IAENG International Journal of Computer Science, 39:3, IJCS_39_3_08

Application of a Mathematical Morphological Process and Neural Network for Unsupervised Texture Image Classification with Fractal Features

M. Talibi-Alaoui & A. Sbihi

***Abstract*—In this paper, we present a new texture image classification algorithm in an unsupervised context, which is based on both Kohonen Maps and Mathematical Morphology. As first part of the proposed algorithm, various features obtained from the fractal dimension computed using differential box counting method, are extracted from the texture image and then applied and projected into a Kohonen map which is 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 algorithm consists in partitioning the Kohonen 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.**

***Index Terms*— Clustering, Mode detection, Texture image, Fractal Features, Kohonen Maps, Watershed Transformation**

I. Introduction

TEXTURE segmentation partitions a texture image into disjoint regions that are sets of connected pixels which share uniform texture characteristics [1]. The approaches to image segmentation can be classified into four groups [2], namely, histogram-based techniques, neighbourhood-based segmentation, physically-based segmentation, and multidimensional data classification methods. In these approaches, most algorithms treated are based on threshold selection or in parameters adjustment which may change the segmentation results. For the multidimensional data classification methods [3], many of them are based on mode detection where there is a one-to-one correspondence between the modal regions of the pdf function and the clusters [4]. The number of clusters and the characteristics of each cluster are generally a priori unknown and are a result of the clustering method.

In a two dimensional space, the data can be examined visually as a scatter plot so that clusters can be identified without a formal mathematical description of similarity between the samples, nor a precise definition of what a cluster is. Clusters are then delineated in an interactive manner by delineating mutually exclusive regions, so that each of them contains a relatively dense concentration of data points.

Unfortunately, this task is difficult for multivariate data. The analyst faces the problem of representing the data graphically in order to highlight the presence of clusters. Due to this problem, there has been a great proliferation of clustering techniques which produce automatically classifications from initially unclassified data [5].

However, the mapping of multivariate data onto a two-dimensional graphic is also a very appealing technique since it takes full advantage of the human skill for organizing the data presented to the eyes of the analyst [6, 7]. The superiority of humans over automatic clustering procedures comes from their ability in recognizing cluster structures and curvilinear relationships between clusters, bridging clusters and all kinds of irrelevant details in the data points distribution. Hence, the mapping procedure used to transform the N-dimensional points representing the sample into points in a two dimensional space is designed to not change too much their relative positions. A human observer can usefully analyze graphic displays without conscious use of any procedure, model or rule. There are a number of methods which permit this mapping by reducing the dimension of the raw data while preserving relationships between data

Manuscript received August 23, 2012.
M. Talibi Alaoui is with the LARI Laboratory, Department of Mathematics and Computer Science, FSO, BP. 717, 60050, Oujda, Morocco (corresponding author to provide phone : 212-0536-500601; fax: 212-0536-500603; e-mail: m.talibialaoui@fso.ump.ma).
A. Sbihi is with the LIRF Laboratory, Department of physics, FSK, University of Ibn Tofail, BP. 133, 14000, Kénitra, Morocco (e-mail : sbihi@ensat.ac.ma).

elements. One of the most commonly used among them in an unsupervised context is the Kohonen self organizing feature map [8, 9, 10]. Nevertheless, this interactive classification manner necessitates the intervention of the analyst in the process.

Hence, this technique, when used alone, does not allow classifying the texture image automatically. To overcome this limitation, we propose a two step methodology. The Kohonen map obtained in the first step is the input of another automatic scheme that aims at partitioning the map into two-dimensional regions corresponding to the clusters in the data set. To be more specific, the detected clusters are automatically extracted by morphological analysis [11], which uses the watershed transformation for delineating the modal regions of the underlying probability density function computed in the 2D space of the Kohonen map.

Indeed, we propose in this paper a new texture image segmentation by a classification approach based on adaptive morphological tools associated with the Kohonen map in order to present an unsupervised clustering method that permit an automatic extraction of significant patterns from N-dimensional texture observations. The goal is to avoid any thresholding procedure in the texture image classification process.

The next section of this paper is devoted to present the data dimensionality reduction by the Self-organizing feature map (SOM) and it's learning process. As first step, and in order to get some insight into the structure of the observations sample, we propose to make a projection onto a two-dimensional plane. As a tool of projection, the Kohonen feature map is used and the resulting map is represented in our application by the underlying probability density function (pdf) estimated, by a non-parametric technique in the N-dimensional space, from the weight vectors resulting of the learning process [12, 18, 25].

In the third section, the efficiency of the proposed algorithm has been demonstrated using local fractal features calculated from the whole texture image. A morphological watershed transformation technique is presented and used in order to detect and extract the different modal regions of the pdf as individualized connected components. Once the modal regions are identified in the projection map, the observations belonging to them are used as prototypes of homogenous region in the image for the classification of the available observations. The observations are assigned to the clusters attached to their nearest neighbours among the prototypes. The performances of the proposed approach are then evaluated using a medical image.

II. Representation of the Image Texture Information on the Kohonen Map

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 fractal features of a texture image.

A. Self-Organizing Feature Map (SOM) and Learning Process

Let's $\Gamma = \{X_1, X_2, \dots, X_q, \dots, X_Q\}$ be a sample of Q observations X_q in a N-dimensional space where $X_q = [x_{q,1}, x_{q,2}, \dots, x_{q,n}, \dots, x_{q,N}]^T$, $q = 1, 2, .., Q$. The Kohonen Network is made of two layers. The first one, or input layer, receives the N attributes of the presented observation X_q. The output layer, or competitive layer, is composed of M units regularly distributed on the map (cf. Figure 1).

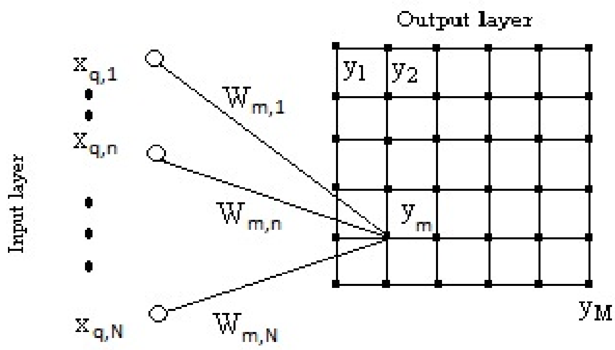

Fig. 1. Kohonen Network

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). The neural units of the second layer are interconnected to elaborate the winning neural units by inhibiting the other units. In the self-organizing feature map of Kohonen, the neural units are arranged in a lattice structure, where each unit is assigned a specific position and a weight vector. Furthermore, the neural units also organize themselves in such a way that the structural relationships between observations in the data space are captured by the lattice structure. During the training process, when an input X_q is presented to the network, the neural unit whose weight vector is the closest to this observation wins the competition and is allowed to take this input into account in the learning process. The output of the winner is then equal to 1 while the outputs of all the other output units are set to 0, such as :

$$y_m = \begin{cases} 1 \;\; if \;\; d\left(X_q(t), W_m(t)\right) \leq d\left(X_q(t), W_{m'}(t)\right) \;\; m' \neq m \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad else \end{cases} \quad (1)$$

Where $d(X_q(t),\ W_{m'}(t))$ 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.

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 and its neighbours are modified according to equations :

$$\begin{array}{ll} W_m(t) = W_m(t-1) + a(t).[X_q(t) - W_m(t-1)] & if \quad m = m^* \\ W_m(t) = W_m(t-1) + a(t).h(m^*,t).[X_q(t) - W_m(t-1)] & if \quad m \in V(m^*, r(t)) \\ W_m(t) = W_m(t-1) & if \quad m \notin V(m^*, r(t)) \quad and \quad m \neq m \end{array} \quad (2)$$

Where m^* denotes the winning unit defined by :

$$m^* = Arg\ \min_m\ [d(X_q(t), W_m(t)] \qquad (3)$$

$a(t)$ is the learning coefficient lower than 1. It is formulated so that it starts from high values at the beginning of the training process and decreases according to iterations. According to this coefficient variation, the learning process began with significant modifications of the weight vectors and finish by refining the analysis at the last iterations.

The coefficient $a(t)$ satisfies the constraints of stochastic approximations given by :

$$\textstyle\sum_t a(t) = \infty \quad and \quad \sum_t a^2(t) < \infty \qquad (4)$$

It can be an hyperbolic, exponential or linear function of t. We model it by an exponential function which decreases towards zero proximity when t increases such as :

$$a(t) = \beta_1 e^{-\beta_2} + \beta_3 \qquad (5)$$

Let T denote the number of iterations for the learning phase which is adjusted experimentally by defining a number of learning cycle. The purpose is to allow a good adaptation of the weight vectors during the learning.

By choosing the values 0.8 and 0.05 as the maximum and the minimum values of $a(t)$, respectively, we are sure that the network begins to learn with high coefficients and learns less during the last iterations. Finally, we select the parameters so that $a(t)$ reaches the value 0.1 at time (2/3)*T, which permit a correct convergence of the weight vectors during the iterations.

Thus, the coefficient $a(t)$ becomes formulated as (cf. Figure 2) :

$$a(t) = 0.75 * \exp(-(8.12) * (10^{-5}) * t) + 0.05 \qquad (6)$$

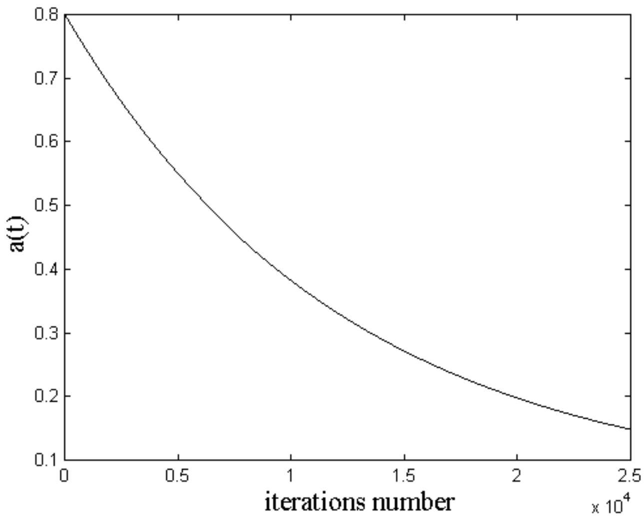

Fig. 2. Learning coefficient function

$V(m, r(t))$ is the neighbourhood of a neural unit m, with a radius $r(t)$, defined by :

$$V\big(m, r(t)\big) = \{m' \in [0, M[,\ m' \neq m \,/\ \ d^A(U_m, U_{m'}) \leq\ r(t)\} \qquad (7)$$

Where $U_m = (u,v)^T$ and $U_{m'} = (u',v')^T$ denote the position vectors on the map of the m and m' neural units.

$h(m^*,t)$ is a neighbourhood function which is used at time t to alter the step size of the m^{th} weight vector. It is a function of the Euclidean distance between its associated neural unit on the lattice and the winning unit m^* given by :

$$h(m^*,t) = \exp\left(-\frac{d^A(U_m, U_{m'})^2}{2r(t)^2}\right) \qquad (8)$$

where $r(t)$ is the radius which depends on the number t of the iteration. It is decreased every $n_r Q$ iterations, where n_r is the epoch number with a constant radius and Q is the number of observations in the sample (cf. Figure 3). Note that one epoch corresponds to one scan of the total data involved in the learning process of the network. $r(t)$ is defined such as :

$$r(t) = \begin{cases} r(t-1) - 1 \; if \;\; t\, mod(n_r Q) = 0 \;\; and \;\; r(t) > 1 \\ \varepsilon \quad if \; t\, mod(n_r Q) = 0 \;\; and \;\; r(t) \leq 1 \\ r(t) \qquad\qquad otherwise \end{cases} \qquad (9)$$

Where $x \, mod(y)$ denotes the remainder after division of x by y.

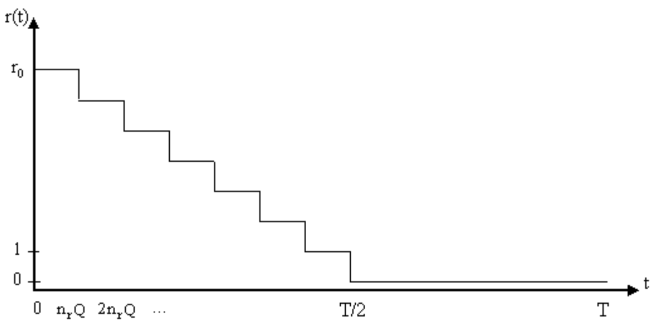

Fig. 3. Interaction radius function

In the learning phase, the observations are presented sequentially one by one to the network randomly and without putting them back to be sure that in each epoch, all the observations were “learned” by the network. We have used a square map with different initializations of the weight vectors.

At the end of the training phase, two neural units close on the map have their weight vectors close in the data space. Moreover, the weight vectors of neighboring neural units in the map converge toward the same area in the data space.

B. Application to Texture Image Classification

In this application we used a texture image (cf. Figure 14).

a. Fractal dimension

The concept of fractal is used in a large number of applications including image analysis, classification pattern recognition, segmentation etc [13]. Fractal objects have irregular shapes and complex structures that cannot be represented adequately by the traditional Euclidean dimension. The concept of fractal dimension (FD) is used as an indicator of surface roughness [14].

Of the wide variety of methods for estimating the fractal dimension that have so far been proposed, the box-counting method [13], as it can be computed automatically and can be applied to patterns with or without self-similarity [15].

The box counting method consists in partitioning the image space into square boxes of equal size. The box covers the image space of the function or pattern of interest and the number of boxes that contain at least one pixel of the function is counted. The process is repeated with different box sizes. The fractal dimension is obtained from the slope of the best fitting straight line to the graph plotting the log of the number of boxes counted versus the log of the magnification index for every stage of partitioning as shown in figure 4. For example, an image measuring size $M \times M$ pixels is scaled down to $s \times s$, where $1 < s < M/2$, and s is an integer. Then, $r = s/M$.

Fractal dimension D is given by, $D = \frac{\log(N_r)}{\log(1/r)}$ (10)

In this paper the differential box counting method is used to calculate the FD and then different fractal features are derived from this fractal dimension.

b. Differential Box Counting Method

N. Sarkar and Chaudhuri had proposed the differential box counting (DBC) method and have compared it with other conventional four methods in [16]

Consider an image of size $M \times M$ pixels. Let it be scaled down to a size $s \times s$ where $M/2 > s > 1$, where s is an integer. Then, $r = s/M$. Now consider the image to be in a 3D space with (x, y) denoting the spatial co-ordinates, while the z axis denotes the gray level. The (x, y) space is partitioned into grids of size $s \times s$. On each grid there is a column of boxes of sizes $s \times s \times s'$. Figure 4 shows the schematic for computing FD using differential box counting method.

If the total number of gray level is G, then $[G/s'] = [M/s]$. Numbers from 1, 2, .. are assigned to the boxes starting from the lowest gray level value. Let the minimum and the maximum gray level of the image in the $(i, j)^{th}$ grid fall in box number k and l, respectively. The contribution of N_r in $(i, j)^{th}$ grid is given by :

$$n_r(i, j) = l - k + 1 \qquad (11)$$

Due to the differential nature in computing n_r, this method is called differential box counting method. The contributions from all grids are found by :

$$N_r = \sum_{i,j} n_r(i, j) \qquad (12)$$

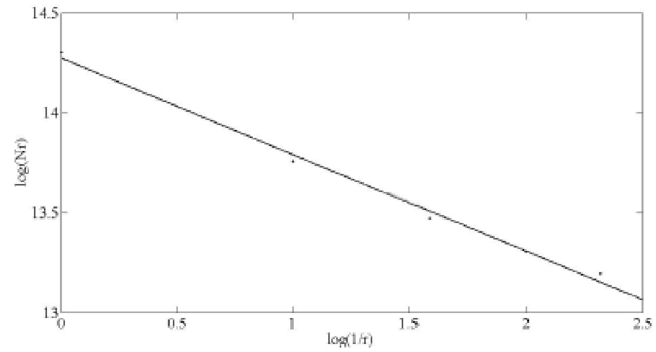

Fig. 4. Plot of $\log(N_r)$ versus $\log(1/r)$

N_r is computed for different values of s i.e. different values of r. Using equation (10), the fractal dimension can be estimated, from the least square linear fit of $\log(N_r)$ along $\log(1/r)$. The slope of the best fitting curve will give the fractal dimension. Figure 4 shows the plot of $\log(N_r)$ versus $\log(1/r)$ from which the FD is computed. A random placement of boxes is applied in order to reduce quantization effects.

c. *Fractal features*

In this paper the differential box counting method is used to calculate the FD and then different fractal features are derived from this fractal dimension which constituted the sample of observations used in the proposed approach.

Five features derived from [14, 17] based on fractal dimension are the FD of original image (f_1), high gray valued image (f_2), low gray valued image (f_3), horizontally smoothed image (f_4) and vertically smoothed image (f_5).

III. CLUSTERING ALGORITHM

A. *Principle of the Algorithm*

In cluster analysis, the existing mode detection approaches are conditioned by the adjustment of some parameters, which becomes crucial for large dimensionality data sets.

The detection of modal regions can be greatly facilitated by mapping data as a first step of the understanding process. The proposed algorithm is based on both neural network and mathematical morphology concepts.

Data projection mapping is done using a Kohonen maps (cf. § II.) represented by the underlying pdf. Modal regions of this pdf are then easily obtained by making concepts of morphological watershed transformations suitable for their detection. The weight vectors existing in the so-detected modal regions are taken as prototypes for image classification.

B. *Clustering Algorithm*

This algorithm is illustrated using a texture image. The proposed clustering algorithm consists in three basic steps :

Step 1. The Kohonen map representation

This first step of the process concerns the self-organizing and the learning of the network which permit to built the Kohonen map (cf. § II.). At the end of the learning phase, the determined weight vectors are used to estimate the underlying probability density function (pdf) in the multidimensional data space. For this purpose, we use the non-parametric Parzen estimate defined by [18] :

$$\hat{p}(W_m) = \frac{1}{Q}.\sum_{q=1}^{Q} \frac{1}{V[D(W_m)]}.\varphi\left(\frac{W_m - X_q}{h_Q}\right) \quad (13)$$

With $\varphi(X) = \frac{1}{\sqrt{2\pi}}.\exp\left(-\frac{1}{2}X^T.X\right)$ (14)

$D(W_m)$ 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 :

$$V[D(W_m)] = \frac{\pi^{N/2}}{\rho(\frac{N}{2}+1)}.h_Q^N \quad (15)$$

With $\rho\left(\frac{N}{2}+1\right) = \frac{(N+1)!\sqrt{\pi}}{2^{(N+1)}\left(\frac{N+1}{2}\right)!}$ (16)

and $h_Q = h_0\sqrt{Q}$ (17)

The parameter h_0 has a great effect on the quality of the estimation. If it is large, the small maxima of the pdf cannot be detected. Inversely, if h_0 is too small, we obtain an estimation with many non significant maxima. However, it can be expected that when true clusters exist, stable connected subsets corresponding to these clusters will appear for a wide range of values of h_0. Based on this assumption, the adjustment of h_0 is governed by the concept of cluster stability [19].

The visualization of the pdf permits to display the Kohonen map as a digital image where each unit of the map is represented by a gray value pixel which corresponds to the pdf value. The visualization of the pdf estimated with $h_0 = 0.02$ is displayed in figure 5. We can observe that the map is constituted by four regions where the pdf presents high values, separated by valleys where the pdf presents low values. We consider that a region is a set of connected pixels in the map with relatively high values of the pdf (cf. Figure 6).

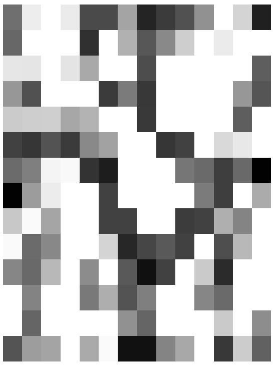

Fig. 5. Representation of the Kohonen Map

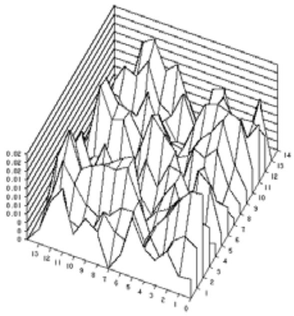

Fig. 6. Graph of the pdf

This data projection method provides a planar display of the high dimensional data set. So, it can be assumed that clusters on the Kohonen map are images of clusters in the raw multidimensional data space, and we can usually analyze graphic displays without conscious use of any analytical model of clusters, or any mathematical decision rule. However this technique, used alone, doesn't allow an automatic data classification. To automate this process, and to give a powerful tool to detect, to extract and to determine the number of clusters from the Kohonen map, we propose to apply the watershed morphological transformation. The following step concerns the problem of modal regions in the Kohonen map.

Step.2 Modal regions extraction

The technique proposed here is designed to detect modal regions of the pdf resulting from the projection process on the Kohonen map, by means of a watershed transformation which constitutes one of the most powerful tools for contour detection and image segmentation provided by Mathematical Morphology [20]. The proposed algorithm for modal regions detection which is based on this transformation consists in two basic phases.

Phase 1 : Preprocessing and watershed determination

Prior to mode detection, some kind of pre-processing is needed to smooth the density function by filtering out its small non significant variations (cf. Figure 6). Among several methods for filtering the pdf, the raw estimate is smoothed by a numerical morphological opening. This transformation is a combination of the two basic numerical morphological transformations, which are the numerical dilatation and erosion [21, 22, 26].

With the flat 3×3 structuring element H within the square grid (cf. Figure 8), numerical dilatation and erosion of $\hat{p}(X)$ representing the pdf function are denoted as :

$$\delta_{\hat{p}}(X) = (\hat{p} \oplus H)(X) = sup\{\hat{p}(Y); Y \in H_X\} \quad (18)$$

$$\varepsilon_{\hat{p}}(X) = (\hat{p} \Theta H)(X) = inf\{\hat{p}(Y); Y \in H_X\} \quad (19)$$

H_X denotes the structuring element shifted to the current point X in the map. The value of the dilated (resp. eroded) pdf at point X is then equal to the supremum (resp. infimum) value of the 3^2 estimates of the pdf at points lying in the 3^2-neighborhood of X. Thus, dilation enhances the modal regions and increases the density function in the valleys. On the other hand, erosion reduces the modal regions of the density function and enlarges the valleys between them. Hence, an erosion followed by a dilation, so-called 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 (cf. figure 7). Let $g(X)$ denote this filtered function.

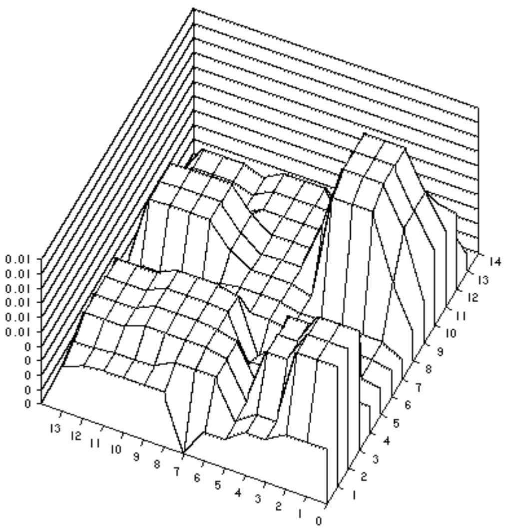

Fig. 7. Opening Smoothing

After this preprocessing, the procedure consists in the localization of the modal regions of the underlying pdf by means of the watershed algorithm based on homotopic thinning of the function [23].

This watershed approach allows determining the so-called catchment basins corresponding to the regional minima of the additive inverse of the pdf, which are the regional maxima of the pdf.

As we use watershed approach, which is well suited for determining the catchment basins corresponding to the regional minima of a function, we introduce the additive inverse $f(X)$ of the function $g(X)$ such as : $f(X) = -g(X)$ (20)

Thanks to this simple transformation, the maxima of $g(X)$ become the minima of the additive inverse $f(X)$.

The watershed of a function can be constructed through consecutive homotopic thinning of this function. The homotopic thinning is a transformation commonly used in mathematical morphology for image skeletonization. It uses a structuring element that preserves connectivity such the one shown in the figure 9, denoted $L^{(1)} = (L_1^{(1)}, L_0^{(1)})$ in the Golay alphabet, in the form of a single 3×3 matrix [24]. A value of one specifies an element that belongs to the part $L_1^{(1)}$ of $L^{(1)}$, while a value zero belongs to the part $L_0^{(1)}$. An asterisk * in the matrix denotes an element which is not used in the process. The analytical definition of the numerical thinning of $f(X)$ by the composite structuring element $L^{(1)}$, denoted $(f \circ L^{(1)})(X)$ is given by :

$$(f \circ L^{(1)})(X) = \begin{cases} \sup\limits_{Y \in L_{1X}^{(1)}} f(Y) \quad iff \quad \sup\limits_{Y \in L_{1X}^{(1)}} f(Y) < f(X) < \inf\limits_{Y \in L_{0X}^{(1)}} f(Y) \\ f(X) \quad otherwise \end{cases} \tag{21}$$

Thinning transformation are generally used sequentially. Let $L = \{L^{(1)}, L^{(2)}, ..., \cdots, L^{(8)}\}$ the family of the eight homotopic structuring elements constructed from $L^{(1)}$ such that $L^{(i+1)}$ is the configuration deduced from $L^{(i)}$ by rotation of $\pi/4$. Sequential thinning of the function $f(X)$ by this family of structuring elements is then obtained as a sequence of eight elementary thinning such as :

$$f(X) \circ L = \left((((f(X) \circ L^{(1)}) \circ L^{(2)}) \circ ...) \circ L^{(8)}\right) \tag{22}$$

Hence, the sequential thinning converges in a number of iterations that depends on the structure of the function. It is stopped when the idempotence is reached so that two consecutive iterations yield the same result. The final thinned function may present non-significant broken divide lines that do not correspond to the expected closed watershed lines. Fortunately, it is possible to "smooth" these lines by means of a sequential pruning operation. When this sequential pruning operation is iterated until the result does not change, spurious lines are shortened from their free ends, and only what is known to be an acceptable approximation of the true watershed lines remains.

1	1	1
1	$\underline{1}$	1
1	1	1

Fig. 8. Configuration of H

0	0	0
*	$\underline{1}$	*
1	1	1

Fig. 9. Configuration of $L^{(1)}$

As the composition of two idempotent mappings is not necessarily idempotent, the whole process is then iterated until idempotence. Let $w(x)$ be the density function resulting from this idempotent process whose additive inverse is presented in figure 10.

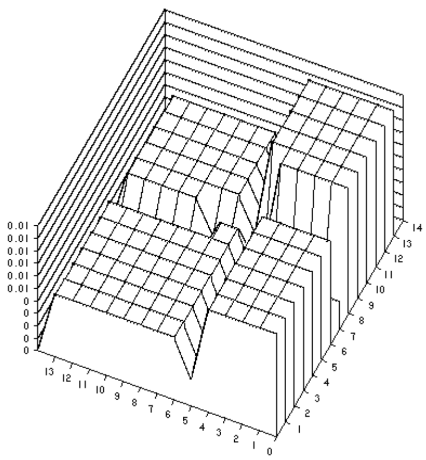

Fig. 10. Modal regions of $\widehat{p}(X)$ with the watershed algorithm

Phase 2 : Modal region extraction

Figure 10 shows that the level of the top of each modal region is constant and two neighbouring modal regions have lower levels than that of the divide separating them. In these conditions, it is evident that performing dilation with the same structuring element as the one used for the opening operation will modify the value of the additive inverse of the thinned function only at points belonging to the divides separating these modal regions. Hence, the divide set and the modal regions set can be easily extracted from the additive inverse of the thinned function by taking the point-to-point difference between the dilated version and the additive inverse of the thinned function itself (cf. Figure 11).

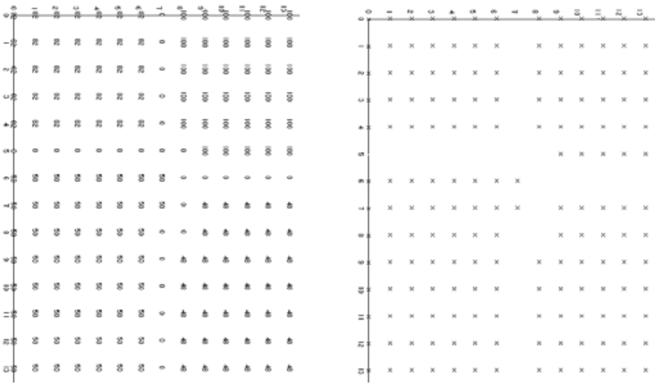

Fig. 11. Modal regions extraction

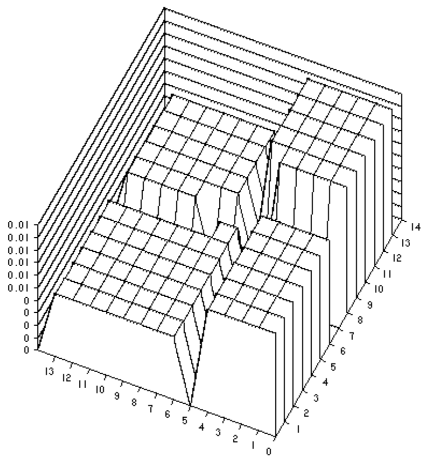

Fig. 12. Graph of the Modal regions extracted

Step.3 Classification Strategy

Under the assumption that the detected modal regions correspond to homogenous regions in the image and a one class can be represented on the Kohonen map by one or more homogeneous regions in the image, the points constitute modal regions are considered as prototypes of classes present in the image (cf. Figure 12) and are the basis of the assignment of any pixel of the image to one of the classes, through the Euclidean distance on the map (cf. Figure 14).

IV. CONCLUSION

In this work, we proposed an algorithm to unsupervised classification of textured image, based on the combination of an algorithm of Mathematical Morphology in a Kohonen map. In this algorithm, we represent at first the Kohonen map by the pdf underlying the sample of observations.

Modal regions of the pdf are then extracted into connected components by the watershed method which corresponds to a homogeneous region in the image. Finally, in classification phase, the weights vectors corresponding to the extracted modal regions are taken as prototypes of classes present in the image, and are used for the assignment of each pixel in the image to one of the classes identified. This approach shows that in an unsupervised context, the tools of mathematical morphology associated with the Kohonen map allows a good automatic classification of the textured image without using any thresholding procedure.

We must indicate at the end the help that gives the Kohonen learning to the watershed transformation, by giving in advance a separated regions even in case of overlapping between the clusters. The advantage of a watershed process is a local analysis given in the map which detects with fineness all the modes representing homogeneous regions in the texture image.

As perspective, we search to apply our approach on 3D image.

Fig. 13. Original Image

Fig. 14. Classified image

REFERENCES

[1] T. Uchiyama et M.-A. Arbib. "Color image Segmentation using competitive learning". *Pattern Analysis and Machine Intelligence*. Vol. 16, N 12, 1994.

[2] A. Gillet, C. Botte-Lecoq, L. Macaire and J.-G. Postaire, "Application of Fuzzy Mathematical Morphology for Unsupervised Color Pixels Classification", ifcs'2000, pp. 69-75, July 11-14, 2000, Namur, Belgium.

[3] A. Verikas, K. Malmqvist, L. Bergman, "Colour image segmentation by modular neural network", Pattern Recognition Letters, 1996.

[4] P.A. Devijver, "A Statistical Approach ", Englewood Cliffts, : Prentice Hall, Pattern Recognition, 1982.

[5] J. G. Postaire, and C.P.A Vasseur, " An Approximate Solution to

Normal Mixture Identification with Application to Unsupervised Pattern Classification", IEEE Trans. Patt. Ann. & Machine intel. Vol. PAMI-3, n°2, pp. 163-179, 1981.

[6] S. Delsert, "Classification interactive non supervisée de données multidimensionnelles par réseaux de neurones à apprentissage compétitif", Thése de Doctorat, Université de Lille, 1996.

[7] M. Betrouni, S. Delsert, and D. Hamad, "Interactive Pattern Classification by means of Aritificial Neural Networks", IEEE, Int. Conf. On SMC, Vol. IV, pp. 3275-3279, October 1995.

[8] T. Kohonen, "Self-Organizing Maps", Springer, Berlin, Heidelberg, 1995. Second Extented Edition 1997

[9] A. Sadeghi, "A Self-organization properly of Kohonen's map with general type of stimuli distribution". Neural Networks, pp. 1637-1643, 1998.

[10] N. R. Pal, J. Bedzek and E. C.-K. Tsao, "Generalized Clustering Networks and Kohonen's Self-Organizing Scheme". IEEE Trans. On Neural Networks, Vol. 4, N°4, pp. 549-557, July 1993.

[11] A. Sbihi and J.G. Postaire, " Mode Extraction by Multivalue Morphology for Cluster Analysis". In W. Gaul and D. Pfeifer (eds) : From DATA to Knowledge : Theoretical and Practical aspects of Classification, Springer, Berlin, pp. 212-221, 1995.

[12] T. Kohonen, "Self-Organisation and Associative Memory", *Springer-Verlag, 2nd Edition*, New York, 1984.

[13] Rosana Esteller, George Vachtsevanos, Javier Echauz, Brian Litt, "A Comparison of Waveform Fractal Dimension Algorithms", IEEE Trans. On Circuits and Systems -I: Fundamental Theory and Applications, Vol.48, No. 2, pp 177-183 Feb 2001.

[14] B. B. Chaudhuri , Nirupam Sarkar , "Texture Segmentation Using Fractal Dimension", IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 17, No. 1, Jan 1995

[15] R. Fazel-Rezai and W. Kinsner, "Texture analysis and segmentation of images using fractals," in Proc. IEEE Canadian Conf. Electrical Computer Engineering, May 9–12, pp. 786–791, 1999.

[16] Nirupam Sarkar, B. B. Chaudhuri, "An Efficient Differential Box-Counting Approach to Compute Fractal Dimension of Image", IEEE Trans. on Systems, Man and Cybernetics, Vol. 24, No. 1., pp-115-120, Jan 1990.

[17] Harsh Potlapalli , Ren C. Luo, " Fractal-Based Classification of Natural Textures", IEEE Trans. On Industrial Electronics, Vol. 45, No. 1, pp 142-150,Feb 1998.

[18] E. Parzen. "On Estimation of a probability density function and mode", *Ann. Math. Stat.*, Vol.33 pp. 1065–1076, 1962.

[19] J.G. Postaire and C.P.A Vasseur, "An Approximate Solution to Normal Mixture Identification with Application to Unsupervised Pattern Classification", IEEE Trans. Patt. Anal. & Machine intel. Vol. PAMI-3, n°2, pp. 163-179, 1981.

[20] S. Beucher et C. Lantuejoul, "Use of Watersheds in Contour Detection". *Int. Workshop on Image Processing, CETT / IRISA*, Rennes, 1979.

[21] J. G. Postaire, R. D. Zhang, C. Botte-Lecoq, "Cluster analysis by binary morphology", IEEE. Trans. Pattern Anal. Machine. Intell., Vol. PAMI-15, N°2, pp. 170-180, 1993.

[22] J. Serra, " Image Analysis and Mathematical Morphology ", Theoritical Advances, Vol.2, London, 1988.

[23] H S. Beucher, " Segmentation tools in Mathematicalns Morphology, Image Algebra and Morphological Image Processing", SPIE, Vol.1350, pp. 70-84, 1990.

[24] J A. Golay, Hexagonal Pattern Transform, IEEE. Trans. on. Computer, Vol.18, n°8, pp 733, 1969

[25] M. Talibi-Alaoui and A. Sbihi, " Fractal Features Classification For Texture Image Using Neural Network and Mathematical Morphology", Lecture Notes in Engineering and Computer Science : Proceedings of The World Congress on Engineering 2012, WCE 2012, 4-6 July, 2012, London, U.K.,pp276-28}.

[26] T. Géraud, P.-Y. Strub et J. Darbon, "Segmentation d'Imags Couleur par Classification Morphologique non supérvisée", *Proceeding of the International Conference on Image and Signal Processing ICISP'2001*, Agadir, Morocco, mai 2001.