Logical Operations in Compact Geospatial Quadtrees

Kirk L. Duffin
Northern Illinois University
duffin@cs.niu.edu

1 Introduction

One of the key data types in family history research is that of location. Locations help uniquely distinguish events that happen to specific individuals. Locations also provide valuable clues to identifying resources for future research.

Like most family history data, locations posess uncertainty, sometimes significant uncertainty. In addition, the defining characteristics of a location, such as boundaries, may change over time. The challenges imposed by this uncertainty often make it necessary to consult experts on a particular time and place in order to perform effective family history research.

The long term goal of the research presented here is the creation of artificial family history experts that can analyze family history data and suggest further research directions. This paper presents continuing progress on research in representing detailed location information in a compact form, along with operations for manipulating the data.

2 Geospatial Quadtrees

Quadtrees are a data structure used to represent two dimensional shapes[Sam84]. Each node in the tree represents a physical area, usually a square or triangle, which can be subdivided into four similar shapes. Nodes are marked with values representing whether the node is in the area, out of the area, or partially contain the area. These values are traditionally denoted, BLACK, WHITE, and GRAY, respectively. See Figure 1 for an example of a quadtree.

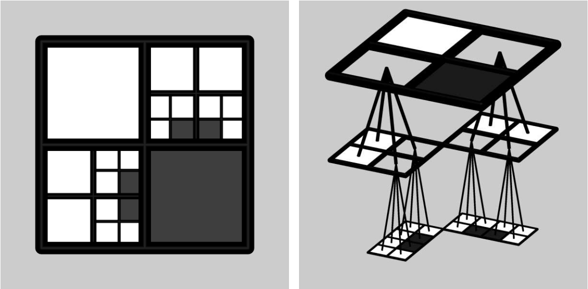

Figure 1: A rectangular quadtree

For geospatial shapes, an alternative form is used, based on triangular subdivision[Dut00]. The base triangles form

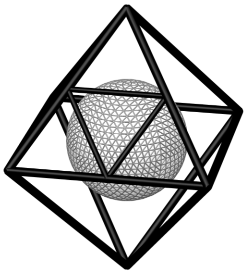

Figure 2: A spherical octahedron for triangular quadtrees.

a spherical octahedron as shown in Figure 2. An advantage of quadtrees is the flexible resolution of detail that exists. The shape can be coarsely defined in some areas and finely detailed in other areas. Only areas of interest need have detail preserved, but new detail is easily integrated into the representation. Another advantage is the easy conversion of detailed shape description into coarser form[Sam85].

3 Compact Forms

Much of the memory associated with a quadtree node goes to the child pointers, so their is considerable interest in reducing quadtree storage requirements. A quick simplification is to note that each node has exactly four children or none at all. Consequently, only a single pointer is needed to a block of four consecutive nodes[RL00].

Even greater savings can be obtained by using pointerless trees, often known as linear quadtrees. Linear quadtrees often take the form of either a depth order listing of node positions [Gar82] or breadth order listing of node values.

3.1 Breadth First Linear Quadtrees

Breadth first linear quadtrees posess the prefix property: they can be truncated at any point in the sequence to produce a complete, but lower resolution representation of the original shape. Minor tweaking of former interior node values may be required for correctness.

Breadth first linear quadtrees have been shown to have smaller storage requirements than depth first linear quadtrees.

Breadth first trees have four node values based on the relationship to the region of interest: BLACK (in), WHITE (out), GRAY (partial) and BRANCH (internal node).

4 Logical Operations

The operations of negation, union and intersection are similar to boolean logical NOT, OR, and AND. Similar node positions are compared using extended truth tables.

Negation is easily performed by walking through the linear quadtree and negating the values. But union and intersection can only be performed by comparing compatible nodes, those that represent the same position at the same level. This necessitates extending leaf nodes of each tree to cover the corresponding spatial division of the other tree.

Tree extension is performed by simultaneously traversing the trees with two indices each, one pointing to a node of the tree and one pointing to its children. As the trees are traversed, copies are made, extending the nodes into the new trees as needed.

The two compatible extended trees are compared node by node according to the desired logical operation and the results placed in another tree. Depending on the values of the nodes, the result may need to be condensed into a smaller tree. This occurs by traversing and copying again, removing nodes whose siblings all have the same leaf node value. Each condensation pass removes only one level of equal valued nodes so multiple condensation passes may be needed, but the number of passes needed is limited by the depth of the tree.

Processing time in non-compact trees is bounded by dynamic memory management issues. Processing time in compact trees is bounded by copying. In most cases, the processing times on compact trees outweigh the times of non-compact trees, but the maximum memory used by compact trees is smaller, enabling larger problems to be processed. Table 1 gives some examples for logical combination of several trees.

5 Comparison

Examples of logical operations on compact regions will be shown, using regions created from the 1990 U.S. Census TigerLine data set. Operations will be performed on regions at the city, county, state, and national levels, illustrating a wide range of memory storage requirements. Comparisons will be made with standard pointer based representations of the same data.

Problem	time	max size
Full quadtree		
Illinois counties	1.28s	21.7MB
Utah counties	0.88s	19.0MB
3 states	0.41s	18.0MB
All U.S.	9.32s	227.3MB
Compact quadtree		
Illinois counties	3.82s	9.49 MB
Utah counties	3.80s	9.49 MB
3 states	0.59s	8.99 MB
All U.S.	114.26s	122.13 MB

Table 1: Processing times and maximum memory usage for several logical union problems. All times on AMD64 3400+

6 Compression Ratios

Most quadtrees conist largely of WHITE and BRANCH nodes at the higher levels. Most of the detail of BLACK and GRAY nodes comes at lower levels. In breadth first traversals, the first part of the representation contains reoccurring patterns that can be compressed further. Table 2 lists the compact size for several objects and the further size reduction obtained by compressing the encoded quadtree with `gzip`. On average, compressed trees require 0.657 bits per node to encode.

References

[Dut00] Geoffrey Dutton. Universal Geospatial Data Exchange via Global Hier-

Object	Size(bytes)	Compressed	ratio
United States	4796119	1429198	0.2980
Alaska	2241834	297219	0.3110
Illinois	285736	88949	0.3113
Utah	248420	37361	0.1504
DeKalb County	22323	4986	0.2234
Utah County	58455	23351	0.3995
DeKalb, IL	11751	5831	0.4962
Provo, UT	10263	4485	0.4370
Average			0.3284

Table 2: Compression of encoded breadth first quadtrees.

archical Coordinates. In *Proceedings of International Conference on Discrete Global Grids*, 2000.

[Gar82] Irene Gargantini. An Effective Way to Represent Quadtrees. *Communications of the ACM*, 25(12):905–910, Dec. 1982.

[RL00] Szymon Rusinkiewicz and Marc Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. In *Proceedings of ACM SIGGRAPH 2000*, pages 343–352, 2000.

[Sam84] Hanan Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2):187–260, June 1984.

[Sam85] Hanan Samet. Data Structures for Quadtree Approximation and Compression. *Communications of the ACM*, 28(9):973–993, 1985.