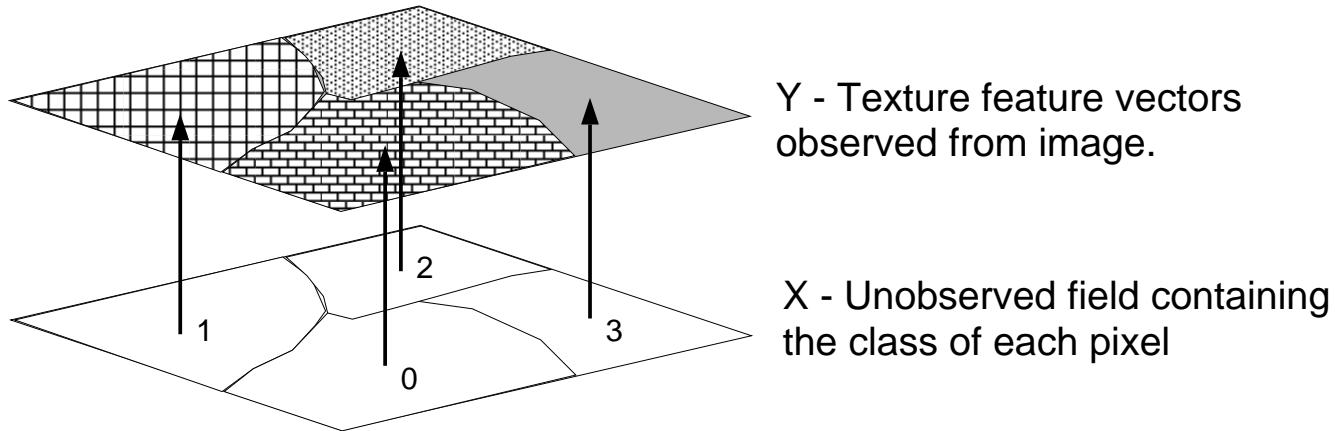


# Application of MRF's to Segmentation

- Topics to be covered:
  - The Model
  - Bayesian Estimation
  - MAP Optimization
  - Parameter Estimation
  - Other Approaches

# Bayesian Segmentation Model



- Discrete MRF is used to model the segmentation field.
- Each class is represented by a value  $X_s \in \{0, \dots, M - 1\}$
- The joint probability of the data and segmentation is

$$P\{Y \in dy, X = x\} = p(y|x)p(x)$$

where

- $p(y|x)$  is the data model
- $p(x)$  is the segmentation model

## Bayes Estimation

- $C(x, X)$  is the cost of guessing  $x$  when  $X$  is the correct answer.
- $\hat{X}$  is the estimated value of  $X$ .
- $E[C(\hat{X}, X)]$  is the expected cost (risk).
- Objective: Choose the estimator  $\hat{X}$  which minimizes  $E[C(\hat{X}, X)]$ .

## Maximum *A Posteriori* (MAP) Estimation

- Let  $C(x, X) = \delta(x \neq X)$
- Then the optimum estimator is given by

$$\begin{aligned}\hat{X}_{MAP} &= \arg \max_x p_{x|y}(x|Y) \\ &= \arg \max_x \log \frac{p_{y,x}(Y, x)}{p_y(Y)} \\ &= \arg \max_x \{ \log p(Y|x) + \log p(x) \}\end{aligned}$$

- Advantage:
  - Can be computed through direct optimization
- Disadvantage:
  - Cost function is unreasonable for many applications

## Maximizer of the Posterior Marginals (MPM) Estimation[12]

- Let  $C(x, X) = \sum_{s \in S} \delta(x_s \neq X_s)$
- Then the optimum estimator is given by

$$\hat{X}_{MPM} = \arg \max_{\hat{x}_s} p_{x_s|Y}(x_s|Y)$$

- Compute the most likely class for each pixel
- Method:
  - Use simulation method to generate samples from  $p_{x|y}(x|y)$ .
  - For each pixel, choose the most frequent class.
- Advantage:
  - Minimizes number of misclassified pixels
- Disadvantage:
  - Difficult to compute

# Simple Data Model for Segmentation

- Assume:
  - $x_s \in \{0, \dots, M-1\}$  is the class of pixel  $s$ .
  - $Y_s$  are independent Gaussian random variables with mean  $\mu_{x_s}$  and variance  $\sigma_{x_s}^2$ .

$$p_{y|x}(y|x) = \prod_{s \in S} \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left\{-\frac{1}{2\sigma_{x_s}^2} (y_s - \mu_{x_s})^2\right\}$$

- Then the negative log likelihood has the form

$$-\log p_{y|x}(y|x) = \sum_{s \in S} l(y_s|x_s)$$

where

$$l(y_s|x_s) = -\frac{1}{2\sigma_{x_s}^2} (y_s - \mu_{x_s})^2 - \frac{1}{2} \log(2\pi\sigma_{x_s}^2)$$

## More General Data Model for Segmentation

- Assume:
  - $Y_s$  are conditionally independent given the class labels  $X_s$
  - $X_s \in \{0, \dots, M-1\}$  is the class of pixel  $s$ .

- Then

$$-\log p_{y|x}(y|x) = \sum_{s \in S} l(y_s|x_s)$$

where

$$l(y_s|x_s) = -\log p_{y_s|x_s}(y_s|x_s)$$

## MAP Segmentation

- Assume a prior model for  $X \in \{0, \dots, M-1\}^{|S|}$  with the form

$$\begin{aligned} p_x(x) &= \frac{1}{Z} \exp\left\{-\beta \sum_{\{i,j\} \in \mathcal{C}} \delta(x_i \neq x_j)\right\} \\ &= \frac{1}{Z} \exp\{-\beta t_1(x)\} \end{aligned}$$

where  $\mathcal{C}$  is the set of 4-point neighboring pairs

- Then the MAP estimate has the form

$$\begin{aligned} \hat{x} &= \arg \min_x \left\{ -\log p_{y|x}(y|x) + \beta t_1(x) \right\} \\ &= \arg \min_x \left\{ \sum_{s \in S} l(y_s|x_s) + \beta \sum_{\{i,j\} \in \mathcal{C}} \delta(x_i \neq x_j) \right\} \end{aligned}$$

- This optimization problem is very difficult

# An Exact Solution to MAP Segmentation

- When  $M = 2$ , the MAP estimate can be solved exactly in polynomial time
  - See [9] for details.
  - Based on *minimum cut* problem and Ford-Fulkerson algorithm [5].
  - Works for general neighborhood dependencies
  - Only applies to binary segmentation case

# Approximate Solutions to MAP Segmentation

- Iterated Conditional Models (ICM) [2]
  - A form of iterative coordinate decent
  - Converges to a local minima of posterior probability
- Simulated Annealing [6]
  - Based on simulation method but with decreasing temperature
  - Capable of “climbing” out of local minima
  - Very computationally expensive
- MPM Segmentation [12]
  - Use simulation to compute approximate MPM estimate
  - Computationally expensive
- Multiscale Segmentation [3]
  - Search space of segmentations using a course-to-fine strategy
  - Fast and robust to local minima
- Other approaches
  - Dynamic programming does not work in 2-D, but approximate recursive solutions to MAP estimation exist[4, 13]
  - Mean field theory as approximation to MPM estimate[14]

## Iterated Conditional Modes (ICM) [2]

- Minimize cost function with respect to the pixel  $x_r$

$$\begin{aligned}
 \hat{x}_r &= \arg \min_{x_r} \left\{ \sum_{s \in S} l(y_s | x_s) + \beta \sum_{\{i,j\} \in \mathcal{C}} \delta(x_i \neq x_j) \right\} \\
 &= \arg \min_{x_r} \left\{ l(y_r | x_r) + \beta \sum_{s \in \partial r} \delta(x_s \neq x_r) \right\} \\
 &= \arg \min_{x_r} \{ l(y_r | x_r) + \beta v_1(x_r, x_{\partial r}) \}
 \end{aligned}$$

- Initialize with the ML estimate of  $X$

$$[\hat{x}_{ML}]_s = \arg \min_{0 \leq m < M} l(y_s | m)$$

# ICM Algorithm

## ICM Algorithm:

1. Initialize with ML estimate

$$x_s \leftarrow \arg \min_{0 \leq m < M} l(y_s | m)$$

2. Repeat until no changes occur

- (a) For each  $s \in S$

$$x_s \leftarrow \arg \min_{0 \leq m < M} \{l(y_s | m) + \beta v_1(m, x_{\partial s})\}$$

- For each pixel replacement, cost decreases  $\Rightarrow$  cost functional converges
- Variation: Only change pixel value when cost *strictly* decreases
- ICM + Variation  $\Rightarrow$  sequence of updates converge in finite time
- Problem: ICM is easily trapped in local minima of the cost functional

## Low Temperature Limit for Gibb Distribution

- Consider the Gibbs distribution for the discrete random field  $X$  with temperature parameter  $T$

$$p_T(x) = \frac{1}{Z} \exp \left\{ -\frac{1}{T} U(x) \right\}$$

- For  $x \neq \hat{x}_{MAP}$ , then  $U(\hat{x}_{MAP}) < U(x)$  and

$$\begin{aligned} \lim_{T \downarrow 0} \frac{p_T(\hat{x}_{MAP})}{p_T(x)} &= \lim_{T \downarrow 0} \exp \left\{ \frac{1}{T} (U(x) - U(\hat{x}_{MAP})) \right\} \\ &= \infty \end{aligned}$$

Since  $p_T(\hat{x}_{MAP}) \leq 1$ , we then know that  $x \neq \hat{x}_{MAP}$

$$\lim_{T \downarrow 0} p_T(x) = 0$$

So if  $\hat{x}_{MAP}$  is unique, then

$$\lim_{T \downarrow 0} p_T(\hat{x}_{MAP}) = 1$$

## Low Temperature Simulation

- Select “small” value of  $T$
- Use simulation method to generate sample  $X^*$  from the distribution

$$p_T(x) = \frac{1}{Z} \exp \left\{ -\frac{1}{T} U(x) \right\}$$

- Then  $p_T(X^*) \cong p_T(\hat{x}_{MAP})$

- Problem:

$T$  too large  $\Rightarrow X^*$  is far from MAP estimate

$T$  too small  $\Rightarrow$  convergence of simulation is **very** slow

- Solution:

Let  $T$  go to zero slowly

Known as simulated annealing

## Simulated Annealing with Gibbs Sampler[6]

### Gibbs Sampler Algorithm:

1. Set  $N = \#$  of pixels
2. Select “annealing schedule”: Decreasing sequence  $T_k$
3. Order the  $N$  pixels as  $N = s(0), \dots, s(N - 1)$
4. Repeat for  $k = 0$  to  $\infty$ 
  - (a) Form  $X^{(k+1)}$  from  $X^{(k)}$  via

$$X_r^{(k+1)} = \begin{cases} W & \text{if } r = s(k) \\ X_r^{(k)} & \text{if } r \neq s(k) \end{cases}$$

$$\text{where } W \sim p_{T_k} \left( x_{s(k)} \mid X_{\partial s(k)}^{(k)} \right)$$

- For example problem:

$$U(x) = \sum_{s \in S} l(y_s | x_s) + \beta t_1(x)$$

and

$$p_{T_k} (x_s | x_{\partial s}) = \frac{1}{z'} \exp \left\{ -\frac{1}{T_k} ( l(y_s | x_s) + \beta v_1(x_s, x_{\partial s}) ) \right\}$$

## Convergence of Simulated Annealing [6]

- Definitions:
  - $N$  - number of pixels
  - $\Delta = \arg \max_x U(x) - \arg \min_x U(x)$

- Let

$$T_k = \frac{N\Delta}{\log(k + 1)}$$

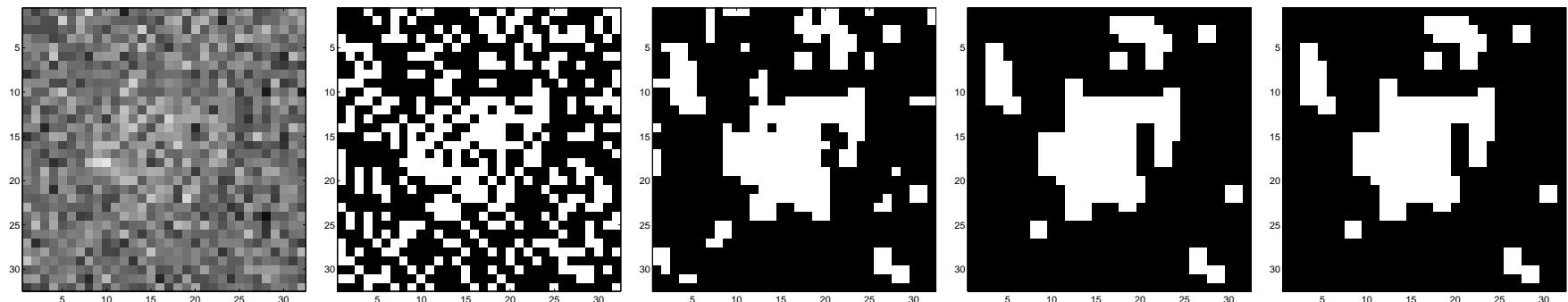
Theorem: The simulation converges to  $\hat{x}_{MAP}$  almost surely. [6]

- Problem: This is very slow!!!
- Example:  $N = 10000, \Delta = 1 \Rightarrow T_{e^{10000}-1} = 1/2$ .
- More typical annealing schedule that achieves approximate solution

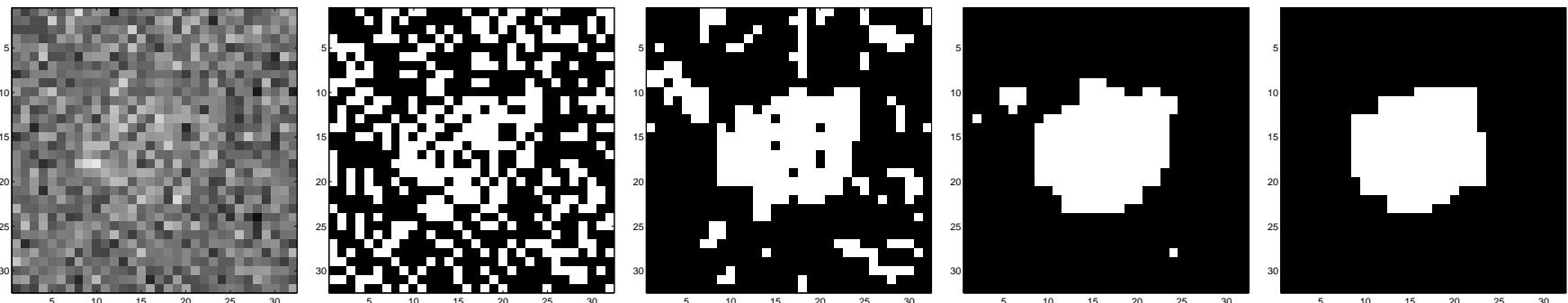
$$T_k = T_0 \left( \frac{T_K}{T_0} \right)^{k/K}$$

## Segmentation Example

- Iterated Conditional Modes (ICM): ML ; ICM 1; ICM 5; ICM 10



- Simulated Annealing (SA): ML ; SA 1; SA 5; SA 10



## Maximizer of the Posterior Marginals (MPM) Estimation[12]

- Let  $C(x, X) = \sum_{s \in S} \delta(x_s \neq X_s)$
- Then the optimum estimator is given by

$$\hat{X}_{MAP} = \arg \max_x p_{x_s|Y}(x_s|Y)$$

- Compute the most likely class for each pixel
- Method:
  - Use simulation method to generate samples from  $p_{x|y}(x|y)$ .
  - For each pixel, choose the most frequent class.
- Advantage:
  - Minimizes number of misclassified pixels
- Disadvantage:
  - Difficult to compute

## MPM Segmentation Algorithm [12]

- Define the function

$$X \leftarrow \text{Simulate}(X_{init}, p_{x|y}(x|y))$$

This function applies one full pass of a simulation algorithm with stationary distribution  $p_{x|y}(x|y)$  and starting with initial value  $X_{init}$ .

### MPM Algorithm:

1. Select parameters  $M_1$  and  $M_2$
2. For  $i = 0$  to  $M_1 - 1$ 
  - (a) Repeat  $M_2$  times

$$X \leftarrow \text{Simulate}(X, p_{x|y}(x|y))$$

- (b) Set  $X^{(i)} \leftarrow X$

3. For each  $s \in S$ , compute

$$\hat{x}_s \leftarrow \arg \max_{0 \leq m < M} \sum_{i=0}^{M_1-1} \delta(X^{(i)} = m)$$

# Multiscale MAP Segmentation

- Renormalization theory[8]
  - Theoretically results in the exact MAP segmentation
  - Requires the computation of intractable functions
  - Can be implemented with approximation
- Multiscale segmentation[3]
  - Performs ICM segmentation in a coarse-to-fine sequence
  - Each MAP optimization is initialized with the solution from the previous coarser resolution
  - Used the fact that a discrete MRF constrained to be block constant is still a MRF.
- Multiscale Markov random fields[10]
  - Extended MRF to the third dimension of scale
  - Formulated a parallel computational approach

## Multiscale Segmentation [3]

- Solve the optimization problem

$$\hat{x}_{MAP} = \arg \min_x \left\{ \sum_{s \in S} l(y_s | x_s) + \beta_1 t_1(x) + \beta_2 t_2(x) \right\}$$

- Break  $x$  into large blocks of pixels that can be changed simultaneously
- Make large scale moves can lead to
  - Faster convergence
  - Less tendency to be trapped in local minima

## Formulation of Multiscale Segmentation [3]

- Pixel blocks

- The  $s^{th}$  block of pixels

$$d^{(k)}(s) = \{(i, j) \in S : (\lfloor i/2^k \rfloor, \lfloor j/2^k \rfloor) = s\}$$

- Example: If  $k = 3$  and  $s = (0, 0)$ , then

$$d^{(k)}(s) = [(0, 0), \dots, (7, 0), (0, 1), \dots, (7, 1), \dots, (0, 7), \dots, (7, 7)]$$

- Coarse scale statistics:

- We say that  $x$  is  $2^k$ -block-constant if there exists an  $x^{(k)}$  such that for all  $r \in d^{(k)}(s)$

$$x_r = x_s^{(k)}$$

- Coarse scale likelihood functions

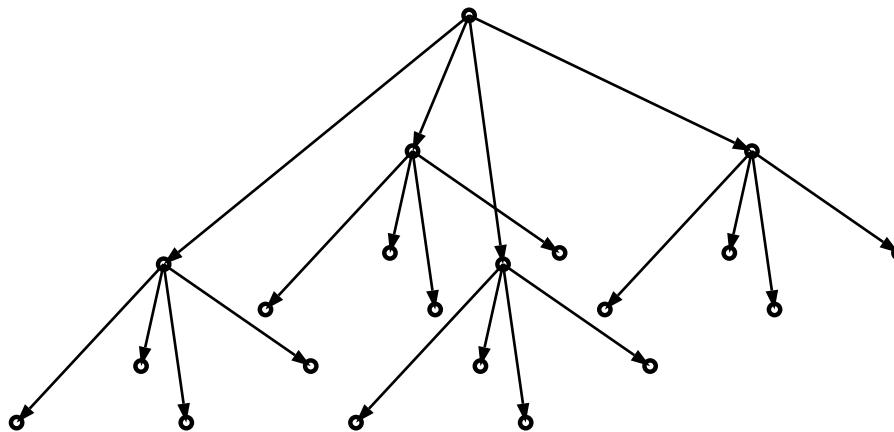
$$l_s^{(k)}(m) = \sum_{r \in d^{(k)}(s)} l(y_r | m)$$

- Coarse scale statistics

$$t_1^{(k)} \triangleq t_1(x^{(k)}) \quad t_2^{(k)} \triangleq t_2(x^{(k)})$$

## Recursions for Likelihood Function

- Organize blocks of image in quadtree structure



- Let  $d(s)$  denote the four children of  $s$ , then

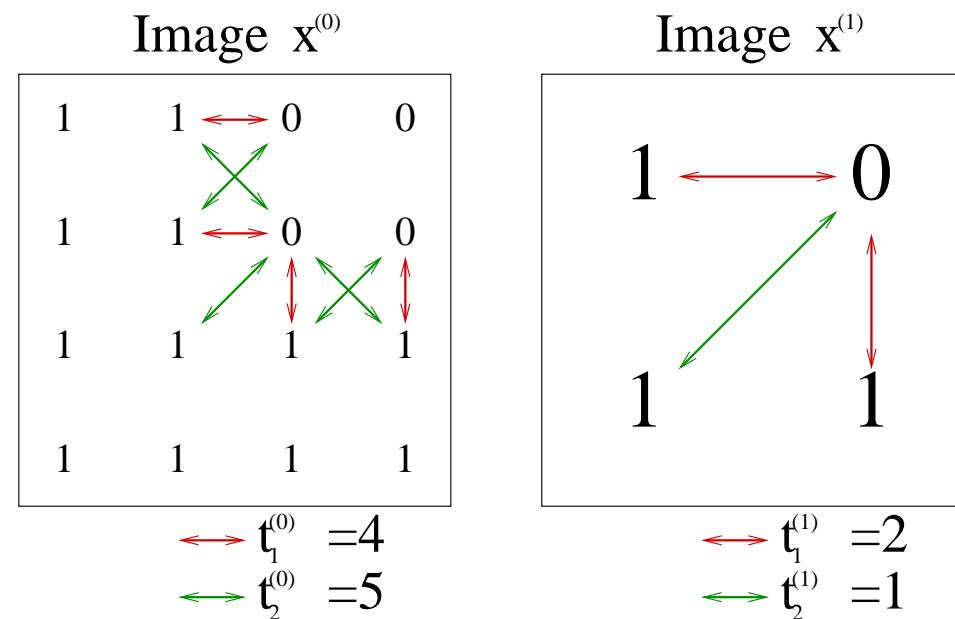
$$l_s^{(k)}(m) = \sum_{r \in d(s)} l_r^{(k-1)}(m)$$

where  $l_s^{(0)}(m) = l(y_s|m)$ .

- Complexity of recursion is order  $\mathcal{O}(N)$  for  $N = \#$  of pixels

# Recursions for MRF Statistics

- Count statistics at each scale



- If  $x$  is  $2^k$ -block-constant, then

$$\begin{aligned}
 t_1^{(k-1)} &= 2t_1^{(k)} \\
 t_2^{(k-1)} &= 2t_1^{(k)} + t_2^{(k)}
 \end{aligned}$$

## Parameter Scale Recursion [3]

- Assume  $x$  is  $2^k$ -block-constant. Then we would like to select parameters  $\beta_1^{(k)}$  and  $\beta_2^{(k)}$  so that the energy functions match at each scale.

This means that

$$\beta_1^{(k)} t_1^{(k)} + \beta_2^{(k)} t_2^{(k)} = \beta_1^{(k-1)} t_1^{(k-1)} + \beta_2^{(k-1)} t_2^{(k-1)}$$

- Substituting the recursions for  $t_1^{(k)}$  and  $t_2^{(k)}$  yeilds recursions for the parameters  $\beta_1^{(k)}$  and  $\beta_2^{(k)}$ .

$$\begin{aligned}\beta_1^{(k)} &= 2 \left( \beta_1^{(k-1)} + \beta_2^{(k-1)} \right) \\ \beta_2^{(k)} &= \beta_2^{(k-1)}\end{aligned}$$

- Courser scale  $\Rightarrow$  large  $\beta$   $\Rightarrow$  more smoothing
- Alternative approach: Leave  $\beta$ 's constant

## Multiple Resolution Segmentation (MRS) [3]

### MRS Algorithm:

1. Select coarsest scale  $L$  and parameters  $\beta_1^{(k)}$  and  $\beta_2^{(k)}$
2. Set  $l_s^{(0)}(m) \leftarrow l(y_s|m)$ .
3. For  $k = 1$  to  $L$ , compute:  $l_s^{(k)}(m) = \sum_{r \in d(s)} l_r^{(k-1)}(m)$
4. Compute ML estimate at scale  $L$ :  $\hat{x}_s^{(L)} \leftarrow \arg \min_{0 \leq m < M} l_s^{(L)}(m)$
5. For  $k = L$  to 0
  - (a) Perform ICM optimization using initial condition  $\hat{x}_s^{(L)}$  until converged
 
$$\hat{x}^{(k)} \leftarrow ICM \left( \hat{x}^{(k)}, u^{(k)}(\cdot) \right)$$

where

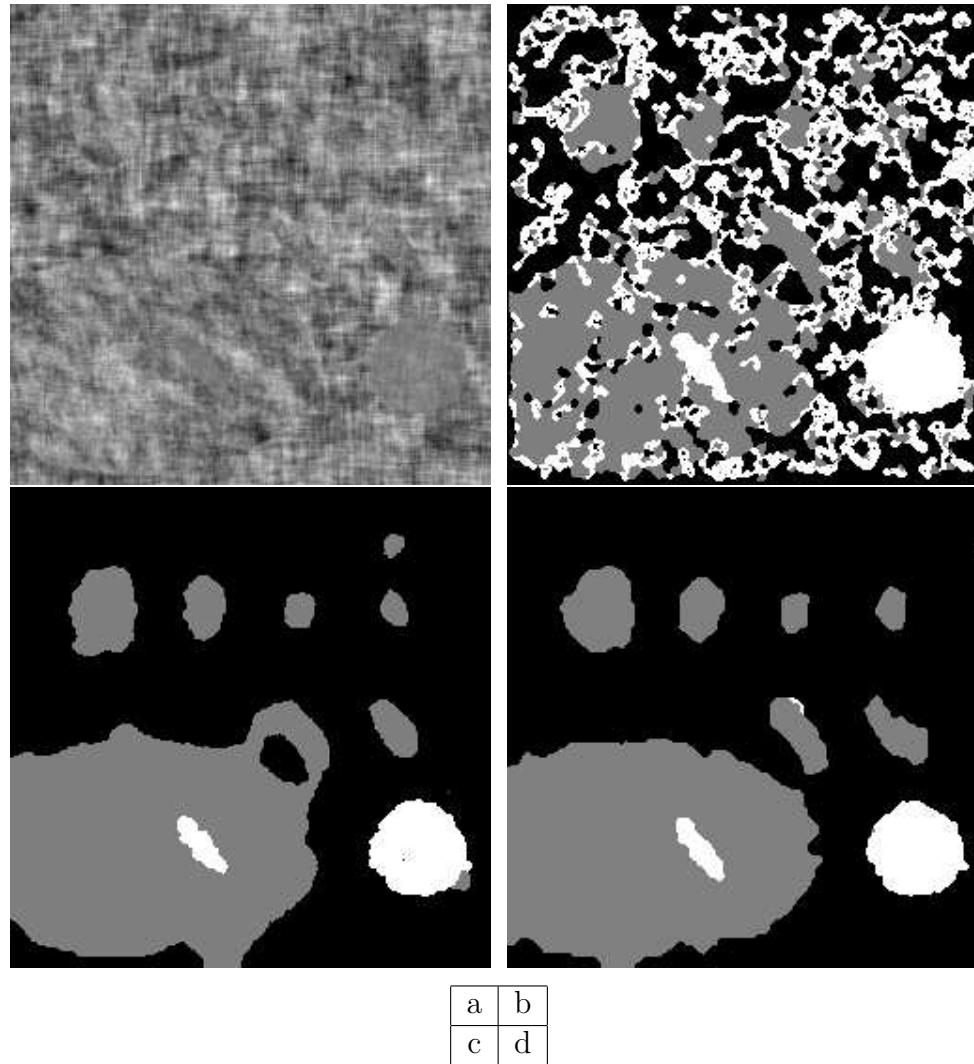
$$u^{(k)} \left( \hat{x}^{(k)} \right) = \sum_s l_s^{(k)}(\hat{x}_s^{(k)}) + \beta_2^{(k)} t_1^{(k)} + \beta_2^{(k)} t_2^{(k)}$$

- (b) if  $k > 0$  compute initial condition using block replication

$$\hat{x}^{(k-1)} \leftarrow Block\_Replication(\hat{x}^{(k)})$$

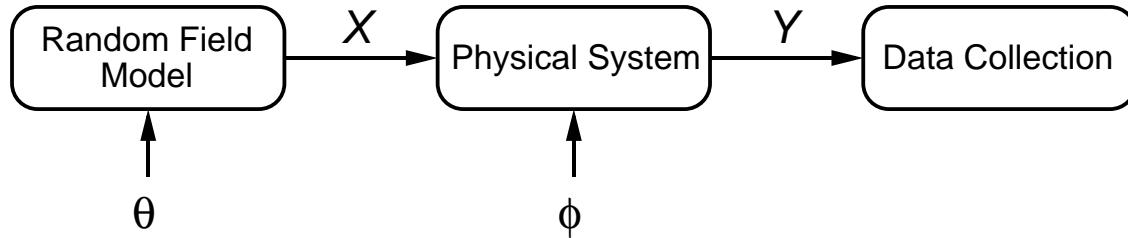
6. Output  $\hat{x}^{(0)}$

## Texture Segmentation Example



a) Synthetic image with 3 textures b) ICM - 29 iterations c) Simulated Annealing - 100 iterations d) Multiresolution - 7.8 iterations

# Parameter Estimation



- Question: How do we estimate  $\theta$  from  $Y$ ?
- Problem: We don't know  $X$ !
- Solution 1: Joint MAP estimation [11]

$$(\hat{\theta}, \hat{x}) = \arg \max_{\theta, x} p(y, x | \theta)$$

- Problem: The solution is biased.
- Solution 2: Expectation maximization algorithm [1, 7]

$$\hat{\theta}^{k+1} = \arg \max_{\theta} E[\log p(Y, X | \theta) | Y = y, \theta^k]$$

- Expectation may be computed using simulation techniques or mean field theory.

## References

- [1] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Statistics*, 37:1554–1563, 1966.
- [2] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3):259–302, 1986.
- [3] C. A. Bouman and B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2):99–113, February 1991.
- [4] H. Derin, H. Elliott, R. Cristi, and D. Geman. Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-6(6):707–719, November 1984.
- [5] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.
- [6] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, November 1984.
- [7] S. Geman and D. McClure. Bayesian images analysis: An application to single photon emission tomography. In *Proc. Statist. Comput. sect. Amer. Stat. Assoc.*, pages 12–18, Washington, DC, 1985.
- [8] B. Gidas. A renormalization group approach to image processing problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(2):164–180, February 1989.
- [9] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51(2):271–278, 1989.
- [10] Z. Kato, M. Berthod, and J. Zerubia. Parallel image classification using multiscale Markov random fields. In *Proc. of IEEE Int'l Conf. on Acoust., Speech and Sig. Proc.*, volume 5, pages 137–140, Minneapolis, MN, April 27-30 1993.
- [11] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):799–813, August 1989.
- [12] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82:76–89, March 1987.
- [13] J. Woods, S. Dravida, and R. Mediavilla. Image estimation using double stochastic Gaussian random field models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(2):245–253, March 1987.

- [14] J. Zhang. The mean field theory in EM procedures for Markov random fields. *IEEE Trans. on Signal Processing*, 40(10):2570–2583, October 1992.