This paper originally appeared in the *Proceedings of the 1998 International Conference on Image Processing*, v. 3, pp. 770-774, IEEE, Chicago, (1998)

MRF-based Algorithms for Segmentation of SAR Images [*]

Robert A. Weisenseel, W. Clem Karl, David A. Castañon, and Richard C. Brower
Multi-Dimensional Signal Processing Laboratory, ECE Dept.
Boston University, 8 Saint Mary's St., Boston, MA 02215

Abstract

In this paper we demonstrate a new method for Bayesian image segmentation, with specific application to Synthetic Aperture Radar (SAR) imagery, and we compare its performance to conventional Bayesian segmentation methods. Segmentation can be an important feature extraction technique in recognition problems, especially when we can incorporate prior information to improve the segmentation. Markov Random Field (MRF) approaches are widely studied for segmentation, but they can be computationally expensive and, hence, are not widely used in practice. This computational burden is similar to that seen in the statistical mechanics simulation of simple MRF models such as certain magnetic models. Recently, Swendsen and Wang and others have had great success accelerating these simulations using so-called "cluster" Monte Carlo methods. We show that these cluster algorithms can provide speed improvements over conventional MRF methods when the MRF prior model has sufficient weight relative to the observation model.

1. Introduction

Robust and accurate segmentations of Synthetic Aperture Radar (SAR) imagery are of great use in the process of automatically recognizing ground targets. Correctly segmented target and shadow regions have proven to be powerful features in the classification process, as evidenced in DARPA's MSTAR program. However, to obtain good segmentations, it is necessary to incorporate prior information about the structure of the image. For example, SAR images have impulsive noise, also known as speckle, which can result in very noisy segmentations. Because we expect the segmentation field to be somewhat smooth, we can introduce a prior model that will reduce segmentation noise caused by impulsive observations. For example, we can introduce a segmentation field model that makes a pixel with a shadow segmentation label more likely to be surrounded by other shadow pixels than it would be with just an observation model.

To include prior information in a statistically rigorous fashion, many researchers study Markov Random Field (MRF) models for segmentation [1, 2, 3]. Through simple local interactions at the scale of a single pixel and its nearest neighbors, MRF models demonstrate complex global behavior able to capture a wide array of effects. This power of MRF techniques has led to their popularity with researchers. Unfortunately, most algorithms based on MRF models are computationally prohibitive, often because their local interactions are slow to numerically propagate global information. Ideally, we desire an approach that retains the power of local MRF models, but can share global information quickly. While researchers have examined multigrid techniques [3] for accelerating MRF algorithms with some success, a related technique for global information sharing has been largely ignored by the image processing community: the cluster Monte Carlo method.

First proposed by Swendsen and Wang [4, 5] for accelerating Monte Carlo simulations of MRF models from statistical mechanics, this acceleration method has demonstrated unequaled speedups in simulating systems described by MRF-type models. It is important to note that these speedups do not arise from reducing the number of computations per iteration but from reducing the number of iterations required to obtain a satisfactory segmentation. In addition to their acceleration capability, clusters may have an exploitable interpretation for prior models in their own right, just as was found for multiscale approaches when these were studied more deeply. In this paper, we develop and apply this cluster approach for a SAR image segmentation problem. We also show that the speed improvement of our cluster algorithm depends on the weight given to the prior model.

[*] Supported by Air Force Office of Scientific Research Grant F49620-96-1-0028, National Institutes of Health Grant NINDS 1 R01 NS34189, and Army Research Office Grant ARO DAAG55-97-1-0013

2. MMSE segmentation

First, our SAR image has been pre-processed to give a 128×128 pixel, strictly positive array of observations. The logarithm of the data is shown in Figure 1. Taking the logarithm is necessary to clarify the image structure.

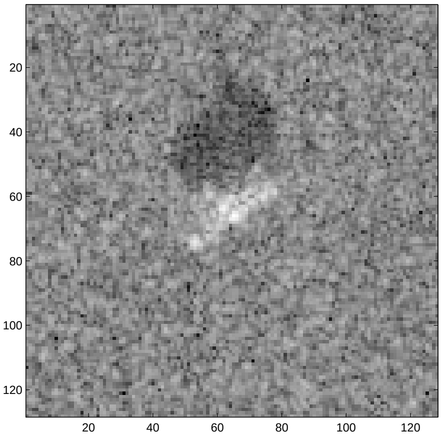


Figure 1: 128x128 vehicle SAR logarithm image

We assume that there are three classes underlying these observations: target, background, and shadow regions labeled 3, 2, and 1, respectively. Thus, letting y_i correspond to an observation pixel and x_i be the segmentation label associated with that pixel,

$$\begin{array}{cc} y_i \in \Re^+ & x_i \in \{1,2,3\} \\ y = [y_1, \ldots, y_{128^2}]^T & x = [x_1, \ldots, x_{128^2}]^T \end{array} \quad (1)$$

We develop our MRF approaches in a Bayesian framework. We use a Minimum Mean-Squared Error (MMSE) criterion for our segmentation. Solving the optimization problem associated with this criterion results in an estimate that would be the conditional expectation in a continuous formulation:

$$\hat{x} = E\{x|y\} \quad (2)$$

Computing this expectation would be prohibitively expensive because it involves a sum over a large number of terms (there are 3^{128^2} possible configurations) and there is no other closed form solution. Thus, we use Monte Carlo methods to approximate it:

$$E\{x|y\} \approx \frac{1}{N}\sum_{k=1}^{N} x^{(k)} \quad (3)$$

where $x^{(k)}$ is a sample from the posterior probability distribution, $p(x|y)$. Because we desire a discrete segmentation, we threshold the resulting average image.

In order to sample from the posterior distribution, we require a likelihood distribution for y given x and a prior distribution for x by Bayes rule,

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (4)$$

We do not require $p(y)$: this prior distribution is just a constant, because y is known. Thus, the mean of $p(x|y)$ is the same as the mean of $p(y|x)p(x)$.

3. Models

For $p(y|x)$, we use an independent, identically distributed model for the observation pixels, y, given the segmentation labels, x:

$$p(y|x) = \prod_{i=1}^{N} p(y_i|x_i) \quad (5)$$

where N is the total number of pixels, and y_i is conditionally Rayleigh distributed to capture the strict positivity of the data and the impulsive nature of the noise.

$$p(y_i|x_i = \ell) = \tfrac{y_i}{r_\ell^2} e^{-\frac{1}{2}(y_i^2/r_\ell^2)} \quad \ell \in \{1,2,3\} \quad (6)$$

The variable r_ℓ is the parameter of the Rayleigh distribution associated with segmentation label ℓ.

To describe the prior model on the segmentation label field, we use a MRF. The basic idea is that each pixel depends only on its nearest neighbors. For simplicity, we will examine a "smoothness" model. Using the MRF-Gibbs equivalence [2] we write our model directly as a Gibbs distribution,

$$p(x) = \frac{1}{Z} e^{-H(x)} \quad (7)$$

$$H(x) = -\beta \sum_{i=1}^{N} \sum_{j \in \eta_i} [\delta(x_i - x_j) - 1] \quad (8)$$

where Z is a normalization constant, β is a field strength parameter (as beta goes to zero, this approaches a uniform distribution), η_i is the set of four nearest neighbors of pixel i, and $\delta(.)$ is the discrete delta function. Clearly, it is more probable for a pixel to have the same label as the majority of its neighbors.

We can combine the prior model with the observation model to obtain a posterior distribution that is also in Gibbs form,

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{1}{Z} e^{-H(x|y)} \quad (9)$$

$$H(x|y) = -\beta \sum_{i=1}^{N} \left(\alpha \left[\log \frac{y_i}{r_{x_i}^2} - \frac{1}{2} \frac{y_i^2}{r_{x_i}^2} \right] + \sum_{j \in \eta_i} [\delta(x_i - x_j) - 1] \right) \qquad (10)$$

where α is just a weighting parameter governing the relative strength of the prior and observation distributions. Note that increasing α *decreases* the amount of smoothing.

4. Markov chains

4.1. Conventional algorithm

The conventional method to compute samples from the distribution described by Equations 7 and 8 is a Markov chain. For each pixel, hold the rest of the field fixed and change it with some probability. For example, in the Gibbs sampler approach,

$$\begin{aligned} p(x_i = \ell|\eta_i) &= \\ &= p(x_i = \ell | x \setminus x_i) = \frac{p(x_i = \ell, x \setminus x_i)}{\sum_{x_i=1}^{3} p(x)} \\ &= \frac{\exp\left(2\beta \sum_{j \in \eta_i} [\delta(\ell - x_j) - 1]\right)}{\sum_{x_i=1}^{3} \exp\left(2\beta \sum_{j \in \eta_i} [\delta(x_i - x_j) - 1]\right)} \qquad (11) \end{aligned}$$

Updating every pixel in the array with this method at each iteration results in a Markov Chain that converges in distribution to the probability function given by Equations 7 and 8. Alternative methods include the Metropolis algorithm [2]. We can obtain a Markov chain that converges to the distribution described by Equations 9 and 10 by following a similar set of steps.

The main problem with this approach is that information propagates through the field as a random walk due to the random local updates. This propagation can be quite slow, resulting in high correlation between one step of the Markov chain and the next. To compute averages such as the one described in Equation 3, we would like independent samples. Using the method described above, many steps may be required before two samples may be considered "independent."

4.2. Cluster algorithm

A cluster algorithm provides an alternative method for computing a Markov chain that converges to the desired distribution. The general idea of a cluster algorithm is to extend the reach of nearest neighbor interactions at each step while retaining the same prior and observation probabilities. To do this, we augment the probability space with a new set of random variables, which we call bonds.

A bond governs the interaction between two neighboring pixels by making it an all or nothing relationship. Instead of having some probabilistic relationship, these two neighboring pixels become locked together and are forced to act together for this step of the algorithm. Pixels linked by bonds form a *cluster* whose destinies are tied for the current step of the Markov chain. The stochastic nature of the interaction between pixels is absorbed in the randomness of the bond formation process. Only the observation field is used to compute the probabilities in a cluster update.

For the case of the posterior distribution described by Equations 9 and 10, the appropriate bond process is to never form a bond between pixels with differing segmentation labels. If two pixels have the same label, however, we form a bond between them with probability $1 - e^{-\beta}$. We then search out every grouping of pixels that are connected by bonds. Call these groupings clusters.

Once an image has been divided into clusters, the state of every pixel in a cluster is updated together using the observation model as a basis for selecting the new state of the cluster. This update can be accomplished in a manner similar to Equation 11,

$$p(C_j = \ell|y) = \frac{\exp\left(\alpha \sum_{i \in C_j} \left[\log \frac{y_i}{r_\ell^2} - \frac{1}{2}\frac{y_i^2}{r_\ell^2}\right]\right)}{\sum_{C_j=1}^{3} \exp\left(\alpha \sum_{i \in C_j} \left[\log \frac{y_i}{r_\ell^2} - \frac{1}{2}\frac{y_i^2}{r_\ell^2}\right]\right)} \qquad (12)$$

where $C_j = \ell$ denotes that the class label of every pixel in cluster j is ℓ:

$$x_i = \ell \quad \forall i \in C_j \qquad (13)$$

After updating every cluster, the old bonds are forgotten and the process is repeated.

Because two distant pixels can be part of the same cluster, state information can reach distant points in the image very quickly. The resulting cluster algorithm can have a much lower correlation between two samples of the Markov chain than the corresponding conventional, local update algorithm. The surprising about this process is that updating clusters results in the correct probability density function.

To summarize, at each iteration perform the following steps:

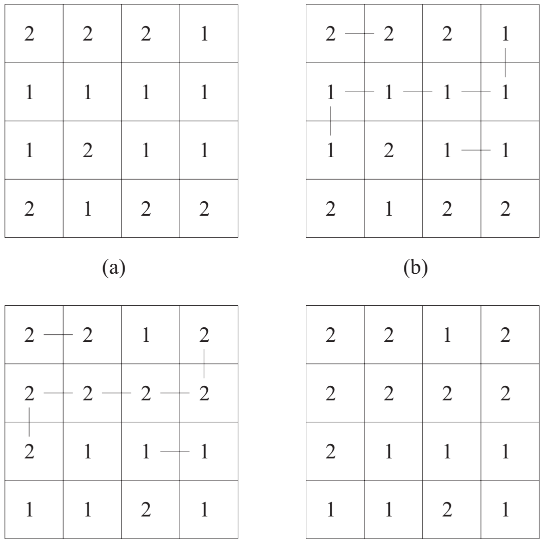
1. randomly form bonds between neighboring pixels conditioned on the segmentation labels with probability $1 - e^{-\beta}$ if the neighbors are equal, and with probability 0 otherwise

2. randomly update the clusters conditioned on the bonds and the data with probabilities as given in Equation 12

3. go to step 1

This Markov chain will converge in distribution to the Gibbs distribution described by Equations 9 and 10.

As an example of a cluster update, take the two-state case with no data, so clusters change to a new state with probability of one-half. This process is illustrated for a single iteration on a 4x4 array in Figure 2. Figure 2a is a typical segmentation field for a binary case that only takes on values 1 and 2. Figure 2b shows the same image with bonds shown; each connected region denotes a cluster. Figure 2c shows the clusters after the state update step. Figure 2d shows the new state, ready for a new set of bonds.



(a) (b)

(c) (d)

Figure 2: Example of a cluster update step

5. Results

Our primary result is that the speed advantages of cluster algorithms for segmentation can depend strongly on the emphasis given to the prior. To compare the decorrelation times of the Markov chains resulting from the conventional and cluster algorithms described in Section 4., we choose to use a summary measure. We take the energies, $H(x|y)$, and compute their correlation function as follows:

$$C(\tau) = \frac{\frac{1}{M}\sum_{t=1}^{M}\left[H(x^{(t)}|y) - \bar{H}\right]\left[H(x^{(t+\tau)}|y) - \bar{H}\right]}{C(0)} \tag{14}$$

where $\bar{H}$ is given by

$$\bar{H} = \frac{1}{M}\sum_{t=1}^{M} H(x^{(t)}|y) \tag{15}$$

This statistic provides a measure of how two samples of the Markov chain are correlated as a function of their separation in time, where time corresponds to iterations of the Markov chain.

For the conventional, local update algorithm, Figure 3 shows the correlation function versus time shift for three different levels values of α, the weighting parameter on the prior model. Note that the decorrelation time is very large for strong smoothing (very low α), but becomes negligible as the amount of smoothing decreases (α increases). This variation in decorrelation time as the weight on the prior model changes makes sense: for weak smoothing, the local information provided by the observation model is valued more strongly than the long-distance information that the prior model can describe. Propagating that long-distance information is what can slow the decorrelation of the Markov chain.

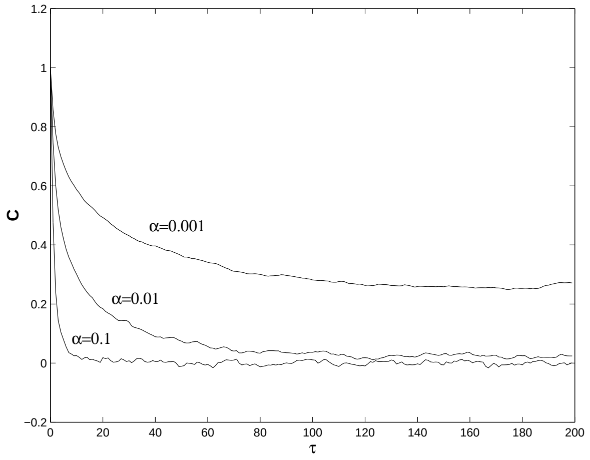


Figure 3: Correlation vs. time shift for conventional MRF

Figure 4 shows the graphs of the correlation function versus time difference for the cluster algorithm. These graphs are at the same values of α as for the conventional case. The correlation time of the cluster algorithm is clearly much lower at low values of α. As we increase α, decreasing the weight on the prior, the difference between the decorrelation times of the conventional and cluster Markov chains shrinks.

We show a typical segmentation using the conventional, local update algorithm for $\alpha = 0.5$ in Figure 5.

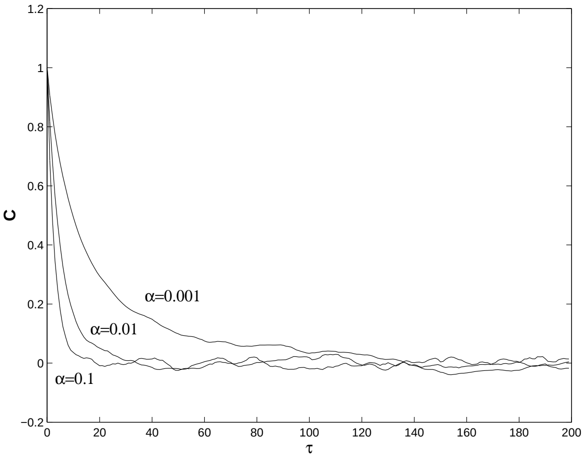


Figure 4: Correlation vs. time shift for cluster MRF

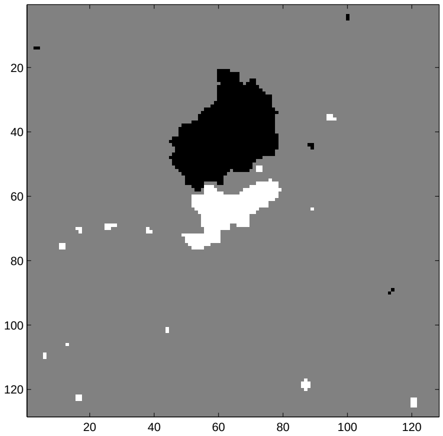


Figure 6: Cluster Segmentation, 300 iterations

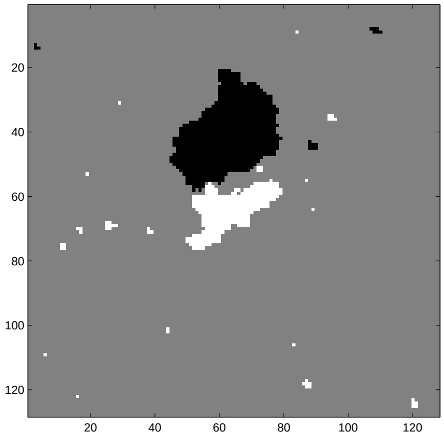


Figure 5: Conventional Segmentation, 300 iterations

In Figure 6, we show a corresponding segmentation output from the cluster algorithm. As we would expect based on our analysis of the correlation times, the cluster algorithm does not offer a significant improvement over conventional algorithms for this level of smoothing and the same number of iterations.

6. Conclusion

While cluster algorithms demonstrate good acceleration for a strongly weighted prior model, their speed advantages appear to be lower for a level of smoothing appropriate to image processing problems. These algorithms may also be useful as a tool for describing Markov Random Fields. By formulating the problem in another framework, they may open new possibilities for prior models, just as multigrid methods did when these techniques were studied more closely. In future work, we would like to extend these algorithms to more general prior models and demonstrate priors that would be difficult to formulate in any other algorithmic paradigm.

7. REFERENCES

[1] N.S. Friedland, "A 'PTBS' Segmentation Scheme for Synthetic Aperture Radar", Proceedings of the SPIE, V.2484, *Signal Processing, Sensor Fusion, and Target Recognition IV*, pp476-493, Jul, 1995

[2] S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, V 8, pp679-698, 1984

[3] M. Mignotte, C. Collet, P. Perez, P. Bouthemy, "Unsupervised Markovian Segmentation of Sonar Images" *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, V. 4, pp2781-2784

[4] Jian-Sheng Wang, R.H. Swendsen, "Cluster Monte Carlo algorithms", Physica A, V. 167, N.3, p. 565-579

[5] R.H. Swendsen, Jian-Sheng Wang, "Nonuniversal critical dynamics in Monte Carlo simulations", *Physical Review Letters*, V. 58, N. 2, pp86-88, 1987