ISPRS Journal of Photogrammetry and Remote Sensing 78 (2013) 116–128

Contents lists available at SciVerse ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Parallel indexing technique for spatio-temporal data

Zhenwen He [a,b,*], Menno-Jan Kraak [b], Otto Huisman [b], Xiaogang Ma [b], Jing Xiao [b]

[a] School of Computer, China University of Geosciences (Wuhan), 388 Lumo Road, Wuhan 430074, China
[b] Faculty of Geo-Information Science and Earth Observation, University of Twente, Hengelosestraat 99, 7514 AE Enschede, Netherlands

ARTICLE INFO

Article history:
Received 22 June 2012
Received in revised form 24 January 2013
Accepted 25 January 2013
Available online 6 March 2013

Keywords:
Spatio-temporal index
Parallel index
R-Tree
Interval

ABSTRACT

The requirements for efficient access and management of massive multi-dimensional spatio-temporal data in geographical information system and its applications are well recognized and researched. The most popular spatio-temporal access method is the R-Tree and its variants. However, it is difficult to use them for parallel access to multi-dimensional spatio-temporal data because R-Trees, and variants thereof, are in hierarchical structures which have severe overlapping problems in high dimensional space. We extended a two-dimensional interval space representation of intervals to a multi-dimensional parallel space, and present a set of formulae to transform spatio-temporal queries into parallel interval set operations. This transformation reduces problems of multi-dimensional object relationships to simpler two-dimensional spatial intersection problems. Experimental results show that the new parallel approach presented in this paper has superior range query performance than R*-trees for handling multi-dimensional spatio-temporal data and multi-dimensional interval data. When the number of CPU cores is larger than that of the space dimensions, the insertion performance of this new approach is also superior to R*-trees. The proposed approach provides a potential parallel indexing solution for fast data retrieval of massive four-dimensional or higher dimensional spatio-temporal data.

1. Introduction

The world is three-dimensional and time is always going forward. Essentially, the real world is a four-dimensional spatio-temporal integrated space. Each object in the world has a presence in both time and space. The temporal dimension marks the lifecycle of an object in terms of its starting and ending times, and also establishes the validity of data. Data valid today may have had no meaning in the past, and similarly, may hold no identity in the future. Location represents the *x*, *y*, *z*, location of an object, which may change over time. This relationship with time adds a temporal identity to most data in a spatial information system. However, the time-varying data with spatial locations is difficult to manage effectively and efficiently. Most current spatial database systems represent a single state of the spatial data and this is most commonly assumed to be its current state. Any modifications normally result in the overwriting of the data with the old data being discarded. Although commercial database management systems offer some capabilities to keep track of old and discarded data, this is solely for the purpose of database recovery and not to retain the previous state of the data.

In the last decades, research on spatio-temporal databases has advanced in various aspects and reported many important results, however, many challenges still remain (Lin, 2012; Ni and Ravishankar, 2007; Stantic et al., 2010). In most previous studies, core concepts of temporal index methods and spatial index methods have been established, but it is not yet shown how they can be applied to manage spatio-temporal data efficiently. Generally, spatio-temporal databases are append-only and usually very large in size. Hence, an efficient access method for spatio-temporal databases is even more important than for conventional databases. Numerous indexing methods have been proposed, however, most of them are usually customized either specifically for temporal data or for spatial data, and most of them cannot adequately handle integrated spatio-temporal data (Ang and Tan, 1995; Berchtold et al., 1996; Ciaccia et al., 1997; Curtis and Michael, 1991; Eo et al., 2006; Ibrahim and Christos, 1993; Kim et al., 2007; Lin, 2008, 2012; Saltenis and Jensen, 2002; Stantic et al., 2010; Zhu et al., 2007). This is because indexing methods often divide the spatio-temporal index into two parts, temporal index and spatial index. This particular constraint makes these access methods unsuitable for many practical spatio-temporal applications.

Many multi-dimensional index methods have been proposed. Some of them have been recommended for handling temporal data (Betty and Vassilis, 1999; Kumar et al., 1998; Stantic et al., 2010), and some of them have been reported for handling spatial data

* Corresponding author at: School of Computer, China University of Geosciences (Wuhan), 388 Lumo Road, Wuhan 430074, China.
E-mail address: zwhe@foxmail.com (Z. He).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2013.01.014

effectively, such as R-Tree (Guttman, 1984), R*-Tree (Norbert et al., 1990), Segment R-Tree (Curtis and Michael, 1991), packing R-Tree (Ibrahim and Christos, 1993), X-tree (Berchtold et al., 1996), M-tree (Ciaccia et al., 1997), 3D R-Tree (Zhu et al., 2007) and some other R-Tree variants (Balasubramanian and Sugumaran, 2012). Most of the spatio-temporal index methods are extensions of data partitioning spatial index methods, for instance, X-Tree and Segment R-Tree. The multi-dimensional R-Tree and its variants can be used as spatio-temporal index structures in theory. These use a spatio-temporal containment hierarchy that clusters data into bounding regions of the parent node, forming a hierarchical tree structure. These bounding regions may not represent the entire data scope and could overlap. Overlapping is a problem for data partitioning index methods because even for a simple point query it may need to examine multiple paths. Especially when they are used in now-relative temporal data and movement data, significant overlapping between nodes and dead space will cause very poor performance of the index (Lin, 2012; Saltenis and Jensen, 2002). In order to solve these problems caused by moving objects data, some spatio-temporal indexing methods based on R*-trees (Norbert et al., 1990) were proposed, such as TPR-tree (Choi et al., 2004; Lin and Su, 2004; Saltenis et al., 2000), TPR*-tree (Kim et al., 2010; Tao et al., 2003), RTR-tree and TP^{2}R-tree (Jensen et al., 2009). These indexing methods mainly improve the continuous update performance of R*-trees and R-trees via the time parameter and velocity parameter. Consequently, they only fit moving objects data, mainly the trajectory data, but not more general spatio-temporal data, such as fields. If we want to handle general spatio-temporal data, we should go back to R-trees and R*-trees, since they are basic data structures for multi-dimensional range query.

Parallel computing is a growing trend in handling massive spatio-temporal datasets, especially the high-dimensional data. Nevertheless, R-Trees are not suitable for applications in high dimensional spaces, because their structures are not suitable for parallel computing. Parallel computing is a form of computation in which many calculations are carried out simultaneously. At present, computer capabilities have increased due to the application of parallelism, for example in multicore processors (Asanovic et al., 2006; Knysh and Kureichik, 2010). However, most of the spatial or temporal indexing methods mentioned previously are implemented as traditional sequential algorithms. Some of these can only handle temporal data or interval data effectively, some can only handle spatial data effectively, and others can be used as general spatio-temporal indexes (such as R-Trees, R*-trees, and some of their variants). But all are difficult to implement as parallel algorithms.

We intend to propose a parallel access method for general spatio-temporal data. The fundamental idea behind this approach is to transform both temporal data and spatial data into multi-dimensional intervals, and then deploy a parallel access method for the multi-dimensional intervals. A number of access methods for interval data have been proposed, including the Interval B-Tree (Ang and Tan, 1995), RI-Tree (Brochhaus et al., 2005), TD-Tree (Stantic et al., 2010) and compressed B$^+$-Tree (Lin, 2008, 2012). The compressed B$^+$-Tree mainly focuses on movement data. Stantic et al. (2010) reported that TD-Tree is superior to RI-Tree which in turn is superior to Window-Lists (Ramaswamy et al., 1997), the Oracle tile index and IST-techniques. The TD-Tree is aimed at one- dimensional time interval data. As a result, it cannot be used in four-dimensional spatio-temporal data directly, although it is efficient for some certain temporal queries. Nevertheless, its triangular decomposition strategy is a good idea. The methods proposed in this paper represent extensions of this approach. Specifically, we will use it as one of parallel branches in our algorithm.

In this paper we present an integrated parallel index method, the "Multi-dimensional Parallel Binary Tree" (MPB-Tree) for general four-dimensional spatio-temporal data. In contrast to previously proposed access methods for intervals or R-Tree variants, this method can efficiently answer a wide range of multi-dimensional spatio-temporal query types using a uniform algorithm. The MPB-Tree is a parallel space partition access method. The basic idea is to manage four-dimensional or multi-dimensional interval using multi-dimensional parallel binary trees. This multi-dimensional index structure relies upon multiple parallel two-dimensional representations of intervals. The MPB-Tree partitions space using a parallel triangular decomposition method extending from the TD-Tree's triangular decomposition strategy. The resulting multi-dimensional parallel binary trees store a bounded number of intervals. The empirical performance of the MPB-Tree is demonstrated to be superior to the R*-tree for general four-dimensional spatio-temporal data.

The remainder of this paper is organized as follows: In the following section, we describe the representation of spatio-temporal data and interval data. In Section 3, we describe the parallel indexing algorithms for patio-temporal data. Section 4 documents the algorithm, experiment and analysis of results using the improved method. Finally we present our conclusion in Section 5.

2. Representation of spatio-temporal data and intervals

We assume a four-dimensional spatio-temporal model in the range [*TMin*, *XMin*, *YMin*, *ZMin*, *TMax*, *XMax*, *YMax*, *ZMax*], for some *TMax* > *TMin*, *XMax* > *XMin*, *YMax* > *YMin* and *ZMax* > *ZMin*. In the four-dimensional space, each point can be described as (*t, x, y, z*) and each spatio-temporal object has a minimum bounding rectangle (MBR) in the range [*TOMin*, *XOMin*, *YOMin*, *ZOMin*, *TOMax*, *XOMax*, *YOMax*, *ZOMax*]. We employ this four-dimensional MBR to represent each spatio-temporal object, like the R-Trees (Guttman, 1984; Nievergelt et al., 1984; Norbert et al., 1990). We can transform this four-dimensional MBR into four intervals: [*TOMin, TOMax*], [*XOMin, XOMax*], [*YOMin, YOMax*] and [*ZOMin, ZOMax*]. For example, there is a building whose spatial MBR is in the range [100,200] in *X* axis, the range [150,400] in *Y* axis, and the range [300,700] in *Z* axis. This building was built in 1900, and it was destroyed in 1978. Then this building may be represented by four intervals: [1900,1978], [100,200], [150,400] and [300,700]. In general, we can describe them in a uniform:

$$[Is(d), Ie(d)],\ IMin \leqslant Is(d) \leqslant Ie(d) \leqslant IMax,$$

in which *IMax* is the maximum value of the intervals, *IMin* is the minimum value of the intervals, *Is* is the starting of interval, *Ie* is the ending of interval, and $d = 0,1,\ldots,n-1$. This representation can be extended to *n*-dimensional space, since in four-dimensional space, $n = 4$. For each dimension of this space, an interval can be described as a point in a two-dimensional plane (Stantic et al., 2010) as shown in Fig. 1. The axis *S* is the starting value of the interval, and axis *E* is the ending value of the interval. Because *Ie* is always larger than *Is*, the triangle of (*IMin, IMin*), (*IMax, IMin*), (*IMax, IMax*) is always empty in the two-dimensional space. If the dimension is four, there will be four parallel two-dimensional spaces. This offers the possibility of handling the intervals in four-dimensional space using a parallel method extended from the two-dimensional space, and the possibility for reducing query computation by skipping the empty triangular regions.

For general intervals, Allen (1983) described 13 distinct interval algebra relationships that may hold between two intervals, as shown in Fig. 2. Each of the 13 interval algebra relationships can be represented as a point, line and region (Figs. 3 and 4) in the two-dimensional space, or the plane *SE*. If the universal interval of any one dimension is [*Is, Ie*], and the input interval of a query