Astronomical Data Analysis Software and Systems IV
ASP Conference Series, Vol. 77, 1995
R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.

Application of the Linear Quadtree to Astronomical Databases

P. Barrett

Universities Space Research Association & Laboratory for High Energy Astrophysics, NASA/Goddard SFC, Greenbelt, MD 20771

Abstract. Quadtrees have a wide range of applications, from graphics to image processing to spatial information systems. The use of linear quadtrees to represent spatial information has been widely used in geography, but rarely in astronomy. With the advent of the Guide Star Catalog and other large astronomical source lists, an efficient method of storing and accessing such spatial data is necessary. We show that encoding astronomical coordinates as a linear quadtree, instead of right ascension and declination as is typically done, can provide significant improvements in efficiency when accessing sources near a given spatial direction. We also discuss how the linear quadtree can aid in the correlation of source positions from different astronomical catalogs and how it can be applied to relational databases.

1. Introduction

This paper is a brief introduction to methods of storing multidimensional point data as it applies to astronomy. Such a vast amount of information cannot possibly be covered in a short paper; we have only tried to present the highlights to stir your interest. This study has been motivated by the publication of the Guide Star Catalog. The amount of data (nearly 800 MB) in this catalog can create problems with the storing and accessing of this data, if it resides in a (relational) database. The preliminary results of this study are intended to find solutions to such problems, so that large astronomical catalogs can be easily and efficiently handled by modern databases. A possible solution is the use of hierarchical data structures which provide for efficient methods of searching and storing of the data. Such problems are not new to the sciences. Geographers have had to deal with these problems since the initial development of Geographic Information Systems (GIS) and have therefore made important advances in the storage and access of vast amounts of point data. The information in this paper comes from such sources.

2. Data Structures for Multi-dimensional Point-Data

A common query of astronomical databases is the *range query* (i.e., region search), which requests all records from a database whose specified attributes are within given ranges. Techniques to search this type of query can be divided into

Data Structures	Search Operations	2-D Operations
non-hierarchical:		
sequential list	$O(N \cdot k)$	~ 200000
inverted list	$O(N^{1-1/k})$	~ 1000
fixed grid	$O(2^k \cdot F)$	~ 4
hierarchical:		
point quadtree	$O(k \cdot N^{1-1/k})$	~ 1000
k-d tree	$O(k \cdot N^{1-1/k})$	~ 1000
range tree	$O(log_2^k N + F)$	~ 400
MX quadtree	$O(2^d + F)$	~ 45
PR quadtree	$O(2^d + F)$	~ 45
bit-interleaving	$O(k \cdot log_2 N + F)$	~ 40

Table 1. A list of data structures and the typical number of operations required for a search. N and k are the number of records and search attributes of the list, respectively. For range queries, F is the number of records found. Column three gives the typical number of operations required for finding a single query ($F = 1$) of two attributes ($k = 2$, e.g., RA and Dec) on a list containing $N = 10^6$ records.

two categories: those that organize data to be stored and those that organize the embedding space from which the data are drawn. The binary search tree is an example of the former since boundaries of different regions in the search space are determined by the data being stored. Address computation methods such as radix searching are examples of the latter, since region boundaries are chosen from among locations that are fixed regardless of the content of the file. In a formal sense, the distinction is between *trees* and *tries*, respectively.

Each method has its advantages and disadvantages; some are more suitable to in-core data, while others, the bucket methods, ensure efficient access to disk data. Table 1 provides a list of these methods which are divided into hierarchical and non-hierarchical methods. For each method, column 2 gives an expression for the average number of operations for a search query, while column 3 gives the order-of-magnitude estimate of a 2-dimensional search for a file containing 10^6 records. It is evident that asymptotic (i.e., $\propto logN$) searches are superior to those that are not.

3. Two Proposed Methods of Astronomical Point-Data

In this section a brief description is given of two possible methods for storing astronomical records based on their coordinates. The first uses bit-interleaving of right ascension and declination, while the other encodes the coordinates as Gray codes. Both methods have the property that neighboring points also tend to be neighboring records. This property of the file should improve the efficiency

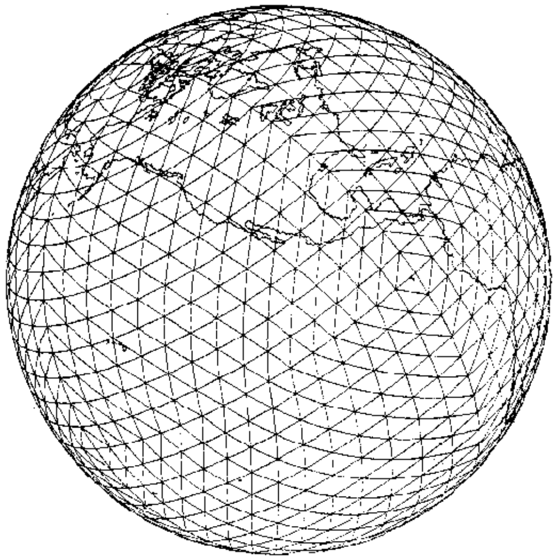

Figure 1. A quaternary triangular mesh (QTM) for a globe. The level 4 system of triangles viewed orthogonally from over the Pacific Ocean.

of range searches. These two methods of encoding data are in effect space-filling curves.

3.1. Bit-Interleaving

This first method of storing astronomical data, bit-interleaving, is the easiest to understand and to implement. A possible disadvantage is that it requires the data to be discrete. For present astronomical use, we believe that this property has minimal impact. Our proposal is as follows:

Right ascension and declination have ranges from $0°$ to $360°$ and $-90°$ to $+90°$, respectively. Normally these coordinates are represented as real numbers, though they can also be represented in a discrete space with minimal impact on the data as long as the discrete space has adequate resolution. Right ascension and declination encoded as 32-bit integers have a resolution of about 1 milli-arcsecond which is completely adequate for almost all current astronomical catalogs. Once the coordinates are made discrete, then by a simple algorithm the bits of the two 32-bit integers are interleaved to form a single 64-bit integer or code.

3.2. Quaternary Triangular Mesh

The second proposed method is the Quaternary Triangular Mesh (QTM; see Figure 1). This method, as for the previous one, is based on the regular tessellation of space which is an infinitely repeatable pattern of a regular polygon (2-dimensional figure) or polyhedron (3-dimensional figure). The first method, bit-interleaving, uses the square as its regular polygon, whereas the QTM method uses the triangle. The QTM method has the properties that: (1) basic units are of almost equal size, (2) basic units are of almost equal shape, and (3) a set of units for which true adjacencies for neighboring elements are preserved. The method first divides the globe into eight triangular regions whose vertices are at the north and south poles and at 0^o, 90^o, 180^o, and 270^o of the equator. This initial octahedron is better for fitting the requirements of polar symmetry and for mapping vertices along the equatorial plane. Figure 1 shows the level 4 system of triangles viewed orthogonally from over the Pacific Ocean.

Two other important properties of these methods are that they reduce k-dimensional space to a 1-dimensional space, and they are bucket methods. The advantage of encoding the data into a 1-dimensional space is that the resulting structure can be readily balanced, since balancing techniques for 1-dimensional data are well known (e.g., AVL trees). It has been shown that whether a data structure is balanced or not has a critical impact on the performance of operations such as search and update. The advantage of bucket methods is that data can be easily stored on disk. If the amount of data is too large to be stored in a single file, it can be divided into several more manageable files with little affect on the performance of searches and updates.

4. Conclusions

We present here two possible methods of storing astronomical point data or coordinates. They have some important properties, including a hierarchical structure for nearly unlimited resolution, and they reduce a k-dimensional space to a 1-dimensional space, which permits the construction of *balanced* binary trees and hence efficient data structures and searches. Finally, neighboring points typically have neighboring records for efficient range searches, and the ability to use bucket methods for storing the data easily and efficiently on disk.

References

Dutton, G. 1989, in Proc. Ninth International Symposium on Computered-Assisted Cartography (Falls Church, ASPRS/ACSM), p. 462

Goodchild, M., & Shiren, Y. 1989, NCGIA Tech. Paper (Santa Barbara, Univ. of California), p. 89

Laurini, R., & Thompson, D. 1992, in APIC Series: Fundamentals of Spatial Information Systems (San Diego, Academic), p. 133

Samet, H. 1989, The Design and Analysis of Spatial Data Structures (Reading, Addison-Wesley), p. 43