Pattern Recognition 48 (2015) 918–932

Contents lists available at ScienceDirect

Pattern Recognition

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

A simple statistics-based nearest neighbor cluster detection algorithm

Gerhard X. Ritter [a], José-A. Nieves-Vázquez [a], Gonzalo Urcid [b,*]

[a] CISE Department, University of Florida, Gainesville, FL 32611, USA
[b] Optics Department, INAOE, Tonantzintla, Pue 72000, Mexico

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form
8 August 2014
Accepted 3 October 2014
Available online 29 October 2014

Keywords:
Clusters
Cluster detection
Clustering
Cluster analysis
Digital geometry
Nearest neighbors
Neighborhoods
Statistics
Pattern recognition

ABSTRACT

We propose a new method for autonomously finding clusters in spatial data. The proposed method belongs to the so called nearest neighbor approaches for finding clusters. It is a repetitive technique which produces changing averages and deviations of nearest neighbor distance parameters and results in a final set of clusters. The proposed technique is capable of eliminating background noise, outliers, and detection of clusters with different densities in a given data set. Using a wide variety of data sets, we demonstrate that the proposed cluster seeking algorithm performs at least as well as various other currently popular algorithms and in several cases surpasses them in performance.

1. Introduction

Cluster analysis divides data into groups that are useful for specific applications. These groups are called *clusters* and the data points in a given cluster are in some sense similar. Similarity of data objects may be defined in terms of color, statistics, spectral values, and a host of other features. Some excellent summaries can be found in [22,10,24,27,25], and [39,1] with the last reference devoting six chapters to cluster analysis. These sources demonstrate that there is no single optimal cluster detection algorithm but a plethora of methods for cluster detection. A scanning of current cluster seeking algorithms available in the open literature makes it clear that cluster detection is still an experiment oriented endeavor in the sense that the performance of a given algorithm is not only dependent on the type of data being analyzed, but is also strongly influenced by the chosen measure of pattern similarity as well as the method used for identifying clusters in the data. For example, suppose we have a set of objects $\mathcal{O}$ specified by a sequence $p_1, \ldots, p_n$ of properties or attributes such as a specific color, shape, and weight range with different objects lacking different properties. Such a set is often transformed into a set of binary vectors $X \subset \mathbb{R}^n$, where $\mathbf{x} = (x_1, \ldots, x_n) \in X$ is defined by $x_i = 1$, if and only if, p_i holds else $x_i = 0$. In this situation, two objects $\mathbf{x}, \mathbf{y} \in X$ are viewed as *similar* if they share a large majority of properties or attributes. Suppose X contains the elements $\mathbf{w}, \mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ given by: $w_i = 1$ if $i = 1$ else $w_i = 0$, $x_i = 1$ if $i = n$ else $x_i = 0$, $y_i = 0$ if $i = 1$ else $y_i = 1$, and $z_i = 0$ if $i = n$ else $z_i = 1$. Employing the Euclidean metric, one obtains $d(\mathbf{w}, \mathbf{y}) = n$, which can be very large in some settings. This shows that $\mathbf{w}$ and $\mathbf{y}$ are spatially far apart when viewed as points in a n-dimensional Euclidean space. This can also be interpreted that the two vectors are very dissimilar as they have no common attributes. However, we also have $d(\mathbf{w}, \mathbf{x}) = \sqrt{2} = d(\mathbf{y}, \mathbf{z})$ even though $\mathbf{w}$ and $\mathbf{x}$ share no attributes and are, therefore, totally dissimilar while $\mathbf{y}$ and $\mathbf{z}$ are very similar. Thus, the Euclidean metric provides little information when used as a clustering tool for this type of data. Likewise, the L_∞ metric is of little use in cluster analysis of binary data since $d(\mathbf{x}, \mathbf{y}) = \bigvee_{i=1}^{n} |x_i - y_i| = 1$ for all distinct pairs $\mathbf{x}, \mathbf{y} \in X$, where X is an n-dimensional binary data set and $\bigvee$ denotes the maximum.

It is pertinent to note that some researchers test their clustering algorithms on well known data sets that are commonly used in machine learning and training of artificial neural networks for pattern classification which, although related, differs from the subject of cluster detection and cluster analysis. A typical example is the 4-dimensional Iris data set consisting of three classes, with each class corresponding to a distinct species of the genus Iris [17,10]. Four features, specific to a given species, are described in vector format. Two of the classes are geometrically closely intertwined in 4-space and can be successfully separated with neural network techniques when using the complete data set as training data but fails when

* Corresponding author. Tel.: +52 222 266 3100; fax: +52 222 247 2940.
E-mail addresses: ritter@cise.ufl.edu (G.X. Ritter), gurcid@inaoep.mx (G. Urcid).

http://dx.doi.org/10.1016/j.patcog.2014.10.009
0031-3203/

using 50% and even 60% of the data for training [34]. The problem is that the two intertwined sets do not form two well defined spatial clusters that can be determined using current clustering techniques. For this reason we do not consider many of the standard data sets that are commonly used in pattern classification tasks for evaluating performance of cluster seeking algorithms.

In our approach we view clusters in terms of their spatial arrangement and distribution by using the old adage that "birds of a feather will flock together." For instance, when observing migrating cranes one sees beautiful V shaped formations, while blackbirds will flock into cloud shaped 3-dimensional globular clusters. Several migrating species of birds form huge rotating 3D doughnut or spiral shaped clusters before assuming a single line or a V shaped formation. In his seminal paper entitled *Data Clustering*: *50 years beyond K-means*, A.K. Jain points out that there are no cluster algorithms available that are able to detect all seven clusters shown in Fig. 1 even though these clusters are readily apparent to a human data analyst [25]. The problems raised by Jain's example are manifold. First, there is the issue of the noisy background that is interspersed with the data clusters. Next, the two globular clusters on the left side of the figure have different densities. Finally, the well defined geometric pattern clusters on the right side have cluster center problems. The circular clusters share the same geometric cluster center, while the center for one of the two spiral cluster may be located inside or closer to the other spiral. These clusters are troublesome for various center based approaches to clustering.

The basic idea underlying center based approaches is to group a set $X \subset \mathbb{R}^n$ of feature vectors into K clusters using an appropriate similarity measure for comparison with the cluster's center. Generally, this measure is the distance between the feature vector and the cluster's center and assigns the feature vector $\mathbf{x}^j$ to cluster C_k whenever the distance from $\mathbf{x}^j$ to the cluster's center $\mathbf{c}^k$ is the minimum over all K clusters. The k-means (hard c-means) clustering algorithm, first developed by MacQueen [29], belongs to this group. The algorithm was later modified by Dunn and Bezdek [11,5,6] to include fuzzy c-means clustering and has become one of the most popular and widely used clustering method. Since the number of actual clusters in high dimensional data is generally not known, the initial input value K can critically affect the algorithm's output. Similarly, different initial centroid values usually result in different output and performance. Consequently, various modifications of c-means algorithms have been proposed in order to get around some of these problems [28,43,9,44,33,45,41,47]. The performance of these modifications always improved on the examples given by their authors but still failed when applied to Jain's example as well as other data sets some of which are given in subsequent sections.

Jain's example will yield very mixed results for many clustering algorithms in vogue today. However, it is our opinion that any automatic clustering algorithm worth its salt should be able to find the three clusters shown in Fig. 2.

Here the data set X consists of 192 points; i.e., $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^{192}\} \subset \mathbb{R}^2$. The statistics associated with X are trivial. Every point $\mathbf{x}^j \in X$ has a neighboring point whose distance from $\mathbf{x}^j$ is of unit length. This is true for the Euclidean as well as the chessboard and the city-block distance metric. More specifically, for $j = 1, \ldots, 192$, the number $\tau_j = \bigwedge_{k=1, k\neq j}^{192} d(\mathbf{x}^j, \mathbf{x}^k) = 1$, where d denotes any of the three distances mentioned and $\bigwedge$ denotes the global minimum. Hence, the average nearest neighbor distance and the standard deviation of the nearest neighbor distances are given by $\mu = \sum_{j=1}^{192} \tau_j / 192 = 1$ and $\sigma^2 = \sum_{j=1}^{192} (\tau_j - \mu)^2 / 192 = 0$, respectively.

Nevertheless, when applying either the c-means or the fuzzy c-means algorithm in Matlab and specifying $K=3$, one may not obtain the 3 correct clusters as shown in Fig. 3 unless one provides the *actual* clusters. Even when using the correct number K and selecting randomly each starting point or *seed* in each of the actual clusters may still result in incorrect identification of the true clusters. This happens because cluster detection in data containing clusters of greatly varying sizes and densities remains problematic when applying the various modified c-means techniques cited earlier. The reason for this is that the fundamental building blocks of these modifications are all based on the classical c-means and fuzzy c-means methodology and as such inherit some of the undesirable properties of their predecessors.

For example, using the alternative c-means clustering algorithm proposed by Wu and Yang [43] produces the results shown in Fig. 3(b). Here the fuzzy membership threshold used was set at 0.4; i.e., data points below 0.4 where not assigned to any cluster. One can assign all points to clusters by simply assigning cluster membership to a cluster if the vector's fuzzy membership is less with respect to the other two clusters. In this case the result is shown in Fig. 3(c). The method of randomly assigning data points to three sets of equal size and defining the seeds to be the center points of these sets may result in the three clusters shown in Fig. 3(d). Given the inherent difficulties encountered by the c-means method listed here and elsewhere [4,32], we considered several cluster seeking methods that were not based on the c-means paradigm.

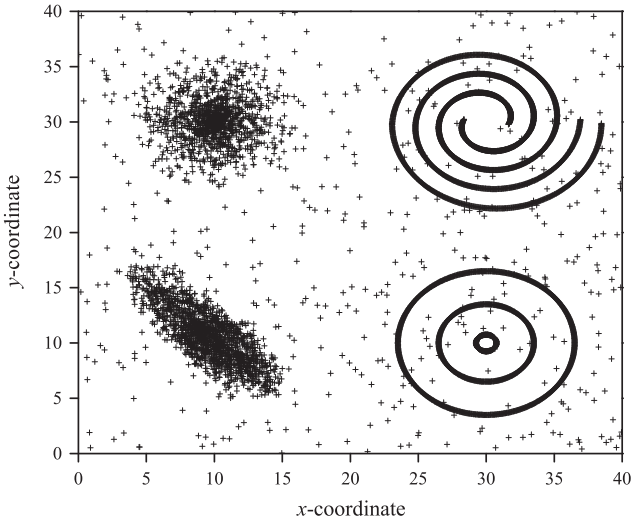

Fig. 1. Seven clusters that differ in shape, size, and density in a noisy background.

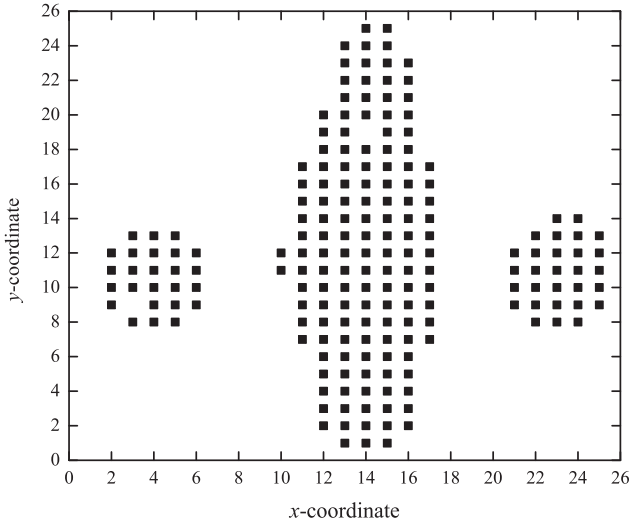

Fig. 2. Three separated globular clusters differing only in size.

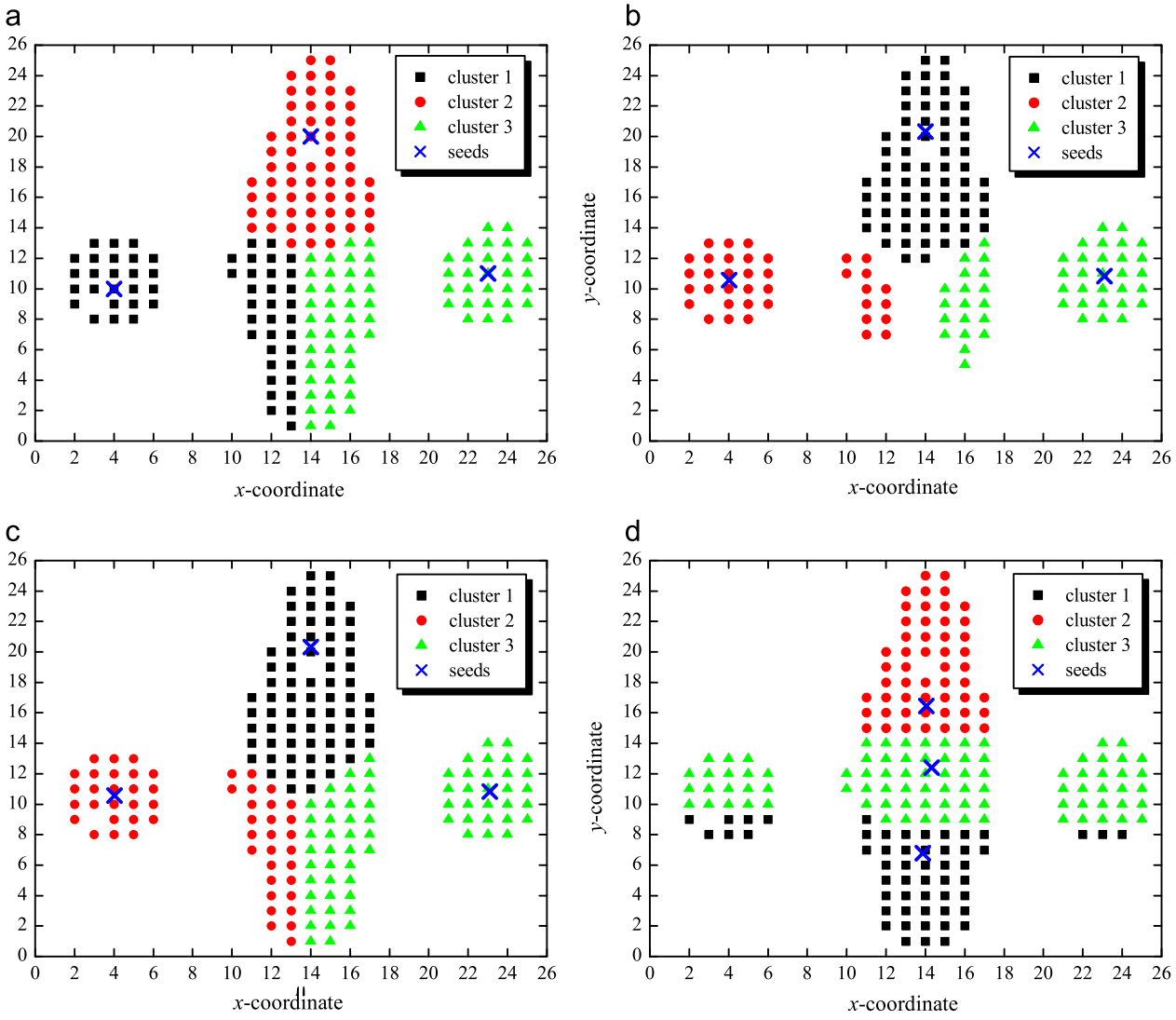

Fig. 3. (a) 3 clusters found by c-means with initial cluster centers as indicated (seeds); (b) 3 clusters found using the alternative fuzzy c-means (fuzzy threshold $=0.4$); (c) same as (b) but with variable threshold, and (d) alternative fuzzy c-means with seeds the centroids of 3 clusters derived by dividing the data into 3 equal parts.

Among these were nearest neighbor algorithms, cluster trees and density-based methods, model and rule-based approaches, and path-based spectral techniques [26,42,14,18,16,8,38,30]. Although several of these methods exhibited various advantages over c-means-based algorithms, they all failed in correctly identifying the seven clusters with noisy background shown in Fig. 1. We briefly summarize some of the reasons for this failure.

Spectral clustering makes use of the spectrum, specifically the eigenvalues of the similarity matrix $S=\{s_{ij}\}$ obtained from the data [31,46]. That is, if X denotes the data set and s the similarity measure, then $s_{ij}=s(\mathbf{x}^i,\mathbf{x}^j)$, where $\mathbf{x}^i,\mathbf{x}^j\in X$. The Shi–Malik algorithm for image segmentation is based on this concept [36]. Image segmentation has been the major application domain for spectral based algorithms. As such, spectral clustering performed well in identifying the three circular clusters in Fig. 1. However, even without noise, spectral clustering had problems in correctly identifying the two spiral clusters.

Path-based clustering measures local homogeneity rather than global similarity of data objects such as pixels in a digital image. It is capable of identifying the two spirals and homogeneous texture patches in digital images. Hence, like spectral clustering, it has proven to be a useful tool in image segmentation [16,15]. Nevertheless, non-homogeneous clusters such as clusters with varying densities remain problematic. Combining path-based clustering with spectral clustering provides a new paradigm known as path-based spectral clustering [8]. This combination is also mostly applied to image segmentation and is able to segment the spiral and circular clusters in Fig. 1. Its user defined neighborhood choice dependency makes it very sensitive to noise as well as a poor candidate for non-homogeneous (e.g. varying density) cluster identification. Additionally, the computational complexity of path-based spectral clustering is fairly high.

Clustering based on shared nearest neighbors (SNNs) using a distance function are appealing due to their intuitive simplicity. Algorithms based on SNNs require a user's choice of a metric, a number k of nearest neighbors, and a distance threshold that is modifiable. The idea of SNNs was introduced by Jarvis and Patrick [26]. They construct a SNN graph from a proximity matrix by creating a link between a pair of points $\mathbf{p}$ and $\mathbf{q}$ if and only if $\mathbf{p}$ and $\mathbf{q}$ have each other in their closest k nearest neighbor list. Several variations and generalizations of the SNN approach have been published since its first introduction [42,14,18,38]. One of the most often cited variants is the DBSCAN shared nearest neighbor method [14] (*Density-Based Spatial Clustering for Applications with Noise*). The DBSCAN algorithm – as well as the proposed *Simple Statistics-based Nearest Neighbor* (SSNN) algorithm described in the next section – correctly identifies the 3 globular clusters shown in

Figs. 2 and 4(b). However, DBSCAN needs two user specified parameters: the ϵ parameter which represents an n-dimensional ball of radius ϵ centered at a data point, commonly referred to as the ϵ-neighborhood of that point and the minimal number k of data points in an ϵ-neighborhood of a data point $\mathbf{p}$ that serves as the basis for deciding the cluster membership of $\mathbf{p}$. As a consequence, DBSCAN performs poorly in correctly identifying clusters that have large variation in densities. To increase performance and overcome some of these problems, several generalizations of DBS CAN have been proposed [35,2,7].

A more successful generalization of DBSCAN is the OPTICS approach [2,7] (*Ordering Points To Identify the Clustering Structure*). However, the basic approach is similar to DBSCAN in that the two parameters ϵ and k are required even though ϵ plays a less important role as each point is also assigned a *core distance*, which is the distance to its k nearest neighbors. The number $k=4$ is often encountered and is probably due to a desire of reducing computational overhead. However, $k=4$ is generally not a good choice for high-dimensional data sets. Once the user supplied choice of k is fixed and ϵ is chosen too large, then using the core distance may create fewer clusters than are actually present in the data (see Example 1 in Section 2).

2. SSNN clustering algorithm

In this section we discuss the SSNN approach to clustering and provide some performance comparisons with several standard algorithms cited in the Introduction. The new algorithm is a consequence of our research efforts in autonomous endmember detection for hyperspectral image segmentation [40]. In this research we were in need of a fairly robust, yet simple, autonomous cluster detection algorithm in somewhat noisy data. None of the standard algorithms that we considered provided satisfactory results on real data. Nearest neighbor approaches were the most attractive because of their intuitive simplicity. The need for some statistical information about point distances is also crucial as such information can provide important insight into the spatial arrangement of the data points. Two fundamental as well as simple measures are the mean and the standard deviation of distances between points and their nearest neighbors. Knowledge of these statistical measures can be used to provide thresholds for noise or outlier removal and also cluster detection. Suppose $X=\{\mathbf{x}^\xi\in\mathbb{R}^n:\xi=1,\ldots,k\}$ denotes the data set under consideration and $T=\{\tau_j:\tau_j=\bigwedge_{\xi\neq j} d(\mathbf{x}^j,\mathbf{x}^\xi),\mathbf{x}^\xi\in X\}$ denotes the set of nearest neighbor distances associated with X. Our basic measuring stick for creating nearest neighborhoods for clustering data is the sum $\mu+\sigma$ of the average and the standard deviation of nearest neighbor distances given by T. Intuitively, one would like to obtain parameters a and b for which $\epsilon=a\mu+b\sigma$ yields an ideal clustering measure of nearest neighbors. Such values need to be sensitive to the maximal nearest neighbor distance $\tau=\bigvee_{j=1}^k\tau_j$ in order to isolate outliers or noise. Although such values a and b can be obtained for simple data with well defined clusters, there is no known method for computing them for arbitrary data sets. In our approach we used the relations $\alpha(\mu+\sigma)=\tau$ and $\beta(\mu+\sigma)=\tau+\sigma$ in order to obtain the sensitivity parameters α and β used in the formulation of the various ϵ values defined in the SSNN algorithm. The different ϵ values are data dependent and a consequence of various possible relationships between the statistical parameters μ, σ, $\mu+\sigma$, τ, and the ratios $\alpha=\tau/(\mu+\sigma)$ and $\beta=\alpha+\sigma/(\mu+\sigma)$. Another value that plays a more indirect role in the relationship between μ and σ is the minimum closest neighbor distance $\tau_{\min}=\bigwedge_{j=1}^k\tau_j$ since $\tau_{\min}\leq\mu\leq\tau$. These relationships are given by the following theorems:

Theorem 1. $\sqrt{\mu^2+\sigma^2}\leq\tau$.

Theorem 2. $\alpha>2/3$, $\beta\geq1$, *and* $0\leq\beta-\alpha<1$.

Theorem 3. $\sigma\leq\mu\Leftrightarrow\beta-\alpha\leq1/2$ *and* $\sigma>\mu\Leftrightarrow\beta-\alpha>1/2$.

Theorem 4. *If* $\alpha\geq1$ *and* $\beta\leq3/2$, *then* $\sigma\leq\mu$.

Note that the lower bound 2/3 of α is the multiplicative inverse of the decision boundary $\beta=3/2$. For the next theorem let $T_{\max}=\{\tau_j\in T:\tau_j=\tau\}$, $T_{\min}=\{\tau_j\in T:\tau_j=\tau_{\min}\}$, $m=|T_{\max}|$, and $\ell=|T_{\min}|$, where $|A|$ denotes the number of elements (cardinality) of set A.

Theorem 5. $\tau-\mu\geq\ell(\tau-\tau_{\min})/k$ *and* $\mu-\tau_{\min}\geq m(\tau-\tau_{\min})/k$.

Proofs and further discussions of the theorems are given in the Appendix. Henceforth, we let $X=\{\mathbf{x}^\xi\in\mathbb{R}^n:\xi=1,\ldots,k\}$ denote the data set under consideration and given sets A and B, $A\setminus B$ denotes set subtraction. The symbols $\approx$ and $\ll$ stand for *approximately equal to* and *much less than*, respectively. The SSNN algorithm consists of two parts. In the first part we apply two neighborhood based filters in order to cluster data and remove noise, outliers and other artifacts. Each filter is determined by a specific fixed number ϵ derived from the nearest neighbor statistics computed in the first step (S1) of the algorithm. These statistics are derived from the collection T of nearest neighbor distances. The first filter provides for a rough listing of possible clusters and sets up a remainder set R_1 of possible outliers, noise, or other artifacts of the data. However, if $R_1=\varnothing$, then the algorithm stops. The output consists of the clusters detected. If $R_1\neq\varnothing$, then Part I of the algorithm repeats the whole procedure one more time using the new set given by $X'=X\setminus R_1$ as the data set under consideration. The ϵ derived from the NN-statistics of X' provides for a refined filter used to detect its clusters. The final output is a set of clusters and a possible new remainder set R_2. The remainder sets R_1 and R_2 are obtained by setting a threshold $M\geq1$ which accumulates all clusters C_ℓ into the remainder set for which $|C_\ell|\leq M$. The set of all clusters C_ℓ for which $|C_\ell|>M$ and the set $R=R_1\cup R_2$ constitute the input to Part II of the algorithm. The value of the threshold M as well as the second part will be discussed after specifying the four major steps that constitute Part I of the algorithm.

SSNN Algorithm-Part I
Initialize $t=0$, $p=0$, $X_0=X$, $T=\varnothing$
S1 **let** $p=p+1$, $\tau=0$ [Compute statistical parameters]
for $\mathbf{x}^j\in X$
$\tau_j=\bigwedge_{\mathbf{x}^\xi\in X,\xi\neq j}d(\mathbf{x}^j,\mathbf{x}^\xi)$
if $p=1$, $T=T\cup\{\tau_j\}$ **else** continue
$\tau=\tau\vee\tau_j$
let $\mu=(1/|X|)\sum_{\mathbf{x}^j\in X}\tau_j$, $\sigma^2=(1/|X|)\sum_{\mathbf{x}^j\in X}(\tau_j-\mu)^2$,
$\alpha=\tau/(\mu+\sigma)$, $\beta=\alpha+\sigma/(\mu+\sigma)$
if $p=1$ **case** $\beta>3/2$, $\epsilon=\alpha\mu+\sigma$
case $\beta\leq3/2$, $\epsilon=\beta(\mu+\sigma)$
else **case** $\beta>3/2$ and $\sigma\geq\mu$, $\epsilon=\beta\mu+\sigma$
case $\beta>3/2$ and $\mu>\sigma$, $\epsilon=\mu+\beta\sigma$
case $\beta\leq3/2$, $\epsilon=\beta(\mu+\sigma)$
S2 [Initialize intermediate sets for the clustering process]
let $t=t+1$, $C_t=N_t=\varnothing$, $X_t=X_{t-1}$,
randomly choose $\mathbf{x}\in X_t$, $B_t=\{\mathbf{x}\}$
S3 [Update intermediate sets and stopping criteria]
let $C_t=C_t\cup B_t$, $X_t=X_t\setminus C_t$
if $X_t=\varnothing$ **call** Histogram to find M, $R_p=\bigcup_{|C_j|\leq M}C_j$
else continue with next step S4
if $p=1$ and $R_p=\varnothing$, STOP [Clusters are given by C_j for $j=1,\ldots,t$]
else continue
if $p=1$ and $R_p\neq\varnothing$, **let** $X=X\setminus R_p$, $t=0$, $X_0=X$,
RETURN to step S1
else let $R=R_1\cup R_2$, STOP
[Part I of the SSNN is completed. All clusters C_j with $|C_j|>M$

are the clusters determined by Part I, R is the remainder set]
S4 **for** $\mathbf{x}^j \in B_t$ [Cluster detection loop]
for $\mathbf{x}^\xi \in X_t$
if $d(\mathbf{x}^j, \mathbf{x}^\xi) \le \epsilon$, $N_t = N_t \cup \{\mathbf{x}^\xi\}$ **else** continue
if $N_t = \varnothing$ RETURN to step S2
else $B_t = N_t$, RETURN to step S3

Before specifying the second part of the SSNN algorithm we believe that it is instructive to discuss some of the parameters generated and provide a few examples in order to better understand the computational aspects and performance of Part I. Observe that the algorithm does not specify a particular metric d. This allows the user to implement his favorite metric. Unless otherwise specified we used the L_∞ metric in the examples given in this paper. This choice was mainly due to the computational simplicity of this metric. The basic statistical parameters μ, σ, and $\tau = \bigvee \tau_j$ are derived from the set T of nearest neighbor distances. Thus, the first objective of step S1 is the derivation of the elements of T. Since T is referenced twice in Part I ($p=1,2$) and again in Part II, it is advantageous to use T as a look-up table in order to reduce the computation time in all steps that require its elements.

The parameters μ, σ, and τ are used to compute the sensitivity parameters α and β. These five parameters determine the ϵ-radius for the nearest neighbor clustering in step S4. Note that all choices are dependent on whether $\beta \ge 3/2$ or $\beta < 3/2$. The decision thres hold 3/2 for determining ϵ is a consequence of the theorems that establish the relationships between μ, σ, α, and τ. Since $\beta(\mu+\sigma) = \tau+\sigma$, β provides a measure as to the difference between the value $\tau+\sigma$ vs $\mu+\sigma$. Thus, $\beta > 3/2$ indicates a large difference between $\mu+\sigma$ and τ. Hence, we can use a radius slightly larger than $\mu+\sigma$ but less than τ by multiplying μ by α in order to isolate outliers and noise, i.e., *single point clusters*. This follows from the observation that for $\beta > 3/2$ we have $\alpha \ge 1$. Therefore, $\tau \ge \alpha(\mu+\sigma) > \alpha\mu+\sigma$. Similarly, a small value of $\beta \le 3/2$ means an even smaller value of α. That is $\alpha \approx 1$ and $\tau \approx \mu+\sigma$ with σ very small when compared to μ. In this case, we want ϵ to be slightly larger than τ to not separate clusters of sets that are very close (within a distance only slightly larger than τ) but whose elements have nearest neighbors a distance equal to or close to τ. Some of the examples provided demonstrate the advantage of this approach. For this reason we chose $\epsilon = \beta(\mu+\sigma) = \tau+\sigma$. The ϵ values for $p=2$ and the new set $X \setminus R_1$ are chosen for analogous reasons.

Another threshold that is in need of elucidation is the value M. Step S4 of the SSNN algorithm computes all possible clusters of points that have nearest neighbors within a distance ϵ, including single point clusters. Setting $M=1$ suffices to eliminate all single point clusters. However, for large data sets, especially those containing extraneous artifacts (e.g. noise) as illustrated in Fig. 1, it often happens that there is a large abundance of single clusters as well as a large number of small clusters consisting of 2, 3, and more points. Since such data sets are often encountered when dealing with real data, it is befitting to establish a histogram once $X_t = \varnothing$ in step S3. When this occurs, the algorithm calls the following function:

Histogram
let $m = \bigvee_{\ell=1}^{t} |C_\ell|$ and $h_i = 0$ for $i = 1,\ldots,m$
for $i = 1$ to m
for $\ell = 1$ to t
if $i = |C_\ell|$, $h_i = h_i + 1$ **else** continue

For small data sets consisting of fewer than 50 points the number h_1 is generally either zero or fairly small. However, for large data sets a large h_1 is a real possibility and usually implies that the data is very noisy or clusters are not well defined. Additionally, in case of extremely large amounts of random noise there most likely will be some noise points that are closer together (within the distance ϵ of each other). As a result, the histogram exhibits an initial rapidly decreasing sequence $h_1, \ldots, h_M > 0$ with most $h_i \ne 0$ for $1 \le i \le M$. This sequence is then followed by a significantly larger sequence consisting entirely of zeros; that is, $h_{M+1} = 0 = \cdots = 0 = h_{j-1} < h_j$. The number $(j-M)$ is called the *zero gap*. The beginning of the zero gap corresponds to the number M used for determining the sets R_p. If there does not exist an initial decreasing sequence h_i, that is $h_i = 0$ for

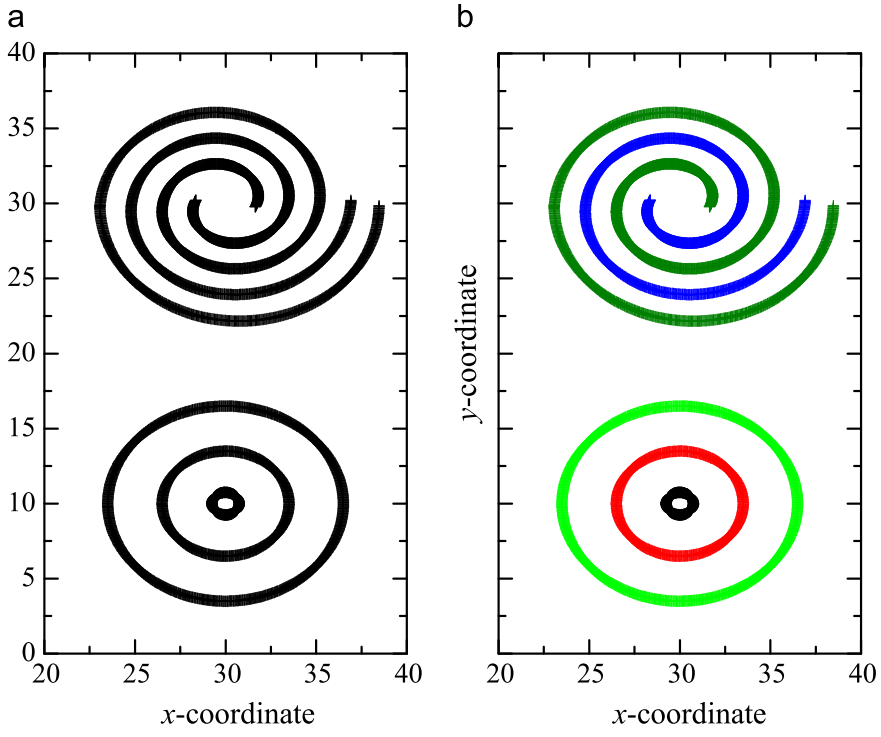

Fig. 4. (a) The raw data set with five well defined clusters; (b) correct identification of the 5 clusters using only Part I of algorithm SSNN.

$i=1,\ldots,\ell$, with $\ell \geq 2$, then we set $M=1$ as we do not consider single point clusters as viable clusters. On the other hand, larger values of M can result in the elimination of clusters consisting of five or more points which may not be desirable. This is one reason for reexamining the background R in Part II of the algorithm. The next examples will illustrate some of these issues.

Example 1. For data sets consisting only of geometrically well defined clusters without noise such as shown in Figs. 2 and 4(a), the algorithm has no problems in correctly identifying all clusters and terminates in Part I. Specifically, for the case presented in Fig. 2 we have $\mu=1$, $\beta=1<1.5$ and $\sigma=0$ so that $\epsilon=\beta(\mu+\sigma)=\tau=1$. The corresponding histogram gives $h_i=0$ for $i=1,\ldots,24$ and $h_{25}=h_{29}=h_{138}=1$. Thus, there is a large zero gap before $h_i\neq 0$ and we set $M=1$. With this value of M the algorithm stops in the $p=1$ cycle since $R_1=\varnothing$. Similarly, for the data set shown in Fig. 4(a), we obtain $\beta=1.452<1.5$ so that $\epsilon=\tau+\sigma=0.0485+0.0102=0.0587$. The histogram gives $h_i=0$ for $i=1,\ldots,298$, with h_{299}, h_{599}, h_{999}, h_{1400}, and h_{1900} equal to 1. Thus, the gap of zeros starting from h_1 is extremely large and, hence, $M=1$ and $R_1=\varnothing$. Again, the algorithm terminates in Part I. Although this example of spirals and circles within circles has a straight forward solution, it has been used by numerous researchers to show that various cluster algorithms have failed when confronted by this data [8,25,32].

Example 2. For a different example, we consider the heterogeneous data set shown in Fig. 5(a). This data set first appeared in [42] as well as in Example 3 of [43] in slightly altered format due to the addition of more points to both clusters, thus making the clusters appear more homogeneous. For $p=1$, one obtains $\mu=0.676$, $\sigma=0.305$, $\alpha=1.53$, and $\beta=1.841>1.5$ so that $\epsilon=\alpha\mu+\sigma=1.339$. The resulting histogram yields $h_1=2$, h_2 to $h_{30}=0$ and $h_{31}=1$. Again, the zero gap between clusters containing a single point and the only other cluster is large, resulting in $M=1$. The resulting cluster and remainder set R_1 are shown in Fig. 5(b). Since $R_1\neq\varnothing$, $p=2$; for this value of p and new set $X\setminus R_1$ one obtains $\sigma<\mu$ and $\beta=1.779$. Therefore, $\epsilon=\mu+\beta\sigma=1.052$ and two clusters C_1, C_2 are obtained containing 14 and 16 points,

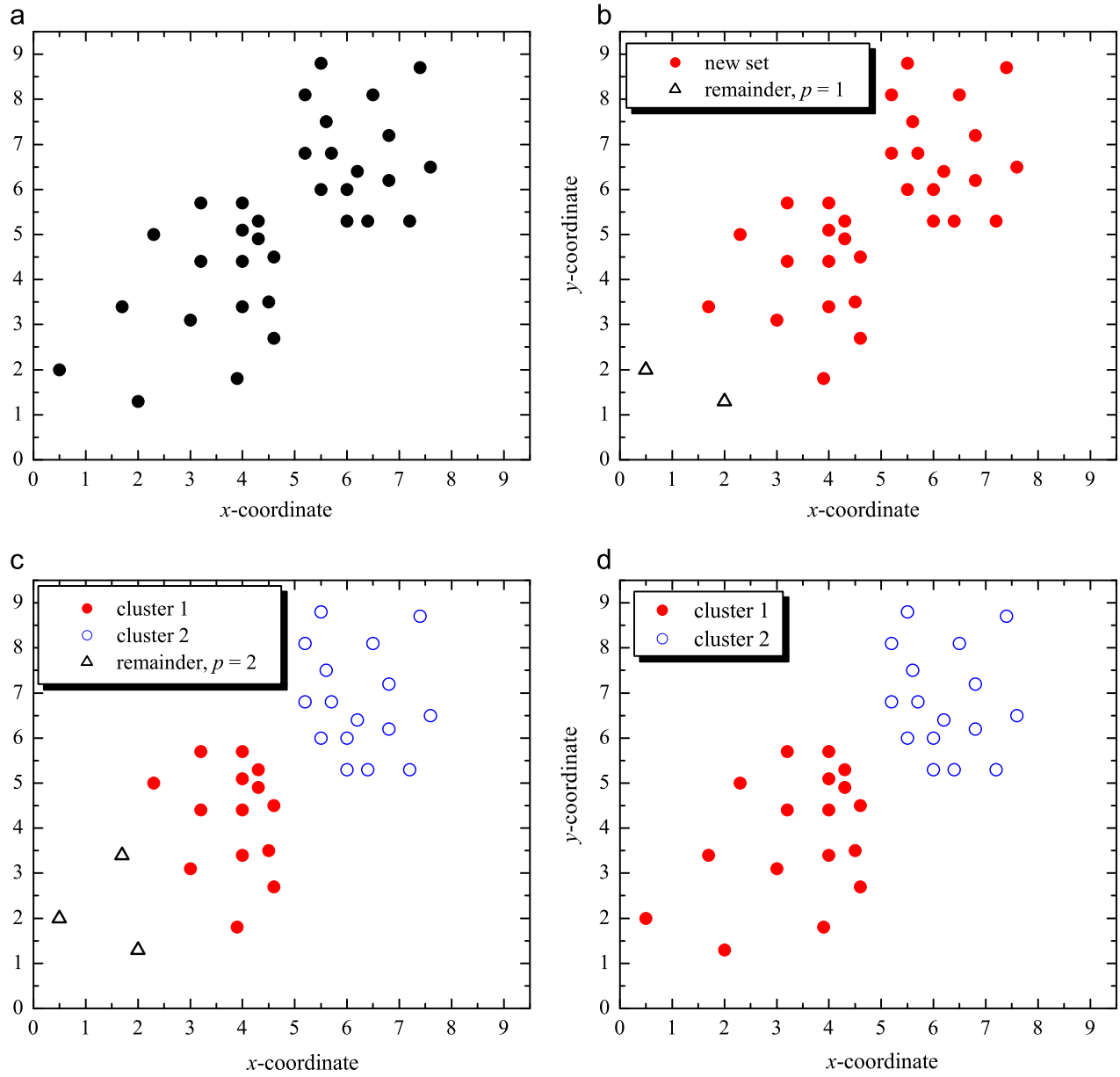

Fig. 5. (a) Heterogeneous data set X with $|X|=33$; (b) for $p=1$ and $M=1$ the result is cluster C_1 having 31 points and $|R_1|=2$; (c) for $p=2$ and $M=1$ two viable clusters are detected and $|R|=3$; (d) final 2 clusters detected by algorithm SSNN-Part II.

respectively, as well as a one point cluster (see Fig. 5(c)). Hence, $M=1$ and $|R|=3$. Since $R\neq\varnothing$, the algorithm continues with Part II.

The input to Part II of the SSNN algorithm consists of the remainder set R and the clusters C_ℓ for which $|C_\ell|>M$. We assume that these clusters have been relabeled sequentially as $C_1,\ldots,C_L$. The first step of Part II of the algorithm computes the statistical parameters τ, μ, σ, α, β, and the appropriate ϵ for each cluster C_ℓ. The objective is to absorb remainder points within the distance ϵ of C_ℓ into C_ℓ. In this scheme we have to provide a distance ϵ that is slightly larger than the maximal nearest neighbor value τ of the set C_ℓ. Note that the algorithm terminates when either $R=\varnothing$ or the ℓ-loop ends. Example 2 and the examples of Section 3 illustrate these two scenarios.

SSNN Algorithm-Part II

S1 Initialize $\tau=0$

for $\ell=1$ to L [Compute statistical parameters of cluster C_ℓ]

for $\mathbf{x}^j\in C_\ell$

$\tau_j=\bigwedge_{\mathbf{x}^\xi\in C_\ell,\xi\neq j} d(\mathbf{x}^j,\mathbf{x}^\xi)$, $\tau=\tau\vee\tau_j$

let $\mu=(1/|C_\ell|)\sum_{\mathbf{x}^j\in C_\ell}\tau_j$, $\sigma^2=(1/|C_\ell|)\sum_{\mathbf{x}^j\in C_\ell}(\tau_j-\mu)^2$,

$\alpha=\tau/(\mu+\sigma)$, $\beta=\alpha+\sigma(\mu+\sigma)$,

case $\beta>3/2$ and $\sigma>\mu$, $\epsilon=\tau+\mu$,

case $\beta>3/2$ and $\sigma<\mu$, $\epsilon=\tau+\sigma$,

case $\beta\leq 3/2$, $\epsilon=\alpha\tau+\beta\sigma$,

case $\alpha\leq 3/2$ and $\beta>3/2$, $\epsilon=\alpha\tau+\sigma$

S2 **for** $\mathbf{x}^j\in C_\ell$ [Merge appropriate background points with C_ℓ]

for $\mathbf{x}^\xi\in R$

if $d(\mathbf{x}^j,\mathbf{x}^\xi)\leq\epsilon$, $C_\ell=C_\ell\cup\{\mathbf{x}^\xi\}$, $R=R\setminus\{\mathbf{x}^\xi\}$

if $R=\varnothing$, Stop [Clusters are C_ℓ for $\ell=1,\ldots,L$] **else** continue

Example 2 (*continued*). As mentioned, the input to Part II are the two clusters C_1, C_2 and the remainder set R obtained from Part I using the heterogeneous data set of Fig. 5(a). For $\ell=1$ one obtains $\tau=1$, $\mu=0.593$, $\sigma=0.24$, $\alpha=1.2$, and $\beta=1.489<1.5$. Thus, $\epsilon=\alpha\tau+\beta\sigma=1.558$. With this ϵ and $\ell=1$, three background points are merged into C_1 so that $R=\varnothing$ and the algorithm terminates. As shown in [43], the c-means, fuzzy c-means, and the alternative hard c-means were not able to identify these two clusters correctly. However, the alternative fuzzy c-means did correctly identify both clusters.

3. SSNN clustering performance

In many scenarios no a priori knowledge exists about the type or number of clusters in a given data set. For most high-dimensional data sets the number of clusters can be extremely difficult if not impossible to ascertain before the start of a clustering procedure. However, any good clustering method should be able to autonomously find clusters if their topology is reasonably well defined. In this section we examine the performance of the SSNN algorithm by using several different data sets.

Example 3. In this example we consider the data set shown in Fig. 6(a). This data set consists of 31 points and appeared in [43] where it was shown that the c-means and fuzzy c-means failed to correctly identify the two different size clusters. Both, the alternative fuzzy c-means and the SSNN algorithm had no problems in correctly identifying the two clusters. The various stages of our algorithm are illustrated in Fig. 6. This experiment was repeated after modifying the data set by adding two outlier points to the data as shown in Fig. 7 that get eliminated in Part I while $p=1$.

Example 4. The data set shown in Fig. 8(a) appeared in [13] and consists of 30 points. Visual inspection shows three disjoint clusters and two outliers near the bottom right side of the square. Part I of the SSNN algorithm with $p=1$ yields the values shown in the 1st row of Table 1 and $\epsilon=\alpha\mu+\sigma$ provides for 3 clusters with $h_1=2$, h_2 to $h_{27}=0$ and $h_{28}=1$. Thus, $M=1$ and $|R_1|=2$. For $p=2$, the new set $X\setminus R_1$ yields the statistics shown in the 2nd row of the same table. Since $\beta>1.5$ and $\mu>\sigma$, $\epsilon=\mu+\beta\sigma$, and the cluster detection step S4 finds 7 clusters with $h_1=4$, h_2 to $h_5=0$ and $h_i=1$ for $i=6,8,10$, resulting in $M=1$. Consequently, $|R_2|=4$ and $|R|=6$.

The remaining clusters are C_1, C_2, and C_3 with $|C_1|=6$, $|C_2|=8$, and $|C_3|=10$; their statistics are given in the last three rows of Table 1 where $\epsilon=\alpha\tau+\beta\sigma$. Starting Part II with $\ell=1$ we obtain 3 clusters with $h_1=2$ and $h_{10}=1$. Thus, 4 background points merged with C_1 and 2 remain in the background. For $\ell=2$ and $\ell=3$ no further merger took place and the algorithm stops. The final result are the clusters C_1, C_2, and C_3 displayed in Fig. 8(d). The unweighted nearest neighbor graph method presented in [13] also eliminated the two outlier points and discovered the C_1 cluster. However, due to thresholding problems clusters C_2 and C_3 were identified as a single cluster. The same problem occurred when employing the NN method suggested in [38].

Example 5. Data sets similar to the one displayed in Fig. 9 have been tested by several researchers [13,2,43,38]. The data set used here consists of 1748 points. The two cluster centers are clearly visible. However, visual inspection cannot ascertain with absolute certainty the cluster membership of several points. Although fuzzy membership approaches may be helpful in this case, most variations of the fuzzy c-means method still require user defined number and location of seed points. Visual inspection, however, is impossible for high-dimensional data sets. If by luck we chose $c=2$ and divide the data set into two sets, each consisting of 874 randomly chosen points, and then apply the alternative hard c-means or fuzzy c-means, the results would be as shown in Fig. 10. Here we assigned a point to a cluster if its membership fraction for this cluster was greater than its membership number for the other cluster. Visually, there is little difference between the alternative hard c-means and the alternative fuzzy c-means. Both exhibit an almost linear separation. In comparison, when using Part I of the SSNN algorithm, then for $p=1$ the initial statistics are shown in Table 2 (row 1). With $\epsilon=\alpha\mu+\sigma$ clustering results in 36 clusters and the histogram yields: $h_1=21$, $h_2=7$, $h_3=4$, $h_4=h_5=1$, $h_6=h_7=0$, $h_8=1$, h_9 to $h_{1683}=0$, and $h_{1684}=1$. Therefore $M=8$, which results in one cluster with 1684 elements and $|R_1|=64$ (cf. Fig. 9(b)).

For $p=2$, the statistics associated with the new set $X\setminus R_1$ are given in Table 2 (row 2). With $\epsilon=\beta\mu+\sigma$ we obtained 23 clusters with histogram $h_1=9$, $h_2=8$, $h_3=1$, $h_4=0$, $h_5=1$, h_6 to $h_8=0$, $h_9=1$, h_{10} to $h_{13}=0$, $h_{14}=1$, h_{15} to $h_{377}=0$, $h_{378}=1$, h_{379} to $h_{1249}=0$, and $h_{1250}=1$. Thus, $M=14$, clusters C_1, C_2 with $|C_1|=378, |C_2|=1250$ were found as well as a remainder set R_2 with $|R_2|=56$ and $|R|=120$ (see Fig. 9(c)). After Part II, cluster C_1 grows by 32 points while C_2 remains unchanged.

Example 6. Another large and noisy data set is Jain's example displayed in Fig. 1. This data set consists of 8537 points and has been the bane of clustering algorithm developers. Applying Part I of SSNN results in the values listed Table 2 (row 3). Thus $\epsilon=\alpha\mu+\sigma$, and clustering yields the histogram $h_1=75$, $h_2=19$, $h_3=9$, $h_4=5$, $h_5=1$, $h_6=3$, $h_7=2$, $h_8=1$, $h_9=2$, h_{10} to $h_{1287}=0$, and $h_{1288}=1$. Therefore, $M=9$ which results in $|R_1|=223$ and 4 viable clusters consisting of the 3 circles and the 2 spirals. The upper and lower left globular sets within each cluster have some

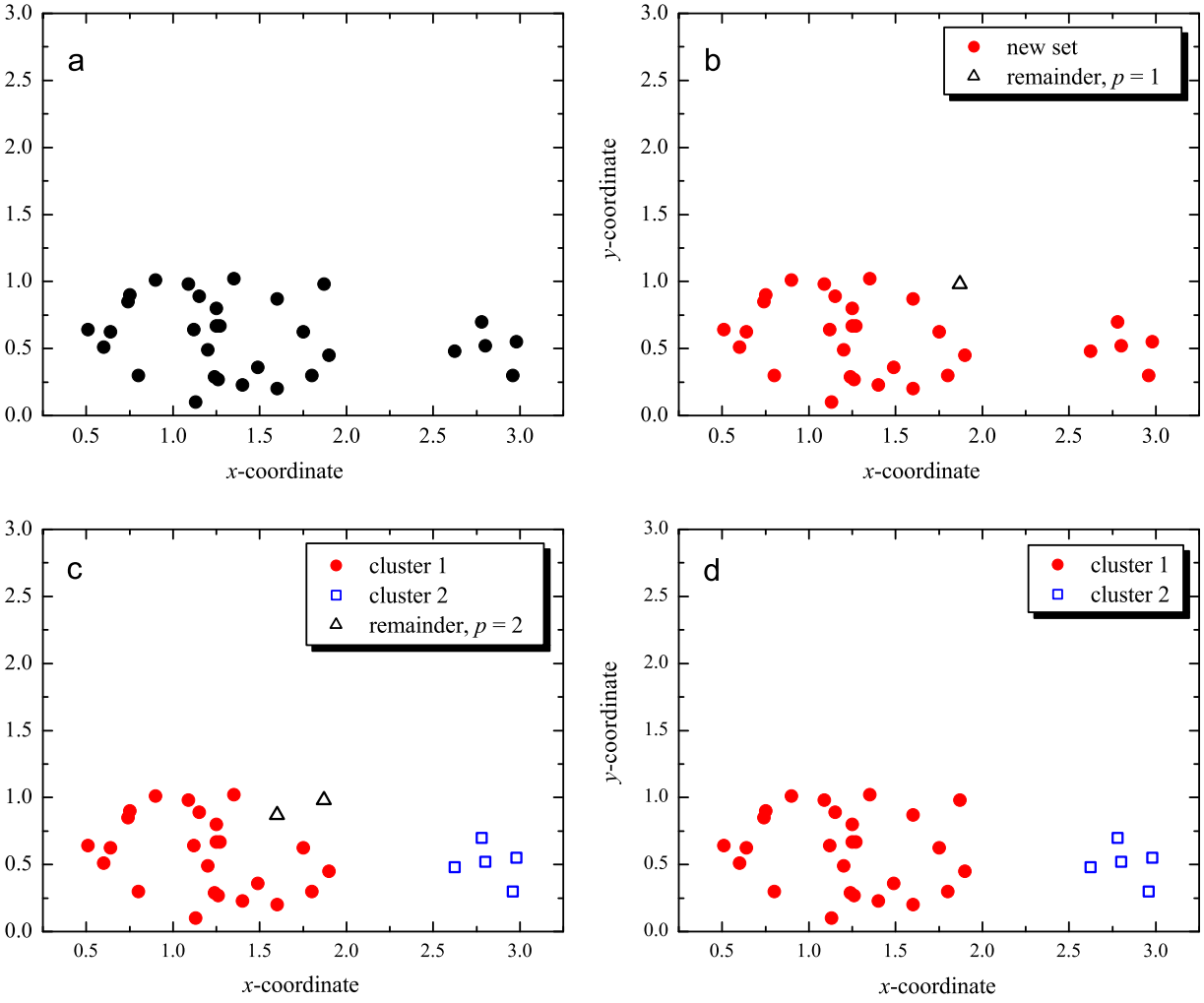

Fig. 6. (a) The data set X, $|X|=31$; (b) for $p=1$, 2 clusters result with histogram $h_1=1=h_{30}$ with $h_i=0$ for $1<i<30$, thus, $M=1$ and $|R_1|=1$; (c) for $p=2$ and $M=1$, 2 viable clusters are detected and $|R|=2$; (d) final 2 clusters identified by SSNN-Part II.

nearby noise attached. The union of these clusters make up the new set $X\setminus R_1$ shown in Fig. 11(a). For $p=2$ the statistics of the new set are provided in row 4 of Table 2. Since $\epsilon=\beta\mu+\sigma$, the clusters found in step S4 yield the histogram $h_1=85$, $h_2=13$, $h_3=6$, $h_4=1$, h_5 to $h_{300}=0$, $h_{301}=1$ so that $M=4$. With this threshold we get 7 clusters $C_1,\ldots,C_7$. Here C_1 is the globular cluster on the bottom left of Fig. 11(b) with $|C_1|=1633$, C_2 (red) is the outer spiral with $|C_2|=1941$, C_3 (green) is the inner spiral with $|C_3|=1412$, C_4 (blue) is the outer circle with $|C_4|=1018$, C_5 is the globular cluster on the top left with $|C_5|=1266$, C_6 is the middle circle with $|C_6|=610$, and C_7 is the innermost circle with $|C_7|=301$. Also, $|R_2|=133$ resulting in a background R consisting of 356 points. During the merging procedure (Part II) clusters C_1, C_3, C_4, C_6, and C_7 remain the same. Cluster C_2 adds 2 points while cluster C_5 accumulates 12 points from the background. Comparing the results of Part II shown in Fig. 11(c) with those shown in Fig. 11(b), it is clear that had we stopped with the results obtained from Part I we would not have included as many noise points with the spiral clusters but would have missed adding valid cluster points to the globular cluster C_5.

The examples given thus far are 2-dimensional sets as such data can be visualized. High dimensional patterns loose this desirable property and are, therefore, seldom used for demonstration purposes. However, one type of high dimensional pattern vectors that can be displayed in 2D are images. Using the standard row-scan method, an $m\times n$ image $\mathbf{p}$ can be converted into an mn-dimensional pattern vector $\mathbf{x}=(x_1,\ldots,x_{mn})$ by defining $x_{n(i-1)+j}=\mathbf{p}(i,j)$ for $i=1,\ldots,m$ and $j=1,\ldots,n$. Clustering images has long been an active area of investigation in computer vision and database research [3,12,19,20,23]. Problems arising in image clustering are numerous. We conclude this section by examining the performance of the SSNN algorithm when confronted with images.

Example 7. The grayscale face images of 5 different persons displayed in Fig. 12 are of size 44×32 and were taken under different illuminating conditions. The 25 images are a small subset of the CMU PIE Face Dataset [37]. This data set as well as the Yale Face database [20] have yielded at best some moderate successes to clustering approaches. Some of the best clustering performances were achieved by two algorithms developed by J. Ho and his colleagues [23]. Applying the SSNN algorithm to the image set $X=\{\mathbf{x}^1,\ldots,\mathbf{x}^{25}\}$ results in the statistics shown in Table 2 (row 5). Accordingly, $\epsilon=\tau+\sigma$ and clustering yields 5 clusters with histogram h_1 to $h_4=0$ and $h_5=5$. Thus, $M=1$, $R_1=\varnothing$, and the algorithm stops.

Example 8. A set of 35 binary images of size 50×50 (see Fig. 13) have been corrupted by random impulsive noise in multiples of 5% applied to five different noiseless images. If f represents the percentage of random noisy pixels, then $f_\lambda=5\lambda$, where $\lambda=0,\ldots,6$. If $\lambda=0$, a noiseless image results while if $\lambda=6$ the image is corrupted by 30%. In this example, instead of using the L_∞ metric we apply the L_1 metric $d(\mathbf{x},\mathbf{y})=\sum_{i=1}^{n}|x_i-y_i|$. With this metric, the initial statistics for $p=1$

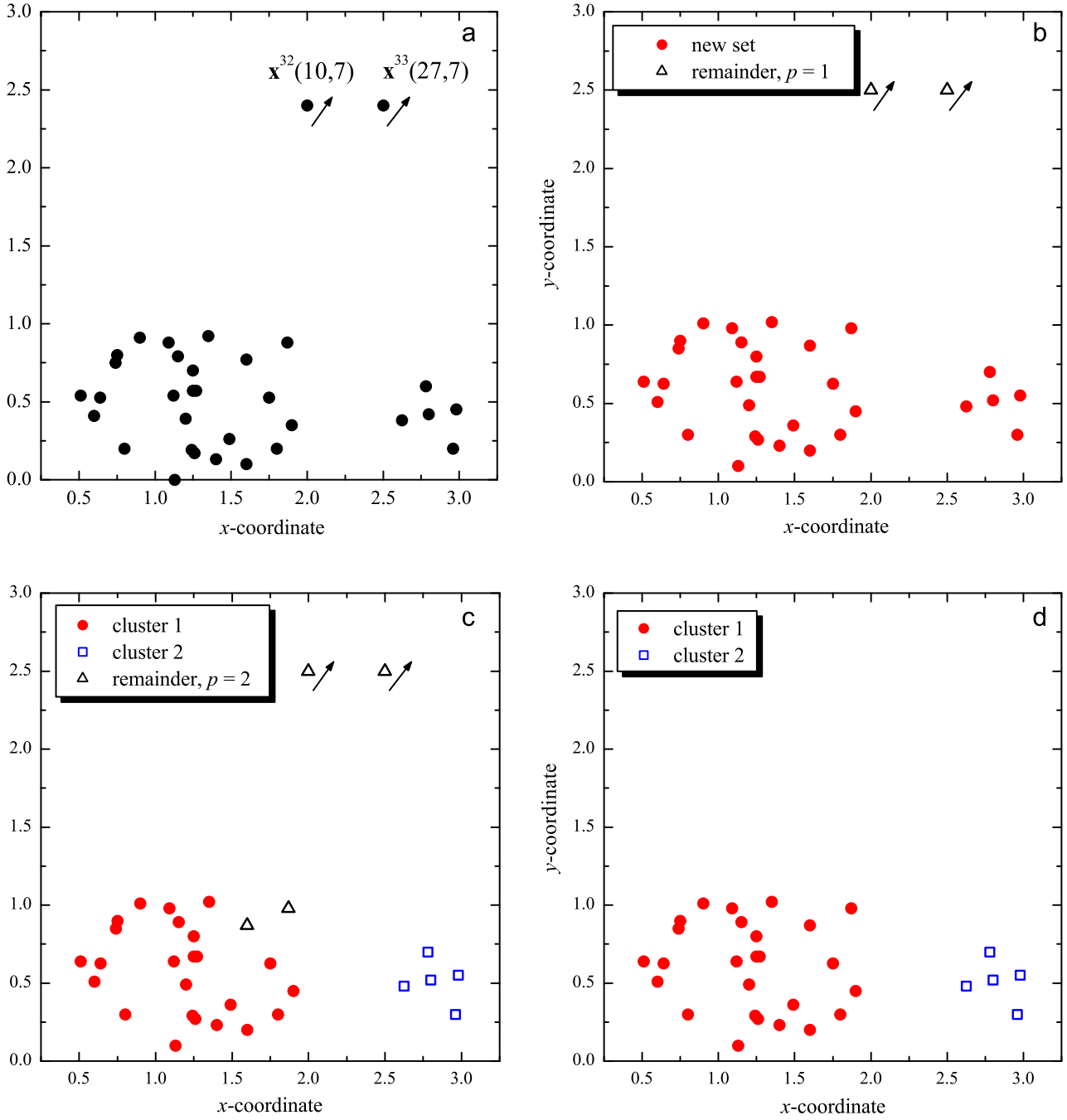

Fig. 7. (a) Data set X with $|X|=33$ and two outlier points $\mathbf{x}^{32}=(10,7)$ and $\mathbf{x}^{33}=(27,7)$ (outside the graph); (b) for $p=1$, $h_1=2$ and $h_{31}=1$ so that $M=1$ and $R_1=2$, the resulting clusters are as shown; (c) for $p=2$, $M=1$ is obtained and two viable clusters are detected where $|R|=4$; (d) final 2 clusters detected by algorithm SSNN-Part II.

of the image set $X=\{\mathbf{x}^1,\ldots,\mathbf{x}^{35}\}$ are listed in Table 2 (row 6). Since $\beta\leq 1.5$, $\epsilon=\tau+\sigma$ yields 5 clusters whose histogram is h_1 to $h_6=0$ and $h_7=5$ so that $M=1$ and $R_1=\varnothing$.

4. Conclusions

We establish a new clustering algorithm based on simple statistical parameters derived from the set of nearest neighbor distances of points of a given data set. The ϵ-value defining *nearness* for clustering points is based on the interrelationship of the maximum value τ, the mean μ, and the standard deviation σ of the set of minimal distances. Two sensitivity parameters α and β connect τ with μ and σ. The values of the five parameters τ, μ, σ, α, and β are the key for the determination of the nearness distance value ϵ. The decision as to which arithmetic combination of these five parameters should be used for the computation of ϵ was derived from the five theorems listed in Section 2 as well as testing clustering performance on a large number of different data sets. On all the data sets that we tested, the SSNN algorithm performed at least as well as other current algorithms (that were either easily available or implementable) and in several cases exhibited superior performance, e.g., Jain's example. In addition to its simplicity, another appealing feature of the SSNN algorithm is that no starting values such as ϵ-parameters, minimum points or the number k for k nearest neighbors have to be set by the user. Hence, the algorithm runs autonomously and differs from algorithms based on the c-means, k-nearest neighbor, OPTICS, or DeLiClu [1] approaches. The data sets used in Examples 1–8 can

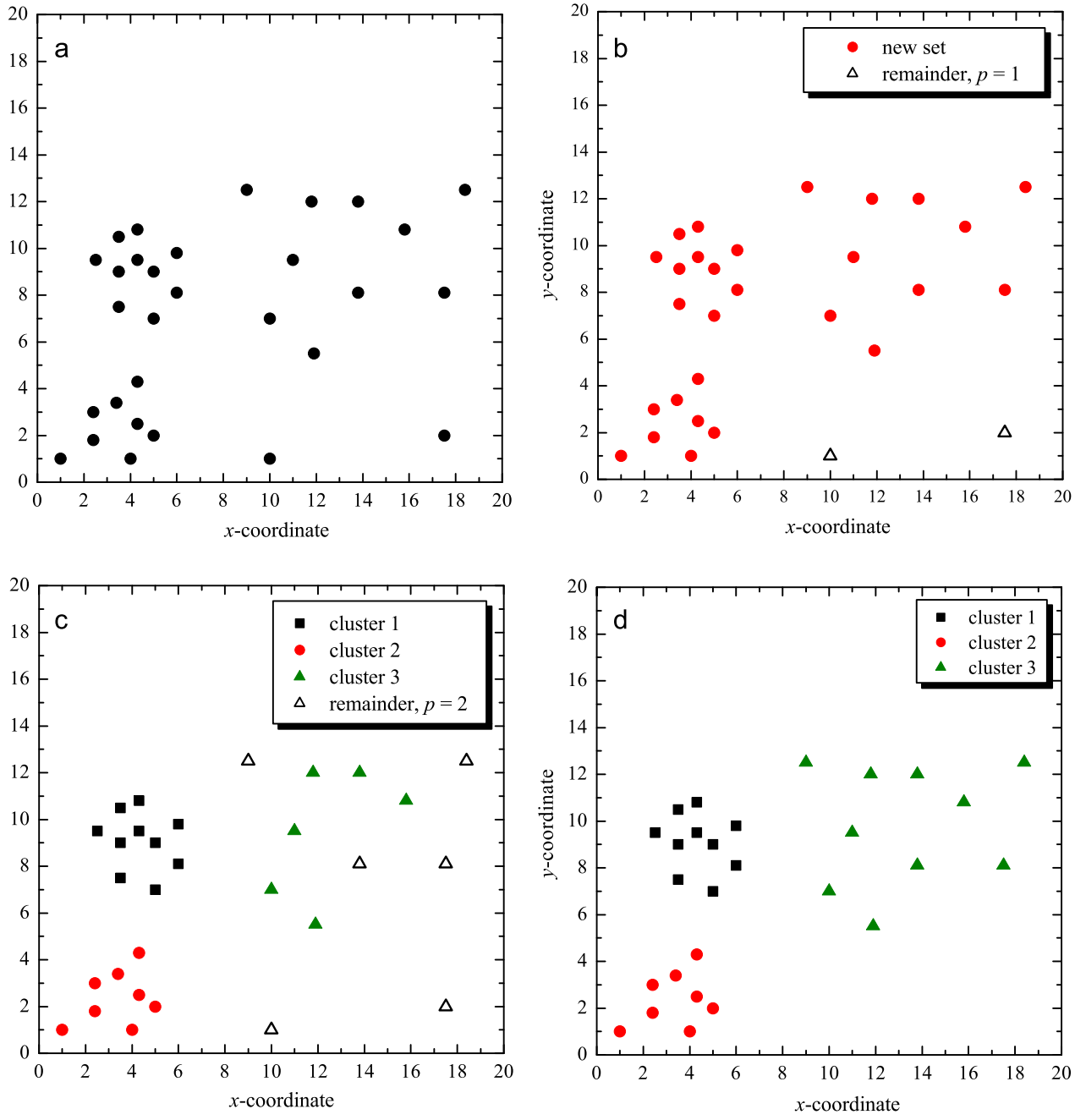

Fig. 8. (a) Data set X, $|X|=30$; (b) for $p=1$ the algorithm finds $M=1$, one viable cluster (28 points), and a remainder set R_1 (2 points); (c) for $p=2$ the results are $M=1$, 3 viable clusters, $|R_2|=4$, and $|R|=6$; (d) the final 3 clusters detected by algorithm SSNN-Part II.

Table 1
Computed statistical parameters and sensitivity values for Example 4.

Variable	τ	μ	σ	α	β	ϵ
$p=1$	5.6	1.677	1.137	1.990	2.394	4.474
$p=2$	2.8	1.436	0.701	1.310	1.638	2.584
C_1	2.5	2.050	0.206	1.081	1.172	2.944
C_2	1.4	0.975	0.222	1.169	1.355	1.937
C_3	1.5	0.940	0.229	1.283	1.479	2.263

be found at http://cise.ufl.edu/~ritter/dataset.zip. Despite its excellent performance on a large variety of data sets, we do not claim that the SSNN technique is superior to other methods when applied to any data set. It has its own inherent weaknesses. Foremost among these is the single link problem [24], i.e., if there is a chain of single points between two clusters, then the two clusters may not be separated. This would be the case when each point in the chain has a nearest neighbor within the computed ϵ distance. However, removing or cutting single chains has its own inherent problems. For instance, consider the configuration of points in Fig. 14.

Cutting single link chains would result in 4 clusters. But should this be a single cluster or four separate clusters? To some researchers it will be one cluster but others will say four. In various non-trivial digital topology applications the data represented by Fig. 14 would be classified as a single connected component and therefore a single cluster [21]. This brings us back to Section 1 of this paper. There is no cure-all clustering mechanism since there is no rigorous mathematical definition of a cluster. Clusters are defined in terms of similarity measures, and these vary widely and are data and application

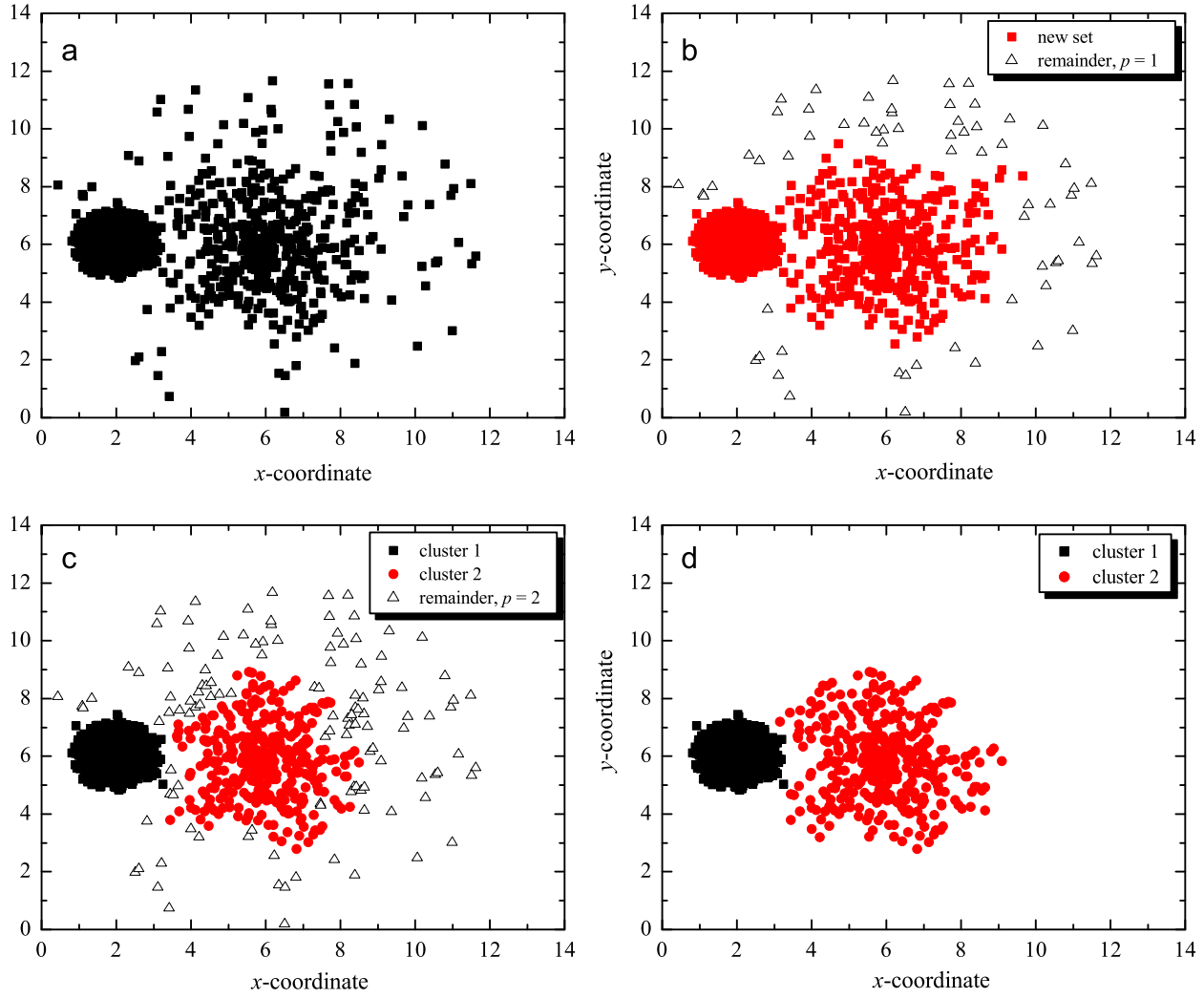

Fig. 9. (a) Data set X with $|X| = 1748$; (b) for $p=1$, threshold $M=8$ provides one viable cluster C_1 with 1684 points and $|R_1| = 64$; (c) for $p=2$, $M=14$ and 2 viable clusters C_1, C_2 are obtained. Also, $|R_2| = 56$ and $|R| = 120$; (d) final 2 clusters obtained by SSNN-Part II.

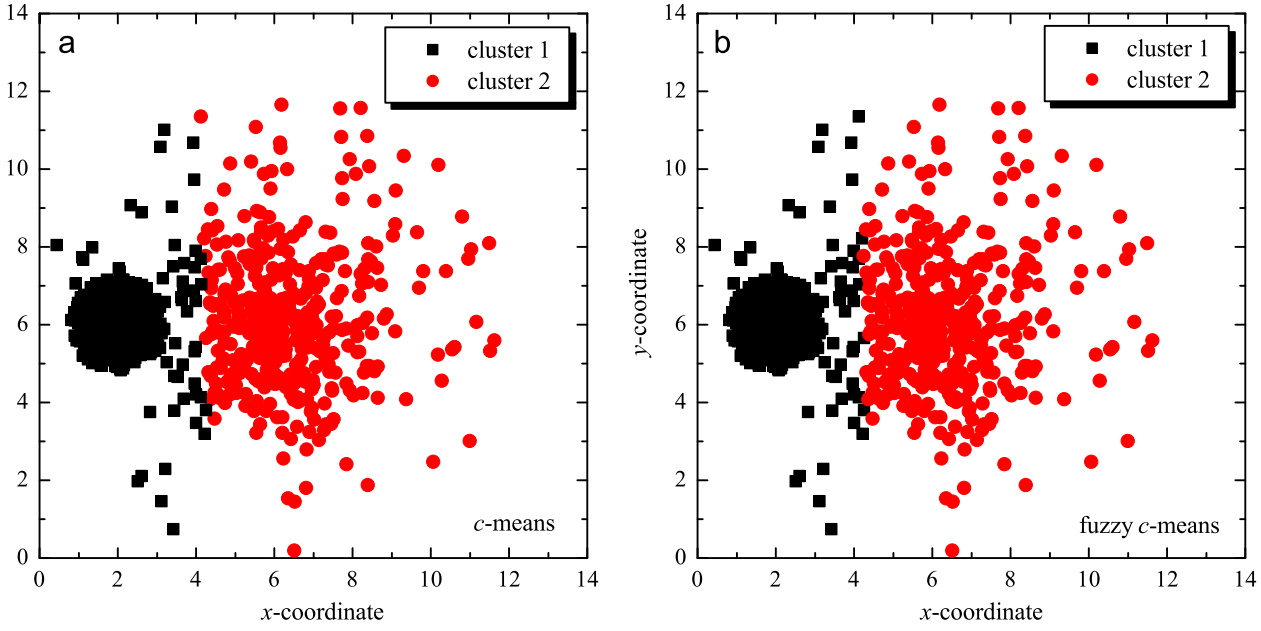

Fig. 10. Left, 2 clusters obtained with the alternative hard c-means algorithm; right, similar result using the alternate fuzzy c-means algorithm.

Table 2
Computed statistical parameters and sensitivity values for Examples 5–8.

Variable	τ	μ	σ	α	β	ϵ
$p=1$, Ex.5	1.256	0.0624	0.1095	7.307	7.944	0.565
$p=2$, Ex.5	2.800	0.0473	0.0626	25.478	26.048	1.295
$p=1$, Ex.6	3.037	0.0881	0.1877	11.012	11.693	1.158
$p=2$, Ex.6	1.079	0.0659	0.0987	6.555	7.155	0.570
$p=1$, Ex.7	28,209	18,206	6907	1.123	1.398	35,116
$p=1$, Ex.8	134	118.91	7.1529	1.063	1.120	141.15

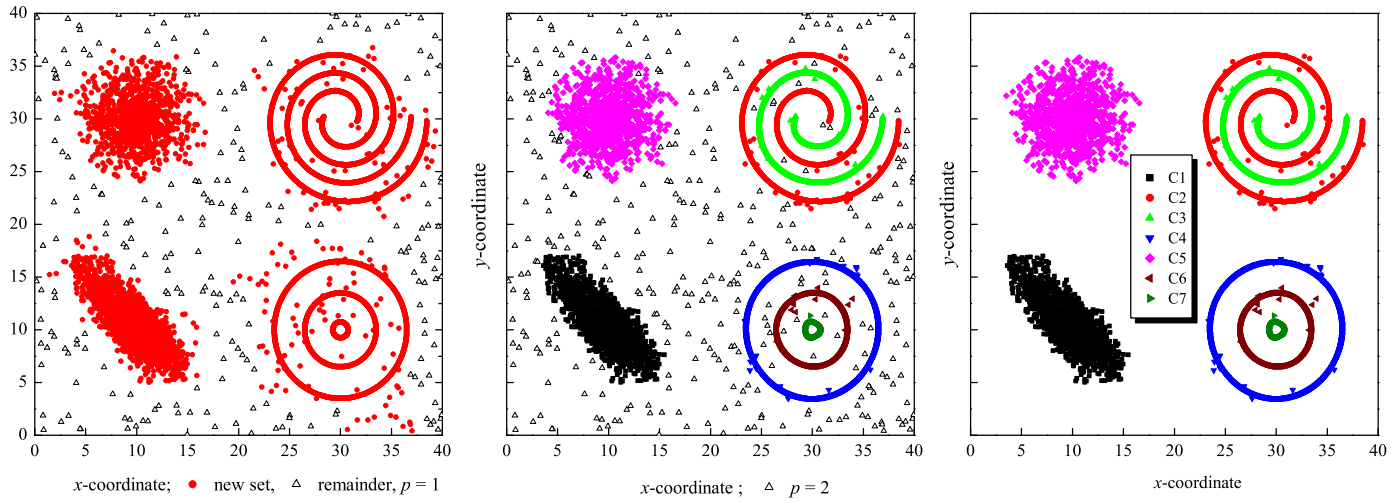

Fig. 11. (a) The new set $X \setminus R_1$ and R_1; (b) background and the 7 clusters obtained for $p=2$; (c) final 7 clusters obtained with algorithm SSNN-Part II.

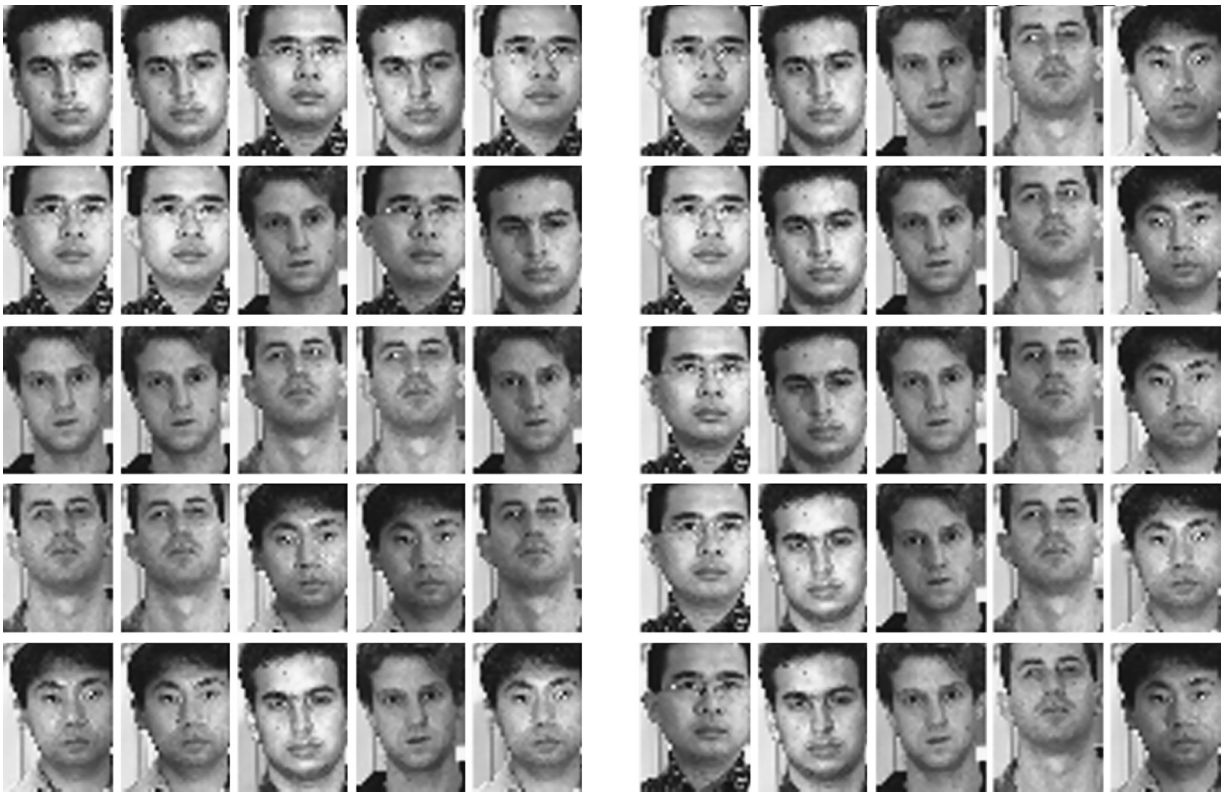

Fig. 12. Left, set of input grayscale images; right, the 5 clusters found by the SSNN algorithm where cluster C_j corresponds to the jth column and $j=1,\ldots,5$.

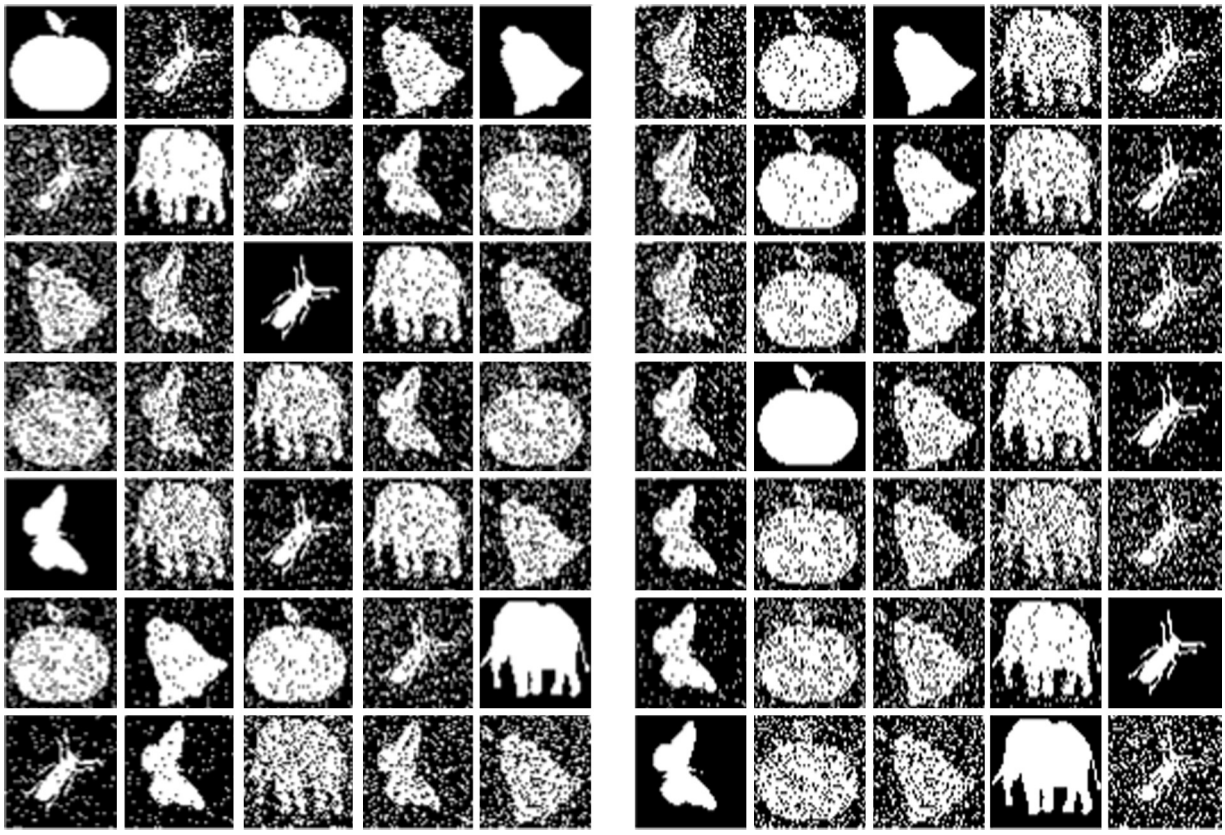

Fig. 13. Left, set of input binary images; right, the 5 clusters found by the SSNN algorithm where cluster C_j corresponds to the jth column and $j=1,\ldots,5$.

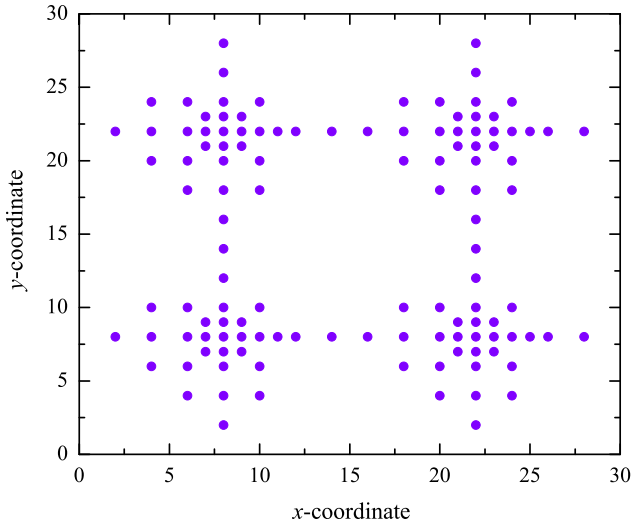

Fig. 14. One cluster or four clusters?.

dependent. For instance, at the beginning of Section 2 we mentioned hyperspectral image segmentation where single point clusters play an important role as they often represent endmembers, which are vital for unmixing pixel spectra.

Conflict of interest

None declared.

Acknowledgments

José A. Nieves-Vázquez thanks for support from the University of Florida and CONACYT in Mexico City for scholarship # 167135 and Gonzalo Urcid is grateful to SNI-CONACYT for grant # 22036.

Appendix

It follows from the definition of μ and σ that $\tau_{\min} \le \mu \le \tau$ and $0 \le \sigma \le \tau$. These two inequalities are the main tools in the proofs of the theorems.

Proof of Theorem 1.

$$\sigma^2 = \sum_{i=1}^{k} (\tau_i - \mu)^2/k = \sum_{i=1}^{k} (\tau_i^2 - 2\mu\tau_i + \mu^2)/k$$

$$\le \sum_{i=1} (\tau_i^2 - 2\mu^2 + \mu^2)/k = \sum_{i=1}^{k} \tau_i^2/k - \sum_{i=1}^{k} \mu^2/k$$

$$\le \sum_{i=1}^{k} \tau^2/k - \sum_{i=1}^{k} \mu^2/k = \tau^2 - \mu^2 \quad \text{or} \quad \mu^2 + \sigma^2 \le \tau^2,$$

where the first inequality follows from $\mu \le \tau$ and the second from $\tau_i \le \tau$ □

A result of Theorem 1 is that for a given finite set T of minimal distances, the domain D_f of function $f(\mu,\sigma) = \mu + \sigma$ must be a subset of the shaded region bounded by the quarter circle shown in Fig. 15. Since we also have the inequality $0 < \tau_{\min} \le \mu$, D_f must be a strict subset of the shaded region. It becomes clear that $\mu \to \tau$ implies that $\sigma \to 0$ and $\sigma < \mu$ or, more precisely, that $\lim_{\mu \to \tau} f(\mu,\sigma) = \tau$. Another fact illustrated in the figure is that if $\mathbf{p} = (p_0, p_0)$ corresponds to the

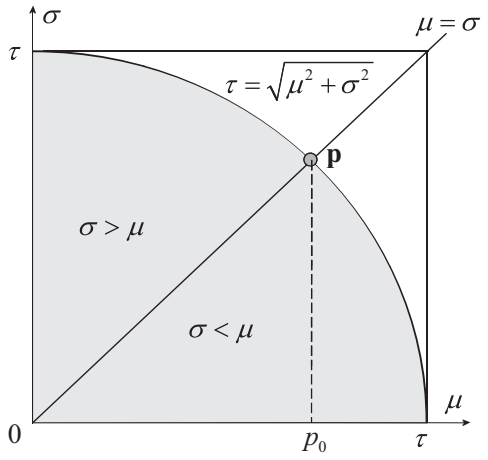

Fig. 15. The domain of $f(\mu,\sigma)$ is a subset of the shaded region.

intersection of the quarter circle and the line $\mu=\sigma$, then $\sigma<\mu$ whenever $p_0<\tau_{\min}$.

Proof of Theorem 2. We first prove that $\alpha>2/3$. Suppose to the contrary that $\alpha\le 2/3$. Then $\tau\le(2/3)(\mu+\sigma)$ or, equivalently, $3\tau\le 2(\mu+\sigma)$. It follows from Theorem 1 that $9(\mu^2+\sigma^2)\le 9\tau^2\le 4(\mu+\sigma)^2=4(\mu^2+\sigma^2)+8\mu\sigma$. Thus, $5(\mu^2+\sigma^2)-8\mu\sigma\le 0$. But $4(\mu^2+\sigma^2)< 5(\mu^2+\sigma^2)$ and we obtain the contradiction that $0\le(2\mu-2\sigma)^2=4(\mu^2+\sigma^2)-8\mu\sigma<0$. To prove the fact that $\beta\ge 1$ we again employ reductio ad absurdum. Thus we suppose that $\beta<1$. Then $(\tau+\sigma)/(\mu+\sigma)<1$ or $\tau+\sigma<\mu+\sigma$. Solving for τ we obtain the contradiction that $\tau<\mu$. The proof of $0\le\beta-\alpha<1$ follows from the definition of α and β. □

Proof of Theorem 3. Since $\beta-\mu=\sigma/(\mu+\sigma)$ we have

$$\sigma\le\mu\Leftrightarrow 2\sigma\le\sigma+\mu\Leftrightarrow 2\le(\sigma+\mu)/\sigma\Leftrightarrow\sigma/(\mu+\sigma)\le 1/2.$$

The argument for $\mu<\tau\Leftrightarrow 1/2<\sigma/(\mu+\sigma)$ is analogous. □

Proof of Theorem 4.

$$\beta-\alpha\le 3/2-\alpha\le 3/2-1=1/2.$$

The result now follows from Theorem 3. □

Proof of Theorem 5.

$$\mu=\sum_{i=1}^{k}\tau_i/k=[\ell\tau_{\min}+\sum_{\tau_i\neq\tau_{\min}}\tau_i]/k\le[\ell\tau_{\min}+(k-\ell)\tau]/k$$
$$=[k\tau+\ell\tau_{\min}-\ell\tau]/k=\tau+\ell(\tau_{\min}-\tau)/k$$

Subtracting μ and $\ell(\tau_{\min}-\tau)/k$ from both sides of the inequality gives the desired result. Similarly,

$$\mu=\sum_{i=1}^{k}\tau_i/k\ge[m\tau+(k-m)\tau_{\min}/k]$$
$$=[m(\tau-\tau_{\min})+k\tau_{\min}]/k=m(\tau-\tau_{\min})/k+\tau_{\min}.$$

Subtracting $\tau_{\min}$ from both sides of the inequality yields the desired result □

Theorem 5 provides lower bounds for $\tau-\mu$ and $\mu-\tau_{\min}$ that are dependent on the number of occurrences of τ and $\tau_{\min}$ in the collection T and the number k, but independent of μ. Since the distance $\tau-\tau_{\min}=(\tau-\mu)+(\mu-\tau_{\min})$, the theorem provides another tool for visualizing the relationships of the basic statistical measures used in the SSNN algorithm. For instance, if $p_0\le\tau_{\min}$, then we have another quarter circle restriction of the domain D_f. In this case the circle is given by $\sqrt{(\tau-\mu)^2+\sigma^2}=\tau-\tau_{\min}$ since the additional restriction is due to the inequality $(\tau-\mu)^2+\sigma^2\le(\tau-\tau_{\min})^2$ whenever $p_0\le\tau_{\min}$.

References

[1] E. Achtert, C. Boehm, P. Kroger, DeLiClu: boosting robustness, completeness, usability, and efficiency of Hierarchical clustering by closest pair ranking, in: Advances in Knowledge Discovery and Data Mining, Springer, Lecture Notes in Computer Science, vol. 3198, Berlin, 2006, pp. 119–128.

[2] M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in: Proceedings of ACM SIGMOD'99, International Conference on Management of Data, Philadelphia, ACM Press, New York, 1999, pp. 49–60.

[3] R. Basri, D. Roth, D. Jacobs, Clustering appearances of 3D objects, in: Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition, 1998, pp. 414–420.

[4] S. Ben-David, D. Pal, H.U. Simon, Stability of *k*-means clustering, in: Proceedings of 20th Conference on Learning Theory, Springer, Berlin, 2007, pp. 20–34.

[5] J.C. Bezdek, Cluster validity with fuzzy sets, J. Cybern. 3 (1974) 58–71.

[6] J.C. Bezdek, A convergence theorem for the fuzzy c-means clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 2 (1) (1980) 1–8.

[7] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, OPTICS-OF: identifying local outliers, in: Principles of Data Mining and Knowledge Discovery, Springer, Berlin, 1999, pp. 262–270.

[8] H. Chang, D-Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 14 (2008) 191–203.

[9] Y.M. Cheung, *k**-Means: a new generalized k-means clustering algorithm, Pattern Recognit. Lett. 24 (2003) 2883–2893.

[10] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, New York, NY, 2001.

[11] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters, J. Cybernet. 3 (1974) 32–57.

[12] S. Edelman, Representation and Recognition in Vision, MIT Press, Boston, MA, 1999.

[13] L. Ertoz, M. Steinbach, V. Kumar, Clusters of different sizes, shapes, and densities in noisy high dimensional data, in: Proceedings of SIAM International Conference on Data Mining, San Francisco, CA, 2003, pp. 47–58.

[14] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large databases with noise, in: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[15] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1411–1415.

[16] B. Fischer, T. Zoeller, J.M. Buhman, Path-based pairwise data clustering with application to texture segmentation, in: Proceedings of 3rd International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, France, 2001, pp. 235–250.

[17] R.A. Fisher, The use of multiple measurements in taxonomic problems, in: Annual Eugenics, vol. 7 (Part II), 1936, pp. 179–188.

[18] C. Fraley, A. Raferty, How many clusters? Which clusters? Answers via model-based cluster analysis, Comput. J. 41 (1998) 578–588.

[19] Y. Gdalyahu, D. Weinshall, Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (12) (1999) 1312–1328.

[20] A. Georghiades, D. Kriegman, P. Belhumeur, From few to many: generative models for recognition under variable pose and illumination, IEEE Trans. Patteren Anal. Mach. Intell. 40 (6) (2001) 643–660.

[21] R.W. Hall, G.T. Herman, Y.T. Kong, R. Kopperman, Digital Topology: Theory and Applications, Springer Monographs in Computer Science, Berlin, 2006.

[22] J.A. Hartigan, Clustering Algorithms, John Wiley & Sons, New York, NY, 1975.

[23] J. Ho, M.H. Yang, J. Lim, K.C. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: Proceedings of the IEEE, Conference on Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 11–18.

[24] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Upper Saddle River, NJ, 1988.

[25] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (2010) 651–666.

[26] R.A. Jarvis, E.A. Patrick, Clustering using a similarity measure based on shared nearest neighbors, IEEE Trans. Comput. 22 (11) (1973) 1025–1034.

[27] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York. NY, 2005.

[28] J. Kolen, T. Hutheson, Reducing the time complexity of the fuzzy *c*-means algorithm, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 263–267.

[29] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, CA, 1967, pp. 281–297.

[30] E.G. Mansoor, FRBC: a fuzzy rule-based clustering algorithm, IEEE Trans. Fuzzy Syst. 19 (5) (2011) 960–970.

[31] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, vol. 40, MIT Press, Cambridge, MA, 2002, pp. 849–856.

[32] F. Nie, D. Xu, X. Li, Initialization independent clustering with actively self-training method, IEEE Trans. Syst. Man Cybern. 42 (1) (2012) 17–27.

[33] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A possibilistic fuzzy *c*-means clustering algorithm, IEEE Trans. Fuzzy Syst. 13 (4) (2005) 517–530.

[34] G.X. Ritter, G. Urcid, Learning in lattice neural networks that employ dendritic computing, in: Computational Intelligence based on Lattice Theory, vol. 67, Springer, Heidelberg, 2007, pp. 25–44.

[35] J. Sander, Generalized Density-Based Clustering for Spatial Data Mining, Herbert Utz, Munich, Germany, 1998.
[36] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[37] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 53–58.
[38] W. Stueltze, R. Nugent, A generalized single linkage method for estimating the cluster tree density, J. Comput. Graph. Stat. 19 (2) (2010) 397–418.
[39] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 3rd ed., Academic Press, Amsterdam, The Netherlands, 2006.
[40] G. Urcid, G.X. Ritter, *C*-means clustering of lattice auto-associative memories for endmember approximation, in: Advances in Knowledge-Based and Intelligent Information Systems, vol. 243, IOS Press, Amsterdam, The Netherlands, 2012, pp. 2140–2149.
[41] Y. Wang, C. Li, Y. Zuo, A selection model for optimal fuzzy clustering algorithm and number of clusters based on competitive comprehensive fuzzy evaluation, IEEE Trans. Fuzzy Syst. 13 (3) (2009) 568–577.
[42] M. Wong, T. Lane, A *k*-th nearest neighbor clustering procedure, J. R. Stat. Soc. Ser. B 45 (3) (1983) 362–368.
[43] K.L. Wu, M.S. Yang, Alternative *c*-means clustering algorithms, Pattern Recognit. 35 (10) (2002) 2267–2278.
[44] J. Yu, M.S. Yang, Optimality test for generalized FCM and its application to parameter selection, IEEE Trans. Fuzzy Syst. 13 (1) (2005) 164–176.
[45] K.R. Zalik, An efficient k'-means clustering algorithm, Pattern Recognit. Lett. 29 (2008) 1385–1391.
[46] L. Zelnik, P. Perona, Self-tuning spectral clustering, in: Proceedings of NIPS, 2004, pp. 1601–1608.
[47] L. Zhu, F.L. Chung, S. Wang, Generalized fuzzy *c*-means clustering algorithm with improved fuzzy partitions, IEEE Trans. Syst. Man. Cybern. 39 (3) (2009) 578–591.

Gerhard X. Ritter received the B.A. (1966) and Ph.D. (1971) degrees from the University of Wisconsin, Madison. He is a Professor Emeritus of Mathematics and of Computer Science of the Computer and Information Science and Engineering Department at the University of Florida. He is a Fellow of SPIE, a member of the European Academy of Science, a recipient of the 1998 General Ronald W. Yates Award for Excellence in Technology Transfer by the Air Force Research Laboratory and the 1989 International Federation for Information Processing(IFIP) Silver Core Award. He is the author of two books and more than 100 refereed publications in computer vision, mathematics, and neural networks. His current research interests include artificial neural networks, pattern recognition, and the mathematical foundations of image processing and computer vision.

José Angel Nieves-Vázquez received his B.E. (2000) from the Technological Institute of Minatitlán, Veracruz, Mexico, and his M.Sc. (2005) and Ph.D. (2009) in optics from the National Institute of Astrophysics, Optics and Electronics (INAOE), Tonantzintla, Mexico. He was a Postdoctoral student in the Computer and Information Science and Engineering Department at the University of Florida during the academic year 2012 and since 2013 works as an Assistant Professor at the University of Quintana Roo, Mexico. His research interests include digital image processing, pattern recognition, and artificial neural networks.

Gonzalo Urcid received his B.E. (1982) and M.Sc. (1985) both from the University of the Americas, Puebla, Mexico, and his Ph.D. (1999) in optics from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Tonantzintla, Mexico. He was a Postdoctoral Associate in the Computer and Information Science and Engineering Department at the University of Florida during the academic years 2001–2002 and since 2000 works as an Associate Professor in the Optics Department at INAOE. He holds the appointment of National Researcher from the Mexican National Council of Science and Technology (SNI-CONACYT) since 2001. His present research interests include applied mathematics, artificial neural networks, pattern recognition, and digital-optical image processing.