The Role of Visualization in Effective Data Cleaning

Yu Qian
Dept. of Computer Science
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
qianyu@student.utdallas.edu

Kang Zhang
Dept. of Computer Science
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
kzhang@utdallas.edu

ABSTRACT
Using visualization techniques to assist conventional data mining tasks has attracted considerable interest in recent years. This paper addresses a challenging issue in the use of visualization for data mining: choosing appropriate parameters for spatial data cleaning methods. On one hand, algorithm performance is improved through visualization. On the other hand, characteristics and properties of methods and features of data are visualized as feedbacks to the user. A 3-D visualization model, called *Waterfall*, is proposed to assist spatial data cleaning in four important aspects: dimension-independent data visualization, visualization of data quality, algorithm parameter selection, and measurement of noise removing methods on parameter sensitiveness.

Keywords
Information visualization, data mining, clustering, noise removal.

1. INTRODUCTION
Capturing and storing data is becoming increasingly easier than analyzing and exploring data. Advanced data processing techniques have thus attracted growing research interest in recent years. Among these techniques, data mining or *knowledge discovery in database* (*KDD*), which refers to the non-trivial process of discovering interesting, implicit, and previously unknown knowledge from large databases [5], has been widely applied to many communities. According to Fayyad and Uthurusamy [7], data mining is primarily concerned with making it easy, convenient, and practical to explore very large databases for organizations and users with lots of data but without years of training as data analysts. A data mining technique should allow users to get insights from the data by extracting patterns or models or relationships among the data that can be easily interpreted and understood. A promising approach along this direction is data visualization. The goal of visualization is to provide qualitative insight into data, processes, and concepts through the use of the visual pattern recognition ability humans possess [16]. Visualization has become an increasingly important component in data analysis for its ability to provide rich overviews and a visual detection of patterns and outliers. Visualization could bridge the two most powerful information-processing systems: human and computer. Humans are limited in the ability of handling scale and are easily overwhelmed by the volumes of data. Data mining, aiming at processing large amount of data automatically, could complement human abilities. Combining the two approaches for knowledge discovery is clearly promising [6].

SAC'05, March 13-17, 2005, Santa Fe, New Mexico, USA.

As a primary data mining technique, clustering is the process of grouping a set of objects into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters [8]. Many clustering algorithms aim at discovering patterns in noisy environments and offers abundant noise removing methods for pattern discovery, most of which, however, require trial-and-error parameter tuning processes for the thresholds used in the noise removing methods. According to Han *et al.* [8], clustering algorithms can be classified into hierarchical method, partitioning method, density-based method, and grid-based method. An important observation is that most representative density-based approaches use a similar pair of parameters on noise removal, which provides us a chance to design a common model for density-based methods to visualize the relationship between noise removing results and corresponding parameters. Since the method-dependent axes of the 3-D model represent only the algorithm parameters, a reassignment of the axes with parameters of a different method can provide similar visual support. The proposed model, called *Waterfall*, typically representing the visualized data like waterfalls, can assist high dimensional noise removal. The waterfall model does not visualize higher dimensional data directly. It instead visualizes the relationship between the output of the data preparation method and the two algorithm parameters in a 3-D coordinate system. Since the data preparation method separates the given data and removes noise, visualizing its output provides an approximate overview on the data distribution. Further, the relationship between the two method parameters can be reflected through a 2-D projection of the produced 3-D graph on the surface decided by the two axes corresponding to the two method parameters. If different combinations of the two parameters produce similar noise removing results, the noise removing method could be insensitive to parameter selection; otherwise, the noise removing method is parameter sensitive. Thus we can evaluate the quality of a noise removing method through a straightforward visualization. Finally, parameter tuning becomes easy. The appropriate pair of the two parameters can be selected through checking the visualization of the algorithm output and the relationship between the two parameters. Compared with direct-plotting approaches, waterfall model has several features: first, it suits high-dimensional data. Second, it allows a quick and approximate overview on data distribution. Third, it discovers the relationship between algorithm parameters. Last, it can measure the parameter sensitiveness of a noise removing method. Waterfall visualization model has been applied to the data clustering framework FAÇADE [12] as effective visual support for noise removal and parameter tuning, which is publicly available at http://viscomp.utdallas.edu/FACADE.

The rest of the paper is organized as follows. Section 2 introduces the background knowledge about spatial noise removal and clarifies our motivation. Section 3 presents waterfall model and its properties. The experimental results of three benchmark data sets are demonstrated in Section 4. Section 5 concludes the paper.

2. BACKGROUND AND MOTIVATION

This section will examine representative noise removing methods and clarify why waterfall model is necessary and applicable to most density-based clustering approaches.

To remove noise effectively, the first task is to define noise. A common way of defining noise is to find features that can be used to distinguish noise from true data. Commonly used features include distances between data points, sizes of data groups, and number of connections among data points in graph-based approaches. Accordingly, existing noise removing methods can be classified into three categories: distance-based, size-based, and degree-based. Typical distance-based noise removal appears in DBSCAN [4], the first density-based clustering method. DBSCAN uses two parameters: *Eps* and *MinPts* for its noise removal. The neighborhood within a radius *Eps* of a given object is called the *Eps*-neighborhood of the object and an object with at least *MinPts* of objects within its *Eps*-neighborhood is called a core object. Then noise is defined as non-core objects which do not lie within the *Eps*-neighborhood of any core object. Although distance-based approaches can discover points distributed sparsely as noise, thresholds like distance between data points are usually difficult to set for different data sets without prior knowledge. OPTICS [1] is proposed to assist DBSCAN on parameter selection. Although OPTICS reduces the impact on clustering result of inappropriate parameter selection, it still requires *Eps* and *MinPts* and cannot compare different sets of parameters. Other clustering methods that use similar parameters include DENCLUE [10].

To overcome the problem of lacking prior knowledge, size-based approaches, like Birch [17] and RandomWalk [9], perform cluster analysis on data sets first and then produce clusters of small sizes as noise. Birch regards clusters with sizes less than 1/4 average as noise while RandowWalk uses half of the average as the threshold. Size-based approaches still require prior information about the ratio of noise to true data; otherwise, the threshold on cluster size for distinguishing noise clusters from true clusters is hard to obtain. To avoid another weakness of DBSCAN of treating sparse true data as noise, a degree-based noise removing method is proposed in SNN [3]. It constructs a shared nearest neighbor graph on the given data set and decides if a data point is noise based on the degree of its corresponding vertex, i.e., the number of its nearest neighbors instead of distances among them. Degree-based approaches can discover sparse true data while it still requires trial-and-error parameter setting on two parameters: k for constructing the k shared nearest neighbor graph and a threshold d for regarding points with degree less than d as noise.

Core-based noise removal [13] is a recent progress on spatial data cleaning, which prepares the data for visualization through a two-step approach: first modeling the given data set with a k-mutual neighborhood graph; then applying a fast graph partitioning method, the k-core algorithm [2, 14], to decompose the k-mutual neighborhood graph into small groups of data points. With the core-based noise removing technique, two parameters affecting the result are k_m and k_c. To study the relationship between the output of the k-core algorithm and the two parameters, we started our experiments by try-and-error on many benchmarks. Our experiments reveal an important phenomenon: for a wide range of k_m, there exists a corresponding k_c so that the pair (k_m, k_c) removes noise effectively. This phenomenon, however, does not help solving the parameter selection problem because: first, this property may not be general enough for all kinds of data sets. Second, even if k_m can be chosen freely, it is difficult to select the corresponding k_c. Therefore, a visualization model supporting the selection of both parameters is needed. Observing that similar situation exists for all aforementioned clustering methods, we propose the waterfall model to visualize the relationship between the parameters and the algorithm output. The two parameters are mapped to two axes of a 3-D coordinate system. The ratio of the size of the data after noise removal to the original data size represents the third axis.

Another motivation of devising waterfall model is due to the high-dimensional data. While high dimensional data becomes more and more popular in real applications, the data properties are hard to discover because straightforward visualization techniques cannot be applied. An important property of waterfall model is its independence from the dimensionality of the given data because it visualizes the noise removing method instead of data.

3. PARAMETER TUNING

This section will demonstrate a waterfall visualization model through two benchmark data sets used by CHAMELEON [11], and a comparison between two representative noise removing methods: DBSCAN and Core-based noise removal. The waterfall model will visualize the relationship between the algorithm output and the parameters so that the user can select right parameters intuitively. Waterfall model not only avoids direct visualization of higher dimensional data but also provides an overview of the data characteristics useful for noise removal. Section 3.1 will describe coordinate assignment of the 3-D model while Section 3.2 will discover more features of both data and methods through a 2-D projection. Section 3.3 will provide a visual comparison between DBSCAN and core-based noise removal.

3.1 The Waterfall Visualization Model

The waterfall model has four graphical dimensions: x-axis, y-axis, z-axis, and color, corresponding to k_m, k_c, the ratio of the data sizes before and after noise removal (denoted by r), and the five colors simulating a waterfall. The colors, from light green to dark blue, correspond to the amount of true data. The more true data identified, the bluer the graph appears. This allows a quick capture of the data quality, i.e. if the waterfall appears to be blue, it means the data set contains little noise.

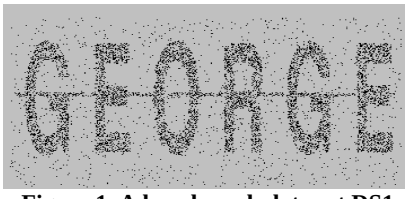


Figure 1. A benchmark data set DS1.

Figure 1 is a benchmark data set used in CHAMELEON [11], a representative method. Figure 2 shows the resulting visualization for DS1. To save computation time, each time we increase the value of k_m by ten or k_c by five and find how many data points are removed. The increment of k_m and k_c determines the size of the mesh grid, as shown in Figure 2. The relationship between the algorithm parameters and the output can be described as follows.

When fixing k_m, the size of the remaining data decreases as k_c increases. The dropping of the size of the remaining data becomes dramatic when the true data has been removed, which produces a waterfall style of visualization. The sudden dropping can be justified as follows. Because true data usually are of large size and have similar properties on distance or shape, the parameter change will make either few of them or most of them removed, which will not be a gradual process. When k_m increases, the connections among both noise and true data will increase, thus a corresponding bigger k_c is required for the removal of the same size of noise.

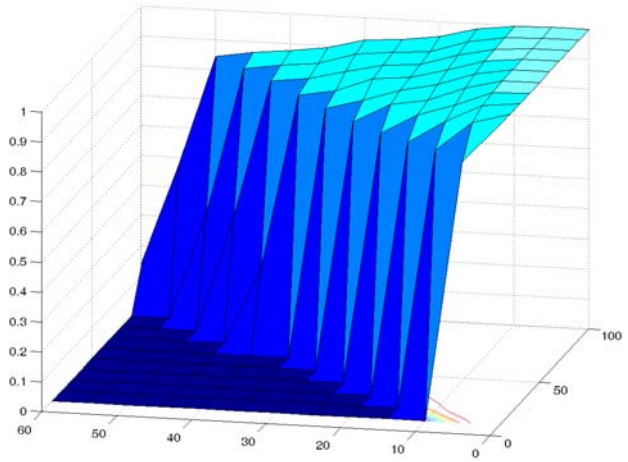


Figure 2. The waterfall model for DS1 with the core-based noise removal.

The waterfall visualization, on one hand, supports our hypothesis that for a wide range of k_m there is a k_c to remove noise effectively, on the other hand, detects the noise ratio in this data set is about 15% because the waterfall drops down dramatically at z=0.85. This is very useful for data mining methods as prior knowledge about the given unknown data. The visualization can also measure the quality of a noise removing method. If a noise removing algorithm is not sensitive to its parameters, for a wide range of one of the parameters, there should exist a value for the other parameter so that the pair can remove noise correctly. Shown in the visualization, the waterfall should drop at similar place if the algorithm is insensitive to parameter selection because the noise ratio is a fixed value for a given data set. For example, in Figure 2 all lines drop at about 0.85.

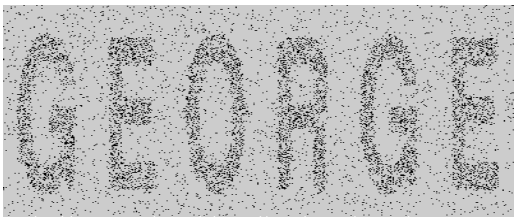


Figure 3. DS2: another benchmark data set.

Generally, the shape of the waterfall is affected by both the method and the data set. Figure 3 shows DS2, another benchmark data set. Its waterfall model, illustrated in Figure 4, shows a different waterfall outline from that of DS1. The comparison between Figures 2 and 4 indicates DS2 has a little higher noise ratio than DS1.

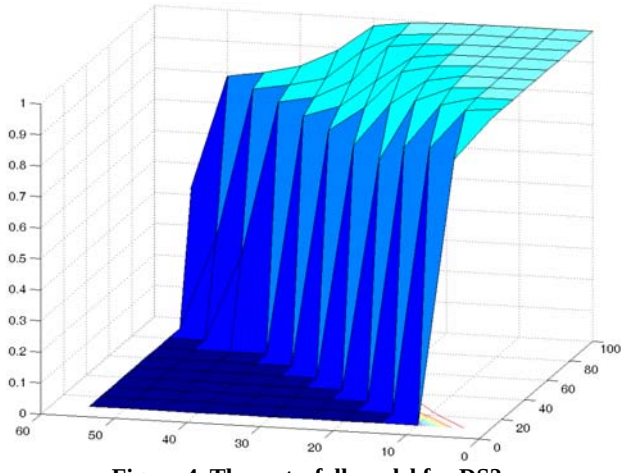


Figure 4. The waterfall model for DS2.

All features of waterfall can be seen clearly through two 2-D projections on the back surface and on the bottom surface, which will be presented in the following section.

3.2 2-D Projections

We first project the 3-D waterfall of DS1 to the surface decided by *z*-axis (r) and *y*-axis (k_c), i.e., the back surface, as illustrated in Figure 5.

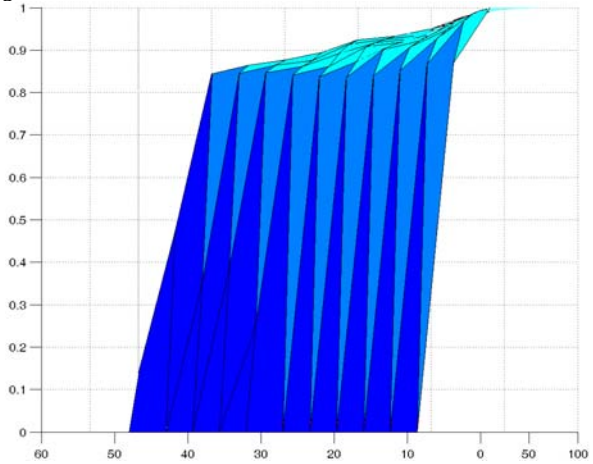


Figure 5. The 2-D projection of DS1 on the back surface.

Similarly, DS2 also has the projection, as shown in Figure 6. From the comparison between Figures 5 and 6, we can discover information about the DS1 and DS2. Firstly, the graph of DS1 has a bigger blue area than that of DS2, so DS1 contains less noise. Secondly, the waterfall of DS1 drops at about z=0.85 while that of DS2 drops at about z=0.77, which represents the approximate noise ratios of them. Thirdly, all lines of the waterfall of DS1 drop at similar place while those of DS2 vary a little, which implies the parameter selection for DS2 is harder than for DS1. Lastly, the waterfall of DS2 drops two times, which implies there exists a large amount of noise in DS2 which differs greatly from the true data on distribution, which makes the noise removal a sudden drop instead of a gradual process as k_c increases. This matches the generation process of DS2 exactly because DS2 is a combination of the true data of DS1 and 3500 random noise points which has a much smaller density compared with the true data. Thus the first

drop represents the removal of these noise points totally and the second drop removes the true data. In contrast, DS1 contains the thick line which reduces the density difference between noise and the true data, so the removing process is a gradual one.

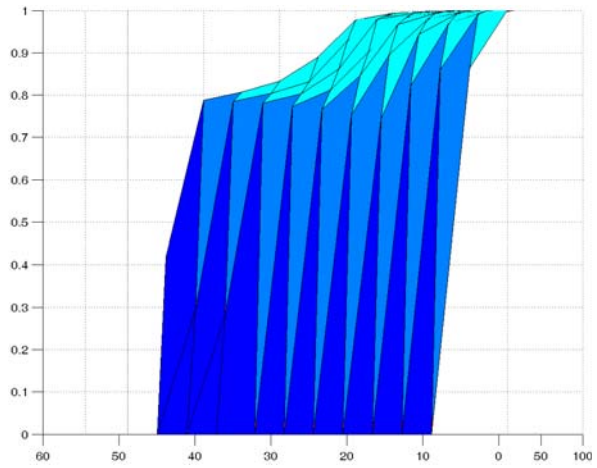


Figure 6. The 2-D projection of DS2 on the back surface.

Now let us study the relationship between the two algorithm parameters through another 2-D projection on the surface decided by *x*-axis (k_m) and *y*-axis (k_c), i.e., the bottom surface. The 2-D projection is obtained through cutting the waterfall horizontally at the dropping position. For example, for DS1, we cut the waterfall with the horizontal surface z=0.85 and get the slice, as shown in Figure 7, in which we can find that the relationship between k_m and k_c is very close to the function $k_m = 2\ k_c$. This relationship can be used to select the parameter when the other parameter can be fixed. All pairs of (k_m, k_c) satisfying $k_m = 2\ k_c$ would remove the 15% noise effectively. By checking the slice of the waterfall model, users are freed from the complicated parameter tuning job.

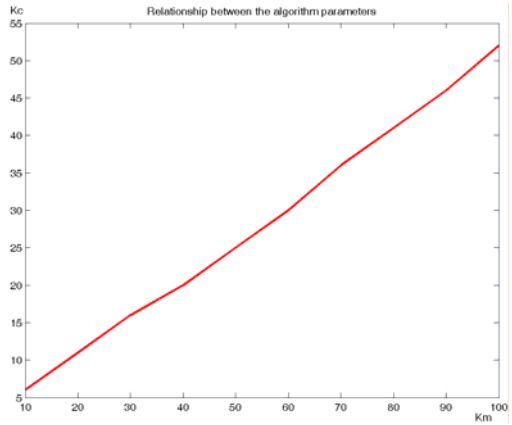


Figure 7. The relationship between k_m and k_c for DS1: a projection on the bottom surface.

3.3 Visual Comparison of Different Clustering Methods

Using similar coordinate assignments, the waterfall visualization model can be applied to most density-based noise removing methods. We produce one for DBSCAN through the following mappings: *Eps* and *MinPts* are mapped to *x*-axis and *y*-axis, respectively, and z-axis still represents the ratio of the size of the remaining data to the original data. The 3-D waterfall visualization for DS2 with DBSCAN is shown in Figure 8 while the 2-D projection on the back side is in Figure 9.

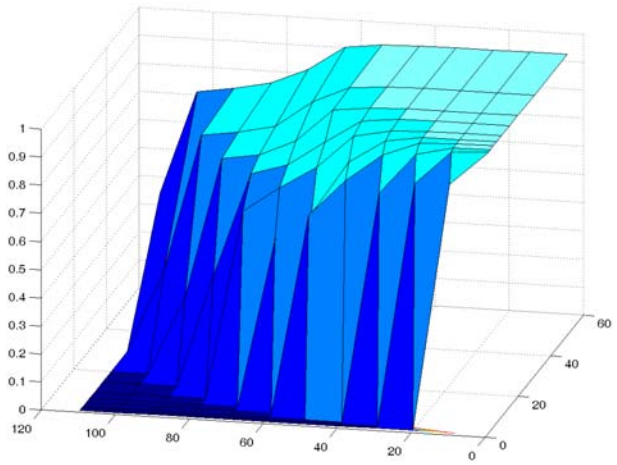


Figure 8. The waterfall model of DS2 with DBSCAN.

A comparison between the visualizations of the two different methods leads to the measurement of the methods on the parameter sensitiveness. The inconsistent dropping shown in Figures 8 and 9 indicates that DBSCAN is more parameter-sensitive than the core-base noise removing approach.

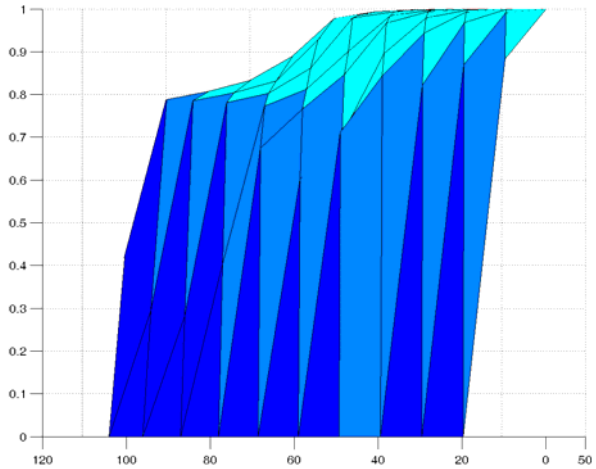


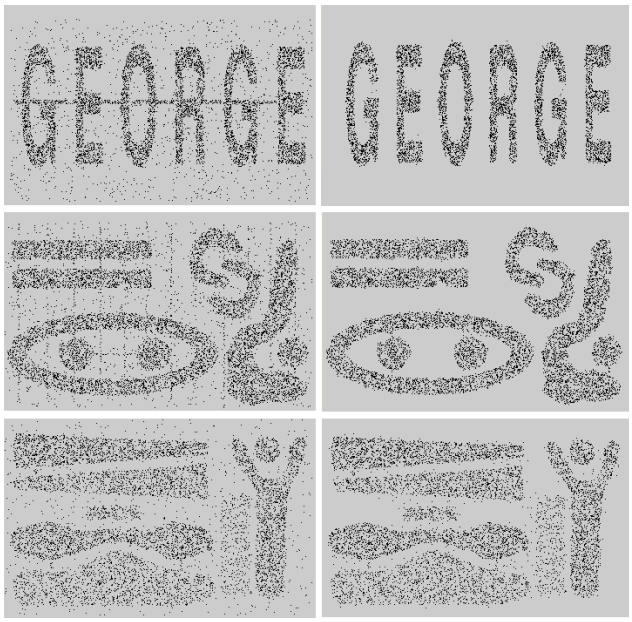
Figure 9. The 2-D projection of the waterfall of DS2 with DBSCAN.

In summary, the waterfall visualization has four advantages:

- Supporting high-dimensional data.
- Showing the approximate noise ratio for a given data set.
- Measuring the quality of a noise removing method on parameter sensitiveness.
- Producing a relationship graph between the two algorithm parameters to assist parameter tuning.

4. EXPERIMENTAL RESULTS

This section will provide the noise removing results guided by waterfall model through performing core-based noise removal on three benchmark spatial datasets used by CHAMELEON [11]. More experimental results can be found at http://viscomp.utdallas.edu/FACADE.



(a) **(b)**

Figure 10. Three data sets (a) before, and (b) after noise removal.

Figure 10 (a) shows the three data sets and their corresponding results are shown in Figure 10 (b). Waterfall model successfully detects the threshold k_c for different shapes and sizes of noise and assist core-based noise removal to separates noise from true data effectively.

5. CONCLUSION

Spatial databases are an important subclass of multimedia databases [15]. Parameter tuning has always been a challenging issue in spatial data clustering. Many current data clustering algorithms hardwire some algorithm parameters to the values that are identified through adhoc and try-and-error experiments. An important observation is that most of these algorithms require two parameters to complete their noise removing processes and the two parameters appear to be related to each other on the output of the algorithm. If we can work out a visualization way to assist the parameter tuning for one of these methods, it can also be applied to other methods. This paper presents a visualization-based approach to solve the parameter tuning problem. A 3-D visualization models, waterfall, is proposed to assist noise removing methods on parameter tuning. The waterfall model supports high-dimensional data and can be used to measure the data quality by outputting the noise ratio of the data set. The appropriate selection of algorithm parameters can be shown in a 2-D projection of the waterfall model for different data sets. Also, two different noise removing methods can be measured and compared at the same time by applying the waterfall model on them. Our experiments demonstrate that the waterfall model can effectively assist core-based noise removing method and similar approaches on threshold selection to produce very clean results. Future work will continue along two directions: applying the waterfall model to real data sets and measure algorithm performance, and discovering more properties of the waterfall model and see if more clustering methods can be visualized through it.

REFERENCES

[1] Ankerst, M., Breunig, M., Kriegel, H. P., and Sander, J., OPTICS: Ordering Points To Identify the Clustering Structure, in *Proc. of 1999 ACM-SIGMOD Conf. on Management of Data (SIGMOD'99)*, pp. 49-60.

[2] Batagelj, V., Mrvar, A., and Zaversnik, M., Partitioning approaches to clustering in graphs, *Proc. Graph Drawing'1999*, LNCS, 2000, pp. 90-97.

[3] Ertoz, L., Steinbach, M., and Kumar, V., Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, In *Proc. of SIAM DM'03*.

[4] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proc. of 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, 1996, pp. 226-231.

[5] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT press, 1996.

[6] Fayyad, U. and Grinstein, G., *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann, 2001, pp. 182-190.

[7] Fayyad, U. and Uthurusamy, R., Evolving data mining into solutions for insights, *Communications of ACM*, 45 (8), 2002, pp. 28-31.

[8] Han, J., Kamber, M., and Tung, A. K. H., Spatial clustering methods in data mining: A survey, H. Miller and J. Han (eds.), *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis, 2001.

[9] Harel, D. and Koren, Y., Clustering spatial data using random walks, In *Proc. 7th Int'l Conf. Knowledge Discovery and Data Mining (KDD-2001)*, ACM Press, New York, pp. 281-286.

[10] Hinneburg, A., and Keim, D. A., An efficient approach to clustering in large multimedia databases with noise. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining*, pp. 58-65.

[11] Karypis, G., Han, E., and Kumar, V., CHAMELEON, A hierarchical clustering algorithm using dynamic modeling, *IEEE Computer*, Vol. 32, 1999, pp. 68-75.

[12] Qian, Y., Zhang, G. and Zhang, K., FACADE: A Fast and Effective Approach to the Discovery of Dense Clusters in Noisy Spatial Data, in *Proc. ACM SIGMOD 2004 Conference*, Paris, France, 13-18 June 2004, ACM Press, pp. 921-922.

[13] Qian, Y. and Zhang, K., Discovering spatial patterns accurately with effective noise removal, in *Proc. 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'04)*, Paris, France, 13 June 2004, ACM press, pp. 43-50.

[14] Seidman, S. B., Network structure and minimum degree, *Social Networks*, 5, 1983, pp. 269-287.

[15] Shekhar, S., Schrater, P. R., Vatsavai, R. R., Wu, W., and Chawla, S., Spatial contextual classification and prediction models for mining geospatial data, *IEEE Trans. on Multimedia*, Vol. 4, No. 2, pp. 174-188.

[16] Ward, M. O. and Zheng, J., Visualization of spatio-temporal data quality, in *Proc. of GIS/LIS*, 1993, pp. 727-737.

[17] Zhang, T., Ramakrishnan, R., and Linvy, M., BIRCH: an efficient data clustering method for very large databases, in *Proc. ACM SIGMOD 1996 Conference*, ACM Press, pp.103-114.