CAR-TR-860 DAAL01-96-2-0001
CS-TR-3801 May 1997

Image Segmentation and Labeling Using the Polya Urn Model

Amit Banerjee† Philippe Burlina† Fady Alajaji‡

†Center for Automation Research
University of Maryland at College Park
College Park, MD 20742-3275
‡Department of Mathematics and Engineering
Queen's University
Kingston, ON K7L 3N6, Canada

Abstract

We propose a segmentation method based on Polya's urn model for contagious phenomena. An initial estimate of the pixel labels is obtained using a Maximum Likelihood (ML) estimate or the Nearest Mean Classifier (NMC), which are used to determine the initial composition of an urn representing the pixel. The resulting urns are then subjected to a modified urn sampling scheme mimicking the development of an infection to yield a segmentation of the image into homogeneous regions. This process is implemented using contagion urn processes and generalizes Polya's scheme by allowing spatial interactions. The composition of the urns is iteratively updated by assuming a spatial Markovian relationship between neighboring pixel labels. The asymptotic behavior of this process is examined and comparisons with simulated annealing and relaxation labeling are presented. Examples of the application of this scheme to the segmentation of synthetic texture images, Ultra-Wideband Synthetic Aperture Radar (UWB SAR) images and Magnetic Resonance Images (MRI) are provided.

The support of the Army Federated Laboratories/Lockheed Sanders under Contract DAAL01-96-2-0001 is gratefully acknowledged.

1 Introduction

Image segmentation is a fundamental problem in computer vision which has been extensively studied. With the advent of new image modalities such as Synthetic Aperture Radar (SAR) and Magnetic Resonance Imaging (MRI), research into methods of segmentation has attracted renewed interest.

We describe a segmentation method using contagion urn schemes that rely on modified versions of the Polya-Eggenberger sampling process [12]. This biologically inspired sampling procedure was originally designed to model the development of contagious phenomena.

For our segmentation purposes, we model a scene as being composed of distinct, contiguous regions, each of which is described by constant or homogeneous attributes such as intensity or texture. An image of the scene is an imperfect description of the scene. The imperfection may be due to noise, which may be additive or multiplicative, or to blurring. In other words, an image is a corrupted version of an underlying piecewise smooth scene [10]. A natural approach to delineating the regions in an image is to statistically estimate the attributes of the regions and use the descriptions to differentiate between the regions.

Many such approaches to segmentation have been studied. Unsupervised segmentation approaches include Nearest Mean Classification (NMC) and the branch-and-bound procedure [3]. Supervised methods generally proceed by formulating statistical model assumptions for the region generation and the image formation processes. Maximum likelihood (ML) or maximum *a posteriori* (MAP) estimation is then used for segmentation. Examples of such approaches abound in the literature [4, 8, 15]. Techniques that model images as Markov random fields (MRFs) have been extensively investigated [4]. MRFs attempt to represent spatial dependencies, and the MRF-Gibbs equivalence allows for the computation of the maximum *a posteriori* (MAP) estimate of the original image [4].

For the corrupted, piecewise smooth image model, segmentation techniques which incorporate contextual information will usually be able to label the pixels more accurately than those which classify pixels independently . MRFs provide a direct mechanism for relating neighboring pixels. Algorithms such as relaxation labeling (RL) also make use of contextual information in the form of constraints on or supports for pixel labels. The presence of a label in a pixel's neighborhood imposes constraints on the possible labels of the pixel. RL is a parallel process in which these constraints reduce, and hopefully eliminate, ambiguity in the label of a pixel.

A "discrete" RL algorithm attempts to find a possibly unambiguous label for each pixel. For real images, a more appropriate strategy is to use the classical "probabilistic" RL algorithm introduced by Rosenfeld *et al.* [16]. In this method, probabilities are assigned to pixel labels. These probabilities are updated according to positive and negative supports provided by pixels in local regions. The nonlinear update mechanism provided in [16] is based on the above heuristic. The algorithm allows local information to propagate via iterative processing [9].

This paper models images using contagion urn processes. The idea behind this method is similar to that of RL; it iteratively propagates local information by contagion. The motivation for employing urn schemes is twofold: First, urn processes can generate Markov chains as well as MRFs [7]. Second, urn schemes are of particular interest because they provide a natural probabilistic representation for the image labels. Therefore, they constitute an attractive generative process for the underlying image regions which exhibit strong spatial dependencies. This approach is related to the Gibbs sampling procedure [4], preserving key features of the Gibbs sampler but using instead a contagion sampling scheme. The spatial dependencies of the pixel labels are captured by the contagious behavior which promotes segmentation of the image into regions. The urn process is analogous to relaxation labeling algorithms, except that the urn process is not deterministic [14], but stochastic. The urn sampling scheme is also iterative and can be performed in parallel at each site or pixel of the image.

In our scheme, each pixel is represented by an urn with a mixture of balls of different colors, one color for each class label. We begin by applying either an ML or NMC segmentation technique to the image to provide initial measures of similarity of the pixels to each class. These measures are directly mapped into the initial composition of each urn. A neighborhood is also defined for each pixel. The balls of the urns in the neighborhood are combined to determine the next state of the urn. The iterative nature of the algorithm incorporates temporal memory, while the use of the neighboring urns in the updating process promotes spatial contagion.

This paper is organized as follows: The initial NMC and ML segmentations are presented in

Section 2. The contagion-based smoothing process is then described in Section 3. In Section 4, the stochastic properties of the resulting image process are discussed. In Section 5 the relationships between the urn sampling scheme, relaxation labeling, and simulated annealing are examined. Finally, experimental results on texture, SAR and MR images are shown in Section 6.

2 Initial Segmentation

When no a priori information on the image statistics is available, general clustering algorithms such as NMC are usually applied. In the NMC method, an initial arbitrary labeling is used from which centroids of the feature vectors of each class are computed. Next, all samples are reclassified to the cluster corresponding to the nearest mean, and the centroids are recomputed. This process is iterated until a stopping criterion is met [3].

On the other hand, when a stochastic model for the image can be justified, it is possible to apply ML segmentation. The conditional distribution of the image, i.e.,

$$p(X_s/C_s = l, C_r; r \in N_s^k) \tag{1}$$

is assumed. Here, X_s is the given image data, C_s is the label for pixel s, and C_r represents the pixel labels of N_s^k, the k^{th} order neighborhood of pixel s [4].

For our ML segmentation purposes, we estimate the pixel labels by assuming that the conditional distribution of the image given each class label, i.e. $p(X_s/C_s = l)$, is a correlated, multivariate Gaussian distribution. The parameters for each class are estimated in the following manner: Training data for each class l is extracted. Non-overlapping $M \times M$ blocks of pixels from the training data are lexicographically ordered into $M^2 \times 1$ vectors $\mathbf{v}_i$. The empirical mean of these vectors serves as the unbiased estimate of the mean vector for class l,

$$\mathbf{m}_l = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i, \;\; l = 1, 2, ..., L. \tag{2}$$

Next, the unbiased estimate of the $M^2 \times M^2$ covariance matrix $\mathbf{\Sigma}_l$ of class l is obtained by

$$\mathbf{\Sigma}_l = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{v}_i - \mathbf{m}_l)(\mathbf{v}_i - \mathbf{m}_l)^T. \tag{3}$$

After obtaining the parameters of the different classes, the ML test determines the label for each pixel in the image. The ML decision rule is

$$\hat{l} = \arg\max_{l} \; p(X_s/C_s = l, C_r; r \in N_s^2). \tag{4}$$

where

$$p(X_s/C_s = l, C_r; r \in N_s^2) = \frac{1}{(2\pi|\mathbf{\Sigma}_l^{-1}|)^{M^2/2}} \, exp\left(\frac{1}{2}(\mathbf{x} - \mathbf{m}_l)^T \mathbf{\Sigma}_l^{-1} (\mathbf{x} - \mathbf{m}_l)\right). \tag{5}$$

The above schemes do not capture the homogeneity of regions. Since the ML test assumes that each pixel label is equally likely throughout the image, it produces a noisy segmentation. This assumption is incorrect, for in a local region dominated by one class, the dominant class has a higher prior probability than the other classes. Such contextual information is not taken into account in either the ML or NMC estimate of the pixel labels.

This drawback is addressed by using MAP segmentation. The MAP estimate of the class label $\hat{l}$ for a pixel given the observed image X_s is

$$\hat{l} = \arg\max_{l} \; p(C_s = l/C_r; r \in N_s^2, X_s) \tag{6}$$

It can be shown that maximizing $p(C_s = l/C_r; r \in N_s^2, X_s)$ is equivalent to maximizing $p(C_s = l/C_r; r \in N_s^2)p(X_s/C_s = l)$.

If segmentation of the image into homogeneous regions is desired, it is intuitively appealing to model the prior distribution $p(C_s = l/C_r; r \in N_s^2)$ using an MRF, as the MRF model relates the label of a pixel to the labels of its neighboring pixels [8]. If the prior is modeled as an MRF, the Gibbs-MRF equivalence can be exploited by techniques such as simulated annealing (SA) or other stochastic relaxation methods to derive the MAP estimate [8].

Unfortunately, techniques such as SA have high computational costs. Indeed, convergence to the MAP estimate is possible only when impractically slow annealing schedules are followed. Instead, we propose to replace the annealing step by an urn contagion process to model the spatial dependencies between neighboring pixels. Our motivation for employing an urn scheme lies in its ability to generate MRFs.

3 Urn Sampling with Contagion

In this section, the concept of temporal and spatial contagion for image segmentation is introduced, and the general urn sampling scheme for pixel classification is outlined.

3.1 Temporal Contagion

Polya [12] introduced the following urn scheme as a model for the spread of a contagious disease through a population. An urn originally contains T balls, of which W are white and B are black ($T = W + B$). Successive draws from the urn are made; after each draw, $1 + \Delta$ ($\Delta > 0$) balls of the same color as was just drawn are returned to the urn. Let $\rho = W/T$ and $\delta = \Delta/T$. Define the binary process $\{Z_n\}_{n=0}^{\infty}$ as follows:

$$Z_n = \begin{cases} 0, & \text{if the } n^{th} \text{ ball drawn is white;} \\ 1, & \text{if the } n^{th} \text{ ball drawn is black.} \end{cases}$$

It can be shown that the process $\{Z_n\}$ is stationary and non-ergodic [2, 12]. The urn scheme has infinite memory, in the sense that each previously drawn ball has an equal effect on the outcome of the current draw.

3.2 Temporal and Spatial Contagion

The urn sampling scheme proposed in this paper incorporates both temporal and *spatial* contagion. Instead of representing an image by a finite lattice of pixels, we consider an image as a finite lattice of urns. In the single-urn sampling described above, the effect of each sample propagates through time. For the lattice of urns, the sampled ball at each iteration must depend not only on the composition of the pixel's urn, but also on the compositions of the neighboring urns to encourage contagious behavior. Thus, we need to allow for spatial interactions at each time instant by involving the urns of the neighboring pixels in the determination of the newly sampled ball.

3.3 Urns as a Probabilistic Image Labeling Representation

The following presentation considers an L-ary labeling problem. Let $I_n = [p_n^{(i,j)}]$ be an L-ary label image of size $H \times K$, where $p_n^{(i,j)} \in \{1, ..., L\}$ is the label of pixel (i,j) at iteration n, $n = 0, 1, \ldots$, $(i,j) \in \mathcal{I}$ where

$$\mathcal{I} : \{(i,j) : i = 0, \ldots, H-1; j = 0, \ldots, K-1\}.$$

We associate an urn $u_n^{(i,j)} : (B_{0,n}^{(i,j)}, B_{1,n}^{(i,j)}, \ldots, B_{L,n}^{(i,j)})$ with each pixel $p^{(i,j)}$ at time n, where $B_{l,n}^{(i,j)}$ is the number of balls of color l in the urn. With this representation we define a similarity function for each pixel as

$$m_F^l(p_n^{(i,j)}) = \frac{B_{l,n}^{(i,j)}}{\sum_{k=1}^{L} B_{k,n}^{(i,j)}}.$$

This can be interpreted as the probability that pixel $p^{(i,j)}$ belongs to class l.

3.4 Contagion-based Segmentation

The general class of algorithms for the contagion-based segmentation process will now be described.

3.4.1 Initialization

Initialization of the urn composition is critical for the algorithm to converge to an appropriate segmentation. The initial segmentation, based on ML or NMC, provides this initialization.

When the correlated Gaussian assumption is used for the ML estimate, the urn initialization proceeds in the following manner. The similarity of pixel $p^{(i,j)}$ to class l is determined by the Mahalanobis distance

$$\text{Distance}_{(i,j)}(l) = (\mathbf{x}_i - \mathbf{m}_l)^T \mathbf{\Sigma}_l^{-1} (\mathbf{x}_i - \mathbf{m}_l), \tag{7}$$

where $\mathbf{x}_i$ is the vectorized M$\times$M neighborhood around pixel (i,j) and $\mathbf{m}_l$, $\mathbf{\Sigma}_l$ are the mean vector and covariance matrix, respectively, for each class. Classification with the correlated Gaussian assumption is also known as a minimum distance classifier. Next, the distances are converted to probabilities by

$$\text{P}_{(i,j)}(l) = \frac{\text{Distance}_{(i,j)}(l)^{-1}}{\sum_{l=1}^{L} \text{Distance}_{(i,j)}(l)^{-1}}. \tag{8}$$

Finally, the probabilities are mapped directly to the urn composition of pixel (i,j) by

$$\mathrm{B}_{l,0}^{(i,j)} = T * \mathrm{P}_{(i,j)}(l), \tag{9}$$

where $\mathrm{B}_{l,0}^{(i,j)}$ is the number of balls of color l in pixel (i,j)'s urn at time 0 and T is the total number of balls initially in the urn.

Likewise, when applying NMC, the relative distances of the feature vectors of a pixel to the centroids in feature space are used to determine the similarity of the pixel to each class, i.e.

$$\mathrm{P}_{(i,j)}(l) = \frac{\mathrm{Distance}_{(i,j)}(l)^{-1}}{\sum_{k=1}^{L} \mathrm{Distance}_{(i,j)}(k)^{-1}}. \tag{10}$$

and

$$\mathrm{B}_{l,0}^{(i,j)} = T * \mathrm{P}_{(i,j)}(l). \tag{11}$$

3.4.2 Iterative Urn Sampling

Once the urns are initialized, the general modified Polya-Eggenberger urn sampling scheme proceeds as follows.

For $n > 0$, the urn composition of each pixel (i,j) at time n is updated by sampling from a combination of the participating urns $\mathcal{V}_{n-1}^{(i,j)}$ with $\mathcal{V}_{n-1}^{(i,j)} : \{u_{n-1}^{(r,s)} : (r,s) \in N_q^k\}$, where N_q^k is the neighborhood system defined as in [4]:

$$N_q^k : \{q = (r,s) \in \mathcal{I} : (i-r)^2 + (j-s)^2 \leq k\}.$$

A simple, yet effective, sampling procedure is as follows: The urn $u_n^{(i,j)}$ for pixel $p^{(i,j)}$ is updated by first combining the balls of $u_{n-1}^{(i,j)}$ and the N neighboring urns:

$$\mathcal{C}_{n-1}^{(i,j)} = \mathrm{ASSOCIATE}\,(\mathcal{V}_{n-1}^{(i,j)}). \tag{12}$$

The ASSOCIATE function forms a collection of balls, $\mathcal{C}_{n-1}^{(i,j)}$, from the urns of the neighborhood. Examples of the ASSOCIATE function include grouping the urns of $\mathcal{V}_{n-1}^{(i,j)}$ into a "super" urn or sampling one ball from each urn to form the collection.

Next, a selection operation on the new collection of balls, $\mathcal{C}_{n-1}^{(i,j)}$, is performed, i.e.

$$Z_n^{(i,j)} = \mathrm{SELECT}\,(\mathcal{C}_{n-1}^{(i,j)}). \tag{13}$$

The SELECT function may determine the next state of the urns by sampling one ball from $\mathcal{C}_{n-1}^{(i,j)}$ or by taking the majority class of $\mathcal{C}_{n-1}^{(i,j)}$.

We denote by $Z_n^{(i,j)}$ the outcome of the SELECT function:

$$Z_n^{(i,j)} = l \text{ if the } n^{th} \text{ selected ball is color } l.$$

If $Z_n^{(i,j)} = l$, add Δ balls of color l to urn $u_n^{(i,j)}$.

This yields a new urn composition for each pixel, given by

$$u_n^{(i,j)} = \begin{cases} B_{l,n}^{(i,j)} = B_{l,n-1}^{(i,j)} + \Delta, & \text{if } Z_n^{(i,j)} = l; \\ B_{l,n}^{(i,j)} = B_{l,n-1}^{(i,j)}, & \text{else.} \end{cases}$$

The above procedure is iterated until $n = N$. At time N, the final composition of each individual urn $u_N^{(i,j)}$, $(i,j) \in \mathcal{I}$ determines the final labeling of the image. As described above, each urn represents a probabilistic membership function on the pixel labels.

For this paper, we have developed two specific methods based on the general urn process. In method 1, the contents of the N urns in the neighborhood of pixel $p^{(i,j)}$ are collected into a "super" urn. One ball is sampled from the "super" urn, and Δ balls of that color are added to the urn of pixel $p^{(i,j)}$, $u_n^{(i,j)}$. In method 2, one ball is sampled from each of the N urns in the neighborhood to form the collection $\mathcal{C}_n^{(i,j)}$. Then Δ balls of the majority color in $\mathcal{C}_n^{(i,j)}$ are added to $u_n^{(i,j)}$.

4 Statistical Properties

The idea behind our urn sampling scheme is to promote spatial contagion of the pixel labels. At the end of the iterative process, homogeneous regions should be described by one label. It is in this sense that the urn process generates MRFs; the label of a pixel is determined by the urns in its neighborhood. In this section, we report asymptotic results to provide insight as to why the urn sampling scheme allows the initial majority color of a region to dominate the population of the urns in that region.

4.1 Temporal Contagion

Consider the original, binary Polya sampling scheme. The asymptotic properties of the joint distribution can be characterized in the temporal case, i.e., when all spatial interactions are inhibited at each sampling step. In this case, it can be shown [11, 12, 13] that the proportion of white balls in each urn after the n^{th} trial $\rho_n^{(i,j)}$, where

$$\rho_n^{(i,j)} = \frac{\rho + \left(Z_1^{(i,j)} + Z_2^{(i,j)} + \cdots + Z_n^{(i,j)}\right)\delta}{1 + n\delta},$$

is a martingale [2] and admits a limit Y as the number of draws increases indefinitely. Indeed, $\rho_n^{(i,j)}$ (or equivalently the sample average $\frac{1}{n}\sum_{k=1}^{n} Z_k^{(i,j)}$) converges with probability 1 to Y [2]. This limiting proportion Y is a continuous random variable with support the interval $(0, 1)$ and beta probability density function with parameters $(\rho/\delta, (1-\rho)/\delta)$:

$$f_Y(y) = \begin{cases} \frac{\Gamma(1/\delta)}{\Gamma(\rho/\delta)\Gamma((1-\rho)/\delta)} y^{\frac{\rho}{\delta}-1}(1-y)^{\frac{1-\rho}{\delta}-1}, \\ if\ 0 < y < 1; \\ \\ 0, \text{otherwise}. \end{cases}$$

$\Gamma(\cdot)$ is the gamma function defined as

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \quad \text{for} \quad x > 0.$$

The behavior of this pdf can be interpreted as follows: Assuming $\delta = 1$ for simplicity, if the original fraction of white balls in the urn is close to 1, then the limiting distribution of $W_n^{(i,j)}$ will be skewed towards 1. A similar behavior is obtained for the case when ρ is close to 0. Therefore, the limiting pattern will reflect the underlying probability

$$Pr\left(p_1^{(i,j)} = x\right) = \rho^x(1-\rho)^{(1-x)}.$$

For the L-ary labeling case, the above observations generalize with convergence to the Dirichlet distribution [7].

4.2 Temporal and Spatial Contagion

We now examine the asymptotic behavior of the two specific sampling schemes given above.

Consider sampling from the "super" urn. Restating the problem, suppose there are N urns in the neighborhood of pixel X_s, each *initially* with b_i black balls and w_i white balls, and $b_i + w_i = T$ for all i, $i = 1, 2, \ldots, N$. We put the contents of all N urns into a "super" urn, sample one ball, and add Δ balls of the same color into the urn of pixel X_s. The following properties are easily derived:

The probability of sampling exactly k black balls from n iterations of the "super" urn is

$$Pr(X = k) = \begin{pmatrix} n \\ k \end{pmatrix} \frac{B(\alpha + k, \beta + n - k)}{B(\alpha, \beta)}, \tag{14}$$

where $\alpha = \sum_i \frac{b_i}{\Delta}$, $\beta = \sum_i \frac{w_i}{\Delta}$, and the beta function $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ (see Appendix for proof).

The above process can be regarded as being generated by a sequence of independent Bernoulli trials with parameter Z, where Z is random with beta distribution. In fact, it is identical with different parameters to the Polya-Eggenberger distribution in the single-urn case given above.

The average number of black balls in the "super" urn at any given time is

$$E\left[B_n\right] = \sum_j b_j \frac{NT + n\Delta}{NT}. \tag{15}$$

Therefore, the average proportion of black balls in the "super" urn is

$$E\left[\frac{B_n}{(NT + n\Delta)}\right] = \sum_j \frac{b_j}{NT}. \tag{16}$$

Remarkably, the average proportion of black balls in the "super" urn at any time equals the original proportion of black balls. This shows that the composition of the urn is highly dependent on the original proportion of the balls. Eventually, the majority class of the urns in a given neighborhood will spread and dominate the population of balls in that neighborhood. Therefore, we conclude that this urn sampling scheme will reinforce the majority class in a spatial neighborhood; it constitutes a positive-feedback system that yields limiting patterns of the self-reinforcing type [1]. The contagion effectively models the Markovian dependencies of the pixel labels.

This second method is described as follows: We sample one ball from each of the urns in pixel $X_{(i,j)}$'s neighborhood, $\mathcal{V}_{n-1}^{(i,j)}$. From this collection of balls, we compute the majority class, denoted by $Z_n^{(i,j)}$. We update urn $u_n^{(i,j)}$ in the same manner described in the previous section, i.e.

$$u_n^{(i,j)} = \begin{cases} B_{l,n}^{(i,j)} = B_{l,n-1}^{(i,j)} + \Delta, & \text{if } Z_n^{(i,j)} = l; \\ B_{l,n}^{(i,j)} = B_{l,n-1}^{(i,j)}, & \text{else.} \end{cases}$$

Eventually, the initial majority class of each urn in the neighborhood will dominate its composition, thereby propagating the label throughout the neighborhood.

It is difficult to find a general closed-form expression for $P(Z_n^{(i,j)} = k)$, the probability that class k is the majority of the individual samples. The difficulty arises because we are trying to find the majority of a set of samples of a non-i.i.d. process. Hence, we resort to heuristic arguments. Results obtained using both methods will be given in the next section.

The sequence of images generated by both methods exhibits both spatial and temporal dependencies represented by a Markovian relationship in terms of the urns $u_n^{(r,s)}$; more specifically:

$$Pr\{u_n^{(i,j)} | U_{n-1}, U_{n-2}, \ldots, U_0\} = Pr\{u_n^{(i,j)} | \mathcal{V}_{n-1}^{(i,j)}\},$$

where $U_n : [u_n^{(i,j)}]$ is the urn matrix associated with I_n, and $\mathcal{V}_{n-1}^{(i,j)}$ is the set of participating urns defined in the previous section.

5 Comparison of Methods

While there are many techniques for image segmentation, here we briefly examine the relationships between the urn sampling scheme and other methods with closely similar flavors, namely relaxation labeling and simulated annealing.

The idea behind the urn sampling scheme is that within a region, one label should be dominant. By repeatedly sampling with replacement from an urn, or a group of urns, the most frequently occurring color or label will asymptotically dominate the populations in the urns. Hence, contagion will promote the homogeneity of local regions.

Another interpretation of the urn process can be derived from the fact that when the urns are initialized by the ML estimate, the urn compositions represent the conditional probabilities of the image pixels. The subsequent iterative process updates these probabilities by adding balls of certain colors. The Polya-Eggenberger sampling process is such that it will naturally emphasize the majority labels.

Relaxation labeling for pixel classification is a procedure which assigns a best label to a pixel under certain constraints. It is also an iterative process which updates "probabilities" according to

$$P_n^{(i,j)}(l) = \frac{P_{n-1}^{(i,j)}(l)\,(1+q_n^{(i,j)}(l))}{\sum_{k=1}^{L} P_{n-1}^{(i,j)}(k)\,(1+q_n^{(i,j)}(k))} \tag{17}$$

$$q_n^{(i,j)}(l) = \sum_{k=1}^{L} R_{(i,j)}(l,k) P_n^{(i,j)}(k), \tag{18}$$

where $P_n^{(i,j)}(l)$ is the probability that pixel (i,j) belongs to class l at time n, $q_n^{(i,j)}(l)$ is a constraint coefficient, and $R_{(i,j)}(l,k)$ is the compatibility coefficient for labels l and k. As noted by Kittler [9], the update mechanism is based on heuristic arguments so that the update and resulting segmentation is influenced by the pre-defined constraints between class labels [6]. In the urn sampling scheme, improved segmentation is achieved without imposing such constraints. Indeed the population of each urn represents the support for the labels at each pixel. The changing of the urn compositions represents the changing of the supports; when one color dominates the population, the ambiguity of the label of a pixel is reduced.

The disadvantages of RL include the slow convergence of the algorithm. Also, studies have shown [5] that little improvement is gained after the first few iterations. Our results in the following section give evidence that the urn sampling scheme converges to the solution at a faster rate.

Simulated annealing may be employed when pixel classification is viewed as a function optimization problem, i.e. when using the MAP estimate of the label. Since, by Bayes' rule, maximizing the posterior probability of the pixel labels is equivalent to maximizing the product of the conditional distribution of the image and the prior probability of the labels, simulated annealing can also be viewed as a mechanism which updates the conditional probability of each pixel. The conditional probability serves as an initial estimate of the optimal point on a highly non-convex surface; an energy function for the prior distribution of the labels is adopted, and the simulated annealing procedure iteratively adds or subtracts from the conditional to arrive at the optimal point. The drawbacks of this method are its computational burden and its impractically slow schedule to converge to the solution.

The concept of simulated annealing is derived from thermodynamics; it models the way metals cool and anneal. When heated to extremely high temperatures, the molecules of the metal are at a high energy state and move relatively freely. When the metal is cooled, the molecules gradually lose their mobility; eventually they cool and converge to a lower energy state. If the temperature is decreased slowly, the molecules will reach their minimum energy state.

This analogy is incorporated in the Boltzmann probability distribution

$$Prob(E_1 \rightarrow E_2) \;\propto\; exp(-(E_2 - E_1)/kT). \tag{19}$$

E_1 and E_2 represent the current and an alternate energy level, respectively, of a function. Equation (18) states that the function will always go from E_1 to E_2 if E_2 is lower than E_1; it will go from E_1 to E_2 with probability $exp(-(E_2 - E_1)/kT)$ if E_2 is higher than E_1. In other words, if the function is presented with a lower energy state, it will change to that state. If it is presented with a higher energy state, it will change to that state with a small probability.

It has been proved that if the temperature is decreased according to the annealing schedule

$$T(k) = \frac{C}{log(1+k)}, \tag{20}$$

where C is a constant and $T(k)$ is the temperature at the kth iteration, the algorithm will converge to the MAP estimate [4]. The advantage of SA over greedy algorithms is that SA is able to avoid convergence to local minima by having the opportunity to jump to higher energy states.

Convergence to the true MAP segmentation of the image requires approximately 10^{10000} iterations [4]. Clearly, for practical applications, this is too long. The urn process does not attempt to achieve function optimization. The asymptotic results of Polya's sampling scheme ensure convergence by eventually forcing one color to dominate the population.

6 Experimental Results

In all of the examples given in this section, urn sampling method 1 is used; the contents of the urns of a pixel's neighborhood are put into a "super" urn, and a sample from the "super" urn is used to choose the label or color that is incremented in the pixel's urn. Each urn is initialized with 100 balls, and Δ, the number of balls added at each iteration, is 10. Urn sampling method 2 gives similar results.

In Figures 1 through 4, the ability of the urn process to segment an image into regions of different textures is demonstrated. The initial ML estimate is found by assuming that the textures can be described by a correlated Gaussian model with M=3. This model is unable to describe the grainy texture of the background, resulting in the inaccurate segmentation 1(b). The urn sampling process operates on the urns to produce a smoother segmentation. Figure 2 shows that the urn process provides a better segmentation than relaxation labeling. Note that the contagion-based segmentation usually preserves the edges of the texture regions better than the RL method. Figure 4 illustrates the iterative improvement resulting from the urn sampling scheme. The noisy background segmentation causes the RL algorithm to diverge to a nonsensical solution. However, the urn representation allows the urn sampling scheme to gradually adjust the urn compositions to achieve a smoother segmentation.

For foliage-penetrating SAR image analysis, it is helpful to segment the image into foliage and non-foliage regions to improve the performance of image exploitation systems. The images used in this study were obtained by the UWB SAR developed at the Army Research Laboratory. The image in Figure 5(a) shows the Aberdeen Proving Grounds in Aberdeen, MD during the summer of 1995. In segmenting this image, we start with ML segmentation with M=3. As shown in Figure 5(b), the resulting labeling is speckled, a characteristic of the ML segmentation technique.

Application of simulated annealing for a limited number of iterations generates only a slightly smoother segmentation of the image (Figure 6(a)). For SAR segmentation, SA is implemented in the following manner: Assigning the energy function $U(C_s = l)$ for the prior distribution is arbitrary, since the MRF-Gibbs equivalence states that $U(C_s = l)$ completely determines the form of $P(C_s/C_r)$. Therefore, we define an energy function [15]

$$U(C_s = l) = -(\beta/N) \sum_{i,j} \delta(C_{i,j} - C_k). \tag{21}$$

The energy is summed over the second order (3×3) neighborhood of pixel k. Minimization of the energy function $U(C_s = l)$ encourages smoothing of the pixel labels into homogeneous regions. The clustering parameter β is a measure of the smoothness incurred by the minimization of the function $U(C_s = l)$. Experimental results have shown that $\beta = 1.4$ provides acceptable segmentations [15]. The algorithm to find the global minimum of this energy function proceeds as follows: ML estimation of the pixel labels provides an initial segmentation. Next, the pixels are revisited randomly at each iteration. At each pixel, the energy is calculated under the current and alternate pixel labels. If the alternate pixel label yields a lower energy value, then the pixel is relabeled as the alternate class. If it yields a higher energy state, the label is changed with a small probability determined by the Boltzmann distribution. At the end of each iteration, the temperature is decreased according to the annealing schedule. Figure 6(b) shows that only ten iterations of the urn sampling scheme operating on the ML segmentation yield a much smoother image.

To segment the MR image in Figure 7 we obtain an initial segmentation by NMC. The proton density and T2 relaxation times are the components of the two-dimensional feature vector used for NMC. Since NMC is based solely on the means of the vectors, the initial segmentation is especially sensitive to the inherent noise of the MR image modality. The leads to the speckled segmentation shown in Figure 8(a). The distances from a pixel's feature vector to the centroids in feature space determine the initial composition of the urns. The urn process then operates on the urns to produce a smoother segmentation. The output after ten iterations is shown in Figure 8(b).

7 Conclusion

In this paper, we have illustrated how modified Polya urn sampling schemes can be implemented for image segmentation. Given an initial speckled segmentation, the contagion process obtains a smoother segmentation into homogeneous regions by its Markovian properties. Two general features result in temporal and spatial contagion. First, iterative updating provides for temporal contagion. Second, sampling from neighboring urns, similar to the Gibbs sampler, yields spatial contagion.

In the context of RL, the urn compositions provide positive support for the labels of each pixel. Polya's sampling with replacement ensures that one label will eventually have the highest degree of support. In terms of SA, the urn scheme also updates the conditional probabilities of each pixel in a stochastic manner. Our examples show that the urn scheme yields better segmentations in relatively small numbers of iterations.

ACKNOWLEDGEMENTS

We wish to thank Azriel Rosenfeld and Rama Chellappa for their guidance and many insightful discussions on relaxation labeling and image segmentation. We also gratefully acknowledge the help of Lockheed Sanders and the Army Research Laboratory in Adelphi, MD for making available the UWB SAR imagery used in this work.

A P-E Distribution of the "super" urn

We present the proof of Equation 14, that the "super" urn is governed by the Polya-Eggenberger distribution.

Consider again a binary sampling problem. Suppose there are N urns in the neighborhood of pixel (i,j), each initially with B_i black balls and W_i white balls, and that $B_i + W_i = T$, $i = 1, \ldots, N$. Now, at each time instant, place all of the balls into the "super" urn, and sample one ball from the "super" urn. If the ball is black, add Δ black balls to $u_n^{(i,j)}$; if it is white, add Δ white balls to $u_n^{(i,j)}$.

The probability of drawing a particular sequence of k black balls out of n samples (such as BBB...BWW...W) is

$$\frac{\sum_i B_i}{NT} \times \frac{\sum_i B_i + \Delta}{NT + \Delta} \cdots \frac{\sum_i B_i + (k-1)\Delta}{NT + (k-1)\Delta} \times \frac{\sum_i W_i}{NT + k\Delta} \cdots \frac{\sum_i W_i + (n-k-1)}{NT + (n-1)\Delta}. \tag{22}$$

Note that the order of a particular sequence does not affect the probability, i.e. the process is exchangeable. Therefore, the total probability of drawing k black balls out of n samples is

$$\mathrm{Prob}(X=k) = \begin{pmatrix} n \\ k \end{pmatrix} \frac{\sum_i B_i}{NT} \times \frac{\sum_i B_i + \Delta}{NT+\Delta} \cdots \frac{\sum_i B_i + (k-1)\Delta}{NT+(k-1)\Delta} \times \frac{\sum_i W_i}{NT+k\Delta} \cdots \frac{\sum_i W_i + (n-k-1)}{NT+(n-1)\Delta}. \tag{23}$$

Knowing that $NT = \sum_i B_i + \sum_i W_i$, and dividing through by Δ, the probability becomes

$$\begin{aligned} \mathrm{Prob}(X=k) &= \begin{pmatrix} n \\ k \end{pmatrix} \frac{\alpha(\alpha+1)(\alpha+2)\ldots(\alpha+(k-1)) \times \beta(\beta+1)(\beta+2)\ldots(\beta+n-k-1)}{(\alpha+\beta)(\alpha+\beta+1)\ldots(\alpha+\beta+(n-1))} \\ &= \begin{pmatrix} n \\ k \end{pmatrix} \frac{\Gamma(\alpha+\beta)\Gamma(\alpha+k)\Gamma(\beta+n-k)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta+n)} \\ &= \begin{pmatrix} n \\ k \end{pmatrix} \frac{B(\alpha+k,\, \beta+n-k)}{B(\alpha,\beta)}, \end{aligned} \tag{24}$$

where $\alpha = \sum_i \frac{B_i}{\Delta}$ and $\beta = \sum_i \frac{W_i}{\Delta}$.

References

[1] W. B. Arthur, Y. M. Ermoliev, and Y. M. Kaniovski, "Path-Dependent Processes and the Emergence of Macro-Structures," *European Journal of Operational Research*, pp. 294–303, 1987.

[2] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 2, Wiley, 1971.

[3] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, 1990.

[4] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 721–741, 1984.

[5] R. Haralick, "An Interpretation for Probabilistic Relaxation," *Computer Vision, Graphics and Image Processing*, Vol. 22, pp. 388–395, 1983.

[6] R. Hummel and S. Zucker, "On the Foundations of Relaxation Labeling Processes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 267–287, 1983.

[7] N. Johnson and S. Kotz, *Urn Models and Their Application*, Wiley, 1977.

[8] P. A. Kelly, H. Derin, and K. D. Hartt, "Adaptive Segmentation of Speckled Images Using a Hierarchical Random Field Model," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1628-1641, 1988.

[9] J. Kittler and J. Illingworth, "Relaxation Labelling Algorithms - A Review," *Image and Vision Computing*, Vol. 3, pp. 206–216, 1985.

[10] Y. LeClerc, "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision*, Vol. 3, pp. 73–102, 1989.

[11] G. Polya, "Sur Quelques Points de la Théorie des Probabilités," *Ann. Inst. H. Poincaré*, Vol. 1, pp. 117–161, 1931.

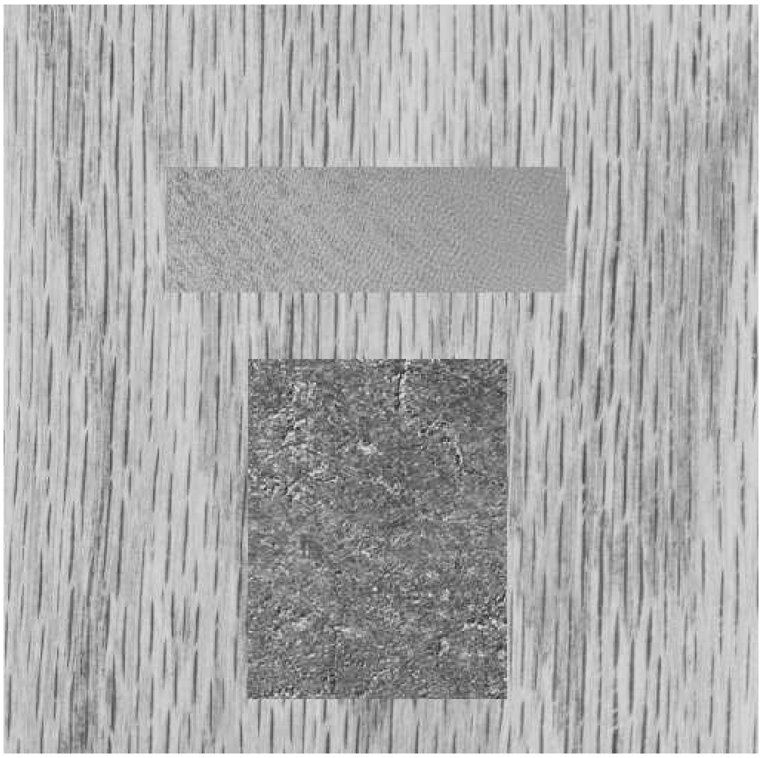
[12] G. Polya and F. Eggenberger, "Über die Statistik Verketteter Vorgänge," *Z. Angew. Math. Mech.*, pp. 279–289, 1923.

[13] G. Polya and F. Eggenberger, "Sur l'Interpretation de Certaines Courbes de Fréquences," *Comptes Rendus C. R.*, pp. 870–872, 1928.

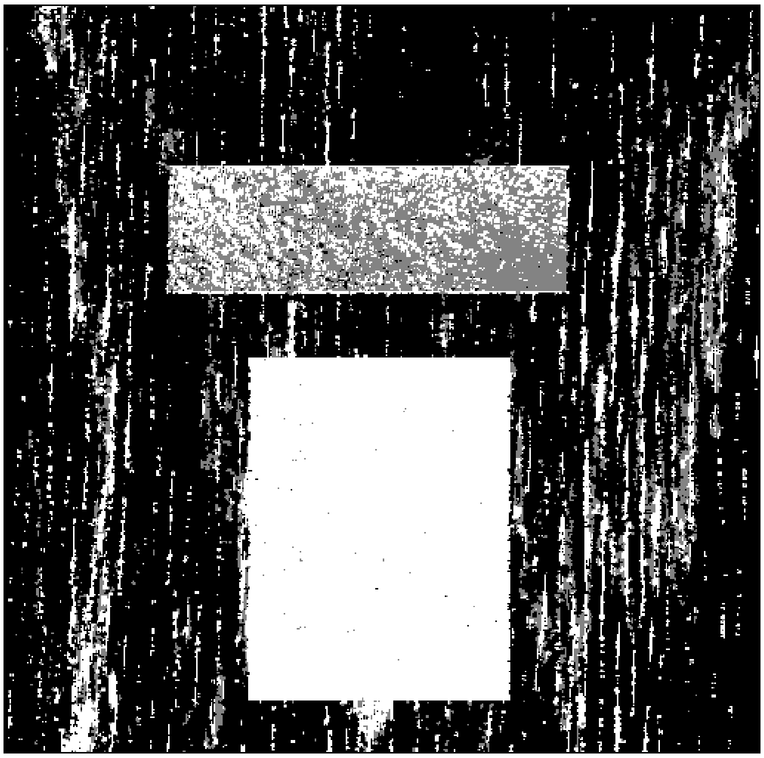
[14] S. Ranade and A. Rosenfeld, "Point Pattern Matching by Relaxation," *Pattern Recognition*, Vol. 12, pp. 269–275, 1980.

[15] E. Rignot and R. Chellappa, "Segmentation of Polarimetric Synthetic Aperture Radar Data," *IEEE Trans. on Image Processing*, Vol. 1, pp. 281–300, 1992.

[16] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 6, pp. 420–433, 1976.

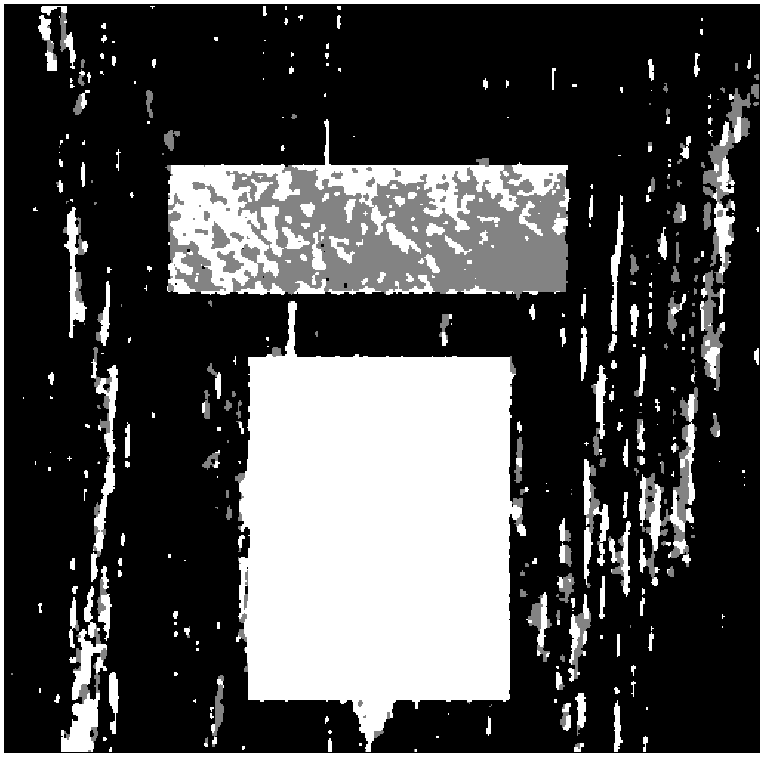


(a) Original texture image.

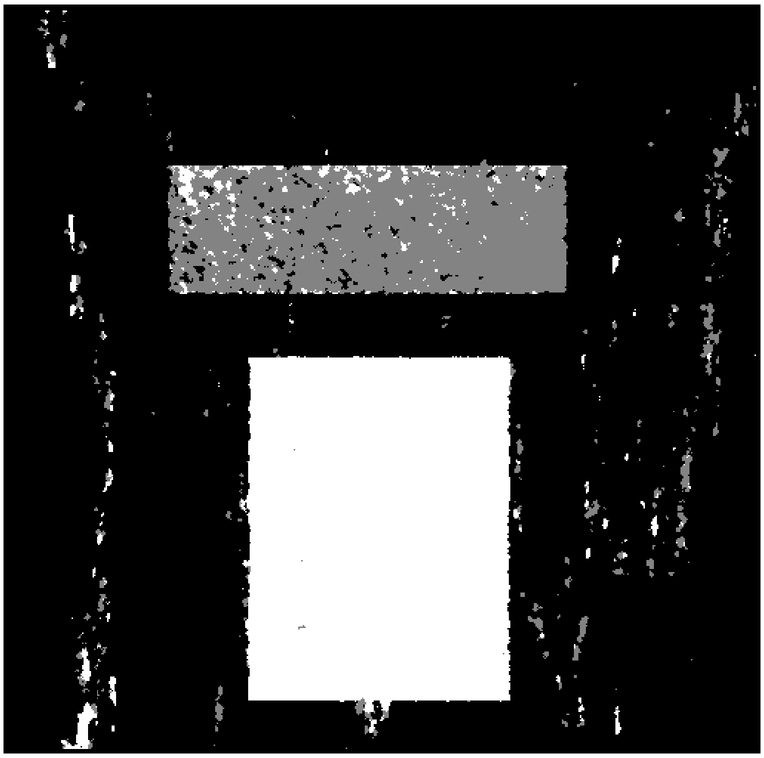


(b) ML segmentation.

Figure 1: ML segmentation of texture image.

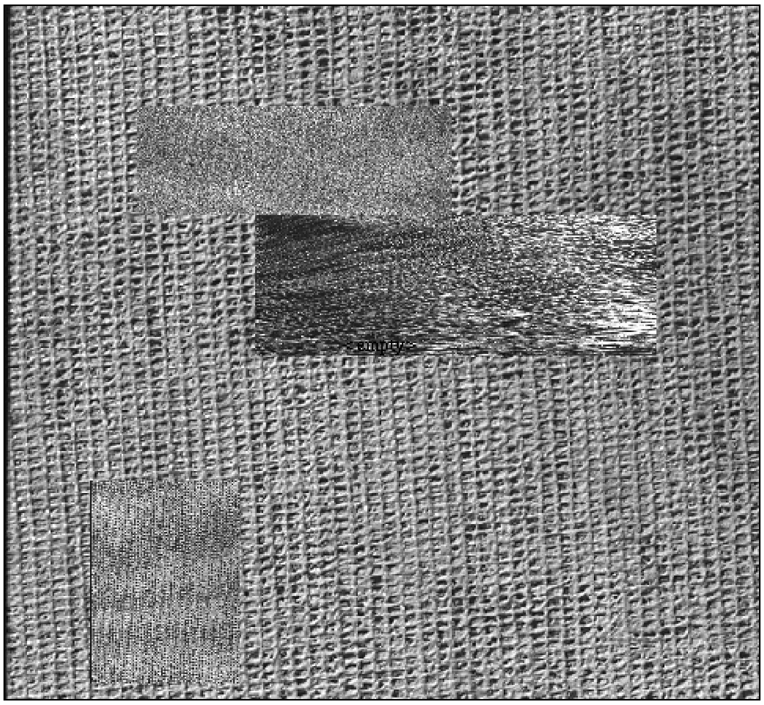


(a) Segmentation using RL after 15 iterations.

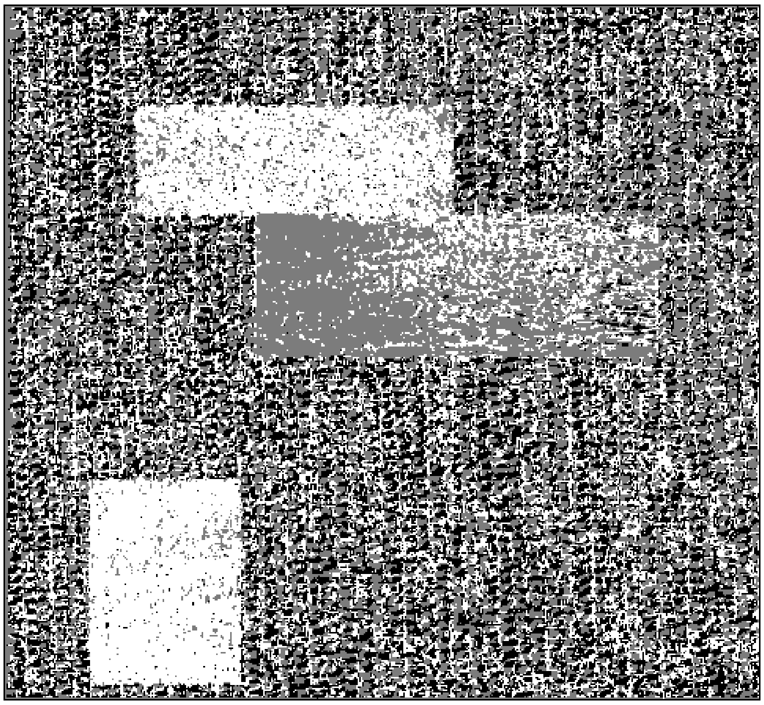


(b) Segmentation using 15 iterations of the urn process.

Figure 2: Comparison of texture segmentations using RL and urn process.

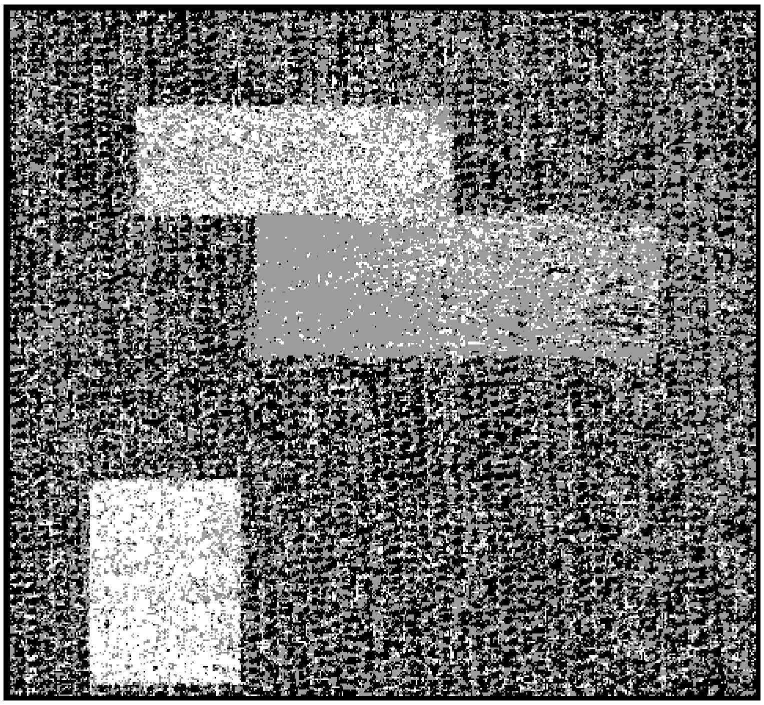


(a) Original texture image.

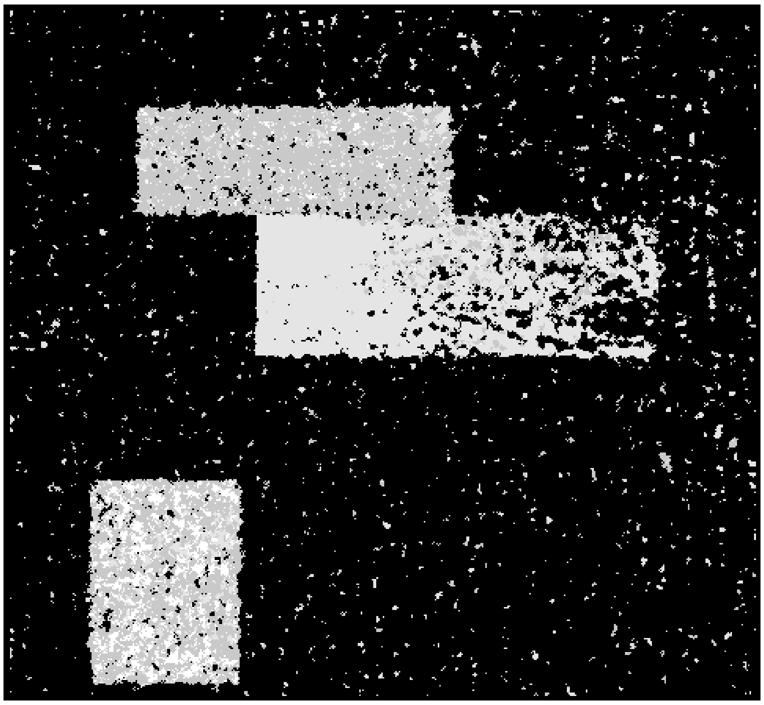


(b) ML segmentation.

Figure 3: ML segmentation of texture image.

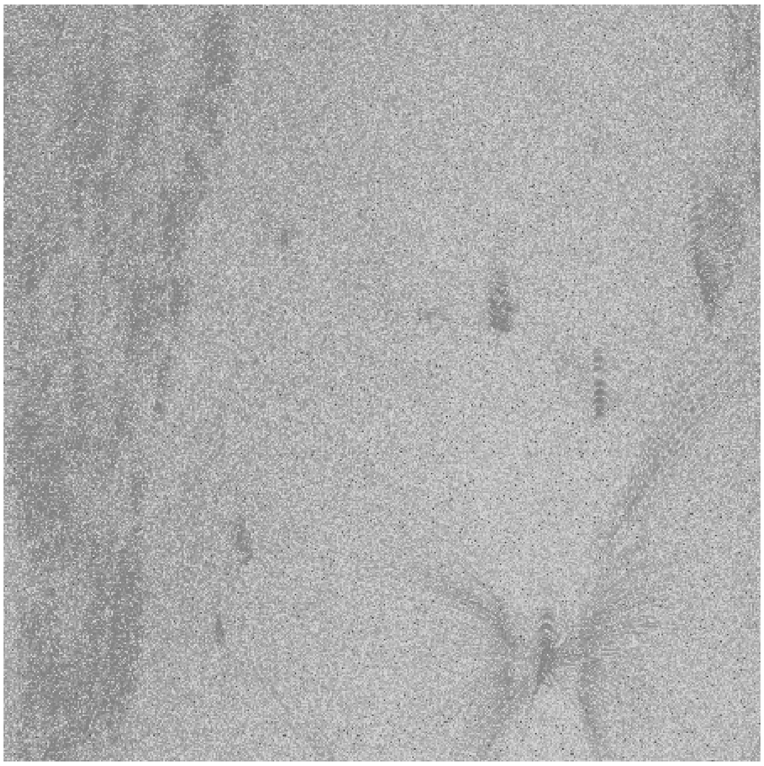


(a) Segmentation using one iteration of urn process.

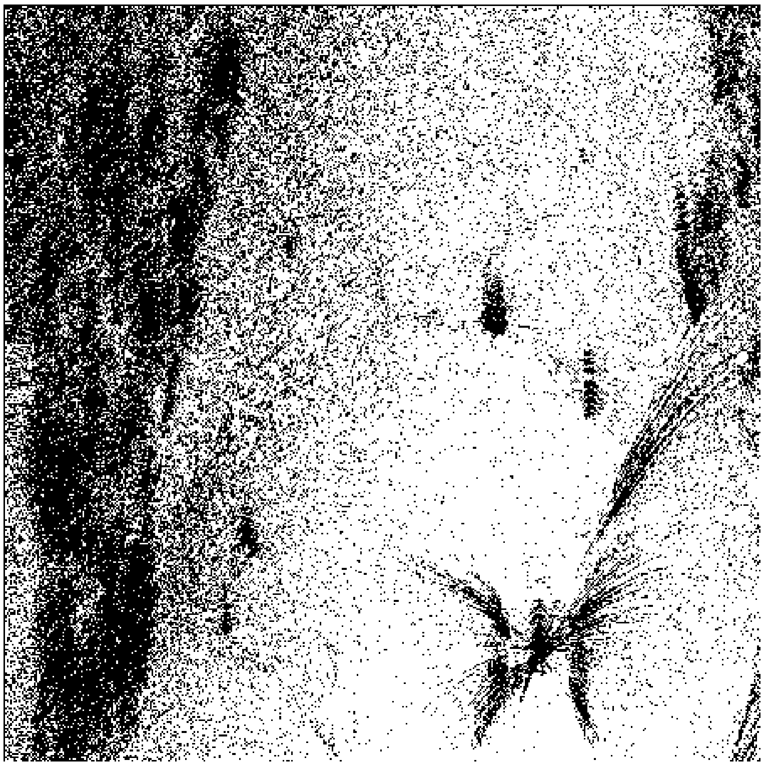


(b) Segmentation using 15 iterations of urn process.

Figure 4: Iterative improvement of texture segmentation

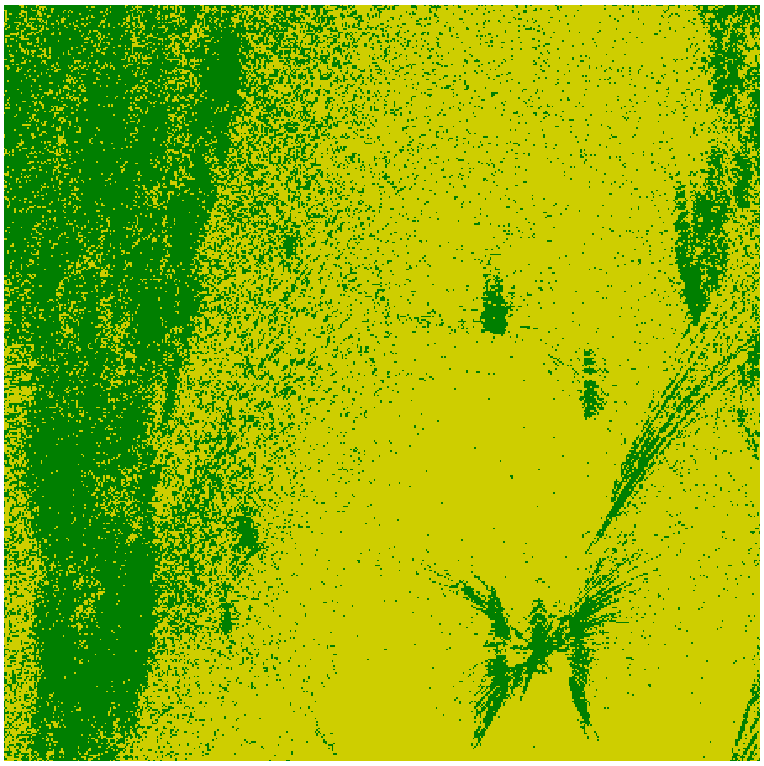


(a) Original UWB SAR image.

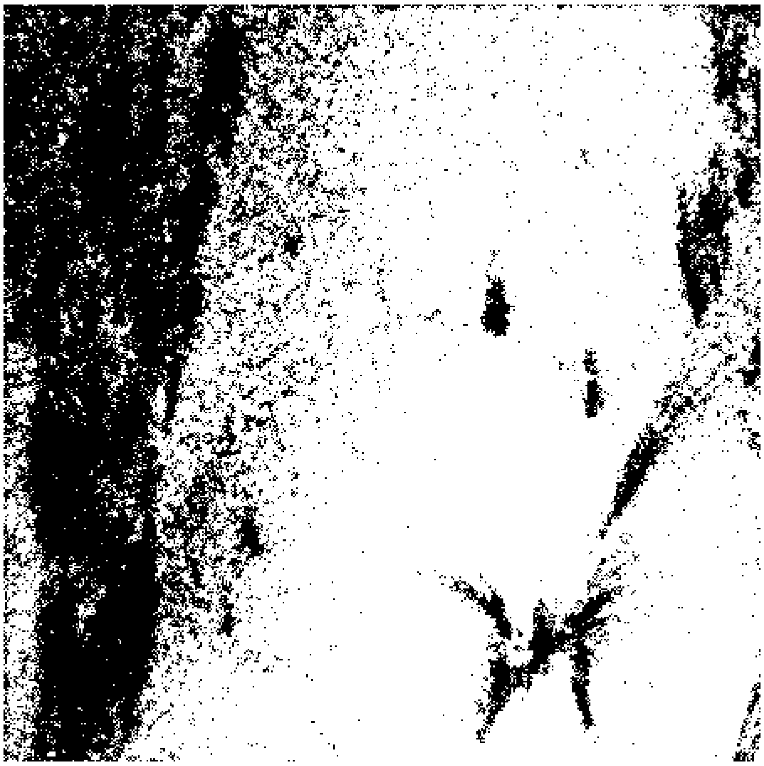


(b) ML segmentation.

Figure 5: ML segmentation of UWB SAR image.



(a) Segmentation using SA after 10 iterations.



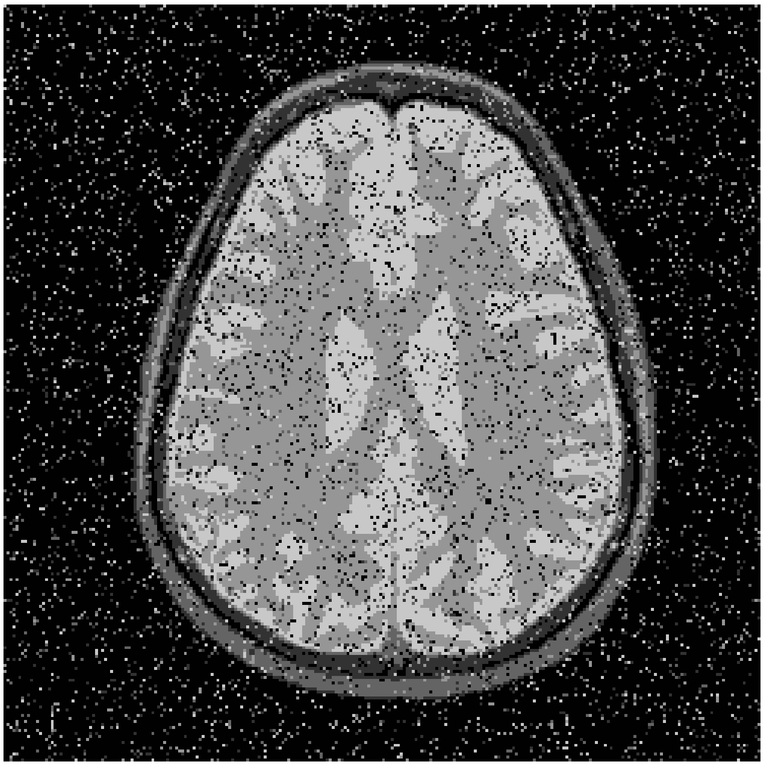
(b) Segmentation using 10 iterations of urn process.

Figure 6: UWB SAR segmentation using SA and urn process.

(a) T2 image.

(b) Proton density image.

Figure 7: Magnetic Resonance Images.



(a) Noisy segmentation.

(b) Segmentation using 10 iterations of urn process.

Figure 8: Segmentation of MR Images using the urn process