International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:12, 2013

Automatic Moment-Based Texture Segmentation

Tudor Barbu

***Abstract*—**An automatic moment-based texture segmentation approach is proposed in this paper. First, we describe the related work in this computer vision domain. Our texture feature extraction, the first part of the texture recognition process, produces a set of moment-based feature vectors. For each image pixel, a texture feature vector is computed as a sequence of area moments. Then, an automatic pixel classification approach is proposed. The feature vectors are clustered using an unsupervised classification algorithm, the optimal number of clusters being determined using a measure based on validation indexes. From the resulted pixel classes one determines easily the desired texture regions of the image.

***Keywords*—**Image segmentation, moment-based texture analysis, automatic classification, validity indexes.

I. Introduction

THIS paper approaches the texture analysis domain, providing a texture-based image segmentation approach. In the computer vision domain, segmentation refers to the process of partitioning a digital image into multiple segments, or regions, which represent sets of pixels [1]. All the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture.

Image segmentation methods can be classified into two main categories: contour-based and region-based techniques [1]-[3]. The contour-based approaches use edge detection operations for the segmentation process. They provide a set of contours extracted from the image. Region-based methods provide a set of regions that collectively cover the entire image. They include histogram-based, clustering based, graph partitioning based, watershed based and neural network based techniques.

We propose a region-based image segmentation approach using a texture-based pixel clustering. Texture segmentation, representing the process of dividing the image space into texture regions, has been a topic of intensive research for over four decades [1]-[3].

Texture analysis has been approached using various techniques, such as those based on Gabor filtering [2], [4], Wavelet Transforms [2], [5], image moments [2], [6], [7], Fourier descriptors [8], geometrical approaches [9], co-occurrence matrices [10] or correlation operations [2]. This related work in texture analysis is described in the next section. We provide an automatic moment-based texture segmentation technique in this paper.

An important application area of texture recognition is object detection. Thus, this process of detecting regions with similar texture and separating regions with different texture could become an essential step in semantic image object extraction. Other potential application include biomedical image analysis, analysis of satellite or aerial imagery, content-based image retrieval, document analysis, biometric person authentication, image coding and robotics [2], [3].

In the third section a texture feature extraction method is provided. One computes a texture feature vector, as sequence of area moments, for each pixel of the image [6], [7]. The second step of the texture recognition process, feature vector classification, is described in the fourth section. Unsupervised *K-means* algorithms [6], [12] and cluster validation indexes [13], [14] are used for pixel classification.

Some texture-based segmentation experiments and comparisons with other techniques are also mentioned in this paper. The work ends with a conclusions section and a list of references.

II. Related Work

Image texture, defined as a function of the spatial variation in pixel intensities, is useful in a variety of applications and has been a subject of intense study by many researchers. Many texture analysis methods have been proposed in the past several decades.

The available techniques might be categorized into geometrical, statistical, probability model-based and signal processing (spectral) methods. Geometrical approaches are characterized by their definition of texture as being composed of *texture primitives* or *texels*. Some important geometrical texture analysis methods use the *Voronoi tessellation* of the image [9].

Another category of geometrical approaches are the structural models. The texture is produced by the placement of the texture primitives according to certain rules. This class of algorithms, work satisfactory when is dealing with very regular textures.

Statistical methods collect image signal statistics from the spatial domain as texture feature descriptors. Commonly applied statistics include 1D histograms, moments, grey-level co-occurrence matrices or autocorrelation functions. Texture analysis usually exploits low-order image statistics, particularly first- and second-order statistics. The first-order statistics, such as the mean, variance, standard deviation and higher-order histogram moments, concern with properties of the individual image pixels.

The second-order statistics treat not only the individual pixel properties but also the spatial inter-dependency or co-occurrence of two pixels at specific relative positions. The most popular second-order statistics for texture recognition include *grey level co-occurrence matrices* (GLCM) [10], *grey level differences*, *autocorrelation function* [2], and the *local binary pattern operator* [2].

T. Barbu is with the Institute of Computer Science of the Romanian Academy, Iaşi, Romania (e-mail: tudbar@iit.tuiasi.ro).

The computational complexity increases exponentially with the order of statistics. For this reason, the statistics higher than second-order are not very common in texture description.

Model based texture analysis techniques are based on the developing of an image model that describes textures. The most common probability models for texture analysis are *Markov Random Fields* and *Gibbs Random Fields* [11].

Markov Random Fields (MRF) capture the local contextual information in an image. These probability models assume that the intensity at each pixel in the image depends on the intensities from the neighborhood. Gibbs random fields (GRF) models represent useful tools for texture discrimination. In a typical modeling scenario, an MRF is defined over the image lattice to encapsulate and represent domain knowledge and observations about the texture to be modeled, e.g., by specifying a neighborhood system and related conditional probabilities. Then, by Markov-Gibbs equivalence, a Gibbs distribution of the MRF is derived as a texture model [11]. The spectral texture analysis techniques are based on two-dimensional signal processing. They include the spatial domain filters, Fourier descriptors, image moments, Gabor filtering and Wavelets models.

Spatial domain filters are the most direct way to capture image texture properties. Some filters, such as the operators Roberts and Laplacian, measure the image edgeness (edges per unit area). Other spatial filter based approaches model the pre-attentive texture perception in human visual system.

Fourier descriptors, usually used for shape recognition, represent another important texture analysis tool. The human visual system analyzes the textured image by decomposing it into frequency and orientation components [8]. The frequency analysis of the textured image is best done in the Fourier domain. *Discrete Fourier Transforms* (DFT) are used to describe the texture regularity. One can obtain discriminative features for regularity computation by using this Fourier spectrum [2], [8].

Gabor filters represent powerful tools for image texture analysis and segmentation [2], [4]. A two-dimensional Gabor filter consists of a sinusoidal plane wave of a certain frequency and orientation modulated by a Gaussian envelope. 2D Gabor filters capture optimally both local orientation and frequency information from the textured image.

Basically, these filters are a group of wavelets, with specific each wavelet capturing energy at a specific frequency and a specific direction. The Gabor filter based texture analysis techniques apply banks of filters at multiple scales and orientations on the image, to obtain texture features [4].

Since human visual system can be modeled as a set of independent channels, each with a particular orientation and spatial frequency, the *Discrete Wavelet Transforms* (DWT) are found useful to extract texture features [5]. Energy features, as well as first and second-order statistics of the wavelet coefficients, have both been used successfully for texture description. Multiple Wavelet analysis produces additional information about image texture.

Image area moments represent another category of spectral methods [2]. The moment-based features have been used successfully in texture segmentation. Our segmentation technique described in the next section represents such a signal processing based approach using area moments.

III. Moment-Based Texture Feature Extraction

Our approach performs a texture recognition process for each image pixel. A texture feature extraction is performed first. We propose a moment-based feature extraction in this section [2], [6], [7].

As one knows from the moment representation theorem, a two-dimensional real bounded function with support on a finite region is unique determined by an infinite set of its $(p+q)^{\text{th}}$ - order moments, m_{pq}. Therefore, we try to represent a texture region using a set of area moments [2].

First, the color digital image to be segmented is converted into the grayscale form. The obtained image could still be affected by various types of noise, such as Gaussian noise, therefore some 2D smoothing filters and restoration techniques have to be applied to them. We proposed our own image denoising and restoration techniques, which are based on *PDE* operations, in some of our previous works [15], [16].

Let I be such an enhanced $[X \times Y]$ grayscale image. For each pixel of I we consider a square $[(2N+1)\times(2N+1)]$ neighborhood of it, the current pixel being located at its center, where the value of N depends on texture density. For this region one computes a sequence of 6 modified central area moments [6]. Therefore, for the i^{th} pixel of image I, its feature vector is obtained as:

$$V(i)=\left[\hat{\mu}_{00}(i),\hat{\mu}_{01}(i),\hat{\mu}_{11}(i),\hat{\mu}_{12}(i),\hat{\mu}_{22}(i),\hat{\mu}_{23}(i)\right], i\in[1,X\cdot Y] \quad (1)$$

where each component of it is computed as a centered moment having the following form:

$$\hat{\mu}_{pq}(i)=\sum_{x=x_i-N}^{x=x_i+N}\sum_{y=y_i-N}^{y=y_i+N} I(x,y)\cdot\left(\frac{x-C_x}{\sigma_x}\right)^p\cdot\left(\frac{y-C_y}{\sigma_y}\right)^q, p,q\geq 0 \quad (2)$$

where coordinates of the center of gravity are computed as:

$$C_x=\frac{m_{10}}{m_{00}}, C_y=\frac{m_{01}}{m_{00}} \quad (3)$$

with

$$m_{pq}(i)=\sum_{x=x_i-N}^{x=x_i+N}\sum_{y=y_i-N}^{y=y_i+N} I(x,y)\cdot x^p\cdot y^q,\ p,q\geq 0 \quad (4)$$

and the standard deviations in x and y directions are

$$\sigma_x=\sqrt{\frac{\mu_{20}}{m_{00}}}, \sigma_y=\sqrt{\frac{\mu_{02}}{m_{00}}} \quad (5)$$

with

$$\mu_{pq}(i)=\sum_{x=x_i-N}^{x=x_i+N}\sum_{y=y_i-N}^{y=y_i+N} I(x,y)\cdot(x-C_x)^p\cdot(y-C_y)^q,\ p,q\geq 0 \quad (6)$$

where (x_i, y_i) represents the position of the i^{th} pixel in image *I*. The moment-based texture feature vectors computed by (1) can be compared using metrics like the Euclidian distance, sum of absolute differences (*SAD*), or the *squared Euclidian metric*, in the classification process.

It is obvious, as resulting from the moment representation theorem, that inserting more $\hat{\mu}_{pq}(i)$ moments into the sequence of $V(i)$ makes this feature vector a more powerful texture descriptor. The main disadvantage of more complex feature vectors is the high computational cost of the texture analysis process.

The calculation of these texture feature vector coefficients $\hat{\mu}_{pq}(i)$ is time-consuming, because it requires many operations. Its algorithm has a polynomial time complexity. The computational complexity for each discrete image moment has the order $O(p\cdot q\cdot n^2)$, where *n* represents here the number of pixels from a square neighborhood. Thus, we obtain $n=(2N+1)^2$ and the time complexity is $O(p\cdot q\cdot(2N+1)^4)$.

Therefore, it is obvious that including more higher order moments into this $V(i)$ feature vector, would increase the processing time of the classification procedure very much. So, to reduce the time complexity of the image segmentation technique, we have considered only these six discrete area moments, which are up to the order $p+q=5$.

IV. Unsupervised Feature Vector Classification Approach

The feature vectors obtained in the previous section have to be clustered in a number of classes. We have developed an automatic classification approach using an unsupervised algorithm [12]-[14].

Obviously, we have to classify $X\cdot Y$ texture feature vectors, therefore the number of classes, *K*, should satisfy: $1\leq K<X\cdot Y$. Usually, the number of texture classes is *much smaller* than $X\cdot Y$, so let us consider a value *T* representing the maximum number of possible classes.

Then, for each $K\in[1,T]$ we apply an unsupervised classification procedure to the feature vector set [12]. We choose some *K-means algorithms* for performing these clustering tasks. Obviously, choosing a high value for *T* raises the computation volume and the time complexity of the classification process.

The *K*-means algorithm, proposed by MacQueen in 1967, represents one of the simplest unsupervised learning algorithms [6], [12]. In statistics and machine learning, *K*-means clustering is a method of cluster analysis which aims to partition *n* observations into *K* clusters in which each observation belongs to the cluster with the nearest mean. The partitioning must be performed so as to minimize the *within-cluster sum of squares* (*WCSS*). This algorithm could be successfully applied to image segmentation. The number of clusters, *K*, is a priori known. In our segmentation case, we consider the following objective function to be minimized by the *K*-means procedure, as a within-cluster sum of squares:

$$F=\sum_{i=1}^{K}\sum_{j\in Ind(C_i)} d(V(j),C_i) \quad (7)$$

where *d* is a proper metric (Euclidian distance, for example), C_i is the centroid of the current cluster and $Ind(C_i)$ represents the set of the indices of pixels corresponding to the feature vectors from that cluster. The clustering procedure uses an iterative refinement technique, being composed of the following steps:

1. Initialize *K* centroids: we define *K* initial centroids, one for each cluster, each centroid representing a point in the feature vector space. We select them as much as possible far away from each other.
2. Each texture feature vector is associated to the closest centroid (in terms of the used distance)
3. The centroids of the obtained clusters are recomputed, as means of the vectors in those clusters.
4. Steps 2 and 3 are repeated until the centroids of classes no longer change their positions.

The final clusters resulted from this *K*-means algorithm represent the texture feature vector classification result for the current value of *K*. The time complexity of this clustering process is $O(n^{Kl+1}\log n)$, where *n* represents here the number of entities to be classified and *l* represents the length of the feature vectors corresponding to these entities.

We have $l=6$, the number of the used image moments, and $n=X\cdot Y$, the number of image pixels to be classified. Therefore, we will obtain $O((X\cdot Y)^{6K+1}\log(X\cdot Y))$ as the necessary time for the texture feature vector clustering process to be performed, if *K* value is fixed. Obviously, increasing the *l* value, by adding more moment-based coefficients to the feature vector $V(i)$, will considerably increase the computational complexity of the clustering *K*-means algorithm. The complexity is also proportional to the image *I* size. If *I* has small *X* and *Y* dimensions, then the time complexity decreases and the recognition process runs faster. Otherwise, if one uses high-dimension images, the computational cost of this clustering process substantially increases.

In the next step, we have to determine the optimal number of clusters, from 1 to *T*. We will use some cluster validity indexes for this purpose [13], [14]. We propose a measure that is based on a combination of Dunn and Davies-Bouldin validity indexes. It has to be minimized for a proper clustering.

The Dunn measure aims to maximize the inter-cluster distances and minimize the intra-cluster distances [13]. The number of clusters that maximize the Dunn index is taken as the optimal number of clusters. The Dunn validity index has a linear time complexity, its computational complexity being *O*

(n). This represents an advantage, the main disadvantage of Dunn index disadvantage being the vulnerability for noise in the data.

The index proposed by Davies and Bouldin is a function of the ratio of the sum of within-cluster scatter to between-cluster separation [14]. Its lowest value indicates an optimal clustering operation. Davies-Bouldin index becomes computationally expensive when the number of clusters grows very large.

If the obtained texture feature vector classes, for a given K, are Cl_1,..., Cl_K, then the optimal number of texture classes is computed as follows:

$$K_{optim} = \arg\min_{K\in[1,T]}\left(DB(K)+\frac{1}{D(K)}\right) \quad (8)$$

where the Davies-Bouldin index is computed in this case as

$$DB(K) = \frac{1}{K}\sum_{i=1}^{K}\max_{i\neq j}\frac{d(Cl_i)+d(Cl_j)}{d(C_i,C_j)} \quad (9)$$

and our Dunn index is obtained as

$$D(K) = \min_{i\in[1,K]}\left\{\min_{j\neq i}\left\{\frac{d(C_i,C_j)}{\max_{t\in[1,K]}d(Cl_t)}\right\}\right\} \quad (10)$$

where C_i is the centroid of the cluster Cl_i, while $d(Cl_i)$ represents the intra-cluster distance of Cl_i, that is the absolute/squared distance between all pairs of points in that cluster. Therefore, the optimal feature vector clustering process is that corresponding to the detected K_{optim} value.

The obtained K feature vector clusters correspond to K classes of pixels, each class grouping pixels having the same texture. Texture region detection becomes a facile process using these texture classes. Thus, any connected set of pixels belonging to the same texture class represents a texture region. Obviously, each texture class may contain one or more texture regions. We could add another processing step that eliminates the very small texture regions surrounded by large regions, these small spots being considered as classification errors.

V. Experiments and Method Comparisons

We have performed a lot of practical experiments, testing the provided texture-based segmentation technique on various image data sets. We have obtained satisfactory texture recognition results.

Our automatic texture segmentation approach has been tested on tens images containing textures. A high texture-based pixel recognition rate, over 80%, has been obtained. The proposed technique runs faster on low-dimension images, because of its computational cost, so the most images we have used were quite small in size (under $[200 \times 200]$ dimension).

We have chosen a limit value $T = 25$ in our experiments. Using a greater limit value for the possible number of clusters would produce better recognition results sometimes, but it also produces a high complexity process and a great processing time. However, the most digital images do not exceed the threshold limit of 25 texture classes.

Also, we have generally used a value $N = 2$ for establishing the neighborhood of each pixel. Each square neighborhood has a $[5 \times 5]$ size, which we consider to be the proper size for our segmentation task. Obviously, the neighborhood size should depend on the texture density. If the analyzed texture has quite large structural units, then the N value should be increased. For example one could choose $N = 3$, producing a $[7 \times 7]$ neighborhood, or $N = 4$, which means a $[9 \times 9]$ neighborhood for each pixel. If the texture is characterized by small area texels, then a lower value can be chosen for N. For example, we could set $N = 1$, obtaining $[3 \times 3]$ - sized neighborhoods.

Such a texture recognition example is depicted in Fig. 1. The RGB color textured image displayed in Fig. 1 (a) is converted to a grayscale form, which is displayed in Fig. 1 (b). Then the grayscale image is filtered for noise removal, and the automatic texture-based segmentation procedure is applied, using $T = 25$ and $N = 2$. The 3 detected texture classes are represented in Fig. 1 (c) and the resulted 4 texture regions are depicted in Fig. 1 (d). One can see that two identified texture regions of the image, which are labeled with 1 in the picture, belong to the same texture class.

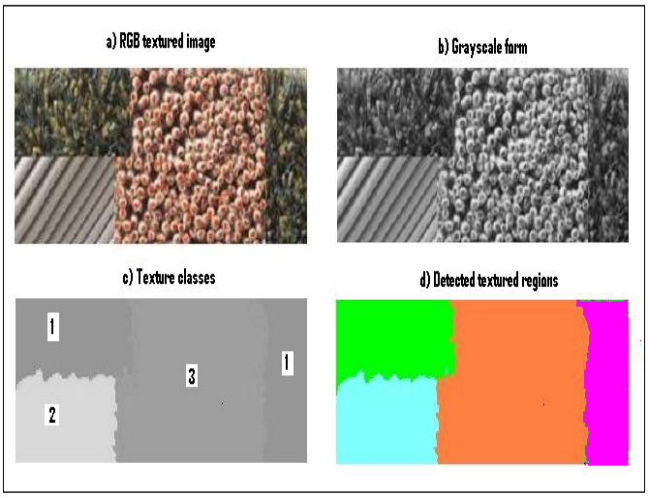

Fig. 1 Textured image segmentation example

The proposed texture segmentation approach works satisfactory on images containing not only textures, but homogeneous color regions also. Such an image segmentation example obtained from our experiments, is described in Fig. 2. The color image displayed in Fig. 2 (a) contains a homogeneous region and a textured one. The segmentation algorithm identifies two texture classes which correspond to the two regions. As one can see in the next figure, this texture segmentation process is a simpler one. In this case the

obtained texture classes are the same with the detected texture/homogeneous regions. Also, the segmentation process is much faster in this case, the detection of color homogeneous regions requiring a lower computation time than the texture region detection process.

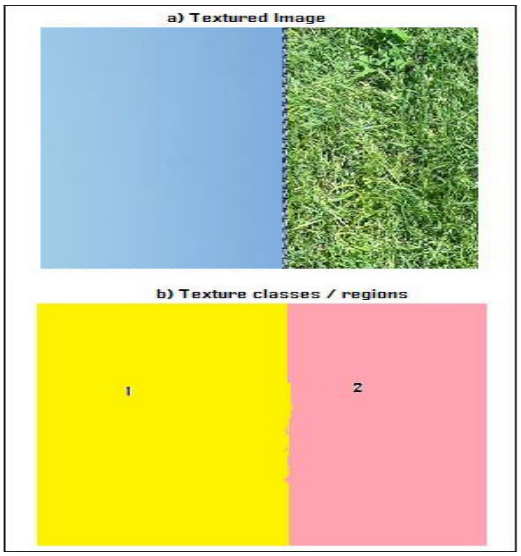

Fig. 2 Textured image segmentation example

We have compared our moment-based texture recognition technique with other texture analysis methods. We have compared both the feature extraction part and the classification part of our approach with these components corresponding to other methods.

Because of the complex structure of textures, the texture based segmentation approaches are usually characterized by high computational complexities and require long processing times. Techniques using moment-based feature extraction have a high computational cost too, but produce better segmentation results than other approaches. Our texture segmentation method achieves higher discrimination rates than the transform-based (such as those based on *FFT*) and structural texture analysis techniques.

Wavelet-based texture features can be used successfully with *K*-means based classifiers, but the main drawback of the Wavelet Transform is not being translation-invariant, like moment-based texture features proposed by us. We have also tested a 2D Gabor filtering approach for feature extraction, computing a 2D texture feature vector for each pixel. We have obtained good recognition results, but also a very long running time, because of the high complexity of the computation process.

We have also compared the performance of our classification algorithm with other procedures' clustering performances. Thus, we have tested a semiautomatic variant of our clustering process, consisting of performing only one *K*-means procedure and setting interactively the number of texture classes. Obviously, we have got a much faster execution time, but the advantages of the automatic character were lost.

Then, we have considered other automatic classification techniques to be applied for the moment-based feature vectors. Thus, we have replaced the *K*-means algorithm with a *region-growing* based *hierarchical agglomerative* procedure [17], in the classification process, keeping the use of validation indexes. While we have obtained good texture recognition results, we have got also a higher computational complexity and a much slower running time.

We have also tried to replace the entire classification system (both the *K*-means and the validation indexes parts) with another region-growing automatic clustering technique proposed several ago [18]. The obtained segmentation system performs poorer than the approach proposed in this paper. It achieves a lower texture recognition rate and also a slower processing time.

There are many other clustering algorithms which can be applied in the classification stage. Thus, one can use *Fuzzy K-Means* (*FKM*) instead of the *K*-means algorithm. *FKM* improves the performance of *K*-means, running faster and adding more stability. Another solution is replacing the *K*-means based clustering system with a *Self-Organizing Feature Map* (*SOFM*), which could provide better classification results.

VI. Conclusions

An automatic unsupervised texture-based image segmentation system has been provided in this paper. The main contributions of this article are the texture feature extraction approach using centered area moments and the automatic pixel classification procedure.

The performed experiments have produced satisfactory results, proving the effectiveness of the proposed approach. The automatic character of our segmentation technique represents an important advantage of it. This texture segmentation approach can be used successfully in processing large textured image databases, where interactivity cannot be used. Thus, our method can be applied in the texture-based image indexing and retrieval domain [19].

The main disadvantage of our segmentation technique is its high computational complexity. Moment-based featuring requires many processing operations, which means the proposed approach has a high execution time. For this reason, in the near future we intend to develop some low complexity discrete moment computation approaches.

Our future work will focus also on some application areas of this texture-based segmentation technique. Besides the cluster-based image indexing and retrieval domains [19], we intend to approach the texture segmentation based object detection and retrieval. Being very useful in object detection and robotics, our automatic texture recognition system can be successfully applied in the video analysis domain. Thus, we intend to integrate this texture segmentation approach into a more complex video object tracking system.

ACKNOWLEDGMENT

This work was mainly supported by the project PN-II-ID-PCE-2011-3-0027-160/5.10.2011, financed by UEFSCDI Romania. It was supported also by the Institute of Computer Science of the Romanian Academy, Iasi, Romania.

REFERENCES

[1] L. Shapiro, G. Stockman, Computer Vision, New Jersey, Prentice-Hall, 2001, pp. 279-325.
[2] M. Tuceryan, A. K. Jain, *Texture Analysis. Handbook Pattern Recognition and Computer Vision*. Singapore: World Scientific, ch. 2, 1993, pp. 235–276.
[3] M. K. Pietikäinen, "Texture Analysis in Machine Vision", *Series in Machine Perception and Artificial Intelligence*, vol. 40, 2000.
[4] V. Levesque, "Texture segmentation using Gabor filters", in *Center for Intelligent Machines Journal*, 2000.
[5] M. N. Do, M. Vetterli, "Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance", in *IEEE Transactions on Image Processing*, 11:2, February 2002.
[6] T. Barbu, "A Pattern Recognition Approach to Image Segmentation", *Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science*, Volume 4, Number 2, 2003, pp. 143 – 148.
[7] T. Barbu, "An Automatic Graphical Recognition System using Area Moments", *WSEAS Transactions on Computers*, Issue 9, Volume 5, 2006, pp. 2142-2147.
[8] B. Abraham, O. I. Camps, M. Sznaier, "Dynamic Texture with Fourier Descriptors", Proceedings of the 4th International Workshop on Texture Analysis and Synthesis, 2005, pp. 53-58.
[9] M. Tuceryan, A. K. Jain, "Texture Segmentation Using Voronoi Polygons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12, 1990, pp. 211 - 216.
[10] R. M. Haralick, K. Shanmugam, I. Dinstein, "Textural features for image classification", *IEEE Transactions on Systems, Man, and Cybernetics*, SMC - 3, 1973, pp. 610 - 621.
[11] F. S. Cohen, D. B. Cooper, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*, 1987, pp. 195-219.
[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Volume 24, Number 7, 2002, pp. 881-892.
[13] J. Dunn, "Well separated clusters and optimal fuzzy partitions", *Journal of Cybernetics* vol. 4, 1974, pp. 95-104.
[14] D. L. Davies, D. W. Bouldin, "A cluster separation measure", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1 (4), 2000, pp. 224-227.
[15] T. Barbu, "Approximations of the filtering problem via fractional steps method", *Communications in Applied Analysis*, Vol. 8, No. 2, Dynamic Publishers, USA, 2004, pp. 263-278.
[16] T. Barbu, V. Barbu, V., Biga, D. Coca, "A PDE variational approach to image denoising and restoration", *Nonlinear Analysis: Real World Applications*, Volume 10, Issue 3, 2009, pp. 1351-1361.
[17] T. Kurita, "An Efficient Agglomerative Clustering Algorithm for Region Growing", Proc. of IAPR Workshop on Machine Vision Applications, MVA '94, Kawasaki, Dec. 13-15, 1994, pp. 210-213.
[18] T. Barbu, "An Automatic Unsupervised Pattern Recognition Approach", *Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science*, Vol. 7, Number 1, January 2006, pp. 73 – 78.
[19] B. S. Manjunath, W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 18, Number 8, Aug. 1996, pp. 837-842.