# 3DR-tree Model Improvement Based on Enhance of Index Performance

Zhang Zhi Tong[1, a]

[1]Faculty of Technology, Harbin University, China

[a]hrbuzzt@126.com

**Keywords:**spatio-temporal database; 3DR-tree, index; node splitting; tree splitting

**Abstract.** 3DR-tree is an index method using the traditional R-tree to index moving objects. Its defect is that a cube generates when an object remains still for a period of time. For those objects that remain still for a long period of time, many strip cubes will generate. MBR will be overlong or overlarge, thus increasing a great deal of overlap and reducing index performance greatly. This paper is to improve the 3DR-tree model to enhance index performance. On the basis of 3DR-tree, this paper will put forward improving historical data index performance of 3DR-tree through node splitting. 3DR-tree is provided with the online data index function by means of tree splitting.

## Introduction

3DR-tree [1] is an index method using the traditional R-tree [2] to index moving objects. When a 3DR-tree index is established, the left and right end values in each dimensional interval of each object must be known. Therefore, 3DR-tree processes only offline data, i.e. data items with defined time interval values. It cannot process line segments with an open time interval.

In the spatio-temporal database, tense attributes of moving objects are divided into historical tense, online tense and future tense. Therefore, when researching index structure, we correspondingly divide index into moving object historical information index, moving object current information index and moving object future information index. 3DR-tree is reconstructed based on this classification. In this paper, we use a “triple model” [3] to indicate the evolution of a spatio-temporal object.

## Node Splitting

3DR-tree regards time as another spatial dimension information of a spatial object, and then utilizes R-tree to perform spatial index. It uses three-dimensional spatial data to indicate a spatio-temporal object (two-dimensional), which is comparatively intuitional and simple. For example, in the two-dimensional space, a spatial object is represented by the minimum bounding rectangle (MBR). Accordingly, a spatio-temporal object can be represented by the MBR in the three-dimensional space. The bottom of the three-dimensional MBR is the MBR of the two-dimensional space, and the height of the three-dimensional MBR is the lifecycle of the spatio-temporal object.

Considering the node form of 3DR-tree, enhancing query effectiveness can start with reducing the volume of the minimum circumscribed cube.

First, load the splitting process manually in the evolution of a spatio-temporal object. Assume that a spatio-temporal object evolves to the moment t2 from the moment t1. Load a splitting at the moment ts. Where, ti<ts<t2. Delete the object manually at the moment ts, and then insert two new objects with the same spatial status. These two objects have the same ID as the original object. As a result, the original spatio-temporal object o[t1，t2] is replaced by two new spatio-temporal objects o[t1，ts] and o[ts，t2]. The number of spatio-temporal objects increases, but the total MBR invalid space reduces, i.e. the volume of the minimum circumscribed cube is reduced by segmentation. Figure 3-1 shows that an interval object in a three-dimensional space operates to the moment t2 from the moment t1. Introduce a manual splitting at the midpoint ts between t1 and t2. It can be seen that the original MBR forms a large invalid space. Compare the MBRs of the two newly introduced spatio-temporal objects with the MBR of the original spatio-temporal object, and it is seen that the invalid spaces E1 and E2 are saved. Therefore, the volume of circumscribed cube

reduces by about 1/2 provided that the number of spatial objects is doubled.

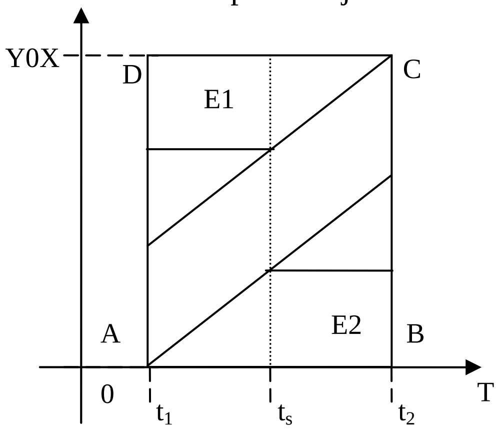

Fig.1 MBR projecting in Y0X when splitting

Second, study the effect of the data set density and the number of spatial objects on index performance. We will use the R-tree forecast model [4] put forward by YANNIS THEODORIDIS to estimate its index performance.

Assume that f is the capacity of each node of R-tree, the query window Q = (qx, qy, qz), and D is the density of the data object set that indicates the number of objects (represented by their MBRs) at a random point of a given space. The number of objects is expressed as m. For the query Q, the number of times of I/O required is about:

$$DA(q) = \frac{m}{f}\left(\left(\frac{D_1 f}{m}\right)^{1/3} + q_x\right)\left(\left(\frac{D_1 f}{m}\right)^{1/3} + q_y\right)\left(\left(\frac{D_1 f}{m}\right)^{1/3} + q_z\right)$$

$$D_1 = \left(1 + \frac{D^{1/3} - 1}{f^{1/3}}\right)^3 。 \qquad (1)$$

In the definition, the size of each dimension of the space is [0，1). If qi = 0, then DA(q) = Dl. It can be seen that reducing the density of large and small sets can reduce I/O operations. If the query window is small, the increase of the number of objects has little effect on I/O operations.

According to the above-mentioned analysis of the R-tree forecast model, the effect of the number of spatial objects on I/O operations is lower than that of the data set density obviously.Therefore, introducing the splitting operation will enhance index performance.

Because leaf nodes do not cause splitting of bottom-layer nodes and due to the efficiency reason, leaf nodes are not affected by recursive splitting. However, the splitting operation on other nonleaf nodes may spread downward.The specific splitting process can be expressed as the following algorithm:

```
BOOL Request_Spliting(n)
If ( n.leaf <= 2 )
//Judge whether a node is a leaf node
Return FALSE
Else
Return TRUE
for (I = 0, j = 1; I <= count(n.leaf);i++, j++)
temp=max(tj-ti)
  //Select the node with the longest time interval from all nonleaf nodes
ts=(t1+t2)/2
//Calculate corresponding time splitting points
```

The process split(n) divides the strip e into two parts along the time axis, i.e. e(o_id, s, [t1, t2]) is divided into e1(o_id, s, [t1, ts]) and e2(o_id, s, [ts, t2]). This process splits a strip cube into a series of short cubes along the time axis, which reduces the zone area, overlapping area and perimeter and thus enhances index performance of 3DR-tree. Figure 2 shows this process. Considering more details, the time area is elaborated. Therefore, the area of the minimum circumscribed rectangle that moving objects account for in the YOX plane reduces obviously, as shown in the figure. This proves that index efficiency can be effectively enhanced by using the node splitting method

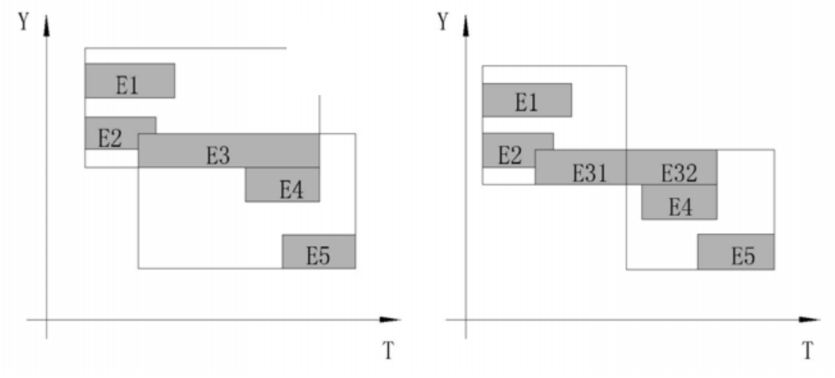

Fig.2 spliting of square along T-axis

## Tree Splitting

In history tree, existence of large amount of blank space influences searching efficiency, e.g. searching section may intersect with node's MBR but without actual object exists in the intersection, or visiting unwanted nodes[4]. In the case of spatial-temporal data processed in 3DR-tree, the time dimension continuously expends upwardly along with increasing update time t in spatial-temporal object data, which induces a large proportion of MBR is occupied by blank space. Therefore, it is considered to take blank space as an inserted cost function in order to control the increasing blank space of nodes in 3DR-tree structure algorithm. This modification has the advantages of more compacted nodes, a reducing possibility of intersection between search space and blank space of node MBR, and an improved searching efficiency.

Two methods are used to expand 3DR-tree so that 3DR-tree can index online spatio-temporal data.In the first scheme, the minimum circumscribed cube (MBR) of an R-tree node grows with time. That is, if a node includes online spatio-temporal data items, its circumscribed cube grows with time linearly. Figure 3 shows this process.

The rectangle R as shown in the figure above includes two data items o1 and o2. o1 is closed while o2 is unclosed and grows with time. The right endpoint of the time dimension of o2 is unclosed, i.e. o2 is online data. Therefore, the minimum circumscribed rectangle of o1 and o2 is also unclosed, and the right endpoint of the time dimension remains growing. If we want to insert data into index and use index to query, all unclosed right time endpoints must be set to the operation execution time. In this way, all unclosed hypercubes can be considered closed, and the R-tree algorithm can be used. RST-tree is a good example of this aspect, but it is a dual-tense index. The defect of this method is that even though there are only a few online data in the index, the circumscribed cubes of nodes will still result in invalid space and space overlapping, thus causing inefficiency of index performance. In addition, this method has an excessive insertion cost.

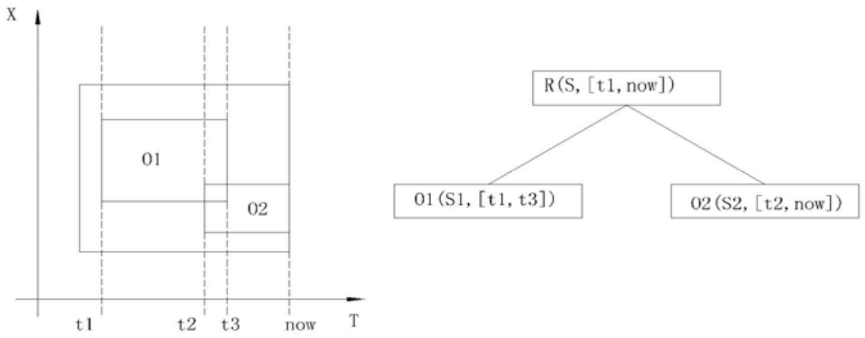

Fig.3 MBR extending with time

The other method is to separately store online data and historical data[5]. Two trees are used to index all data. The first tree indexes online data and mainly inspects spatial attributes of data with tense information as auxiliary information. The second tree uses improved 3DR-tree to index historical data. When any spatio-temporal data become historical data, the time interval attribute is changed to [ti, tj] from [ti, now], and data move from tree I to tree II. This method is efficient for spatio-temporal data with a long evolution period.

## Conclusion

It regards time as another dimension of the space and is applicable to spatio-temporal objects whose position and range do not or slightly change. However, it does not take into account the particularity of the time dimension and processes only offline data. In addition, many strip cubes will generate for those objects that remain still for a long period of time, thus reducing index performance greatly.Therefore, we use the combination of node splitting and tree splitting to design and expand the 3DR-tree index structure. In addition, historical data index and online data index are separately established on a uniform 3DR-tree according to different characteristics of historical spatio-temporal data and active spatio-temporal data, thus realizing the spatio-temporal index mechanism for effective indexing of both historical data and online data and breaking through the limitation that the original 3DR-tree only index historical data.

## References

[1] TAO Y, PAPADIAS D. MV3R-tree: A Spatio-Temporal Access Method for Timestamp and Interval Queries[C]. Proceedings of the 27th International Conference on Very Large Databases, San Francisco, 2001: 431-440.

[2] YUFEITAO, PAPADIAS DIMITRIS, SUN JIMENG. The TPR*-tree: An Optimized Spatio-Temporal Access Method for Predictive Queries[C]. Proceedings of the 29thVLDB Conference, Berlin, 2006: 790-801.

[3] CHON D, AGRAWAL D, ABBADI A E. Storage and Retrieval of Moving Objects[C]. In Proc.of the Intl.Conf. on Mobile Data Management, Hong Kong, China, 2007: 173-184.

[4] CAIAND M, REVESZ P. Parametric R-Tree: An Index Structure for Moving Objects[C]. In Proc.of the Intl.Conf. on Management of Data, Pune, India, 2005: 57-64.

[5] PEUQUET D, DUAN N. An Event-Based Spatiotemporal Data Model(ESTDM) for Temporal Analysis of Geographical Data[J]. International Journal of Geographical Information Systems, 1995, 9(1): 7-24.