Gamma-partition: a clustering method for spatial point pattern analysis

Ping Yin

Department of Geography, University of Georgia, GG Building, 210 Field St., Room 204, Athens, GA 30602, U.S.A.

E-mail address: yinping9@uga.edu

Lan Mu

Department of Geography, University of Georgia, Athens, GA 30602, U.S.A.

E-mail address: mu@uga.edu

John Radke

Department of Landscape Architecture and Environmental Planning, University of California, Berkeley, CA 94720 U.S.A

E-mail address: ratt@gisc.berkeley.edu

Abstract

Spatial clustering methods are essential for point pattern analysis since they can discover strong relationships in point sets. Based on computational morphology, this paper proposes a theoretic framework, where spatial clustering methods are complementary to shape hulls and skeletons in describing geometric properties of point sets. Inspired by the well-accepted work of alpha-hull/shapes and beta-skeletons, a gamma-partition method is developed to divide a point set into clusters based on a Maximal Covering Criterion (MCC) with a fixed-radius disk, gamma. As a succession to concepts and methods of alpha-hull/shapes and beta-skeletons, gamma-partition links the three geometric descriptors for point sets, shape hull, skeleton, and clustering, at the method level. An algorithm of gamma-partition is elaborated, and some potential applications of gamma-partition are investigated in such fields as global spatial point pattern testing, location-based planning, and other clustering method construction, among others.

Keywords: spatial point patterns, spatial clustering, computational morphology, spatial analysis, GIScience

1. Introduction

A point cloud, a set of vertices or points in a dimensioned coordinate system, is a common way to map and represent the location of sampled phenomenon. With advances in remote sensing and GPS technology, point sets are also effective for monitoring, mapping and detecting patterns in migration, growth, and decline of populations under study. Identifying, recognizing, and characterizing sets, subsets, their shape, spread and density are all effective tools for characterizing morphology. With the digital revolution producing unprecedented volumes of data, we must develop metrics to synthesize and decompose points into defined clusters or subsets to help recognize patterns and better hypothesize underlying process.

Although a single point has no spatial dimension, the geography of a set of points embedded in some coordinate space can be described in terms of that space. It is common in spatial analysis to consider points in space as a set where extent, coverage and pattern can be detected, measured and reported in terms of the characteristics of the space itself.

It is often necessary for many methods of spatial point pattern analysis to test whether the observed pattern is clustered, random, or regular by statistically comparing some measures of the observed to those of a benchmark, for instance, a point pattern generated from a stochastic process in which points occur continuously and independently of one another creating a completely spatial random point set. These measures are mainly separated into two categories (Radke 1988), "those dealing with absolute measurements of the dispersion of points about the area occupied by a point set, and those examining the association of points relative to other points in the set". The former category includes measures of density, distance and direction that can be made operational as: density of points per quadrat, nearest neighbor distance, and standard deviational ellipse, for example. The latter category includes relative measures associated with point neighbors such as: counts of first and/or higher order neighbors, or the proximity of neighbors within a defined search geometry. All the measures listed above aim to provide a simple, summary, and quantitative descriptor for the spatial structures of point sets.

Computational morphology exploits ideas and techniques to characterize and describe spatial structure in point patterns that provide perceptually meaningful information of point sets. The decomposition of a point set can result in two types of shapes: external and internal shapes (Radke 1988). The external shape is defined as the shape hull, which is the external polygonal (area) characterization of shapes of point sets. Common shape hulls for point sets include the minimum bounding box or circle, and the convex hull. Internal shape is characterized as a skeleton that describes the internal link structure of a set of points based on point-to-point neighborhood properties. Common skeletons of point sets are the nearest neighbor graph, the minimum spanning tree, the Gabriel graph and the triangulated irregular network (TIN).

Both the shape hull and skeleton of point sets have been extensively applied in computer science, geography, ecology, urban planning, image processing and other GIScience fields. The minimum bounding box of geographic objects is the basis of R-trees (Guttman 1984) and R*-trees (Beckmann et al. 1990), which are popular methods of indexing geographic data in GIS because of their flexibility and excellent performance. The construction of the convex hull of point sets is often regarded as a first or bounding step in some algorithms that create other more complex graphs or data models, such as the Delaunay Triangulation or TIN (Longley et al. 2005). In addition, the convex hull is widely applied in the processing of LiDAR point clouds (Sampath and Shan 2007). The minimum spanning tree is directly applied to the design of computer, communication and transportation networks, and indirectly as a metric of surface homogeneity and automatic speech recognition (Graham and Hell 1985).

Shape hulls and skeletons are two geometric descriptors for summarizing both global and local spatial properties of point sets. These descriptors, external versus internal, or global versus local, can be generally applied in describing phenomenon on the surface of the earth. For example, the development boundary of a town or city, at another scale the outline of neighborhoods, and at yet another scale the outer shell of individual houses. In each of these examples subsets or components likely exist and identifying them would enhance pattern description. This spatial clustering and identifying subsets can lead to more sensitive descriptions of form and better hypothesizing of underlying processes (Turner 1989; McGarigal and Marks 1995).

In a point set, components are defined as groups of points or spatial clusters that have small within-group variance and large between-group variance. Han *et al.*(2001) provide a thorough review on existing spatial clustering methods in data mining, such as *K*-means, DBSCAN (Density-based Spatial Clustering of Applications with Noise), and CURE (Clustering Using REpresentatives), which also can be used for point sets. Many clustering metrics and tools are used in landscape ecology, forest management, location-based services, public health, urban planning and other GIScience fields. For example, FRAGSTAT has a large collection of such metrics (McGarigal and Marks 1995), and there are more methods and tools in spatial modeling and geostatistics (Toussaint 1988; Gold 1992; Boots 1994; Anselin 1995; Gatrell et al. 1996; Gold 1996; Boots and South 1997; Okabe et al. 2000; Radke and Mu 2001; McGarigal 2002; Wu and Hobbs 2002; Mu and Wang 2008).

Most existing clustering methods use distance as single metric. In terms of the type of space, the distance could be either Euclidean distance or non-Euclidean distance, such as Manhattan distance and traffic network distance. In terms of the type of information, the distance could be the physical spatial distance between spatial objects, or the similarity measure for the attributes of spatial objects. However, single distance-based metric may be not enough to solve some kinds of problems. For example, when we plan to select the sites for wireless stations, we need to identify the clusters of the residents which are not only required to be within the service area of the wireless stations, the counts of residents in the clusters are also needed to be maximal in order to optimize the network planning. In this case, therefore, the clustering of residents is the result considering both distance and count. In addition, the complex phenomena in the world may not be explained by single distance-based metric. Multiple considerations for clustering may aid in understanding the complex processes underlying the patterns or forms. Therefore, it is meaningful to develop the clustering methods using combined metrics.

We consider shape hulls, skeletons, and clustering as a set of three logically related strategies for detecting the geometric properties of a point set. After briefly reviewing common features of α-hull/shapes and β-skeletons we introduce the γ-partition concept and algorithm followed by an investigation of potential applications.

2. *α*-Hull/Shape and *β*-Skeleton

2.1 *α-Hull/Shape*

The α-hull/shape (Edelsbrunner et al. 1983) is a generalization of the Voronoi diagram and thus the convex hull of a finite set of points in the plane. Given a set S of points in the plane and a real α, the α-hull is the intersection of all 'closed disks' of radius $1/\alpha$ that contain all points in S. Different definitions are given for the 'closed disk' based on α value. When $\alpha > 0$, it refers to the disk of radius $1/\alpha$; when $\alpha = 0$, it refers to the half plane; when $\alpha < 0$, it refers to the complement of a disk with radius $-1/\alpha$. The points lying on boundary of the same disk are called α-neighbors. The α-shape is the graph connecting all α-neighbors in the point set.

From negative infinity to positive infinity, all α values construct a spectrum of α-hull/shape for a point set. Fig.1 shows examples of α-hull and α-shape for a given point set. In this example, when $\alpha = 2.5$, the α-shape of the point set is a simple triangle; when $\alpha = 0$, the α-shape is the convex hull of the point set; and when $\alpha = -9.5$, the α-shape reveals more detail, often concavity in the overall shape and subsets of the point set.

Fig.1 Examples of α-hull and α-shape for a given point set

2.2 *β-skeleton*

Inspired by *α*-hull/shapes, *β*-skeletons (Kirkpatrick and Radke 1985; Radke 1988) describe the inner skeletons of point sets. Given a point set *S* and a positive real *β*, if the 'region of influence' of two points x and y contains no point of *S* other than themselves, x and y are regarded as *β*-neighbors. The 'region of influence' is defined based on the parameter *β* and the distance between the two points *d(x, y)*. When $0 < \beta <= 1$, it refers to the intersection of two circles of radius $d(x, y)/(2\beta)$ passing through both x and y; when $\beta > 1$, it refers to the union of the two circles of radius $\beta d(x, y)/2$ that pass through x and y. The network graph linking all *β*-neighbors in a point set is defined as *β*-skeleton. Especially, when $\beta = 1$, the *β*-skeleton is the Gabriel network[1].

From zero to positive infinity, all *β* values construct a spectrum of *β*-skeleton for a point set. From the examples of *β*-skeleton for a given point set shown in Fig.2, we can see that the number of *β*-neighbors decreases with increase of *β* value.

Fig.2 Examples of region of influence for points x and y (distance = 12) and *β*-skeleton for a given point set

2.3 *Common Features*

α-hull/shapes and *β*-skeletons of point sets share the common features that influence the development of the *γ*–partition.

1) Both are computational geometric methods for decomposing point sets;
2) Both involve a single parameter (*α* or *β*) defined here as circle or disk-based;
3) Both can create a spectrum of descriptions of point patterns for a given point set;
4) Both parameters (*α* and *β*) reflect the scales at which the geometric characteristics of point sets are described.

[1] The Gabriel network for a point set is created by adding edges between pairs of points in the source set if there are no other points from the set contained within a circle whose diameter passes through the two points (Gabriel and Sokal 1969).

3. *γ*-Partition

3.1 *Concept*

The *γ*–partition method is developed based on the above common features of *α*-hull/shapes and *β*-skeletons. Some analogy is helpful to shape the idea[2]. *α* -hull/shapes use the disks to identify regions void of points where γ–partition uses disks to identify regions with points in common. The following concepts are given for a point set.

Definition 1: Given a set *S* of *n* points in the plane, the point subset with the maximum count of unclustered points enclosed by a circle of radius *γ* is a cluster. We define this selection criterion as the *Maximal Count Criterion (MCC)*.

Definition 2: The *MCC* is applied iteratively to the point set *S* excluding all points in existing clusters until all points in *S* are assigned to clusters. The partition process is called *γ-partition*.

The parameter *γ* is a positive real that controls the scale at which the inner components of point sets will be analyzed. The larger the *γ* is, the coarser the scale is. For example, given a point set *S* $\{P_i, i=1 \text{ to } 6\}$, Fig.3(*a*) and Fig.3(*b*) show the possible situations of moving the circles of two different radii, i.e. *γ* values, to different locations on the plane where the points are located. In Fig.3(*a*), the distance between points P_1 and P_2 is less than 2γ, and the distances between either point and any other point P_i ($i = 3 \text{ to } 6$) is larger than 2γ. Therefore, P_1 and P_2 will be grouped together and separated from other points in the final *γ*-partition result. When the scale becomes coarser and *γ* increases from 1.5 to 2.5 (Fig.3(*b*)), the inter-point distances among points $\{P_1, P_2, P_3, P_5\}$ are all less than 2γ, which makes the four points possible to be grouped together at this scale.

Fig.3 Possible situations to moving circles on the plane of point sets: (a) circle radius $\gamma = 1.5$; (b) circle radius $\gamma = 2.5$

[2] Imaging the points in a point set are chocolate chips distributing across the ice cream, a scoop of radius $1/\alpha$ ($\alpha > 0$) or $-1/\alpha$ ($\alpha < 0$) is used to dig out the ice cream as much as possible but not to move or remove any chocolate chips. When no more ice cream can be scooped, the shape of the remaining ice cream containing all of the chocolate chips is the *α*-hull. In *γ*–partition method, instead of digging out the ice cream, a scoop of radius *γ* ($\gamma > 0$) is used to scoop out the chocolate chips as many as possible every time until all the chocolate chips are dug out. The ‘chocolate chips’ dug out by each scoop are regarded as a subset (of the point set). Finally, all the chocolate chips are grouped into several mutually exclusive and totally exhaustive subsets.

In γ-partition, the MCC adopts a density-based priority. For example, in Fig.3(*a*), point P_5 can be covered by either circle C_2 or C_3. Since circle C_2 can cover more points than C_3, or the point density in circle C_2 is higher than that in circle C_3, point P_5 is grouped with points P_3 and P_4 instead of P_6 in the final partition result.

In some cases, a unique γ-partition result cannot be reached only with the MCC, such as a regular lattice of points in a point set. Therefore, other constraints may be needed according to the requirements of specific applications. Here, two examples of constrains are given: distance constraint and corrected perimeter/area ratio (CPA) constraint.

The distance constraint is defined such that, with the same count of points, the point subset with the shorter average inter-point distance has a higher priority of becoming a cluster. Fig.4 shows two different point subsets that result using the MCC where the counts of points in both subsets are four. However, the average inter-point distance of the point subset $\{P_1, P_2, P_3, P_4\}$ is smaller than that of the point subset $\{P_4, P_5, P_6, P_7\}$. Applying the distance constraint, point P_4 will be grouped as Fig.4(*a*) instead of Fig.4(*b*).

Fig.4 Distance constraint for two different point subsets by using the MCC: (a) point subset $\{P_1, P_2, P_3, P_4\}$; (b) point subset $\{P_4, P_5, P_6, P_7\}$

The CPA, calculated with Eq.1, is a standardized index to characterize perimeter/area ratio. CPA ranges from 1.0 to infinity, and the value of 1.0 represents perfect circles.

$$CPA = \frac{1}{2\sqrt{\pi}}\left(\frac{P}{\sqrt{A}}\right) \qquad (1)$$

Where: *P: Perimeter of the convex hull of the point subset*
A: Area of the convex hull of the point subset

The CPA constraint, with the same counts of points, defines the point subset with the more compact convex hull shape as having a higher priority to become a cluster. Fig.5 shows two different point subsets that result using the MCC with the point counts in both subsets being four. However, the CPA of the convex hull of the point subset $\{P_1, P_2, P_3, P_4\}$ is 1.128, whereas that of the point subset $\{P_4, P_5, P_6, P_7\}$ is 1.197. Under the CPA constraint, point P_4 will be grouped as Fig.5(*a*) instead of Fig.5(*b*).

Fig.5 The CPA constraint for two different point subsets by using the MCC*:* (a) point subset $\{P_1, P_2, P_3, P_4\}$; (b) point subset $\{P_4, P_5, P_6, P_7\}$

Fig.6 illustrates the results using the γ–partition method with a distance constraint to partition a point set involving 18 points. In order to show the partition results clearly, both the convex hull and covering circle of each cluster are drawn. The results are coherent with our common understanding that the general trend of the count of groups decreases with the increase of γ values. When γ is small, 0.05, each point is a cluster, and when γ is large, 2.2, all points will be grouped into one cluster. The cluster count does not continuously decrease with the increase of γ values, as shown in the cases of $\gamma = 1.2$ and $\gamma = 1.4$.

Fig.6 The γ-partition results for a 18-point set with different γ values (*the circles may not reflect the real γ value due to the arrangement of graphs*)

3.2 *Algorithm*

The core of the algorithm implementing the γ-partition solves how to find the point subset with the MCC in a point set. Fig.7 shows an example where there are three points $\{P_1, P_2, P_3\}$. We have the following definition and observation.

Fig.7 Common intersecting area $\{m_1, m_2, m_3\}$ among buffer circles(r) C_1, C_2 and C_3

Definition 3: For a point set S $\{P_i, i=1 \text{ to } n\}$, the circle centered at the point P_i with radius r is defined as the *buffer circle(r)*of point P_i.

Observation 1: These three points can be covered by circle C with radius r. And the buffer circles(r) of these points $\{C_1, C_2, C_3\}$ have a common intersecting area $\{m_1, m_2, m_3\}$ (i.e. shadowed area in Fig.7)

Based on above observation, we present the following lemma.

Lemma 1: The sufficient and necessary condition for a number of points (the count of points is larger than 1) can be covered by a circle with radius *r* is that all the buffer circles(r) of these points have a common intersecting space (either point or area).

Proof: Given a point set *S* including more than one point, if there is a common intersecting part in the buffer circles(r) of all points, with a random location *P* in this common intersecting part as the center, a circle *C* with radius *r* can be drawn. Since *P* is within all buffer circles(r), the distance from *P* to any point in *S* is smaller than or equal to *r*. That means all the points in *S* are all within the circle *C*. Thus a common intersecting space is the sufficient condition. If the points in *S* can be covered by a circle *C* with radius *r*, the distance from any point to the center of circle C is smaller than or equal to *r*. Then, the center of circle *C* is necessarily within the buffer circles(r) of all points. In other words, there exists the common intersecting space among the buffer circles(r), at least including the center of circle *C*. Therefore, a common intersecting space is also a necessary condition.

From the example in Fig.7, the following definition and observation are derived.

Definition 4: If the intersecting arcs of a buffer circle(r) *C* with other buffer circles(r) have one or more common spaces (point or arc), these common spaces are defined as *Common Intersecting Arcs (CIA)* of buffer circle(r) *C* including the case where the common space is just a point. The count of intersecting arcs forming the *CIA* is defined as an *intersecting numbe*r. Among all the *CIA*s on the same buffer circle(r), the one with the maximum *intersecting number* is defined as *Maximal Common Intersecting Arc (MCIA).*

Observation 2: The arcs comprising the boundaries of common intersecting space {m_1, m_2, m_3} are all the CIAs and all their intersecting numbers are two. For example, the boundary *arc*[m_1,m_2] is the CIA of the intersecting arcs of buffer circle(r) C_3 with C_1 *and* C_2 (*arc*[n_1, m_2] and *arc*[m_1, n_2]). Since there are a total of three buffer circles(r) {C_1, C_2, C_3}, each boundary of this area is also the MCIA of each buffer circle(r).

It is worthy of noting that even if there is a common intersecting space existing among *n* buffer circles(r), the count of the boundaries does not necessarily equal to *n*. Fig.8 shows a scenario where there are only two boundaries of the common intersecting area existing among three buffer circles(r) {C_1, C_2, C_3}.

Fig.8 Two-boundary common intersecting area among three buffer circles(r)

Based on the above observations, the following lemma is presented.

Lemma 2: The sufficient and necessary condition for a common intersecting space existing among *n* buffer circles(r) (*n* is larger than 1) is that at least one of these buffer circles(r) has a CIA whose intersecting number is *n-1*.

Proof: As one buffer circle(r) can only have one intersecting arc with another buffer circle(r), from the definition 4, we know that the CIA whose intersecting number is *n-1* is necessarily on one buffer circle(r) and within other *n-1* buffer circles(r). It is an obvious and sufficient condition. If these *n* buffer circles(r) have one common intersecting space, one boundary of the area or the point if the common intersecting space is a point must be on one of the buffer circles(r). Since this point or boundary also lies on or within all other buffer circles(r), it must be the CIA of this circle with all other *n-1* circles (its intersecting number is *n-1*). The necessary condition is proved.

Based on lemmas 1 and 2, we can identify the points with the MCC in a planer set *S* of *n* points by identifying the maximum count (*m*) of points so that at least one of their buffer circles(r) has a CIA with other *m-1* buffer circles(r). The steps of algorithm for the *γ*–partition are listed as below.

Step 1: Construct buffer circle(r) intersection matrix.

An *n by n* matrix is used to record the locational relationship between each pair of the buffer circles(r) of all points in the point set. Assuming only one point is located in each position, if buffer circles(r) of point *i* and *j* are intersected, information about the intersecting arc on buffer circle(r) of point *i*, denoted as $Arc_{i->j}$, and that on buffer circle(r) of point *j*, denoted as $Arc_{j->i}$, are recorded as matrix elements m_{ij} and m_{ji}, respectively. It is defined that the buffer circle(r) does not intersect with itself.

The direction of intersecting arc is described counter clockwise (Fig.9). The data structure of $Arc_{i->j}$ is:

$Arc_{i->j}$ {
i (x,y) *Center point coordinates of the circle where* $Arc_{i->j}$ *is*
p (x,y) *Starting point coordinates of* $Arc_{i->j}$
q (x,y) *Ending point coordinates of* $Arc_{i->j}$
θ *Starting angle of* $Arc_{i->j}$*, range: [0 - 2π)*
φ *Ending angle of* $Arc_{i->j}$*, range: [0 - 2π)*
}

The parameters of $Arc_{i->j}$ can then be calculated for all buffer circles(r) using the circle equation and planar Cartesian coordinate system translation.

Fig.9 The sketch map of $Arc_{i->j}$

Step 2: Find the MCIA on each buffer circle

For each row of the buffer circle(r) intersection matrix, the MCIA on each buffer circle(r) can be found. Fig.10 shows an example of an efficient algorithm for implementing this task. The left graph in Fig.10 shows that there is a buffer circle(r) C_1 which simultaneously intersects with other four buffer circles(r) $\{C_2, C_3, C_4, C_5\}$. $\{Arc_{1\text{->}j}, j = 2 \text{ to } 5\}$ represents the intersecting arcs on buffer circle(r) C_1 with each of other four buffer circles(r). In the right graph in Fig.10, buffer circle(r) C_1 is transformed from a circular metric to a linear metric ranging from 0 to 2π. All intersecting arcs on this circle are mapped as line segments from their starting angles to ending angles. The variable *ArcOverlap* is used to record the intersecting numbers of CIAs on the buffer circle(r) C_1. Assuming there is a vertical cursor moving horizontally from 0 to 2π, *ArcOverlap* will increase by 1 when the cursor intersects with a starting point of an arc, and decrease by 1 when the cursor moves away from an ending point of an arc. Finally, the maximum *ArcOverlap* indicates the intersecting number of the MCIA on this buffer circle(r). In the example shown on Fig.10, the maximum *ArcOverlap* for buffer circle(r) C_1 is 3, and the MCIA on this buffer circle(r), the segment between the two parallel dash lines, is caused by the overlap of 3 intersecting arcs $\{Arc_{1\text{->}2}, Arc_{1\text{->}3}, Arc_{1\text{->}4}\}$. In turn, it is demonstrated that the center points of the four buffer circles(r) $\{C_1, C_2, C_3, C_4\}$ can be covered by a circle with radius r.

Fig.10 Seek the maximum common intersecting arc or point on buffer circle C_1

The starting angles and ending angles of all intersecting arcs on a buffer circle(r) need to be ranked in ascending order before obtaining the maximum *ArcOverlap*. In a special situation, if an arc crosses the zero degree line, i.e., its starting angle is larger than its ending angle, this arc needs to be divided into two consecutive arcs: *arc[original starting angle, 2π]* and *arc[0, original ending angle]*.

Step 3: Identify the point subset(s) with the maximum count in whole point set

By comparing the MCIA(s) of each buffer circle, the MCIA(s) with the largest intersecting numbers in the entire point set can be determined, and the point subset(s) with the maximum count in the entire point set can be identified.

Step 4: (If applicable) Apply other constraints.

According to the requirements of specific applications, some constraints, such as distance and CPA, may be applied to prioritize the point subsets with same count of points as discussed in earlier sections.

Step 5: Remove existing point clusters from the point set, and repeat step 1–4 until all points in the point set are clustered.

The γ–partition is implemented in ArcGIS™ to facilitate its application (Fig.11).

Fig.11 User interface of γ–partition in ArcGIS™

4. Potential Applications

The γ–partition is likely useful in a variety of fields. In practice, the γ–partition method might likely need to be tailored to optimally serve unique applications.

1) A statistic measure for testing global spatial point pattern analysis

Similar to metrics such as the nearest neighbor distance and the point density per quadrat methods used to examine global spatial point patterns (clustered, random, or regular), some measures can be derived from the γ– partition results. The count of clusters in the γ–partition result can be used as a statistical measure and compared to those that would result from a Poisson distribution. When γ is the same, the more clustered the point set, and the fewer clusters there are likely to be in the γ-partition result. Fig.12 shows the results comparing the counts of clusters in γ–partition results of two given point sets to those of 99 completely spatial random point sets with the significance level of 0.01. Fig.12(*a*) shows a clustered point set of 100 points simulated by a Poisson cluster process (Diggle 2003), while Fig.12(*b*) shows a regular point set of 100 points simulated by a Matern inhibition process (Diggle 2003) with inhibition distance of 0.04. All of the point sets are simulated using Spatstat ™ package (Baddeley and Turner 2005) for software R™ (R Development Core Team 2009).

Fig.12 The γ–partition results for global spatial point pattern analysis: (a) simulated clustered point pattern; (b) simulated regular point pattern; (c) count of clusters against γ value.

From Fig.12(c), it is observed that the curve of the clustered point set is obviously below the envelope of complete spatial randomness, while the curve of the regular point set almost completely falls within the envelope, except in the range of γ from 0 to 0.04, where the curve is a little bit beyond the envelope of the complete spatial randomness. As a result, the count of the clusters in the γ-partition results can reflect the characteristics of the global spatial point pattern.

2) A heuristic method for maximal point covering

The γ–partition also has potential in location-based planning, where there is often a need to locate a limited number of facilities so that the service areas of these facilities can serve as many points (residential disease locations) as possible. It is quite similar to the Maximal Covering Location Problem (Church and ReVelle 1974). The difference being in the Maximal Covering Location Problem there is a set of pre-defined sites for the facilities to choose, while in the γ–partition there is not. Since the maximum count of points covered by one facility service area is identified in each iteration during the procedure of the γ–partition, the γ–partition can be considered a *greedy* algorithm which always uses the best immediate, or local, solution while finding an answer (Black 2005). Therefore, the γ–partition may provide globally optimal or sub-optimal solutions. Fig.13 shows the locations of first five facilities and the points covered.

Fig.13 Locations of first five facilities and the count of points each can cover

3) A building block for constructing hierarchical clustering method

If the γ–partition is directly used to partition a point set, the results may not be ideal. Since the *MCC* is only applied to a point set excluding existing clusters, some clusters with overlapping convex hulls are actually more reasonable to be grouped. In addition, a circle-based method is not good for detecting clusters with irregular shapes. To tackle this problem, a hierarchical γ-partition method should be constructed to improve the clustering effect.

In a hierarchical γ-partition, a point set is first divided into initial groups using the γ–partition method. These groups are then merged if their centroids can be grouped using the γ– partition method. The process of clustering can run repeatedly until no more groups can be merged. Fig.14 illustrates the result of using a hierarchical γ-partition of a given point set with three γ values. The hierarchical clustering method brings more flexibility in the basic γ– partition and can broaden the utilization of the method.

Fig.14 Examples of hierarchical γ-partition method for a given point set

5. Conclusions

Spatial clustering methods are essential tools for point pattern analysis because they can maps and help one discover strong relationships between points. In this paper, a theoretic framework based on computational morphology is proposed, where spatial clustering methods play a role - a complementary process to shape hulls and skeletons in describing point patterns. The development of the γ–partition extends geometric descriptors of point pattern concepts and methods such as α-hull/shapes and β-skeletons as they share the concept of a parameterized disk or circle based kernel in their generation. The γ–partition has practical value for a broad number of applications including: global spatial point pattern testing, location-based planning, and spatial clustering. Heuristic modifications to the method described here increase its flexibly and usefulness.

Acknowledgement

The research is partially sponsored by the University of Georgia Research Foundation (UGARF), through the Faculty Research Grant.

References

Anselin, L. 1995. Local indicators of spatial association - LISA. *Geographical Analysis* 27 (2):93-115.

Baddeley, A., and Turner, R. 2005. spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software* 12 (6):1-42.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. 1990. The R*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.* 19 (2):322-331.

Black, P. 2005. "greedy algorithm". In *Dictionary of Algorithms and Data Structures [online]*, ed. P. E. Black, U.S. National Institute of Standards and Technology. 2 February 2005 [cited 7 May 2010]. Available from http://www.itl.nist.gov/div897/sqg/dads/HTML/greedyalgo.html

Boots, B. N. 1994. Visulizing spatial autocorrelation in point data. *Geographical Systems* 1:255-266.

Boots, B. N., and South, R. 1997. Modelling retail trade areas using higher-order, multiplicatively weighted Voronoi diagrams. *Journal of Retailing* 73 (4):519-536.

Church, R., and ReVelle, C. 1974. The maximal covering location problem. *Papers in Regional Science* 32 (1):101-118.

Diggle, P. 2003. *Statistical analysis of spatial point patterns*. 2nd ed. London: Arnold.

Edelsbrunner, H., Kirkpatrick, D. G., and Seidel, R. 1983. On the Shape of a Set of Points in the Plane. *Ieee Transactions on Information Theory* 29 (4):551-559.

Gabriel, K. R., and Sokal, R. R. 1969. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology* 18 (3):259-270.

Gatrell, A. C., Bailey, T. C., Diggle, P. J., and Rowlingson, B. S. 1996. Spatial Point Pattern Analysis and Its Application in Geographical Epidemiology. *Transactions of the Institute of British Geographers* 21 (1):256-274.

Gold, C. M. 1992. The Meaning of "Neighbour". In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 220-235. Berline: Springer-Verlag.

Gold, C. M. 1996. Outside-in: an Alternative Approach to Forest Map Digitizing. *Internatinal Journal of Geographical Information Systems* 10 (3):291-310.

Graham, R. L., and Hell, P. 1985. On the History of the Minimum Spanning Tree Problem. *Annals of the History of Computing* 7 (1):43-57.

Guttman, A. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*. Boston, Massachusetts: ACM.

Han, J., Kamber, M., and Tung, A. K. H. 2001. Spatial clustering methods in data mining: A survey. In *Geographic data mining and knowledge discovery*, eds. H. J. Miller and J. Han, 201-231. London ; New York: Taylor & Francis.

Kirkpatrick, D. G., and Radke, J. D. 1985. A framework for computational morphology. In *Computational Geometry*, ed. G. T. Toussaint, 217-248. Amsterdam: North-Holland: Elsevier Science Pub. Co.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. 2005. *Geographical Information Systems and Science*. 2nd ed. Chichester: Wiley.

McGarigal, K. 2002. Landscape pattern metrics. In *Encyclopedia of Environmentrics*, eds. A. H. El-Shaarawi and W. W. Piegorsch, 1135-1142. Sussex, England: John Wiley & Sons.

McGarigal, K., and Marks, B. J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Tech. Rep. PNW-351: USDA Forest Service.

Mu, L., and Wang, F. H. 2008. A scale-space clustering method: Mitigating the effect of scale in the analysis of zone-based data. *Annals Of The Association Of American Geographers* 98 (1):85-101.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. 2000. *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*. New York: John Wiley & Sons Ltd.

R Development Core Team. 2009. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0, Available from http://www.R-project.org

Radke, J., and Mu, L. 2001. Spatial Decompositions and Modeling to Level Access to Social Programs. In *21st Annual ESRI International User Conference*. San Diego, CA.

Radke, J. D. 1988. On the shape of a set of points. In *Computational Morphology*, ed. I. G. Toussaint, 105–136. North Holland: Elsevier Science Publishers B.V. .

Sampath, A., and Shan, J. 2007. Building boundary tracing and regularization from airborne lidar point clouds. *Photogrammetric Engineering and Remote Sensing* 73 (7):805-812.

Toussaint, G. T. ed. 1988. *Computational Morphology*. New York: Elsevier Science Publishers B.V.

Turner, M. G. 1989. Landscape ecology: the effect of pattern on process. *Annual Review of Ecology and Systematics* 20:171-197.

Wu, J. G., and Hobbs, R. 2002. Key issues and research priorities in landscape ecology: An idiosyncratic synthesis. *Landscape Ecology* 17 (4):355-365.

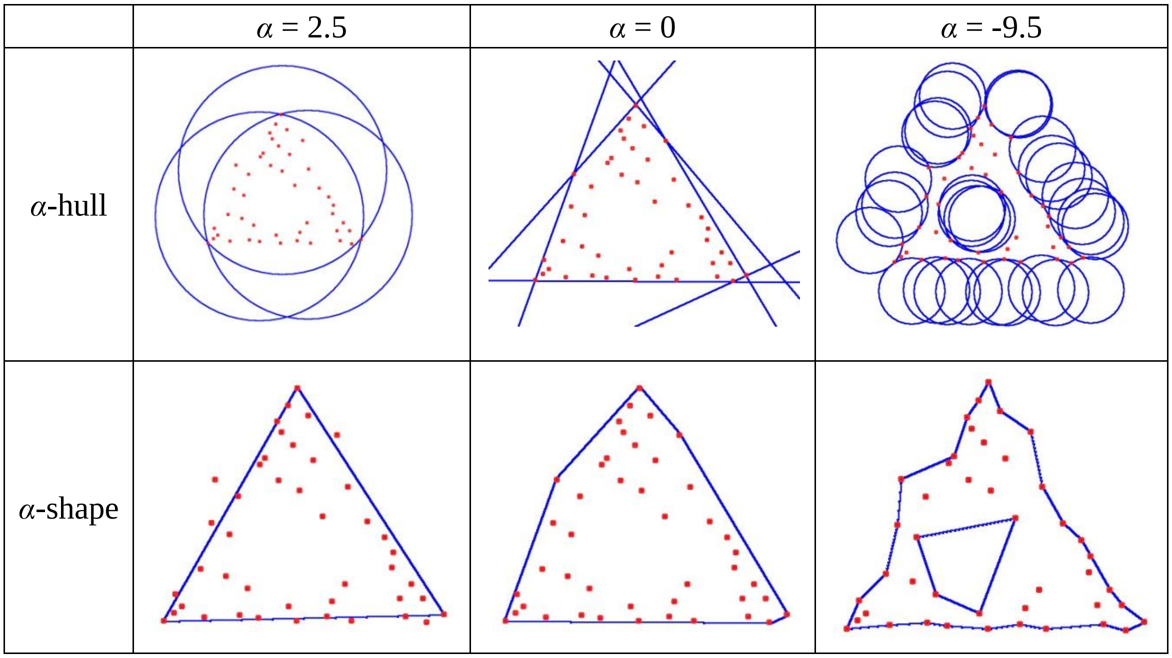


Fig.1 Examples of α-hull and α-shape for a given point set

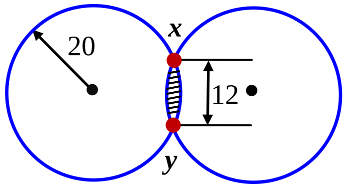
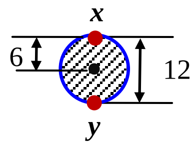
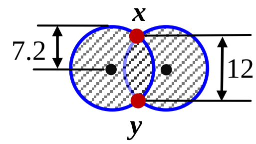
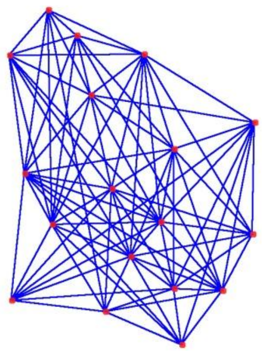
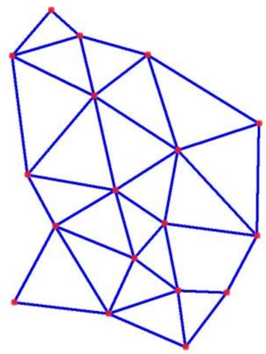
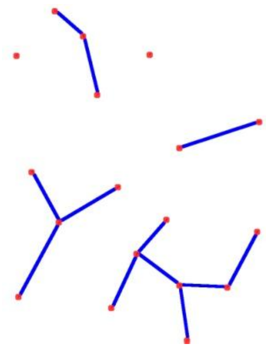
	$\beta = 0.3$	$\beta = 1$	$\beta = 1.2$
Region of influence (*Shaded area*)			
β-skeleton			

Fig.2 Examples of region of influence for points x and y (distance = 12) and β-skeleton for a given point set

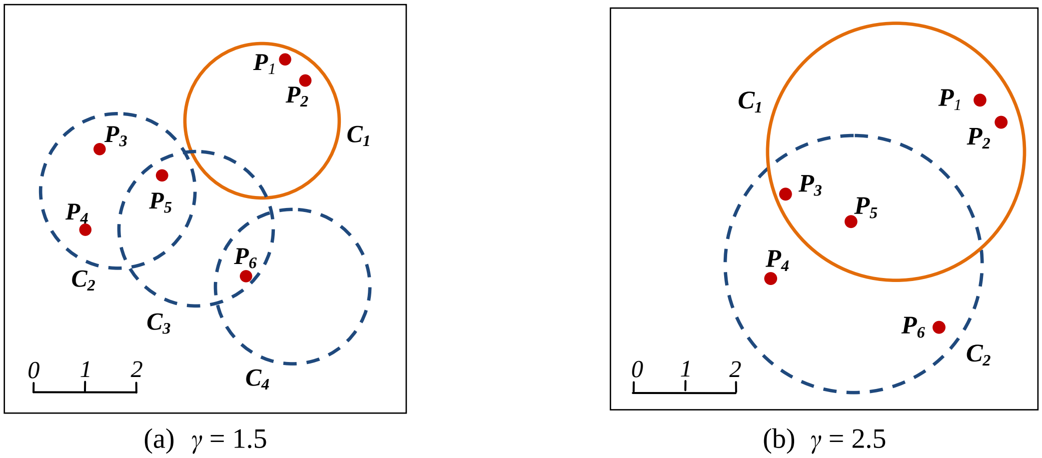
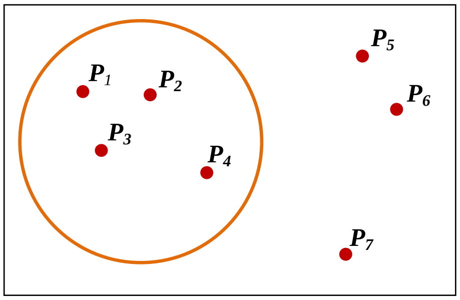
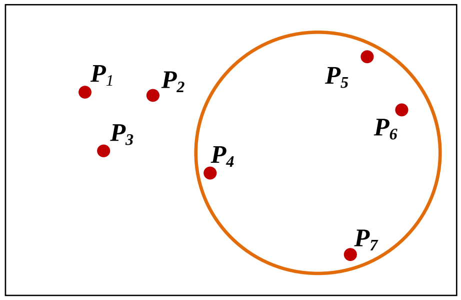


Fig.3 Possible situations to moving circles on the plane of point sets: (a) circle radius $\gamma = 1.5$; (b) circle radius $\gamma = 2.5$

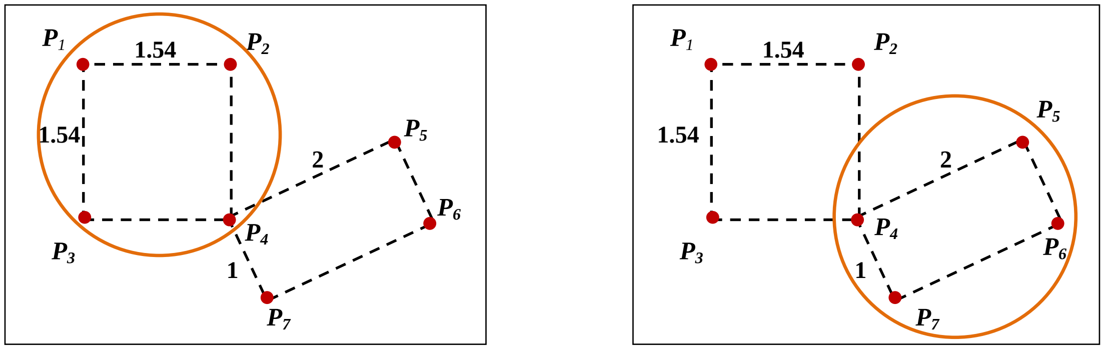


(a) Point subset {P_1, P_2, P_3, P_4}



(b) Point subset {P_4, P_5, P_6, P_7}

Fig.4 Distance constraint for two different point subsets by using the MCC*:* (a) point subset {P_1, P_2, P_3, P_4}; (b) point subset {P_4, P_5, P_6, P_7}



(a) Point subset $\{P_1, P_2, P_3, P_4\}$ (b) Point subset $\{P_4, P_5, P_6, P_7\}$

Fig.5 The CPA constraint for two different point subsets by using the MCC: (a) point subset $\{P_1, P_2, P_3, P_4\}$; (b) point subset $\{P_4, P_5, P_6, P_7\}$

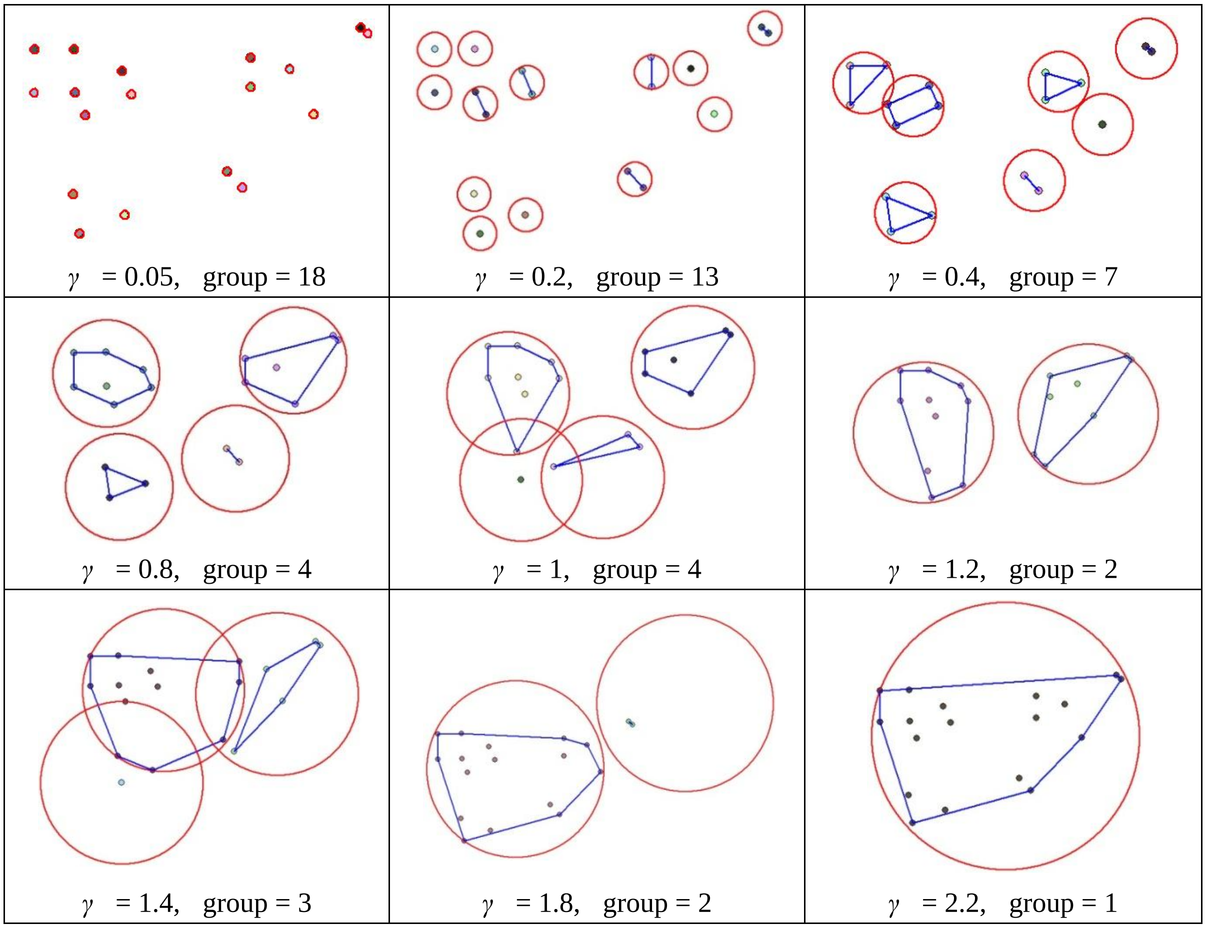


Fig.6 The γ-partition results for a 18-point set with different γ values (*the circles may not reflect the real* γ *value due to the arrangement of graphs*)

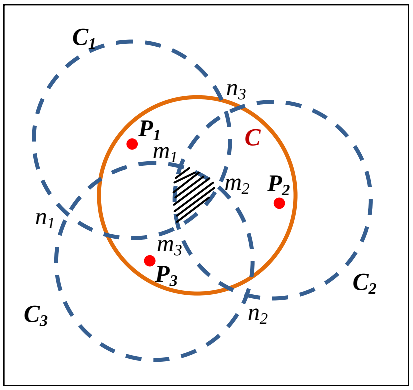


Fig.7 Common intersecting area {m_1, m_2, m_3} among buffer circles(r) C_1, C_2 and C_3

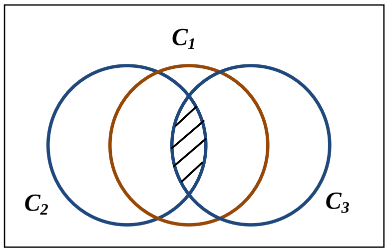


Fig.8 Two-boundary common intersecting area among three buffer circles(r)

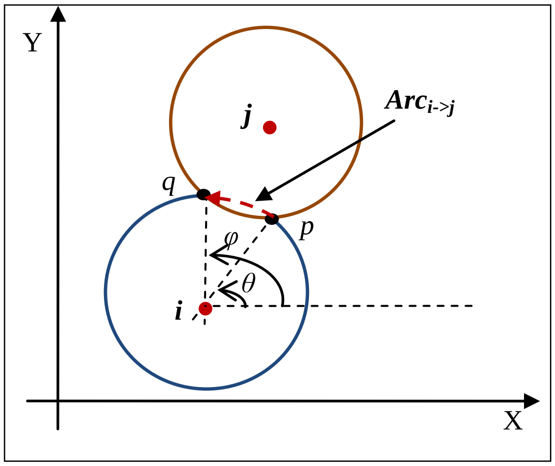


Fig.9 The sketch map of $Arc_{i\text{->}j}$

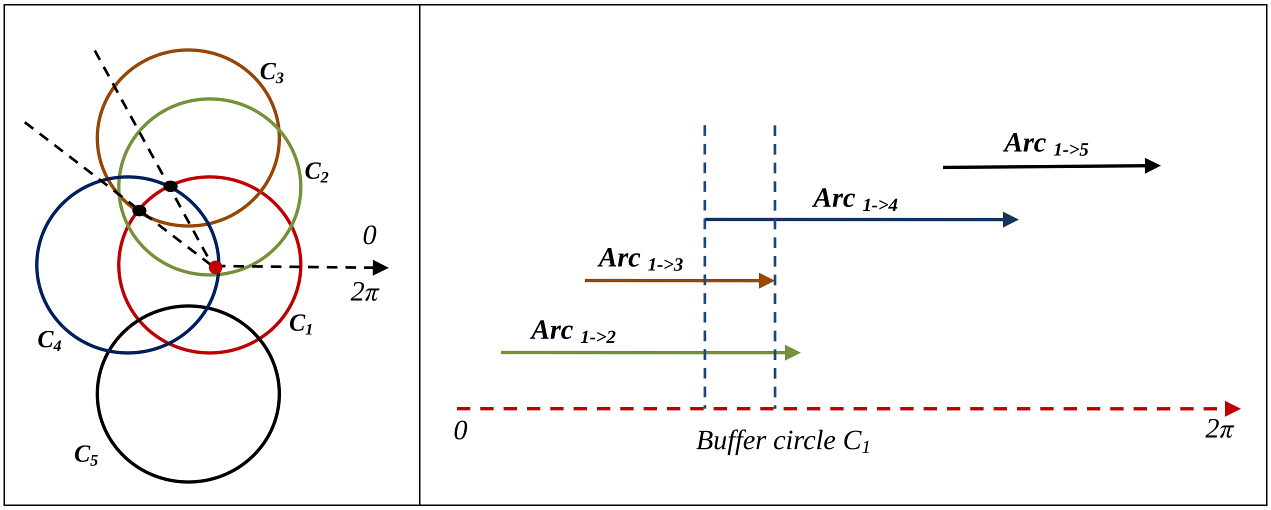


Fig.10 Seek the maximum common intersecting arc or point on buffer circle C_1

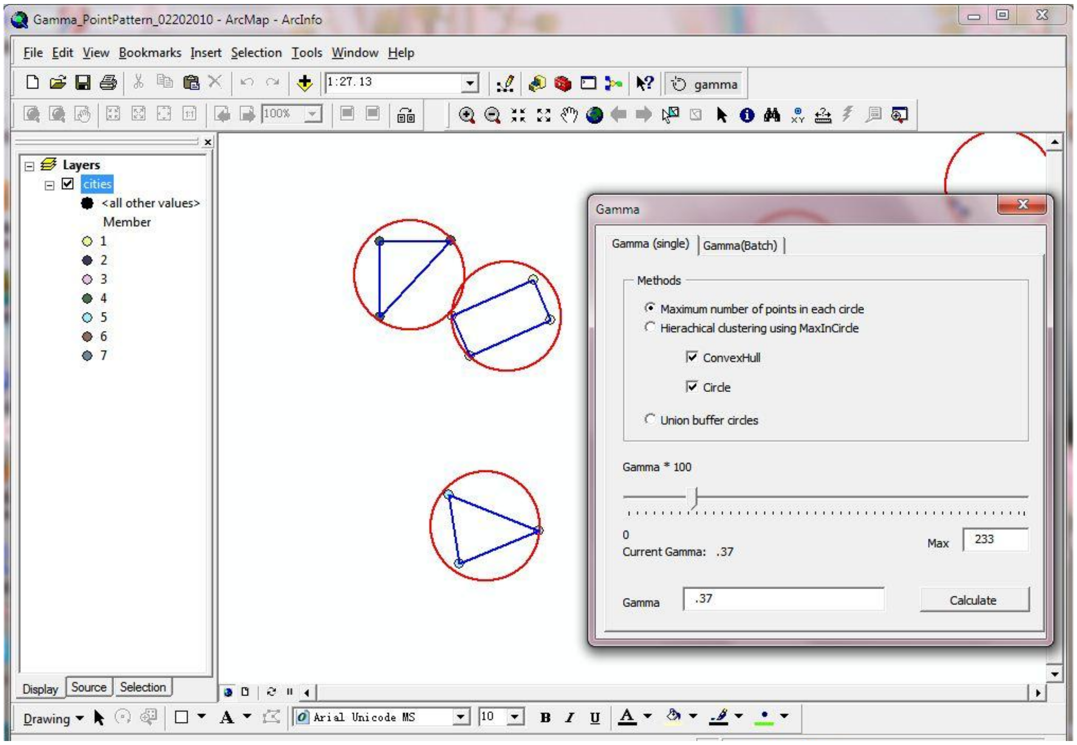
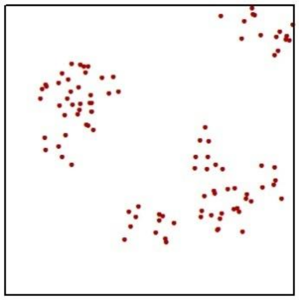
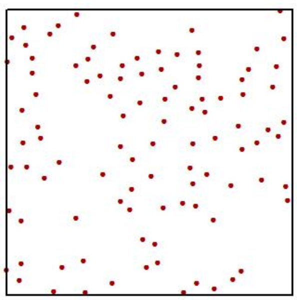


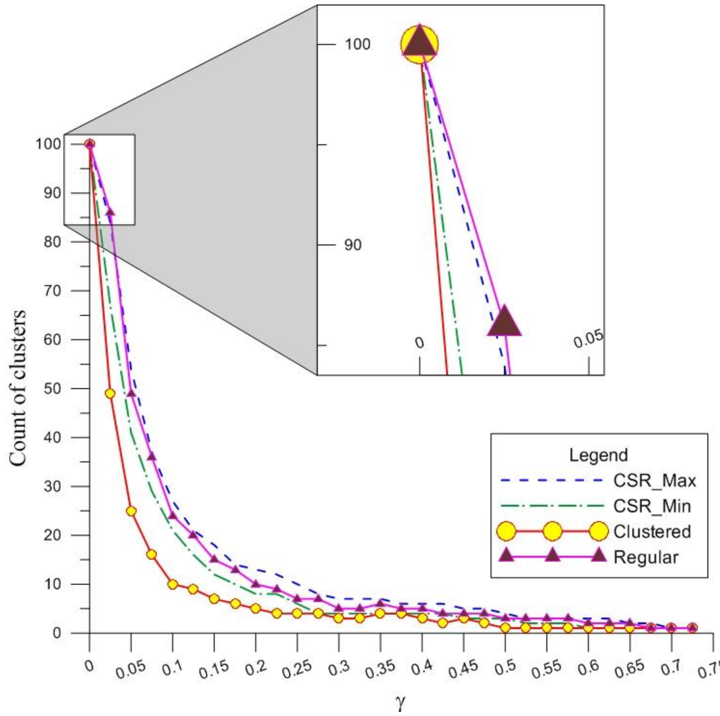
Fig.11 User interface of γ–partition in ArcGIS^{TM}



(a) Simulated clustered point pattern



(b) Simulated regular point pattern



(c) Count of clusters against γ value

Fig.12 The γ–partition results for global spatial point pattern analysis: (a) simulated clustered point pattern; (b) simulated regular point pattern; (c) count of clusters against γ value.

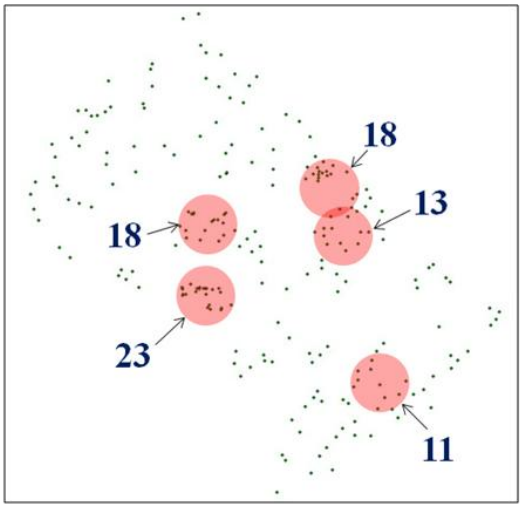


Fig.13 Locations of first five facilities and the count of points each can cover

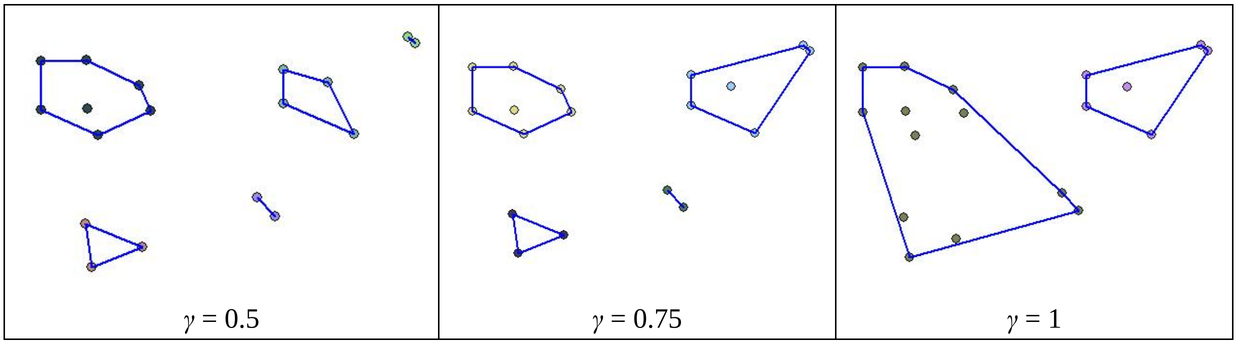


Fig.14 Examples of hierarchical γ-partition method for a given point set