Multidimensional Access Methods: Trees Have Grown Everywhere

Timos Sellis
Computer Science Division
Dept. of Electrical and Computer Engin.
National Tech. Univ. of Athens
Zografou 15773, Greece
timos@dbnet.ece.ntua.gr

Nick Roussopoulos
Computer Science Dept.
University of Maryland
College Park, MD 20742, USA
nick@cs.umd.edu

Christos Faloutsos
Computer Science Dept.
University of Maryland
College Park, MD 20742, USA
christos@cs.umd.edu

1 Extended Abstract

This paper is a survey of work and issues on multi-dimensional search trees. We provide a classification of such methods, we describe the related algorithms, we present performance analysis efforts, and finally outline future research directions.

Multi-dimensional search trees and Spatial Access Methods, in general, are designed to handle spatial objects, like points, line segments, polygons, polyhedra etc. The goal is to support spatial queries, such as nearest neighbors queries (*find all cities within 10 miles from Washington D.C.*), or range queries (*find all the lakes on earth, within 30 and 40 degrees of latitude*), and so on. The applications are numerous, including traditional database multi-attribute indexing, Geographic Information Systems and spatial database systems, and indexing multimedia databases by content.

¿From the spatial databases viewpoint we can distinguish between two major classes of access methods:

- Point Access Methods (known as PAMs) which are used to organize multidimensional point objects (e.g. cities, where each city is represented by three coordinates longitude, latitude and altitude, or traditional records assuming one dimension per attribute).
- Spatial Access Methods (known as SAMs) which are used to organize point objects as well as arbitrary shaped objects (e.g. line segments, polygons).

Efficient access methods for point objects (PAMs) include: the LSD-tree, the hB-tree, the Buddy-tree and the TV-tree. All these structures use a hierarchical directory and a set of buckets where all data objects are stored. In Spatial Access Methods (known as SAMs), the fundamental idea for spatial indexing of non-point objects is the use of approximations. In other words, the index handles simple approximations of the actual objects index rather than their actual geometry. The most common approximation used by the majority of existing SAMs is the Minimum Bounding Rectangle (MBR) of the object, i.e., the minimum rectangle with sides parallel to the axes that totally encloses it.

Existing proposals for SAMs are grouped in three different techniques to organize spatial objects.

- The first technique uses *transformation* of non-point objects to points in a higher / lower dimensional space. Therefore, any access method for point / alphanumeric data can be used for the

Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997

indexing of the (transformed) data set. Many of these methods are based on the use of space-filling curves, to derive one-dimensional values for objects; examples of such curves include the Peano curve and the z-ordering, the Hilbert curve and Gray ordering. The basic advantage of this approach is that no specialised access methods need to be implemented for non-point objects since the problem of indexing such objects is reduced to the problem of indexing multi-dimensional points or numeric values.

- The second technique handles *overlapping regions*; the data set is partitioned into groups, whereby two different groups of objects may share portions of the data space (overlapping) but each spatial object is associated with only one group. Access methods in this category organize data directly in the native space (without any transformation) in possibly overlapping buckets. They use simple techniques to maintain the directory and to split buckets when overflow occurs. Access methods that follow the "overlapping regions" technique include the R-tree, and its variations like the R^*-tree, etc.

- The third technique uses clipping and therefore an object may be split to several sub-objects in order to be stored. The main motivation behind clipping is to avoid overlapping regions in the directory all together instead of trying to minimize it. This is also the main advantage of these methods. However, in order to achieve zero overlap between buckets, a spatial object may be decomposed in several components stored in different buckets. If the resulting redundancy cannot be controlled then the space consumption of the method may increase, resulting in performance degradation. Access methods in this category include the R^+-tree, the Cell-tree, and the linear quadtrees.

In the full paper we provide an overview of algorithms that are used to answer point/range queries (further subdivided to intersection and containment) on tree search structures, nearest-neighbor queries, as well as more advanced operations such as spatial joins and direction queries (e.g. find all objects north of another, etc.) In addition, we provide results on the performance of the methods, based on either the mathematical analysis of their behaviour or the experimental use of them.

Having reached a maturity level, multi-dimensional search trees are incorporated into products: linear quadtrees are used in the TIGER system of the U.S. Bureau of Census; R-trees are included in Informix/Illustra in the form of Spatial DataBlades, to name a few.

What is interesting also to note after more than a decade of research and development in the area of multi-dimensional search trees and access methods, is the numerous proposals for using such structures in other than the traditional spatial database applications they were initially proposed for. We have seen multidimensional methods being used to index rules in active database systems, to index multimedia databases by content, to support OLAP and DataCube processing, to index time sequences, and many other novel applications.

Some of the issues that we see being examined in the future include:

Benchmarking. Provide the designers community with statistically well founded workloads sufficient for a variety of benchmarking applications. Such an environment should at least include: a rich database with several real and synthetic datasets, an attractive user interface for user- benchmark communication and a set of tools for visualisation and statistical processing of access methods.

Performance Evaluation of Access Methods. A thorough experimental examination of the various approaches is mandatory, in order to guarantee the nice behavior of the organization under real workloads. This suggests the evaluation of the approaches with real datasets and realistic query types, which will be components of a benchmarking environment. Several criteria for "blind" performance evaluation should be present: *common assumptions* (e.g. about hardware parameters), *extensibility* (i.e., support of a wide range of queries) and *scalability* (i.e., comparison on growing volumes of data) among others.

Query Optimization. The optimization of composite procedure execution is an important issue, which constitutes a relatively undeveloped field in the research area of spatial databases. The term "optimization", although commonly used, is a misnomer, because in many cases (especially in non- conventional DBMSs, like geographic DBMSs) the execution strategy chosen by the DBMS is not the optimal strategy; it is just a reasonably efficient strategy for executing a sequence of operations. The use of heuristics rules for ordering the operations in a procedure execution strategy, as well as the use of systematic cost estimates of the cost of different execution strategies must be further exploited.

There is no question that trees have grown everywhere and will continue to grow in areas where high performance is needed, marking yet another strong contribution from the area of database systems.