1.4 Quadtrees

Recall from Section 1.1 that the quadtree is the result of imposing a tree access structure on a grid so that spatially adjacent empty grid cells are merged into larger empty grid cells. In essence, it is equivalent to marrying a k-ary tree, where $k = 2^d$, with the fixed grid. There are two types of quadtrees. The first is a point quadtree [614] where the subdivision lines are based on the values of the data points. The second is trie-based and forms a decomposition of the embedding space from which the data points are drawn.

In the rest of this section, we elaborate further on the differences between the point quadtree and the trie-based quadtree by showing how they are updated (i.e., node insertion and deletion) and how they are used for region searching. In particular, Section 1.4.1 presents the point quadtree. Section 1.4.2 discusses trie-based quadtrees, such as the PR and MX quadtrees. Section 1.4.3 contains a brief comparison of the point, PR, and MX quadtrees.

1.4.1 Point Quadtrees

In two dimensions, the point quadtree, invented by Finkel and Bentley [614], is just a two-dimensional binary search tree. The first point that is inserted serves as the root, and the second point is inserted into the relevant quadrant of the tree rooted at the first point. Clearly, the shape of the tree depends on the order in which the points were inserted. For example, Figure 1.14 is the point quadtree corresponding to the data of Figure 1.6.

The rest of this section is organized as follows. Section 1.4.1.1 shows how to insert a point into a point quadtree. Section 1.4.1.2 discusses how to delete a point from a point quadtree. Section 1.4.1.3 explains how to do region searching in a point quadtree.

1.4.1.1 Insertion

Inserting record r with key values (a,b) into a point quadtree is very simple. The process is analogous to that for a binary search tree. First, if the tree is empty, then allocate a new node containing r, and return a tree with it as its only node. Otherwise, search the tree for a node h with a record having key values (a,b). If h exists, then r replaces the record associated with h. Otherwise, we are at a NIL node, which is a child of type s (i.e.,

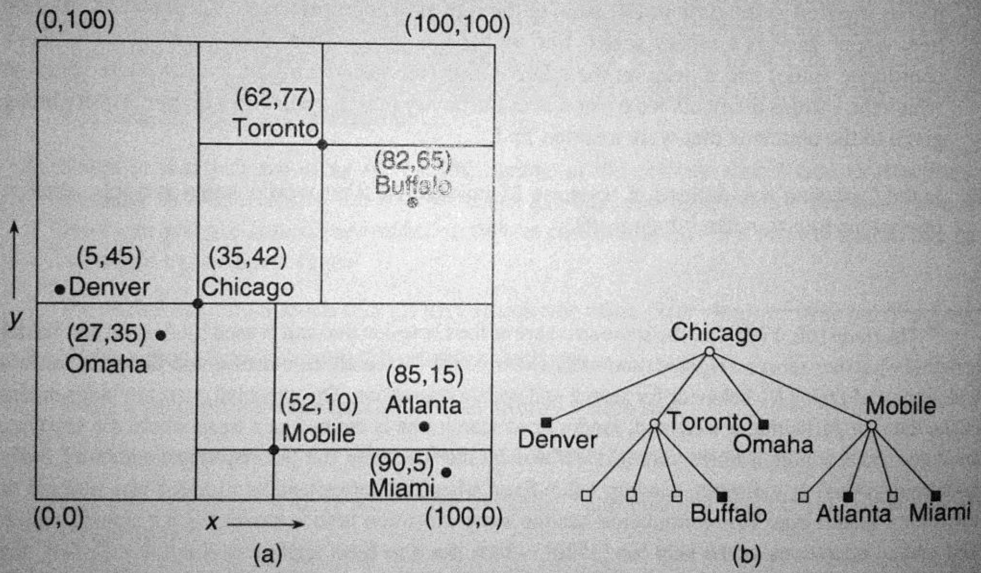
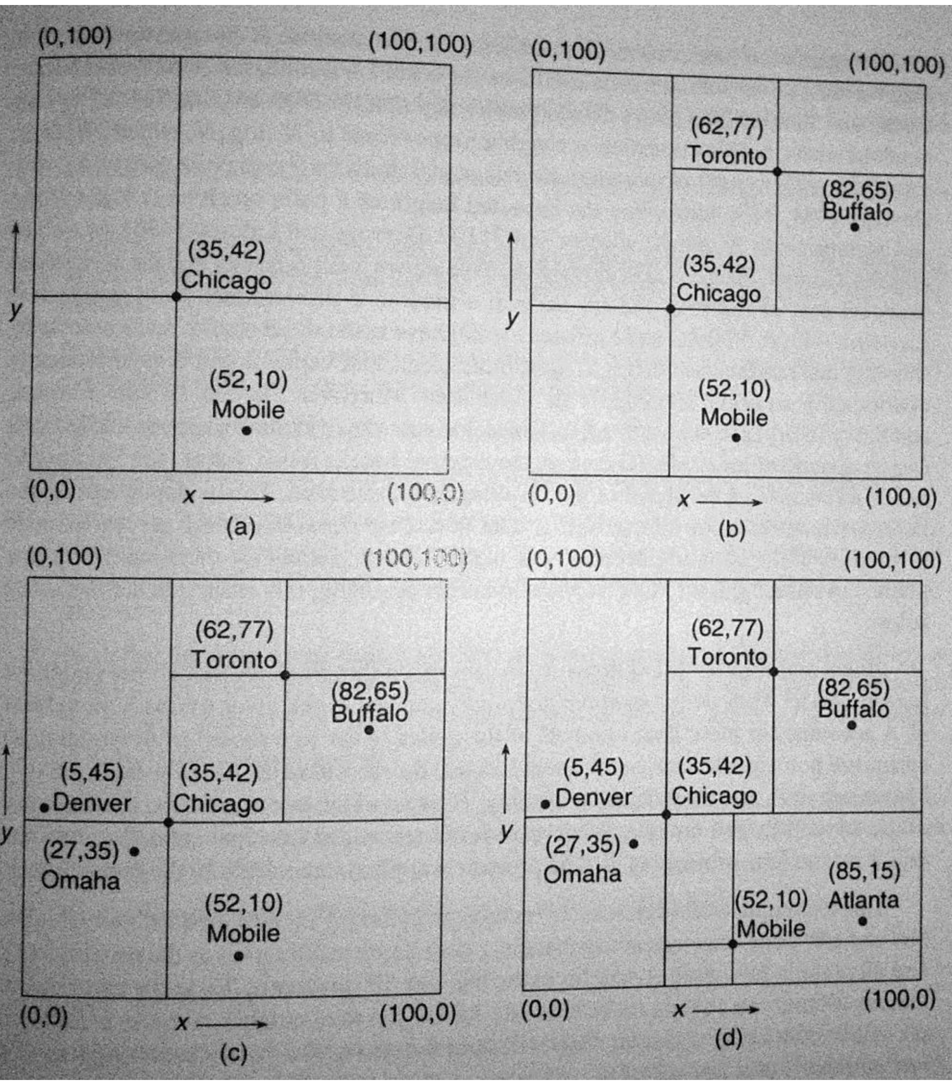


Figure 1.14
A point quadtree and the records it represents corresponding to the data of Figure 1.1: (a) the resulting partition of space and (b) the tree representation.


(0,100)
(100,100)
(35,42)
Chicago
(52,10)
Mobile
(0,0)
(100,0)
x
y
(a)
(0,100)
(100,100)
(62,77)
Toronto
(82,65)
Buffalo
(35,42)
Chicago
(52,10)
Mobile
(0,0)
(100,0)
x
y
(b)
(0,100)
(100,100)
(62,77)
Toronto
(82,65)
Buffalo
(5,45)
Denver
(35,42)
Chicago
(27,35)
Omaha
(52,10)
Mobile
(0,0)
(100,0)
x
y
(c)
(0,100)
(100,100)
(62,77)
Toronto
(82,65)
Buffalo
(5,45)
Denver
(35,42)
Chicago
(27,35)
Omaha
(85,15)
Atlanta
(52,10)
Mobile
(0,0)
(100,0)
x
y
(d)

2.1.2.10 Hexagonal Blocks

As we pointed out in Section 2.1.1.1, the drawback of a decomposition into hexagons is that they cannot be decomposed beyond that atomic tiling without changing the basic tile shape. Also, hexagonal unit-size cells cannot be aggregated into larger hexagonal cells. Nevertheless, there are a number of different hexagonal hierarchies of blocks that are distinguished by classifying the shape of the first-level molecular tile on the basis of the number of hexagons that it contains. Three of these tiling hierarchies are given in Figure 2.34 and are called *n-shapes*, where *n* denotes the number of atomic tiles in the first-level molecular tile. Of course, these n-shapes are not unique.

The 4-shape and the 9-shape have an unusual adjacency property in the sense that no matter how large the molecular tile becomes, contact with two of the tiles (i.e., the one above and the one below) is only along one edge of a hexagonal atomic tile, while contact with the remaining four molecular tiles is along nearly one-quarter of the perimeter of the corresponding molecular tile. The hexagonal pattern of the 4-shape and 9-shape molecular tiles has the shape of a rhombus. In contrast, a 7-shape molecular tile has a uniform contact with its six neighboring molecular tiles.

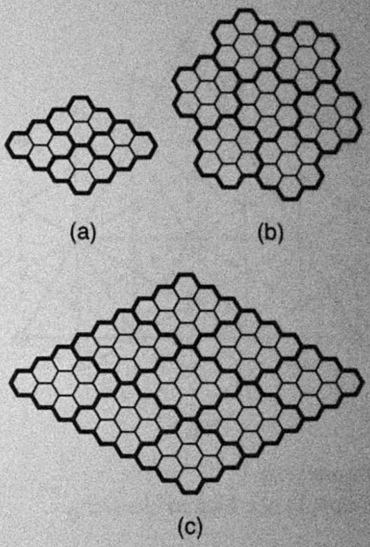


Figure 2.34
Three different hexagonal tiling hierarchies: (a) 4-shape, (b) 7-shape, and (c) 9-shape.

Given the various block shapes described above, the choice is usually made on the basis of the grid formed by the image sampling process. Square blocks are appropriate for square grids, and triangular blocks are appropriate for triangular grids. In the case of a hexagonal grid [285], the 7-shape hierarchy is frequently used since the shape of its molecular tile is more like a hexagon. It is usually described as *rosettelike* (i.e., a *septree*). Note that septrees have jagged edges as they are merged to form larger units (e.g., Figure 2.34(b)). The septree is used by Gibson and Lucas [706], who call it a *generalized balanced ternary* (GBT), in the development of algorithms analogous to those existing for region quadtrees.

Although the septree can be built up to yield large septrees, the smallest resolution in the septree must be decided upon in advance since its primitive components (i.e., hexagons) cannot later be decomposed into septrees. Therefore, the septree yields only a partial hierarchical decomposition in the sense that the components can always be merged into larger units, but they cannot always be broken down. For region data, a pixel is generally an indivisible unit and thus unlimited decomposition is not absolutely necessary. However, in the case of other data types such as points (see Chapter 1) and lines (see, e.g., Section 2.2.2.6), we find that the decomposition rules of some representations require that two entities be separated, which may lead to a level of decomposition not known in advance (e.g., a decomposition rule that restricts each square to contain at most one point). In this book, we restrict our discussion to hierarchies of rectangular blocks.

shape, and (c) 9-shape.

2.1.2.11 Triangular Blocks

Triangular blocks are used in triangular quadtrees that are based on a recursive decomposition of an equilateral triangle into four congruent blocks. Such a triangular decomposition results from using the [6^3] tiling. Triangular blocks are also used in triangular bintrees that are based on a recursive decomposition of an isoceles triangle into two congruent blocks. Such a triangular decomposition results from using the [4.8^2] tiling.

When the data lies on the surface of a sphere, a number of researchers have proposed the use of a representation based on an icosahedron (a 20-faced polyhedron whose faces are regular triangles) [520, 603]. The icosahedron is attractive because, in terms of the number of faces, it is the largest possible regular polyhedron. Each of the triangular faces can be further decomposed in a recursive manner into n^2 ($n > 1$) spherical triangles (the [6^3] tiling).

Fekete and Davis [603] let $n = 2$, which means that, at each level of decomposition, three new vertices are generated by halving each side of the triangle; connecting them together yields four triangles. They use the term *property sphere* to describe their representation. The property sphere has been used in object recognition; however, it is also of potential use in mapping the Earth as it can enable accurate modeling of regions around the poles. For example, see Figure 2.35, which is a property sphere representation of some spherical data. In contrast, planar quadtrees are less attractive the farther we get from the equator due to distortions in planarity caused by the Earth's curvature. Of course, for true applicability for mapping, we need a closer approximation to a sphere than is provided by the 20 triangles of the icosahedron. Moreover, we want a way to distinguish between different elevations.

Dutton [520] lets $n = \sqrt{3}$, which means that, at each level of decomposition, one new vertex is created by connecting the centroid of the triangle to its vertices. The result is an alternating sequence of triangles so that each level is fully contained in the level that was created two steps earlier and has nine times as many triangles as that level. Dutton uses

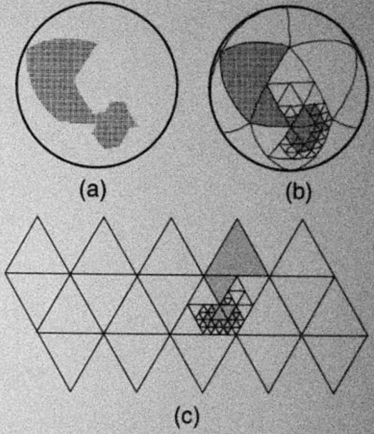


Figure 2.35
Property sphere representation of some spherical data: (a) data, (b) decomposition on a sphere, and (c) decomposition on a plane.