MRF Model-Based Algorithms For Image Segmentation ‡

R. C. Dubes A. K. Jain S. G. Nadabar C. C. Chen

Department of Computer Science
Michigan State University
East Lansing, Michigan 48824-1027

ABSTRACT

This paper empirically compares three algorithms for segmenting simple, noisy images: Simulated Annealing (SA), Iterated Conditional Modes (ICM), and Maximizer of the Posterior Marginals (MPM). All use Markov random field models to include prior contextual information. The comparison is based on artificial binary images which are degraded by Gaussian noise. Robustness is tested with correlated noise and when the object and background are textured. The ICM algorithm is evaluated when the degradation and model parameters must be estimated, both in supervised and unsupervised modes and on two real images. The results are assessed by visual inspection and through a numerical criterion. We conclude that contextual information via Markov random field models improves segmentation when the number of categories and the degradation model are known and that parameters can be effectively estimated. None of the three algorithms is consistently best but the ICM algorithm is the most robust. We also demonstrate that a-posteriori energy is not always minimized at the "best" segmentation.

1. Introduction

We view image segmentation as the problem of recovering a "true" image consisting of a few homogeneous regions from a noisy image by labeling individual pixels according to region type. This paper compares a class of image segmentation algorithms that use Markov Random Fields (MRF) as models of context in the true image. Context is important in image segmentation because contiguous pixels are likely to belong to the same region. Markov random fields are appropriate prior contextual models because they can specify the local properties of image regions. The MRF model of prior information need not be an accurate model of the true image itself. An MRF model is seen here as a convenient means for introducing context, or dependence among neighboring pixels.

This pixel-labeling problem has been called image restoration [2, 8] as well as image segmentation [5, 6]. The type of labels used distinguishes among problems of restoration, segmentation, and edge detection. The set of pixel labels can be unordered, as when segmenting Landsat satellite images according to land use category, or ordered, as when recovering actual gray level intensities. In image restoration, the labels are from the set of gray values {0, 1, ..., 255}. In image segmentation, the labels are from {1, 2, ..., C}, where C is the number of categories. Edge detection can be viewed as a special case of pixel labeling with C=2 (edge and no edge). Haralick and Shapiro [9] give a survey of segmentation techniques. Relaxation labeling is similar to MRF model-based segmentation and iteratively changes pixel labels to optimize a probabilistic index. Relaxation labeling does not mathematically guarantee a unique optimal solution but seeks a practical suboptimal solution.

Comparing image segmentation algorithms is not an easy task. First comes the problem of specifying the segmentation algorithm. Several variations of MRF-based segmentation algorithms have been proposed in the literature [2, 4, 5, 6, 8, 11, 13, 12]. They differ in the details of the type of prior model, in the model for corrupting the true image by noise, in the type of optimization algorithm, and in the details of the algorithms themselves. Choices of models for the image and the noise are often dictated by the problem at hand. Reports of MRF model-based segmentation algorithms in the literature exhibit at least one of the following drawbacks making accurate reproduction of results impossible: (i) Mathematical formulas are not stated explicitly; (ii) Algorithms are not clearly specified; (iii) Parameters are not given; (iv) Details of test images, such as actual gray levels, are not provided. We try to clarify these details in our experiments and make a fair and unbiased assessment of relative capability and utility. The goal is a practical comparison of segmentation algorithms, not a study of their mathematical properties. We have limited our comparison to three methods which have appeared in the current literature: Simulated Annealing (SA), Iterated Conditional Modes (ICM), and Maximizer of the Posterior Marginals (MPM).

The second problem in comparing segmentation algorithms is the difficulty in choosing a criterion for assessing performance. The most common criterion is the percentage of pixels misclassified. The obvious difficulty with this criterion is that it ignores the capability of an algorithm to recover important details, such as straight lines. This criterion also depends on the number of true segments in the underlying image. Our comparison is based on several synthetic images which cannot be segmented easily by heuristic algorithms and on two real images. We rely on a visual comparison of the true and segmented image to judge the performance of the algorithms and report a simple numerical index of performance. The parameters of the models are assumed known in our initial experiments. We extend our study of the ICM algorithm to situations in which parameters are estimated, with and without training samples.

2. Background

This section briefly defines our notation. Complete definitions of the underlying concepts are given elsewhere [2, 8, 6, 7]. An (intensity) image specifies the gray levels for all pixels in an $M \times N$ lattice. The gray levels belong to the set $A = \{0,1,...,G\text{-}1\}$. The "true" or "perfect" image is represented by the vector random variable $\mathbf{X} = \{X_1, X_2, \ldots, X_{MN}\}$ which is modeled by a (discrete) Gibbs random field. The range set for each X_t is the set of labels $\{1,...,C\}$. The gray level at site t is denoted by μ_{x_t} which can be any value in A. Only two gray levels have non-zero probability in binary images. It is often assumed that the gray levels associated with all pixels having identical labels are the same.

‡ This work was supported by NSF Grants IRI-8901513 and ECS-8603541

Figure 1 demonstrates the labeling of pixels and notation used in this paper. For example, Figure 1(a) labels first-order spatial neighbors of site t as "1", second-order neighbors as "2", and so forth to fifth order. A linear ordering of sites from 1 to MN is assumed in vectors such as $\mathbf{X}$. The relative ordering in Figure 1(b) provides a convenient labeling for the neighbors of each pixel. Periodic boundaries are assumed so every pixel has the same number of spatial neighbors.

5	4	3	4	5
4	2	1	2	4
3	1	t	1	3
4	2	1	2	4
5	4	3	4	5

(a)

t:-11	t:-7	t:-6	t:+8	t:+12
t:-9	t:-3	t:-2	t:+4	t:+10
t:-5	t:-1	t	t:+1	t:+5
t:-10	t:-4	t:+2	t:+3	t:+9
t:-12	t:-8	t:+6	t:+7	t:+11

(b)

Figure 1. Definitions of Neighborhoods and Relative Neighborhoods

The image observed is denoted by the MN-vector random variable $\mathbf{Y}$ and is obtained by adding a noise process to the perfect image. We assume i.i.d. Gaussian noise. This is the usual degradation model studied in the literature [2, 6, 11]. Therefore, the density for $\mathbf{Y}$, given the perfect image, is

$$f(\mathbf{y} \mid \mathbf{X} = \mathbf{x}) = \prod_{t=1}^{MN} f_t(y_t \mid x_t).$$

Note that $f_t(. \mid x_t)$ is the conditional density function for Y_t, the gray level at pixel t, given the true label x_t at pixel t. We take $f_t(y_t \mid x_t)$ to be the Gaussian density function with mean μ_{x_t} and variance $\sigma_{x_t}^2$. The variance $\sigma_{x_t}^2$ depends on the additive noise and the gray level variation associated with each image label.

Contextual information enters the image segmentation problem through a Markov random field (MRF) model of the statistical dependence among the labels on neighboring pixels in $\mathbf{X}$ which is equivalent to a Gibbs process [1]. The (a-priori) probability function for $\mathbf{X}$ under a Gibbs random field with respect to a neighborhood system of cliques is given below [1, 6, 8]. Notation P refers to the probability mass function for discrete random variables, such as $\mathbf{X}$, and f refers to the density function for continuous random variables, such as $\mathbf{Y}$.

$$P(\mathbf{X} = \mathbf{x}) = e^{-U(\mathbf{x})}/Z,$$

where Z is the partition function or the sum of the numerator over all possible labelings and $U(.)$ is the energy function. This paper is limited to pairwise interaction processes in which spatial neighbors occur only in pairs.

$$U(\mathbf{x}) = \sum_{t=1}^{MN} \sum_{r=1}^{c} [\theta_r J(x_t, x_{t:+r})], \tag{1}$$

where $J(a,b) = -1$ if $a=b$, 0 if $a \neq b$; $c = 2$ for a first-order model and $c = 4$ for a second-order model; $\{\theta_1, \ldots, \theta_c\}$ are the clique parameters. We emphasize that the random field model is used only to insert context into the problem and is not expected to be an accurate model of the true image. Techniques for fitting models to images are discussed elsewhere [3].

Under our assumptions, the a-posteriori probability mass function for the pixel labels $\mathbf{X}$, given the observed image $\mathbf{Y} = \mathbf{y}$ also has the form of a Gibbs random field with respect to a neighborhood system of cliques.

$$P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y}) = e^{-U(\mathbf{x} \mid \mathbf{y})}/Z_{\mathbf{y}}, \tag{2}$$

where $Z_{\mathbf{y}}$ is a normalizing constant, and the corresponding energy function is

$$U(\mathbf{x} \mid \mathbf{y}) = \sum_{t=1}^{MN} [\frac{1}{2} \ln(\sigma_{x_t}^2) + \frac{(y_t - \mu_{x_t})^2}{2\sigma_{x_t}^2} + \sum_{r=1}^{c} [\theta_r J(x_t, x_{t:+r})]] \tag{3}$$

The local properties of an MRF can be derived from the Gibbs random field. Let $\mathbf{X}_{\partial t}$ be a (vector) random variable representing the gray levels of neighbors of pixel t, denoted by $\{x_{t:+r}, x_{t:-r}\}$ for r from 1 to c. The conditional probability of X_t can be written as follows [1].

$$P(X_t = x_t \mid \mathbf{X}_{\partial t} = \mathbf{x}_{\partial t}, \mathbf{Y} = \mathbf{y}) = e^{-U_t(x_t, \mathbf{x}_{\partial t}, \mathbf{y})}/Z_t,$$

where Z_t is a normalizing constant, and

$$U_t(x_t, \mathbf{x}_{\partial t}, \mathbf{y}) = \frac{1}{2} \ln(\sigma_{x_t}^2) + \frac{(y_t - \mu_{x_t})^2}{2\sigma_{x_t}^2} + \sum_{r=1}^{c} \theta_r [J(x_t, x_{t:+r}) + J(x_t, x_{t:-r})] \tag{4}$$

The segmentation problem can now be stated as the problem of observing vector $\mathbf{y}$ and estimating the labels in the perfect image. The MAP (Maximum A-Posteriori) estimate is the vector $\hat{\mathbf{x}}$ which maximizes $P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y})$ with respect to $\mathbf{x}$. Maximizing a function of MN variables is a formidable task. Section 3 defines three algorithms for approximating the MAP estimate. Contextual information is represented by the rightmost sums in Eq. (3) and (4). If these sums were removed, segmentation algorithms would assign labels independently to each pixel.

3. Pixel Labeling Algorithms

This section defines the MRF-based segmentation algorithms compared in this paper. These iterative algorithms attempt to optimize a statistical criterion by approximating the MAP estimate.

3.1. Simulated Annealing (SA)

Geman and Geman [8] proposed an algorithm based on simulated annealing to find the MAP estimate of the true image, which minimizes the energy function $U(\mathbf{x} \mid \mathbf{y})$ in Eq. (3) over all C^{MN} possible labelings $\mathbf{x}$. An exhaustive search for a global optimum creates 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 can 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 (1953) method of simulating systems containing large numbers of particles. Van Laarhoven and Aarts [14] provide a complete background and a summary of all aspects of simulated annealing algorithms. The algorithm given below was used in this paper.

Algorithm for MAP Estimation by Simulated Annealing

(1) Choose an initial temperature T.

(2) Initialize $\hat{\mathbf{x}}$ by maximizing $f_t(y_t \mid x_t)$ for each pixel t.
(This is the maximum-likelihood estimate of pixel label.)

(3) Perturb $\hat{\mathbf{x}}$ into $\hat{\mathbf{z}}$. Let

$$\Delta = U(\hat{\mathbf{z}} \mid \mathbf{y}) - U(\hat{\mathbf{x}} \mid \mathbf{y})$$

If $\Delta > 0$ then replace $\hat{\mathbf{x}}$ by $\hat{\mathbf{z}}$;
else replace $\hat{\mathbf{x}}$ by $\hat{\mathbf{z}}$ with probability $e^{\Delta/T}$.

(4) Repeat (3) N_{inner} times.

(5) Replace T by $\phi(T)$ where ϕ is a monotonically decreasing function.

(6) Repeat (3)-(5) K_{max} times.

The a-posteriori distribution of $\mathbf{X}$, given $\mathbf{Y} = \mathbf{y}$, must be a Gibbs random field for this algorithm to work. The energy function $U(\cdot \mid \mathbf{y})$ is defined in Eq. (3). It is important to realize that the partition function $Z_{\mathbf{y}}$ need not be computed or estimated to compute Δ.

The choice of function ϕ that defines the cooling schedule, the initial temperature T, the number of inner loops N_{inner}, the number of outer loops K_{max}, and the perturbation method must all be chosen experimentally to simulate the annealing process. Changing temperature slowly permits the estimator to escape local maxima and seek a global maximum. To perturb $\hat{\mathbf{x}}$ into $\hat{\mathbf{z}}$, a single pixel was selected and a new label was randomly assigned.

3.2 Iterated Conditional Modes (ICM).

Besag [2] proposed the ICM method as a computationally feasible alternative to MAP estimation. One difficulty with MRF a-priori models is their tendency to exhibit a phenomenon known as phase transition, in which realizations of the process are "uni-color" images [10]. The ICM segmentation algorithm was specifically designed to overcome this tendency [2].

The key to the ICM method is the following equation of proportionality for the probability of the label at pixel t, given the observed image $\mathbf{y}$ *and* the current estimates $\mathbf{x}_{\partial t}$ of the labels of all pixels in the neighborhood of pixel t. The notation $\mathbf{x}_{S\setminus t}$ refers to the labels of all pixels in the image, excluding the label at site t itself.

$$P(X_t = x_t \mid \mathbf{y}, \mathbf{X}_{S\setminus t} = \mathbf{x}_{S\setminus t}) \propto f_t(y_t \mid x_t) P(X_t = x_t \mid \mathbf{X}_{\partial t} = \mathbf{x}_{\partial t}) \qquad (5)$$

Maximizing the conditional probability in Eq. (5) is equivalent to minimizing the energy function in Eq. (4). The ICM algorithm is described below. The MN-vector $\mathbf{y}$ is given and an MN-vector of estimated pixel labels $\hat{\mathbf{x}}$ is computed.

Algorithm for Estimating Pixel Labels by ICM Method

(1) Initialize $\hat{\mathbf{x}}$ by maximizing $f_t(y_t \mid x_t)$ for each pixel t.
(2) For t from 1 to MN
  Update $\hat{x}_t$ to the value of x_t that maximizes $U_t(x_t, \mathbf{x}_{\partial t}, \mathbf{y})$.
(3) Repeat (2) N_{iter} times.

Experience has shown that 5 or 6 raster scans of an image are sufficient for convergence. The computation is a few orders of magnitude faster than the simulated annealing approach, and, as shown in Section 4, produces reasonable results.

The most difficult part of applying MRF-based segmentation algorithms is choosing the prior model. As an illustration, Besag [2] suggests using a simple, one-parameter pairwise-interaction model, with all the θ's in Eq. (4) equal to β. A second-order neighborhood is used so the label of a pixel is influenced by the labels of its eight nearest neighboring pixels. The larger β, the greater the influence of neighboring pixels. Besag [2] allows β to vary with the iteration. If one has detailed prior information and can choose an accurate prior model for the true labeling, the ICM method should work even better than with this one-parameter model. However, the problem of translating statements about the true labels and region shapes and sizes into parameter values for a MRF has not been solved.

3.3 Maximizer of Posterior Marginals (MPM).

Marroquin et al. [13] used an MRF model for the true labels but avoided the computational difficulties inherent in MAP estimation by proposing to minimize segmentation error, ε. Let $\delta(z) = 1$ if $z = 0$ and 0 otherwise and let $\mathbf{x}^*$ be the perfect image. A labeling $\hat{\mathbf{x}}$ is sought which minimizes

$$\varepsilon = \frac{1}{MN} \sum_{t=1}^{MN} [1 - \delta(x_t^* - \hat{x}_t)] \qquad (6)$$

which is the number of mistakes in the labeling $\hat{\mathbf{x}}$.

The MPM approach requires that the a-posteriori distribution of $\mathbf{X}$, given noisy image $\mathbf{y}$, be an MRF. The labeling that minimizes segmentation error can be shown to maximize the marginal a-posteriori distribution so the label $\hat{x}_t$ at pixel t is taken to be the one that satisfies:

$$P(X_t = \hat{x}_t \mid \mathbf{Y} = \mathbf{y}) \geq P(X_t = x_t \mid \mathbf{Y} = \mathbf{y}) \text{ for all } x_t$$

Besag [2] called the approach to maximizing this marginal conditional probability the MMP (Maximum Marginal Probability) method and viewed it as an approximation to MAP estimation. The point of departure of MPM is the manner in which this marginal conditional distribution is computed. Marroquin et al. [13] propose to simulate the MRF representing the a-posteriori distribution of $\mathbf{X}$, given $\mathbf{Y} = \mathbf{y}$ by a sampling algorithm such as the Gibbs sampler [8] which simulates a Markov chain over the states representing all possible labelings. Once this 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 configurations. Specifically, if $x_t^{(i)}$ is the label achieved at iteration i of the simulation, the a-posteriori marginal distribution of X_t is approximated as follows.

$$\hat{P}(X_t = g \mid \mathbf{Y} = \mathbf{y}) = \frac{1}{n-k} \sum_{i=k+1}^{n} \delta(x_t^{(i)} - g) \qquad (7)$$

Parameter k is the number of iterations needed for the Markov chain to reach steady state and n is chosen large enough for accurate estimation at reasonable computational cost. Both k and n are chosen heuristically. This approach is similar to simulated annealing at a single temperature. The algorithm is summarized below.

Algorithm for Estimating Pixel Labels by MPM Method

(1) Initialize $\hat{\mathbf{x}}$ by maximizing $f_t(y_t \mid x_t)$ for each pixel t.
(2) For site s from 1 to MN
  (a) Choose $g \in A$ at random and let $z_s = g$. Let $z_r = \hat{x}_r$ for all $r \neq s$.
  (b) Let $p = \min\{1, \dfrac{P(\mathbf{X} = \mathbf{z} \mid \mathbf{Y} = \mathbf{y})}{P(\mathbf{X} = \hat{\mathbf{x}} \mid \mathbf{Y} = \mathbf{y})}\}$
  (c) Replace $\hat{\mathbf{x}}$ by $\mathbf{z}$ with probability p.
(3) Repeat (2) n times, saving realizations $\hat{\mathbf{x}}^{(k+1)}$ to $\hat{\mathbf{x}}^{(n)}$.
(4) Form $\hat{P}(X_t = g \mid \mathbf{Y} = \mathbf{y})$ from Eq. (7) for all g.
(5) For site t from 1 to MN
  Choose label estimate $\hat{x}_t$ so that, for all labels g,

$$\hat{P}(X_t = \hat{x}_t \mid \mathbf{Y} = \mathbf{y}) \geq \hat{P}(X_t = g \mid \mathbf{Y} = \mathbf{y})$$

In addition to choosing the parameters of the prior distribution and the parameters relating observations to the label process, one must select the "magic" numbers k and n. The MPM algorithm requires more computation than the ICM algorithm, but far less computation than simulated annealing.

3.4. Other MRF Model-Based Pixel Labeling Algorithms

The SA, ICM, and MPM algorithms are representative of the class of MRF-based segmentation algorithms. Several other algorithms related to these three have been proposed. Derin and Elliott [6] expressed the a-posteriori distribution of $\mathbf{X}$, given $\mathbf{Y}$ as a recursive function and estimated an optimal labeling by dynamic programming. The heuristic used in the procedure does not guarantee either a global or a local optimum. Cohen and Cooper [5] assumed an auto-binomial [1] prior model for $\mathbf{X}$ and an auto-normal [1] distribution for the conditional density of $\mathbf{Y}$ given $\mathbf{X}$. They proposed a hierarchical algorithm for labeling pixels. The algorithms in [5] are essentially ICM-based algorithms. Lakshmanan and Derin [11] simultaneously estimated parameters for the MRF prior distribution and seg-

mented the image. It is clear that a variety of MRF model-based approaches are possible and that they differ in the type of information that must be provided to the algorithm and in computational details.

4. Experiments

Our experiments are based on 64×64 perfect images containing two gray levels (100 and 160) representing "object" and "background" and on two gray-level real images captured by a CCD camera. The two perfect images consist of two regions; the first is a checkerboard image, and the second image consists of a numeral that captures 11% of the pixels. We assume an isotropic second-order Ising model [10] for the prior distribution of **X** so the parameters in Eq. (1) are $c = 4$ and $\theta_1 = \theta_2 = \theta_3 = \theta_4 = \beta$. We have considered two values of β, 0.4 or 1.5, which are kept constant through each segmentation. A realization of an MRF with β=1.5 tends to have very unbalanced numbers of labels because of phase transition [10]. The values of other parameters are given below. In Algorithm SA, $K_{max} = 300$, $N_{inner} = 200$, $T_1 = 2$, and $T_{k+1} = \phi(T_k) = T_1/ln(1+k)$ for $k \geq 2$. Algorithm ICM is stopped after 6 complete scans. The magic numbers for MPM are $k = 200$ and $n = 300$.

In addition to judging the results of pixel labeling by visual inspection, we recorded the energy function of the a-posteriori distribution evaluated at the estimated labeling, $\hat{\mathbf{x}}$, $U(\hat{\mathbf{x}}|\mathbf{y})$. The smaller this energy, the better $\hat{\mathbf{x}}$ approximates the MAP estimate. However, energy is not always minimized at the visually optimal labeling.

Section 4.1 describes experiments when the MRF model parameters and the parameters controlling the degradation in the observed image are all known. Since the ICM algorithm required least computation and generally provides reasonable segmentations, we extended our study of the ICM algorithm to situations in which the parameters were learned in Section 4.2 and to real images in Section 4.3.

4.1. Synthetic Images with Known Parameters

Three experiments were performed to compare the three segmentation algorithms. Experiment 1 creates images that follow the noise model described in Section 2. Experiment 2 uses correlated noise and Experiment 3 represents each region by a texture. The assumptions of Section 2, under which the segmentation algorithms are derived, do not hold in the latter two experiments. Thus, we are viewing the algorithms as general purpose segmentation algorithms and testing their robustness in Experiments 2 and 3.

Experiment-1 Binary Images With i.i.d. Gaussian Noise

The images segmented in this experiment are obtained by adding a sample of a Gaussian random variable (mean 0, variance 3600) independently to each pixel in a binary image having gray levels 100 and 160 (SNR = 1). This implies that in Eqs. (3) and (4), $x_t \in \{1,2\}$, $\mu_1 = 100$, $\mu_2 = 160$, $\sigma_1 = \sigma_2 = 60$. Figures 2 and 3 each show a perfect image, the noisy, or observed image, the MLE image (obtained by labeling each pixel independently) that serves as the starting point for all algorithms, and the results of the three segmentation algorithms for two values of β. The noise has a standard deviation equal to the difference of the gray levels in the perfect image so the histograms for the noisy images are unimodal. Thus, simple thresholding methods are of no value. Figures 2(c) and 3(c) demonstrate that MLE estimates do not segment accurately.

Table 1 summarizes our visual comparison of the segmentations in Figures 2 and 3. Symbol "=" means "judged similar". We emphasize that these assessments are inherently subjective. Entries

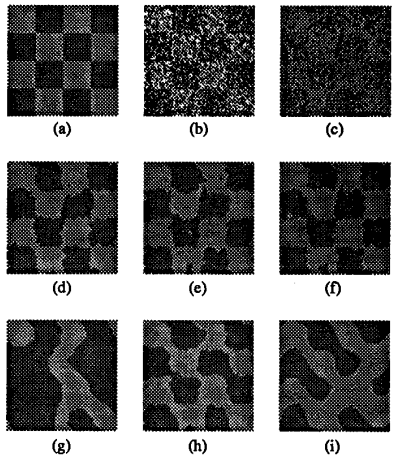

Figure 2. Segmentation results with i.i.d. noise
(a) perfect image (b) noisy image (c) MLE (d) SA ($\beta = 0.4$) (e) ICM ($\beta = 0.4$) (f) MPM ($\beta = 0.4$) (g) SA ($\beta = 1.5$) (h) ICM ($\beta = 1.5$) (i) MPM ($\beta = 1.5$)

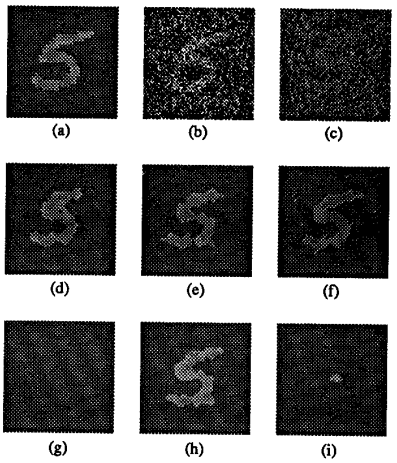

Figure 3. Segmentation results with i.i.d. noise
(a) perfect image (b) noisy image (c) MLE (d) SA ($\beta = 0.4$) (e) ICM ($\beta = 0.4$) (f) MPM ($\beta = 0.4$) (g) SA ($\beta = 1.5$) (h) ICM ($\beta = 1.5$) (i) MPM ($\beta = 1.5$)

Table 1. Summary of Segmentation Comparisons by Visual Inspection

Experiment	Figure(s)	β	Assessment
1	2-3	0.4	ICM=SA=MPM
	2,3	1.5	ICM
2	4	0.4	ICM=SA=MPM
	4	1.5	ICM
3	5	1.5	ICM
	5	0.4	None

"ICM" in Table 1 indicate that only the ICM method produced a reasonable segmentation. Table 2 shows energies of a-posteriori distributions (see Eq. (3)) along with the energy function evaluated for the perfect image. Note that the energy function is not a reliable indica-

tor of segmentation quality. For example the SA algorithm, when $\beta = 1.5$, produced the smallest energy for the checkerboard in Figure 2(g), but the segmentation was the worst visually. Curiously, all three algorithms terminated with images having lower energy than the energy of the perfect image when $\beta = 1.5$ for the checkerboard.

Table 2. Values of Energy function for Segmentations

Experiment	Figure	β	SA	ICM	MPM	Perfect
1	2	1.5	-4135	-3614	-3864	-3531
	3	1.5	-5503	-4979	-5432	-4877
2	4	0.4	12861	12907	12965	13066
	4	1.5	-4332	-3440	-3785	-3336
3	5	1.5	-7511	-7023	-7453	-7081

Experiment-2 Binary Images With Correlated Noise

Experiment 1 is generalized by changing the additive noise process to a first-order Gaussian Markov Random Field [1] with mean 0, variance 3600, and parameters $\theta_1 = \theta_2 = 0.2$. This imposes a correlation between the noise added to neighboring pixels, independent of the true image. As in Experiment 1, the gray level histograms of the noisy images are unimodal with strong clipping at levels 0 and 255 and are useless for segmentation.

Figure 4 exhibits the results of segmenting the noisy checkerboard. The minimum energy solution was not visually the best in all situations in this experiment. Comparing the results of Experiments 1 and 2 suggests that segmentations under the correct noise model (Experiment 1) are better than results under an incorrect model. However, the algorithms are reasonably robust to the level of correlated noise used in this experiment.

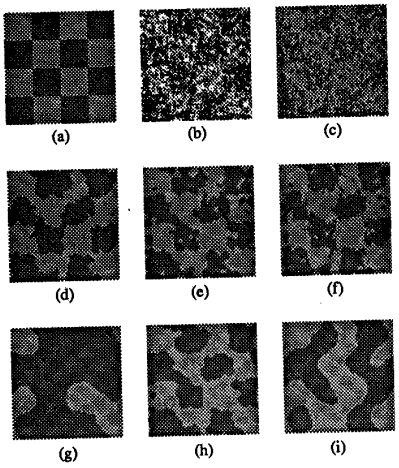

Figure 4. Segmentation results with correlated noise
(a) perfect image (b) noisy image (c) MLE (d) SA ($\beta = 0.4$) (e) ICM ($\beta = 0.4$) (f) MPM ($\beta = 0.4$) (g) SA ($\beta = 1.5$) (h) ICM ($\beta = 1.5$) (i) MPM ($\beta = 1.5$)

Experiment-3 Textured Regions

Experiment 3 violates all the assumptions under which the three algorithms were derived and presents a severe test of robustness. Two 64×64 textured images were generated from a first-order Gaussian Markov Random Field with parameters $\{\mu, \sigma, \theta_1\ \theta_2\}$ of {160, 20, 0.4, -0.08} and {160, 40, -0.08, 0.4}. The two textures have the same mean gray levels but differ in variance and directionality. Pixels at level 100 in the perfect image were assigned gray levels from one process and pixels at level 160, from the other. Gray level histograms were unimodal with sharp peaks at 160 and significant clipping at level 255 and provide no information for segmentation.

Figure 5 shows segmentations of the numeral image. The segmentation algorithms do not incorporate the values of parameters θ_1 and θ_2 of the Gaussian Markov random fields which were used to generate the textured regions. Only the ICM algorithm performs reasonably under the severe conditions of this experiment.

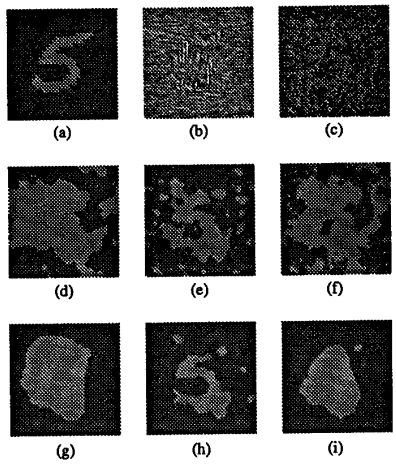

Figure 5. Segmentation results with textured regions
(a) perfect image (b) noisy image (c) MLE (d) SA ($\beta = 0.4$) (e) ICM ($\beta = 0.4$) (f) MPM ($\beta = 0.4$) (g) SA ($\beta = 1.5$) (h) ICM ($\beta = 1.5$) (i) MPM ($\beta = 1.5$)

In summary, none of the algorithms performed consistently best; ICM proved to be the most robust algorithm. A large value of β appears to be desirable with ICM. In a typical experiment on a 64×64 image, the computation time for ICM is about 6 seconds, while MPM needs 3.7 minutes and SA demands several hours on a Sun 4/280 computer. Thus, ICM has clear computational advantages.

4.2. Synthetic Images with Parameters Estimated

The results of Experiments 1-3 suggest that the ICM algorithm with β of 1.5 is the best tradeoff between performance and computational speed. However, all parameters of the noise process were assumed known in Experiment 1, which is not realistic in applications. This section tests the performance of ICM when the parameters are estimated by supervised and unsupervised procedures. Noisy images are generated as in Experiment 1, by adding i.i.d. Gaussian noise with mean 0 and variance 3600 to the two perfect images. The objective is to determine the effect of using estimated parameters on the performance of the ICM segmentation algorithm.

Experiment-4 Labeled Training Samples

The estimation problem is to learn the means $\{\mu_1, \mu_2\}$, and the variances $\{\sigma_1^2, \sigma_2^2\}$ of two normal distributions when the true means are {100, 160} and the true variance for both distributions is 3600. We randomly selected 100 pixels from each region, observed the noisy gray levels at these pixels, computed the maximum likelihood estimates for the means and variances, and used the estimates in the ICM segmentation algorithm with β of 1.5. This simulates the situation when an observer can correctly classify about 5% of the pixels by eye. Rather than repeating this process with both images, we did

Figure 6. Segmentation results of ICM using supervised learning for images in Figures 2-3(b)

it once and used the same estimates for both noisy images. The estimated parameters are $\{\hat{\mu}_1, \hat{\mu}_2\} = \{104, 158\}$ and $\{\hat{\sigma}_1^2, \hat{\sigma}_2^2\} = \{2964, 2838\}$. The segmentations are shown in Figure 6. A visual comparison with the results of Experiment 1 (Figures 2(h)-3(h)) indicates that ICM performs as well with estimated parameters as with known parameters when the noise model is known.

Experiment-5 Unlabeled Training Samples

The assumed knowledge for parameter estimation here is more realistic than in Experiment 4 since no classified training pixels are available. The problem of estimating parameters can be treated as the classical problem of resolving a Gaussian mixture. As in Experiment 4, the parameters to be estimated are the means and the variances of the two regions. In this experiment, however, we also estimate the parameter β of the MRF model. We choose to reformulate the ICM algorithm to incorporate unsupervised estimates as defined in the following algorithm.

Modified ICM Algorithm with Unsupervised Estimation

(1) Obtain an initial segmentation. Initialize $\hat{\beta}$.
(2) Obtain parameter estimates $\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2$ by maximum likelihood estimation.
(3) Perform Step (2) of the ICM algorithm (Section 3.2) using estimated parameters $\{\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\beta}\}$.
(4) Obtain $\hat{\beta}$ by the coding method [1]
(5) Repeat Steps (2)-(4) above N_{iter} times.

An initial segmentation of the image (Step 1 of the algorithm) can be obtained by several methods. We use the histogram concavity analysis method to automatically find the valley in the histogram. Since our level of noise results in a unimodal gray level histogram, we create a modified histogram whose bin values are the geometric means of the corresponding bin values of histograms at three successive image resolutions (1×1, 2×2, 4×4). The modified histogram suppresses the effect of noise and indicates the correct location of the valley. This method has been remarkably successful in locating proper threshold values for an initial segmentation. The initial histogram was unimodal with strong clipping at level 0, and conveys misleading information about the threshold value.

Estimating μ_i and σ_i is straightforward and computationally inexpensive. However, procedures for estimating β are more complex and computationally intensive. To study the effect of estimating β, we considered the following three variations of the modified algorithm specified above:

(i) $\hat{\beta}$ in Step 1 is obtained from the initial segmentation by the Coding method.

(ii) $\hat{\beta}$ in Step 1 is set to 1.5.

(iii) $\hat{\beta}$ is fixed at 1.5 throughout the algorithm (*i.e.*, Step 4 is omitted).

Segmentation results are given in Figure 7. The segmentations for all the images compare favorably with those from supervised learning in Figure 6 and known parameters in Figures 2-3. The numeral image is an exception. Option (i) for estimating β does not segment the numeral image properly (Figure 7(b)). In this case, $\hat{\beta}$ estimated from the initial segmentation is very small (0.13), and remains small throughout ending up with a value of 0.185. As a result, the contextual information is not fully utilized. Options (ii) and (iii) result in good segmentations for the numeral image. The final value of $\hat{\beta}$ for the numeral image with option (ii) is 2.815. Option (iii) is computationally inexpensive and should be preferred to option (ii). We conclude that it is possible to learn the parameters of the ICM algorithm in an unsupervised manner when the noise is i.i.d. Gaussian.

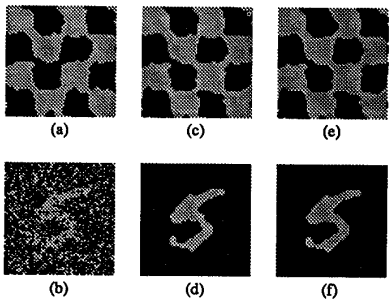

Figure 7. Segmentation results of ICM using supervised learning for images in Figures 2-3(b)
(a)-(b): β estimated in all iterations, (c)-(d): β estimated but initialized to 1.5, (e)-(f): β not estimated, but set to 1.5

4.3. Segmentation of Real Images

Segmenting a real image is a more difficult problem than segmenting a noisy version of a perfect test image. In our experiments, a perfect test image has only two gray levels corresponding to the two labels "object" and "background". The real images we have used also have two labels but each region supports a different distribution of gray levels. That is, the noise process is not identical for the two regions (labels). In addition, the shadow effects and gray level variations in the object and the background imply that the assumption of independent noise is questionable.

Experiment-6 ICM with Unsupervised Estimation

Two real images were used in this experiment. The first is a 120×128 image containing four flat industrial parts and the second is a 256×256 image of printed characters. Option (iii) of the modified ICM algorithm given in Section 4.2 ($\hat{\beta} = 1.5$) was used to segment the images.

The results are given in Figures 8 and 9. The histogram of the "printed characters" image is bimodal whereas that of the "industrial parts" image is essentially flat with a large peak near gray level zero. Figures 8(c) and 9(c) show the initial segmentations (obtained by Step 1 of the modified ICM algorithm), and Figures 8(d) and 9(d) show the final segmentations. For the "industrial parts" image, the initial segmentation is slightly better than the final segmentation. This shows that the use of contextual information by the ICM algorithm does not always improve the segmentation results. For the "printed characters" image, the ICM segmentation is better than the initial segmentation. The final segmentations for both the images are of acceptable quality. We also ran MPM and SA algorithms on the "industrial parts" image, but those segmentations were of no better quality than the ICM segmentation. We conjecture that the failure of these algorithms to enhance the quality of the segmentation was due to the violation of the noise model on which the algorithms are based.

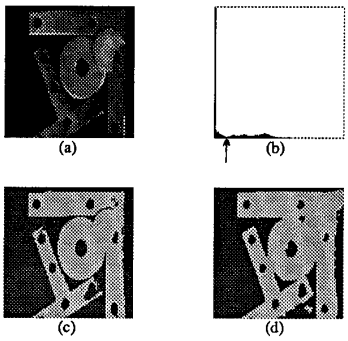

Figure 8. Segmentation results of "Industrial parts" image
(a) Observed image (b) gray level histogram of the image (c) Thresholding of the image at the gray level indicated by the arrow in the histogram (d) ICM segmentation with parameters estimated by unsupervised learning

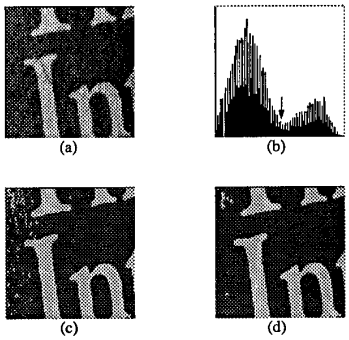

Figure 9. Segmentation results of "Printed characters" image
(a) Observed image (b) gray level histogram of the image (c) Thresholding of the image at the gray level indicated by the arrow in the histogram (d) ICM segmentation with parameters estimated by unsupervised learning

5. Conclusion and Discussion

We have reviewed and listed three MRF model-based pixel labeling algorithms which approximate the MAP segmentation. The three algorithms are derived under the i.i.d additive Gaussian noise assumption. The ICM algorithm performed consistently well when the synthetic images were corrupted with i.i.d. Gaussian noise and the SNR = 1. The robustness of the algorithms was tested on images corrupted by correlated noise and on images with textured regions. The ICM algorithm provided reasonable segmentation under these conditions and was the most robust.

These MRF model-based labeling algorithms require that the parameters of the image model and the degradation model be known as well as the number of labels. Our experiments indicate that the parameters of the degradation model can be estimated reliably for SNR of nearly 1, both in supervised and unsupervised modes under i.i.d. additive Gaussian noise. The parameter β of the ICM algorithm could not be estimated reliably for one of the images but could be adjusted from image data so as to improve segmentation. However, the small improvement in quality may not justify computational overhead.

Intuition suggests that the smaller the energy of the a-posteriori distribution, the better the segmentation. Our limited experiments showed that the true labeling need not have the smallest energy and the labeling with the smallest energy need not be visually the best. The phase transition phenomenon [2, 10], by which realizations of a Markov Random Field tend to be single-label images when the parameters are large, may cause the poor performance of SA and MPM when β is 1.5. The ICM algorithm was designed to be impervious to phase-transition [2] and our experiments show this characteristic.

Our experiments with real images indicate that all three algorithms may fail to enhance the quality of segmentation for real images when the assumed noise model is violated.

REFERENCES

[1] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. R. Statist. Soc. Ser.B*, Vol. 36, pp. 192-236, 1974.

[2] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. R. Statist. Soc. Ser.B*, Vol. 48, pp. 259-302, 1986.

[3] C. C. Chen and R.C. Dubes, "Experiments in Fitting Discrete Markov Random Fields to Textures," *Proc. CVPR '89*, pp. 298-303, 1989.

[4] P. B. Chow, C. M. Brown, and R. Raman, "A Confidence-Based Approach to the Labeling Problem," *IEEE Workshop on Computer Vision*, Miami, pp. 51-56, 1987.

[5] F. S. Cohen and D. B. Cooper, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9, pp. 195-219, 1987.

[6] H. Derin and H. Elliott, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9, pp. 39-55, 1987.

[7] R. C. Dubes and A. K. Jain, "Random Field Models in Image Analysis,"
J. Applied Statistics, Vol. 16, pp. 131-164, 1989.

[8] S. Geman and D. Geman, "Stochastic Relaxation: Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, pp. 721-741, 1984.

[9] R. M. Haralick and L. G. Shapiro, "Survey: Image Segmentation," *Computer Vision, Graphics, and Image Processing*, Vol. 29, pp. 100-132, 1985.

[10] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, American Mathematical Society, Volume I, 1980.

[11] S. Lakshmanan and H. Derin, "Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, pp. 799-813, 1989.

[12] K. V. Mardia and T. J. Hainsworth, "A Spatial Thresholding Method for Image Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 10, pp. 919-927, 1988.

[13] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *J. American Statistical Association*, Vol. 82, pp. 76-89, 1987.

[14] P. J. M. Van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Dordrecht: D. Reidel, 1987.