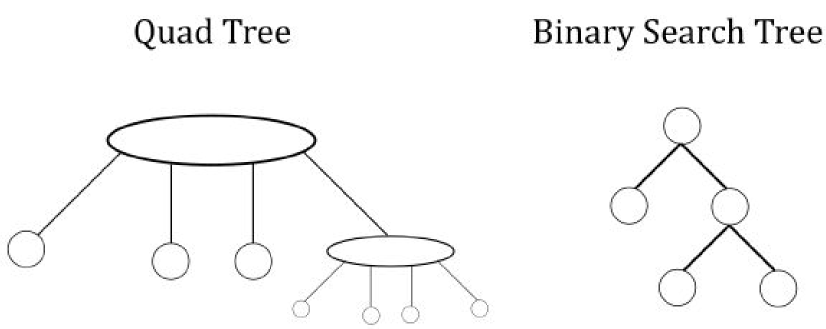
Servando Varela
Joshua Thompson

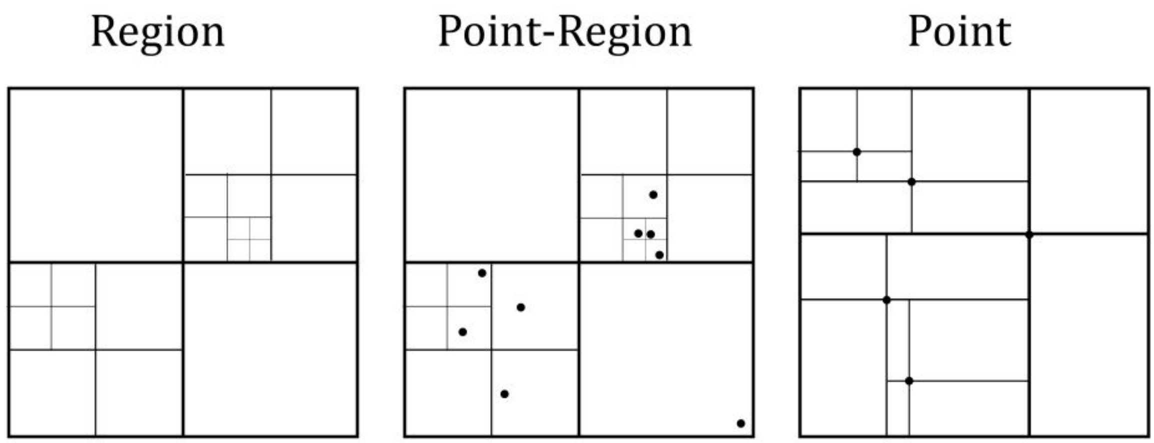
Quadtrees

Introduction

In 1974 Raphael Finkel and Jon Bentley developed quadtrees as an adaptation of the binary search tree (BST) that allows for 2+ dimensional data representation, and today, the primary application of quadtrees is representing multidimensional data. Although this is the primary application for the broad category that is “quadtree”, there is a wide variety of quadtrees that are tailored for specific tasks. However, in this paper we will be focusing on the three most common types of quadtrees: region quadtrees, point quadtrees, and point-region quadtrees.

Pictured below is a quad tree in its tree representation, a binary search tree, and the graph representations of region, point-region, and point quadtrees.





What are quadtrees, and how do they work?

Quadtrees are similar to 234 trees with regards to the strict allowance of children and the way that they are searched. The difference is that in a quadtree, every node represents a region and not necessarily a single value, and each node is restricted to having either no children or exactly 4 children. However, searching the quadtree is very similar to searching a 234 tree. Without getting into too much detail yet, if we wanted to find the region in which a single point lied on the quadtree, we would compare the coordinates of that point to the boundaries of every child of the root, then every child of that child, and so on, until the smallest child was reached.

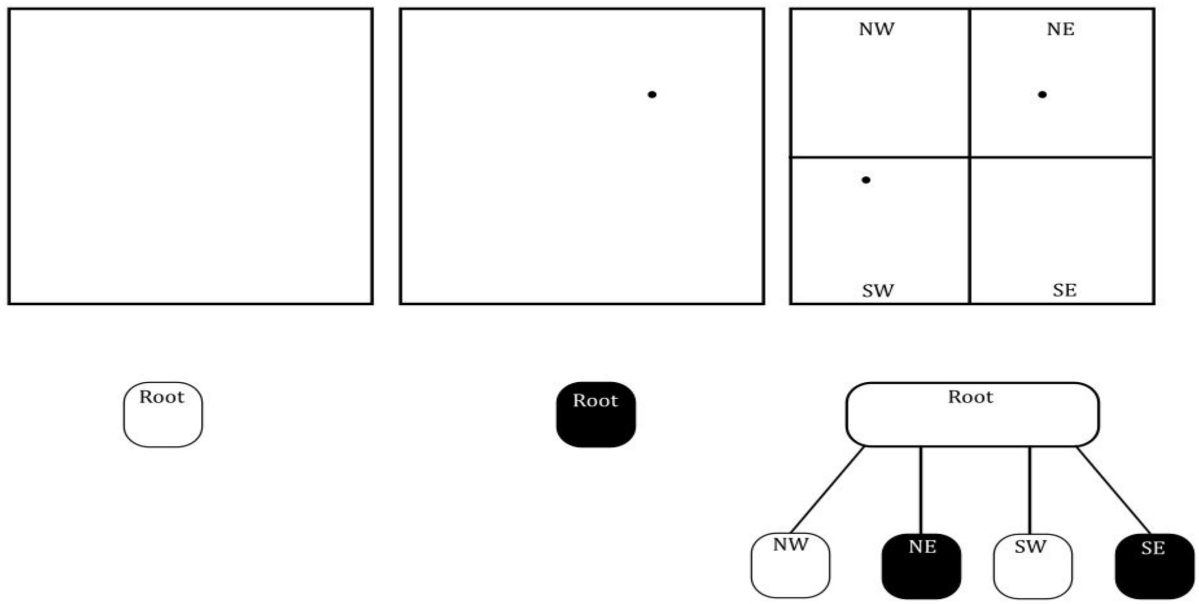
To discuss how quadtrees function, we should first discuss their components. A quadtree's nodes represent regions as shown in the above images. Each region/node is given a capacity, and boundaries. The capacity refers to point-region quadtrees and represents the number of points that may lie within a region before that region is divided into 4 more subregions, the equivalent of getting 4 new children. In the above point-region graph, the capacity is one, which is obvious because there is only one point in each region. The next component is the set of boundaries.

The boundaries represent the borders shown in the above graphs. Each region has leftmost, rightmost, uppermost, and lowermost boundaries. These boundaries are essential for defining what each region actually is, and what it contains. When searching to find what region a point lies in, the boundaries of regions are compared to the coordinates of the point.

Constructing a point-region quadtree

To construct a quadtree, we start with a root node with no children, which is equivalent to a region with no subregions. Suppose the capacity for our nodes will be one. When we add one point to the region, it will be divided into four equal regions. Now suppose our capacity is two. After adding one node to the region, nothing happens. However, after the second point is added, the region will then be split.

Pictured below is the construction of a quadree from an empty root with capacity of two. Notice how the children are represented as NW (North West), NE (North East), etc… This is a common way to represent the four children, as well as numbering one through four. Additionally, nodes may be colored black when they contain a point. That is, when that region contains a point itself, not more subregions.



So what can we do with this? Common queries we would want to perform would include:

- Finding the region in which a point lies
- Finding the points near a given point
- Finding all points within a specified region

This follows nicely into the next topic, actual uses and applications for quadtrees.

Applications of quadtrees

The axes associated with quadtrees can be represent anything. For example, say we would like to compare age with average annual income. We can put age on the Y-axis and income on the X-axis. Each point would represent a person that is Y years old and makes X income. The quadtree would allow us to find how many people are in a certain range of income, or perhaps find the age-income bracket that a particular person sits in. This is a simple application, but remember that quadtrees allow for infinite dimensions, and so an infinite number of factors can be added. For example, maybe we also want to include ethnicity, number of fingers, and glasses of water drank per day. Quadtrees would support this bizzare multidimensional comparison.

Another common application includes image compression. Where region is the average color of all colors within that region on the image. As the regions subdivide, the image gets clearer and clearer as it gets more detailed in areas with a lot of different colors. Meanwhile, a region with only one color will not subdivide any further. Example pictured below.