Unsupervised Texture Image Segmentation Using MRF-EM Framework

Marzieh Azarian[1]✉, Reza Javidan[2], Mashallah Abbasi Dezfuli[3]

(1) Department of Computer Engineering and Information Technology, Science and Research Branch, Islamic Azad University, Khouzestan-Iran
(2) Department of Computer Engineering and It, Shiraz University of Technology, Shiraz, Iran
(3) Department of Computer Engineering and Information Technology, Science and Research Branch, Islamic Azad University, Khouzestan-Iran

Azarian_2011@yahoo.com; Reza.javidan@gmail.com; M.abbasi@ khouzestan.srbiau.ac.ir

Abstract

Texture image analysis is one of the most important working realms of image processing in medical sciences and industry. Up to present, different approaches have been proposed for segmentation of texture images. In this paper, we offered unsupervised texture image segmentation based on Markov Random Field (MRF) model. First, we used Gabor filter with different parameters' (frequency, orientation) values. The output image of this step clarified different textures and then used low pass Gaussian filter for smoothing the image. These two filters were used as preprocessing stage of texture images. In this research, we used K-means algorithm for initial segmentation. In this study, we used Expectation Maximization (EM) algorithm to estimate parameters, too. Finally, the segmentation was done by Iterated Conditional Modes (ICM) algorithm updating the labels and minimizing the energy function. In order to test the segmentation performance, some of the standard images of Brodatz database are used. The experimental results show the effectiveness of the proposed method.

Keywords: EM algorithm, Image segmentation, Markov Random Field (MRF), Texture image

1. Introduction

Segmentation is an important process in digital image processing which has found extensive applications in several areas such as content-based image retrieval, medical image processing, and remote sensing image processing. It aims to find the homogeneous regions for labeling objects and background. In other words, image segmentation is the process of grouping pixels of a given image into regions with respect to certain features and semantic content.

Segmentation methods are often classified as: region-based, boundary-based and edge-based. Region-based method assigns each pixel to a particular object or region. Boundary-based method attempts only to locate the boundaries that exist between the regions. Edge-based method seeks to identify edge pixels first and then links them together to form the required boundaries [1, 2].

In this study, we use Markov Random Fields (MRF) for segmenting texture images without supervision because we used K-means algorithm for initial segmentation which

1

is unsupervised algorithm for solving clustering problem. MRF is a powerful tool to model the joint probability distribution of the image pixels in terms of local spatial interactions. In [3] a method is proposed, called the Iterated Conditional Modes (ICM), which uses the local MRF for segmenting of the true image from the noisy one. In [4], MRF is used for segmenting the color textured images.

In this paper, we use MRF for segmenting the texture images when there is no prior information about the model parameters. Many methods, assume that the number of classes is known in advance. Here, we use Expectation Maximization (EM) algorithm for parameter estimation. Different approaches such as Maximizer of the posterior marginal (MPM), Simulated Annealing (SA) and ICM are used for optimization of MRF model. In this study, we use ICM algorithm because in comparison to the other two methods, this method is more flexible. The computation is also faster than the others and produces reasonable results [4, 5].

The rest of the paper is organized as follow: in section 2, we review related works briefly. In section 3, the proposed method is given. In section 4, we discuss experimental result. In section 5, finally we conclude the paper.

2. Related Works

In this section, we explain some filters that are used in preprocessing stage. Some well-known implementation approaches in MRF models are reviewed and also we mention the general statement of EM algorithm.

Gabor filter: Gabor filter is one of the filters that is used for denoising and texture analysis. This filter is a multichannel filter and the best choice for preprocessing stage of the image. Gabor filters have directional selective capability in detecting oriented features and fine tuning to specific frequencies and scale. These filters act as low-level oriented edge discriminators and are especially important in filtering out the background noise of the images. Several filters with varying parameters are applied to an image to acquire the response. A Gabor filter is a sinusoid function modulated by a Gaussian and is defined by the following equation:

$$+\exp\left(\frac{2\pi x_{\theta_i}}{\lambda}\right) \quad G(x, y) = \exp\left(\frac{-x_\theta^2 \ -\gamma_\theta^2 \ y_\theta^2}{\sigma^2}\right) \tag{1}$$

Where

$$x_\theta = x.\cos\theta + y.\sin\theta \quad , \quad y_\theta = -x.\sin\theta + y.\sin\theta \tag{2}$$

That σ is the standard deviation of the Gaussian function, λ is the wavelength of the harmonic function, θ is the orientation, and γ is the spatial aspect ratio which is left constant at 1/2. The spatial frequency bandwidth is the ratio σ /λ and is held constant and equal to 0.56 [6].

Gaussian filter: A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically used to reduce image noise and reduce detail. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales. The Gaussian blur is a type of image-blurring filter using a Gaussian function which also expresses the normal distribution in statistics for calculating the transformation applied to each pixel in the image. The equation of a Gaussian function in one dimension is:

2

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

In two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point.

MRF for texture image segmentation: In this subsection, we offer a brief overview of the MRF theory. MRF is n-dimensional random process defined on a discrete lattice. Usually the lattice is a regular 2-dimensional grid in the plane. In any MRF model, the probability distribution at any site depends only upon its neighborhood [7]. The Hamersley-Clifford theorem states that the joint probability distribution of any MRF can be written as a Gibbs distribution. Moreover for any Gibbs distribution, there exists an MRF for which there is a joint. That is to say, Hamersley-Clifford establishes the equality of the MRF and Gibbs models:

$$p(x) = \frac{1}{Z} e^{(-U(x))} \tag{5}$$

Where x is the random field, Z is the normalization constant and the energy function U(x) is defined as:

$$U(x) = \sum_{c \in C} V_c(x) \tag{6}$$

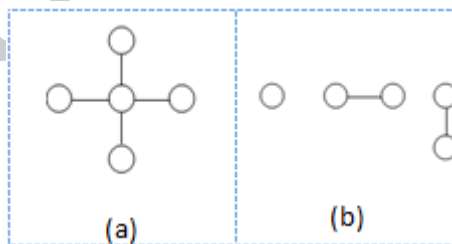Where V(x) is clique potential or potential function and C is the set of all possible cliques.

Figure 1. (a) The first-order neighborhood system. (b) All cliques for first-order neighborhood system

We assume that the image is define on an H × W rectangular lattice   Y={y = (i , j) | $1 \leq i \leq H$, $1 \leq j \leq W$ }and c is a set of pixels, called a clique that consists of either a single pixel or a group of pixels. Figure 1-(a) shows the first-order neighborhood system and figure 1-(b) all cliques for first-order neighborhood system. In the image domain, we assume that one pixel has at most 4 neighbors, the pixels in its 4-neighborhood. Then the clique potential is defined based on pairs of neighboring pixels:

3

$$v_c\left(x_i, x_j\right) = \frac{1}{2}\left(1 - I_{x_i, x_j}\right) \quad I_{x_i, x_j} = \begin{cases} 0 & x_i \neq x_j \\ 1 & x_i = x_j \end{cases} \tag{7}$$

If the pairs of neighboring pixels are the same, $V_c(\text{x}) = 0$ And if are not $V_c(\text{x}) = \frac{1}{2}$

Different kinds of MRF's pixel labeling algorithms: In this subsection, we offer a brief overview of three MRF's implementation approaches. In general, three approaches, SA, MPM, ICM, are used for implementation of MRF. These iterative algorithms attempt to optimize a statistical criterion by approximating the MAP estimate. SA (simulated annealing) is one way for MAP estimation of the true image which minimizes the energy function. An exhaustive search for a global optimum provides an impossible computational burden because the labels for all pixels must be estimated simultaneously. Although simulated annealing is theoretically guaranteed to find a globally optimal labeling, it may fail in actual problems because compromises are needed to overcome the computational burden. Simulated annealing is in the class of stochastic relaxation algorithms and is based on the classical Metropolis method of simulating systems containing large numbers of particles.

ICM algorithm is a computationally feasible alternative for MAP estimation. The computation is a few order of magnitude faster than the simulated annealing approach. ICM algorithm is conceptually simple and computationally viable. The ICM algorithm works on a pixel-by-pixel basis, accepting only those changes which take us successively nearer to our goal of minimizing the potential. This is in contrast to other approaches, such as simulated annealing which allows temporary increases in the potential function to achieve the overall goal of minimization.

MPM is used for the true labels but avoids computational difficulties inherent in MAP estimation because it causes to minimize segmentation error. The labels that minimize segmentation errors can be shown to maximize the marginal a posteriori distribution. The point of departure of MPM is the manner in which this marginal conditional distribution is computed. When Markov chain has reached steady state, the marginal a posteriori probability is estimated by counting the number of times each label is achieved at each pixel in a series of configuration. The MPM algorithm requires more computation than the ICM algorithm, but far less computation than simulated annealing.

Application of EM algorithm: In many practical learning settings, only a subset of the relevant features might be observable. Many approaches have been proposed to handle the problem of learning in the presence of unobserved variables. If some variable is sometimes observed and sometimes not, then we can use the cases for which it has been observed to learn to predict its values when it is not observed. The EM algorithm is widely used to learning in the presence of unobserved variables. The EM algorithm can be used even for variables whose value is never directly observed, provided that the general form of the probability distribution governing these variables is known. The EM algorithm has been used to train Bayesian belief networks as well as radial basis function networks. The EM algorithm is also the basis for many unsupervised clustering algorithms and it is the basis for the widely used Baum-Welch forward-backward algorithm for learning Partially Observable Markov Models. The EM algorithm can be used for the problem of estimating means of a mixture of Normal distributions. More generally, the EM algorithm can be applied in many settings where we wish to estimate

4

some set of parameters that describe an underlying probability distribution, given only the observed portion of the full data produced by this distribution. We use *h* to denote the current hypothesized values of the parameters, and *h′* to denote the revised hypothesis that is estimated on each iteration of the EM algorithm. The EM algorithm searches for the maximum likelihood hypothesis *h′* by seeking the *h′* that maximizes E[ln P(Y (h'))]. In its general form, the EM algorithm repeats the following two steps until convergence:

Step 1: *Estimation* (E) *step:* Calculate Q(h′|h) using the current hypothesis *h* and the observed data X to estimate the probability distribution over Y*.*

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X] \tag{8}$$

Step 2: *Maximization (M) step:* Replace hypothesis *h* by the hypothesis *h′* that maximizes this Q function.

$$h \leftarrow \arg\max_{h'} Q(h'|h) \tag{9}$$

3. The Proposed Method

In this study, we used two filters for preprocessing. First, we used Gabor filter for denoising and then used Gaussian blur for smoothing. When using Gabor filter, we examined different orientations and frequencies to get better results. After preprocessing stage, we generated an initial segmentation using k-means clustering. The initial segmentation provided the initial labels for segmentation algorithm and the initial parameters for EM algorithm. Then we ran EM algorithm to estimate parameters such as μ, σ and run ICM algorithm to generate final labels. Block diagram of segmentation algorithm is shown in figure 2.

Given an image $y = \{y_1, y_2,....,y_N\}$ where each y_i is the intensity of a pixel, we want to infer a configuration of labels $x = \{x_1, x_2,......,x_N\}$ where $x_i \in L$ and L is the set of all possible labels[8].for binary segmentation we use $L = \{0,1\}$ and for multi texture image such as three texture image, we use $L = \{0,1,2\}$. The problem of MRF segmentation is to find the most possible label for each pixel that is named MAP estimation that seeks the labeling x^* which satisfies:

$$x^* = \arg\max_{x \in l}(p(y|x)) = \arg\min_{x \in l}(U(y|x), \Theta) + U(x) \tag{10}$$

The prior probability P(x) is a Gibbs distribution, and the joint likelihood probability is:

$$P(y|x, \Theta) = \prod_i p(y_i|x, \Theta) = \prod_i p(y_i|x_i, \theta_{x_i}) \tag{11}$$

Where $p(y_i|x_i, \theta_{x_i})$ is a Gaussian distribution with parameters $\theta_{x_i} = (\mu_{x_i}, \sigma_{x_i})$. $\Theta = \{\theta_l | l \in L\}$ is the parameter set, which is obtained by the EM algorithm. In the second equation of x^*, the likelihood energy is:

$$U(y|x, \Theta) = \sum_i U(y_i|x_i, \Theta) = \sum_i \left[\frac{(y_i - x_i)^2}{2\sigma_{x_i}^2} + \ln \sigma_{x_i}\right] \tag{12}$$

And the prior energy function U(x) has the form of equation (6).

5

3.1 EM algorithm

For estimating the probability distributions of the labels in the observed image, we need to estimate the mean μ and the variance σ^2 of the each class label [9]. EM algorithm consists of an E-step and an M-step and it starts with initial values μ^0 _and_ σ^0 for the parameters and iteratively performs these two steps until convergence. We suppose the parameter set $\Theta = \{ \theta_l \,|\, l \in L\}$ that $\theta_l = (\mu, \sigma)$.

1) Start: Assume we have an initial parameter set Θ^0.

2) E-step (expectation): At the tth iteration, we have Θ^t and we calculate the conditional expectation:

$$Q(\Theta \,|\Theta^{(t)}) = E\left[\ln P(x, y|\Theta)|y, \Theta^{(t)}\right] \quad =\sum_{x\in X} P(x|y, \Theta^{(t)})\ln P(x, y|\Theta) \qquad (13)$$

3) M-step (maximization): Now maximize $Q(\Theta \,|\Theta^{(t)})$ to obtain the next estimate:

$$\Theta^{(t+1)} = \text{argmax } Q(\Theta \,|\Theta^{(t)}) \qquad (14)$$

Then let $\Theta^{(t)} \rightarrow \Theta^{(t+1)}$ and repeat from the E-step.

3.2 ICM algorithm: minimizing U (y|x)

ICM is an optimization method. We have developed this iterative algorithm to solve the MAP estimation. The final step in the segmentation is to find a viable means to minimize U(y|x). The ICM algorithm is an example of a 'greedy algorithm', as it works on a pixel-by-pixel basis, accepting only those changes which take us successively nearer to our goal of minimizing the potential. This is in contrast to other approaches, such as simulated annealing which allows temporary increases in the potential function to achieve the overall goal of minimization [10].

1) To start, we have an initial estimate $x^{(0)}$, which is from the previous loop of the EM algorithm.

2) Provided $x^{(k)}$, for all $1 \leq i \leq N$, we find

$$x_i^{(k+1)} = \text{argmin}_{l\in L}\{U(y_i|\, l) + \sum_{j\in N_i} V_c\,(l, x_j^{(k)})\} \qquad (15)$$

3) Repeat step 2 until U(y|x , θ)+ U(x) converges or a maximum k is achieved.

```
                          ┌─────────────────┐
                          │   Input image   │
                          └─────────────────┘
                                   │
                          ┌──────────────────────┐
                          │ Filtering with Gabor filter │
                          └──────────────────────┘
                                   │
                          ┌─────────────────┐
                          │  Filtered image │
                          └─────────────────┘
                                   │
                          ┌────────────────────────────┐
                          │ Smoothing using Gaussian filter │
                          └────────────────────────────┘
                                   │
                          ┌─────────────────────┐
                          │ K-means clustering  │
                          └─────────────────────┘
                                   │
                          ┌─────────────────────────┐
                          │ Initial segmented image │
                          └─────────────────────────┘
                                   │
                          ┌─────────────────┐
                          │  EM algorithm   │
                          └─────────────────┘
                                   │
                          ┌───────────────────────┐
                          │ Parameter estimation  │
                          └───────────────────────┘
                             │              │
                     ┌──────────┐      ┌──────────────────┐
                     │   mean   │      │ Standard variation│
                     └──────────┘      └──────────────────┘
                                   │
                          ┌─────────────────┐
                          │ Feature vector  │
                          └─────────────────┘
                                   │
                          ┌─────────────────┐
                          │ Segmenting using│
                          │      MRF        │
                          └─────────────────┘
                             │              │
                  ┌──────────────┐    ┌──────────────┐
                  │Final Segmented│    │  Diagram of  │
                  │    image      │    │energy function│
                  └──────────────┘    └──────────────┘
```
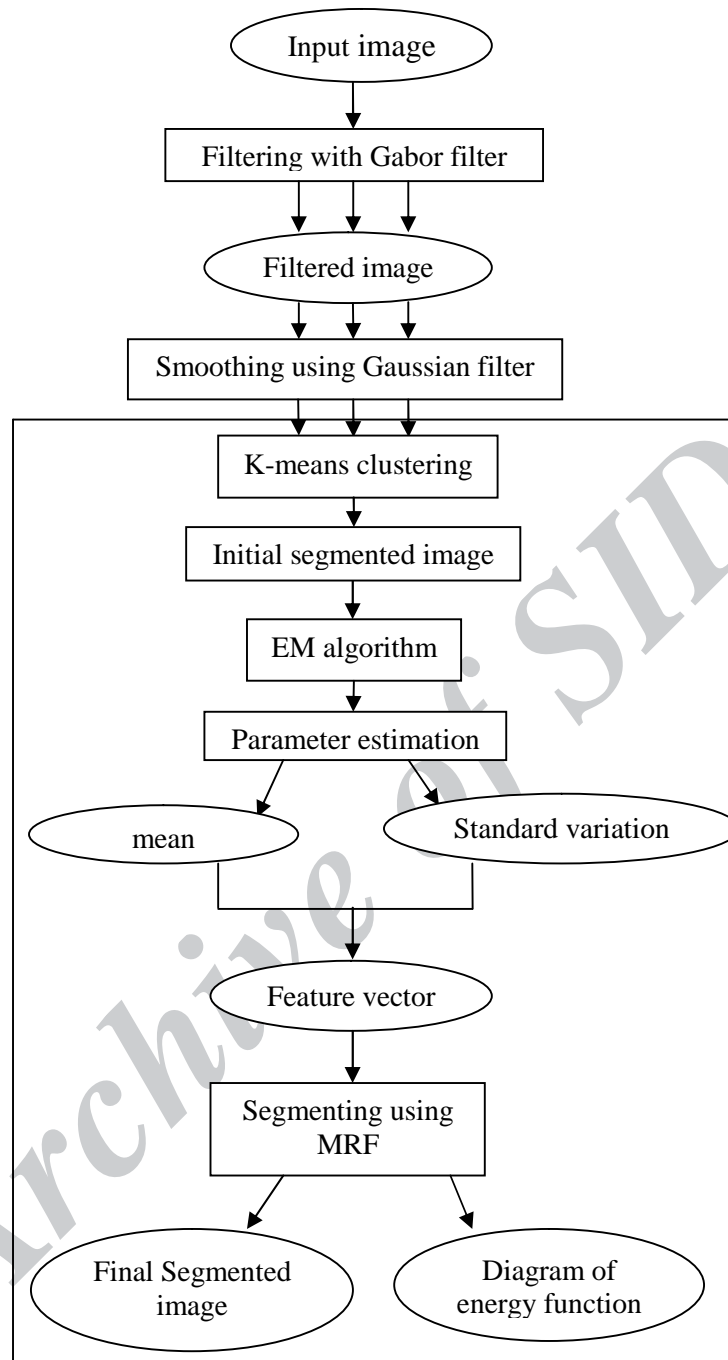
Figure2 .Diagram of Segmentation algorithm

4. Experimental study and results

In this study, an unsupervised texture image segmentation schema is proposed based on ICM algorithm. The parameters of the components are estimated by using the EM algorithm. For testing the performance of the unsupervised segmentation algorithm, we generate several Brodatz texture images of size 256×256 on MATLAB

7

environment. First we use Gabor filter and then Gaussian blur to denoise the original image. It's necessary to mention that when we use Gabor filter, we should examine each image with different value of orientation and frequency and the specific value of these parameters leads us to favorable results. We generate an initial segmentation using k-means clustering. The initial segmentation provides the initial labels for MAP estimation and the initial parameters for EM algorithm. So, before running ICM algorithm for segmentation, we should use EM algorithm for estimating parameters of each class label. In this study, we use 10 iteration for EM algorithm and 10 iteration for ICM algorithm. Eventually, final labels are obtained by running of EM & ICM algorithm. Total processes of algorithm on some of Brodatz image, are shown in figure 4. Another output of these algorithms is a diagram that shows total posterior energy in each iteration of the EM algorithm.

	Original image	Output of Gabor filter	Output of Gaussian filter	Segmented image using ICM: final labels
a				
b				
c				
d				
e				

Figure 3.Total processes of segmentation algorithm using Brodatz image.(a) θ *=135° ,* ω *=0.12 (b)* θ *=135° ,* ω *=0.12 (c)* θ *=135° ,* ω *=3.3 (d)* θ *=135° ,* ω *=0.4 (e)* θ *=135° ,* ω *=0.3*

In the following, figure 4 shows diagram of the total posterior energy in each iteration of the EM algorithm which is related to labels b & d in figure 3.
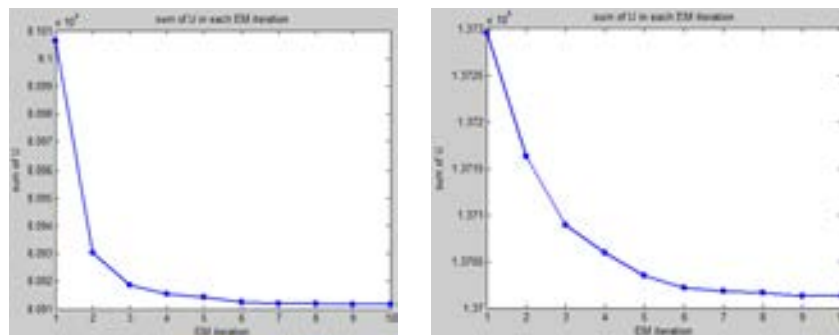

Figure4 .Diagram of the total posterior energy in each iteration of the EM algorithm. (a) Diagram of label b in figure3. (b) Diagram of label d in figure 3.

Because we don't have any specific criterion to calculate the error percent, we use Mean square error approach to calculate errors. Table 1 shows the error percent and running time of segmented image in figure 3.

Table1.Error percent & running time of image in figure 3

Segmented image	a	b	c	d	e
Error percent	27%	6%	24%	15%	16%
Running time(s)	825	118	122	343	320

4.1 Compare with previous approaches

In this subsection, we discuss about the performance and error percent of our proposed method with previous methods in image segmentation using MRF. Deng and Clausi in [2] used SA algorithm for segmentation and EM algorithm for parameter estimation. The initial values for EM algorithm are selected randomly. Sengur in [7], used ICM algorithm for segmentation and EM algorithm for parameter estimation. The initial values for EM algorithm are selected randomly and the initial values for labels in ICM algorithm determine with MLE approach. Kato and Pong in [4], used ICM algorithm and Gibbs sampler for segmentation and EM algorithm for parameter estimation. In this study, combined feature is used for segmentation .The initial values for EM algorithm are selected randomly and the initial values for labels in ICM algorithm determine with MLE approach but in the proposed method in this paper, we used k-means algorithm for initial segmentation and determining the initial value of parameters in both algorithms. It should be mentioned that, the proposed method used two filters in preprocessing stage and this stage caused final result that was more acceptable. Figure 5 shows the result of proposed method and the result of Deng and Clausi method in SAR image. The result of proposed method is closer to Deng and Clausi method when used variable weighting parameter. Figure 6 shows the result of the

9

proposed method and the result of Sengur method in real world image. Figure 7 shows the result of proposed method and the result of Kato and Pong method in real image.
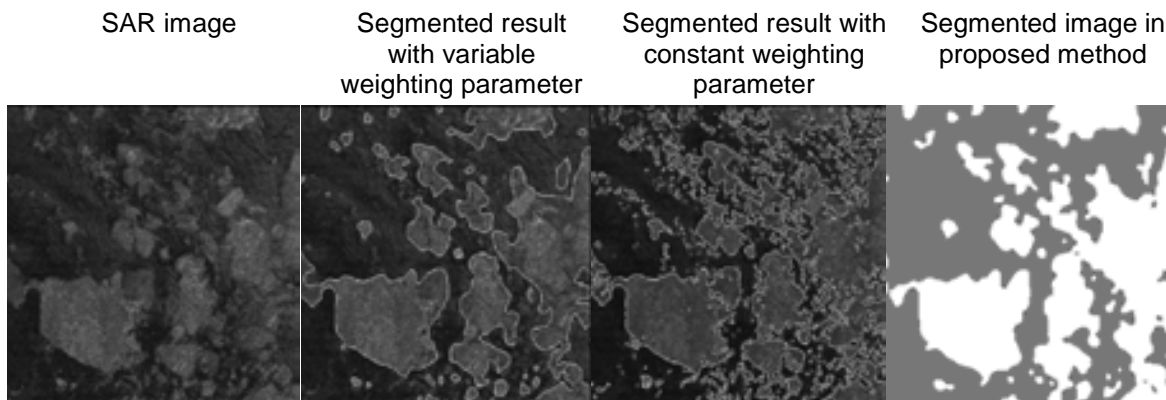
| SAR image | Segmented result with variable weighting parameter | Segmented result with constant weighting parameter | Segmented image in proposed method |

Figure 5. Compare the proposed method with Deng and Clausi method

| Real world image | Segmented image in Sengur method(first order neighborhood system) | Segmented image in proposed method |
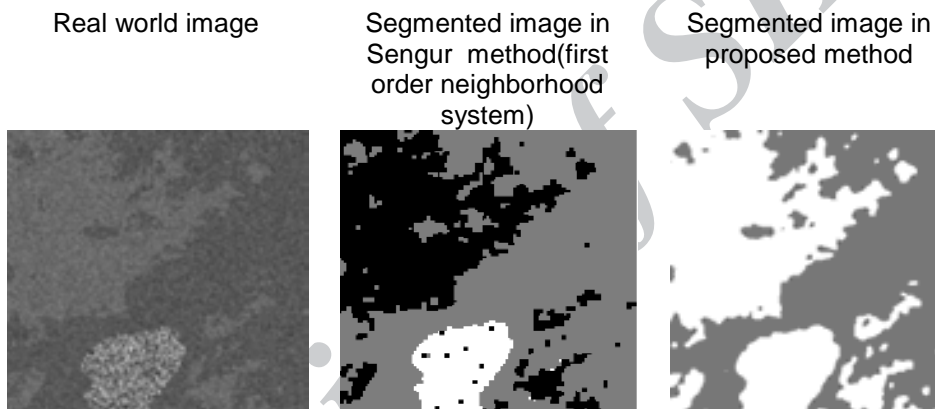
Figure 6. Compare the proposed method with Sengur method in first order neighborhood system

| Real image | Segmented image in Kato method using combined feature | Segmented image in proposed method |
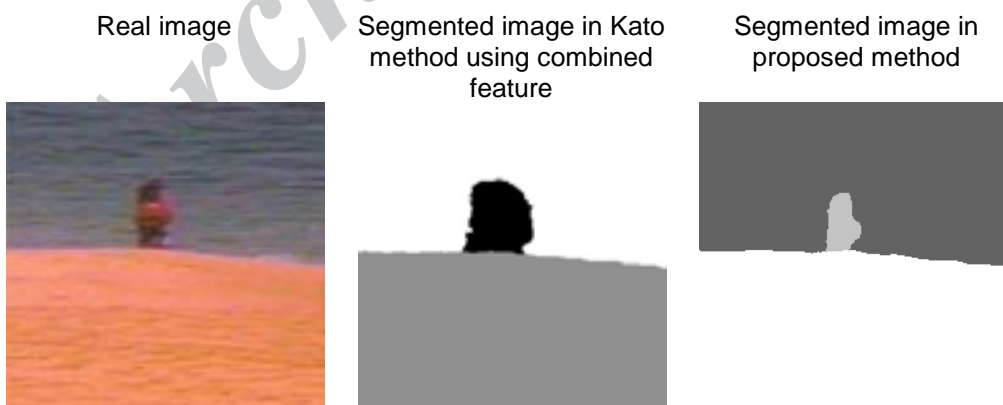
Figure 7. Compare the proposed method with Kato method that using combined features

Table 2 shows the error percent of the proposed method and Deng method. Table 3 shows the error percent of the proposed method and Sengur method. Table 4 shows the error percent of the proposed method and Kato method. Because we don't have any

10

specific criterion to calculate the error percent, we use Mean square error approach to calculate errors. Figure (8, 9, 10) shows the error percent diagrams.

Table 2. Error percent of the proposed method and Deng method

	Deng method with variable parameter	Deng method with constant parameter	Proposed method
Error percent	6%	32%	5%

Table 3. Error percent of the proposed method and Sengur method

	Sengur method	Proposed method
Error percent	21%	4%

Table 4. Error percent of the proposed method and Kato method

	Kato method	Proposed method
Error percent	15%	7%
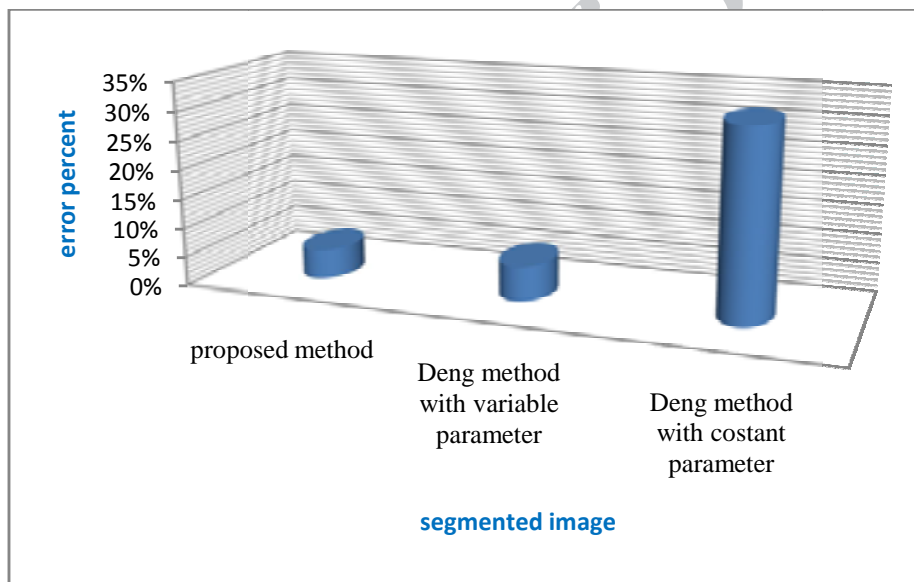
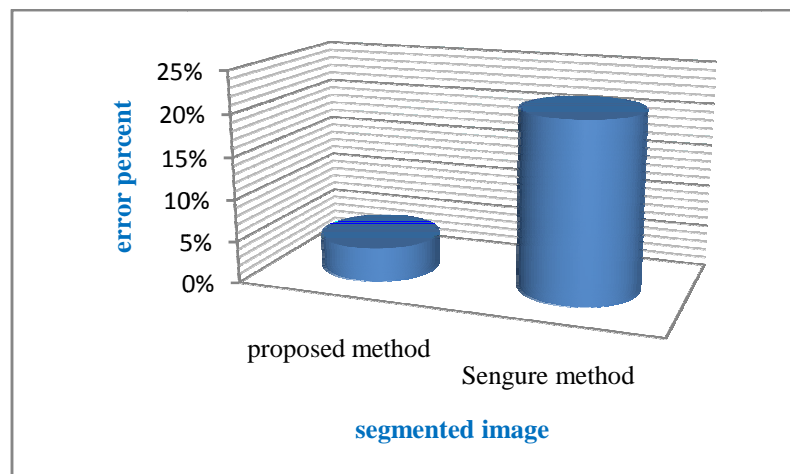*Figure8. Error percent diagram of the proposed method and Deng method*

Figure9. Error percent diagram of the proposed method and Sengur method
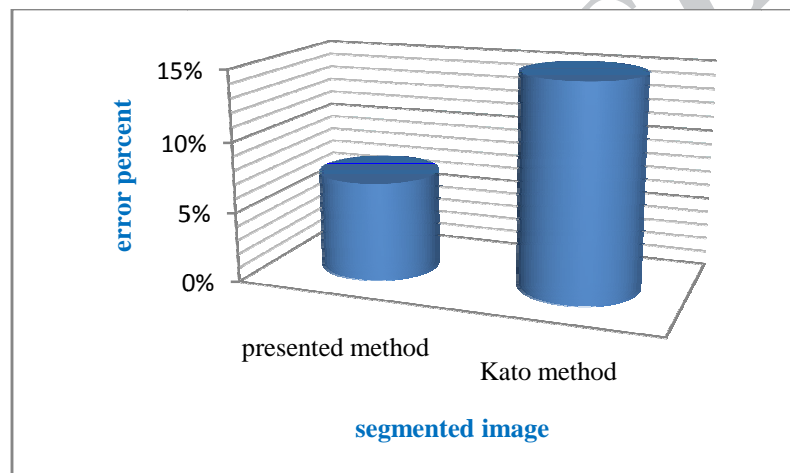

Figure10. Error percent diagram of the proposed method and Kato method

5. Conclusion and Remarks

In this paper, we have examined an unsupervised texture image segmentation algorithm. The segmentation model is defined in a MRF framework. The examined algorithm is fully unsupervised because we used K-means algorithm for initial segmentation which is one of the simplest unsupervised learning algorithms that solve clustering problem. To estimate the component parameters, we use an iterative EM algorithm. Then we use ICM algorithm for completing the segmentation procedure. The algorithm has been tested on a variety of Brodatz texture image and results are very satisfactory. In comparison with Sengur method, Deng method and Kato method, the proposed method had fewer errors.

6. References

[1] F. Shih, "Image processing & pattern recognition, pp. 119-120, 2010.
[2] H.Deng and D.A.Clausi, "Unsupervised image segmentation using a simple MRF model with a new implementation scheme, *Pattern Recognition*,pp. 2323-2335, 2004.

12

[3]   J.Besag, "On the statistical analysis of dirty pictures," *J.Roy.Statist Soc.48*, pp. 259-302, 1986.

[4]   Z.Kato and T.Pong, "A Markov random field image segmentation model for color textured images," *image and vision computing*, pp. 1103-1114, 2006.

[5]   R.C.Dubes,A.K.Jain, S.G.Nadabar,C.C.Chen "MRF model-based algorithm for image segmentation, " *IEEE*, pp.808-814,1990.

[6]   N.Petkov,S.E.Grigoresco "comparison of texture features based on gabor filters ",*proceeding of the 10th international conference on image analysis and processing*,pp.142-147, 1999.

[7]   A.Sengur,I.Turkoglu and M.Cevdet "Unsupervised image segmentation using Markov random field ", *Springer- Verlag Berlin Heidellberg*, pp.158-167,2006.

[8]   Q.Wang, "HMRF-EM-image:Implementation of the Hidden Markov Random Field Model and its Expectation – Maximization Algorithm ", 2012.

[9]   T.M.Mitchell, "machine learning".pp.191-196, 1997.

[10] C.Solomon, T.Breckon, "Fundementals of digital image processing", first edition by JOHN WILEY and sons, ltd, 2011.