**Summary of Chapter 6**

This chapter mainly introduced the database structure and access methods. As a database, the typical organization is as a collection of files, records, and stored on a disk. A file is organized by a sequence of record. A record is a sequence of fields related to a single logical entity. And field is a place for a data item in a record.

For the unordered file, it does not have structure except entry order, so new records are inserted into it in the next physical position on the disk. And when a record deleted, a “hole” will be created. Linear search could be used on the unordered files. Linear search required to scan every record, and the time complexity is O(n).For an ordered file organization, each record is inserted into the file in the order of values in its field. Binary search may be used on ordered fields. B-tree is an index structure that handles the modifications. The B-tree has some operations, such as search insert and delete. From the book, we can get the B-tree has several properties. First of all, the B-tree is completely balanced. Secondly, insertion and deletion have time complexity O(logn), search has time complexity O(logn). The last one is each node is guaranteed to be at least half-full at all stages of the tree’s evolution.

The spatial dimensions and spatial data are orthogonal, but they are depended with each other. There are two types of queries on the planar points: point and range queries. There are six common title indexes: row order, row-prime order, Cantor-diagonal order, spiral order, Morton order and Peano-Hilbert order. Each title index has different features. The row order corresponds to the scan of a raster image across the rows of pixels. The row prime order provides a simple modification. The Cantor-diagonal order and spiral order are good at preserving nearness. Morton and Peano-Hilbert orderings are therefore space filing and can be constructed at arbitrary levels of detail.

Raster is able to represent a large range of computable spatial objects. A raster when stored in a raw state with no compression cannot be efficient in terms of computer storage. In order to represent the raster boundary of a region, chain codes could be used. Run-length encoding (RLE) is an alternative representation that counts the length of “run” of consecutive cells. Block encoding is a generalization of run-length encoding to two dimensions. Region quadtree is used for holding planar areal raster data, and its principle is a recursive subdivision of a nonhomogeneous square array of the cells into four equal-sized quadrants.

The point object structures are considered as grid files, point quadtrees and 2D-trees. The grid structure allows binary search methods to be applied in more than one dimension. The point quadtree combines the grid approach with a multidimensional generalization of the binary search tree. The 2D-tree solves the exponential increase in the number of descendant of non-leaf nodes as the dimension of the embedding space increases.

PM quadtree is a variant of region quadtree designed to storing linear objects. It has several types of PM quadtree, including PM1, PM2 and PM3 quadtrees.

Efficient indexes for rectangles are important because rectangles can be used to get bounded planar spatial objects. Each geometric object is enclosed in its MMB (minimum bounding box). The R-tree is a rooted tree, and its node represents a rectangle. The leaf nodes represent containers for the actual rectangles to be indexed. The R+-tree is a refinement of the R-tree that does not permit over-lapping rectangles associated with non-leaf nodes. The binary space-partitioning (BSP) tree is a binary tree, which hierarchically decomposes the plane into polygonal regions. Because, the structure of BSP-tree is dependent on the order of insertion, it could be unbalanced.

Spherical tessellations provide some methods to surface of the Earth; the quaternary triangular mesh (QTM) recursively can get locations on the surface of a sphere by nested collection of equilateral triangles.

**Summary of Encyclopedia of GIS**

The article Voronoi Diagram talked about the definition, key applications, and future direction of Voronoi Diagram. And, the author used several examples to explain Voronoi Diagram. In general, this method decomposes a set of objects in a spatial space to a set of polygonal partitions. There are many variants to the Voronoi Diagram such as the k-th order, the Farthest-Site. In order to create a Voronoi Diagram, we need to find all the points that are the nearest to the generated object and thepoints that may have more than one closest generated point at first. Then, we also need to find the polygonal shapes around each object. Thus, the polygonal shapes comprise the completed first order Voronoi Diagram. Voronoi Diagram had already been used in many domains, such as science, astronomy, biology, nearest neighbor problem, and route planning. Voronoi Diagram can be widely used in everything that represented spatially.

The article R-Trees talked about the R-Tree, which has been show on the textbook from page 250 to 253. The R-tree structure is the most important access methods in the area of Spatial Data Management. The R-tree comprises a generalization of the B+-tree structure for multiple dimensions. The splitting criteria are as the following: Linear Split, Quadratic Split, and Exponential Split. R-tree is diverse. Base on it, some other structure derived, such as R+-tree, R\*-Tree, Static R-tree, Packed R-tree, Hilbert Packed R-tree, STR R-tree and Time-Evolving R-tree. R-tree had already been used in many areas. For example, it can be used for processing spatial queries. It also can be used in spatio-temporal database, multimedia database and data warehousing and data mining. Based on the research, it covers almost every aspect of concerning a database. R-tree will be a ubiquitous data structure and it will be important in modern system and application.

R\*-tree is an improvement of R-tree and it is an efficiently access method for points and rectangles. R\*-tree supports point and spatial data at the same time and its implementation cost is a little higher than of other R-tree variants. In the article, the author introduces the optimization criteria for R-trees and how to choose the subtree. Then, the author analysis how to split a node and forced reinsert. Now, R\*-tree has been used in geographic information system (GIS), digital mock-up (DMU), and multidimensional feature vectors. The R\*-tree is a popular access method in database systems organizing both multidimensional point data and spatial data. R\*-tree will be more suitable for the more complex object representation.

The article Mobile Object Indexing introduces the definition of mobile object indexing and the way for indexing in one dimension and two dimensions. The traditional way to record the position of moving objects in one and two dimensions is storing the location, start time, and velocity vector. Once the object’s motion changed, the motion information will be updated. The drawback of this way is it can’t be applied when object moves to infinity. Mobile object indexing is pretty useful in “location-aware applications such as traffic monitoring, intelligent navigation and mobile communications management require the storage and retrieval of the locations of continuously moving objects” (Kollios, Tsotras, Gunopulos, “Encyclopedia of GIS”, 670).

Reference

1. Vornoi Diagram, J. Kang, Encyclopedia of GIS, page 1232-1235.
2. R-tree, M. Hadjieleftheiou, et al, Encyclopedia of GIS, page 993-1002.
3. R\*-tree, H. Kriegel, P. Kunath, Encyclopedia of GIS, page 987-992.
4. Mobile Object Indexing, G. Kollios, V. Tsotras, Encyclopedia of GIS, page 663-670.