Texture Segmentation Using Gaussian Markov Random Fields and LEGION

Erdogan Çesmeli[†] and Deliang Wang[‡]
[†]Center for Biomedical Engineering
cesmeli@cis.ohio-state.edu
[‡]Department of Computer and Information Science and Center for Cognitive Science
dwang@cis.ohio-state.edu
The Ohio State University, Columbus, OH 43210, USA

Abstract

An image segmentation method is proposed for texture analysis. The method is composed of two main parts. The first part determines a novel set of texture features based on Gaussian Markov Random Field (GMRF). Unlike other GMRF-based methods, our method is not limited by a fixed set of texture types. The second part is LEGION (Locally Excitatory Globally Inhibitory Oscillator Networks) which is a 2D array of neural oscillators. The coupling strengths between neighboring oscillators are calculated based on texture feature differences. When LEGION is simulated, the oscillators corresponding to the same texture tend to oscillate in synchrony, whereas different texture regions tend to attain different phases. Results demonstrating the success of our method on real texture images are provided.

1. Introduction

Image segmentation is a fundamental and difficult task in machine vision. Texture is one of the visual features believed to have an important role in scene segmentation. Intuitively, texture is related to patterned variations of intensity across an image. It is, therefore, natural to deduce that texture is a region property, not a point property. Accordingly, models have been proposed to capture features from a region of various sizes. Among several other models, Markov Random Fields (MRF) have been shown to be quite successful for texture modeling [4, 6, 8]. This model captures the local characteristics of an image by determining the dependency of one pixel on its neighboring pixels. It is the structure of this dependency which is hypothesized as a Markov Random Field with a certain set of parameters. Based on GMRF parameters a new set of texture features is proposed in our method. The differences of the proposed texture features between different textures make it possible to discriminate one texture from another. LEGION [9], which exhibits emergent properties based on local computations, has been proposed to deal with image segmentation. LEGION is based on the idea of oscillatory correlation [7, 9], whereby phases of neural oscillators encode region labeling. Oscillators corresponding to one region have the same phase which is different from that of other regions [12]. The coupling strengths among the oscillators have a critical role in LEGION's behavior. In this paper, we describe a method which combines GMRF and LEGION for texture segmentation.

2. Model components

2.1 Markov random field

MRF modeling has found several applications in texture processing, including texture synthesis, classification, and segmentation [8]. In MRF an image is represented by determining the local dependency structure of pixel intensities. The notion of locality is captured by means of a local conditional probability distribution. In a general framework of MRF, there are six basic steps. First, a structural form for the local conditional probability distribution is assumed, such as a Gaussian distribution. Second, the joint probability distribution of the entire image is determined. This step obtains a global representation of the image in terms of local information. Third, the process by which the observations are generated from the true image is captured in degradation probability distribution. Following that, the posterior probability distribution of the image given the observations is determined using the Bayes theorem. After the parameters of MRF are estimated from the training data, a cost can be associated with each configuration. A measure of the cost is finally minimized with respect to image intensities or image attributes [8].

2.1.1. Model description. If the structural form for the local conditional probability distribution in MRF is assumed to be Gaussian, the model is called Gaussian Markov Random Field (GMRF). The size and the topology of the local dependence are designated as the order of the model. For example, a second order MRF contains

2	1	2
1	C	1
2	1	2

Figure 1. Structure of a GMRF model. The numbers indicate the order of the model relative to the center pixel, *C*.

eight-nearest neighbors (Fig. 1). Before introducing the framework of the model, some definitions are provided. Let $\Omega = \{(i, j), 1 \le i, j \le M\}$ denote the set of grid points in the $M \times M$ lattice corresponding to the image. Let $\{L_c, c \in \Omega\}$ and $\{Y_c, c \in \Omega\}$ denote the labels and an array of gray levels with zero mean, respectively. Let N_c be the symmetric second order neighborhood of a site C (Fig. 1). Assuming that all the neighbors of C have the same label as that of C, we can write the following expression for the conditional probability density of the intensity at pixel site C:

$$P(Y_c = y_c | Y_r = y_r, r \in N_c, L_c = l)$$
$$= \frac{exp\left(-U'(Y_c = y_c | Y_r = y_r, r \in N_c, L_c = l)\right)}{Z'(l | y_r, r \in N_c)} \quad (1)$$

where $Z'(l | y_r, r \in N_c)$ is the partition function which is the sum of the numerator for all possible realizations of N_c and

$$U'\left(Y_c = y_c | Y_r = y_r, r \in N_c, L_c = l\right)$$
$$= \frac{1}{2\sigma_l^2}\left(y_c^2 - 2\sum_{r \in N_c} \theta_{r,c}^l y_r y_c\right) \quad (2)$$

where σ_l and $\theta_{r,c}^l$ are the GMRF model parameters of the texture type l. The model parameters satisfy $\theta_{r,c}^l = \theta_{r-c}^l = \theta_{c-r}^l = \theta_\tau^l$. Following that, the joint probability in a window R_c can be written as

$$P\left(\vec{Y}_c | L_c = l\right) = \frac{\exp\left(-U(\vec{y}_c | L_c = l)\right)}{Z(l)} \quad (3)$$

where $Z(l)$ is the partition function and

$$U\left(\vec{y}_c | L_c = l\right) = \sum_{c \in R_c} U'\left(Y_c | Y_{c+\tau_i}, L_c\right)$$

$$= \frac{1}{2\sigma_l^2}\sum_{c \in R_c}\left(y_c^2 - \sum_{\tau_i \in N^* | c+\tau_i \in R_c} \theta_i^l y_c\left(y_{c+\tau_i} + y_{c-\tau_i}\right)\right) \quad (4)$$

where $N^* = \{\tau_1, \tau_2, \tau_3, \tau_4\}$
$= \{(0, 1), (1, 0), (1, 1), (-1, 1)\}$

is a set of shift vectors corresponding to the second order GMRF model and $\vec{y}_c$ represents the intensity array in the window R_c.

2.1.2. Model parameter estimation. It is our assumption that the differences between texture features based on a GMRF model make segmentation possible. In LEGION, coupling strengths among the neighboring oscillators have a critical role in segmentation. Hence, we determine the coupling strengths using GMRF-based texture features. Except for the above assumption, our approach differs from other approaches using MRF. In other approaches, an image which contains only one type of texture is analyzed and the GMRF parameters particular to that texture are estimated and stored prior to segmentation. Parameter estimation for other texture types is repeated as to obtain a fixed set of texture types whose parameters are known *a priori*. Therefore, images which have only these textures can be segmented. Our approach, on the other hand, does not need prior texture parameter knowledge for different texture types. During the segmentation process, texture features are estimated on-line within windows which do not necessarily contain only one type of texture. Thus, unlike others, our approach is not limited by a fixed set of texture types.

There are several methods for estimating GMRF parameters. Among them are maximum likelihood (ML) estimation, coding scheme [1], and least square estimation. None of the methods guarantees both consistency (estimates converging to the true values of the parameters) and stability (positive definite covariance matrix in the expression of the joint probability density function) [3]. We adopt the least square estimate [1, 6] among other methods of estimation, because it is easy to compute [8].

In the least square estimation method, a window of size $u \times u$ containing a single texture is considered. Let Ω be the lattice, Ω_I be the interior region of Ω, and Ω_B be the boundary, i.e., $\Omega_I = \Omega - \Omega_B$ where

$$\Omega_B = \{c = (i, j) \text{ and } c \in \Omega, c + \tau_i \notin \Omega$$
$$\text{for at least some } \tau_i \in N^*\} \quad (5)$$

Let $Q_c = [y_{c+\tau_1} + y_{c-\tau_1}, \ldots, y_{c+\tau_4} + y_{c-\tau_4}]^T$. Then, define $\hat{\Theta} = [\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4]^T$ as the least square estimate of parameter vector [1]:

$$\hat{\Theta} = \left[\sum_{\Omega_I} Q_c Q_c^T\right]^{-1} \left[\sum_{\Omega_I} Q_c y_c\right] \tag{6}$$

We modify this solution slightly and introduce a new set of texture features given by:

$$\phi_i = \frac{1}{u^2} \sum_{\Omega_I} \left[y_c - \hat{\theta}_i Q_c(i)\right]^2 \tag{7}$$

where $i = 1, 2, 3$, or 4. Unlike other approaches, we use ϕ_i's instead of $\hat{\theta}_i$'s as the texture feature set.

2.2 LEGION

LEGION is based on the idea of oscillatory correlation, where the phases of the neural oscillators encode the binding of the features [9, 11]. A single oscillator i of LEGION is defined as a feedback loop between an excitatory unit x_i and an inhibitory unit y_i:

$$\frac{dx_i}{dt} = 3x_i - x_i^3 + 2 - y_i + I_i + S_i + \rho \tag{8a}$$

$$\frac{dy_i}{dt} = \varepsilon\left(\gamma\left(1 + \tanh\left(\frac{x_i}{\beta}\right)\right) - y_i\right) \tag{8b}$$

where ρ denotes the variance of a Gaussian noise term. I_i represents external stimulation to the oscillator. The x-nullcline of (8a) is a cubic curve while the y-nullcline of (8b) is a sigmoid function, as shown in Figure 2. If these curves intersect along the middle branch of the cubic nullcline, then the system is oscillatory. If the nullclines intersect at a point along the left branch of the cubic (a

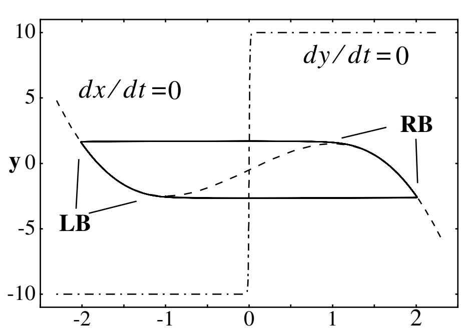


Figure 2. The limit cycle of a single oscillator. The x-nullcline is the dashed line, and the y-nullcline is the dash-dot line. The solid line is the limit cycle of the oscillator. The parameters are $\varepsilon = 0.005$, $\gamma = 10.0$, $\beta = 0.02$, $I_i = -2.0$, and $\rho = 0.02$.

stable fixed point), the system does not oscillate. γ and β are system parameters to control the steepness of y-nullcline. S_i denotes coupling from other oscillators in the network:

$$S_i = \sum_{k \in N(i)} W_{ik}\, H(x_k - \theta_k) - W_z\, H(z - \theta_{xz}) \tag{9}$$

where W_{ik} is the connection weight from oscillator k to oscillator i, H is the Heaviside step function, and $N(i)$ represents the neighborhood topology in the array which is 8-nearest neighbors around oscillator i (Fig. 3). The parameter ε is chosen to be small so that (8) defines a

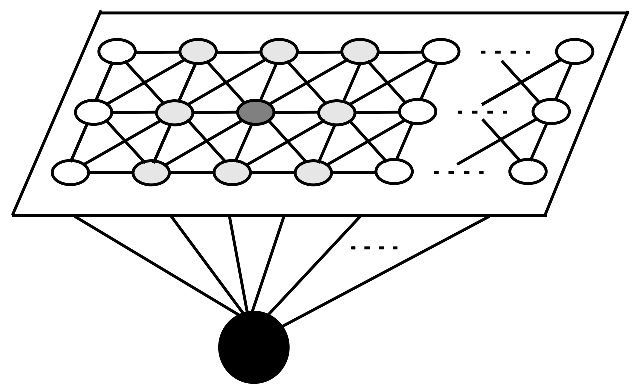


Figure 3. Architecture of a 2-D LEGION with 8-Nearest Neighbor coupling. The global inhibitor is indicated by the black circle.

relaxation oscillator with two time scales. θ_k and θ_{xz} are thresholds, and W_z is the weight of the inhibitory connection from the global inhibitor z, defined as

$$\frac{dz}{dt} = \varphi(\sigma_\infty - z) \tag{10}$$

where $\sigma_\infty = 1$ if $x_i \geq \theta_{zx}$ for at least one oscillator and $\sigma_\infty = 0$ otherwise, and φ is a constant. The neural network structure used in our image analysis is a two dimensional array of oscillators and one global inhibitor as shown in Figure 3. LEGION has been applied to intensity image segmentation [12].

3. The Model: GMRF and LEGION

It is well known that texture perception is a local process rather than a global one [5]. This feature is naturally incorporated into LEGION which functions primarily through local couplings. Determining the weights of LEGION is crucial for the network to perform texture segmentation. In our method, the weights are formed based on GMRF modeling.

First, for each location in the input image a window of size $u \times u$ is considered for GMRF-based texture feature estimation. Consequently, a texture feature vector, $\Phi = [\phi_1, \phi_2, \phi_3, \phi_4]^T$, is determined for each location except for those locations close to the boundary of the image where estimation windows cannot be formed. Therefore, four new feature images each corresponding to one component of texture feature vector are obtained. Following the extraction of texture feature vectors, these feature images are processed by an edge preserving noise suppressing quadrant filter (EPNSQF) [10]. Among five subwindows formed within a filtering window, the one with a minimum variance becomes the winner (Fig. 4). Therefore, center pixel of the filtering window is given a value of the mean of the winner subwindow. As an extension to the original

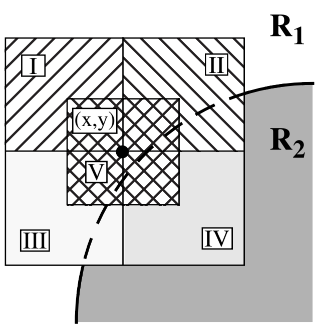


Figure 4. Five rectangular windows of the same size around the point (x, y) shown as the black dot. In this case, the point is located in region R_1 and close to the boundary of the neighboring region R_2. Region information is not known when processing is performed.

EPNSQF, four feature images are processed simultaneously in our analysis. In this filtering process, five subwindows for each location are formed in each feature image. Five subwindows are positioned same in each feature image. Mean and variance of each component of the feature vector within each subwindow are determined. Variances corresponding to a set of four subwindows with the same positioning in four feature images are summated to obtain the total variance of that subwindow set. Following that, five sets of four subwindows compete with each other so that the one with the minimum total variance becomes the winner set. A mean feature vector, where each component is the mean from one subwindow in the winner set is formed. As a result, the feature vector of the current location is replaced by the mean feature vector based on the winner set. This process is repeated for each location.

Once the feature vectors are filtered by EPNSQF, the weights necessary for LEGION can be calculated. We employ dynamic weights in the process. Change in the weights between two oscillators continues until the two oscillators stabilize their phase relations. The weights between two oscillators depend on local feature statistics of two oscillators defined within a neighborhood of each oscillator. When the local average of the texture feature vectors stays the same, the weights in that neighborhood do not change. In the realization of the network, a simple clustering method which is equivalent to LEGION, is developed and implemented. In this method, randomly selected leader points are allowed to form their regions. During this growing (synchronization) process, a point next to a region labeled l, is recruited by that region if its feature vector satisfies a distance criterion. For this purpose, a neighborhood which lies in the region l and next to the point to be recruited is formed. An average feature vector in this neighborhood is calculated. Following that, a normalized Euclidean distance is determined:

$$\Psi_{l,P} = \frac{\sum_{i=1}^{4}\left(\Phi_{l,P}(i) - \Phi_{P}(i)\right)^2}{\|\Phi_{l,P}\|\|\Phi_P\|} \tag{11}$$

where $\Phi_{l,P}$ and Φ_P are the average feature vector of the neighborhood in the region l next to point P, and the feature vector of the point P, respectively. Thus, when this normalized distance is less than a certain threshold determined for the input image, the point P is included in the region l. In the case where there is more than one region competing for the same point, the region with the minimum distance recruits the point.

Each region is initiated by a randomly selected leader point. This point expands in the way described above until there are no more points left around its boundary satisfying the distance criterion. Selection of leader points and expansion of the regions are repeated until there is a preset number of points left unlabeled. Different labels for different regions correspond to different phases in LEGION.

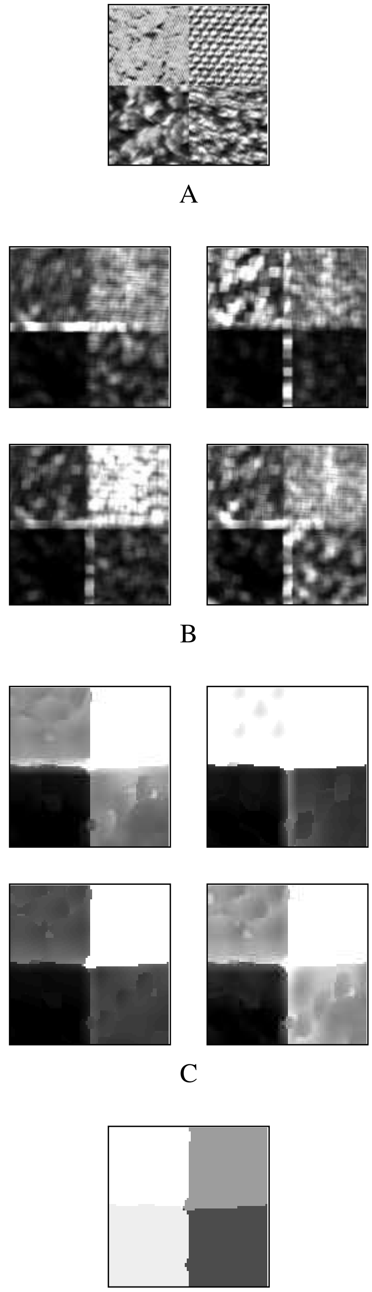
4. Results

To demonstrate the performance of our method, natural textures from the Brodatz Photography Album [2] and MIT Vision Texture Image Archive are used. We present three images of size 128×128 containing different texture types. A second order Gaussian Markov Random Field is used to model the textures.

We illustrate the segmentation process using the example shown in Figure 5A. The example image of size 128×128 is composed of four quadrant regions, each with a different texture. Our method starts with the extraction of texture features across the image. For this purpose, a feature estimation window of size 9×9 is moved across the image. Four GMRF-based texture features are estimated at each position of the window. The estimated parameters form the texture feature vector for the location corresponding to the center location of the estimation window. Because of image boundaries, the resulting 2D parameter vector field has a size of 120×120. One can also view this 2D array of texture feature vectors as four separate feature images (Fig. 5B). Each feature image contains one component of the texture feature vector for all locations.

Following their estimation, texture features are filtered by EPNSQF. A filtering window of size 41×41 is moved across four feature images simultaneously. For each location, means and variances of features for each subwindow of size 21×21 in feature images are calculated. Variances from a set of four subwindows positioned same in four feature images are summated to obtain total variance of each subwindow set. The subwindow set with the minimum total variance becomes the winner among five subwindow sets. A mean feature vector is formed by calculating the means in the winner subwindow set. The texture feature vector of the center location of the filtering window is replaced by the mean feature vector. The filtering window visits all locations across the image. The resulting new feature images due to filtering by EPNSQF for the example in Figure 5A are depicted in Figure 5C. Note that while small variations within different texture regions are removed, borders between the regions are mostly preserved. Because of image boundaries, the resulting 2D array of feature vectors has a size of 80×80.

Finally, clustering on the filtered texture features is performed by LEGION to complete segmentation. The following clustering algorithm employed is equivalent to LEGION. Initially, a leader point is randomly selected. Neighbors around the leader point are recruited and labeled if their texture feature vectors lie within a prespecified distance to the texture feature vector of the leader point. New neighbors are recruited if their texture feature vectors satisfy the distance criterion with respect to a local average texture feature vector. A local average feature vector is based on the feature vectors in a labeled region next to a neighboring point to be recruited. The leader point continues to form its region until neighbors surrounding its region boundaries cannot be recruited. Following that, another leader point is selected in an unlabeled part of the image. Selection of a leader point and forming its region



A

B

C

D

Figure 5. A) Input image. B) Each image represents one component of the texture feature vector, namely, by collecting the values corresponding to the same location from different images, texture feature vector corresponding to that location may be formed. C) After applying EPNSQF. D) Resulting segmentation. Each region is artificially assigned with an intensity value for illustration purposes.

are repeated until there are only 200 locations (about 3% of the total number of locations in the filtered image) left unlabeled. For the example image, the segmentation result is shown in Figure 5D. Regions with different labels are assigned with different intensity values for illustration.

Two additional segmentation examples are provided to demonstrate the performance of our method (Fig. 6A). Similar to the example in Figure 5, each input image is of size 128×128 and composed of four quadrant regions each with a different texture. Segmentation results are provided in Figure 6B.

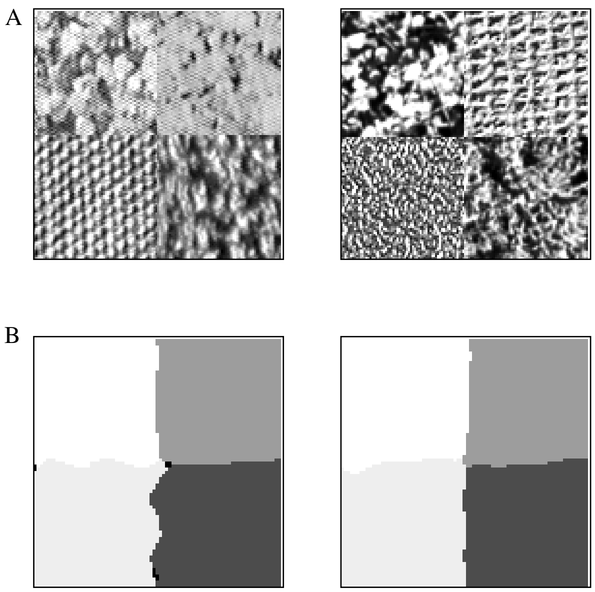


Figure 6. A) Input images of size 128×128, B) Resulting segmented images of size 80×80.

5. Conclusion

Our method is composed of two parts, GMRF-based texture feature estimation and LEGION, both are based on the notion of local computations which is important for parallel and distributed processing. Four main factors give rise to the performance of our method. The first one is the assumption that the dependency of one pixel on its neighboring pixels can be described by a second order GMRF. Second, a different set of texture features based on GMRF is proposed. Third, a multidimensional version of EPNSQF is suggested and employed. Finally, a similarity criterion is introduced for determining coupling strengths in LEGION.

Acknowledgments

This work is supported in part by an ONR grant (N00014-93-10335), an NSF grant (IRI-9423312), and an ONR Young Investigator Award (N00014-96-1-0676) to DLW.

References

[1] J. Besag and P. A. P. Moran, “On the estimation and testing of spatial interaction in Gaussian lattice processes,” *Biometrika*, 62:555-562, 1975.

[2] P. Brodatz, *Textures: A photographic Album for Artists and Designers*. New York: Dover, 1966.

[3] R. Chellappa, “Two dimensional discrete Gaussian Random Markov Field models for image processing,” *Progress in Pattern Recognition 2*, ed. L. N. Kanal and A. Rosenfeld, pp. 79-112, Elsevier Science Publishers, 1985.

[4] G. R. Cross and A. K. Jain, “Markov Random Field texture models,” *IEEE Trans. Patt. Anal. and Mach. Intel.*, PAMI-5:25-40, 1983.

[5] B. Julesz, *Dialogues on perception.* The MIT press, 1995.

[6] R. L. Kashyap and R. Chellappa, “Estimation and choice of neighbors in spatial interaction models of images,” *IEEE Trans. Inform. Theory*, IT-29:60-72, 1983.

[7] C. von der Malsburg and W. Schneider, “A neural cocktail-party processor,” *Biol. Cybern.*, 54:29-40, 1986.

[8] A. Rangarajan A. and R. Chellappa, “Markov Random Fields model in image processing,” in *The Handbook of Brain Theory and Neural Networks*, ed. by M. Arbib, pp. 564-567, MIT Press, 1995.

[9] D. Terman and D. Wang, “Global competition and local cooperation in a network of neural oscillators,” *Physica D*, 81:148-176, 1995.

[10] F. Tomita and S. Tsuji, “Extraction of multiple regions by smoothing in selected neighborhoods,” *IEEE Trans.*, SMC-7:107-109, 1977.

[11] D. L. Wang and D. Terman, “Locally excitatory globally inhibitory oscillator networks,” *IEEE Trans. Neural Networks*, 6:283-286, 1995.

[12] D. L. Wang and D. Terman, “Image segmentation based on oscillatory correlation,” *Neural Computation*, in press; see also *Tech. Rep. #19*, Center for Cognitive Science, The Ohio State University, 1996.