A File Organization Scheme for Polygon Data

Chung Hee Hwang & Wayne A. Davis

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1

Abstract[1]

This paper presents a file organization scheme for representing polygon data by a quadtree. The proposed scheme is an adaptive cell method that is based on *extendible hashing* and *interpolation-based index maintenance*. It aims, on the average, to locate the record associated with a given key with one disk access (or at most two), maintaining a high storage utilization ratio. It also aims to process range search and set operations efficiently. The dynamic file organization capabilities of the scheme and the algorithm for range search are described.

1. Introduction

Given a sequence of k points $P_i=(x_i,y_i)$, for $1 \le i \le k$, in a plane, a polygon with vertices P_i is the sequence of line segments, called edges, P_1P_2, P_2P_3,..., P_kP_1. If these k edges do not have any intersection points, the polygon is simple. A simple polygon divides the plane into two distinct regions. The interior of a simple polygon is called a *polygonal region*. A *complex polygonal region* is a polygonal region which is allowed to have one or more holes in it. Hereafter, a complex polygonal region is called a *polygon*. A *polygon network* for a study area is a set of disjoint polygons overlapping the study area such that the set of polygons yields a total partition of the study area. Each polygon in a polygon network has a unique name.

Although polygon networks have been traditionally represented in vector format, recently quadtree encoding of a polygon network has received increased attention. The quadtree [7] is a dynamic data structure developed to reduce the storage requirement of raster representation by aggregating homogeneous cells. Nevertheless, as the original quadtree concept was based on the assumption that quadtrees were resident in main memory, quadtree structures may not be directly applicable to data resident in external memory. For example, the need to follow pointers may lead to a larger number of page faults than are acceptable in an interactive environment.

In an effort to overcome the frequent page fault problem, there have been studies to represent a quadtree as a linear quadtree [5] and use a B-tree file structure in organizing the data [1,8]. While the B-tree organization of a linear quadtree is a significant improvement over the original quadtree organization in the expected number of disk accesses for single record retrieval, the absence of a localization property is a primary disadvantage. A query usually requires the whole file to be retrieved even though the query can be answered with information from a small part of the file. For example, range search is very awkward and set operations are not efficient because there is no implied connection between data buckets in physical storage and regions in the search space. Furthermore, the B-tree organization of quadtree encoded data still needs several disk accesses to retrieve each record because it is essentially a tree that is accessible with $O(\log n)$ I/O operations, where n is the number of records in the file.

From this perspective, a file organization scheme is developed for polygon networks encoded as a linear quadtree with an aim to locate the record associated with a given key with an average of one disk access (or at most two), maintaining a high storage utilization ratio. Furthermore, the following types of spatial queries are to be supported efficiently: the point-in-polygon query, range search and set operations such as polygon union or intersection and polygon overlay. The scheme is an adaptive cell method and it is based on extendible hashing [4] and interpolation hashing [2], a k-dimensional generalization of linear hashing [6].

2. Definitions and Notation

Let the study area, $\mathbf{U} = [0, 2^n)^2$, be an image of $2^n \times 2^n$ unit square pixels that intersects a polygon network, and let each of the pixels have a polygon name (hereafter called *color*) associated with it. Furthermore, let the polygon network on $\mathbf{U}$ be represented by a region quadtree. To yield an arbitrary but consistent total ordering among the blocks of a quadtree, the following hash function is introduced:

Definition 1. Let $(x,y) \in \mathbf{U} = [0,2^n)^2$ be the x and y coordinates of the lower-left corner of a block of a quadtree defined on $\mathbf{U}$ and have the following binary representation:

$$x = \sum a_i 2^i, \text{ and } y = \sum b_i 2^i, \text{ for } 0 \le i \le n-1,$$

where $a_i, b_i \in \{0,1\}$. Then, an *order preserving hash function*, s, that maps (x,y) onto the key space of $[0,4^n)$ is defined by:

$$s(x,y) = \sum(a_i 2^{2i+1} + b_i 2^{2i}), \text{ for } 0 \le i \le n-1.$$

Notice that the key produced for each of the blocks in this manner is essentially the same as the locational code of the block in linear quadtree encoding [5]. Now, a file that represents a polygon network by a quadtree is defined:

Definition 2. A file $\mathbf{F}$ representing a polygon network for the study area $\mathbf{U}$ by a quadtree is the set,

$$\mathbf{F} = \{(K(L),S(L),C(L)): L \in R\},$$

where R is the set of all leaf nodes (blocks) of a region quadtree on $\mathbf{U}$,

K(L) is the key of L produced by s,
S(L) is the size, or alternatively the level, of L, and
C(L) is the color of the pixels intersecting L.

[1]This research was supported in part by Grant NSERC A7634.