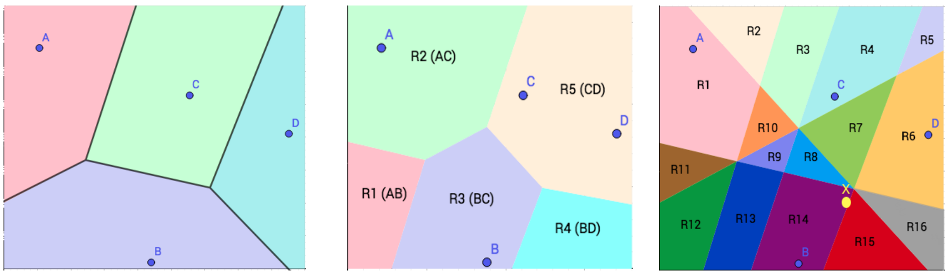
Chapter I

Introduction

1.1 Overview

Indexing has been used to accelerate a search process and minimize effort to read the actual data [4]. In spatial database, indexing is a data structure to increase the access speed of data by trimming the search area data. Indexing is useful to search the closest region and speed up the search [8]. There are various objects in indexing, such as point, regular polygon, and irregular polygon [8]. Irregular polygon is the shape of Voronoi Diagram. Voronoi Diagram used to divide a region into smaller regions based on its nearest neighbor [1].



(a) First Order Voronoi Diagram (b) Higher Order Voronoi Diagram (c) Highest Order Voronoi Diagram

Figure 1.1: The example of Voronoi Diagram

Voronoi Diagram has three orders, namely: (1) First Order Voronoi Diagram shows in Figure 1.1a, (2) Higher Order Voronoi Diagram (HOVD) shows in Figure 1.1b, (3) Highest Order Voronoi Diagram (HSVD) shows in Figure 1.1c, which is the latest order. Although Voronoi diagram has already been used in many variations of spasial query, it still has some disadvantages for the First Order and Higher Order Voronoi Diagram, that are limited and not dynamic. These disadvantages can be solved by using the latest order of Voronoi Diagram, that is Highest Order Voronoi Diagram (HSVD), which can be used for all orders in Voronoi diagram [1].

HSVD has some advantages to identify the farthest point and the region, as well as to identify all distances for each region. However, this method still has the disadvantage, that the data object to the fragmentation (the process of divided regions into smaller region) can not be directly retrieved (bring back the region) by the search data. Thus, for accessing the fragment (fractional part Voronoi Diagrams can be called a Voronoi cell), it will take a long time. The fragment is difficult to access and requires high computing for polygon-shaped fragment [1]. The fragmentation can be accessed using a linear search for checking but the fragment will be checked one by one that can cause a long time [2]. These problems can be solved by using the index shown in the Figure 1.2.

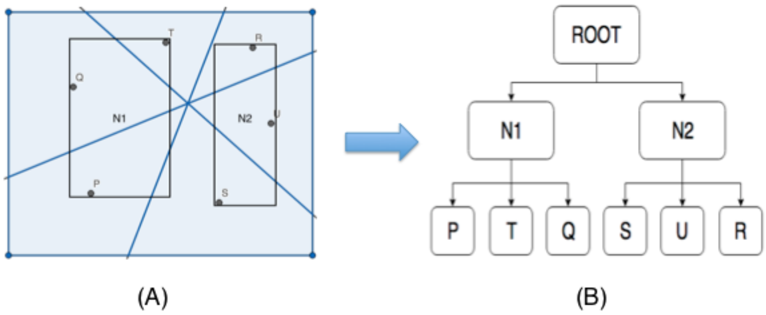


Figure 1.2: Index use VoR Tree

The VoR-Tree (Voronoi R-Tree) already designed for Voronoi Diagram but only for order-1 Voronoi Diagram. VoR-Tree has never been studied on HSVD. There are several kinds of index in spatial database especially in HSVD, such as VoR-Tree, Quadtree and K-D Tree. VoR-tree used to Voronoi Diagram specifically that allows us to examine at each step. They only examine the points inside the current search region [9]. Quadtree is a common spatial index can divide it into four equal sub-spaces, but the balance of the index tree is difficult to control [6]. K-D Tree is a partitioning the space into two parts of data for organizing points in k-dimensional space [7].

Previous studies in the past decade utilized R-trees as their underlying index structures to solve problems efficiently [9]. R-Tree is a method to split the network space into Minimum Bounding Rectangle (MBR) [9]. The figure 1.3 shows the irregular polygon that use MBR can cause a high level of overlapping. VoR-Tree (Voronoi R-Tree) can be used to minimize overlapping.

Basically, the VoR Tree has the same method with R-Tree but VoR Tree is an R-tree on a point combine with the Voronoi Diagram and Delaunay graph. The rectangle nodes of R-tree enable us to search region in logarithmic time, while the polygons of Voronoi diagram allow us to efficiently tile or cover the regions and find the result [9].

This final project presents an index VoR Tree structure to prove VoR Tree can be implemented effectively on Highest Order Voronoi Diagram.

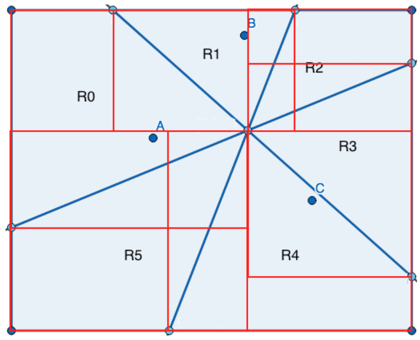


Figure 1.3: MBR in the Highest Order Voronoi Diagram

1.2 Research Problem

Based on overview described above, then the research problem as follows :

1. How to implement VoR Tree on Highest Order Voronoi Diagram effectively.
2. How VoR tree influence Voronoi cell to find region on the execution time, depth of tree and node traversed.
3. How does VoR tree data structure effect on search performance compare to K-D Tree, Quadtree and Linear Search.

1.3 Objective

The purpose of this final project is as follows.

1. To implements VoR Tree on HSVD
2. To analyze the effect of VoR Tree on Voronoi cell to find region on the execution time, depth of tree and node traversed.
3. To analyze VoR Tree data structure effect on search performance compare to K-D Tree, Quadtree and Linear Search.

1.4 Scope

The scope of this final project is as follows.

1. Dataset using random uniform distributed, the meaning that the object data is spread evenly to all areas that are in the Voronoi diagram
2. Each random point has a region label
3. Leaf node has one point

1.5 Summary

In this chapter, describes the problems that have occurred previously. The problem that occurs is the Highest Order Voronoi Diagram can not be accessed directly for the region. Accessing can be done by grouping the point region using labeled and linear search. However, the performance search time becomes slow because must be checked one by one. Therefore, the index is important to resolve this problem. Index will split a partition in the region with point coordinates. This method can improve the execution time to discover the region.