Multidimensional Texture Analysis for Unsupervised Pattern Classification

K. Hammouche[1] and J.-G. Postaire[2]
[1]*Université Mouloud Mammeri, Département d'Automatique Tizi Ouzou*
[2]*Université des Sciences et Technologies de Lille –*
LAGIS 59655 VILLENEUVE D'ASCQ Cedex
[1]*Algeria*
[2] *France*

1. Introduction

Clustering techniques aim to regroup a set of multidimensional observations, represented as data points scattered through a *N*-dimensional data space, into groups, or clusters, according to their similarities or dissimilarities. Each point corresponds to a vector of observed features measured on the objects to be classified. Among the different approaches that have been developed for cluster analysis [Jain et al., 1999; Theodoridis & Koutroumbas, 2003; Tran et al., 2005; Xu & Wunsch, 2005; Filipone et al., 2008], we consider the statistical approach [Devijver & Kittler, 1983]. In this framework, many clustering procedures have been proposed, based on the analysis of the underlying probability density function (pdf). The high density of data points within the clusters gives rise to modal regions corresponding to the modes of the pdf that are separated by valleys of low densities [Parzen, 1962].
Independently from cluster analysis, a large amount of research effort is devoted to image segmentation. Starting from an unstructured collection of pixels, we generally agree about the different regions constituting an image due to our visual grouping capabilities. The most important factors that lead to this perceptual grouping are similarity, proximity and connectedness. More precisely, segmentation can be considered as a partitioning scheme such that:

- Every pixel of the image must belong to a region,
- The regions must be composed of contiguous pixels,
- The pixels constituting a region must share a given property of similarity.

These three conditions can be easily adapted to the clustering process. Indeed, each data point must be assigned to a cluster, and the clusters must be composed of neighbouring data points since the points assigned to the same cluster must share some properties of similarity. Considering this analogy between segmentation and clustering, several image segmentation procedures based on the gray-level function analysis have been successfully adapted to detect the modes or to seek the valleys of the pdf for pattern classification purpose [Botte-Lecocq et al., 2007].
In this framework, the underlying pdf is generally estimated on a regular discrete array of sampling points [Postaire & Vasseur, 1982]. The idea of using a pdf estimation for mode

Source: Pattern Recognition Techniques, Technology and Applications, Book edited by: Peng-Yeng Yin,
ISBN 978-953-7619-24-4, pp. 626, November 2008, I-Tech, Vienna, Austria

seeking is not new [Parzen, 1962] and in very simple situations, the modes can be detected by thresholding the pdf at an appropriate level, using a procedure similar to image binarization [Weszka, 1978]. A "mode" label is associated with each point where the underlying pdf is above the threshold. Otherwise, the corresponding point is assigned a "valley" label.
However, in practical situations, it can be difficult, or even impossible, to select an appropriate global threshold to detect the significant modes. A solution for improving this simple scheme is to consider the spatial relationships among the sampling points where the underlying pdf is estimated, rather than making a decision at each point independently of the decisions at other points. Probabilistic labeling, or relaxation, is a formalism through which object labels are iteratively updated according to a compatibility measure defined among the neighboring labels. This approach, which has been successfully applied to image processing [Dankner, 1981], has been adapted to cluster analysis to reduce local ambiguities in the mode/ valley discrimination process [Touzani & Postaire, 1988].
The segmentation of an image can also be considered as a problem of edge detection [Davis, 1975]. Similarly, in the clustering context, a mode boundary can be localized at important local changes in the pdf. It can be detected by means of generalized gradient operators designed to perform a discrete spatial differentiation of the estimated pdf [Touzani & Postaire, 1989]. Although these spatial operators enhance substantially the discontinuities that delineate the modes, a relaxation labeling process, similar to the one used for thresholding, can be necessary for mode boundary extraction [Postaire & Touzani, 1989].
Beside procedures based on the concepts of similarity and discontinuity, mathematical morphology has proven to be a valuable approach for image segmentation. This theory has been adapted to cluster analysis by considering the sets of multidimensional observations as mathematical discrete binary sets [Postaire et al., 1993]. Binary erosions and dilations of these discrete sets eliminate irrelevant details in the shapes of the clusters without geometric distortions [Botte-Lecocq & Postaire, 1991]. Multivalue morphological operations, such as numerical erosions, dilations and homotopic thinnings have also been defined for processing multidimensional pdf using the umbra concept [Sbihi & Postaire, 1995]. With these operators, the clusters are delineated by means of the watershed transform [Benslimane et al., 1996].
Modeling spatial relationships between pixels by means of Markov random fields has proved to be relevant to the image segmentation problem [Manjunath & Chellappa, 1991; Panjwani & Healey, 1995]. The Markovian approach has been adapted to the mode detection problem in cluster analysis. The hidden field containing the "mode" and the "valley" labels is derived from the observable field representing the data set by means of the estimation-maximisation algorithm combined with the maximum a posteriori mode criterion [Sbihi et al., 2000; Moussa et al., 2008] .
All the above-mentioned clustering methods tend to generalize bi-dimensional procedures initially developed for image processing purpose. But, even though texture properties have been intensively used to solve image segmentation problems, they have not been extended to pattern classification problems. Following the main idea of adapting image processing techniques to cluster analysis, the objective of this chapter is to show how the texture concept can be used in the framework of clustering. The basic idea behind this new approach is the characterization of the local spatial distribution of the data points in the multidimensional data space in terms of textures [Hammouche et al., 2006].

Similarly to texture segmentation, the approach consists first in selecting a set of texture features that characterize the local multidimensional texture around each sampling point of the data space. These multidimensional textures, which reflect the spatial arrangement of the data points, are then classified on the basis of these features. The data points with similar local textures are aggregated in the data space to define compact connected components of homogeneous textures. Some of these multidimensional domains of uniform texture are finally considered as the cores of the clusters.

The chapter is organized as follows. We first describe the discretization process of the input data that yields an array of sampling points well conditioned for multidimensional texture analysis (Section 2). We then introduce the multidimensional texture concept itself, as an alternative to describe the spatial distribution of observations through the data space (Section 3). Such textures are locally characterized by a number of parameters that can be extracted from co-occurrence matrices or from sum and difference histograms, defined as straightforward generalizations of the tools used in textured image processing.

The mode detection strategy is based on the assumption that the texture is homogeneous within the modes of the data distribution, and different from the texture in the valleys between the clusters. Hence, similarly to segmentation of textured images, the sampling points where the local underlying texture is evaluated are classified into different texture classes in order to partition the data space into domains with homogeneous texture properties (Section 4). The determination of the set of the most discriminating texture parameters among all those that are available is based on a performance-dependent feature selection scheme (Section 5).

Many examples are presented to demonstrate the efficiency of this clustering strategy based on multidimensional texture analysis (Section 6). As the computational load could to be prohibitive for data sets of high dimensionality and large size, a specific attention is devoted to the implementation of the clustering procedure in order to improve the computation speed (Section 7).

2. Discretization of the data set

In order to adapt texture analysis tools to clustering, it is necessary to introduce a discrete array of sampling points [Postaire & Vasseur, 1982]. Let us consider Q observations $X_q = [x_{q,1}, x_{q,2}, ..., x_{q,n}, ..., x_{q,N}]$, $q = 1, 2, ..., Q$, where $x_{q,1}, x_{q,2}, ..., x_{q,n}, ..., x_{q,N}$ are the N coordinates of the observation X_q in the data space. The range of variation of each component of the multivariate observations is normalized to the interval $[0, S]$, where S is an integer, by means of the transformation:

$$x'_{q,n} = S \frac{x_{q,n} - \min\limits_{q=1}^{q=Q}\left\{x_{q,n}\right\}}{\max\limits_{q=1}^{q=Q}\left\{x_{q,n}\right\} - \min\limits_{q=1}^{q=Q}\left\{x_{q,n}\right\}} .$$

Let $X'_q = [x'_{q,1}, x'_{q,2}, ..., x'_{q,n}, ..., x'_{q,N}]^T$, $q = 1, 2, ..., Q$, be the Q new observations in the normalized data space. Each axis of this space is partitioned into S exclusive and adjacent intervals of unit width. This discretization defines an array of S^N hypercubes of unit side length. The centers of these hypercubes constitute a regular lattice of sampling points denoted P_r , $r=1,2,...,S^N$. The unit hypercubic cell centered at point P_r is denoted $H(P_r)$. It is defined by its coordinates $h_{r,1}, h_{r,2}, ..., h_{r,n}, ..., h_{r,N}$, which are the integer parts of the coordinates of its center P_r . The qth normalized observation X'_q falls into the unit cell $H(P_r)$ of coordinates $h_{r,n} = \mathrm{int}(x'_{q,n})$, $n=1,2, ..., N$, where $\mathrm{int}(x'_{q,n})$ denotes the integer part of the real number $x'_{q,n}$.

Taking the integer parts of the coordinates of all the available normalized observations yields the list of the non-empty cells whose coordinates are defined on the set Z^{+N} . If several observations fall into the same cell, this one appears many times in the list of non-empty cells. It is easy to determine the number $q\left[H(P_r)\right]$ of observations that fall into the hypercubic cell of center P_r by counting the number of times the cell $H(P_r)$ appears in that list. As this number can be considered as proportional to a rough estimate of the local density of observations, it will be referred as the "density" $W(P_r) = q\left[H(P_r)\right]$ associated to the point P_r in what follows. Subsequently, the distribution of the data points can be approximated on the discrete multi-dimensional array of points P_r . The result of this sampling procedure is a multidimensional regular array of discrete integers in the range $\left[0, G\right]$, where G is the maximum value of $W(P_r)$, $r=1,2,...,S^N$, that is well conditioned for multidimensional texture analysis. Fig. 1 shows a raw data set of bi-dimensional observations (cf. Fig 1(a)) and the corresponding array of discrete densities obtained for S =25 (cf. Fig 1(b)).

3. Multidimensional texture characterization

To illustrate the basic ideas behind the proposed approach, let us consider the bi- and three-dimensional uniform random distributions of data points of Fig. 2. A close visual attention to this figure shows that the arrangement of the observations appears to be more or less coarse and more or less sparse, depending on the density of data points in the bi- or the three-dimensional data spaces. Thanks to the capacities of perception of the human visual system, it is easy to distinguish various random textures associated with these distributions. These considerations led to consider the texture as a property of the data points distribution. In this chapter, it is assumed that the texture tends to be uniform within the core associated with each cluster, so that these cores can be searched as domains of the data space characterized by a relative homogeneity of suitable texture descriptors.

When considering the examples of Fig. 2, it is clear that structural models based on primitive placement rules cannot satisfactorily describe the texture of the distribution of the data points. Therefore, one is led to consider the textural properties in terms of statistical models and the main difficulty is the selection of a set of relevant features to describe the properties of the spatial distribution of the data. A number of textural parameters have been proposed in the image processing literature, derived from autoregressive models [Comer & Delp, 1999], Markov random fields models [Cross & Jain, 1983], Gabor filters [Jain & Farrokhnia, 1991], wavelet coefficients [Porter & Canagarajah, 1996], fractal geometry [Keller & Crownover, 1989] and spatial gray-level dependence analysis [Haralick, 1978]. We have chosen to generalize the concepts of co-occurrence matrices and of sum and difference histograms to multidimensional data spaces since a large variety of features can be derived from such texture models that combine spatial information with statistical properties [Reed & Hans du Buf, 1993].

3.1 Co-occurrence matrices

In the framework of image processing, an element T(i,j) of a co-occurrence matrix is a count of the number of times a pixel $P_{r'} = [x_{r,1}, x_{r,2}]^T$ with gray-level i is positioned with respect to a pixel $P_r = [x_{r',1}, x_{r',2}]^T$ with gray level j such as:

$$P_{r'} = P_r + \begin{bmatrix} d\cos\theta \\ d\sin\theta \end{bmatrix}$$

where d is the distance in the direction θ between the two pixels.

A similar co-occurrence matrix is determined to characterize the local distribution of the data points in a given neighborhood of each sampling point P_r where the "density" value is not null. We use a classical hypercubic neighborhood. As directionality and periodicity are obviously irrelevant characteristics of the data point distributions, it is not necessary to determine co-occurrence matrices for different values of the distance d and the orientation θ between the pairs of sampling points taken into account. Hence, only one co-occurrence matrix is determined for each sampling point. Furthermore, the use of a small neighborhood reduces the computational load, while yielding local information on the distribution of the data points. The co-occurrences T(i,j) of any given pair (i,j) of "density" values are simply counted for all the couples of adjacent sampling points encountered within a hypercubic neighborhood of side length equal to 3, without constraints on their orientations. Two sampling points are considered as adjacent if they are the centers of two hypercubes that have at least one point in common. As the "densities" are quantized on a set of $G+1$ discrete values, the co-occurrence matrices have $G+1$ rows and $G+1$ columns.

As in [Haralick et al., 1973], several local texture features can be extracted from these specific co-occurrence matrices (COM) which accumulate information on the data distribution in the neighborhood of each sampling point (cf. Table 1). These features are expected to characterize such textural properties as roughness, smoothness, homogeneity, randomness or coarseness rather than properties such as directionality or periodicity, since each co-occurrence matrix summarizes the number of occurrences of pairs of histogram values for all possible pairs of adjacent sampling points lying within a given neighborhood, without constraints on their orientations.

Fig. 3 shows the spatial variations of the 7 first features of table 1 for the data set of Fig.1 that is composed of observations drawn from three normal distributions of equal weights. The values of the features f_4, f_6 and f_7 decrease from the centers of the clusters to their peripheries. On the contrary, the values of f_1, f_2, f_3 and f_5 increase from the centers to the peripheries of the clusters. Although these seven texture features reflect the local distribution of the data points, they can be more or less correlated and more or less relevant for the detection of the cluster cores. Furthermore, it would be unrealistic to believe that the performance of the cluster core detection scheme will grow with an increasing number of features.

3.2 Sum and difference histograms

In the image processing framework, statistical texture features can also be extracted from gray-level sum and difference histograms [Unser, 1986]. These histograms are associated to couples of pixels P_r and $P_{r'}$, separated by specific distances d along a set of directions θ. For each couple (d,θ), the value $h_\Delta(i)$ of the ith bin of a difference histogram indicates the number of times such pixels have a gray-level difference $(g_r - g_{r'})$ equal to i, where g_r and $g_{r'}$ are the gray-levels at P_r and $P_{r'}$, respectively. Similarly, the value $h_\Sigma(j)$ of the jth bin of a sum histogram represents the number of occurrences of pairs of pixels P_r and $P_{r'}$ which, in the same geometrical configuration, have a sum of gray-levels $(g_r + g_{r'})$ equal to j.

This gray-level sum and difference histogram concept can be easily extended to summarize the distribution of the sums and differences of densities between pairs of sampling points. In this case, the ith bin $h_\Delta(i)$ of the density difference histogram is equal to the number of times two sampling points P_r and $P_{r'}$ of the discretized data space, separated by the displacement defined by the couple (d,θ), have a difference between their densities equal to i, i.e. $W(P_r) - W(P_{r'}) = i$, $i = -G,...,G$. Similarly, the jth bin $h_\Sigma(j)$ of the density sum histogram is equal to the number of times two sampling points P_r and $P_{r'}$ have the sum of their densities equal to j, i.e. $W(P_r) + W(P_{r'}) = j$, $j = 0,...,2G$. As the values of the densities are quantized on a set of integers in the range $[0, G]$, the sum and difference histograms have $(2G+1)$ bins each.

As when using the co-occurrence matrices, the multidimensional texture of the spatial distribution of the observations is analyzed locally around each sampling point P_r of the data space where the density is not null. For this purpose, the density sum and difference histograms (density SDH) are determined in a hypercubic neighborhood of side length equal 3, centered at point P_r, and without constraints on their orientations

In the image processing framework, several features can be computed from the gray-level sum and difference histograms [Unser, 1986; Clausi & Zhao, 2003]. Nine of the most commonly used texture features, denoted f_m, $m = 1,2,...,9$, are described in table 2. Analogously, the texture at P_r can be evaluated by means of some of the nine features of table 2 derived from the density sum and difference histograms.

Fig. 4 shows the spatial variations of the 8 first features of table 2 for the data set of Fig.1. The values of the features f_1, f_4, f_5, f_7, f_8 and f_9 decrease from the centers of the clusters to their peripheries. On the contrary, the values of f_2, f_3 and f_6 increase from the centers to the peripheries of the clusters. As for the features extracted from co-occurrence matrices, these features could be more or less suitable to describe the structure of the distribution.

A specific problem that must be addressed is now the selection of meaningful features among those of table 1 or table 2 to describe the textural information that will be used to identify the cluster cores in the data space. Each sampling point will then be characterized by a feature vector $F(P_r) = [f_1(P_r), f_2(P_r),..., f_m(P_r),..., f_M(P_r)]^T$, in a M-dimensional feature space. The selection of the M most relevant features, specifically adapted to each data set, will be discussed in section 5.

4. Cluster core extraction

4.1 Texture classification

Similarly to image segmentation, it is expected that sampling points with similar texture properties could be aggregated in the data space to detect the clusters in the data.

When the sampling points are characterized by a set of texture features, they can be represented as feature vectors in a multidimensional feature space. Texture classification consists in assigning the sampling points of the discrete data space to different texture classes defined in the feature space. This is an unsupervised classification problem since no *a priori* knowledge about the feature vectors associated with the textures to be identified is available. A simple solution is to use the basic K-means algorithm where the desired number of classes of feature vectors has to be specified [Macqueen, 1967]. The ability of varying the number of expected classes makes it possible to give some insight into the significance of the clusters that can be identified within the data.

Fig. 5 shows the domains of homogeneous textures associated with the discrete data set of Fig.1(b) when the K-means algorithm requires 2, 3 and 4 classes of different textures characterized by means of co-occurrence matrices. The texture discrimination is performed in a 2-dimensional feature space defined by two features, namely the homogeneity f_3 and the correlation f_4 of table 1, with S=25. When two classes are required, the two domains correspond to the cluster cores and the valleys, respectively (cf. Fig. 5(a)). When the sampling points are assigned to 3 classes of textures, one of them corresponds to the cores; the second to their boundaries and the last one to the valleys (cf. Fig. 5(b)). In the case of 4 classes, Fig. 5(c) shows that the cores are surrounded by concentric domains corresponding to different distribution characteristics that are obviously linked to the local data point densities.

We have kept the parameter S and the two texture features unchanged in order to show the influence of the required number of texture classes on the resulting domains of homogeneous textures. A procedure to optimize the value of S, to select an appropriate set of texture features and to determine the number of texture classes will be presented in section 5.

4.2 Core extraction

Under the assumption that the cluster cores are multidimensional domains with homogeneous textures, it is expected that the hypercubes centered on the sampling points assigned to the same class of texture give rise to connected components in the data space. These components can be extracted by means of an aggregation procedure where two hypercubes whose centers belong to the same class of texture are assigned to the same component if they have at least one point in common. Small components resulting from this aggregation procedure may correspond to non significant domains with only a few data points. Therefore any domain containing less than 5% of the total number Q of observations is discarded.

Among the remaining components, those corresponding to the cores of the clusters are expected to be more compact than those corresponding to their boundaries or to the valleys between them. Hence, they can be discriminated from other components by analyzing their compactness defined as:

$$C = \frac{[\text{total number of hypercubes}]}{[\text{number of boundary hypercubes}]^N}$$

This compactness, which is as much as high as the component is compact, depends mainly on the dimensionality and on the structure of the data. In practice, the selection of the domains with a compactness higher than 50% of the highest compactness value among all the detected domains has proved to be a good strategy to identify the cluster cores [Hammouche et al., 2006].

Table 3 indicates the compactness of the domains resulting from the aggregation of the connected sampling points of Fig. 5. It is clear that the cluster cores are much more compact than the other domains. Cluster core detection is straightforward by simple thresholding of the compactness. Fig. 6 shows the cores identified among the domains of homogeneous texture of Fig. 5.

Due to irregularities in the distribution of the data points, especially for small data sets, the boundaries of the selected domains may present irrelevant details. In such situations, multidimensional binary morphology has proved to be an efficient solution to eliminate details in the data structure without changing the global shape of unsuppressed domains [Botte-Lecocq & Postaire, 1991]. A classical closing-opening operation, using a hypercubic structuring element of side length equal to 3, generally yields regularly shaped cluster cores.

Finally, many supervised classification procedures can be used to assign the observations to the clusters attached to the detected cores. One solution is to use the observations falling into the cores as prototypes. The remaining observations are assigned to the cluster attached to their nearest neighbor among these prototypes. They are assigned one by one to the clusters in a specific order depending on their distances to the prototypes. At each step of this procedure, we consider the distances between all the unassigned observations and all the prototypes. The smallest among these distances indicates the specific observation that must be assigned to the cluster attached to its nearest neighbor. It is integrated within the set of prototypes defining this cluster. This updating rule is iterated until all the observations are classified [Botte-Lecocq & Postaire, 1994].

5. Algorithm tuning and feature selection

The performance of the above described algorithm depends mainly on the adjustment of the discretization parameter S and on the relevance of the chosen texture features.

5.1 Discretization tuning

Let us first consider the effect of the resolution of the discretization process. In fact, the adjustment of S depends on the sample size Q, on the dimensionality N of the data and on the structure of the distribution of the observations. It can be expected that, when true clusters exist, stable connected subsets of data points with similar texture properties appear for a wide range of values of S. Based on this assumption, the adjustment of S can be governed by the concept of cluster stability [Eigen et al., 1974]. Choosing such a parameter in the middle of the largest range where the number of detected clusters remains constant, and different from one, has proved to be a good procedure to optimize a number of clustering algorithms when nothing is *a priori* known about the structure of the distribution of the observations [Postaire & Vasseur, 1981]. Note that the larger the range, the more reliable is the tuning procedure.

5.2 Feature selection

In the framework of multidimensional texture analysis, the key problem is the selection of a set of suitable texture features. For choosing relevant features while reducing the dimensionality of the texture classification problem, we propose a performance-dependent feature selection scheme which is directly related to the above mentioned concept of cluster stability. The effectiveness of a subset of features is evaluated by means of the width of the largest range of values of the discretization parameter S leading to the appropriate number of detected cluster cores. As mentioned at the end of § 5.1, the larger this range, the more reliable is the number of detected cores. This criterion is used to select a set of relevant features among the available ones by means of a sequential forward selection technique [Siedlecki & Sklansky, 1988].

To evaluate the relative relevance of M features $f_1,...,f_m,...,f_M$, we consider the feature subspaces $R^1,...,R^m,...,R^M$, taking into consideration an increasing number of texture features, from one to M. The algorithm starts with the M possible R^1 spaces. The feature which maximizes the range of values of S corresponding to a stable number of detected cores, different from one, is the first selected feature. This feature is combined, in a R^2 feature space, with each of the $M-1$ remaining ones. The corresponding $M-1$ lengths of the stable ranges for S are then determined and the pair of features that maximizes the length is kept.

When m features out of M have been chosen, the algorithm proceeds in the R^{m+1} feature space of $m+1$ dimensions to select the $(m+1)^{th}$ feature that maximizes the length of the range of S when combined with the m previously chosen features. This procedure is iterated until the M features have been ordered by diminishing relevance. The sequence $L(m)$ of length values thus obtained allows selecting a subset of relevant features within the set of M features. These salient features are those that correspond to the starting increasing phase of the length values in the sequence $L(m)$. All the features that follow the first decrease in the sequence $L(m)$ are discarded.

To demonstrate the efficiency of the proposed feature selection technique, we use the bi-dimensional data set of Fig. 1 constituted of three Gaussian clusters. The length $L(m)$ of the longest range of values of S where the same number of cluster cores is detected by the clustering procedure is plotted against the number m of selected features (cf. Fig. 7). The feature selected at each step is indicated at the corresponding point of the plot. The series $(f_4, f_5, f_1, f_3, f_6, f_2, f_7)$, $(f_5, f_1, f_2, f_4, f_7, f_3, f_6)$ and $(f_5, f_2, f_6, f_4, f_3, f_1, f_7)$ represent the 7 first selected features among the 13 computed from the co-occurrence matrices, ordered by decreasing relevance when 2, 3 and 4 classes of textures are required by the K-means algorithm, respectively. As expected, the number of required classes influences the feature selection. When 2 classes are required, the selected features are f_4, f_5 and f_1 since $L(m)$ begins to decrease when f_3 is selected (cf. Fig. 7(a-1)). With 3 classes of textures, the plot of Fig. 7(b-1) shows that the two first features f_5 and f_1 are selected for detecting the 3 clusters. When 4 classes of textures are used, it appears that only the first feature f_5 is selected for detecting the 3 clusters (cf. Fig. 7(c-1)). Fig. 7(a-2), 7(b-2) and 7(c-2) show the ordered features extracted from density sum and difference histograms when 2, 3 and 4 classes of textures are required by the K-means algorithm, respectively.

5.3 Number of texture classes

The next parameter that remains to be adjusted is the number of texture classes required by the K-means algorithm. This number is not determined automatically by the basic, but well controlled, version of the algorithm used in this work. In fact, the concept of cluster stability allows specifying this number by selecting the number of texture classes that leads to the longest range of variation of S where the number of detected cores remains constant. Fig. 7 shows that this wider range is reached when the textures are assigned to two classes with the three features f_4, f_5 and f_1 extracted from the COM and with the five features f_9, f_3, f_6, f_1 and f_4 extracted from the density SDH.

5.4 Hypercubic neighborhood size

The neighborhoods used to determine the local values of the texture features have been defined as hypercubes of side length equal to 3 (cf. § 2.2). But we could have used larger neighborhoods constituted of $(2h+1)^N$ unit cells centered at the sampling points. We have analyzed the effect of the parameter h on the behavior of the algorithm. For each neighborhood size varying from h =1 to h =4, we have selected the relevant texture features as explained in § 5.2 to classify the bi-dimensional data of figure 1(a), asking always for two texture classes. Table 4 indicates the largest ranges of the discretization parameter S where the numbers of detected clusters remain constant for each neighborhood size. It appears that the largest of these ranges are obtained for h =1 . Furthermore, beside being the best choice in terms of reliability of the results, the choice of the minimal neighborhood size (h =1) reduces the computation time while improving the sensitivity of the procedure to local texture properties.

6. Experimental results

The following examples have been chosen to provide some insight into the behavior of the proposed texture based clustering procedure and to demonstrate the interest of this approach for pattern classification.

6.1 Example 1

The first example illustrates all the steps of the algorithm and demonstrates the ability of the procedure to detect clusters of unequal weights. The data set is presented in Fig. 8(a). It is composed of 950 bidimensional observations drawn from the four normal distributions of unequal weights specified in table 5.
The local texture features are computed from the co-occurrence matrices, and, for comparison, from the density sum and difference histograms. In order to tune the algorithm, the number of required texture classes is varied from 2 to 4. In the two cases, the largest range where the number of detected clusters remains constant appears for two classes of textures. It corresponds to a partition of the data set into four clusters (cf. Figs. 8(c-1) and 8(c-2)). With the density SDH based texture features, the largest range of S where the number of detected cores remains constant is [13-38] (Fig. 8(c-2)). It is slightly larger than that obtained with the COM texture features, which is [26-50] (Fig. 8(c-1)).
Figs. 8(d-1) and 8(d-2) show the discrete data sets obtained for S = 38, which is the middle of the range associated with the co-occurrence features, and for S = 26 when the features are extracted from density sum and difference histograms, respectively. The four cores, detected as domains of homogeneous textures, are displayed in Fig. 8(e-1) and 8(e-2). The texture features extracted from the COM are f_1, f_4, f_3 and those extracted from the density SDH are f_1, f_9, f_5 .

The result of the classification is shown in Fig. 8(b). Table 5 summarizes the statistics of the four detected clusters. The performance of the classifier is measured by the classification error-rate, estimated as the ratio of the number of misclassified observations to the total number of observations. The error-rates obtained with the two proposed algorithms are identical and equal to 3.15% . In this example, the classes do not overlap too much and the actual error-rate is very close to the theoretical minimum error-rate achieved by use of the Bayes decision rule associated with the true statistics of the data set, which is equal to 2.63%. The difference between these two error-rates corresponds to only five observations misclassified out of over 950.

6.2 Example 2

The major difficulties in cluster analysis are with non spherical clusters, bridges between clusters and non linearly separable clusters. The bivariate data set of Fig. 9(a) has been generated keeping these well-known difficulties in mind. It is composed of three populations of 1000 data points each drawn as:

$$x_1 = A_1 \cos\Theta + B_1$$

$$x_2 = A_2 \cos\Theta + B_2$$

where Θ is a normal random variable with mean m and standard deviation s , and where B_1 and B_2 are normal random variables with means μ and variances σ (cf. Table 6).

For this example, the largest range of S where the three clusters have been identified is [24-50] when the co-occurrence features are used (cf. Fig. 9(c-1)), while it is [24-46] for the features extracted from density SDH (cf. Fig. 9(c-2)). Figs. 9(d-1) and 9(d-2) show the discrete data sets obtained for S=37 and S=35 respectively, i.e. the middles of these ranges that are very similar.

The three detected cores are displayed in Fig. 9(e-1) and 9(e-2). Two texture features, namely f_2 and f_6 , have been extracted from the COM to obtain the two cores shown in Fig. 9(e-1) and four texture features extracted from density SDH, namely f_6, f_3, f_1 and f_2 , have been selected to obtain the two cores shown in Fig. 9(e-2). The classification results achieved with the two algorithms are identical. They are shown in Fig. 9(b). The error-rate obtained with the texture clustering procedures is 1.12%, whereas it reaches 6.3% with the ISODATA algorithm [Ball & Hall, 1965]. This example shows that when central points cannot represent the clusters globally, the texture based approach, which takes into account the local properties of the distribution of the input data, performs much better than algorithms dedicated to globular clusters.

6.3 Example 3

We now present a multidimensional case, which demonstrates the ability of the procedure to identify interlaced clusters for data of higher dimensionality. The data shown in Fig. 10(a) consists of two clusters generated as circular torus formed by the rotation of a plane circular Gaussian distribution about an axis in the plane of that distribution. These two torus are interlaced as the rings of a chain.

The cluster cores detected by the clustering procedure based on the selected co-occurrence features f_5 and f_4 with two texture classes and with S=34, which is the middle of the [17-50] largest range where the number of detected clusters remains constant, are presented in Fig. 10(c). Figure 10(d) shows the two cluster cores detected with the features f_9, f_6, f_1, and f_8 extracted from the density sum and difference histograms with two texture classes and with S=32, i.e. the middle of the [14-50] largest range where the number of detected clusters remains constant. The classification results achieved with the two algorithms are identical. They are shown in Fig. 10(b). The error-rate associated with the two texture clustering procedures is 0.1% whereas it reaches 12.17% with the ISODATA algorithm. This result demonstrates the effectiveness of the approach in a non trivial situation.

7. Computational load

The proposed texture clustering algorithms are based on the same 3 steps scheme:
1. Data conditioning
2. Texture characterization
3. Clustering based on texture properties.

In the first data conditioning step, the distribution of the data points is approximated by the discrete multi-dimensional histogram constituted of S^N cells. Thanks to the fast algorithm proposed in [Postaire & Vasseur, 1982], the number of elementary operations required by this procedure is NQ .

In the last clustering step, the sampling points where the local underlying texture is evaluated are first assigned to different texture classes using the K-means algorithm that requires RKt operations, where R is the number of non-empty hypercubes, K is the number of texture classes and t is the number of iterations necessary for the algorithm to converge.

Then, the connected components are extracted by means of an aggregation procedure where two hypercubes that belong to the same class of texture are assigned to the same component if they have at least one point in common. As $(3^N - 1)$ adjacent neighbors of each sampling point are considered, $R3^N$ operations are required by the connected components extraction procedure.

The core extraction procedure requires the determination of the compactness of all the detected connected components, involving also $R3^N$ elementary operations. Using a hypercubic structuring element of side length equal to 3, the classical closing-opening morphological filtering process requires $4R3^N$ operations.

We now focus our attention on the complexity of the second step since the first and third steps are independent of the texture features extraction process. This second step, which consists in the characterization of the distributions in terms of texture, is split into two phases. The co-occurrence matrices or the density sum and difference histograms are generated in a first one, while the texture features are extracted from the matrices or from the histograms in a second phase.

The computational loads associated with the generation of the COM and the density SDH for each non-empty hypercube are similar, and depend on the number of the couples of adjacent sampling points encountered within a hypercubic neighborhood of size length $(2h+1)$. As there are $(3^N - 1)$ adjacent sampling points for each of the

$(2h+1)^N$ sampling points falling in the hypercubic neighborhood, $(3^N-1)(2h+1)^N$ couples of sampling points are considered to compute the co-occurrence matrix or the density sum and difference histogram at each sampling point of the discrete multidimensional histogram. As h is set to 1 (cf. § 5.4), the number of elementary operations is approximately equal to $\left(3^N\right)\times\left(3^N-1\right)\approx 9^N$. Hence, the determination of all the co-occurrence matrices or the density sum and difference histograms requires $R9^N$ operations.
The second phase is significantly different for the COM based and the density SDH based algorithms. It deserves a particular attention to avoid computational burden.

7.1 Complexity of the co-occurrence matrix based algorithm

In the case of the COM, each matrix must be looped through once or twice depending on the feature to be extracted. $\left(G+1\right)^2$ operations are necessary to explore the matrix so that the total complexity of the texture characterization using the co-occurrence matrices for R non-empty hypercubes is equal to:

$$O\left(R\left((9)^N+(\alpha+\beta)(G+1)^2\right)\right).$$

where α and β are the numbers of features using 1 and 2 loops, respectively.
When the quantization level G of the density and/ or when the dimension N are large, the computation cost for computing the features becomes prohibitive. Several algorithms have been proposed in the texture analysis literature to overcome this problem. Some solutions are the reduction of the quantization level G [Clausi, 2002], the updating the features determined in a hypercubic neighborhood from those obtained in the adjacent neighborhoods [Argenti et al., 1990] or the storage of only the non-zero co-occurring density values [Clausi & Jernigan, 1998; Clausi & Zhao, 2002]. This last solution is well-adapted for large quantization levels G, i.e. when the co-occurrence matrices become large and sparse. We have used a hybrid data structure which combines a linked list and hash tables [Clausi & Zhao, 2002] to avoid the storage of the pairs of values of the co-occurrence matrices that have zero probability. This data structure is called hereafter the Hybrid Co-occurrence Matrix (HCM).
Each node of that linked list is a structure containing one of the pairs of co-occurring values effectively encountered in the hypercubic neighborhood, its probability of co-occurrence for neighboring sampling points and a link to the next node in the list. To include a new pair in a linked list, a node having the same pair of density values is searched. If such a node is found, then its probability is incremented. Otherwise, a new node is added at the end of the list. However, the search of a particular node is time consuming. To avoid this drawback, we use a hash table with the same size than the co-occurrence matrix, in order to give a direct access to each node of the linked list. The access to the hash table is provided by the pair of density values (*i,j*). Each entry in the hash table contains a pointer. If the pointer is null, then the particular co-occurring pair of density value (*i,j*) does not have a representative node on the linked list. In this case, a new node is created and inserted at the end of the linked list. If the pointer is not null, then it points to the existing corresponding node in the linked list and its probability is incremented.
The length L of the linked list is equal to the number of distinct pairs of values found in the considered hypercubic neighborhood. A total of $R\left((9)^D+\left(\alpha+\beta\right)L\right)$ operations are

required to calculate the texture features for all the sampling points. The value of L, depends on the data structure, on the dimension N and on the discretization parameter S. As L is generally significantly smaller than $\left(G+1\right)^2$, the computational load to determine the texture features can be greatly reduced by means of the HCM.

7.2 Complexity of the density sum and difference histogram based algorithm

Let us now consider the algorithm based on the density sum and difference histograms that must be looped through once or twice to extract one feature. As the histograms are one-dimensional structures, they are explored in $\left(2G+1\right)$ operations and the resulting complexity of the whole characterization procedure is equal to:

$$O\left(R\left((9)^D+(\alpha+\beta)(2G+1)\right)\right).$$

As $\left(2G+1\right)\prec\left(G+1\right)^2$, it appears that the complexity of the density SDH based algorithm is significantly smaller than that of the COM based algorithm, especially for high values of G. However, the comparison between the complexities of the density SDH and HCM based algorithms is not easy since the value of L depends on many parameters. The complexity of the density SDH based algorithm is smaller than that of the HCM based algorithm only if $\left(2G+1\right)\prec L$.

7.3 Processing times comparison

In order to compare the processing speeds produced by the feature extraction procedures based on the COM, the HCM and the density SDH , we use data sets constituted of two well separated Gaussian distributions of observations with means μ =[2, 2, 2,…, 2]^T and μ =[-2, -2, -2,…, -2]^T and with unit covariance matrices $\Sigma_1=\Sigma_2=I_N$. For N-dimensional data, μ_1 and μ_2 are N-dimensional vectors, while Σ_1 and Σ_2 are $N\times N$ unit covariance matrices.

Since the main purpose of these simulations is to compare the computation times of the texture characterization procedures, the tuning of these algorithms is not optimized as proposed in section 5. On the contrary, all runs are made with $S=10$ and with the largest number of available texture features for each algorithm. This strategy allows running the feature extraction procedures under comparable conditions for different dimensionalities N of the data and different sample sizes Q. For the density SDH based algorithm, we compute the 9 available features of table 2. For a fair comparison, we have selected the 9 most discriminatory features among the 13 that can be extracted from the COM and HCM (Table 1).

As the number of non-empty hypercubes depends on the structure of the data distribution, we have determined the average computation time per non-empty hypercube. Table 7 indicates these computation times for twelve data sets obtained with three different sample sizes (Q=1000, 5000 and 10 000) and for N varying from 2 to 5, using a Pentium M processor 715A/ 1.5GHz PC with 512 Moctets memory. Although the running times are computer dependent, they give an idea of the computation time improvement in a non trivial case.

We indicate, for each data set, the number R of non-empty hypercubes and the maximum value G of the densities. As the number of sampling points increases with the dimensionality N, the number of non-empty hypercubes is an increasing function of N for a given number Q of data points. As a consequence, the number of data points in each non-empty hypercube tends to decrease with increasing values of N , so that G is a decreasing function of N. The mean value of the lengths L of the linked lists produced by the HCM based procedure is also indicated in table 7. This mean value is denoted $\overline{L}$.
As expected, the processing times for the generation of the COM and the density SDH are similar. They increase with the dimensionality N and are independent of the number Q of data points.
Table 7 allows to compare the processing times of the feature extraction process from the COM, the HCM and the density SDH for different couples of values of Q and N.
The procedure based on the HCM is always faster than that based on the COM. The speed improvement is important when the value G is high and the mean value $\overline{L}$ of the lengths of the linked lists is low. For example, with Q =10 000 and with the lower dimension N =2, G reaches the value 869, $\overline{L}$ is equal to 63 and the extraction of the features from the HCM is more than 6000 times faster than the extraction from the COM. On the opposite, the improvement of the processing times is less significant for lower values of G and higher values of $\overline{L}$. For the same number Q =10 000 and a higher dimension (N =5), the maximum value G is equal to 61 and $\overline{L}$ reaches the value 804. In this case, the speed with the HCM is only twice faster than that with the COM (cf. table 7).
The procedure based on the density SDH is also always faster than that based on the COM. The larger the maximum value G, the more important is the speed improvement. For example, the extraction of the features from the density SDH is more than 400 times faster than the extraction from the COM when G reaches the value 869, i.e. with Q =10 000 and with the lower dimension N =2. For the same number Q =10 000 and a higher dimension (N =3), the maximum value G is equal to 415 and the speed with the density SDH is only 100 times faster. But, even for G=61, a significantly lower value corresponding to N=5, the speed remains more than 10 times faster with the feature extraction procedure based on the density SDH than that based on the COM.
If we compare the density SDH with the HCM based procedures, we can show that these two procedures provide comparable computation times. When the average value $\overline{L}$ of the lengths of the linked lists is smaller than the size $(2G+1)$ of the SDH, the extraction of the features from the HCM becomes faster than the extraction from the density SDH. For example, with Q=10 000, and with the dimension N =2, the extraction of the features from the HCM is more than 10 times faster than the extraction from the density SDH. On the contrary, for the same example and a higher dimension (N =5), the density SDH is more than 5 times faster than the HCM based procedure.

8. Conclusion

After a series of adaptations of classical image processing tools to cluster analysis such as thresholding, edge detection, relaxation, Markov field models and mathematical morphology, this chapter shows how texture analysis concepts can be introduced in the field of pattern classification. A general-purpose clustering procedure has been presented,

based on multidimensional texture analysis. The basic idea behind this approach is the characterization of the local distribution of the data points in the multidimensional data space in terms of textures. A set of texture features extracted from co-occurrence matrices or density sum and difference histograms that accumulate spatial and statistical information is used to evaluate locally the multidimensional textures that characterize the data distributions. The clustering scheme is based on the classification of texture feature vectors rather than on a direct processing of the observations themselves in the data space. Experimental results show that the density SDH and the COM based clustering algorithms are nearly as accurate in terms of error rates. However, the processing time using the COM tends to be prohibitive, especially for large data sets. This time processing can be greatly reduced by means of an hybrid structure including a linked list associated with hash tables.
The main advantage of sum and difference histograms for clustering is a substantial reduction in computation time and memory requirement without any loss of accuracy of the results.
When the texture based clustering procedures are compared with classical classification schemes for globular clusters detection, they perform comparably well. However, the new procedures are much more efficient in difficult clustering situations such as non spherical or non linearly separable clusters since they are sensitive to the local characteristics of the observation distributions.

9. References

Argenti, F. Alparone, L., Benelli, G., 1990. Fast algorithms for texture analysis using co-occurrence matrices, IEE Proc. 137 Pt. F, 6, 443-448.

Ball, G.H., Hall, D.J, 1965. ISODATA, a novel method of data analysis and pattern classification. NTIS rep. AD699 616 Stanford Res. Inst. Stanford CA.

Benslimane, R., Botte-Lecocq, C., Postaire, J.-G., 1996. A watershed algorithm for cluster analysis by mode detection. J. Européen des Systèmes Automatisés, RAIRO-APII series, 30, 9, 1169–1200.

Botte-Lecocq, C., Postaire, J.-G., 1991. Iterations of morphological transformations for cluster separation in pattern recognition. Symbolic-Numeric Data Analysis and Learning, Nova Science Pub., New-York, pp. 173-185.

Botte-Lecocq, C., Postaire, J.-G., 1994. Mode detection and valley seeking by binary morphological analysis of connectivity for pattern classification, in: Schade, M., Bertrand, P.,

Botte-Lecocq, C., Hammouche, K., Moussa, A. Postaire, J.-G., Sbihi, A. and Touzani A., 2007. Image processing techniques for unsupervised pattern classification. In Vision Systems : Scene Reconstruction, Pose Estimation and Tracking, Rustam Stolkin Eds., I-Tech, Vienna, Austria, Chap. 25, pp. 358-378.

Clausi, D.A., Jernigan, M.E., 1998. A fast method to determine co-occurrence texture features. IEEE Trans. Geosci. Remote Sensing, 36, 1, 298-300.

Clausi, D.A., Zhao, Y., 2002. Rapid co-occurrence texture features extraction using a hybrid data structure. Computers and Geosciences, 28, 6, 763-774.

Clausi, D. A., 2002. An analysis of co-occurrence texture statistics as a function of grey level quantization. Can. J. Remote Sensing, 28, 45-61.

Clausi, D. A., Zhao, Y., 2003. Grey level co-occurrence integrated algorithm (GLCIA): A superior computation method to rapidly determine co-occurrence probability texture features. Computer & Geosciences, 29, 837-850.

Comer, M.L., Delp, E.J, 1999. Segmentation of textured images using a multiresolution Gaussian autoregressive model. IEEE Trans. Image Process., 8, 408-420.

Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Info. Theory., 13, 1, 21-27.

Cross, G., Jain, A., 1983. Markov random fields texture models, IEEE Trans. Pattern Anal. Machine. Intell., 5, 25-39.

Dankner, A.J, Rosenfeld, A., 1981. Blob detection by relaxation, IEEE Trans. Pattern Anal. Machine. Intell., 3, 79-82.

Davis, L.S., 1975. A survey of edge detection techniques, Comput. Graphics Image Process., 4, 248-270.

Devijver, P.A., Kittler J, 1982. Pattern Recognition: A statistical Approach, Prentice-Hall, Englewood Cliffs, NJ.

Diday, E., Le Chevallier, Y., Burschy, B. (Eds.), New Approaches in Classification and Data Analysis, Springer-Verlag, pp. 194–203.

Eigen, D.J, Fromm, F.R., Northouse, R. A., 1974. Cluster Analysis based on dimensional information with applications to feature selection and classification. IEEE Trans. Syst. Man. Cybern., 284-294.

Filippone, M., Camastra, F., Masulli, F., Rovetta S., 2008. A survey of kernel and spectral methods for clustering, Pattern recognition, 41, 176-190.

Hammouche, K., Diaf, M., Postaire J-G., 2006. Clustering method based on multidimensional texture analysis. Pattern Recognition, 39, 1265-1277.

Haralick, R. M., Shanmugam, K., Dinstein, I., 1973. Texture features for image classification. IEEE Trans. Syst. Man. Cybern. 3, 6, 610-621.

Haralick, R.M., 1978. Statistical and structural approaches to texture, Proc. 4th IJCPR, 45-60.

Jain, A. K., Farrokhnia, F., 1991. Unsupervised texture segmentation using Gabor filters. Pattern Recognition. 24, 1167-1186.

Jain, A.K., Murty, M.N., Flynn, P.J, 1999. Data clustering: A review. ACM Computer Surveys, 31, 265–323.

Keller, J. M., Crownover, R. M., 1989. Texture description and segmentation through fractal geometry. CVGIP: Graphical Models and Image Process., 45, 150-166.

Macqueen, J, 1967. Some methods for classification and analysis of multivariate observations. Proc. 5th Symp. Math. Stat. Prob., 281-297.

Manjunath, B. S., Chellappa, R., 1991. Unsupervised texture segmentation using Markov random field models. IEEE Trans. Pattern Anal. Machine. Intell., 13, 478-482.

Moussa, A., Sbihi, A., Postaire, J-G., 2008. A Markov random field model for mode detection in cluster analysis. Pattern Recognition Letters, 29, 9, 1197-2008.

Panjwani, D.K., Healey, G., 1995. Markov random field models for unsupervised segmentation of texture color images. IEEE Trans. Pattern Anal. Machine. Intell., 17, 939-954.

Parzen, E. 1962. On estimation of a probability density function and mode, Ann. Math. Statist., 33, 1065-1076.

Porter, R., Canagarajah, N., 1996. A robust automatic clustering scheme for image segmentation using wavelets. IEEE Trans. Image Process., 5, 662-665.