A Spatial Data Mining Method by Delaunay Triangulation

In-Soo Kang, Tae-wan Kim, and Ki-Joune Li

Department of Computer Science, Pusan National University
Kumjeong-Gu, Jangjeon-Dong, Pusan, Korea 609-735
{iskang,twkim,lik}@chronos.cs.pusan.ac.kr
Tel : +82 51 582 1182
Fax : +82 51 515 2208

Abstract

It becomes an important task to discover significant pattern or characteristics which may implicitly exist in huge spatial databases, such as geographical or medical databases. In this paper, we present a spatial data mining method named *SMTIN* (Spatial data Mining by Triangulated Irregular Network), which is based on Delaunay Triangulation. *SMTIN* demonstrates important advantages over the previous works. First, it discovers even sophisticated pattern like nested doughnuts, and hierarchical structure of cluster distribution. Second, in order to execute *SMTIN*, we do not need to know a priori the nature of distribution, for example the number of clusters, which is indispensable to other methods. Third, experiments show that *SMTIN* requires less CPU processing time than other methods such as *BIRCH* and *CLARANS*. Finally it is not ordering sensitive and handles efficiently outliers.

1. Introduction

The database research community has been considerably attended to GIS(Geographic Information Systems) due to huge amount of spatial data[4]. Research focus of spatial databases has been concentrated on storing and retrieving spatial objects efficiently rather than analyzing pattern and distribution of spatial data. Recently, spatial database researchers have turned to their concerns on mining spatial objects[1]. Spatial data mining is the analysis of geometric or statistical characteristics and relationships of spatial data which may exist implicitly. The major approach of spatial data mining is how to cluster spatial data to discover implicit information. In terms of spatial data mining, cluster means grouping of relevant spatial objects.

Several requirements were proposed in spatial data mining techniques. First, they should be fast, since the amount of data they process is very huge. Second, they should provide rich information. Such information includes sets of spatial objects in a cluster, their shape and distribution, and statistical results like density, diameter, etc. Third, outliers should be treated properly. Outliers refer to spatial objects which are not contained in any cluster and should be discarded during mining process. But, when new spatial objects are inserted, these outliers must be considered, since outliers may form a cluster with the newly inserted objects.

Previous researches have been studied in distance-based or probability-based ways. Both approaches satisfy first and third requirements to some degree. But, the second requirement is not quite sufficiently satisfied so far. In this paper, we propose a spatial data mining method, *SMTIN* which has the following objectives.

- It concentrates on discovering the pattern of distribution of spatial objects and provides information of rich contents. For example, it shows whether the pattern contains holes and nested clusters.
- It should be stable. In other words, the insertion order of spatial objects does not influence on the results and it does not require any a priori knowledge.
- It should be fast.

This paper is organized as follows. Section 1 introduces this paper. We briefly investigate previous clustering methods in section 2. In section 3, *SMTIN* clustering algorithm is introduced. We show the characteristics of *SMTIN* method in section 4. And in section 5, we compare *SMTIN* with two important previous methods by experiments; *CLARANS* and *BIRCH*. Finally we conclude the paper and propose our future researches.

2. Related Works

Partitioning *N* objects into *k* clusters is one of major issues in statistics, and is called cluster analysis. It has been applied to many areas, such as medicine, psychology, archeology, etc. Clustering is defined as partitioning or grouping of relevant objects based on their attributes or geometric properties. Recently, this technique is adopted in spatial data mining fields. In this

GIS 97 LasVegas Nevada USA
Copyright 1997 ACM 1-58113-017-1/97/11..$3.50

[1] In this paper, we assume that the shape of spatial object is point.

section, we introduce three best-known methods which are based on cluster analysis. One common fact is that these methods use distance as a measure of clustering.

PAM[6] was developed to find *k*-medoids which represent *k* clusters. Medoid is a representative object that is the most centrally located in the cluster. *PAM* selects *k* objects arbitrarily as medoids and swaps repeatedly with other objects until all *k* objects qualify as medoids. The major disadvantage of *PAM* comes from the fact that it compares an object with the entire data set to find a medoid. This fact results in slow processing time, $O(k(n-k)^2)$.

CLARANS[8] was developed to overcome disadvantages of *PAM*. It uses sample data set to find medoids. Thus it needs less processing time at each step when it clusters objects into *k* medoids. *CLARANS* clusters objects around *k* medoids based on randomized search algorithm. It selects arbitrary *k* objects as current objects. And they are compared with sampled neighbor objects and swaps each other when one of neighbor objects qualifies certain conditions. *CALRANS* swaps repeatedly until it finds *k* medoids. *CLARANS* also exhibits several disadvantages. First, it is still slow since it uses randomized search algorithm, although it is faster than *PAM*. Second, it could not guarantee optimal clustering, due to its randomized approach. Third, *k*-medoids approach does not present enough spatial information when the patterns and the distribution of a data set are complex. Especially, it does not present hierarchical structure of a data set.

BIRCH[11, 12] improves the lacks of *CLARANS* in that firstly it supports localized clustering. At each clustering, it does not scan all spatial objects and refer all currently existing clusters. Second, it treats outliers in an efficient way. Third, it minimizes both CPU processing time and disk I/O time. Fourth, it inserts dynamically new spatial objects into existing clusters without modifying the all clusters. *BIRCH* clusters data by using *CF-Vector* (Clustering Feature) and *CF-Tree*. *CF-Vector* is a triple <*N*, *LS*, *SS*>, where *N* means the number of objects in a cluster, *LS* is *N* over a linear sum of each data object in a cluster, and *SS* is the square sum of each data object in a cluster. The elements of *CF-Tree* are *CF-Vector*. While *CF-Vector* contains distance relationships among spatial objects, *CF-Tree* contains information about clusters. Since the size of each node is small, *CF-Tree* can be loaded in main memory. It outperforms *CLARANS* in that it constructs *CF-Tree* in a single scan and accesses disk less than *CLARANS*. But the deficiencies of *BIRCH* are as follows. First, *BIRCH* concentrates on clustering spatial objects instead of finding patterns of distribution. Thus, it cannot provide enough information about complex and hierarchical patterns. Second, *BIRCH* generates different clusters for the same input data set according to the input order and selection of seed points. In other words, this method is order-sensitive and also seed-sensitive. Finally, we must have a priori knowledge about the nature of distribution, for example, the number of clusters and other input parameters, which is often unrealistic.

In order to overcome these shortcomings, we propose a spatial data mining method named *SMTIN* that is efficient and effective. It is efficient in terms of processing time and effective in a sense that it does not only cluster spatial objects according to the patterns of distribution but also presents complex and hierarchical patterns of distribution. While previous clustering methods use radius as a distance measure and calculate distance from a datum point, like *k* medoids in *CLARANS* and centroid X_0 in *BIRCH*, to other spatial objects, *SMTIN* clusters spatial objects as it traverses from any object to qualified neighboring spatial objects. The traversal property of *SMTIN* mainly comes from Delaunay Triangulation method. By the property of Delaunay Triangulation, *SMTIN* presents a cluster as an encompassing polygon, which makes possible to discover the shape and the hierarchical structure of clusters. We call an encompassing polygon a ***contour***. The motivation of our research lies in the facts that previous distance-based approaches such as *CLARANS* and *BIRCH* cannot cluster spatial objects as they are distributed and cannot present conceptually reasonable clusters of sophisticated and hierarchical data set. *SMTIN* is also motivated to generate clusters which are independent of the input data sequences and of seeds.

3. Clustering Spatial Objects by Delaunay Triangulation

According to [5], clustering of a set is a partition of its elements that is chosen to minimize some measure of dissimilarity. And the dissimilarity has been defined as diameter of cluster. Conventional clustering methods such as *PAM*, *CLARANS* and *BIRCH* therefore partition *N* spatial objects into *k* clusters with *k* mediods, so that the diameter should be minimized. Since they do not fully consider geometric properties of spatial objects and they rather rely on geo-statistical methods, they fail to discover geometric information like shape of clusters. But the information that spatial data mining discovers should include not only statistical regroupement but also geometric characteristics. This is a basic motivation of our approach.

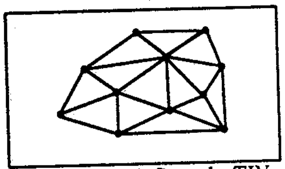
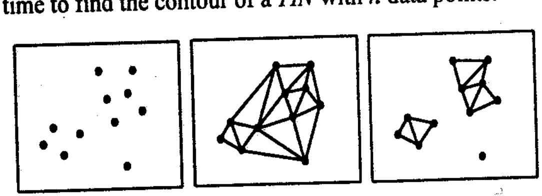


Figure 1. A Sample *TIN*

One of the efficient way to investigate geometric properties of spatial objects is Delaunay Triangulation [7], which is the dual graph of *Voronoi Diagram* [7]. And Delaunay triangles or the dual graph of Voronoi diagram are represented by *TIN* (Triangulated Irregular Network), where nodes represent spatial objects and edges means nearest couples among spatial objects as shown by Figure 1. It has an important property that the nearest objects to a given spatial object are always linked by edges, which allows us to analyze proximity relationship between spatial objects. By Euler's formula [7], *TIN* has at most $3n$-6 edges and $2n$-4 triangles for *n* input data points. We can see that it takes $O(n)$ time to find the contour of a *TIN* with *n* data points.



Initial Dataset Phase 1 Phase 2

Figure 2. *SMTIN* Procedure

Now, let us explain *SMTIN*. It consists of two phases; the first phase is building *TIN* from spatial objects. And on the second phase, we eliminate all edges whose distance is greater than a given threshold, T. Figure 2 shows the procedure of *SMTIN*.

```
Algorithm SMTIN
  1. Input spatial objects and construct their TIN.
  2. Remove edges whose length is greater than threshold $T$,
     and find connected components. Each connected
     component becomes a cluster.
  3. Remove clusters whose number of objects is less than a
     given number $n_0$.
  4. Find the contour lines of remaining clusters.
```

As a result, *SMTIN* generates clusters and their contour lines. If one or a number of spatial objects are isolated from other clusters, we consider them as outliers and exclude them from clustering. We can control outlier by n_0. If $n_0 = 1$, it means that we do not exclude any outliers. We can also control the granularity of clusters. By setting threshold T as great value, we can obtain coarse clustering, and when T is small, clusters contain small number of elements. Normally, we commence with relatively great threshold to see an outlook of clustering and by decreasing it, we get finer clustering.

Suppose that the number of input spatial objects is n. Then, step 1 of *SMTIN* requires $O(nlogn)$ time to generate Delaunay triangulation[9]. And since the number of edges is at most $3n$-6, the time complexity of step 2 is $O(n)$. It is obvious that it takes $O(n)$ time for step 3, because the maximal number of clusters does not exceed the number of objects. As explained previously, it takes $O(n_i)$ to find contour line of i-th cluster with n_i nodes, where k is the number of clusters and $n_1 + n_2 + ... + n_k \leq n$. Therefore for step 4, we need $O(n_1) + O(n_2) + ... O(n_k) = O(n)$ time. As consequence, it takes linear time of input size except step 1, and the total time complexity of *SMTIN* is $O(nlogn)$.

4. Clustering Sophisticated and Hierarchical Pattern

Spatial data mining methods that assume the shape of clusters are not adequate in analyzing geometric characteristics of spatial objects[3]. Since there is no general rule about the shape of clusters[10], cluster may be spherical, linear, or of other shapes. In comparison with previous spatial data mining methods, *SMTIN* does not assumes any a priori shape of distribution and clusters points as it visits arbitrarily around neighboring points. Therefore, it generates clusters as data points are distributed.

In the case when spatial objects are distributed in a very complex pattern, it works properly by virtue of the above property. For example, when houses are distributed along a lake, *SMTIN* traverses along the lake and clusters houses as a doughnut shape while letting inside of a doughnut empty. *SMTIN* demonstrates its strength when the shape of distribution is sophisticated and the distribution has a hierarchical structure.

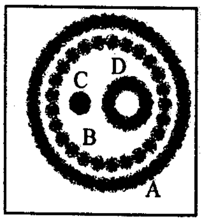
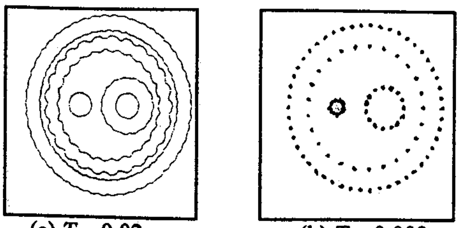


Figure 3. A sophisticated shape of distribution.

Figure 3 shows an example of sophisticated distribution with four clusters and it also implies a hierarchical structure. We can obtain clustering as Figure 4, with different thresholds. It clearly separates four clusters and discovers the original shape of clusters as Figure 4(a). And if we need more fine clustering, we can get a result as Figure 4(b) with smaller threshold.



(a) $T = 0.02$ (b) $T = 0.002$

Figure 4. *SMTIN* Clusters

We can also find a hierarchical relationship between two outputs. For example, cluster A in Figure 4(a) consists of several sub-clusters in Figure 4(b), which forms a hierarchical relationship. In the real world, we often find such relations and *SMTIN* is very helpful to analyze them. For example, the distribution of buildings in a city may contain the distribution of residential houses, commercial buildings, factories, and so on. In this case, *SMTIN* generates a cluster of buildings. And, it repeatedly generates several clusters of residential houses, commercial buildings and factories with a smaller threshold value. A tree for clusters may be constructed by iteration of threshold values range from T_1 to T_2 ($T_1 > T_2$).

5. Comparisons with *BIRCH* and *CLARANS*

In this section, we compare the performance and functions of *BIRCH* and *CLARANS* with *SMTIN* by experiments. We use Delaunay Triangulator [9] to implement *SMTIN*. We have used the same data sets of [11, 12] for the experiments, which are *DS1*, *DS2*, and *DS3* shown in Figure 5, 6, and 7. Each dataset consists of 100,000 points. The points in *DS3* are randomly distributed while *DS1* and *DS2* are distributed in grid and sine curve patterns, respectively. And their ordering are random, which favors *BIRCH* and *CLARANS* than *SMTIN*, since it is not order-sensitive.

We assume that the main memory is enough for loading the whole dataset. It is evident that this assumption favors *SMTIN* and *CLARANS* than *BIRCH*, since one of the advantages of *BIRCH* is reduction of the number of disk I/O. But, by using spatial access method such as *R*-tree* [1], we expect that the number of disk I/O could be considerably reduced and we plan to combine *R*-tree* with *SMTIN* for improving its disk I/O performance.

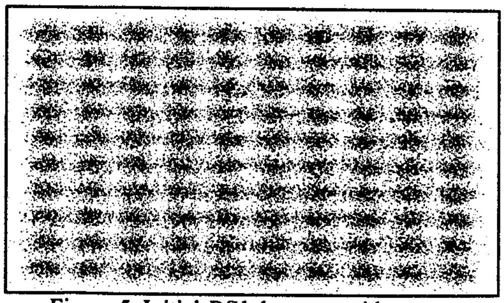


Figure 5. Initial *DS1* dataset : grid pattern

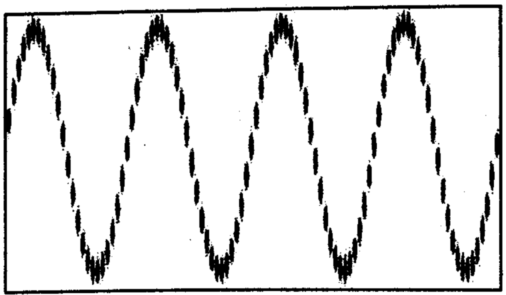


Figure 6. Initial *DS2* dataset : sine curve pattern

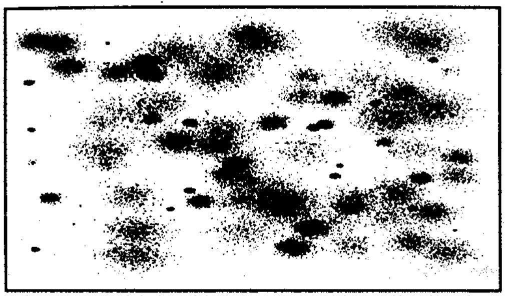


Figure 7. Initial *DS3* dataset : random distribution

Table 1 shows the execution time of three methods, *CLARANS*, *BIRCH*, and *SMTIN* on Sun UltraSPARC 2 with 128 M bytes main memory for *DS1*, *DS2*, and *DS3* respectively.

Data set	*CLARANS*	*BIRCH*	*SMTIN*
DS1 (grid)	1146.1	64.3	39.62
DS2 (sine)	923.2	56.4	39.39
DS3 (random)	1818.1	59.2	39.53

Table 1. Execution time of *CLARANS*, *BIRCH*, and *SMTIN*

In this table, we excluded disk writing time required by *BIRCH* to fairly compare the CPU processing time. By comparing them, we observe that: (1) *SMTIN* is faster than the other methods, *CLARANS* and *BIRCH*. (2) The execution time of *SMTIN* is independent of the distribution of spatial objects and consequently a stable clustering method. The reason is that Delaunay triangulation process mainly determines its execution time, which is almost independent from the nature of distribution.

In Figure 8, we find that: (1) The presentation method of the clustering results by *SMTIN* differs from those of *CLARANS* and *BIRCH*. The contour line of cluster in Figure 8c is actual shape of each cluster, whereas the ellipses in Figure 8a and 8b do not exactly correspond the contour lines of cluster. The numbers respectively represent the number of points in each cluster. (2) *BIRCH* and *SMTIN* cluster the points nearly same as the actual clusters, while *CLARANS* is far from them. (3) We have a priori defined the number of clusters for *CLARANS* and *BIRCH* as 100, which is actual number of clusters. It means that we must know the number of clusters a priori for *CLARANS* and *BIRCH*, which is often unrealistic. If we give an incorrect number of cluster, the result may be totally different from the actual. But we do not need to give the number of clusters for running *SMTIN* and in any way, it finds a correct clusters.

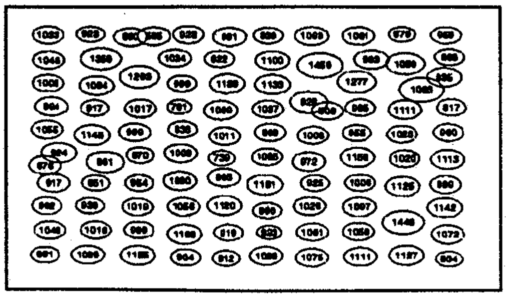


Figure 8a. *CLARANS* Clusters of *DS1*

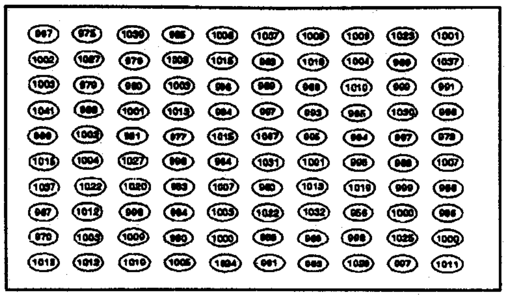


Figure 8b. *BIRTH* Clusters of *DS1*

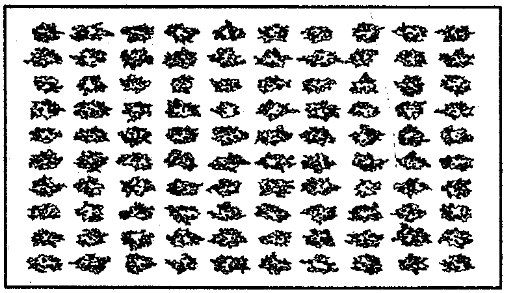


Figure 8c. *SMTIN* Clusters of *DS1*

We find a great difference between *SMTIN* and the other two methods in Figure 9. (1) While *BIRCH* and *CLARANS* discover only a set of local clusters as shown by Figure 9a and 9b, *SMTIN* discover a shape of clusters given by sine curve in addition to set of local clusters by applying different thresholds. For two thresholds T_1 and T_2, the clusters discovered by *SMTIN* applying T_1 and T_2 respectively are hierarchical as explained in the previous section. It means that we can find the shapes of distribution for several scales by *SMTIN* in a hierarchical way. (2) We observe that the local clusters with T_2 exactly correspond to the actual clusters. Obviously, it is important to find a proper threshold value. In order to find a proper threshold, we commence with relatively large threshold and decrease it gradually until we get a good clustering.

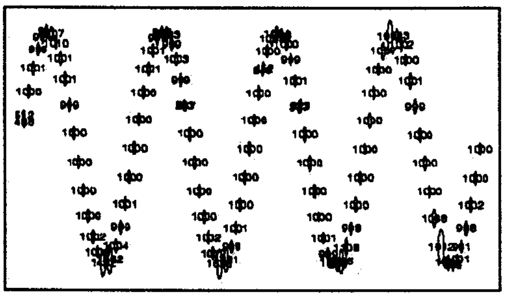


Figure 9a. *CLARANS* Clusters of *DS2*

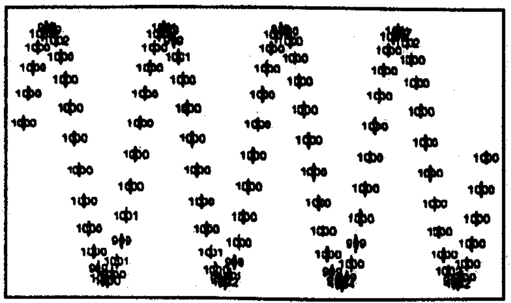
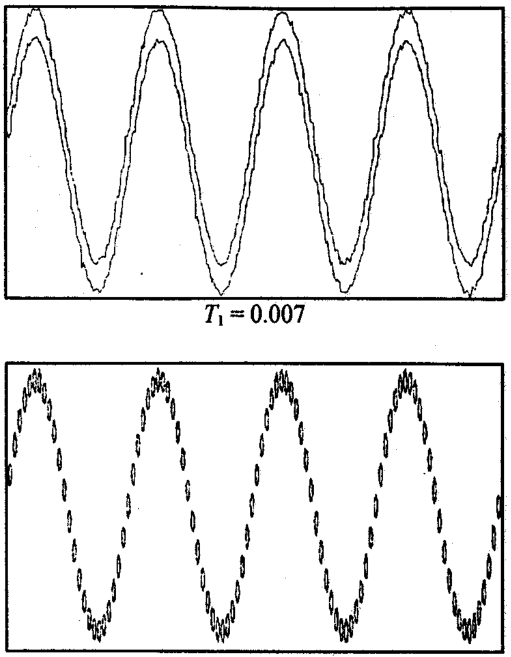


Figure 9b. *BIRTH* Clusters of *DS2*

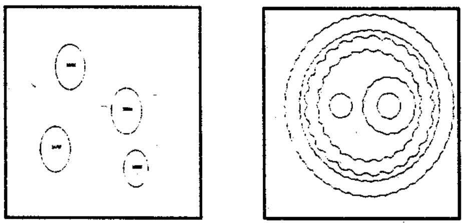


$T_1 = 0.007$

$T_2 = 0.0006$

Figure 9c. *SMTIN* Clusters of *DS2*

We finally compare *BIRCH* and *SMTIN* with dataset *DS4* given by Figure 3. The distribution of *DS4* is complicated and its shape is like nested doughnuts. Figure 10(a). and 10(b). respectively show the clusters discovered by *BIRCH* and *SMTIN*. In Figure 10(b), we see that *SMTIN* correctly carries out clustering with *DS4*, and each cluster is clearly separated from others. *BIRCH*, however fails to discover the actual clusters as shown by Figure 10(a).



(a) *BIRCH* Cluster (b) *SMTIN* Cluster

Figure 10. Clusters of *DS4*

6. Conclusion

In this paper, we presented a spatial data mining method, *SMTIN* which is based on Delaunay Triangulation. By comparison with other spatial data mining methods, such as *CLARANS* or *BIRCH*, it has the following advantages;

- It discovers rich information about the distribution of spatial objects, such as shape of clusters and hierarchical structure of cluster distribution, even though the distribution has sophisticated shape, like nested doughnuts.
- We do not need to know the nature of cluster distribution a priori, such as the number of clusters.
- It requires less CPU processing time than *CLARANS* and *BIRCH*.
- It efficiently handles outliers and is not ordering sensitive method.

An important drawback of our method is that it requires more disk I/O than *BIRCH*, which needs only one scan of spatial objects. We however expect that we could considerably reduce the number of disk I/O by using spatial access method[2]. Future works therefore include reconstruction of *SMTIN* on *R*-tree*. We will also extend our method to deal with non-point spatial objects such like lines and regions, as well as points.

Acknowledgements

This work was partially supported by the ministry of science and technology of South Korea under the contract of National GIS Development Project.

References

[1] Beckmann, N., Kriegel, H-P., Schneider, R., and Seeger, B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," *Proceedings of SIGMOD*, 1990, pp. 322-331

[2] Brinkoff, T. and Kriegel, H-P., "The Impact of Global Clustering on Spatial Database Systems," *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994, pp. 168 - 179

[3] Faloutsos, C. and Kamel, I. "Beyond Uniformity and Independence : Analysis of R-trees Using the Concept of Fractal Dimension," *Proceedings of ACM Conference Principles on Database Systems(PODS)*, 1995, pp.4-13

[4] Guenther, O. and A. Buchmann, "Research Issues in Spatial Database," ACM *SIGMOD Record, vol. 19, no.4*, 1990, pp. 61-68

[5] Hartigan, J.A., *Clustering Algorithms, John Wiley & Sons*, 1975

[6] Kaufman, L. and Rousseeuw, P.J., *Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley & Sons*, 1990

[7] Preparata, F.P. and Shamos, M.I., *Computational Geometry: An Introduction, Springer-Verlag*, 1985

[8] Raymond, T. Ng and Han, J., "Efficient and Effective Clustering Methods for Spatial Data Mining," *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994, pp. 144 - 155

[9] Schewchuk, J.R., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," *Proceedings First Workshop on Applied Computational Geometry*, ACM, 1996.

[10] Seber, A.J., *Multivariate observations, John Wiley & Sons*, 1984

[11] Zhang, T., Ramakrishnan, R., and Livny, M., "BIRCH: An Efficient Data Clustering Methods for Very Large Databases," *Proceedings of ACM SIGMOD International Conference of Management of Data*, 1996, pp. 103 -114

[12] Edwin M. Knorr and Raymond T. Ng, "Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 8, No. 6, 1996, pp. 884-897