Color-Texture Image Clustering Based on Neuro-morphological Approach

Khalid Salhi, El Miloud Jaara, Mohammed Talibi Alaoui, and Youssef Talibi Alaoui

***Abstract*—Image cluster identification is one of the main problems in many computer vision tasks. In this paper, a neural approach, combined with a mathematical morphology concept, is proposed as a solution to the problem of segmenting color-texture images in unsupervised mode. This approach involves three steps. First, both the color and texture features are extracted from each pixel of our image, using several color spaces and various textural descriptors based on fractal dimension or cooccurrence matrix. Second, the result of the concatenation of these features is used to train the Kohonen's neural network. Finally, a morphological watershed transformation will serve to segment the representation of the Kohonen's map on several modal regions, this will help on the segmentation of our image, by assigning each pixel to one region extracted from the map. Some experimental results of color-texture segmentation using various images composed from colored Brodatz Texture database are presented, these results demonstrate the improved performance of our algorithm even if it requires no a priori knowledge of the number of regions in our image. To highlight the effectiveness of our approach, our experimental results are compared first, to texture-based methods and color-based methods only, and then to the K-Means method.**

***Index Terms*—segmentation, color-texture features, neural networks, watershed.**

I. Introduction

SEGMENTATION is a very important low-level operation for the interpretation of an image, its purpose is to divide the image into several homogeneous areas called regions, each region is a connected set of pixels having common properties (gray level, color, texture, etc.). Various applications require the use of segmentation such as object recognition, medical image analysis, robotics and the domain of satellite image analysis.

There are several approaches to segmenting an image [1]–[3], usually they are grouped into three classes: pixel-based, region-based or contour-based. Our work is in the first category, which is generally divided into two steps, the extraction of information for each pixel of the image and the representation of these in the form of a vector, this step is followed by a classification of these vectors, each class will represent a uniform region of our image. The modules of our segmentation approach are shown in the Fig. 1.

In the first part of this paper, we present our approach of calculating and extracting descriptors, instead of choosing between color or texture features [4], we opted for the combination of these two to create a color-texture descriptor, this combination is done by a concatenation of the two color and texture vectors for each pixel in a single descriptive vector. In this work we will use two features for texture, the first are local fractal features computed by the differential box counting method, and the second are the Haralick features extracted from Grey Level Co-occurrence Matrix (GLCM). These two descriptors will be used in combinations with color vectors represented in the RGB and HSV color spaces.

The second section of this paper presents our unsupervised classification approach. This approach is based on the combination of the neuronal and morphological concept, unlike other classification method like K-Means, this combination gives us the advantage of classifying without knowing the number of classes before.

First, for each vector to classify, we make a projection on a two-dimensional self-organizing map, named Kohonen self-organizing feature map. When the learning phase is over, we apply the probability density function (PDF) on each neuron of the map in order to distinguish the different regions on the map. This helps in the second step to an automatic extraction of the regions of the map using our watershed transformation approach. Knowing that each region of the map represents a different class, we can thus assign to each vector the class of the region with the nearest weight vector.

All the descriptors presented before are compared with each other using our clustering approach in the last part of our article. Additionally, we present the results of the comparison of the efficiency of our pixel-based segmentation approach with the K-Means approach results.

II. Pixel-features extraction

The first step of our segmentation approach and all the approaches based on pixel classification is feature extraction for each pixel of the image. The purpose of this step is to represent the information of each pixel in the form of a vector in order to make a classification of the set of vectors that represents the entirety of the image.

Manuscript received July 17, 2018.

Khalid Salhi (corresponding author) is with the LaRI laboratory, Department of Computer Science, FSO, Mohammed First University, Oujda, Morocco (email: salhi.0.khalid@gmail.com).

El Miloud Jaara is with the LaRI laboratory, Department of Computer Science, FSO, Mohammed First University, Oujda, Morocco (email: emjaara@yahoo.fr).

Mohammed Talibi Alaoui is with the SIA laboratory, Faculty of Sciences and Technology, Sidi Mohammed Ben Abdellah University, Fez, Morocco (email: talibialaouim@yahoo.fr).

Youssef Talibi Alaoui is with the LaRI laboratory, Department of Computer Science, FSO, Mohammed First University, Oujda, Morocco (email: y.talibi.alaoui@gmail.com).

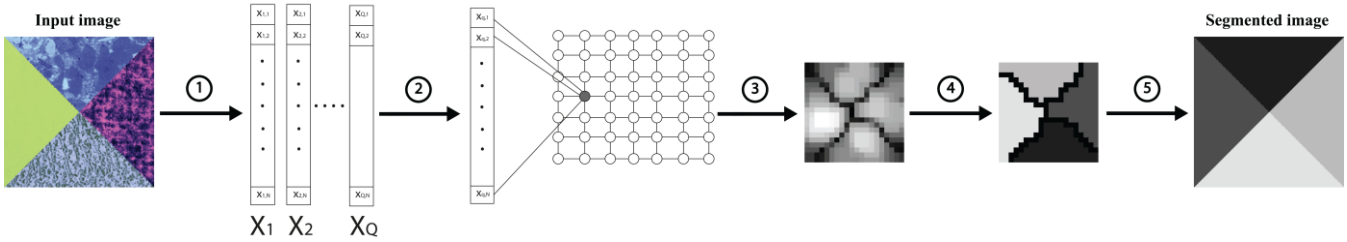

Fig. 1. The proposed approach. Step 1: color-texture features extraction, Step 2: Training the Kohonen map, Step 3: PDF visualization of the Kohonen map, Step 4: extraction of the PDF homogeneous regions, Step 5: Segmentation of the image.

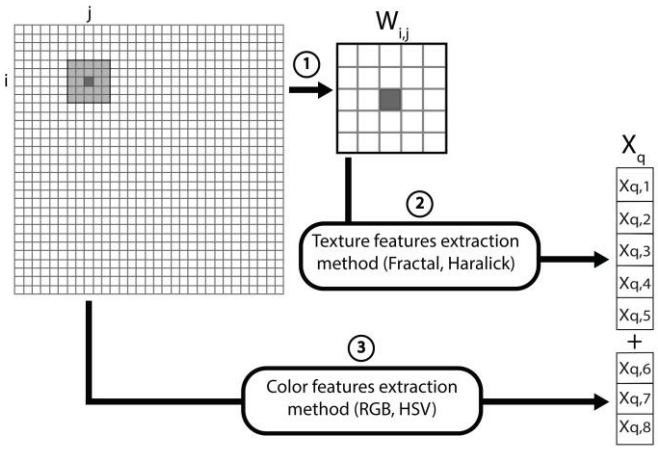

Fig. 2. Color-texture features extraction principle: (1) Sliding window, (2) Texture features extraction, (3) Color features extraction.

In this work we chose to combine the color and texture information of each pixel. This combination is done by means of the concatenation of the texture vector with the color vector of each pixel, the Fig. 2 illustrates the method used to achieve this combination.

A. *Fractal features extraction*

In 1982, fractal geometry was introduced by Mandelbrot to describe, understand and analyze some complex and irregular phenomena and forms found in nature [5]. This new concept is applied to several areas of image processing such as image coding, classification of images and image segmentation.

Generally, the fractal is used in the image analysis domain by means of the fractal dimension (FD). There are several methods in the literature that allow us to estimate this dimension. In our case, we have chosen to calculate it with the differential box counting method [6]. Its advantage is that it can automatically be applicable to patterns either with or without self-similarity.

The differential box counting method begins by dividing the image space into boxes of various sizes r, then for each one of these boxes. The probability N(r) is calculated as the difference between their maximum and minimum gray levels, after that, the fractal dimension can be estimated using the equation:

$$FD = \lim_{r \to 0} \frac{\ln[N(r)]}{\ln\left(\frac{1}{r}\right)} \tag{1}$$

The process of extracting Haralick features from each pixel I(i,j) of the image I is defined by the following algorithm:

Step 1: Extract the sliding window $W_{i,j}$ from the pixel I(i,j).

Step 2: For various r ∈ [0,1] do:

a. Divide $W_{i,j}$ into $(1/r)^2$ boxes.

b. Divide the range of intensities [0,255] into 1/r levels numbered 1..1/r .

c. For each box b(p,q) ∈$W_{i,j}$ do:

(i) l ← minimum(b(p,q)).

(ii) k ← maximum(b(p,q)).

(iii) $n_{p,q}(r)$ ← l-k+1.

d. $N(r) = \sum_{p,q} n_{p,q}(r)$.

Step 3: Do line-fit of N(r) and ln(1/r).

Step 4: The fractal dimension FD is obtained by linear regression of this line-fit.

The application of this method will allow us to extract a single piece of information for each pixel, therefore, in addition to the original image I_1, the method is also applied to the four derived images:

- High gray valued image (I_2):

$$I_2(i,j) = \begin{cases} I_1(i,j) - L_1, \text{ if } I_1(i,j) > L_1 \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

- Low gray valued image (I_3):

$$I_3(i,j) = \begin{cases} 255 - L_2, \text{ if } I_1(i,j) > (255 - L_2) \\ I_1(i,j), \text{ otherwise} \end{cases} \tag{3}$$

- Horizontally smoothed image (I_4):

$$I_4(i,j) = \frac{1}{2\omega + 1} \sum_{k=-\omega}^{\omega} I_1(i, j+k) \tag{4}$$

- Vertically smoothed image (I_5):

$$I_5(i,j) = \frac{1}{2\omega + 1} \sum_{k=-\omega}^{\omega} I_1(i+k, j) \tag{5}$$

Finally, we have for each pixel I(i,j) a vector $X_q=\{f_1,f_2,f_3,f_4,f_5\}$ where f_k is the fractal dimension of the sliding window $W_k(i,j)$ from I_k .

B. *Haralick features extraction*

The Haralick features are the most commonly used texture measures. They are derived from the Gray Level Co-occurrence Matrix (GLCM), which is a second-order statistic method introduced by Robert Haralick in 1970 [7].

A Gray level co-occurrence matrix P(*i*,*j*) is a square matrix, which describes the relative frequencies with which two pixels separated by a distance d = d(dx,dy) on the image

I, one with grey-tone *i* and the other with gray-tone *j*. Mathematically, The GLCM can be defined as:

$$P_{(dx,dy)}(i,j)=\sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & \text{if } I(p,q)=i \text{ and } I(p+dx,q+dy)=j\\ 0, & \text{otherwise}\end{cases} \quad (6)$$

In order to reduce the size of the co-occurrence matrix and also gain the calculation time for the extraction of the texture features of our image, we opted for a pre-treatment approach [8], using the coding methods as a solution to reduce the number of the grey levels in the image.

After applying the pre-processing step on our image, we calculate the five least correlated coefficients defined by Haralick: Homogeneity (f_1), Energy (f_4), Entropy (f_3), Contrast (f_4) and Correlation (f_5):

$$f_1=\sum_{i,j}\frac{P(i,j)}{1+|i-j|} \quad (7)$$

$$f_2=\sum_{i,j}P(i,j)^2 \quad (8)$$

$$f_3=\sum_{i,j}P(i,j).\ln\left[P(i,j)\right] \quad (9)$$

$$f_4=\sum_{i,j}(i-j)^2.P(i,j) \quad (10)$$

$$f_5=\sum_{i,j}\frac{(i-\mu x)(j-\mu x).P(i,j)}{\sigma x.\sigma y} \quad (11)$$

C. Color features extraction

In order to extract the color information from the image, we have opted for two color spaces, the first and most popular is the RGB, is an additive color system represented by 3 components Red, Green and Blue.

We also use the HSV color space, which refer to Hue, Saturation, and Value components. The HSV is different from RGB space in order to decouple chromatic information from intensity information, which makes it closer to human visual perception, in addition that it is easily invertible transform from RGB.

III. Clustering Approach

A. Kohonen Self-organizing Map

The self-organizing map algorithms are a part of the numerous tools for data analysis, in 1984, inspired by a kind of biological neural network in human cortex and his phenomenon of self-organization, Teuvo Kohonen created a sort of SOM known as the self-organizing Kohonen map (KSOM) [9], which was considered as a very high performing algorithm to analyze and visualize multidimensional data, and on top of that, Kohonen's algorithm prove to be an effective clustering method.

The main advantage of KSOM lies in its architecture, which is different than the classical neural networks by the organization of neurons within the output layer. Thus, the neurons are represented by nodes arranged in a one, two or three-dimensional grid, which will allow close input data to be represented by the same node or a close node in the map, thereby, the topological relationships between the input vectors are preserved in the output map.

In our work we chose to use a two-dimensional KSOM, as shown in Fig. 3, each unit of the input vector X_q is linked to the nodes of the output layer, which will create an interconnection $W_{m,i}$ for each input i and output m, thereby we can represent each neuron m of the output layer by its weight vector $Wm = [W_{m,1}, W_{m,2},\ldots\ldots, W_{m,N}]^T$.

in order to represent the image extracted features in our Kohonen map, a learning algorithm must be defined by the following steps:

Step 1: Initializing the weights of each neuron in the output layer by giving them random values.
Step 2: Presenting a random input color-texture vectors X_q to the neural network.
Step 3: Finding the closest neuron m^* by calculating the Euclidian distance between the input vector and each neuron of the map.
Step 4: Updating the weights of the winner neuron and each neighbouring neuron using the equations:

$$\begin{cases}W_m(t)=W_m(t-1)+\alpha(t).[X_q-W_m(t-1)]\\ if\ m=m^*\\ \\ W_m(t)=W_m(t-1)+\alpha(t).h_m(t).[X_q-W_m(t-1)]\\ if\ m\in V(m^*,r(t))\end{cases} \quad (12)$$

Step 5: Reducing the learning coefficient a(t).
Step 6: Reducing the size of the neighbourhood area of the winner neuron.
Step 7: Repeat from the Step 2. until the end of learning.

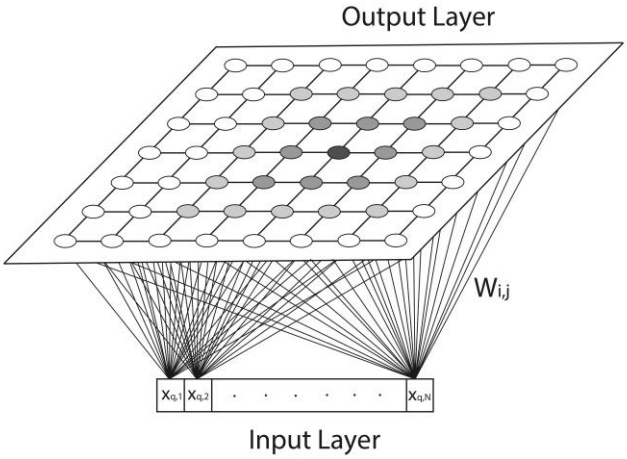

Fig. 3. Kohonen map architecture

After the training of our KSOM, the most effective method of interpreting the map is to visualize it using a visualization technique. In our work we opted for the underlying probability density function (PDF) [10]. This technique will allow us to display the borders between the distant classes by calculating the distances between the weight vectors of each node of the map. By this means it will help to distinguish visually the different classes detected in the map, even in case of important overlaps.

In order to visualize the PDF of our map, we calculate the estimate for each point of the space defined by the components of the weight vectors W_m defined by the following equation:

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

B. Watershed Transformation

In order to automate the extraction of classes, we should apply a segmentation algorithm on the PDF of our map, and due to the fact that PDF estimation has allowed us to clearly distinguish edges between classes, watershed image segmentation algorithm will be more suitable in our case [11].

In fact, the watershed technique will consider the estimated PDF on the Kohonen map as three-dimensional mountains, where the third dimension is determined by the intensity value of each pixel in the PDF. Thus, the pixels with the highest intensity values will be visualized as tops, and those with a low intensity as valleys.

The idea of this technique then is to calculate the watersheds lines of this 3D visualization, that represent the regions boundaries. The watersheds thus obtained correspond to the different classes of our map.

Knowing that the watershed transform is highly sensitive to noise, we apply a morphological opening on our PDF estimation, which will allow us to remove all the extremities of its peaks.

Let p(x) be the opened PDF function, our method allows the construction of the watershed lines through consecutive homotopic thinning operations using a family of the eight homotopic structuring elements $L = \{L_i, i=1,2,\ldots,8\}$ Fig. 4(a), followed by a sequential pruning operation using the structuring family $E = \{E_j, j=1,2,\ldots,8\}$ Fig. 4(b). These two morphological operations are applied on the inverse -p(x) and repeated consecutively until idempotence [11], [12] Fig. 5.

(a)

0	0	0
*	1	*
1	1	1

(b)

0	0	0
0	1	*
0	0	*

Fig. 4. Structuring elements family: (a) the first structuring element L1 for thinning (b) the first structuring element E1 for pruning.

Fig. 6 shows the result of our watershed technique steps on three PDF estimations containing three, four and five different classes, respectively. We can notice that the resulting function p'(x) of the application of our watershed transform technique takes a unique constant value within each different region, which equals the regional maximum of the same region; furthermore, the watershed line separating each neighboring modal regions does not exceed one cell in thickness. By exploiting these two properties, we can extract the different regions easily by applying a dilation on the function p'(x).

After the extraction of the modal regions, each region will represent a different class of the image, by this means we can calculate the Euclidean distance between the pixel, which is represented by its texture-color vector X_q, and each node m of the map represented by its weight vector Wm, thereby, we can assign to each pixel of the image the class of the closest node on the map.

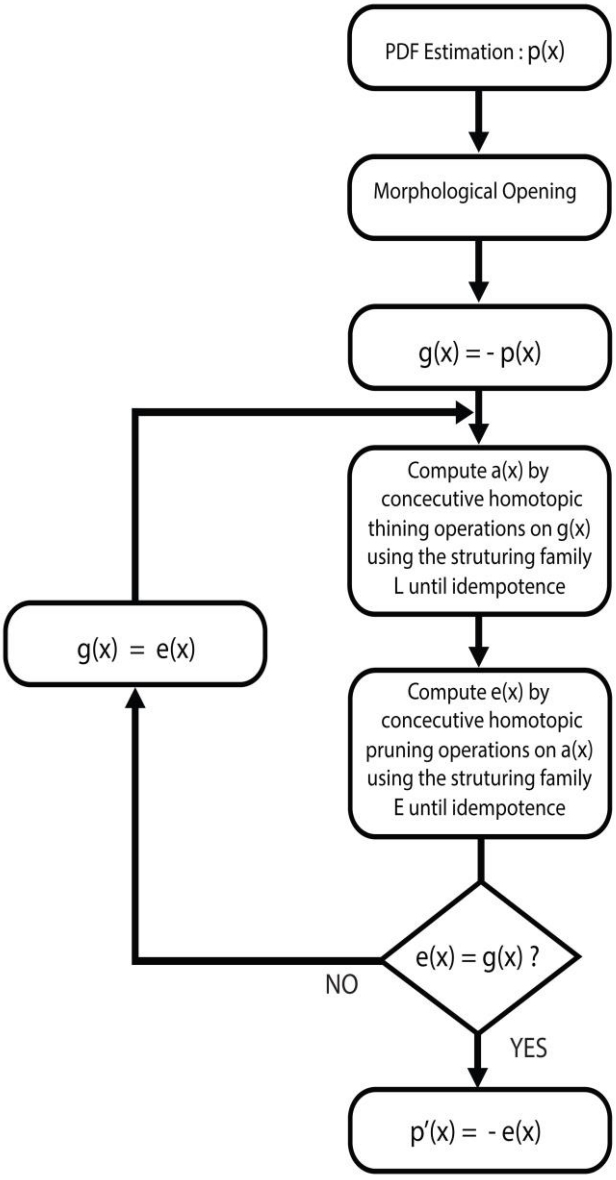

Fig. 5. Proposed watershed transform algorithm

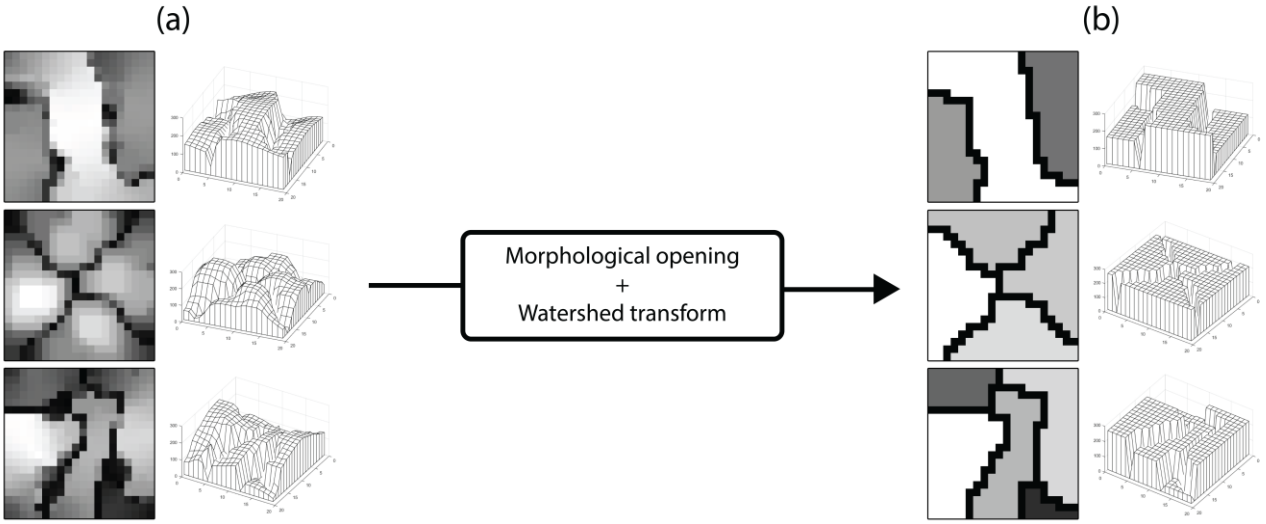

Fig. 6. Watershed transformation: (a) PDF, (b) PDF modal regions

IV. Results and Discussion

We have made a series of experiments to compare the different results obtained by applying the color and textural features presented in this paper to our approach. Afterwards, in order to evaluate the effectiveness of the proposed approach, we compared its results with that obtained by k-means method.

In our experiments, we applied both approaches on a group of 9 images of 256×256 pixels composed each one of two to five different textures Fig. 8, chosen from the colored Brodatz texture database [13].

The choice of the size for the KSOM has a considerable impact on modal regions number. Consequently, to optimize the choice of this parameter, we have experimented several sizes for the map from 8×8 to 30×30, calculating it from several images which contains 2 to 5 regions. According to the results shown in Fig. 7, we found that the choice of a size of 20×20 for the Kohonen map, allowed us to have good results of segmentation using our watershed method without having problems of over-segmentation. Regarding the size of the features window, we must avoid choosing a very large size to minimize the overlap of different textures that are in the border of different regions of the image. At the same time, the choice of a small size will lose a lot of textural information. Thus, we opted for an optimal size of 7×7 for Haralick features, and 11×11 for fractal features.

Especially for the Haralick features, we have implemented a pre-processing rank coding method shown in a previous work [8], that allows us to gain a lot of time to extract these features.

Finally, as the quality criterion, we use a misclassification error measure, based on the number of misclassified pixels. For that, we created for each image a manually segmented ground truth image, in order to compare it with the images segmented by our approach and get the clustering accuracy defined by the equation (14). All these results are presented in Table 1.

$$\text{Clustering accuracy (\%)} = \left[\frac{\text{Number of misclassified pixels}}{\text{Total number of pixels}}\right] \quad (14)$$

All the experiments are developed using the C++ library OPENCV3 and run on a Windows10 pc with core i7(7700k) and 16GB RAM.

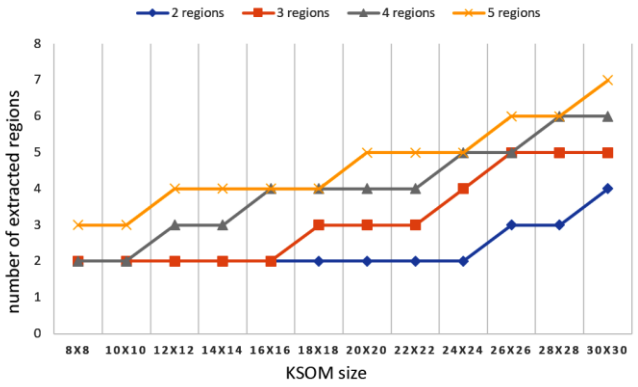

Fig. 7. Impact of KSOM size on extracted regions number from images with 2 to 5 regions.

For each image we first calculate the segmentation rate using the RGB and HSV color vectors, the texture vectors extracted by the Haralick method and the fractal method, and the combined texture-color vectors from the four possible combinations. All segmentation results of our 9 images obtained by the proposed approach and K-Means are shown in Table 1.

Tests have shown that combining textured attributes with color usually gives better results than using each information separately. In fact, we can notice in Fig. 9 that the addition of RGB color features to the texture information extracted by the Haralick method in the image gives more efficient segmentation results. The same remark goes for fractal attributes in image results, where the addition of HSV information has increased the segmentation rate. Besides, it can be seen that the Haralick descriptors suffer from the misclassification of regions which contains macro-texture, unlike the fractal features that gives good results no matter the type of textures.

In addition to the equivalent or better results for all the images of our experiment (see Table 1), our proposed approach allowed us, thanks to the neuro-morphological combinations, to avoid one of the biggest problems of K-Means; which suffers from being confined to run with a fixed number of regions represented by K, contrary to our approach which allows us to go beyond this step by the automatic extraction of the number of the regions of the image from the Kohonen map thanks to our morphological

watershed technique.

V. Conclusion

In this study, we present our neuro-morphological clustering approach for color-texture images, which has been done with the application of our morphological watershed approach on a visualization of the Kohonen self-organizing map. The KSOM has been trained by a color-texture vectors extracted from each pixel of the image, and these vectors are a result of a combination of texture and color attributes. In this paper, we present two textural features, the first based on the calculation of the local fractal dimension and the second is extracted from the GLCM matrix. These two vectors are both combined with color attributes extracted from the HSV and RGB color spaces.

Our results are very encouraging comparing it with K-Means, knowing that the proposed approach does not need to fix the number of regions of the image.

In future the work is extended in the direction of segmenting 3D images using this approach. Additionally, we plan to propose a parallel version of it, using both CPU and GPU acceleration.

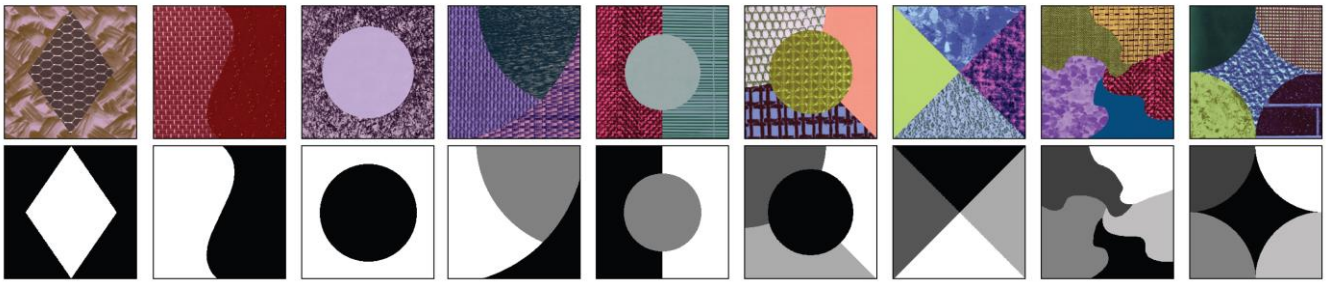

Fig. 8. Test images. The first row: input images 1 to 9 from left to right. The second row: the corresponding ground truth.

TABLE I
Clustering rate result

		FRACTAL	HARALICK	RGB	HSV	FRACTAL + RGB	FRACTAL + HSV	HARALICK + RGB	HARALICK + HSV
Image 1	Our approach	94,44%	82,36%	59,50%	72,36%	**96,02%**	94,14%	88,74%	72,91%
	K-Means approach	94,32%	81,48%	58,85%	70,95%	95,00%	94,17%	87,00%	71,60%
Image 2	Our approach	99,27%	96,97%	54,26%	99,56%	99,39%	99,50%	98,68%	**100,00%**
	K-Means approach	99,19%	96,65%	56,52%	99,56%	99,30%	99,49%	98,35%	100,00%
Image 3	Our approach	96,03%	97,99%	85,38%	79,46%	97,24%	96,16%	90,98%	**99,43%**
	K-Means approach	96,11%	97,97%	81,18%	75,94%	97,32%	96,23%	86,68%	99,39%
Image 4	Our approach	97,00%	58,59%	54,15%	47,69%	**97,45%**	97,26%	66,56%	61,58%
	K-Means approach	96,31%	56,69%	56,98%	48,00%	96,79%	96,64%	64,76%	59,88%
Image 5	Our approach	96,31%	96,52%	73,91%	76,54%	96,22%	96,25%	**98,65%**	98,50%
	K-Means approach	95,91%	95,36%	72,97%	77,03%	96,27%	96,11%	97,86%	98,01%
Image 6	Our approach	94,65%	56,75%	76,60%	79,89%	**97,09%**	96,73%	81,74%	87,49%
	K-Means approach	94,49%	56,03%	74,37%	78,92%	96,35%	96,59%	80,86%	86,16%
Image 7	Our approach	87,04%	47,68%	66,87%	69,54%	90,86%	88,19%	69,65%	82,58%
	K-Means approach	87,20%	46,91%	65,34%	68,45%	**91,14%**	88,32%	67,89%	80,23%
Image 8	Our approach	94,66%	71,58%	90,75%	90,83%	92,33%	**95,96%**	70,38%	91,43%
	K-Means approach	94,31%	69,93%	84,60%	88,09%	90,56%	95,20%	66,06%	89,10%
Image 9	Our approach	94,51%	49,48%	59,54%	86,93%	**96,07%**	95,41%	75,35%	87,43%
	K-Means approach	94,31%	48,16%	60,38%	86,33%	95,85%	95,29%	72,68%	84,63%

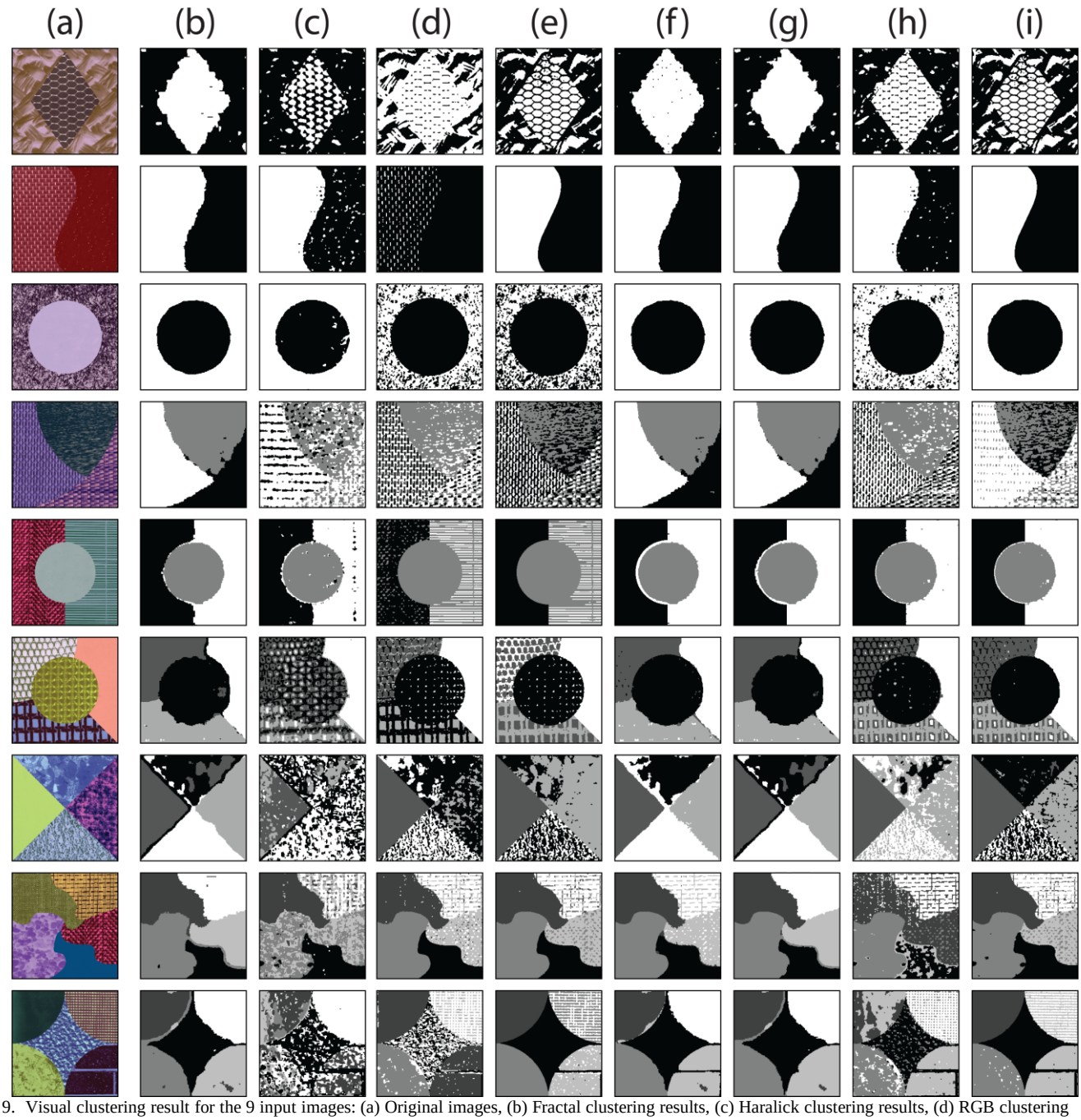

Fig. 9. Visual clustering result for the 9 input images: (a) Original images, (b) Fractal clustering results, (c) Haralick clustering results, (d) RGB clustering results, (e) HSV clustering results, (f) Fractal + RGB clustering results, (g) Fractal + HSV clustering results, (h) Haralick + RGB clustering results, (i) Haralick + HSV clustering results.

REFERENCES

[1] M.A.-M. Salem, A. Atef, A. Salah, and M. Shams, "Recent survey on medical image segmentation," in *Computer Vision: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2018, pp. 129–169.

[2] N. Dhanachandra and Y. J. Chanu, "A survey on image segmentation methods using clustering techniques," *European Journal of Engineering Research and Science*, vol. 2, no. 1, pp. 15–20, 2017.

[3] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.

[4] L. Liu, J. Chen, P. Fieguth,G. Zhao, R. Chellappa, and M. Pietikainen, "A survey of recent advances in texture representation," *arXiv preprint arXiv:1801.10324*, 2018.

[5] B. B. Mandelbrot, "The fractal geometry of nature/revised and enlarged edition," *New York, WH freeman and Co., 1983, 495 p.*, 1983.

[6] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 17, no. 1, pp. 72-77, 1995.

[7] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786-804, 1979.

[8] K. Salhi, E.M. Jaara, and M. T. Alaoui, "Pretreatment approaches for texture image segmentation," *Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on*. IEEE, 2016, pp. 221–225.

[9] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52-65, 2013.

[10] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no 3, pp. 1065-1076, 1962.

[11] K. Salhi, E. M. Jaara, and M. T. Alaoui, "Texture image segmentation approach based on neural networks," *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, vol. 6, no 1, pp. 19-32, 2018.

[12] M. Talibi-Alaoui and A. Sbihi, "Application of a mathematical morphological process and neural network for unsupervised texture image classification with fractal features," *IAENG International Journal of Computer Science*, vol. 39, no. 3, pp. 286–294, 2012.

[13] S. Abdelmounaime, and H. Dong-Chen, "New Brodatz-based image databases for grayscale color and multiband texture analysis," *ISRN Machine Vision*, vol.2013, 2013.