Computers, Environment and Urban Systems 35 (2011) 320–332

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/compenvurbsys

An adaptive spatial clustering algorithm based on delaunay triangulation

Min Deng [a,*], Qiliang Liu [a], Tao Cheng [b], Yan Shi [a]

[a] Department of Surveying and Geo-informatics, Central South University, Changsha, China
[b] Department of Civil, Environmental and Geomatic Engineering, University College London, Gower St., WC1E 6BT, London, United Kingdom

ARTICLE INFO

Article history:
Received 26 August 2010
Received in revised form 11 February 2011
Accepted 14 February 2011
Available online 12 March 2011

Keywords:
Spatial clustering
Adaptive
Delaunay triangulation
Spatial data mining

ABSTRACT

In this paper, an adaptive spatial clustering algorithm based on Delaunay triangulation (ASCDT for short) is proposed. The ASCDT algorithm employs both statistical features of the edges of Delaunay triangulation and a novel spatial proximity definition based upon Delaunay triangulation to detect spatial clusters. Normally, this algorithm can automatically discover clusters of complicated shapes, and non-homogeneous densities in a spatial database, without the need to set parameters or prior knowledge. The user can also modify the parameter to fit with special applications. In addition, the algorithm is robust to noise. Experiments on both simulated and real-world spatial databases (i.e. an earthquake dataset in China) are utilized to demonstrate the effectiveness and advantages of the ASCDT algorithm.

1. Introduction

More attention has been paid to spatial data mining in the last decade (Miller & Han, 2009), in order to discover valuable information or knowledge from spatial databases. Spatial clustering has played an important role in the process of spatial data mining. It aims to classify a spatial database into several clusters without any prior knowledge (e.g., probability distribution and the number of clusters). In general, spatial points in the same cluster are similar to one another and dissimilar to the points in different clusters. Spatial clustering can be generally employed to segment geographic regions with different characteristics and extract meaningful spatial patterns. Spatial clustering also can help to generalize the aggregate point features and find the optimal positions of the public facilities.

There are mainly two types of spatial clustering methods. One considers only the spatial attributes of spatial points (i.e. geometric coordinates), and the other considers both spatial and non-spatial attributes of spatial points. Currently, spatial clustering has a wide range of applications, such as crime hot-spot analysis (Estivill-Castro & Lee, 2002a), land use detection (Ester, Kriegel, Sander, & Xu, 1996; Sander, Ester, Kriegel, & Xu, 1998), earthquake analysis (Pei, Zhu, Zhou, Li, & Qin, 2009; Xu, Ester, Kriegel, & Sander, 1998), vehicle crashes (Yamada & Thill, 2007) and agricultural environments (Schoier & Borruso, 2004, 2007). In this paper, the spatial clustering of geo-referenced 2-D point data is emphasized, where only the spatial attribute of point data is considered. As a matter of fact, this is the most fundamental clustering task for exploratory spatial analysis, which plays a critical role in further investigations (Estivill-Castro & Lee, 2002a; Ng & Han, 1994).

A number of clustering algorithms have been developed for spatial databases. They can be classified roughly into partitioning algorithms (MacQueen, 1967; Kaufman & Rousseeuw, 2005; Ng & Han, 1994), hierarchical algorithms (Estivill-Castro & Lee, 2002a; Guha, Rastogi, & Shim, 1998; Karypis, Han, & Kumar, 1999; Xu & Wunsh, 2009; Zhang, Ramakrishnan, & Livny, 1996), density-based algorithms (Ankerst, Breunig, Kriegel, & Sander, 1999; Ester et al., 1996; Hinneburg & Keim, 1998; Nosovskiy, Liu, & Sourina, 2008), graph-based algorithms (Estivill-Castro & Lee, 2002b; Liu, Nosovskiy, & Sourina, 2008; Zahn, 1971; Zhong, Miao, & Wang, 2010), grid-based algorithms (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998; Sheikholeslami, Chatterjee, & Zhang, 1998; Wang, Yang, & Muntz, 1997), model-based algorithms (McLachlan & Krishnan, 1996; Xu et al., 1998) and various combinational algorithms (Lin & Chen, 2005; Tsai & Yen, 2007). All the above-mentioned algorithms have proved able to handle some specific applications.

Currently, the applications on complicated spatial databases bring new demands for clustering algorithms. A complicated spatial database may contain clusters adjacent to each other, with arbitrary geometrical shapes and/or different densities. In addition, a large amount of noise possibly exists. Here noise refers to spatial points which do not belong to any clusters, and can be classified into two kinds, i.e., background noise and chains. The background noise is some isolated points which often distribute randomly in the database; the chains are formed of noises which connect two separated clusters. New clustering algorithms should be developed in order to

* Corresponding author. Address: Dept. of Surveying and Geo-informatics, Central South University, Changsha, Hunan, China.
E-mail address: dengmin028@yahoo.com (M. Deng).

0198-9715/$ - see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.compenvurbsys.2011.02.003

be adaptive to different cases. In the light of this, an adaptive spatial clustering algorithm based on Delaunay triangulation (ASCDT for short) is developed in this paper. The development of this algorithm will be expected to enhance the applications in the fields as earthquake analysis (Pei et al., 2009; Xu et al., 1998), cartographic generalization (Li, 2007), geographic customer segmentation (Miller & Han, 2009) and other exploratory data analysis in geographic information systems (Estivill-Castro & Lee, 2002a).

The rest of the paper is organized as follows. In Section 2, related work and the strategy on adaptive spatial clustering are described in detail. The ASCDT algorithm is performed fully in Section 3. Experiments on both simulated and real databases are shown in Section 4. Section 5 summarizes the main findings and highlights the directions for future research.

2. Related work and our strategy on discovering spatial clusters

2.1. Related work

As mentioned above, spatial clustering algorithms are designed to discover the clusters of certain structures. In this section, the performance of some classical clustering algorithms will be examined in detail, according to the requirements of the spatial clustering algorithms. This examination will be made based upon several comparative experiments and further used to substantiate the analysis of the ASCDT algorithm proposed in this paper (to be discussed in Section 4).

There are several challenges in spatial clustering. The first is to discover clusters of arbitrary shapes. Partitioning methods, such as K-means (MacQueen, 1967), PAM (Kaufman & Rousseeuw, 2005), and CLARANS (Ng & Han, 1994), are suitable only for detecting clusters of spherical shape and similar sizes. Traditional hierarchical algorithms, such as single-link and complete-link, have difficulty in discovering clusters with different shapes. Improved hierarchical algorithms, such as BIRCH (Zhang et al., 1996) and CURE (Guha et al., 1998), can deal with clusters of more complex shapes, but they are unable to discover very complicated clusters. Density-based methods, such as DBSCAN (Ester et al., 1996), DENCLUE (Hinneburg & Keim, 1998), and OPTICS (Ankerst et al., 1999) can discover arbitrary-shaped clusters, but they perform well only in cases in which the clusters are well separated and have similar densities.

The second challenge is to handle spatial data with mixed density. There are two cases. One case is that the density of one cluster is even but different from the other cluster, as shown in Fig. 1a; the other case is that the density of the cluster is not even, as shown in Fig. 1b. Density-based algorithms, such as DBSCAN, DENCLUE, and OPTICS, are all influenced by the density problem. Several improved graph-based algorithms, such as AMOEBA (Estivill-Castro & Lee, 2002a), CHAMELEON (Karypis et al., 1999), and AUTOCLUST (Estivill-Castro & Lee, 2002b can be utilized to discover clusters with different densities in a spatial database (the case in Fig. 1a). However they cannot detect clusters with uneven internal densities (the case in Fig. 1b).

The third challenge is that spatial clustering algorithms should be robust to noise and the "touching problems" (Zhong et al., 2010). When the noise is isolated from the spatial points, the density-based and graph-based algorithms, such as DBSCAN and AUTOCLUST, can deal with the noise well. However if the noise from one or more chains connects two clusters, which is also called the "chaining problem" (Fig. 2a), this has been proven to be a very challenging problem. Touching problems can be classified into two types. One is the famous "neck problem" (Fig. 2b) (Zahn, 1971); the other is the "adjacent problem" (Fig. 2c) (Estivill-Castro & Lee, 2002b), in which spare clusters are adjacent to high-density clusters. The MST algorithm (Zahn, 1971) can resolve the single chain problem and the touching problem. However, this algorithm on the one hand relies heavily on the prior information in the structure of the database; and on the other hand it cannot deal with multi-chains. The AUTOCLUST algorithm can be used to remove single or multi-chains, but it does not always work well. The AUTOCLUST algorithm still suffers from the neck problem. The 2-MSTClus algorithm (Zhong et al., 2010) can handle the chaining effect and the touching problem, but it is seriously affected by noise.

Lastly, spatial clustering algorithms should not rely on too much prior knowledge to obtain satisfactory results, because prior knowledge of a spatial database is usually unavailable for many applications. Partitioning algorithms, such as K-means, PAM and CLARANS, require the users to set the number of clusters. Hierarchical algorithms are involved in the setting of the merge or split conditions. Among the density-based algorithms, which include DBSCAN, DENCLUE, and OPTICS, it is indeed hard to determine the threshold of the density. Model-based algorithms, such as EM (McLachlan & Krishnan, 1996) and DBCLASD (Xu et al., 1998) require users to make an assumption about the probability distribution of the data. Graph-based algorithms, such as TRICLUST (Liu et al., 2008) require domain knowledge to set the weights of the different statistical features.

The analysis of the classical spatial clustering algorithms mentioned above is summarized in Table 1. It shows that the existing clustering algorithms have difficulty in satisfying various challenges of spatial clustering, and thus may encounter difficulties when dealing with practical applications. As a result, a new strategy needs to be developed.

2.2. A new strategy on discovering spatial clusters in spatial database

It is well known that geographic phenomena are the combination of global effect (large scale effect) and local effect (small scale effect) (Haining, 2003). The global effect influences the geographic phenomena first, and later the local effect (Bailey & Gatrell, 1995). Spatial clusters can be viewed as certain geographic phenomena that are merged into a spatial database. To detect spatial clusters more accurately, it is reasonable to remove the global effect first, and then the local effect. Therefore, a two-level strategy is employed to detect the clusters in a spatial database.

In the process of spatial clustering, the construction of the spatial proximity relationship is a critical issue. Delaunay triangulation has been proven to be a powerful tool for capturing spatial proximity in spatial analysis and spatial clustering (Estivill-Castro

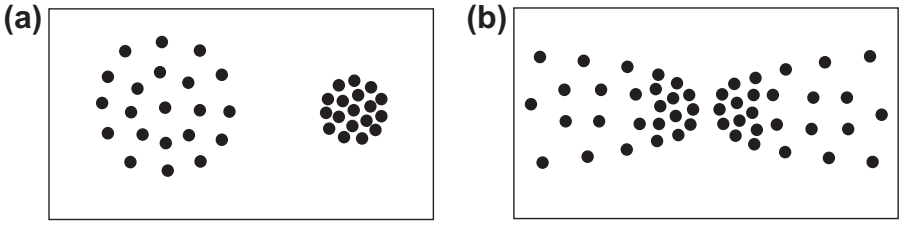

Fig. 1. Examples of clusters with uneven densities. (a) represent a database with clusters of various densities; (b) shows clusters with uneven densities.

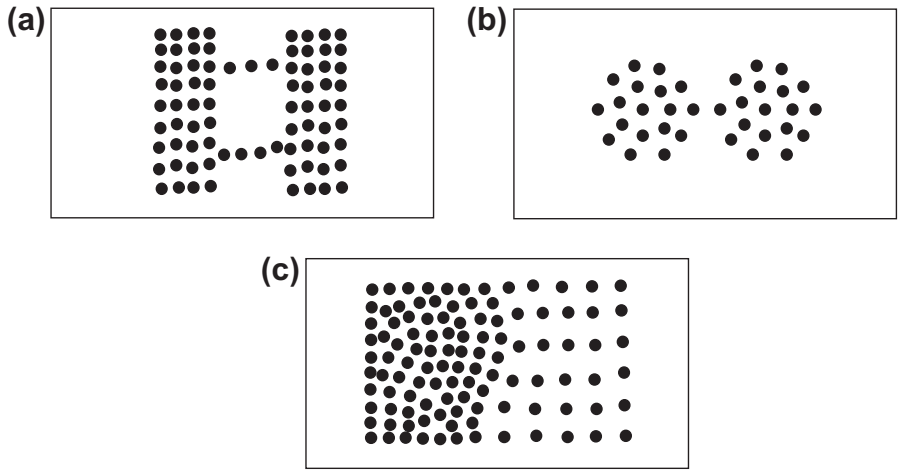

Fig. 2. Examples of the touching problem. (a) is the chaining problem; (b) is the neck problem; (c) is the adjacent problem.

Table 1
Comparisons of some classical spatial clustering algorithms.

Algorithm	Problem				
	Discovery of arbitrary shaped cluster	Discovery of clusters with uneven density	Handle touching Problems	Requirement of prior knowledge	Handle chaining problem
K-means	×	×	√ (for spherical clusters)	√	√ (for spherical clusters)
CLARANS	×	×	√ (for spherical clusters)	√	√ (for spherical clusters)
CURE	×	×	×	√	√ (for spherical clusters)
BIRCH	×	×	×	√	×
AMEOBA	√	√ (partial solution)	×	×	×
CHAMELEON	√	√ (partial solution)	×	√	×
DBSCAN	√	×	×	√	×
OPTICS	√	√ (partial solution)	×	√	×
DENCLUE	√	×	×	√	×
MST	√	√	√ (partial solution)	√	√ (partial solution)
AUTOCLUST	√	√ (partial solution)	√ (partial solution)	×	√ (partial solution)
EM	×	×	√ (for spherical clusters)	√	√ (for spherical clusters)
STING	√	×	×	√	×
WaveCluster	√	×	×	√	×
CLIQUE	√	×	×	√	×

& Lee, 2002a, 2002b; Liu et al., 2008; Lu & Thill, 2008; Tsai, 1993). In this paper, Delaunay triangulation is employed to model the spatial proximity among the spatial points. Further, the removal of inconsistent edges (such as edges which are too long, on the chain or on the neck) from the Delaunay triangulation is implemented to detect the clusters in a spatial database. As a result, in the two-level strategy, a global edge cut-off statistical criterion is proposed to remove those edges that are too long at the global level. Two local criteria are developed to remove those inconsistent edges at the local level. One criterion is a local edge cut-off statistical feature, which is used to remove locally long edges. The other is based on a novel spatial proximity definition, which aims to deal with chains and necks. In the following, some basic definitions are introduced, and then the ASCDT algorithm will be fully performed.

3. ASCDT algorithm

3.1. Basic definitions

Let $\boldsymbol{S} = \{P_1, \ldots, P_N\}$ denote a spatial database of N spatial points. $\boldsymbol{DT}(\boldsymbol{S})$ is the corresponding Delaunay triangulation of $\boldsymbol{S}$. Each spatial point P_i represents a vertex in $\boldsymbol{DT}(\boldsymbol{S})$. Here, a particular case is that some points may have same locations. In this case, the Delaunay triangulation cannot be constructed. Therefore, the strategy used by ADACLUS (Nosovskiy et al., 2008) is employed to solve this problem. More specifically, if two points share same location, such points are glued together as a point and labeled by the number of points glued. This operation is only made in the process of clustering, and the original data is not changed.

Definition 1. Given a graph $\boldsymbol{G}$, for each vertex P_i in $\boldsymbol{G}$, the K-order neighbors of P_i is the set of points that belongs to a path of K or less edges starting at P_i, and is denoted by $N_G^K(P_i)$.

Definition 2. Given a graph $\boldsymbol{G}$, the mean length of edges in $\boldsymbol{G}$, denoted by $Global\_Mean(G)$, is represented as

$$Global\_Mean(G) = \frac{\sum_{i=1}^{N} |e_i|}{N}, \tag{1}$$

where N denotes the number of edges in $\boldsymbol{G}$, and $|e_i|$ denotes the Euclidean length of edge e_i in $\boldsymbol{G}$.

Definition 3. Given a vertex P_i in $\boldsymbol{G}$, $Mean_G^K(P_i)$ is the mean length of the edges that belong to a path of K or less edges beginning at P_i, and is expressed as

$$Mean_G^K(P_i) = \frac{\sum_{j=1}^{n}|e_j|}{n} \quad (2)$$

where n denotes the number of edges formed by the points in $N_G^K(P_i)$, and $|e_j|$ denotes the Euclidean length of edge e_j, which belongs to a path of K or less edges starting at P_i.

Definition 4. Given a graph ***G***, the global variation of ***G***, denoted by *Global_Variation*(G), is defined as the standard deviation of the length of all edges in ***G***, represented as

$$Global\_Variation(G) = \sqrt{\frac{\sum_{i=1}^{N}(|e_i| - Global\_Mean(G))^2}{N-1}}. \quad (3)$$

Definition 5. Given a graph ***G***, the local variation of a vertex P_i, denoted by *Local_Variation*(P_i), is defined as the standard deviation of the length of all edges that are directly connected to P_i in ***G***, thus having

$$Local\_Variation(P_i) = \sqrt{\frac{\sum_{i=1}^{n}(|e_i| - Mean_G^1(P_i))^2}{n-1}}, \quad (4)$$

where n is the number of edges linking to P_i directly; and $|e_i|$ denotes the Euclidean length of edge e_i linking to P_i.

Definition 6. Given a graph ***G***, the mean variation of ***G***, denoted by *Mean_Variation*(G), is defined as the mean of the local variations of all the points in ***G***, which can be expressed as

$$Mean\_Variaton(G) = \frac{\sum_{i=1}^{N} Local\_Variation(P_i)}{N}, \quad P_i \in G \quad (5)$$

3.2. Removal of global effect

At a global level, only features that are well separated and have high densities are clustered, such as regions ***I*** and ***II*** in Fig. 3a. That is to say, it is necessary to delete the unduly long edges from the Delaunay triangulation in order to take the global effect into account. A threshold named global cut-off value is utilized to identify the unduly long edges at a global level. Thus, in a Delaunay triangulation ***DT***, all the edges that directly connect to a point P_i can first be classified into two types, according to the global cut-off value, namely *Global_Long_Edges*(P_i) and *Global_Other_Edges*(P_i). The former refers to the edges that are too long at the global level. The latter refers to the edges that are not too long at the global level.

Both the *Global_Mean* and the *Global_Variation* of the Delaunay triangulation will be two reasonable statistical measures to define the global cut-off value. As mentioned in Section 2.1, the densities of clusters in a spatial database may vary widely. The global cut-off value is able to not label those long edges in a relatively low density cluster, as suggested that the value should be relatively large for most edges in a cluster, and relatively low for the long edges. The ratio of $Mean_{DT}^1(P_i)$ to *Global_Mean*(DT) can be utilized to define adaptive factor. Thus, for each point P_i, global cut-off value, denoted by *Global_Cut_Value*(P_i), can be represented as

$$Global\_Cut\_Value(P_i) = Global\_Mean(DT) + \frac{Global\_Mean(DT)}{Mean_{DT}^1(P_i)} \cdot Global\_Variation(DT) \quad (6)$$

where $Mean_{DT}^1(P_i)$ is the mean length of the edges linking to P_i directly. $\frac{Global\_Mean(DT)}{Mean^1(P_i)}$ is the adaptive factor of *Global_Cut_Value*(P_i) .

Global_Long_Edges(P_i) and *Global_Other_Edges*(P_i) can then be defined as follows:

Definition 7. For each point P_i in Delaunay triangulation ***DT***, an edge e_j, which directly connects to P_i, belongs to *Global_Long_Edges*(P_i) if the length of e_j is longer than or equal to *Global_Cut_Value*(P_i), thus having there

$$Global\_Long\_Edges(P_i) = \{e_j | |e_j| \geqslant Global\_Cut\_Value(P_i)\} \quad (7)$$

where $|e_j|$ is the length of edge e_j.

Definition 8. For each point P_i in Delaunay triangulation ***DT***, an edge e_j, which directly connects to P_i, belongs to *Global_Other_Edges*(P_i) if the length of e_j is shorter than *Global_Cut_Value*(P_i), thus having there

$$Global\_Other\_Edges(P_i) = \{e_j | |e_j| < Global\_Cut\_Value(P_i)\} \quad (8)$$

where $|e_j|$ is the length of edge e_j.

Global long edges can be classified into four types: edges between well-separated clusters (e.g., P_1P_2 in Fig. 3b), edges between a cluster and global noise (e.g., P_3P_4 in Fig. 3b), edges between global noise (e.g., P_4P_5 in Fig. 3b), and edges between local noise and global noise (P_5P_6 in Fig. 3b). Global well-separated clusters and global background noise will be extracted from the data by removing the long edges belonging to *Global_Long_Edges*. In Fig. 4a, we see that the global clustering patterns have been discovered successfully. The *Global_Cut_Value* here is similar to the tolerance value used by AMOEBA (Estivill-Castro & Lee, 2002a), but it is not used to remove the inconsistent edges in sub-clusters. Though *Global_Cut_Value* is effective in identifying and removing *Global_Long_Edges*, it cannot handle local inconsistent edges effectively. To obtain spatial clusters more accurately, in the following *Global_Other_Edges* will be further refined, at a local level, in order to remove the local effect.

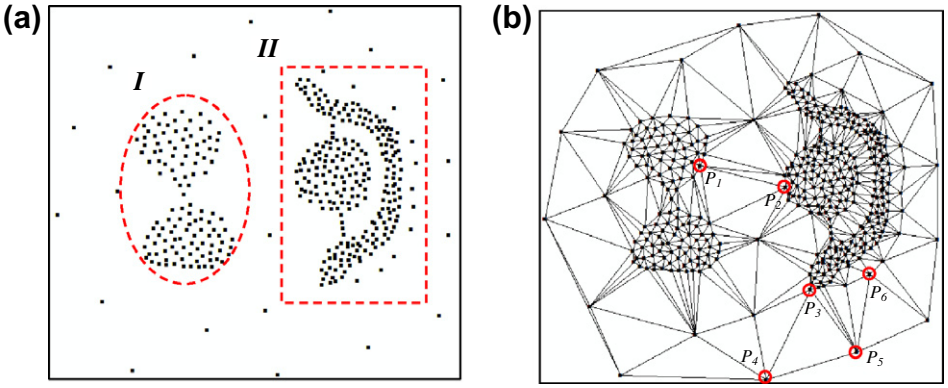

Fig. 3. A simulated spatial database and its Delaunay triangulation. (a) is the simulated database; (b) is the Delaunay triangulation of the simulated database.

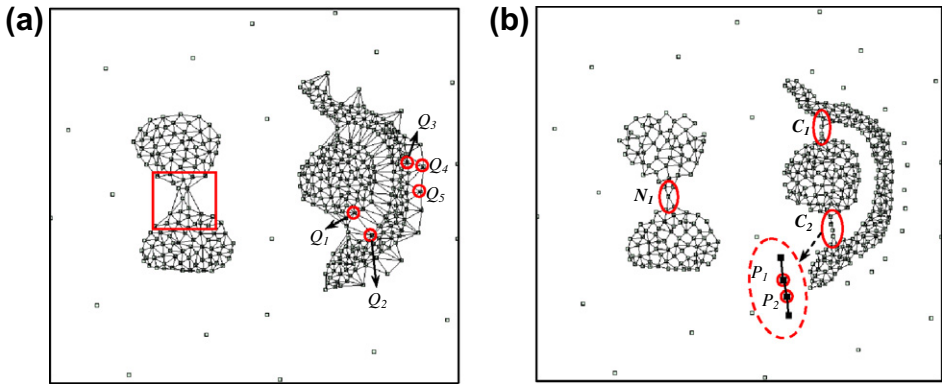

Fig. 4. Removal of *Global_Long_Edges* and *Local_Long_Edges*. (a) is the rough clustering patterns with a zoom window after removing *Global_Long_Edges*; (b) is the graph after removing *Local_Long_Edges*.

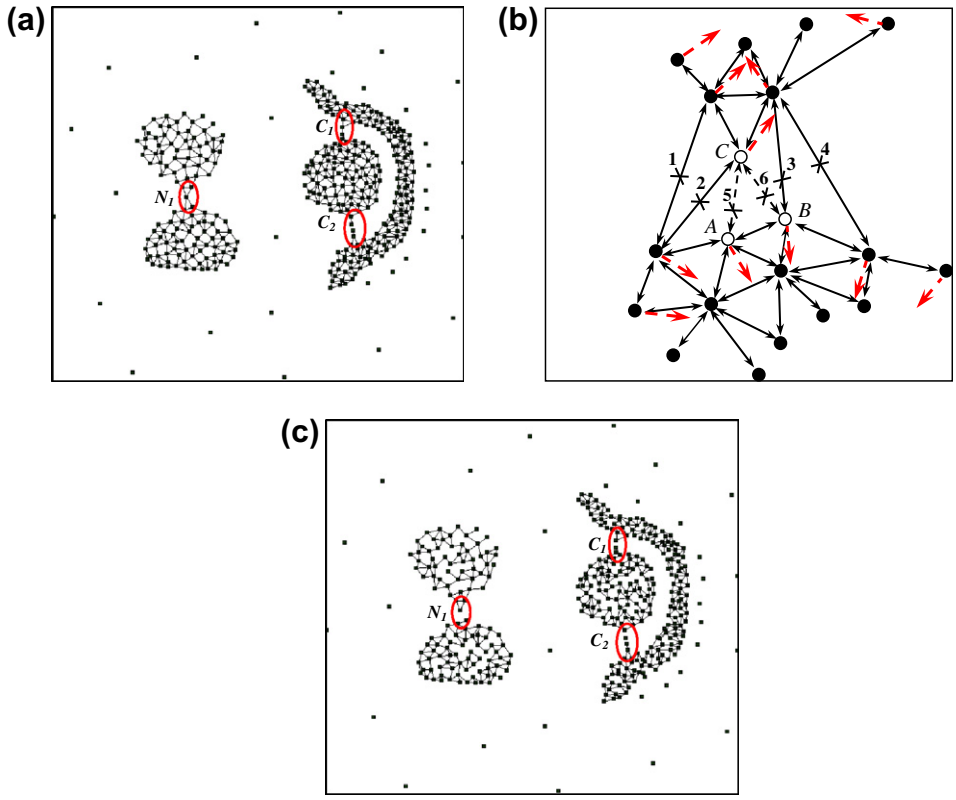

Fig. 5. Removal of chains and necks. (a) is the graph after removing chain $\boldsymbol{C}_2$ based on observation 1; (b) is the zoomed part of (a), which shows the procedure for removing the neck between two clusters; (c) shows the sub-graphs obtained by the ASCDT algorithm.

3.3. Removal of local effect

After removing the global effect, the local effects should again be considered. This means detecting and removing the local inconsistent edges from *Global_Other_Edges*.. Previous spatial clustering algorithms based on Delaunay triangulation, such as AUTOCLUST and AMOEBA, employ only edge length constraints to remove edges that are too long or too short in the Delaunay triangulation. In fact the local inconsistent edges include not only too long and too short ones, but also some normal edges, such as edges on the neck or on the short chains ($\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{N}_1$ in Fig. 4b). Thus the clustering results obtained by previous algorithms may not be robust. To handle this condition, local inconsistent edges are classified into two types: *Local_Long_Edges* and *Local_Link_Edges*.. The former refers to those edges that are substantially longer than the others. The latter refers to those edges that link clusters by necks or chains. Two criteria are developed to deal with these two types of inconsistent edges (i.e. *Local_Long_Edges* and *Local_Link_Edges*).

First, a local cut-off value is proposed, to identify those edges that are too long at a local level. After deleting *Global_Long_Edges* from the Delaunay triangulation ***DT***, there will remain a set of sub-graphs $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_n$. Then, for each point P_j in $\boldsymbol{G}_i$, $Mean^2_{G_i}(P_j)$ and $Mean\_Variation(G_i)$ are employed to define the local cut-off value, denoted by $Local\_Cut\_Value(P_j)$. This can be expressed as

$$Local\_Cut\_Value(P_j) = Mean^2_{G_i} + \beta \cdot Mean\_Variation(G_i) \tag{9}$$

where $Mean^2_{G_i}(P_j)$ is the mean length of the edges formed by the points in the second-order neighbors of P_j. β is a control factor that is used to control the sensitiveness of the $Local\_Cut\_Value(P_j)$. The

smaller the value of β, the more sensitive the $Local\_Cut\_Value(P_j)$ is. Thus, more edges may be recognized as long edges. In practice, β can be set from 1 to 1.5 and β is set to 1 by default in this paper. For some special applications, users need to have an option to set the parameter to obtain specific clustering results. The clusters will be a little rough when the parameter is set to a somewhat large value.

The points in second-order neighbors can be viewed as a scanning window that looks through a microscope, and so enables more details to be considered. $Local\_Cut\_Value(P_j)$ aims to identify the long edges at the local level only. These are distinctly longer than the average value of all the edges formed by the points in the second-order neighbors of P_j. In statistics, edges formed by the points in second-order neighbors in a sub-graph can be regarded as a small sample. In this case, there may be some error in the calculation of the variation. Thus, the mean variation of the sub-graph is employed as the variation indicator, which combines the local variation and the variation in the sub-graph. Next, *Local_Long_Edges* can be defined based on *Local_Cut_Value*

Definition 9. For each point P_j in a sub-graph $\boldsymbol{G}_i$, an edge e_j, formed by the vertexes in the second-order neighbors of P_j, belongs to $Global\_Other\_Edges(P_j)$. If the length of e_j is larger than or equal to $Local\_Cut\_Value(P_j)$, and so there we have

$$Local\_Long\_Edges(P_j) = \{e_k \,|\, |e_k| \geqslant Local\_Cut\_Value(P_j)\} \tag{10}$$

where $|e_k|$ is the length of edge e_k.

The graphs obtained after removing *Global_Long_Edges* are shown in Fig. 4a. *Local_Long_Edges* includes mainly the edges between local separated clusters (e.g., Q_1Q_2 in Fig. 4a), edges between clusters and local noise (e.g., Q_3Q_4 in Fig. 4a), and edges between local noise (e.g., Q_4Q_5 in Fig. 4a). The results after deleting *Local_Long_Edges* are shown in Fig. 4b, and the local background noise can be also identified.

Second, *Local_Link_Edges* should be further considered ($\boldsymbol{C}_1$, $\boldsymbol{C}_2$, and $\boldsymbol{N}_1$ in Fig. 4b). The edges linking to the points on the chains between clusters are sometimes *Global_Long_Edges* and sometimes *Local_Long_Edges*. After removing these two types of edge, some chains may be refined, e.g., $\boldsymbol{C}_2$ in Fig. 4b. To deal with this kind of chain, the following observation can be employed.

Observation 1. Given a sub-graph $\boldsymbol{G}_i$, two points P_j and P_k ($P_j \in \boldsymbol{G}_i$, $P_k \in \boldsymbol{G}_i$) on a chain are connected directly. If the number of points that directly link to P_j and P_k are both two, and there are no other edges linking P_j with P_k, then the edge between P_j and P_k should be deleted in order to cut the chain.

Taking $\boldsymbol{C}_2$ in Fig. 4b as an example, P_1, P_2 are the points at observation 1 described. Thus the edge between P_1 and P_2 should be cut, and the chain $\boldsymbol{C}_2$ is then broken, correspondingly. Although this strategy is useful for some long chains, it will not always work well for the condition that a chain is not trimmed as a line, such as the chain $\boldsymbol{C}_1$ in Fig. 4b. To handle the chaining problems that cannot be solved by observation 1 and the neck problem, a novel spatial proximity definition is developed. Here, the proximity between a spatial point and its spatial neighbors can be described as a local aggregation force and has a reverse relationship with distance. If the distance between two points is small, the local aggregation force between them is strong and it means the spatial proximity between them is small. If two points are far from each other, the local aggregation force between them will be very weak, and so can be neglected. One can find that the definition of local aggregation force has the similar function as traditional distance measures. However, it is a vector here. This characteristic can be further employed to deal with the chains. In the sub-graph $\boldsymbol{G}_i$, for each point P_i only its second-order neighbors are considered. Thus, the local aggregation force can be defined as follows.

Definition 10. Given a point P_j in a sub-graph $\boldsymbol{G}_i$, the local aggregation force between a vertex P_j and each vertex P_k in its second-order neighbors is denoted by $\vec{F}(P_j, P_k)$, and expressed as

$$\vec{F}(P_j, P_k) = k \cdot \frac{1}{d^2(P_j, P_k)} \vec{e}_{P_jP_k}, P_k \in N^2_{G_i}(P_j) \tag{11}$$

where k is the constant, which is here set to 1; $d(P_j, P_k)$ is the Euclidean distance between P_j and P_k; $\vec{e}_{P_jP_k}$ is the unit vector from P_j to P_k.

Definition 11. Given a point P_j in a sub-graph $\boldsymbol{G}_i$, the cohesive local aggregation force exerted on P_j is the sum of all the local aggregation forces exerted on P_j, denoted by $\vec{F}_C(P_j)$, and expressed as

$$\vec{F}_C(P_j) = \sum \vec{F}(P_j, P_k), P_k \in N^2_{G_i}(P_j). \tag{12}$$

Based on definitions 10 and 11, it can be reasonable that P_j is to be classified into a set of entities that the local aggregation forces between P_j and them are large and the cohesive local aggregation force on P_j will point to those entities. Thus, the cohesive local aggregation force of the points on the border of a cluster will point to the internal points in the cluster, and will have a tendency to move into the cluster. Such property is called the "border shrink phenomenon." If two clusters connected by neck or short chain can be distinguished, there will be large difference in the "shrink directions" (moving tendency of the points on the border) between them. Taking the neck problem shown in Fig. 5b as example, in which the dotted vectors represent the directions of the cohesive local aggregation force. One can see that the moving tendencies of the points on the border are nearly opposite, and that there is a natural boundary between the two clusters. Thus, a local aggregation criterion can be defined, and be used to handle the necks or short chains, based on the "border shrink phenomenon." This is defined as follows:

Definition 12. Given a point P_j in a sub-graph $\boldsymbol{G}_i$, the local aggregation set of P_j is formed by those points that seriously attract P_j, and directly connect to P_j, denoted by $\boldsymbol{Local\_Agg\_Set}(P_j)$, which is represented by

$$\boldsymbol{Local\_Agg\_Set}(P_j) = \{P_k \,|\, \theta(\vec{F}_T(P_j), \vec{F}(P_j, P_k)) < 90^\circ \wedge P_k \in N^1_{G_i}(P_j)\} \tag{13}$$

where $\theta(\vec{F}_T(P_j), \vec{F}(P_j, P_k))$ is the angle between $\vec{F}_T(P_j)$ and $\vec{F}(P_j, P_k)$..

Further, the local aggregation criterion can be defined as follows:

Definition 13. For each point P_j in the sub-graph $\boldsymbol{G}_i$, the edge that directly connects to P_j will be reserved if another vertex of that edge is in the local aggregation set, and others will be deleted.

From the view of the synthesis and decomposition of the force, the smaller the angle between a component force and the cohesive force, the more contribution the component force will make to the cohesive force. If the angle is smaller than 90°, this means that the component force makes a positive contribution to the cohesive force; if the angle is larger than 90°, the component force makes a negative contribution to the cohesive force; if the angle is equal to 90°, the component force makes no contribution to the cohesive force. For a point P_i, 90° can be a natural threshold from which to determine whether the neighbors seriously attract it or not.

Take the neck problem in Fig. 5c as example, in which the angle between $\vec{F}_T(C)$ and $\vec{F}(C, A)$ is larger than 90°. The angle between

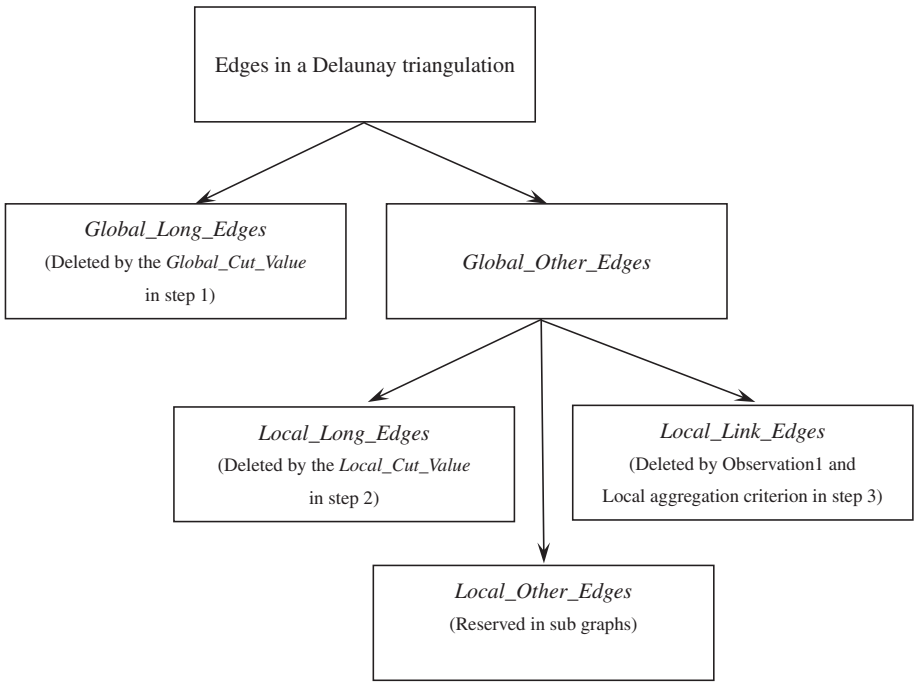

Fig. 6. Main steps of ASCDT algorithm.

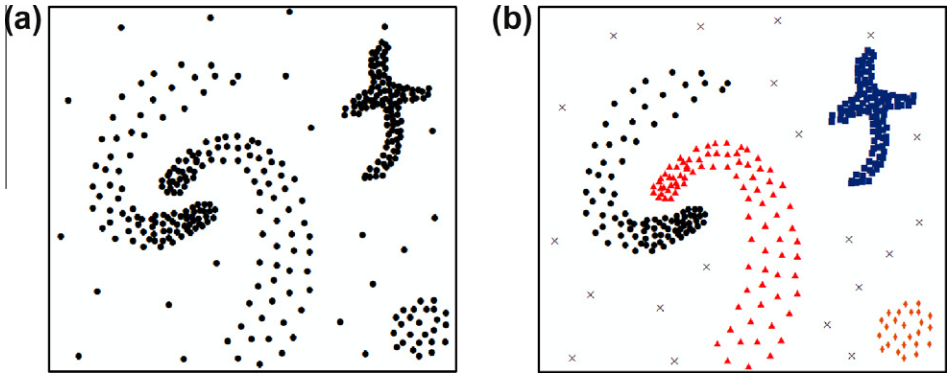

Fig. 7. The testing of ASCDT on ***S1***. (a) is the original database; (b) is the clustering resulting from ASCDT.

$\vec{F}_T(C)$ and $\vec{F}(C,B)$ is also larger than 90°. Thus A and B are not in ***Local_Agg_Set***(C). For points A and B, C is also not a member of ***Local_Agg_Set***(A) or ***Local_Agg_Set***(B). Thus edges 5 and 6 are deleted, according to the local aggregation criterion. Edges 1–4 are *Local_Long_Edges* and have been removed, so the neck between the two clusters is removed. The short chains can be handled in the same way. In Fig. 5c, it can be seen that the neck N_1 and the short chain C_1 are successfully removed.

After removing the local effect, each sub-graph of the Delaunay triangulation will be identified as a cluster, as shown in Fig. 5c. The edges reserved in the final sub-graphs are defined as *Local_Other_Edges*.

3.4. Algorithm description

Therefore, the ASCDT algorithm can be summarized as in the three steps in Fig. 6. The function of each step and its time complexity are elaborated as follows:

Step 1 aims to delete the edges that are too long at the global level. This process involves three operations, as follows:

(i) Construct Delaunay triangulation ***DT*** of the spatial database ***S*** with N points. This requires O (Nlog N) time.
(ii) Calculate the *Global_Mean* and *Global_Variation* of ***DT***, and $Mean^1_{DT}(P_i)$ for each point. The time complexity is linear to N.
(iii) Delete *Global_Long_Edges*. This requires about O (N) time.

Step 2 removes the edges that are too long at the local level. This process consists of the following two aspects:

(i) Calculate $Mean\_Variation(G_i)$ and $Mean^2_{G_i}(P_i)$ for each point in each sub-graph $\boldsymbol{G}_i$, which requires about O (N) time.
(ii) Remove *Local_Long_Edges*. The time complexity is also linear to N.

Step 3 deals with necks and chains. This process can be divided into the following two aspects:

(i) The implementation of Observation 1 requires about O (N) time.
(ii) Remove *Local_Link_Edges* and find final clusters. These two operations can be finished in O (N) time.

Thus the total complexity of the ASCDT algorithm is about O (Nlog (N)).

4. Experimental results and comparisons

The proposed spatial clustering algorithm ASCDT is tested on four 2-D simulated spatial databases, ***S***1–***S***4, and on a real-world spatial database. For comparison, five classical spatial clustering algorithms, i.e., K-means, CURE, AMOEBA, DBSCAN and AUTOCLUST, are also applied to these databases. For the CURE algorithm, the method of noise elimination is the same as in the original paper. Moreover, the shrink factor α is set to 0.5, and the number of representative points is set to 15, as the original paper suggested (Guha et al., 1998). For the DBSCAN algorithm, the parameters are set as $k = 4$ and MinPts = 4, which are based on the method given in the original paper (Ester et al., 1996).

4.1. Experiments on synthetic databases

S1 was shown in Fig. 7a. It is a complicated database with clusters of arbitrary shape, distinctly uneven internal density, clusters of different density and noise. As shown in Fig. 7b, the ASCDT algorithm is able to detect all four clusters, and the noise is also correctly identified. Five classical clustering algorithms are performed, and their results are shown in Fig. 8a–e. One can see from the results that none of them can separate all the clusters. More specifically, K-means and CURE algorithms cannot find clusters with a complex shape. When the densities of the clusters are different from each other, some points in the clusters with somewhat low density will be wrongly identified as noise by CURE algorithm. Both AMOEBA and AUTOCLUST algorithms cannot recognize clusters of uneven internal density. The DBSCAN algorithm is seriously influenced by the density difference among clusters.

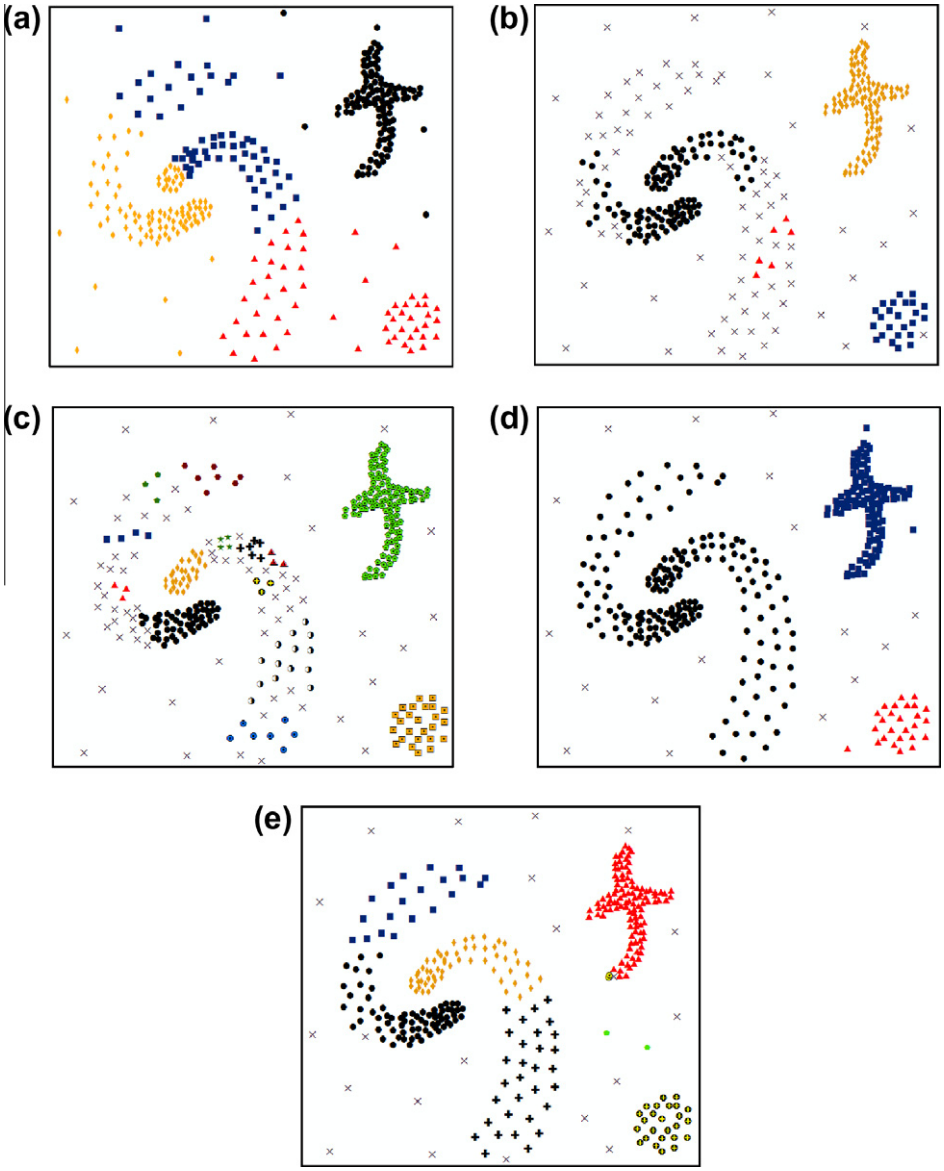

Fig. 8. Clustering results of ***S1*** by classical algorithms. (a) is the clustering result by K-means; (b) is the clustering result by CURE; (c) is the clustering result by AMOEBA; (d) is the clustering result by DBSCAN(Eps = 5.40); (e) is the clustering result by AUTOCLUST.

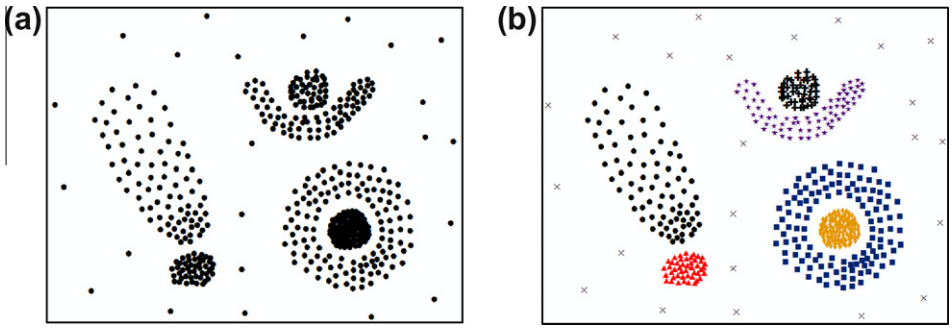

Fig. 9. The testing of ASCDT on ***S2***. (a) is the original database; (b) is the clustering result obtained by ASCDT.

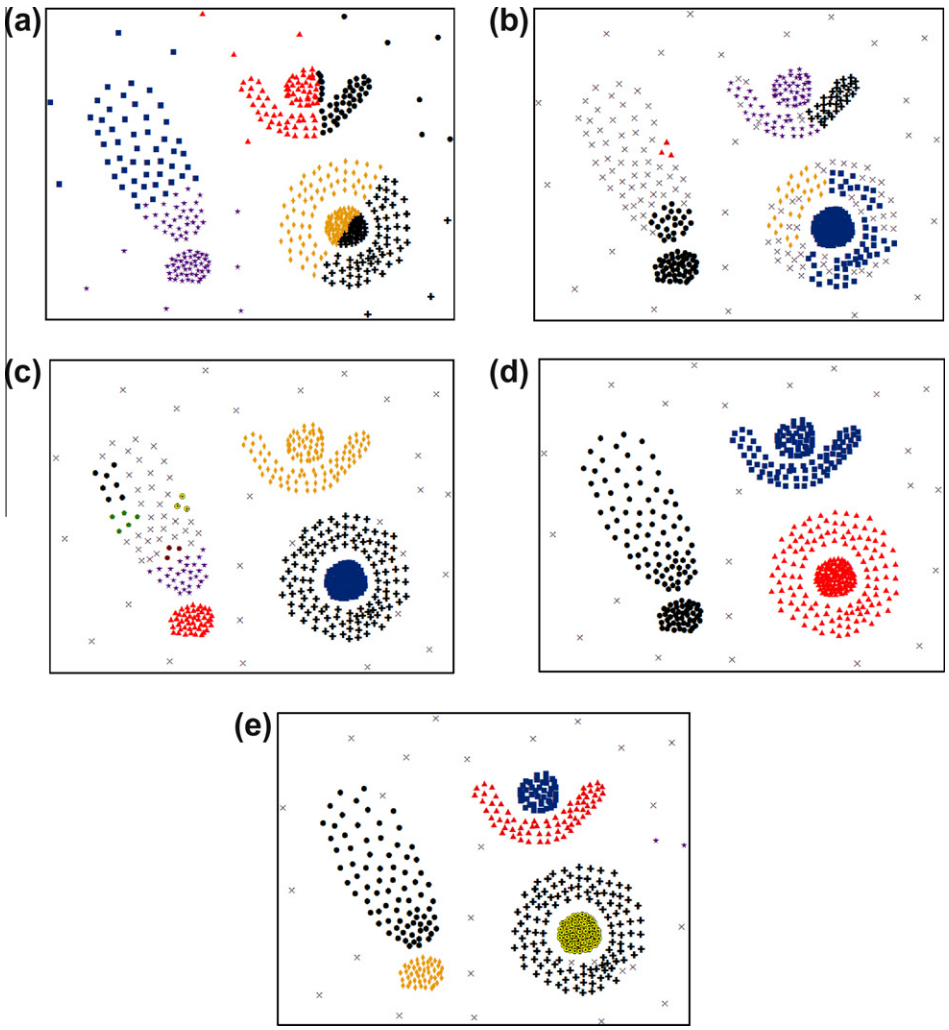

Fig. 10. Clustering results of ***S2*** by classical algorithms. (a) is the clustering result by K-means; (b) is the clustering result by CURE; (c) is the clustering result by AMOEBA; (d) is the clustering result by DBSCAN(Eps = 4.17); (e) is the clustering result by AUTOCLUST.

S2 is another challenging database shown in Fig. 9a. There are clusters with different densities, which are adjacent to each other. The clustering result by the ASCDT algorithm is shown in Fig. 9b, and the good performance of the algorithm is clearly demonstrated. In Fig. 10a–e, the clustering results of five other algorithms are shown. Except for the AUTOCLUST algorithm, none of them is able to identify the six clusters in *SDB2*.

S3 shown in Fig. 11a is utilized to test the performances of algorithms in the case in which the neck and chaining problems exist. In Fig. 11b, one can see that the ASCDT algorithm separates the six clusters well. In Fig. 12a–f, all the five classical clustering methods fail on this database. As shown in Fig. 12e, the AUTOCLUST algorithm can remove the long chains effectively, but it still suffers from the short chains and necks.

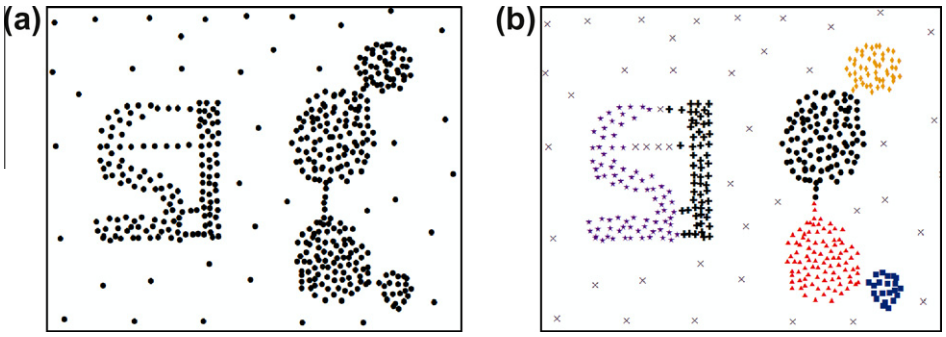

Fig. 11. The testing of ASCDT on ***S3***. (a) is the original database; (b) is the clustering result by ASCDT.

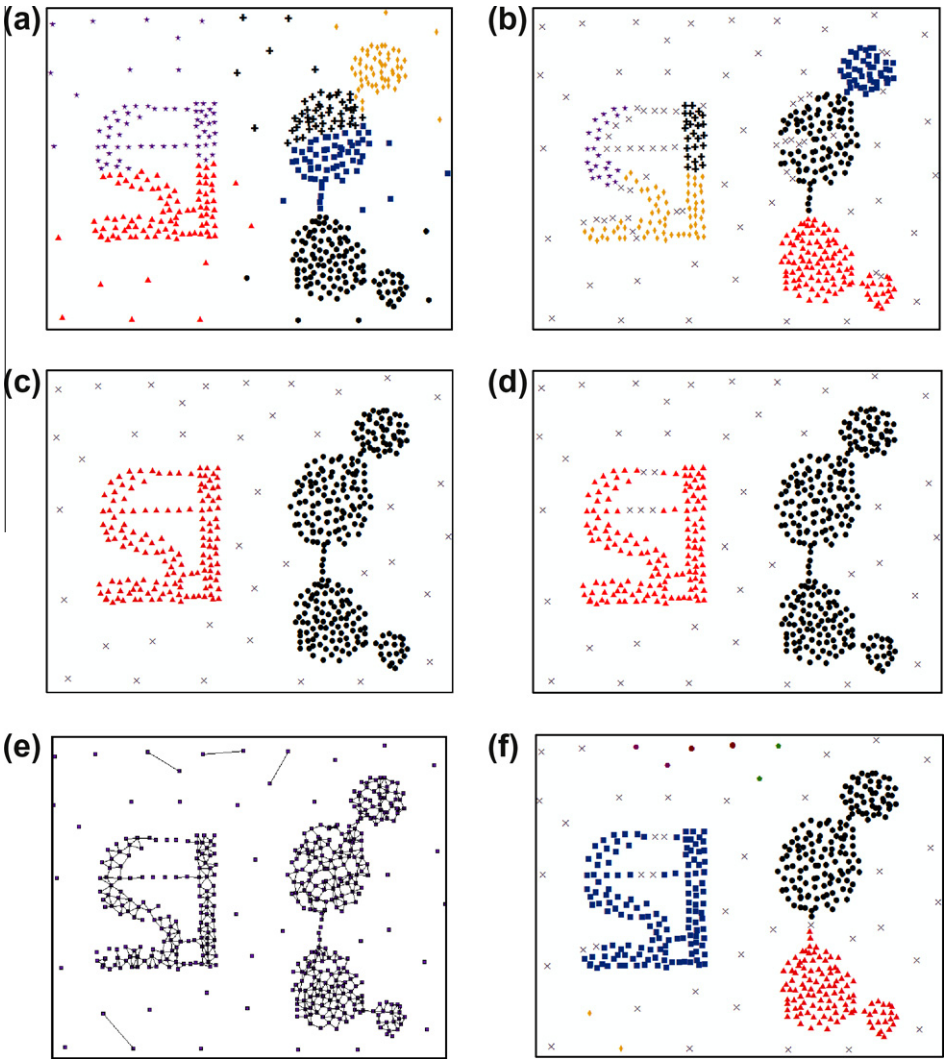

Fig. 12. Clustering results of ***S3*** by classical algorithms. (a) is the clustering result by K-means; (b) is the clustering result by CURE; (c) is the clustering result by AMOEBA; (d) is the clustering result by DBSCAN(Eps = 6.77); (e) is the sub-graphs obtained by AUTOCLUST; (f) is the clustering result by AUTOCLUST.

S4 is an extremely challenging database. As shown in Fig. 13a, there are eight clusters. These involve several challenging clustering problems mentioned in ***S1***, ***S2***, and ***S3***, such as clusters with internal uneven density, clusters with different densities, sparse clusters adjacent to high-density clusters, the neck problem, the chaining problem, clusters with complicated shapes, and large amount of noise. ***S***4 aims to test the performance of a spatial clustering algorithm when all these problems appear simultaneously. Comparing the clustering result by ASCDT, shown in Fig. 13b, and the results by five classical algorithms, shown in Fig. 14a–e, it can be found that, apart from the ASCDT algorithm, no algorithm can deal with this spatial clustering task.

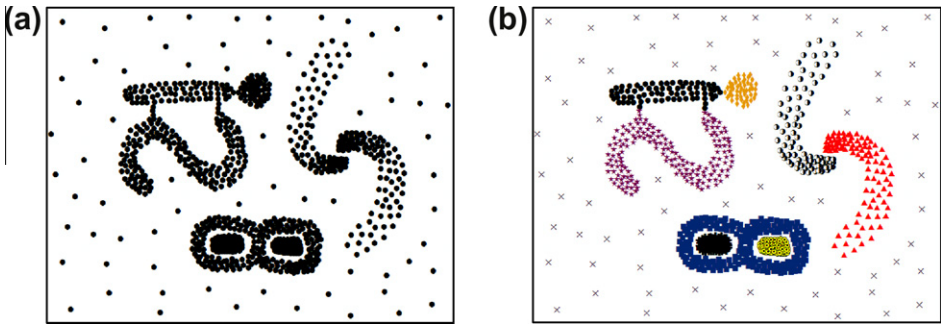

Fig. 13. The testing of ASCDT on ***S4***. (a) is the original database; (b) is the clustering result by ASCDT.

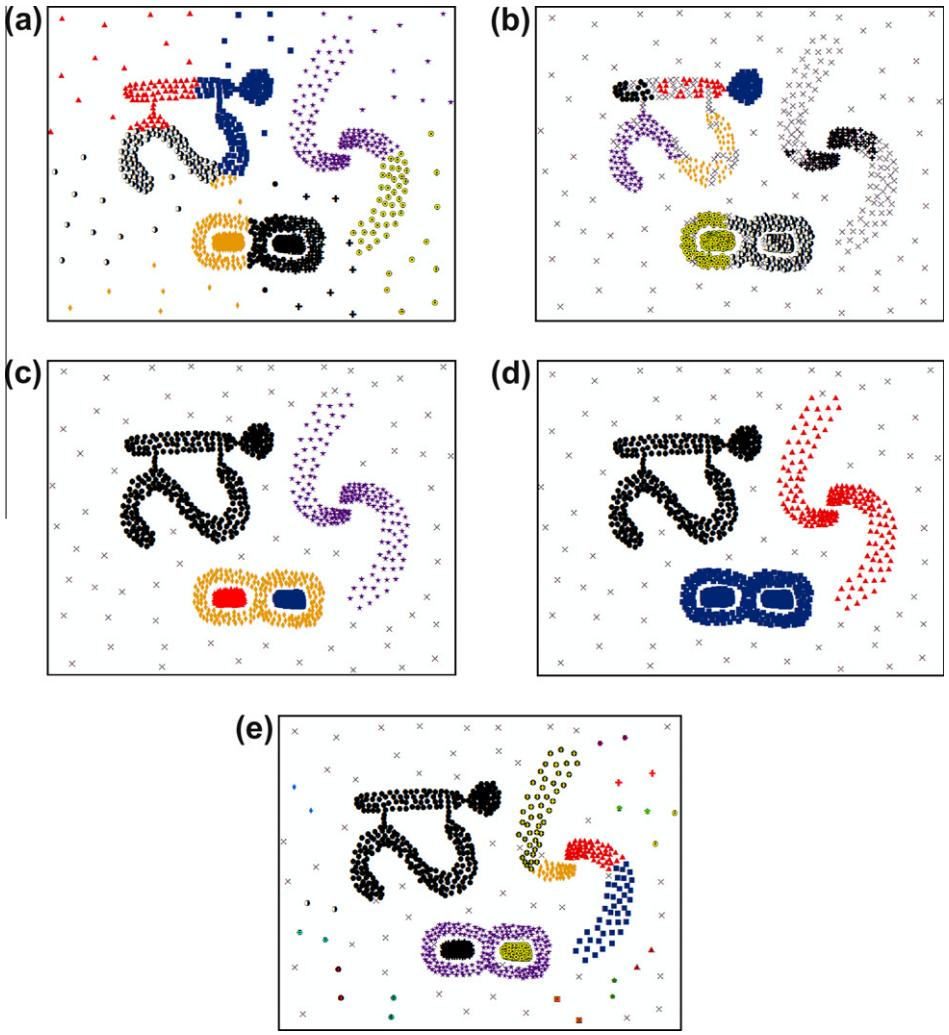

Fig. 14. Clustering results of ***S4*** by classical algorithms. (a) is the clustering result by K-means; (b) is the clustering result by CURE; (c) is the clustering result by AMOEBA; (d) is the clustering result by DBSCAN(Eps = 4.25); (e) is the clustering result by AUTOCLUST.

4.2. Practical applications of ASCDT

In order to illustrate the practicability of the ASCDT algorithm, it was applied to a spatial database of earthquakes. The determination of clustered earthquakes can be utilized to recognize active faults, or to understand the changing trend of earthquakes (Pei et al., 2009; Xu et al., 1998). The earthquake data used in this paper are from the Seismic Catalog of Mainland of China (1970–2006, M larger than or equal to 5) (Jiang, Fu, Liu, & Lv, 2007). The earthquake dataset is obtained from the project "the research of earthquake prediction in China during the 10th five-year plan". In China, the records of the strong earthquakes (i.e., M larger than or equal to 5) are more complete, so they are collected and used for the project. However, the earthquake data given by Jiang et al. (2007) is only from 1970 to 2004. The study area here is located between 90° and 104°E, and 21–38°N. Earthquakes in this area are the most

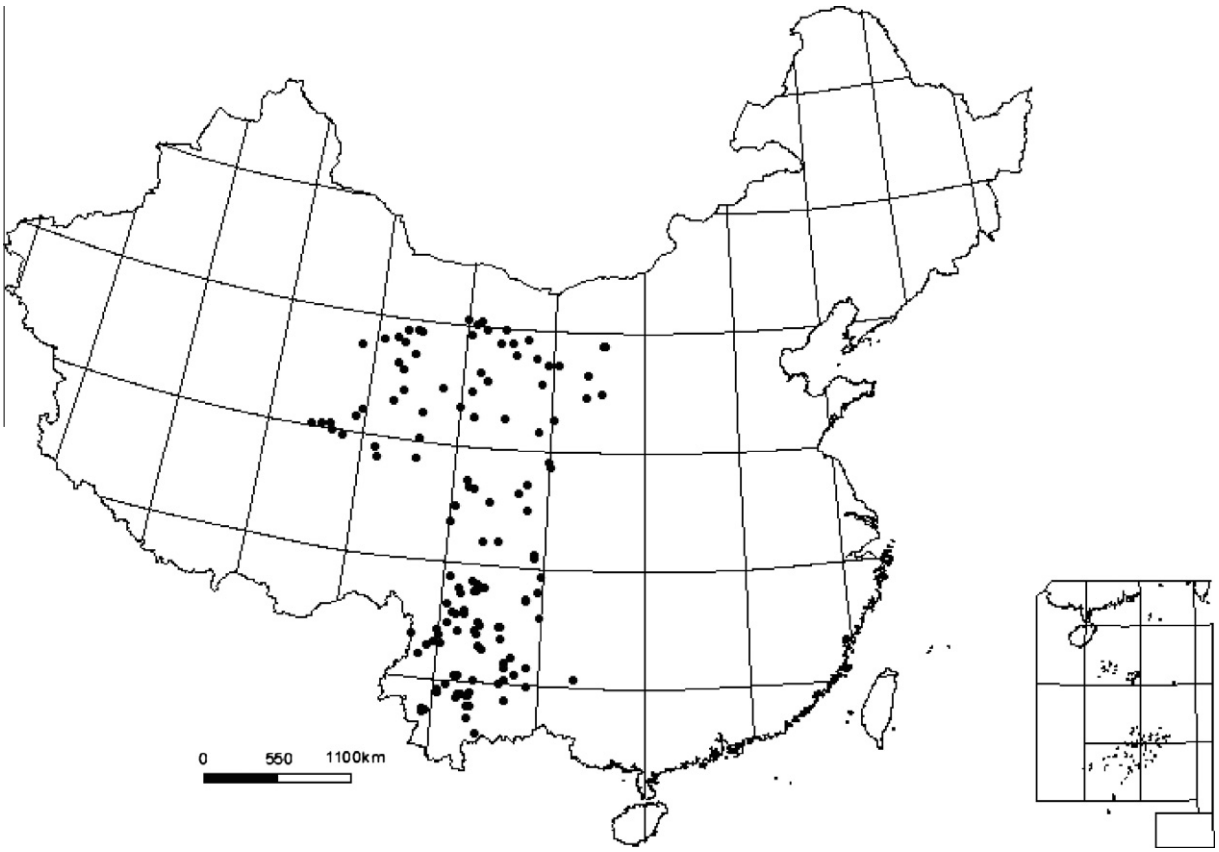

Fig. 15. Locations of the earthquakes (M ⩾ 5.0) in West China (1970–2004).

active in China, and have caused serious damage. There are 139 epicenters, and information on their locations is provided in the spatial database, as shown in Fig. 15.

The clustering result of ASCDT is shown in Fig. 16. There are five line-shaped clusters ($\boldsymbol{C}_1$, and $\boldsymbol{C}_3$ to $\boldsymbol{C}_6$), five spherical clusters ($\boldsymbol{C}_2$, $\boldsymbol{C}_7$–$\boldsymbol{C}_{10}$), four small clusters ($\boldsymbol{I}_1$ to $\boldsymbol{I}_4$) and two isolated earthquakes (labeled by symbol ×). Earthquake epicenters usually occur along seismically active faults, and an observed earthquake should be clustered along such faults over time (Xu et al., 1998). Taking the large faults in this area into account (Jiang et al., 2007), some interesting relationships between the clusters and faults can be found. The biggest line-shaped cluster $\boldsymbol{C}_1$ is located around the Qilian Mountain fault zone, and the directions of all the faults are from northwest to southeast. There are 17 large earthquakes in $\boldsymbol{C}_1$, which shows that the Qilian Mountain fault zone has been very active in the past 30 years. The line-shaped clusters $\boldsymbol{C}_5$ and $\boldsymbol{C}_6$ are located in various parts of the East Sichuan-East Yunnan fault zone; $\boldsymbol{C}_4$ is located in the Lijiang-Xiaojinhe fault zone, and the faults in these parts should be the active ones in the fault zone. The line-shaped cluster $\boldsymbol{C}_3$ shows interesting characteristic, that is, the direction of $\boldsymbol{C}_3$ almost crosses the main faults in that region. The cause of $\boldsymbol{C}_3$ maybe should be further investigated. The spherical clusters $\boldsymbol{C}_7$, $\boldsymbol{C}_8$, $\boldsymbol{C}_9$, and $\boldsymbol{C}_{10}$ are all located in a region that belongs to the Sichuan-Yunnan rhombic block. There are many faults in this region, and their directions differ from each other. Nearly half of the large earthquakes occurred in the Sichuan-Yunnan rhombic block in the past 30 years, and the clustered earthquakes may be helpful in identifying the rules of the crustal movement of this block. The loose cluster $\boldsymbol{C}_2$ is mainly located on the edge of Qinghai-Tibet plateau. It shows that there are no concentrated earthquakes in the past 30 years. The small clusters and isolated earthquakes usually represent sudden events, as are maybe useful to identify the movement tendency or rule of a certain fault.

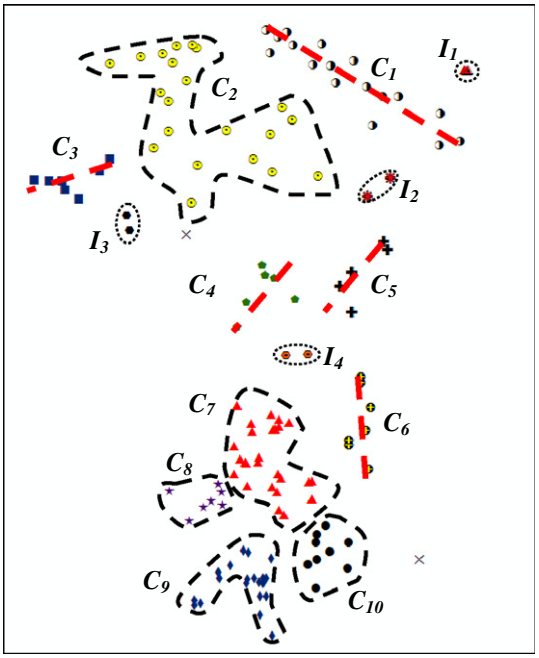

Fig. 16. Spatial clustering results on earthquake database by ASCDT.

5. Conclusions and future works

This paper proposes an adaptive spatial clustering algorithm based on the Delaunay triangulation (ASCDT). Through simulation

and practical experiments, and comparison with some classical algorithms, the good adaptiveness of the ASCDT algorithm is demonstrated in full. Based on global and local criteria, the ASCDT algorithm can automatically discover clusters of uneven density, clusters with arbitrary shapes, clusters of different densities, clusters connected by chains and/or necks, and in addition it is robust to noise. The time complexity of the ASCDT algorithm is about O (N (log$N$)), where N is the size of the spatial database.

Future works will focus on two directions. One is to apply the ASCDT algorithm to a large-scale spatial database, such as seismic data and satellite image data. The other is to consider those non-spatial attributes in the spatial clustering process that may lead to a deep analysis of the spatial database.

Acknowledgements

The work described was supported by the National High Technology Research and Development Program of China (863 Program), No. 2009AA12Z206 and National Science Foundation of China (NSFC), Nos. 40871180 and 40830530.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data* (pp. 94–105). ACM Press.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (pp. 49–60). Philadelphia, USA.

Bailey, T. C., & Gatrell, A. C. (1995). *Interactive Spatial Analysis*. New York: Wiley.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). Portland, OR.

Estivill-Castro, V., & Lee, I. (2002a). Multi-level clustering and its visualization for exploratory spatial analysis. *GeoInformatica, 6*, 123–152.

Estivill-Castro, V., & Lee, I. (2002b). Argument free clustering for large spatial point-data sets. *Computers, Environment and Urban Systems, 26*, 315–334.

Guha, S., Rastogi, R., Shim, K. (1998). CURE: An efficient clustering algorithm for large database. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (pp. 73–84). USA, New York.

Haining, R. (2003). *Spatial data analysis: Theory and practice*. United Kingdom: Cambridge University Press.

Hinneburg, A., Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* (pp. 58–65). USA, New York.

Jiang, H. K, Fu, Z. X., Liu, J., & Lv, P. L. (2007). *Research on earthquake sequences in Mainland of China*. Beijing: Seismological Press. in Chinese.

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer, 32*, 68–75.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding groups in data: An introduction to cluster analysis* (2nd ed.). New York: Wiley.

Li, Z. L. (2007). *Algorithm foundation of multi-scale spatial representation*. London: CRC Press.

Lin, C.-R., & Chen, M.-S. (2005). Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging. *IEEE Transaction on Knowledge and Data Engineering, 17*, 145–159.

Liu, D., Nosovskiy, G. V., & Sourina, O. (2008). Effective clustering and boundary detection algorithm based on Delaunay triangulation. *Pattern Recognition Letters, 29*, 1261–1273.

Lu, Y., & Thill, J.-C. (2008). Cross-scale analysis of cluster correspondence using different operational neighborhoods. *Journal of Geographical Systems, 10*, 241–261.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). California: University of California Press.

McLachlan, G. J., & Krishnan, T. (1996). *The EM algorithm and extensions* (1st ed.). New York: Wiley.

Miller, H., & Han, J. (2009). *Geographic data mining and knowledge discovery* (2nd ed.). New York: CRC Press.

Ng, R. T., Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases* (pp. 144–155). Chile, Santiago.

Nosovskiy, G. V., Liu, D., & Sourina, O. (2008). Automatic clustering and boundary detection algorithm based on adaptive influence function. *Pattern Recognition, 41*, 2757–2776.

Pei, T., Zhu, A. X., Zhou, C. H., Li, B. L., & Qin, C. Z. (2009). Detecting feature from spatial point processes using collective nearest neighbor. *Computers, Environment and Urban Systems, 33*, 435–447.

Sander, J., Ester, M., Kriegel, H.-P., & Xu, X. (1998). Density-based clustering in spatial database: The algorithm GDBSCAN and it applications. *Data Mining and Knowledge Discovery, 2*, 169–194.

Schoier, G., Borruso, G. (2004). A clustering method for spatial databases. In *Proceedings of the ICCSA 2004 International Conference* (pp. 1089–1095). Italia, Perugia.

Schoier, G., Borruso, G. (2007). A modification of the DBSCAN algorithm in a spatial data mining approach. In *Meeting of the Classification and Data Analysis Group of the SIS* (pp. 395–398). Italia, Macerata.

Sheikholeslami, G., Chatterjee, S., Zhang, A. (1998). Wave Cluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of 24th International Conference on Very Large Databases* (pp. 428–439). USA, New York.

Tsai, V. J. D. (1993). Delaunay Triangulations in TIN creation: An overview and a linear-time algorithm. *International Journal of Geographical Information Systems, 7*, 501–524.

Tsai, C.-F., & Yen, C.-F. (2007). ANGEL: A new effective and efficient hybrid clustering technique for large databases. *Lecture Notes in Computer Science, 4426*, 817–824.

Wang, W., Yang, J., Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In *Proceedings of the 1997 International Conference on Very Large Databases* (pp. 186–195). Greece, Athens.

Xu, X., Ester, M., Kriegel, H.-P., & Sander, J. (1998). A distribution-based clustering algorithm for mining in large spatial database. In *Proceedings of the 14th International Conference on Data Engineering* (pp. 324–331). USA, Washington, DC.

Xu, R., & Wunsh, D. II, (2009). *Clustering*. New Jersey: John Wiley & Sons.

Yamada, I., & Thill, J.-C. (2007). Local indicators of network-constrained clusters in spatial point patterns. *Geographical Analysis, 39*, 268–292.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transaction on Computer, 20*, 68–86.

Zhang, T., Ramakrishnan, R., Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 103–114), Canada, Montreal.

Zhong, C., Miao, D., & Wang, R. (2010). A graph-theoretical clustering method based on two rounds of minimum spanning trees. *Pattern Recognition, 43*, 752–766.