tree being too tall. KDB-Tree combines B-Tree with KD-Tree and adds the number of the nodes’ fan-out. KDB-Tree is an index that represents the space partitioning, the method performs well on uniform data, but performs poorly on uneven data. The KDB-Tree has the feature of splitting down, resulting in excessive number of nodes which reduces the physical utilization.

6. Conclusion

In this paper, aiming at the dynamic resources in the system resources of location service, a trajectory model separating the dynamic information from the static information is proposed. Based on this model, a two-state separating index construction of STIR-tree based on historical trajectories is proposed. a nearest-neighbor query algorithm based on STIR-tree is proposed an experimental analysis was conducted to verify the effectiveness of the method.

However, the future location of the resources cannot be queried in the query of dynamic resources. In order to solve this problem, the next paper will dig out useful knowledge from a large number of historical trajectories to solve the index query problem of the location of future resources.

Acknowledgments

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

References

[1] CHEN, Xu, et al. HIB-tree: an efficient index method for the big data analytics of large-scale human activity trajectories[J].Future Generation Computer Systems, 2018.
[2] Zheng K, Zheng B, Xu J, et al. Popularity-aware spatial keyword search on activity trajectories[J]. World Wide Web, 2017, 20(4): 749-773.

[3] Wang S, Bao Z, Culpepper S, et al. Reverse k Nearest Neighbor Search over Trajectories[J]. IEEE Transactions on Knowledge and Data Engineering, 2017.

[4] JIN, Xiongnan, et al. Collective Keyword Query on a Spatial Knowledge Base[J]. IEEE Transactions on Knowledge and Data Engineering, 2018.

[5] Piorkowski M, Sarafijanovoc-Djukic N, Grossglauser M. A Parsimonious Model of Mobile Partitioned Networks with Clustering[M]. Bangalore, India: 2009.

[6] Chen N, Zhong X, Li L. Research on optimized R-Tree high-dimensional indexing method based on video features[C]//Cloud Computing and Big Data Analysis (ICCCBDA), 2017 IEEE 2nd International Conference on. IEEE, 2017: 93-97.

[7] T R J. The KDB-tree:A search structure for large multidimensional dynamic indexes [C]. in: Proc of ACM SIGMOD 1981. New York: ACM,1981. 10-18.