ELSEVIER

Pattern Recognition Letters 17 (1996) 11–17

Pattern Recognition Letters

A new definition of neighborhood of a point in multi-dimensional space

B.B. Chaudhuri

Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700035, India

Received 2 May 1994; revised 17 August 1995

Abstract

Given a set of points in multi-dimensional space, we propose a new definition for the neighbors of an arbitrary point P. The definition tries to capture the idea that the neighbors should be as near to P and as symmetrically placed around P as possible. In contrast, the conventional nearest neighborhood considers only nearness as the criterion for neighborhood. We propose an iterative procedure to compute the neighbors where the first neighbor is the nearest neighbor. The second and other neighbors are chosen so that at any stage the distance between the centroid of the neighbors and P is as small as possible. The centroid criterion takes care of symmetrical placement of the neighbors. One can use median instead of centroid to define the neighbors. The new definition is free from any user-specified parameter and can be used for pattern classification, clustering and low-level description of dot patterns.

Keywords: Neighborhood; Classification; Clustering; Pattern recognition; Image processing

1. Introduction

The concept of neighborhood is useful and important in pattern clustering, computational geometry and image processing problems. Consider a set of points in $\mathbb{R}^n$. The neighbors of a point P can be loosely defined as the points in the "proximity" of P. Usually, the Euclidean norm is used to compute proximity. Thus, the neighbors of a point P can be defined as the points that are nearest to P in terms of Euclidean distance. Such neighbors are called *nearest neighbors* (NN).

However, the concept of neighborhood should be such that (a) the neighbors are as near to the candidate P as possible and (b) the neighbors sit as symmetrically around P as possible. The *nearest neighborhood* takes care of property (a) only. Thus, NNs may not be symmetrical around P if the data set is not homogeneous in its neighborhood. To tackle the problem, O'Callaghan (1975) proposed an alternative definition of neighborhood in $\mathbb{R}^2$ with two constraints. One is the distance constraint, which excludes points further than a specified distance, say d, as neighbors. The other is the direction constraint which excludes points essentially behind other chosen points with respect to P. The direction constraint is imposed by an angle parameter θ. Thus, the definition is dependent on two user-specified quantities d and θ, and the number of neighbors obtained for some specified d and θ may not be equal to the desired number of neighbors. Also, the approach does not provide any ordering scheme of the neigh-

SSDI 0167-8655(95)00093-3

bors. In addition, extension of the approach to data in higher-dimensional space is not a straightforward task.

To obtain symmetrical neighbors that is data driven and free from user-specified parameters we initially considered Voronoi tesselation (Toriwaki and Yokoi, 1988). In this case, a data point P is enclosed by a convex polygon $C(P)$ so that any point in $C(P)$ is nearer to P than any other datum. Each polygon has a few neighboring polygons sharing common edges. The data enclosed by these neighboring polygons denote the first layer of neighbors of P. The second and higher layers can be similarly defined. If the user wants a specified number k of neighbors, he/she can start from the first layer and pick data nearest to P. If the first layer contains fewer than k data, then the second and higher layers are considered in a similar manner. However, the problem of the method is that its extension in $\mathbb{R}^n$, $n > 2$, is computationally very costly.

We propose a simple and intuitively appealing definition of neighborhood that does not need any user-specified parameter and can be readily computed in any multi-dimensional space. The algorithm is an iterative one where the kth neighbor is found using the $k-1$ neighbors computed earlier. Extension of the definition to object is also possible. The proposed neighborhood, called *nearest centroid* (NC) *neighborhood* is defined and its properties are discussed in Section 2. A variant of the proposal named *nearest median* (NM) *neighborhood* is also stated. Possible applications of the proposed neighborhood are demonstrated in Section 3.

2. Nearest centroid (NC) and nearest median (NM) neighbors

The basic idea behind the nearest centroid neighborhood is as follows. Let P be a point whose k neighbors should be found in a set of points S_0. These k neighbors are such that (a) they are as near to P as possible, and (b) their mean or centroid is also as near to P as possible.

To satisfy the two conditions, we propose an iterative procedure where the first neighbor of P is its nearest neighbor, say R. The second neighbor Q is such that the centroid of R and Q is nearest to P. Note that the nearness of R forces us to choose Q to be near to P. Thus, if we compute the kth neighbor using the $k-1$ neighbors chosen previously by the nearest centroid criterion, both conditions (a) and (b) are met quite well. The basic steps of the nearest centroid neighbor (NCN) algorithm are given below.

Algorithm NCN

Step 1. (Initialization) $S \Leftarrow S_0 - P$; $T = \emptyset$; $j = 0$.
Find the nearest neighbor (in terms of Euclidean distance) of P in S. Let it be R (resolve tie arbitrarily).
Make $T \Leftarrow R$; $S \Leftarrow S - R$.

Step 2. $j \Leftarrow j + 1$; For each point $Q \in S$ find the centroid M of points in $T \cup Q$. Choose the point as Q_0 for which M is nearest to P in terms of Euclidean distance. In case of a tie choose as Q_0 the point that is farthest from the neighbor chosen in the previous iteration.

Step 3. Make $T \Leftarrow T \cup Q_0$; $S \Leftarrow S - Q_0$;
If $j = k$ return. [T is the subset of k NC neighbors of P.] Else, go to Step 2.

The concept is explained through one example in Fig. 1. Here the first neighbor of P, shown by the number 1, is the first nearest neighbor. The second neighbor is not the second nearest neighbor (which is B in Fig. 1). Rather, the algorithm tries to pick a point in the opposite direction (and with equal dis-

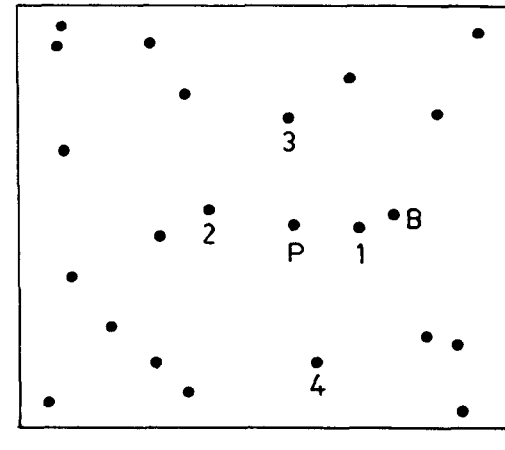

Fig. 1. An example of nearest centroid neighbors (the NC neighbors are numbered in increasing order).

tance) of the first neighbor with respect to P so that the centroid is minimally away from P. Because of the centroid criterion, the chosen neighboring points sit all around P. Also, P being nearest to the centroid, acts as the unbiased representative of its neighbors.

Clearly, unlike O'Callaghan's (1975) method our proposal is free from user-specified parameters. It can be computed inexpensively for data in any dimension. More specifically, computation of one NC neighbor of any point requires at most N centroid and distance computations as well as N comparisons to find the minimum of the distances. Therefore, k NC neighbors of a point can be computed in $O(kN)$ time. Note that k nearest neighbors of one point can also be computed in $O(kN)$ time.

On the other hand, Voronoi diagram computation is dependent on the dimensionality of data. In 2D, the Voronoi diagram and its dual Delaunay triangulation can be optimally computed in $O(N \log N)$. At higher dimensions, $d \geq 3$, the Voronoi diagram can be found from the convex hull in $d+1$ dimensions. The convex hull in d dimensions can be computed optimally in $O(N^{\lfloor d/2 \rfloor})$ by the algorithm of Chazelle (1991). Thus the Voronoi diagram based neighborhood is computationally more expensive than the proposed approach. The Delauney triangulation which also defines the neighborhood can be computed in expected $O(N^3)$ by solving a linear programming problem (Megiddo, 1984), although the worst-case complexity is much higher. Even $O(N^3)$ is more than the complexity of our proposed algorithm.

Instead of nearest centroid we can use nearest median to define the neighbors. We use the following definition for the purpose.

Definition 1. The *median point* of a set of points S is the point whose coordinates are the medians of the respective coordinates of the points in S.

If in Step 2 of the above procedure, M denotes the median of points in $T \cup Q$ then the resulting neighbors may be called *nearest median* (NM) neighbors.

NC and NM neighborhood for objects of finite size

To extend the proposed nearest centroid neighborhood definition for objects of finite size (a) a definition of distance between objects and (b) a representative point for each object is necessary. The representative points are needed to find the centroid for NC neighborhood computation.

One may be tempted to use the centroid of the object as its representative point. But we did not consider the centroid for several reasons. The centroid may not be a point belonging to the object (e.g. for an annular ring). Also, the distance between centroids does not always reflect our intuitive idea about distance. For example, two concentric rings of different diameters would have the same centroid, leading to zero distance. Thus, the Euclidean distance between centroids does not define a metric for the object distance.

The *Hausdorff distance* (HD), on the other hand, is a metric for objects in space and we use it for our purpose. For two objects A and B the HD is defined as follows. For each point P of A find the Euclidean distance of the nearest point of B. Let the largest of these distances be $d(A, B)$. Compute $d(B, A)$ in a similar manner. The larger of $d(A, B)$ and $d(B, A)$ is the HD between A and B.

For our purpose we mark the points $P \in A$ and $Q \in B$ so that the Euclidean distance between P and Q is equal to the HD between A and B. Let us call P and Q as *Hausdorff points* (HP) of A and B. In Fig. 2, P_{01} and Q_{01} are HP of objects numbered 0 and 1 and so on.

Let the object numbered 0 be the one whose NCNs should be found. The first NCN is the object

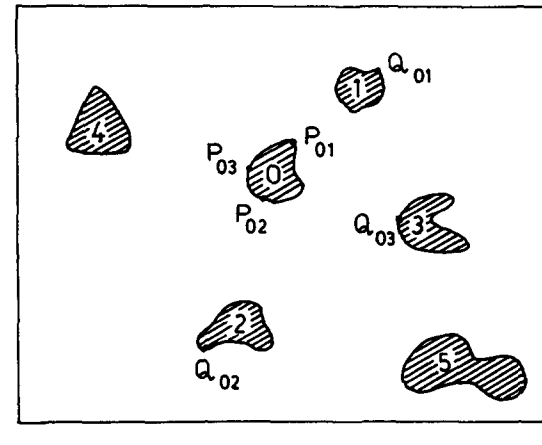

Fig. 2. Nearest centroid neighborhood for objects of finite area.

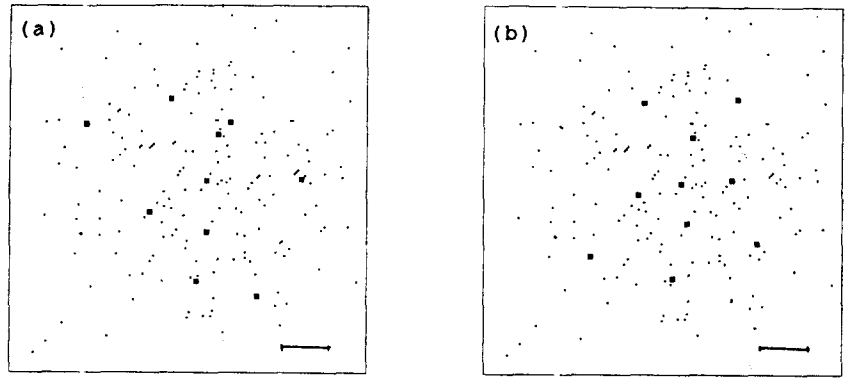

Fig. 3. Representative points using (a) NC neighborhood, (b) Nearest neighborhood. (Line of unit length is shown at bottom right side.)

whose HD from object 0 is the smallest. Thus, the object numbered 1 is chosen. To choose the next NCN, find HD of each object X (other than 1) and object 0 and mark their HP's as P_{0x} and Q_{0x}. Find the centroid of P_{01} and P_{0x}, say C_{0x}. Find the centroid of Q_{01} and Q_{0x} say, C_{1x}. Choose the object for which C_{0x} is nearest to C_{1x} as the next NCN. Thus object 2 is chosen in Fig. 2. The process may be repeated to find any number of NCNs. In general, to find the $(r+1)$th NCN, the centroid of the points $P_{01}, P_{02}, \ldots, P_{0r}$ and P_{0x}, say C_{0rx}, and the centroid of the points $Q_{01}, Q_{02}, \ldots, Q_{0r}$ and Q_{0x}, say C_{rx}, are found and the distance between C_{0rx} and C_{rx} is computed. The object X for which this distance is minimum is marked as the $(r+1)$th NCN.

The nearest median neighborhood for objects can be defined in a similar manner as above if the centroid is replaced by the median of the co-ordinates of the HPs.

3. Some applications

The definition of neighborhood can be applied to a wide variety of problems. To demonstrate the relative advantage of NCN over conventional NN we consider two problems.

The first problem is to choose a representative subset (RS) from a set of data in multi-dimensional space. This problem was recently considered by Chaudhuri (1994) in the form of choosing one out of each k data units. Briefly, the algorithm has the following three steps.

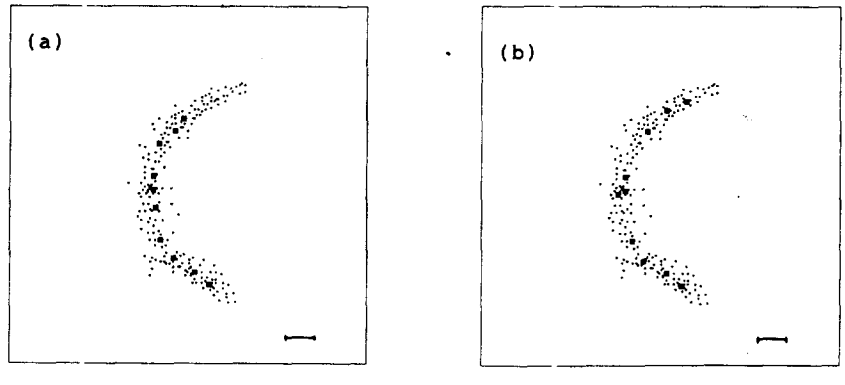

Fig. 4. Representative points using (a) NC neighborhood, (b) Nearest neighborhood. (Line of unit length is shown at bottom right side.)

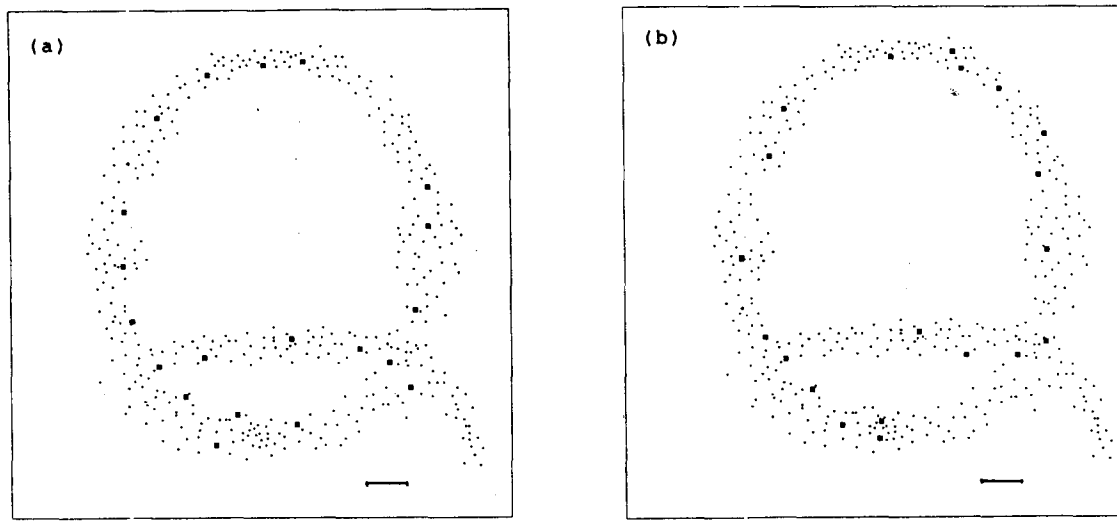

Fig. 5. Representative points using (a) NC neighborhood, (b) Nearest neighborhood. (Line of unit length is shown at bottom right side.)

Algorithm RS

Step 1. (Initialization) Find the density at each datum and order the data in decreasing magnitude of density. Let L be the ordered list. Let $i \leftarrow 1$. Define sets S and S_r where initially $S \Leftarrow S_0$ and $S_r \Leftarrow \emptyset$.

Step 2. Choose the datum P that tops the list L as the ith representative datum. Make $S_r \Leftarrow S_r \cup P$.

Step 3. Count the number of data in the current S. If the number is less than $k-1$ then stop. Else, from the current S find the $k-1$ nearest neighbors of the datum P which has been chosen in Step 2. Delete P and these $k-1$ neighbors from L and S. Make $i \leftarrow i+1$ and go to Step 2.

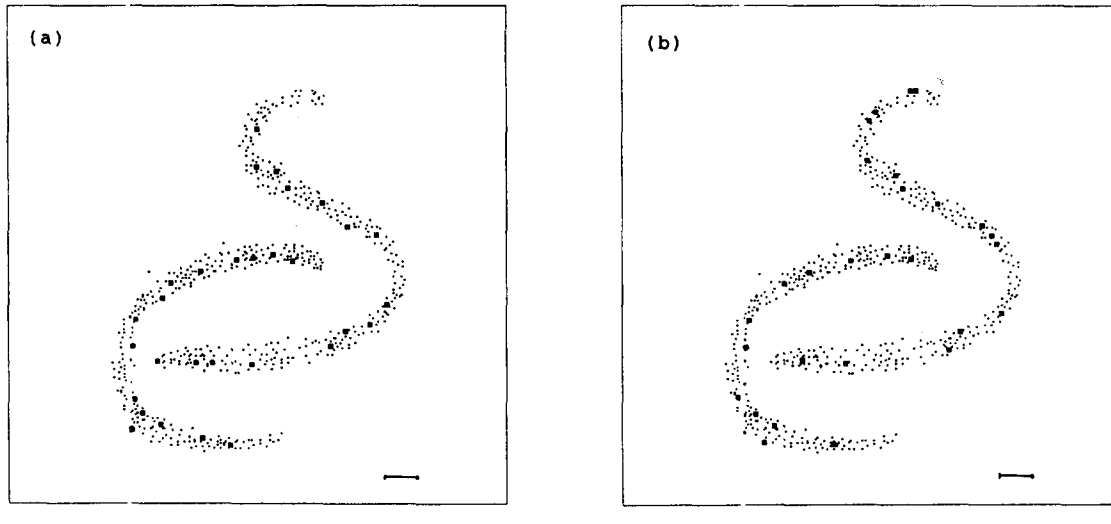

Fig. 6. Representative points using (a) NC neighborhood, (b) Nearest neighborhood. (Line of unit length is shown at bottom right side.)

Table 1

Fig. no.	Representation cost	
	Using NN	Using NCN
3	85.8552	71.0064
4	65.6734	35.2823
5	60.6415	42.7742
6	72.8988	18.4963

S_r is the set of representative points returned by the algorithm. The *density* referred to in Step 1 can be the probability density estimated by the method of kernels or by the K-nearest neighbor approach.

To make a comparative study of NN and NCN we considered a few data sets shown in Figs. 3–6. In Step 3, we computed the neighbors as NNs and NCNs. A cost of representation is defined as follows.

For each datum x in the data set S_0 find the nearest representative point $p_x \in S_r$. The representation cost is given by

$$C = \sum_{x \in S_0} d(x, p_x)$$

where $d(x, p_x)$ is the Euclidean distance between x and p_x.

The cost computed for representatives found by NN and NCN neighborhood is called as C_{nn} and C_{ncn}, respectively.

In the data sets of Figs. 3–6 5% representative points using both NN and NCN were computed. The results are also displayed in Figs. 3–6 where the representative points are shown by small dark squares while the C_{nn} and C_{ncn} are presented in Table 1. Note that C_{ncn} is less than C_{nn} in all cases indicating that NCN always leads to a better choice of representative points.

Given a homogeneous set of points in 2D and 3D space, we have a perceptual notion about the points lying on the border as compared to those of the interior of the data set. A low-level description of the shape of a data set can be made in terms of border points, interior points and stray or noisy points. Labeling of border, interior or stray points may be useful in clustering and related problems. On the shape of point patterns see (Zucker, 1979; Radke, 1988; Zahn, 1971) as well as (Ahuja and Tuceryan, 1989).

Detection of the border points and the interior points of a dot pattern is not a straightforward job. In a previous paper we described an approach where the border points are found on the basis of density (Chaudhuri et al., 1994). But density alone cannot capture the notion of border points. One heuristic is that border points are not surrounded by other points in all directions while the interior points are. The present approach of border point detection is based on this observation.

A point $x \in S_0$ is said to be an *opposite point* of $y \in S_0$ with respect to $z \in S_0$ if x, z and y almost lie in a straight line, i.e., if

$$I(x, y)_z = \frac{d(x, y)}{d(x, z) + d(z, y)} \approx 1$$

where $d(x, y)$ means the Euclidean distance between two points x and y.

Fig. 7 shows that (x, y) are nearly opposite points with respect to z. Note that if x is an opposite point of y then y is also an opposite point of x with respect to z. $I(x, y)_z$ may be called the *degree of oppositeness* of x and y with respect to z.

Consider the k-neighborhood D around z. Let I_z be the average of $I(x, y)_z$; $x, y \in D$. If I_z has a small value then z should be a border point in the neighborhood. Intuitively, we make a threshold at $\frac{1}{2}$. Thus, a point $x \in S$ is said to be *border point* if $I_x < \frac{1}{2}$. A point $x \in S$ is said to be *interior point* if $I_x \geqslant \frac{1}{2}$.

Let m be the desired number of border points. The border point detection algorithm of a data set $S = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^q$ is as follows.

Algorithm BPD

Step 1. Find the value of I_{x_i} for all $i = 1, \ldots, n$.

Step 2. Rearrange the points according to the increasing order of their value of I_x provided $I_x < \frac{1}{2}$.

Step 3. Declare the first m ranking points as m border points, if they exist.

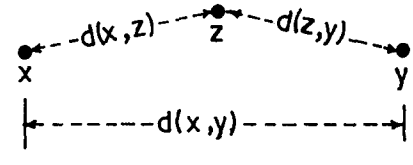

Fig. 7. Degree of oppositeness.

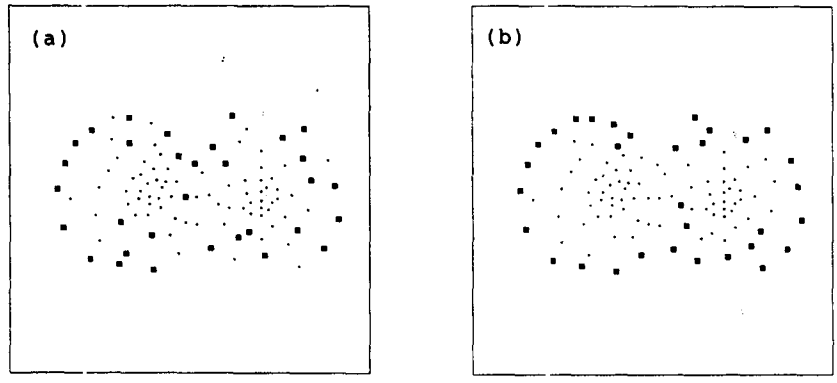

Fig. 8. Border points using (a) NC neighborhood, (b) Nearest neighborhood.

Fig. 8 shows a data set whose border points are marked by small dark squares. Here m is 30% of the total data. The border points defined by NCN are visually better than those defined by NN. This is because NC neighbors are picked all around the candidate point P, and it is convenient to detect if P lies "inside" the data or not.

Detection of border points and hence the shape of a dot pattern and their applications will be described in detail in a separate correspondence.

4. Conclusion

A new definition of neighborhood is proposed that captures the notion of both proximity and symmetric placement. The definition is free from any user-specified parameter and its implementation has the same order of computer complexity as the nearest neighbors. For a point P the proposed NC neighbors have centroid nearest to P, making it an unbiased representative of its neighbors. Two applications are presented to show the efficiency of NC neighbors over the nearest neighbors. It is expected that the proposed neighborhood definition will stimulate further study and show improved performance in many other problems.

Acknowledgment

Help rendered by Mr. Debasis Chaudhuri and Mr. Anirban Roy Chowdhuri in preparing the manuscript is acknowledged with thanks. Useful suggestions given by an unknown referee are sincerely acknowledged.

References

Ahuja, N. and M. Tuceryan (1989). Extraction of early perceptual structures in dot patterns: Integrating regions, boundary and component gestalt. *Computer Vision, Graphics, and Image Processing* 48, 304–356.

Chaudhuri, B.B. (1994). How to choose a representative subset from a set of data in multi-dimensional space. *Pattern Recognition Lett.* 15, 893–899.

Chaudhuri, D., C.A. Murthy and B.B. Chaudhuri (1994). Finding a subset of representative points in a data set. *IEEE Trans. Syst. Man Cybernet.* 24 (9), 1416–1424.

Chazelle, B. (1991). An optimum convex hull algorithm and new results on cutting. *Proc. Foundations of Computer Science* 32, 29–38.

Jarvis, R.A. and E.A. Patrick (1973). Clustering using a similarity measure based on shared near neighbors, *IEEE Trans. Comput.* 22 (11), 1025–1034.

Megiddo, N. (1984). Linear programming in linear time when the dimension is fixed. *J. ACM* 31, 114–127.

O'Callaghan, J.F. (1975). An alternative definition for "neighborhood of a point". *IEEE Trans. Comput.* 24, 1121–1125.

Radke, J.D. (1988). On the shape of a set of points. In: G.T. Toussaint, Ed., *Computational Morphology*. North-Holland, Amsterdam, 105–136.

Toriwaki, J. and S. Yokoi (1988). Voronoi and related neighbours on digitized two-dimensional space with applications to texture analysis. In: G.T. Toussaint, Ed., *Computational Morphology*. North-Holland, Amsterdam, 207–228.

Zahn, C.T. (1971). Graph theoretic methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.* 20, 68–86.

Zucker, S.W. (1979). Towards a low-level description of dot clusters. *Computer Graphics and Image Processing* 9, 213–233.