

Texture Classification Based on Co-occurrence Matrix and Neuro-Morphological Approach

Mohammed Talibi Alaoui¹ and Abderrahmane Sbihi²

¹ Laboratoire de Recherche en Informatique, LARI, Université Mohamed I,
FSO, BP. 717, 60000, Oujda, Morocco
m.talibialaoui@fso.ump.ma

² Laboratoire LTI, ENSA, Université Abdelmalek Essaadi, Tanger, Morocco
sbihi@ensat.ac.ma

Abstract. This article proposes a hybrid approach for texture-based image classification using the gray-level co-occurrence matrices (GLCM), self-organizing map (SOM) methods and mathematical morphology in an unsupervised context. The GLCM is a matrix of how often different combinations of pixel brightness values (grey levels) occur in an image. The GLCM matrices extracted from an image are processed to create the training data set for a SOM neural network. The SOM model organizes and extracts prototypes from various features obtained from the GLCM matrices. These prototypes are represented by the underlying probability density function (pdf). Under the assumption that each modal region of the underlying pdf corresponds to a one homogenous region in the texture image, the second part of the approach consists in partitioning the self-organizing map into connected modal regions by making concepts of morphological watershed transformation suitable for their detection. The classification process is then based on the so detected modal regions. We compare this approach to other texture feature extraction using fractal dimension.

Keywords: Image Processing, Texture, Clustering, Co-occurrence Matrix, Self-Organizing Map, Watershed Transformation.

1 Introduction

Texture is an important feature of objects in an image. The perception of texture is believed to play an important role in the human visual system for recognition and interpretation. There has been a great interest in the development of texture based pattern recognition methods in many different areas, especially in the areas of industrial automation, remote sensing and medical diagnosis [1].

Texture classification passes through the difficult step of texture representation or description. What is seen as a relatively easy task to the human observer becomes a difficult challenge when the analysis is made by a computational algorithm. How can we copy the human brain in its capability to analyze, classify and recognize textures?

Putting aside these questions about human brain workings, and focusing mainly on the necessity of how to describe a texture from its content, different approaches and models have been proposed [2].

Several authors have worked in finding descriptors and features for texture identification. Existing features and techniques for modeling textures include Gabor transforms [3], bidirectional texture function [4], local binary patterns [5], wavelet-based features [6], fractal dimension [7], Haralick features [8], and some invariant feature descriptors such as Zernike moments [9].

Among these, Haralick features are the most widely used. In his work, Haralick suggested the use of gray-level co-occurrence matrices (GLCM) to extract texture features from an image. Since then, GLCMs became widely used for image texture features extraction in many types of applications.

The focus of this paper is to show the effectiveness of the hybrid approach in handling and clustering from a texture image. The Performances are evaluated and compared in term of classification error. In order to test our choice of the GLCM matrices as a texture feature extraction methods, we compare it with a fractal extracted features [10], [11].

The rest of the paper is organized as follows. In section 2, we propose our approach for texture classification. We first calculate features extracted from the GLCM descriptor, and then place the feature vector of each pixel into the feature space which forms a cloud of observations. To help discover the different classes present in this cloud of N-dimensional observations, we propose as first treatment phase, the projection of these observations on a two-dimensional map (SOM) [12]. The information in each cell of this map is represented by the probability density function value (pdf) estimated by a nonparametric procedure [13], from the distribution in multidimensional space of weight vectors resulting from the learning of the neural network. Under the assumption that each regional maximum of the pdf [14], [15] is a modal area of this function, which corresponds to a homogenous region in the image, the second step of treatment consists in extracting automatically regions of the pdf modal regions into connected and individualized components. This extraction is based on the exploitation of the watershed technical, generally used for segmentation of digital images using mathematical morphology [11][15][16].

In section 3, the classification phase is to take the weight vectors corresponding to the modal regions detected as prototypes of homogeneous regions in the image. Weights from each of these prototypes are the basis of the assignment of any pixel of the texture image to one of the classes extracted. In section 4, the experimental results using the proposed approach are shown and the performance is discussed. Section 5 gives the concluding remarks.

2 The Proposed Approach for Texture Classification

Every classification process begins with an acquisition step of observations which consists in determining relevant attributes that characterize better the objects. The sample of observations is constituted by features extracted from GLCM matrices of a texture

image. These (GLCM) matrix [8] describes the relative frequencies with which two pixels separated by a distance d under a specified angle occur on the image.

Four main modules are the building blocks of the proposed method: (i) Self-Organising feature map, (ii) Texture descriptor and learning process, (iii) Clustering, and (iv) Classification.

2.1 Self-Organising Feature Map (SOM)

Let's $\Lambda = \{X_1, X_2, X_3, \dots, X_Q\}$ be a sample of Q observations in a N -dimensional space such as $X_q = [x_{q,1}, x_{q,2}, \dots, x_{q,n}, \dots, x_{q,N}]^T$, $q = 1, 2, \dots, Q$. The Kohonen network is made of a two layers [12]. The first one, the input layer is composed of N neural units representing the N attributes of the observation X_q . The output layer is composed of M neural units regularly distributed on the map which elaborates prototypes of the data (cf. Figure 1). The neural units of the first layer are connected to the units of the second layer. Each interconnection from an input unit j to an output unit m has a weight $W_{m,j}$. That means that each output unit m has a corresponding weight vector $W_m = [W_{m,1}, W_{m,2}, \dots, W_{m,n}, \dots, W_{m,N}]^T$ (cf. figure 1).

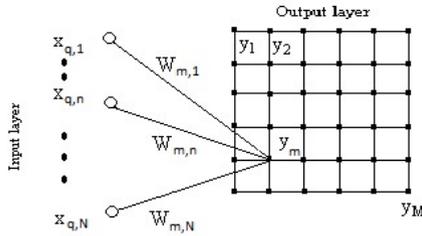


Fig. 1. Kohonen Network

Each neural unit in the output layer is assigned with a specific position and a weight vector. When an input $X_q(t)$ is presented to the network, the neural unit whose weight vector is the closest to this observation wins the competition and is allowed to learn it even better. The neural units of the second layer are so interconnected to elaborate the winning neural units by inhibiting the other units.

The winning neural unit and its neighbours are updated. The size of the neighbourhood is decreased as the training goes on. The weight vector of this winning unit, noted m^* , and its neighbours m are modified according to equation :

$$W_m(t) = W_m(t-1) + a(t)h(m^*,t) \cdot [X_q(t) - W_m(t-1)] \text{ if } m \in V(m^*, r(t)) \quad (1)$$

where :

- m^* is the winning unit defined by :

$$m^* = \text{Arg min}_m [d(X_q(t), W_m(t))] \quad (2)$$

- $d(X_q(t), W_{m'}(t))$ is the Euclidean distance between the observation $X_q(t)$ and the weight vector $W_{m'}(t)$ of the unit m' in the output layer.
- $r(t)$ is the interaction radius which depends on the number t of the iteration.
- $a(t)$ is the learning coefficient at the time t .
- $V(m, r)$ is the neighbourhood of a neural unit m with a radius r , defined by :

$$V(m, r) = \{m' \in [0, M[, m' \neq m / d(U_m, U_{m'}) \leq r\} \quad (3)$$

- $d(U_m, U_{m'})$ denotes the Euclidean distance between the position vectors U_m and $U_{m'}$ of the m and m' neural units.
- $h(m^*, t)$ is the interaction function that depends on the proximity radius $r(t)$ defined by :

$$h(m^*, t) = \exp\left(-\frac{d(U_m, U_{m'})^2}{2r(t)^2}\right) \quad (4)$$

2.2 Learning Process and Application to Texture Image Classification

Texture Descriptor Module. This module has two main tasks. First it computes the GLCM for image. Then it processes the GLCM matrix in order to obtain a vector features which, in turn, will constitute the input data for the clustering module.

The gray co-occurrence matrix (GLCM) is a traditional statistical method for texture analysis [8]. We are interested in this second order method since it characterizes the relationship between the values of neighboring pixels.

A Gray level co-occurrence matrix $P_d(i, j)$ is a square matrix, which describes the relative frequencies with which two pixels separated by a distance d under a specified angle occur on the image, one with graytone i and the other with graytone j . Such matrices of graytone spatial dependence frequencies depend on the angular relationship between neighboring pixels and on the other distance between them. The GLCM can be defined as:

$$P_{d,\theta}(i, j) = \Pr(I(p_1) = i \wedge I(p_2) = j \wedge \|p_1 - p_2\| = d) \quad (5)$$

Where P is the probability, p_1 and p_2 are positions in the gray scale image I and θ usually take the range of values: $0^\circ, 45^\circ, 90^\circ,$ and 135° . The algorithm schema describes how the texture descriptor module works:

1. Browsing image by a sliding window of size $k * k$. The choice of k depends on the image being processed.
2. The GLCM matrices contain a wealth of very important information and are therefore difficult to handle. Thus, fourteen indices (defined by Haralick) which correspond to descriptions of characters textures can be calculated from these matrices [8]. In our case study, we only use and compute the following five main Haralick's coefficients on GLCM for texture analysis and the classification: Contrast (f_1), Entropy (f_2), Energy (f_3), Correlation (f_4) and Homogeneity (f_5) parameters. The correlation feature based on the variance (σ) and Mean parameter (μ). The coefficient σ is specified by the following equations:

$$\sigma = \sqrt{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (1-\mu)^2 .P_d(i, j)} \quad , N_g = \text{Number of gray levels} \tag{6}$$

$$\mu = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i.P_d(i, j)$$

Next, we calculate GLCM and the mean values in four directions. Therefore we extract the Haralick's parameters from GLCM matrix and then assign the pixel center of the window [8][17].

3. Each pixel is specified by a feature vector $X_q(t) = \{f_1, f_2, f_3, f_4, f_5\}$
4. In this module, the feature vector $X_q(t)$ is used as input data for SOM algorithm.

The SOM algorithm, each node contains a vector of weights of the same dimension as the input vectors [12].

The Haralick's coefficients are specified as following [8]:

The Contrast (f_1) feature is a measure of image intensity contrast or the local variations present in an image to show the texture fineness. This parameter is specified by the following equation:

$$f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (1-j)^2 .P_d(i, j) \tag{7}$$

The Entropy coefficient (f_2) is a descriptor of randomness produces a low value for an irregular GLCM. It achieves its highest value when all elements of the GLCM are equal for an irregular image. Furthermore, an irregular GLCM is produced by a

regular image, such as one with a repetitive pattern. This coefficient is defined by the following expression:

$$f_2 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_d(i, j) \cdot \ln[P_d(i, j)] \tag{8}$$

The Energy feature (f_3) returns the sum of squared elements in the GLCM as expressed by the following equation :

$$f_3 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_d(i, j)^2 \tag{9}$$

The descriptor Correlation (f_4) measures the linear dependence of gray level values in the co-occurrence matrix or describes the correlations between the rows and columns of the co-occurrence matrix. This parameter is specified by the following equation:

$$f_4 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)(j - \mu_y) \cdot \frac{P_d(i, j)}{\sigma_x \sigma_y} \tag{10}$$

Homogeneity (f_5) feature returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. This coefficient is specified by the following equations:

$$f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_d(i, j) / (1 + |i - j|) \tag{11}$$

All these features parameters are defined for a value of d . The choice of this parameter is very important in obtaining a quality result.

Visualization of the Pdf on the SOM. In the learning phase, the observations $X_q(t)$ are presented sequentially one by one to the network randomly. This first step of the process concerns the self-organizing and the learning of the network which permit to represent the SOM map. The proposed procedure is based on numerical mathematics morphological operators; we propose so to take a digital representation on the SOM. Thus, the information from the learning process is represented by the underlying probability density function (pdf). For this purpose, we use the non-parametric Parzen estimate defined by [18]:

$$\hat{p}(W_m) = \frac{1}{Q} \cdot \sum_{q=1}^Q \frac{1}{V[D(W_m)]} \cdot \varphi\left(\frac{W_m - X_q}{h_Q}\right) \tag{12}$$

With $\varphi(X) = \frac{1}{\sqrt{2\pi}} \cdot \exp(-\frac{1}{2} X^T \cdot X)$

$D(W_m)$ is the domain estimation. When it corresponds to an hypersphere with radius h_Q , and centered at the point defined by W_m . Its volume $V[D(W_m)]$ is given by:

$$V[D(W_m)] = \frac{\pi^{N/2}}{\rho(\frac{N}{2} + 1)} \cdot h_Q^N \quad \text{With } \rho(\frac{N}{2} + 1) = \frac{(N + 1)! \sqrt{\pi}}{2^{(N+1)(\frac{N+1}{2})!}} \tag{13}$$

and $h_Q = h_0 \sqrt{Q}$

The visualization of the pdf permits to display the SOM map as a digital image where each unit of the map is represented by a gray value pixel which corresponds to the pdf value.

This data projection method is a very appealing technique since it takes full advantage of the human skill for organizing the data presented to the eyes of the analyst. He has to interactively handle the data. Or, this map wins more interest if it could be exploited by automatic procedures in an unsupervised classification i.e without the intervention of the analyst in the decision. To overcome this limitation, and to give a powerful tool to detect, to extract and to determine the number of clusters from the SOM map, we propose to apply the watershed morphological transformation. The following step concerns the problem of modal regions in the SOM map.

2.3 Modal Regions Extraction with Watershed Technical

This principal phase of the procedure consist in the localization of the modal regions of the underlying pdf by means of the watershed algorithm based on homotopic thinnings of the function [11] [15][16].

Prior to mode detection, some kind of pre-processing is needed to enhance significant local variations of the density function. Opening operation, tends to smooth the function by filling up small holes and removing insignificant peaks in the function, while preserving the global shape of the function [11] [16].

As we use the watershed approach, which is well suited for determining the catchment basins corresponding to the regional minima of a function [11], we introduce the additive inverse $f(W_m)$ of the function $p(W_m)$. Thanks to this simple transformation, the maxima of $p(W_m)$ become minima of $f(W_m)$.

The watershed of a function can be constructed through consecutive homotopic thinnings of this function. The homotopic thinning is a transformation commonly used in mathematical morphology for image skeletonization [11][15][16]. In the watershed graph, modal regions of the pdf that are homotopic and geodesic extensions of regional maxima are separated by lines and are easily extracted in connected components through a simple morphological transformation.

3 Classification Module

Once the different modal domains of the underlying pdf are identified and extracted as connected components, the observations corresponding to them are used as prototypes in the raw space. Each pixel in the texture image can finally be assigned to the clusters attached to their nearest neighbour among the prototypes in the modal domains detected [11].

CLASSIFICATION ALGORITHM SCHEMA:

1. Compare $X_q = [x_{q,1}, x_{q,2}, \dots, x_{q,n}, \dots, x_{q,N}]^T$ with all the node vectors m_d ($d = 1, \dots, M$) of the SOM using the Euclidean distance.
2. Obtain an ordered list of all best matching units (BMUs). The BMU is the winning node of the SOM.
3. Determine the first BMUs in the SOM.
4. Affect the prototype associated with the selected BMU(s) to X_q .
5. Repeat the step 1.

4 Experimental Results

This section presents experiments that evaluates the proposed method and compares its performance with other methods in literature.

In order to assess the performance of the proposed hybrid approach, experiment with VisTex texture image [20], is carried out. Texture image have 200×200 pixels, with 8 bits/pixel (cf. Figure 7).

4.1 SOM Map Size

The adaptation of the parameter of the map size is based on the concept of stability of the number of modes extracted from this map. When the sample of observations to be

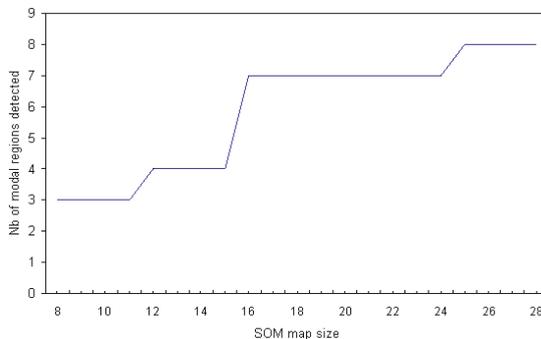


Fig. 2. Varying the number of modal regions extracted based on the map size

analyzed really consists of several different modes where the number is unknown a priori, the modes of the classes should indeed appear for a big interval of the values of the map size. Consequently, to adapt this parameter, we have analyzed the results obtained for a big margin of its variation from 8×8 to 30×30 (cf. figure 2). It was proved that the choice of this parameter in the middle of the big interval of stability in which the number of modes discovered constant rest is a good procedure to optimize an important number of algorithms of classification [19]. According to this figure, we use a map with 20×20 size.

4.2 Opening Smoothing and Watershed Transformation

The visualization of the pdf estimated with $h_0 = 0.0035$ is displayed in figure 3. Figure 4 shows the binary representation estimate of the pdf on the SOM map. We can observe that after opening operation, the map is constituted by five regions where the pdf presents high values, separated by valleys where the pdf presents low values (cf. Figure 5). We consider that a region is a set of connected pixels in the map with relatively high values of the pdf.

Figure 6 shows the graph of the function resulting from the watershed transformation. In the resulting graph, modal regions of the pdf that are homotopic and geodesic extensions of regional maxima are separated by lines and are easily extracted in connected components through a simple morphological transformation. We can observe that the map is constituted by seven regions and not five. This is due to the fact that watershed is a local analysis procedure and remains sensitive to any irregularity in the pdf and risk to divide a homogeneous region into subregions.

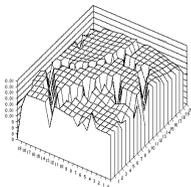


Fig. 3. Pdf graph

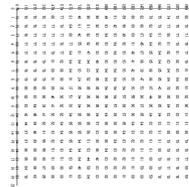


Fig. 4. Estimate of the pdf on the SOM Map

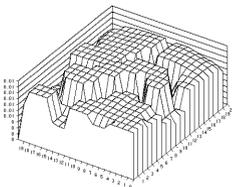


Fig. 5. Opening Smoothing Pdf

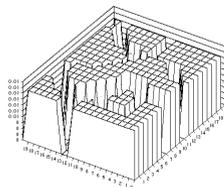


Fig. 6. Pdf modal regions extracted

This is however, an advantage in the type of analysis proposed here especially for complex real images since we not start from the assumption that any modal pdf region represents a class in the image, but rather the assumption that any modal region represents a homogeneous region in the image, which can be a class or subclass in the image. The same result can be represented by a binary representation of the SOM map, where modal regions of the pdf are well represented by an individualized connected component.

4.3 Classification Results

The classification image is displayed in figure 8. The error rate achieved by the procedure is equal to 3.56 %, while the K-means error rate is equal to 3.28 % [21]. By using the fractal dimension, as other texture feature extraction method [11], we can observe the classified image result in figure 9. The error rate achieved is 41.5%.

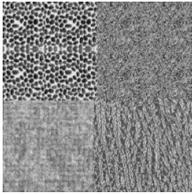


Fig. 7. Original Image

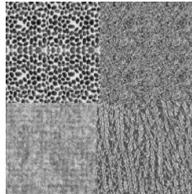


Fig. 8. Classified Image with GLCM features

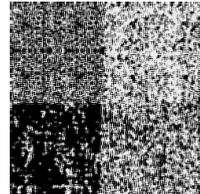


Fig. 9. Classified Image with Fractal features

Table 1.

Texture image Size	SOM map Size	Learning Time process (with 5 HC) (in minutes)	Classification Time (in seconds)	Learning Time process (with 3 HC) (in minutes)	Classification Time (in seconds)
512×512	30×30	25	14.5	12.76	6
256×256	25×25	20	12.6	10.25	5.04
200×200	20×20	12	5.4	5.03	2.19
128×128	14×14	4.35	1.8	1.85	0.75
64×64	10×10	0.3	0.6	10 Seconds	0.24

Otherwise, when we use the fractal feature as another texture feature extraction method, the time consumed during learning process decrease in a ratio of more than 65%.

We can observe, when using only three parameters (Correlation, Entropy and Contrast) of the Haralick's Coefficients (HC), the time consumed during learning process was decreased in a ratio of more than 50%, in a machine with 4 GO RAM, and a processor Intel Core i5. Table 1, shows and compare the learning time and the classification time for our texture image in many sizes and, for each size a corresponding map size:

5 Conclusion and Perspectives

In this work, we proposed an approach to unsupervised classification of textured image using the gray-level co-occurrence matrices (GLCM), with the combination of an algorithm of Mathematical morphology in a SOM map.

This approach shows that in an unsupervised context, the tools of mathematical morphology associated with the SOM map allow a good automatic classification of a texture image without using any thresholding procedure even if in the case of overlapping features.

Using GLCM features gave a good result in classification when compared with the fractal features. Fractal features used always for medical image to model the heterogeneity of the pixel distribution. We can conclude that for each texture image, we can use a texture feature extraction method different to another. This depends on its attributes and internal texture and is the analyst to decide which method to use.

Otherwise, the experimental results demonstrate that only the Contrast, Entropy and Correlation of the Haralick's texture parameters allow classifying the texture image which reduces the calculation time generated by our approach.

References

1. Li, S., Kwok, I.T., Zhu, H., Wang, Y.: Texture classification using the support vector machines. *Pattern Recognition* 36(12), 2883–2893 (2003)
2. Barcelos, C.A.Z., Ferreira, M.J.R., Rodrigues, M.L.: Texture image retrieval: A feature-based correspondence method in fourier spectrum. In: *Third International Conference on Advances in Pattern Recognition*, pp. 424–433 (2005)
3. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. *Pattern Recognition* 24(12), 1167–1186 (1991)
4. Kautz, B.S., Durand, F.: Interactive editing and modeling of bidirectional texture functions. *ACM Transactions on Graphics* 26(3), 53 (2007)
5. Zhao, G., Pietikainen, M.: Local binary pattern descriptors for dynamic texture recognition. In: *ICPR 2006: Proceedings of the 18th International Conference on Pattern Recognition*, pp. 211–214 (2006)
6. Charalampidis, D., Kasparis, T.: Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Trans. Image Process.* 11, 825–837 (2002)
7. Varma, M., Garg, R.: Locally invariant fractal features for statistical texture classification. In: *International Conference on Computer Vision*, pp. 1–8 (2007)
8. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. *IEEE Transactions on In Systems, Man and Cybernetics* 3(6) (1973)

9. Khotanzad, A., Hong, Y.H.: Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(5), 489–497 (1990)
10. Chaudhuri, B.B., Sarkar, N.: Texture Segmentation Using Fractal Dimension. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 17(1) (1995)
11. Talibi Alaoui, M., Sbihi, A.: Fractal Features Classification for Texture Image Using Neural Network and Mathematical Morphology. In: *Proceeding of World Congress on Engineering*, London, UK, Vol. I, pp. 286–294 (2012)
12. Kohonen, T.: *Self-Organisation and Associative Memory*, 3rd edn. Springer, New York (2001)
13. Asselin de Beauville, J.P.: l'estimation des modes d'une densité de probabilité multidimensionnelle. *Revue statistique et Analyse des Données* 8(7), 16–40 (1983)
14. Beucher, S.: Extrema of Gray Tone Fonctions and Mathematical Morphology. *Rapport du C.G.M.M., Ecole des Mines, Fontainebleau, n°793* (1983)
15. Beucher, S., Lantuejoul, C.: Use of Watersheds in Contour Detection. In: *Int. Workshop on Image Processing, CETT/IRISA, Rennes* (1979)
16. Beucher, S.: Segmentation Tools in Mathematical Morphology. In: *Image Algebra and Morphological Image Processing, SPIE*, vol. 1350, pp. 70–84 (1990)
17. Rouquet, C., Chausse, F., Chapuis, R., Bonton, P.: Segmentation non supervisée d'images de scènes routières, Une approche multi-critère. *TS. Traitement du Signal* 13(3), 196–208 (1996)
18. Parzen, E.: On Estimation of a probability density function and mode. *Ann. Math. Stat.* 33, 1065–1076 (1962)
19. Sbihi, A., Postaire, J.G.: Mode Extraction by Multivalued Morphology for Cluster Analysis. In: *Gaul, W., Pfeifer, D. (eds.) From DATA to Knowledge: Theoretical and Practical aspects of Classification*, pp. 212–221. Springer, Berlin (1995)
20. <http://vismod.media.mit.edu/pub/VisTex>
21. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, PBSPSP 1967*, pp. 281–297. University of California Press (1967)