# Effective Clustering & Boundary Detection Algorithm Based on Delaunay Triangulation

Dongquan Liu, Gleb V. Nosovskiy, Olga Sourina

**Abstract**

In this paper, a new spatial clustering algorithm TRICLUST based on Delaunay Triangulation is proposed. This algorithm treats clustering task by analyzing statistical features of data. For each data point, its values of statistical features are extracted from its neighborhood which effectively models the data proximity. By applying specifically built criteria function, TRICLUST is able to effectively handle data set with clusters of complex shapes and non-uniform densities, and with large amount of noises. One additional advantage of TRICLUST is the boundary detection function which is valuable for many real world applications such as geo-spatial data processing, point-based computer graphics, etc.

*Key words:* Clustering Algorithms, Data Mining, Delaunay Triangulation

## 1 Introduction

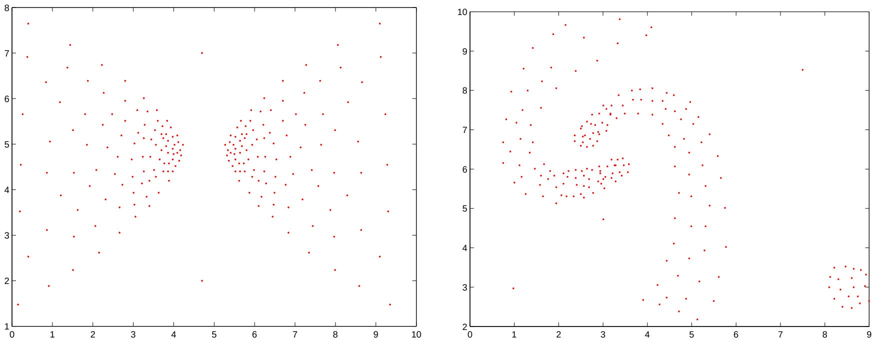
As spatial databases have been used in more and more areas, such as geo-spatial data processing, biomedical data analysis, point-based graphics, etc, the size of spatial database increases dramatically and the structure of data becomes more and more complicated. To discover valuable information in those databases, spatial data mining techniques [1,2] have been paid significant attention during recent years. Clustering [3-5], or unsupervised learning, plays an indispensable role in spatial data mining. Thus, a variety of spatial clustering methods have been developed.

In general, clustering is a method to separate data into groups without prior labeling so that the elements inside the same group are most similar as long as elements belonging to different groups are most dissimilar. We can roughly divide the main existing spatial clustering methods into five categories: partition methods [6,7], hierarchical methods [8-10], density-based methods [11-13], graph-based methods [14-16], and learning-network methods [17,18]. All these approaches have been proven successful in dealing with different data sets in different application domains. But there are still limitations of the most existing famous clustering algorithms.

First, the performance of classic clustering algorithms always relies on users specified parameters or prior knowledge of data. The traditional partition method K-MEANS [6] and its derivative methods [19] need the user to input the number of clusters in advance. Hierarchical methods like CURE [10] and BIRCH [9] are sensitive to the values of pre-set parameters related to merging condition. Model-based partition methods [20,21] make assumptions on data distribution to achieve good results. But due to the increase of complexity of data, it is very difficult to provide best fit parameters or assumptions before clustering process. Moreover, most existing methods use global parameters, for example, such as the $EPS$ value for density-based method DBSCAN [11] and the $MinPts$ value for OPTICS [13]. The drawback is that global parameters lack the ability to apply different discrimination criterion on different parts of data with the specific local information. When the data distribution becomes complex, by only employing global parameter, clustering algorithm can hardly achieve the best result. The combination of both global and local information should be considered when designing parameters.

Second, since the shape of expected clusters becomes more and more complicated, the demand for clustering algorithm to detect clusters with arbitrary shapes rises. Most existing clustering methods can not deal with clusters of irregular shapes or clusters with different sizes, like CLARAS [19] and BIRCH. The density-based methods DBSCAN and DENCLUE [12] are able to discovery clusters with irregular shapes, but their abilities are still limited to handling clusters of similar densities. In the work of new graph-based methods, such as AMOEBA [15] and AUTOCLUST [16], the situation where clusters have different density is studied. Cluster with high density surrounded by cluster with low density can be correctly detected. However, when we think of this problem further, we can notice that the density of clusters could be different not only between clusters but also between the different parts of the same cluster. In Fig.1, a 2-dimensional example of this situation is given. Although the density varies inside those clusters, just by visual inspection, we can easily discover those non-uniform clusters because of the gradually changing densities. Moreover, clusters with non-uniform density widely exist in real world applications such as geo-spatial data processing and point-based graphics [22-25]. But most existing clustering algorithms lack the capacity to analyze this kind of data set. Thus, it is necessary to design a new method to fit this new requirement.

Third, for real world applications, in addition to the variety of data distributions, there are always noises introduced by systems during data collection or data transformation procedures. The distribution of noise also significantly influences the result of clustering. If the noises are isolated points, the ef-



(a)

(b)

Fig. 1. Examples of clusters with non-uniform density

fect of them may be relatively trivial. Many methods such as DBSCAN and Single-linkage hierarchical method, can deal well with the noises. But if the noises form a chain which connects two clusters that should be separated from each other, by using the above methods, the result will be totally not expectable. This kind of problems is the so-called "short bridges" problem [16,26] or "chaining effect" problem [3,27], or "multiple bridges" [16] problem if there are several chains formed by noises. The well-known solution is cut-point finding which only works for single bridge cases.

Last, the cluster boundary detection becomes more and more valuable for many real world applications, such as geo-spatial data analysis and point-based computer graphics. But there are quite few clustering methods with this function. Moreover, from data classification prospective, knowledge of cluster boundary makes it possible to classify new data without repeating the whole clustering process [7]. Thus, methods with boundary detection will have the advantage to deal with dynamic data sets.

In this paper, we treat clustering task by analyzing statistical features of data and design our novel effective clustering algorithm TRICLUST using triangulation. The proposed method TRICLUST is able to automatically detect not only clusters with non-uniform inner density but also with short bridges formed by noises. By developing this algorithm, we can effectively deal with a wider range of data sets which have different distributions of data density and noises. TRICLUST can also build the cluster boundaries by identifying the data points on them for all clusters.

The rest of this paper is organized as following. In Section 2, the relation of TRICLUST to previous methods is described. In Section 3 and 4 the definitions used in TRICLUST and the basic idea of it are given. In Section 5 and 6, the algorithm description and theoretical analysis are provided. In Section 7 and 8, the performance comparisons with other clustering methods and real world application are given. In the last section, conclusions are drawn and future work suggestions are given.

## 2 Relationship to Previous Works using Triangulation

The Delaunay Triangulation or Delaunay Diagram [28-30] is a method used to build topology of data set. It represents proximity relationships of data by building the connected graph. It has the important properties [14,31] such as the nearest data points to a given spatial data point are always connected by edges, which provide us a good description of proximity, and the circumcircle (circumshpere for higher than 2-dimensional case) of every triangle does not contain any other data points. It represents data topology explicitly by building the succinct graph. Based on it, we can conveniently extract information such as statistical features from the graph. This kind of information can effectively characterize the relationship between data points so that it can lead to good clustering method.

There are a few methods developed based on Delaunay Triangulation, such as the method proposed by Eldershaw and Hegland [14], AMOEBA [15], AUTOCLUST [16] and the method proposed by Hader and Hampercht [32]. The approach of Hader and Hampercht applies Delaunay Triangulation as a path building tool. After the graph is generated, along the paths found, the value of the density function is checked to find the local maxima in order to start the density-based clustering process. The approach proposed by Eldershaw and Hegland as well as AMOEBA and AUTOCLUST mainly focus on classifying the edges of the graph built by Delaunay Triangulation or Delaunay Diagram into several groups. Then, by removing the inter-cluster edges, clusters are isolated. Among these previous approaches, AUTOCLUST is the latest and the most effective one. It applies Delaunay Diagram on the data set, then by analyzing the statistic information extracted from the graph built by Delaunay Diagram, unusually long and short edges are removed. After the edges recovery and bridges looking phases, the data set is separated into clusters and outliers[1] . This method has an advantage that the values of the parameters can be found from the statistic information of the edges of the graph. It makes AUTOCLUST an effective clustering method. Comparing to its prior method AMOBEA, AUTOCLUST has the ability to deal with more complicated data sets, such as in which clusters are connected by multiple bridges and/or nearby clusters are with different density, but the density inside these clusters is still relatively uniform.

The proposed algorithm TRICLUST is also based on Delaunay Triangulation.

[1] In this paper, when we analyze clustering results, we call a point which does not belong to any clusters an outlier no matter it is a data point or a noise point.

But our method has the following novel features:

(1) **Data set with clusters of non-uniform density**

First, we extended our concern on data distribution to non-uniform inner cluster density cases, which have been introduced in Section 1. Based on the needs risen recently in both research and application areas, we developed a new algorithm which can be applied on wider range of data sets. Thus, the variation of distribution of data density between and inside clusters is one of our main concerns.

Second, if the density of data set varies not only between clusters but also inside clusters, the classification of edges of the graph will be more difficult, since the long edges could connect data points of the thin part of the same cluster, and short edges could connect local outliers with nearby clusters. The discrimination of edges in the way how it was done in AMOBEA and AUTOCLUST, would be much more difficult in such situation, moreover, for some complex data distribution, it could just fail. Therefore, we use data point investigation instead of edge investigation in our algorithm. The data points are classified into different categories according to the statistic information which is extracted from their neighborhood which are built by triangulation. An advantage of studying on data points rather than edges is that we can minimize the effect of density variation and achieve a more clear description of data distribution. In addition, the time complexity will decrease during the checking and searching procedures, since, in general, there are more edges than data points in a graph built by triangulation.

(2) **New statistic features**

In order to characterize data points based on data distribution, we designed a few statistic features. Different from the ones used in previous methods, we consider on data point instead of edges together with the effect of density changing both inter clusters and inner clusters. The target which we study on and extract statistic information from is the length of all edges inside the neighborhood of each data point. The neighborhood for each data point is the sub-graph of Delaunay Triangulation. The statistic features we applied here are mean of the length, standard deviation of the length divided by mean, and the positive part of the derivative of mean. The details of definitions of these statistic features are given in Section 3.

## 3 Definitions

Here, we regard each data point as a point $P$ in the $n$-dimensional data space $\mathbf{R^n}$. The data set $D$ is a set of $n$-dimensional points $D = \{P_1, \ldots, P_N\}$, and $N$ is the number of data points.

**Definition 1** *Neighborhood:*

*The neighborhood $Ne$ of one data point $P$ is the sub-graph of the Delaunay Triangulation of the data set. This neighborhood is constructed by all data points which are directly connected to $P$ based on Delaunay Triangulation and all edges between those data points. We call two data points directly connected if they are linked by the same edge of Delaunay Triangulation.*

**Definition 2** *Mean of one data point:*

*The Mean of one data point $P$ is the mean of the length of all edges inside its neighborhood $Ne$. $Mean(P)$ is defined in 1.*

$$Mean(P) = \sum_{L_i \in Ne} |L_i|/M_e = \sum_{i=1}^{M_e} |L_i|/M_e \tag{1}$$

*where $|L_i|$ denotes the Euclidean length of edge $L_i$, and $M_e$ denotes the number of edges in $Ne$.*

**Definition 3** *Standard deviation ($STD$) of one data point:*

*The standard deviation of one data point $P$ is the standard deviation of the length of all edges inside $P$'s neighborhood $Ne$. It is defined in 2.*

$$STD(P) = \sqrt{\sum_{i=1}^{M_e} (Mean(P) - |L_i|)^2/(M_e - 1)} \tag{2}$$

*where $Mean(P)$ is the mean of data point $P$.*

**Definition 4** *The quotient of standard deviation ($STD$) divided by Mean of one data point:*

*The quotient of $STD$ divided by Mean of one data point $P$, which we denoted as $DM$, is defined in 3.*

$$DM(P) = STD(P)/Mean(P) \tag{3}$$

**Definition 5** *The $PDM$ value of one data point:*

*The $PDM$ value of one data point $P_i$ is the mean of positive parts of the derivative of the Mean along all edges connected $P_i$ to its neighboring data points in its neighborhood $Ne_i$. The $PDM$ is defined in 4.*

$$PDM(P_i) = \sum_{j=1}^{M_p} PD(P_i, P_j)/M_p \tag{4}$$

*where $PD(P_i, P_j)$ is the positive part of the derivative of Mean along the edge connected $P_i$ and $P_j$. $PD$ is defined in 5; $M_p$ is the number of data points in $Ne_i$ with smaller $Mean$ value than $Mean(P_i)$.*

$$PD(P_i, P_j) = \begin{cases} 0, & if \quad Mean(P_i) \leq Mean(P_j) \\ \frac{Mean(P_i) - Mean(P_j)}{|L|}, & if \quad Mean(P_i) > Mean(P_j) \end{cases} \tag{5}$$

*where $|L|$ is the length of the edge connected $P_i$ and $P_j$.*

## 4 Basic Ideas

The basic idea of TRICLUST, which is an effective clustering algorithm based on Delaunay Triangulation, is to cluster data set by classifying all data points into two categories. These two categories are inner cluster data points and boundary data points. By using proposed statistic features extracted from the neighborhood of data points, we build the criteria function according to both global view and local view for different situations. We use K-MEANS method as the threshold detecting method to choose the threshold of our criteria function. The classification of data points is executed by applying this threshold. Thus, the $n$-dimensional clustering problem can be transferred to one dimensional classification problem. We will give the description of our statistic features in more detail in this section with 2-dimensional examples.

Let us elaborate on basic ideas of TRICLUST.

- **Statistic features calculated based on triangulation**

  The statistic features we employed in TRICLUST to classify one data point are the $Mean$ of it, $DM$ which is the quotient of standard deviation of it divided by its $Mean$, and $PDM$ which is the mean of the positive parts of the derivative of $Mean$.

  To illustrate the meaning of the value of each statistic feature, we consider the different locations of one data point inside its data set. In Fig.2, we show all possible locations for one data point in a 2-dimensional data set. The data point drawn with circle represents the situation when the data point is located in the dense part of the cluster. The data point drawn with triangle represents the situation when the data point is located in the thin (low density) part of the cluster. The data point drawn with diamond represents the situation when the data point is located on the cluster boundary. The data point drawn with square represents the situation when the data point is an outlier.

  To give a clearer view of the meaning of statistic feature values in situations shown in Fig.2, we listed all situations with the corresponding neigh-

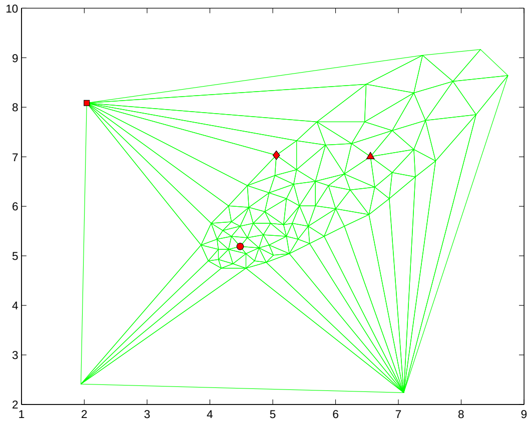


Fig. 2. Illustration of all possible locations of one data point in the data set

borhood graph of the data point in Fig. 3. From the sub-pictures (A) to (H), the neighborhood graphs of the center red points which are the data points we are investigating into are shown. In each situation, the red center point corresponds to the data point drawn with different symbols shown in Fig.2. The neighborhood graphs are built by Delaunay Triangulation. The situations of (A) and (B) represent the situations when the center data points are located in the dense part of a cluster. The situations of (C) and (D) represent the cases when the data points are in the thin part of a cluster. The situations of (E) and (F) represent the cases when the center data points are on the boundary of one cluster. And (G) and (H) represent the cases when they are outliers. We provided two sub-pictures for each situation to illustrate that, as the center data point is in the same situation, the different positions of the neighboring data points will not affect our study on the statistic features extracted from its neighborhood.

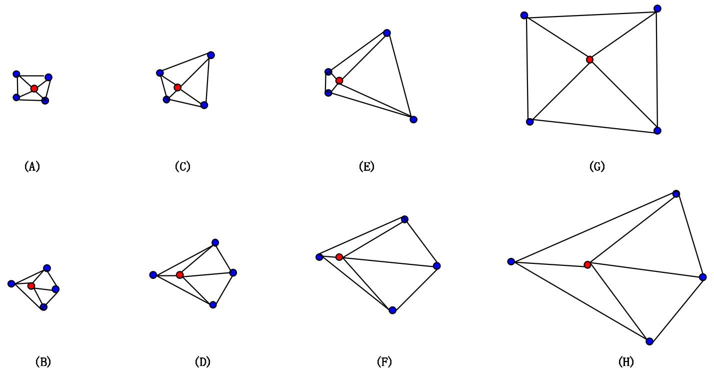


Fig. 3. Different neighborhood graphs of the center data point regarding its different locations in the data set

From the four types of situations shown in Fig.3, we can notice the differences between those neighborhood graphs so that we are able to study the variation of values of the statistic features defined in Section 3. For situations in sub-picture (A) and (B), since the center data points are in the dense part of a cluster, the $Mean$ of them and the standard deviation $STD$ of them are both with small values, also the value of their $DM$ and the value of their $PDM$. For the situations in sub-picture (C) and (D), since the center data points are located in the thin part of one cluster, the values of $Mean$ and $STD$ of them become bigger than the ones in situation (A) and (B). But the values of $DM$ will not increase with the same extent, because the scale of the increase of $STD$ and $Mean$ is the same. The values of $PDM$ of the center points in these situations will also increase. For the situations shown in (E) and (F), the center data points are on the boundary of one cluster, thus, values of the $Mean$ and the $STD$ of them will both increase. But they will not increase to the same extent, there will be more increase of the values of $STD$ than of $Mean$. Therefore, the values of $DM$ will increase greatly because of the different scales between increase of $DM$ and $STD$. The values of $PDM$ of the center data points will also increase dramatically since the sudden change of the derivative of $Mean$ occurs. For situations (G) and (H), since the center data points represent outliers, the values of $Mean$ of them will be much bigger than the values in previous situations. The values of $STD$ and $DM$ will also have big values. The values of $PDM$ in these situations will be also big. In general, they will be quite similar to the values in situation (E) and (F), since the values of the $Mean$ of both the center data points and their neighbors which are data points on cluster boundaries are big.

- **Criteria Function**

  The three statistic features we used in TRICLUST have different characteristics with respect to their capacities to reflect the data distribution. $Mean$ can distinguish inner cluster data point from boundary point in the case when the densities of clusters are uniform and similar to each other. But for non-uniform density clusters or clusters with different densities, such as when sparse clusters are near to dense clusters, it can not work well. On the other hand, $DM$ is a good statistic feature for detecting non-uniform density clusters. It has the strong ability to identify boundary data points when both uniform and non-uniform density clusters exist. But it is very sensitive to the irregular data distribution inside clusters. It means when there are several data points inside the same cluster with quite different distributions from others, such as when they are more close to each other than to other data points, these data points may be identified as boundary points by their $DM$ value. $PDM$ is more robust to the irregular data distribution inside clusters than $DM$. But for detecting boundary data points, it is not as effective as $DM$, especially when ”short-bridges” exist.

  Another important issue for modern clustering research is applying different criteria on different parts of data set according to both global and local information of it. It means that clustering data set under global view or local view only can not achieve ideal result. We should combine the global and local information based on certain data distribution and clustering requirement.

To maximize the advantage of each statistic feature and consider both global and local information when we cluster our data set, we build the criteria function by combining our three statistic features with different weights. The values of these weights reflect the degree to which we want to cluster the data set under global view or local view. In general, the bigger size of the data set is, the more important is the global information; the smaller size of the data set is; the more important is the local information.

Since the $Mean$ of data point can serve as a good criterion under global view, we let it play a more significant role when the size of our data set is big. On the contrary, $DM$ is a local and very sensitive criterion, when the size of our data set is small, it should be in charge. $PDM$ is a good complement of $DM$ with regard to dealing with irregular inner cluster data points. And it is better than $mean$ to detect boundary data points. So the importance of $PDM$ should increase when $DM$ is not in charge. Thus, by combining the three statistic features into one criteria function, we can have good discrimination in classifying data points according to both global and local views so that our algorithm TRICLUST is able to effectively detect clusters with non-uniform density and is robust to noises, even when the "short bridges" are existing.

The criteria function $f_c$ is defined as in 6, the value of it becomes the final feature for every data point. For data point $P$, $f_c(P)$ is:

$$f_c(P) = a * Mean(P) + b * DM(P) + c * PDM(P) \tag{6}$$

where $a, b, c$ are the weights which are defined in 7.

$$a = \begin{cases} \frac{N}{2000}, & N < Smin \\ \frac{1.9N + Smax/10 - 2Smin}{Smax - Smin}, & Smin \leq N < Smax \\ 2, & N \geq Smax \end{cases}$$

$$b = \begin{cases} 1, & N < Smin \\ 1 - \frac{N - Smin}{2(Smax - Smin)}, & Smin \leq N < Smax \\ 0.5, & N \geq Smax \end{cases} \tag{7}$$

$$c = \begin{cases} 0.5, & N < Smin \\ \frac{0.5N + 0.5Smax - Smin}{Smax - Smin}, & Smin \leq N < Smax \\ 1, & N \geq Smax \end{cases}$$

where $N$ is the number of data points inside the data set.

The values of parameter $a$, $b$ and $c$ reflect the level to which we want to cluster our data set under local or global view. If there is a small amount of data points that one can notice any of the data points "at one glance", then the natural clustering should depend exclusively on the local details of the data points distribution. Moreover, the small number of data points is not enough to generate reliable global features of distribution which could be useful for clustering. We denote $Smin$ as the number of data points, starting from which human visual inspection starts missing individual points and therefore the manner of discrimination starts shifting from local to global one. Another threshold is $Smax$ the number of points, starting from which global features of the data set become fully important for human visual inspection. These two thresholds should be estimated by biological research of human vision for eye simulation cases or be selected based on requirements of certain applications. Here, they are very approximatively set to $Smin = 200$ and $Smax = 10000$, but the clustering results are quite robust to their variations, which will be shown in Section 8.

## 5 Algorithm Description

The main steps of TRICLUST are elaborated in this section.

**Step 1.** Apply the Delaunay Triangulation on our data set to get the triangulation graph of it. The length of all edges for each data point is also calculated.
**Step 2.** Calculate the three statistic features for each data point according to the definitions introduced in Section 3.
**Step 3.** Set the values of parameters $a, b, c$ of the criteria function based on the method proposed in Section 4. For every data point $P_i$, we calculate the value of its final feature $f_c(P_i)$ which is the value of criteria function on the data point $P_i$. After that, we get the distribution of the final feature values for all data points by drawing the frequency histogram. We find the value $R_c$ of the frequency histogram. $R_c$ is defined in 8.

$$\begin{aligned} R_c &= \begin{cases} R_1, & \text{if} \quad N \leq 5000 \\ min\{R_2, R_1\}, & \text{if} \quad N > 5000 \end{cases} \qquad (8) \\ R_1 &= \min\{x : g^{f_c}_{emp}(x) = 0\} \\ R_2 &= \min\{x : P_{emp}(f_c < x) = 0.97\} \end{aligned}$$

where $g^{f_c}_{emp}(x)$ denotes the empirical density (frequency histogram) of the $f_c$ distribution. $R_1$ is the first zero. $R_2$ is the 97% of the distribution. The 3% cut is for algorithm robustness when dealing with large data set. We select the final feature values less than $R_c$ for threshold deciding in the next step. In Fig.4, we provided an illustration of the frequency histogram of final feature.

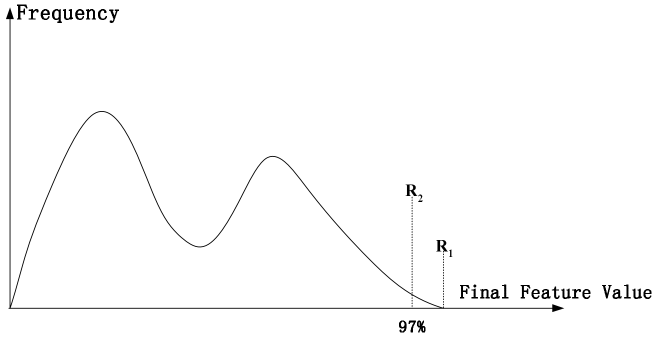


Fig. 4. An illustration of frequency histogram of final feature

**Step 4.** By applying K-MEANS algorithm (the parameters setting for K-MEANS: number of clusters equal to 2; initial cluster centers are the minima and mean of the selected final feature values) on the final feature values achieved from Step 3, we can get the threshold $th$ which is used to distinguish boundary data points from inner cluster data points. The data points with the final feature value bigger than $th$ are labeled as boundary data points (this category also includes outliers). The rest data points are labeled as inner cluster points.

**Step 5.** We start the clustering process by selecting the first inner cluster data point $P_i$ based on indexing. A new cluster $C_j$ is generated by assigning this data point $P_i$ to it as its first element.

**Step 6.** The cluster expansion progress is executed by continuously adding the inner cluster data points in the neighborhoods of the assigned data points. After the first data point $P_i$ is assigned to the new cluster $C_j$, we initialize the cluster neighborhood $Ned_j$ which is a set of data points. Initially, it includes the data points in the neighborhood of $P_i$. We find all the inner cluster data points inside $Ned_j$. We assign these data points to $C_j$ and update the cluster neighborhood $Ned_j$ by adding data points in the neighborhoods of all newly added data points. The achieved cluster neighborhood $Ned_j$ does not include all data points which have already been assigned to cluster $C_j$. After that, we add the inner cluster data points in $Ned_j$ to cluster $C_j$ and update $Ned_j$ until there is no suitable data point to be assigned to cluster $C_j$. During the cluster expansion progress, we update the index of one data point when it is assigned to a cluster.

**Step 7.** Repeat Step 5 and Step 6 until no new cluster can be generated. Till now, the clustering process complete. All data points have been assigned to clusters or labeled as outliers.

**Step 8.** If we want to build complete cluster boundaries for boundary detection applications, besides the data points that have been already labeled as boundary points, the boundary data points which are only connected to their own clusters are also labeled. In 2-dimensional case, the data points which are the ends of such edge that belongs to only one triangle are also labeled as boundary points.

## 6 Complexity Analysis

Since we are mainly interested in spatial data which is usually low dimensional, the analysis here is for 2-dimensional cases. The time complexity of constructing Delaunay Triangulation graph is $O(NlogN)$, where $N$ is the number of data points. The time complexity of finding neighboring data points and calculating the length of edges in the neighborhood for all data points is linear to $N$. The most time consuming procedure is cluster neighborhood updating. Because average number of neighboring data points to one data point is bounded [16,31], the time complexity of this procedure can be linear to $N$. So the total time complexity of TRICLUST is $O(NlogN)$.

The experimental testing of the time complexity of TRICLUST is shown in Fig.5. The $X$ direction is the number of data points. The $Y$ direction is the CPU time required by TRICLUST in second.

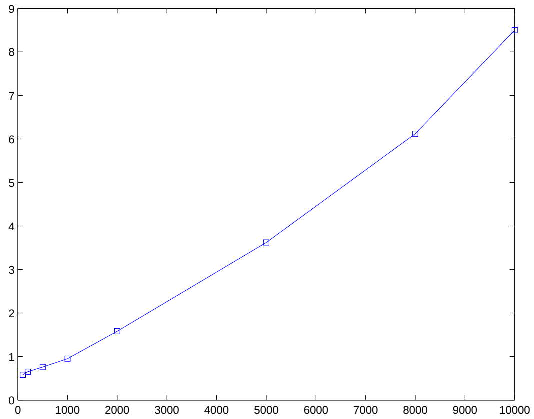
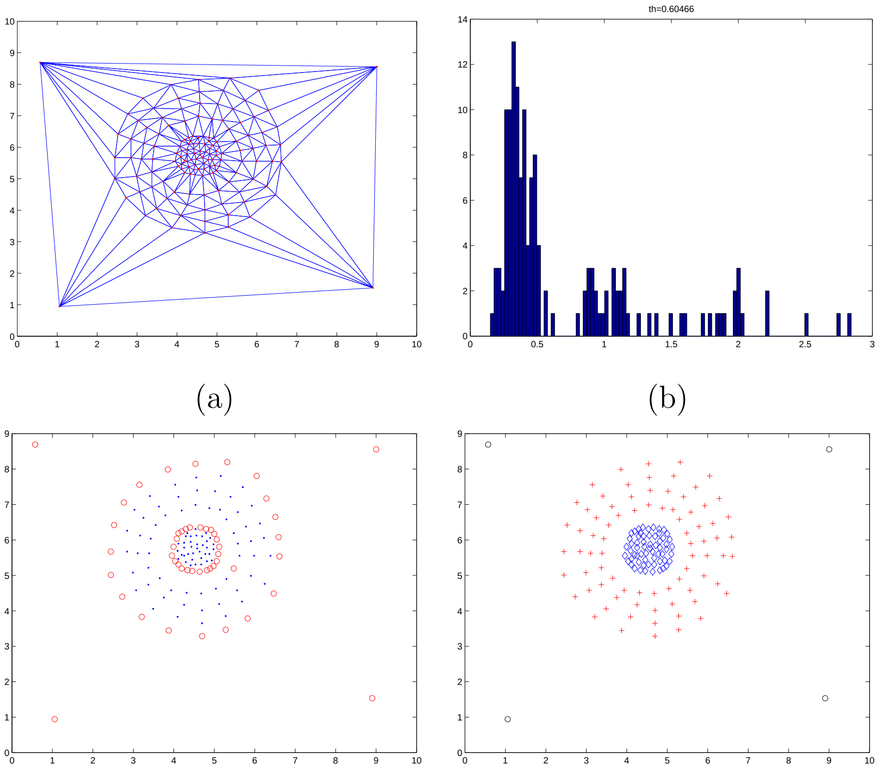


Fig. 5. Time required by TRICLUST in seconds

## 7 Comparisons and Experimental Results Analysis

To show the capability of TRICLUST to handle data with complex distribution, we designed two 2-dimensional testing data sets $D1$ and $D2$. The new challenges for clustering algorithms from the distribution of data, such as non-uniform cluster density, complicated cluster shape, and short bridges built by noises, are implemented in these testing data sets. By showing clustering results, the cluster boundaries built by TRICLUST, and the distributions of final feature values, the performance and characteristics of TRICLUST are well demonstrated.



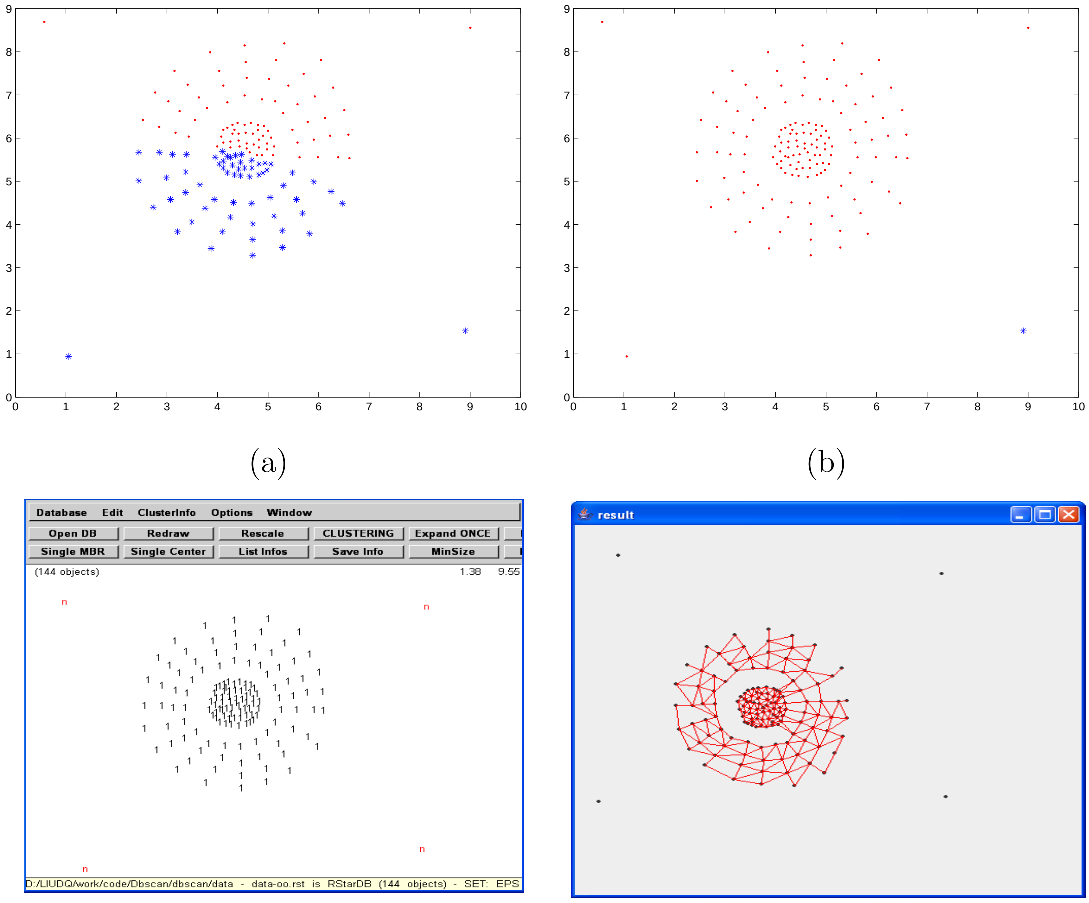
(a) (b)

(c) (d)

Fig. 6. The testing on data set $D1$ of TRICLUST: (a)The graph built by triangulation of $D1$; (b)The distribution of final feature values of $D1$; (c)The boundary data points of $D1$ detected by TRICLUST; (d)The clustering result of $D1$ by TRICLUST

For comparison, we also applied classic clustering methods such as K-MEANS, hierarchical Single Linkage, DBSCAN and AUTOCLUST on $D1$, $D2$ data sets. We choose these classic clustering methods because they are representative in this area, and standard implementations of these methods are available. We can avoid the potential problems causing by different implementations, which could lead us to bias in results comparisons. K-MEANS method is still the most widely used method, and it is based on the optimization principles which form a foundation for many well-known data mining techniques. We employed the squared Euclidean distance and random selection of initial cluster centers for K-MEANS in our tests. Single Linkage hierarchical method has a good ability of handling clusters with complicated shapes. DBSCAN is another well known density-based clustering method for dealing with arbitrary shape clusters and outliers [2] . The implementation of DBSCAN is provided by its author and all parameter settings are based on the instruction given in the original

[2] In this paper, when we analyze clustering results, we call a point which does not belong to any clusters an outlier no matter it is a data point or a noise point



(a)

(b)

(c)

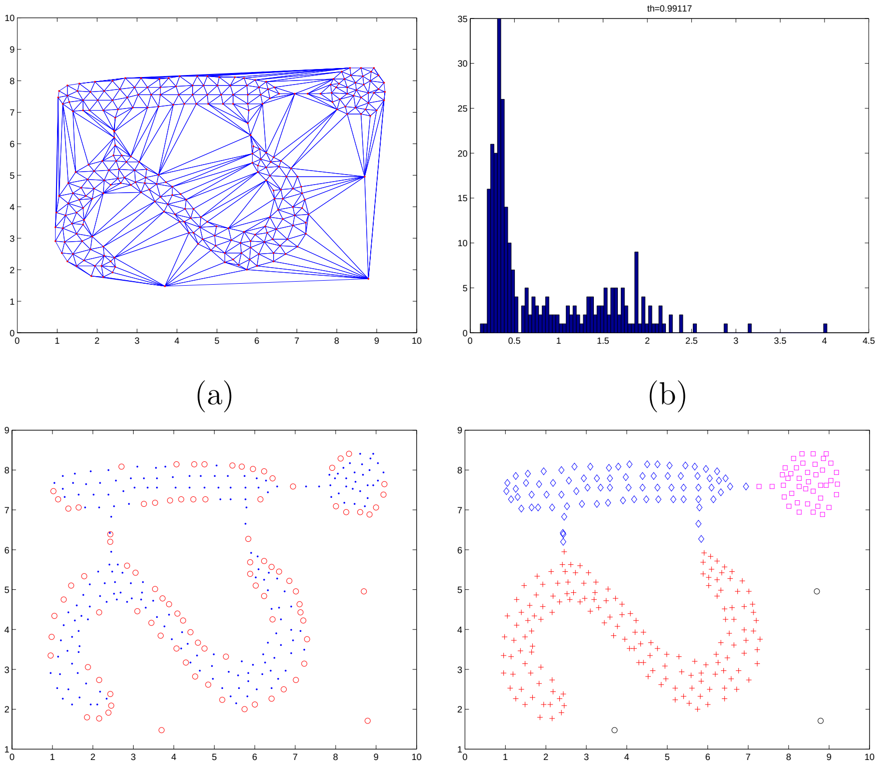
(d)

Fig. 7. Clustering results of data set $D1$ by comparison methods: (a)Clustering result of $D1$ generated by K-MEANS; (b)Clustering result of $D1$ generated by Single-linkage algorithm; (c)Clustering result of $D1$ generated by DBSCAN, where $EPS = 0.7937$; (d)Clustering result of $D1$ generated by AUTOCLUST algorithm

paper ($k = 3$ and $MinCard = 4$ for our 2-dimensional data sets). AUTOCLUST is one famous clustering algorithm based on Delaunay Triangulation. The implementation of AUTOCLUST is obtained from the GEOTOOLS package (http://geotools.codehaus.org/).

In Fig.6 and Fig.8, the Delaunay Triangulation graph of testing data set $D1$ and $D2$, the boundary data points of clusters in $D1$ and $D2$ detected by TRICLUST, distribution of the final feature values of $D1$ and $D2$, and clustering results of testing data sets $D1$ and $D2$ by TRICLUST are provided.

In data set $D1$, there are two clusters inside this data set together with four outliers. The small round cluster is surrounded by the large ring shape cluster. The densities of these two clusters are quite different that the round cluster is denser than the ring shape cluster. Moreover, the density of ring shape cluster is non-uniform comparing with the density of another cluster. These



(a)

(b)

(c)

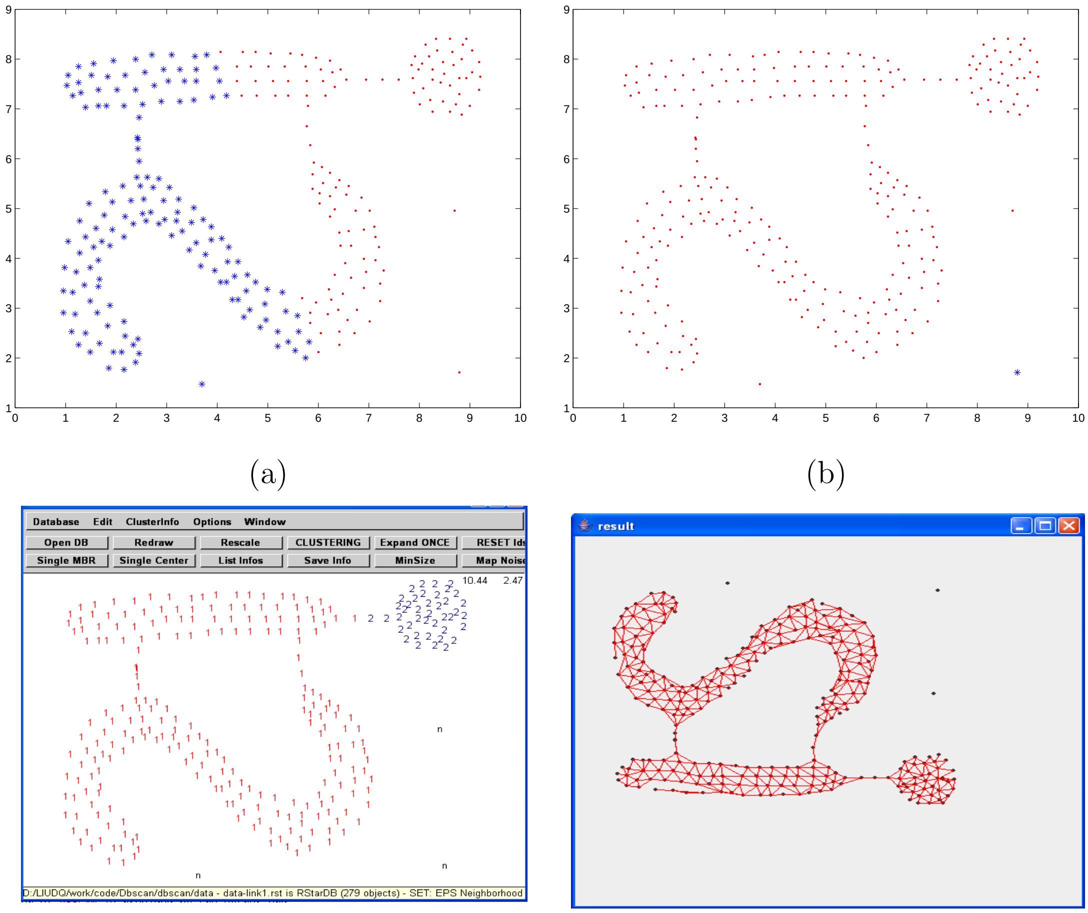
(d)

Fig. 8. The testing on data set $D2$ of TRICLUST: (a)The graph built by triangulation of $D2$; (b)The distribution of final feature values of $D2$; (c)The boundary data points of $D2$ detected by TRICLUST; (d)The clustering result of $D2$ by TRICLUST

two clusters are also very close to each other so that the smallest distance between data points belonging to different clusters smaller than the largest distance between data points of the ring shape cluster. From Fig.6, we can notice that TRICLUST separated the two clusters very well and also found the outliers.

In Fig.7, the clustering results of $D1$ by K-MEANS, Single Linkage method, DBSCAN and AUTOCLUST are shown. None of them can get the ideal result. K-MEANS separated the clusters into half and joined parts of two clusters together. Single Linkage method, DBSCAN and AUTOCLUST are not able to distinguish these clusters from each other at all. Comparing with the results shown in Fig.6, the advantage of TRICLUST to handle clusters with non-uniform density is well demonstrated.

In data set $D2$, there are three clusters together with three short bridges and two isolated outliers. Comparing with the previous data sets, data set $D2$ is more challenging because not only the clusters are with complex shapes but also short bridges exist. From the clustering result by TRICLUST shown in Fig.8, the good performance of it on $D2$ is clearly demonstrated.



D:/LIUDQ/work/code/Dbscan/dbscan/data - data-link1.rst is RStarDB (279 objects) - SET: EPS Neighborhood

(a)

(b)

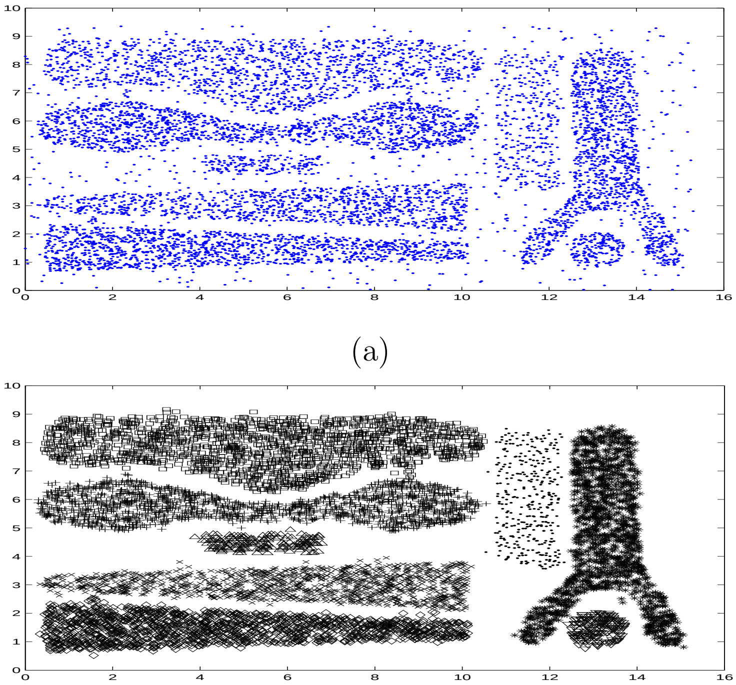
(c)

(d)

Fig. 9. Clustering results of data set $D2$ by comparison methods: (a)Clustering result of $D2$ generated by K-MEANS; (b)Clustering result of $D2$ generated by Single-linkage algorithm; (c)Clustering result of $D2$ generated by DBSCAN, where $EPS = 0.7598$; (d)Clustering result of $D2$ generated by AUTOCLUST algorithm

In Fig.9, the clustering results of $D2$ by four comparison clustering methods are shown. K-MEANS method can not discover two of the three clusters. Single Linkage method and AUTOCLUST joined all three clusters together because of the short bridges. DBSCAN joined two of the three clusters together.

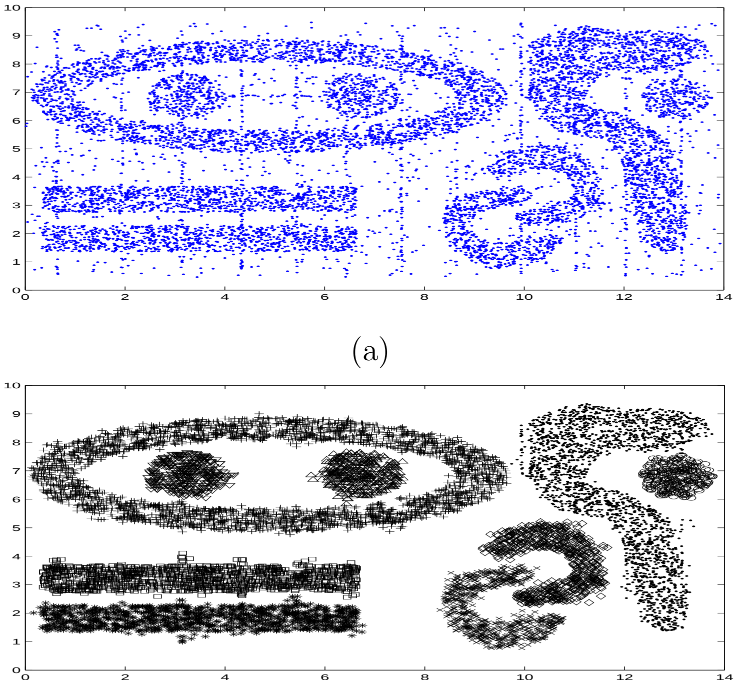
We also tested TRICLUST with the famous CHAMELEON data sets which have been regarded as benchmark data sets in this area. From Fig.10 to Fig.11, the pictures of three CHAMELEON data sets $C1$ to $C2$ and clustering results of them by TRICLUST are provided. In order to illustrate the results better, we drew clusters with different symbols without outliers in those figures of clustering results. All clusters in these CHAMELEON data sets have been correctly detected by TRICLUST, even there are short bridges between clusters. The ability of TRICLUST to deal with large complex data sets with masses of noises is clearly shown.



(a)

(b)

Fig. 10. (a)The picture of CHAMELEON data set $C1$; (b)The clustering result of CHAMELEON data set $C1$ by TRICLUST



(a)

(b)

Fig. 11. (a)The picture of CHAMELEON data set $C2$; (b)The clustering result of CHAMELEON data set $C2$ by TRICLUST

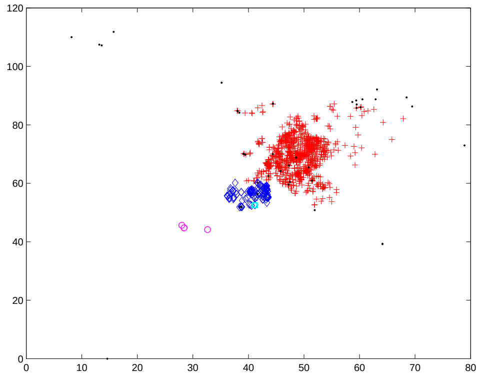


Fig. 12. An example of real world application of TRICLUST

In this section, all clustering results are generated according to the parameters setting method proposed in Section 4. Users can also set the values of parameters based on their knowledge or special requirements. This algorithm also can be applied on high dimensional data set due to Delaunay Triangulation can be performed in any dimensional space [33,34].

## 8 Stability of TRICLUST to deviation of intrinsic parameters

In this section, we will discuss the stability of TRICLUST to three intrinsic parameters: $Smin$, $Smax$ in 7 and 5000 in 8.

Parameters $Smin$ and $Smax$ are designed to simulate clustering feature of human vision: the more data points are inside the data set the less particular details in relatively small parts of data one can notice. In principle, these two parameters should be estimated by research of human vision. But TRICLUST algorithm appears to be robust to their deviation. The robustness of TRICLUST to variation of these parameters is illustrated below by tests on data set $D2$ and large data set $C1$, where $Smin$ varies from 100 to 400 and $Smax$ from 5000 to 20000. In Table.1, we provide corresponding values of $a$, $b$ and $c$. The clustering results are the same as the ones shown in previous section.

The number 5000 in 8 is large data set threshold. For large data set, we cut 3% to increase algorithm robustness. This is a standard statistical procedure. The choice of large data set threshold highly depends on application and, in general, the variation of it would not affect the performance of TRICLUST.

Table 1
Variation of values of intrinsic parameters

| | | $a$ | | $b$ | | $c$ | |
|---|---|---|---|---|---|---|---|
| $Smin$ | $Smax$ | $D2$ | $C1$ | $D2$ | $C1$ | $D2$ | $C1$ |
| 100 | 5000 | 0.1694 | 2 | 0.9817 | 0.5 | 0.5183 | 1 |
| | 10000 | 0.1344 | 1.6162 | 0.9910 | 0.6010 | 0.5090 | 0.8990 |
| | 20000 | 0.1171 | 0.8453 | 0.9955 | 0.8015 | 0.5045 | 0.6985 |
| 200 | 5000 | 0.1313 | 2 | 0.9918 | 0.5 | 0.5082 | 1 |
| | 10000 | 0.1153 | 1.6122 | 0.9960 | 0.6020 | 0.5040 | 0.8980 |
| | 20000 | 0.1076 | 0.8485 | 0.9980 | 0.8030 | 0.5020 | 0.6970 |
| 400 | 5000 | 0.1395 | 2 | 1 | 0.5 | 0.5 | 1 |
| | 10000 | 0.1395 | 1.6042 | 1 | 0.6042 | 0.5 | 0.8958 |
| | 20000 | 0.1395 | 0.8367 | 1 | 0.8061 | 0.5 | 0.6939 |

## 9 Real World Application of TRICLUST

We applied TRICLUST on the real world data set collected from European Topic Center on Air and Climate Change (ETC/ACC), which established the European air quality database system which contains next to multi-annual time series of measurement data and their statistics for a representative selection of stations throughout Europe. In this example, we used TRICLUST to study the distribution of the locations of stations for particular value of certain air quality feature. Here, the data set is about the locations of stations with more than 95% coverage for ozone value monitoring. The clustering result is shown in Fig.12. All 1181 stations have been divided into four clusters according their location distribution under both global and location view. The black dots represent stations that are far from others or with irregular locations. With TRICLUST, people who work in this area can conduct further investigations on the relationship between station positions and the coverage of stations. This result is generated automatically by TRICLUST. We can also set the parameters according to the knowledge on data set to fit the specific application requirements.

## 10 Conclusion and Future work

By the theoretical analysis and experimental tests shown above, the ability of TRICLUST to handle data sets with clusters of complicated shapes and non-uniform densities, and with large amount of noises is well demonstrated. The specifically built criteria function based on statistical feature values extracted from triangulation graph serves as a flexible discrimination according to both global and local information, so that our algorithm can work effectively. Moreover, the boundary detection function of TRICLUST has a good potential for real application. In the future, we will employ more complex statistical techniques, such as density estimation methods for our threshold determination to fit certain application requirements. And we will extend our algorithm to high dimensional applications where the efficiency of our method need to be improved. For future applications, we plan to apply our algorithm in environmental research for spatial analysis of large-scale data originating from satellite imagery and grounded-based sensors, and geo-spatial data processing.

## References

[1] Trevor Bailey and Tony Gatrell, *Interactive Spatial Data Analysis*, Prentice Hall, 1996.

[2] Usama.M. Fayad, *Advances in Knowledge Discovery in Databases*, MIT Press, 1996.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[4] Pavel Berkhin, “Survey Of Clustering Data Mining Techniques,” Tech. Rep., Accrue Software, San Jose, CA, 2002.

[5] Rui Xu and II. D.Wunsch, “Survey of Clustering Algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, pp. 645 – 678, 2005.

[6] J. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” in *Proc. 5th Berkeley Symp. Math. Statist. Prob.*, 1967, pp. 281–297.

[7] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 1st edition, 2000.

[8] F. Murtagh, *Multidimensional Clustering Algorithms*, Compstat Lectures, Vienna: Physika Verlag, 1985.

[9] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an Efficient Data clustering method for very large databases,” in *Proc. of the 1996 ACM SIGMOD international conference on Management of data*, 1996, pp. 103–114.

[10] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, “CURE: An Efficient Clustering Algorithm for Large Databases,” in *Proc. of the 1998 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1998, pp. 73–84, ACM Press.

[11] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proc. of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[12] A. Hinneburg and D.A. Keim, “An Efficient Approach to Clustering in Large Multimedia Databases with Noise,” in *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

[13] M. Ankerst, Markus M.Breunig, H.-P. Kriegel, and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure,” in *Proc. ACM SIGMOD Int. Conf. on Management of Data SIGMOD’99*, 1999, pp. 49–60.

[14] C. Eldershaw and M. Hegland, “Cluster Analysis using Triangulation,” in *Proc. Computational Techniques and Applications: CTAC97*, Singapore, 1997, pp. 201–208.

[15] V. Estivill-Castro and I. Lee, "AMOEBA: Hierarchical Clustering Based on Spatial Proximity Using Delaunay Diagram," in *Proc. of the 9th International Symposium on Spatial Data Handling*, 2000, pp. 7a.26–7a.41.

[16] V. Estivill-Castro and I. Lee, "AUTOCLUST: Automatic Clustering via Boundary Extraction for Mining Massive Point-Data Sets," in *Proc. of the 5th International Conference on Geocomputation*, 2000.

[17] T. Kohonen, "The Self-organizing Map," *Proceedings of the IEEE*, vol. 9, pp. 1464–1479, 1990.

[18] Y.J. Zhang and Z.Q. Liu, "Self-Splitting Competitive Learning: A New Online Clustering Paradigm," *IEEE Trans. Neural Networks*, vol. 13, pp. 369–380, 2002.

[19] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," in *Proc. of the 20th International Conference on Very Large Data Bases VLDB 94*, Santiago, Chile, Sept 1994, pp. 144–155.

[20] C. Fraley and A.E. Raftery, "MCLUST: Software for Model-Based Clustering, Density Estimation and Discriminant Analysis," Tech. Rep. 415, Department of Statistics, University of Washington, 2002.

[21] C. Fraley and A.E. Raftery, "Model-Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, vol. 97, pp. 611–631, Jun 2002.

[22] A. Adamson and M. Alexa, "Approximating Bounded, Non-orientable Surfaces from Points," in *Proc. of Shape Modeling International 2004*, 2004, pp. 243–252.

[23] H. Pfister and M. Gross, "Point-Based Computer Graphics," *IEEE Computer Graphics and Applications*, vol. 4, pp. 22–23, 2004.

[24] M. Andersson, J. Giesen, M. Pauly, and B. Speckmann, "Bounds on the k-Neighborhood for Locally Uniformly Sampled Surfaces," in *Proc. of the 1st Symposium on Point-Based Graphics*, 2004, pp. 167–171.

[25] M. Pauly, M. Gross, and L. Kobbelt, "Efficient Simplification of Point-Sampled Surfaces," in *Proc. of the conference on Visualization '02*, 2002, pp. 163–170.

[26] C. T. Zahn, "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. on Computers*, vol. 20, no. 1, pp. 68–86, 1971.

[27] G. Nagy, "State of the Art in Pattern Recognition," *Proceedings of the IEEE*, vol. 56, pp. 836– 863, 1968.

[28] C. M. Gold, "Problems with Handling Spatial Data-The Voronoi Approach," *CISM Journal ACSGC*, vol. 45, pp. 65–80, 1991.

[29] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, 2nd edition, 2000.

[30] Lee D.T. and B.J. Schacter, "Two Algorithms for Constructing a Delaunay Triangulation," *International Journal of Computer and Information Sciences*, vol. 3, pp. 219–241, 1980.

[31] In-Soo Kang, Tae wan Kim, and Ki-Joune Li, "A Spatial Data Mining Method by Delaunay Triangulation," in *Proc. of the 5th ACM international workshop on Advances in geographic information systems*, New York, NY, USA, 1997, pp. 35–39, ACM Press.

[32] S. Hader and F.A. Hamprecht, "Efficient density clustering using basin spanning trees," in *Proc. of the 26th Annual Conference of the Gesellschaft für Klassifikation (GfK1)*, 2003, pp. 39–48.

[33] D. F. Watson, "Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes," *The Computer Journal*, vol. 24, pp. 167–172, 1981.

[34] A. Bowyer, "Computing Dirichlet Tessellations," *The Computer Journal*, vol. 24, pp. 162–166, 1981.