ECSE-626

Statistical Computer Vision

MRF Bayesian Estimation

Bayesian Estimation with Markov Random Fields

As we have seen earlier in the course, most vision problems can be thought of as inverse problems, where, given some measurements, u, we want to find some quantities, z, which gave rise to these measurements.

Now, suppose that we modeled the unknown quantities as a Markov random field.

Can we make use of this modeling assumption to help solve for these quantities?

Bayesian Estimation with Markov Random Fields

We can use a Bayesian estimation process, which involves acting on the *posterior* distribution:

$$p(z|u) = p(u|z)\, p(z) \,/\, p(u).$$

Bayesian Estimation with Markov Random Fields

For example, suppose that we know that:

- The underlying quantity that we are trying to find is actually an MRF with known parameter values.
- Our measurements are the field values corrupted with Gaussian noise with known mean and covariance.

Bayesian Estimation with Markov Random Fields

In this case, we have all the information needed to specify the posterior distribution using Bayes rule.

In particular, the *prior* is a Gibbs distribution (since we know from the Hammersley-Clifford Theorem that all MRFs have a Gibbs distribution as their joint distribution).

We could then use a technique such as MAP estimation to find a solution for z given the measurements u.

Bayesian Estimation with Markov Random Fields

We can often model the measurement process statistically as:

$$P(u \mid f) \propto \exp(-\sum_{j \in R} V_j^L(f_{d_j}, u_j))$$

where R is the set of measurement sites and $\{d_j, j \in R\}$ is a set of small subsets of the sites in S.

If we take the quantity f to be an MRF, then the prior is a Gibbs distribution:

$$P(f) \propto \exp(-\sum_{c \in C} V_c^P(f_c))$$

Then the resulting posterior is a Gibbs distribution parametrized by the measurement, u :

Bayesian Estimation with Markov Random Fields

$$P(u \mid f) \propto \exp(-\sum_{j \in R} V_j^L(f_{d_j}, u_j))$$

$$P(f) \propto \exp(-\sum_{c \in C} V_c^P(f_c))$$

Then the resulting posterior is a Gibbs distribution parametrized by the measurement, u :

$$P(f \mid u) = \frac{1}{Z(u)} \exp\{-\sum_{j \in R} V_j^L(f_{d_j}, u_j) - \sum_{c \in C} V_c^P(f_c)\} = \frac{\exp(-U(f,u))}{\mathrm{Z}}.$$

Bayesian Estimation with Markov Random Fields

Finding the f that maximizes the posterior probability requires a search over all possible configurations, and hence is intractable.

Let us modify our problem slightly and take as our Gibbs distribution the following:

$$P(f \mid u) = \frac{\exp(-\frac{1}{T}U(f,u))}{Z(u,T)}$$

Bayesian Estimation with Markov Random Fields

$$P(f \mid u) = \frac{\exp(-\frac{1}{T}U(f,u))}{Z(u,T)}$$

It is seen for a given fixed value of T this distribution has the same mode as the one we considered initially. Thus, for the purposes of MAP estimation, the above distribution will suffice.

Simulated Annealing

The Geman and Geman paper referred to earlier in the context of the Gibbs sampler provides an approach to MAP estimation using a so-called "annealing" process.

This process finds an equlibrium value for f using the Gibbs sampler, at a given value of the parameter T.

Simulated Annealing

This "temperature" parameter is then reduced slowly towards zero, maintaining a quasi-equilibrium throughout.

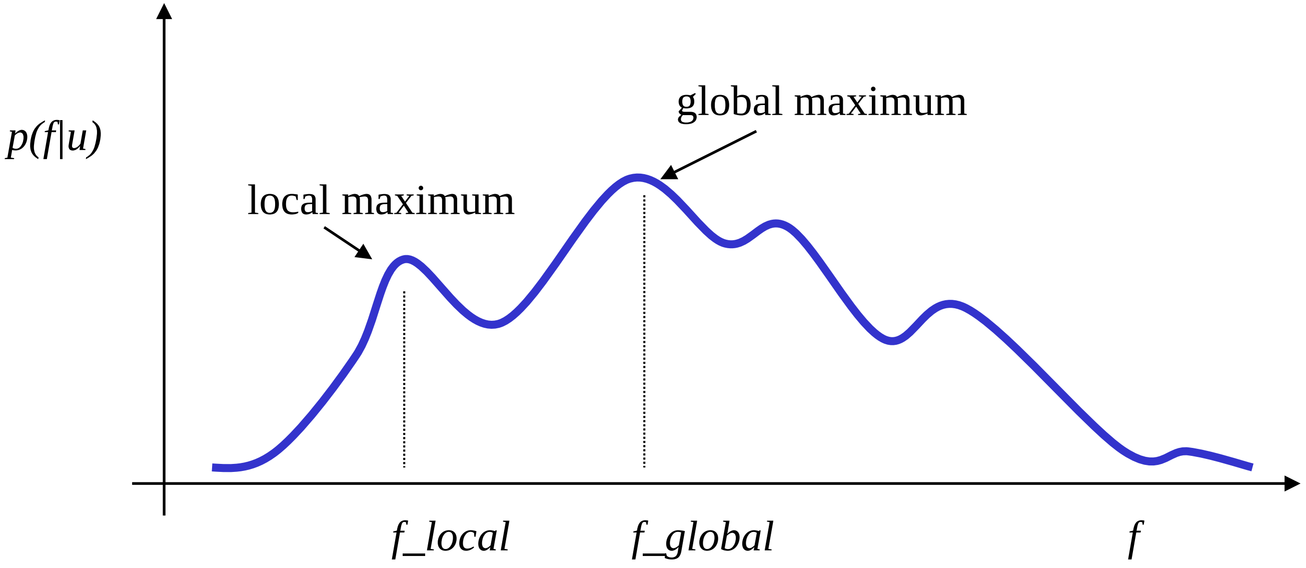
As T approaches zero the exponential form of the Gibbs distribution tends to make the field values with the global minimum of energy more and more likely.

Simulated Annealing

Thus the samples provided by the Gibbs sampler will approach those configurations of minimum energy. These then correspond to the MAP estimate.

Simulated Annealing

Simulated annealing is a means to avoid getting trapped in local maxima when trying to find the maximum of the posterior distribution $p(f|u)$.



Simulated Annealing

Simulated annealing is an iterative process, which at each iteration provides a sample from the Markov Random Field corresponding to the current temperature parameter value.

Using the Gibb's sampler, we can obtain statistical equilibrium for a given temperature T, and therefore obtain a sample of the MRF for that temperature.
This sample can be taken as an estimate of the field that is most probable.

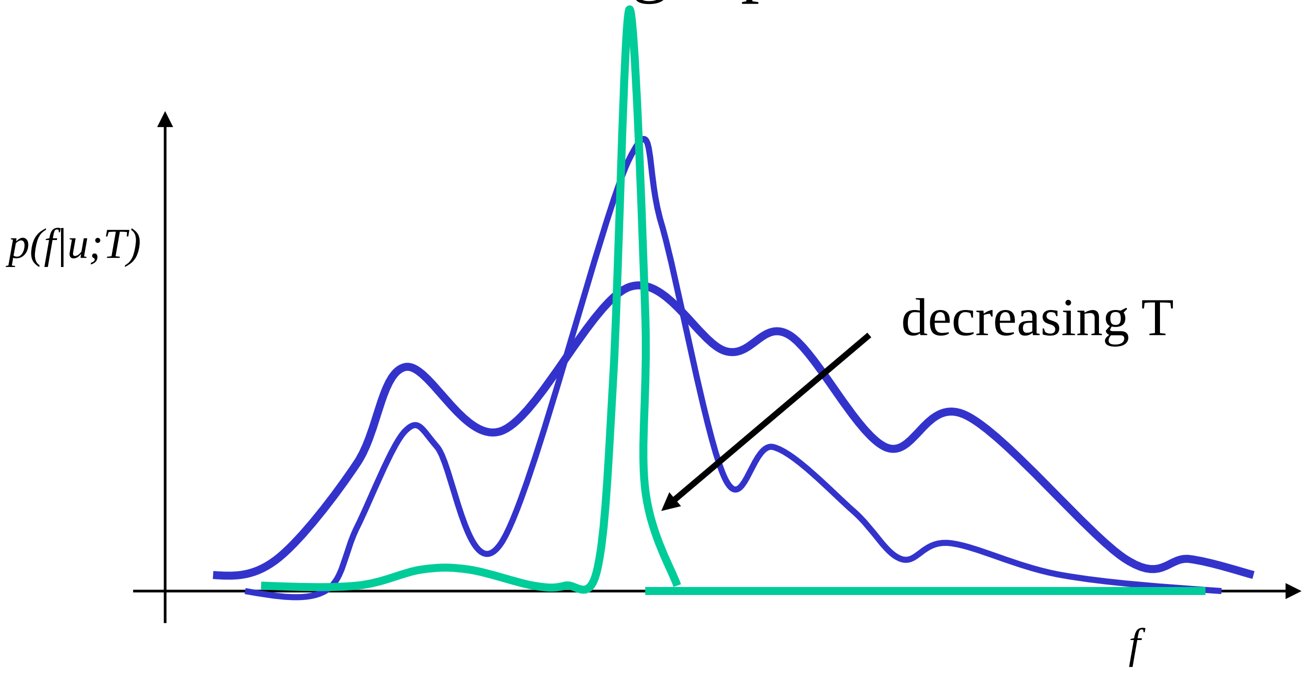
Simulated Annealing

As the iterations proceed, the temperature is reduced.

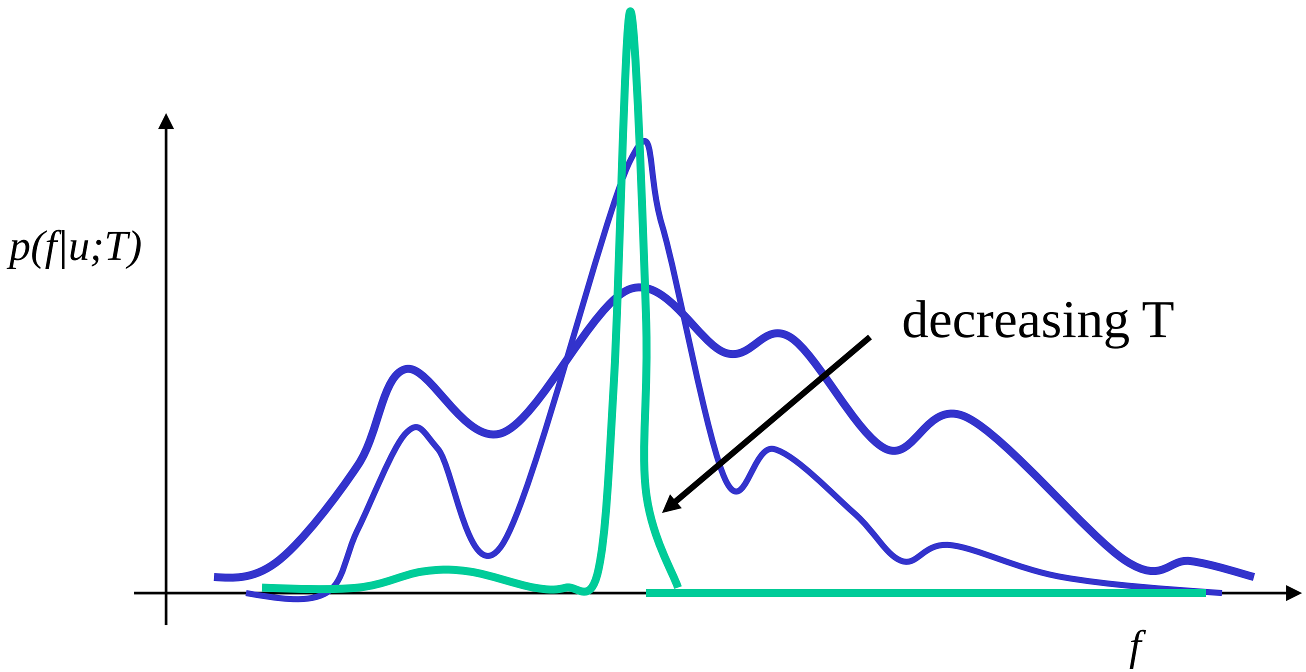
Every time the temperature drops, we are no longer in statistical equilibrium, and we need to re-run the Gibbs sampler to regain the equilibrium.

Simulated Annealing

As the temperature parameter drops, the shape of the posterior distribution becomes more and more focused on a single peak.



Simulated Annealing



If the temperature decrease is done slowly enough, so that a statistical quasi-equilibrium is maintained, the distribution will become peaked at the location of the global maximum of *p(f|u)*.

Annealing Schedules

How should we lower the temperature parameter in the simulated annealing process in order to guarantee convergence to the global maximum?

Can we do this quickly? Or does the decrease have to proceed very slowly?

Proof of Annealing Theorem (for your interest…)

Simulated Annealing

The Geman and Geman paper proves the following theorem:

Theorem B (Annealing):

Assume that there exists an integer $\tau \geq N$ such that for every $t = 0, 1, 2, \ldots$ we have $S \subseteq \{n_{t+1}, n_{t+2}, \ldots, n_{t+\tau}\}$.

$\{n_{t+1}, n_{t+2}, \ldots, n_{t+\tau}\}$ is the sequence of sites that are visited.

N is the total number of sites.

Simulated Annealing

Let $T(t)$ be any decreasing sequence of temperatures for which:

a) $T(t) \rightarrow 0$ as $t \rightarrow \infty$

b) $T(t) \geq N\Delta / \log t$

for all $t \geq t_0$ for some integer $t_0 \geq 2$.

Simulated Annealing

Then, for any starting configuration f^{0} and for every $f \in F$,

$$\lim_{t\to\infty} P(F(t) = f \mid F(0) = f^{0}) = \pi_0(f).$$

In the above, $\pi_0(f)$ is the uniform distribution on F_0 where

$$F_0 = \{ f \in F : U(f) = \min_{f^{0}} U(f^{0}) \}$$

Simulated Annealing

This theorem basically states that if we lower our temperature slowly enough, then after a very long time, the samples provided by the Gibbs sampler will approach those configurations that have the globally minimum energy (i.e. will be the global MAP estimate).

Simulated Annealing

The annealing schedule provided by the theorem is impractically slow, unfortunately!
It is merely an upper-bound, however, and good results might be obtained with a more rapid schedule.

Geman and Geman found that using the schedule $T(k) = C / \log(1+k)$ where k is the current number of iterations (complete sweeps of the Gibbs sampler over the set of sites) with C = 3 and C = 4, they acheived acceptable results.

Applications of Simulated Annealing

Simulated annealing is a technique for finding the ***mode*** of a distribution that is modeled by a Gibbs distribution.

We can therefore apply this to MAP estimation problems where we want to find the mode of a posterior distribution $p(f|u)$.

Applications of Simulated Annealing

Bayes rule lets us write $p(f|u) = p(u|f)p(f)/Z$

If we can specify Gibbs distributions for the likelihood $p(u|f)$ and for the prior $p(f)$ then we have the Gibbs distribution for the posterior, to which we can apply simulated annealing.

Example: Noise Reduction or Image Restoration

Let us look at an example, where we have noisy measurements of an Ising texture field.
Assume that the noise is additive, zero-mean and white with a Gaussian distribution.

The measurement equation is therefore:

$$u = f + \eta$$

where $\eta \approx N(0,1/\varepsilon)$, and f is an Ising MRF with given β.

Example: Noise Reduction or Image Restoration

The likelihood is therefore a Gibbs distribution:

$$P(u \mid f) = \frac{1}{Z_L(u)} \exp(-U_L(f,u))$$

where $U_L = \sum_{i \in S} \varepsilon (f_i - u_i)^2$

The prior distribution is also a Gibbs distribution:

$$P(f) = \frac{1}{Z_P} \exp(-U_P(f,u))$$

where $U_P = \sum_{i \in S} \sum_{j \in N(i)} \beta f_i f_j$

Example: Noise Reduction or Image Restoration

Thus, the posterior Gibbs distribution can be written as:

$$P(f \mid u) = \frac{1}{Z(u)} \exp(-U(f, u))$$

where $U = \sum_{i \in S} \sum_{j \in N(i)} \beta f_i f_j + \sum_{i \in S} \varepsilon (f_i - u_i)^2$

Example: Noise Reduction or Image Restoration

Simulated annealing requires a sampling scheme, such as the Metropolis algorithm or the Gibbs sampler.

To implement the Gibbs sampler, we need to compute the conditional probability of the field value at a site given the values at sites in the neighborhood.

Example: Noise Reduction or Image Restoration

The conditional probability for f at site i can be written as:

$$P(f_i \mid f_{N_i}) = \frac{\exp(-\frac{1}{T}[\beta f_i \sum_{j \in N(i)} f_j + \varepsilon(f_i - u_i)^2])}{\exp(-\frac{1}{T}[\beta \overline{f}_i + \varepsilon(1-u_i)^2]) + \exp(-\frac{1}{T}[-\beta \overline{f}_i + \varepsilon(1+u_i)^2])}$$

where $\overline{f}_i = \sum_{j \in N(i)} f_j$.

(remember, we assume that f can take on values +/- 1)

Example: Noise Reduction or Image Restoration

Expanding out the quadratic terms in the denominator and grouping, we get:

$$P(f_i \mid f_{N_i}) = \frac{\exp(-\frac{1}{T}[\beta f_i \sum_{j \in N(i)} f_j + \varepsilon (f_i - u_i)^2])}{\exp(-\frac{\varepsilon}{T}(1+u_i)^2)\left[\exp(-\frac{1}{T}(\beta \overline{f}_i - 2\varepsilon u_i)) + \exp(+\frac{1}{T}(\beta \overline{f}_i - 2\varepsilon u_i))\right]}$$

Example: Noise Reduction or Image Restoration

or

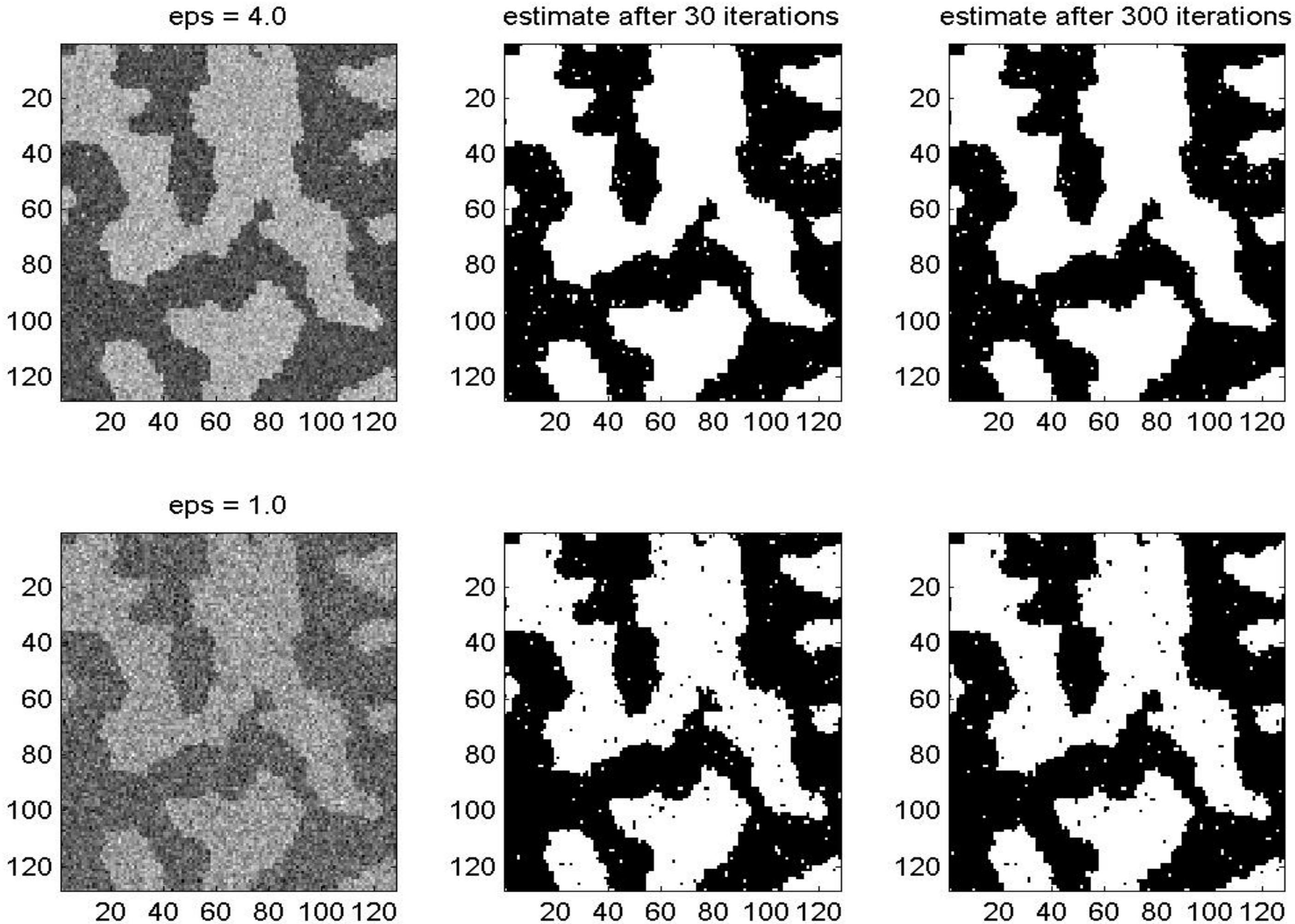
$$P(f_i \mid f_{N_i}) = \frac{\exp(-\frac{1}{T}[\beta f_i \sum_{j \in N(i)} f_j + \varepsilon (f_i - u_i)^2])}{2\exp(-\frac{\varepsilon}{T}(1+u_i)^2)\cosh(\frac{1}{T}(\beta \overline{f_i} - 2\varepsilon u_i))}$$

This is the form of the conditional probability that we will use in the implementation of the Gibbs sampler.

Matlab program for Noise Reduction of Ising Textures using Simulated Annealing

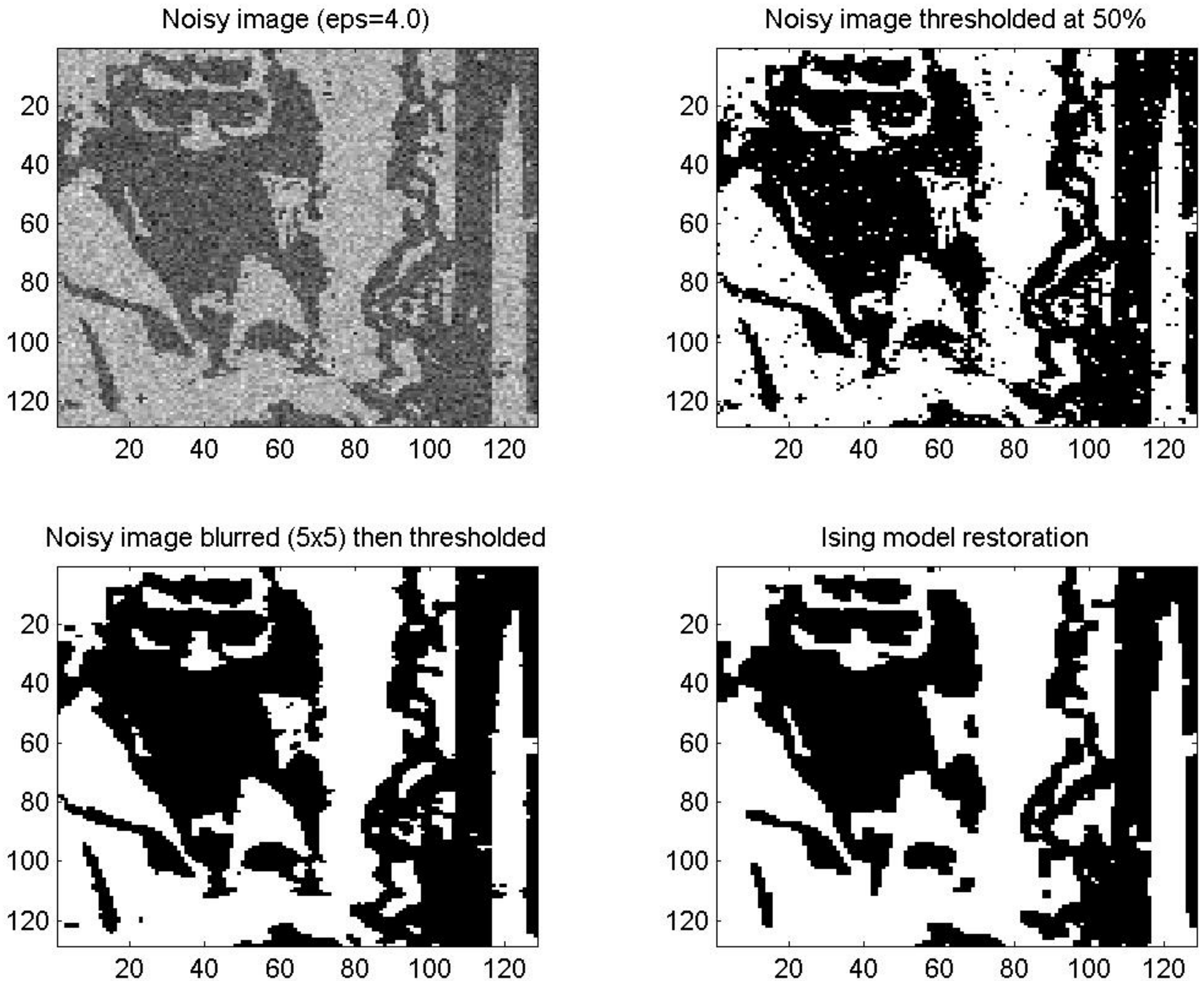
```
function im1=Ising_Restoration(im0,beta,eps)
%
% uses simulated annealing to restore a noisy Ising texture with known beta
% the statistics of the noise are assumed to be known (zero mean, white, variance 1/eps)
%
[n,m]=size(im0);
im1=2*(im0>0)-1; % take the initial field to be a thresholded version of the measurements.

for k=1:300,
    Tinv = log(1+k)/3.0;
    for row=1:n,
        for col=1:m,
            nbrs = 0;
            if(row > 1) nbrs = nbrs + im1(row-1,col); end
            if(row < n) nbrs = nbrs + im1(row+1,col); end
            if(col < m) nbrs = nbrs + im1(row,col+1); end
            if(col > 1) nbrs = nbrs + im1(row,col-1); end
            cond=exp(-Tinv*(beta*nbrs+(1-im0(row,col))*(1-im0(row,col))))/
(2.0*exp(-Tinv*eps*(1+im0(row,col)*im0(row,col)))*cosh(Tinv*(beta*nbrs-2*eps*im0(row,col))));
            im1(row,col)=-1;
            if(rand < cond)
                im1(row,col) = 1;
            end
        end
    end
end
```



β=0.8

Restoration results using an annealing schedule of T=3/ln(1+k).



Noise reduction of a natural image (binarized and then noise added)

Example: Distorted Image Restoration

As noted in the Geman and Geman paper, we can easily handle the case of nonlinear measurement functions, as well as noise models other than additive (e.g. multiplicative).

Let's look at an example, where $u = \eta\sqrt{H(f)}$ where η is white, Gaussian noise, with known mean and variance, and $H(f)$ is a known linear blurring process.

Example: Distorted Image Restoration

In other words, the measurement is obtained from the MRF by:

(1) First, blurring the random field.

(2) Passing the field through a point-wise square-root nonlinearity,

(3) Perturbing the result by a multiplicative Gaussian noise field.

Example: Distorted Image Restoration

The likelihood is therefore a Gibbs distribution:

$$p(u \mid f) = \frac{1}{Z_L(u)} \exp\left(-\frac{1}{2\sigma^2} \sum_{i \in S} \left(\mu - \frac{u_i}{\sqrt{H(f_i)}} \right)^2 \right)$$

(essentially, this is saying that $\frac{u_i}{\sqrt{H(f_i)}}$ is Gaussian distributed)

Example: Distorted Image Restoration

Let us consider the case where f is known to be a Multi-Level Logistic MRF, with given α,β values.

Then the prior model will be:

$$p(f) = \frac{1}{Z_P} \sum_{i \in S} \exp\left(-[\beta n(f_i) + \alpha f_i]\right)$$

Example: Distorted Image Restoration

Let us also assume that the mean of the noise field is 1.

The conditional probability for f at site i can then be written as:

$$P(f_i \mid f_{N_i}) = \frac{\exp\left(-\frac{1}{T}[\beta n(f_i) + \alpha f_i + (1 - u_i / \sqrt{H(f_i)})^2]\right)}{Z(u_i)}$$

where $n(f_i)$ is the number of sites in the neighborhood of site i that have the same label as site i.

Example: Distorted Image Restoration

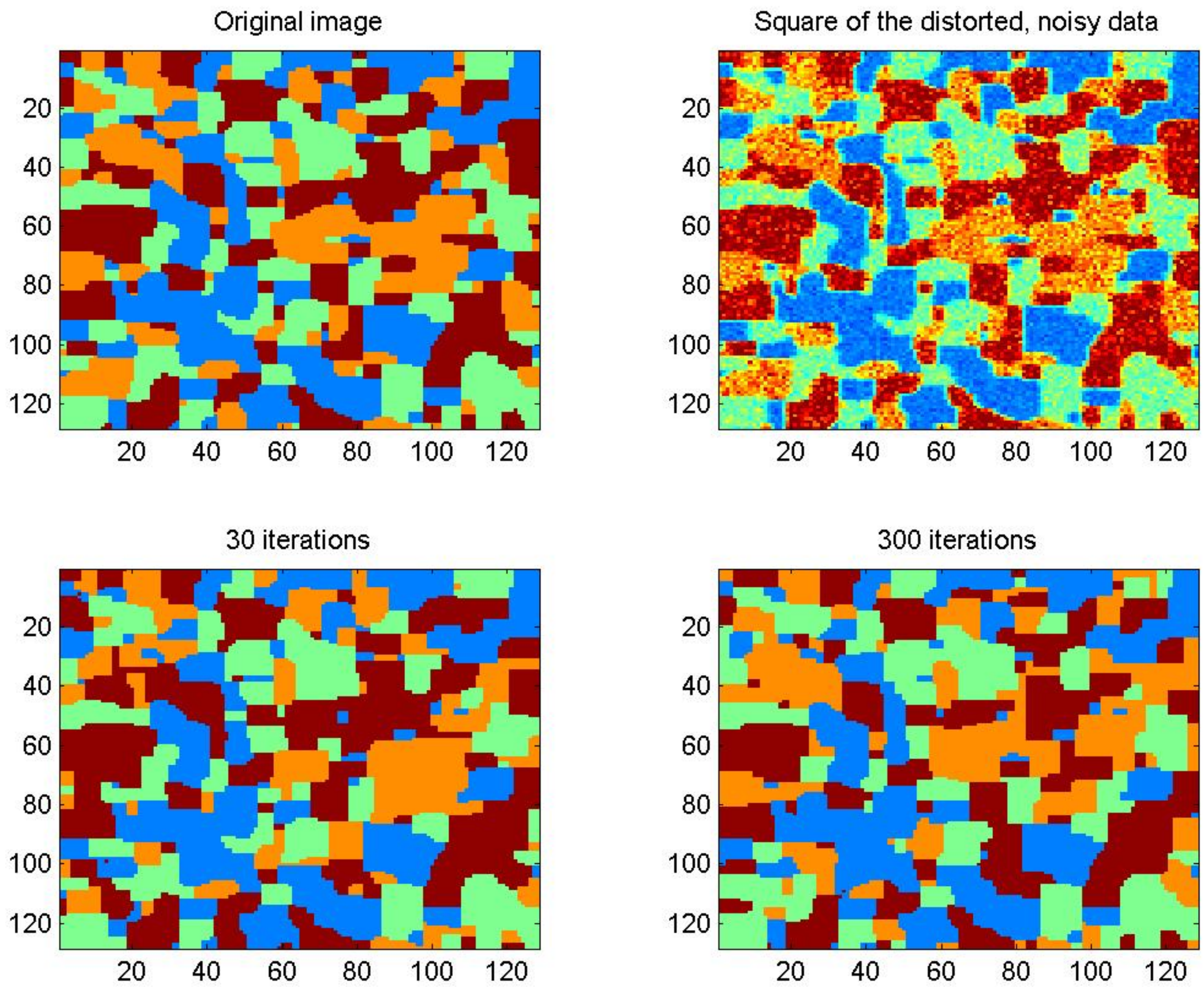
$H(f_i)$ is computed by applying the blurring convolution kernel to just the site i.

The partition function $Z(u)$ is computed by summing the numerator over all possible values of f_i.

0	1/6	0
1/6	1/3	1/6
0	1/6	0

Blurring process convolution kernel

```
function f=MLL_Restoration(im0,beta,alpha,nlevels,variance,niters)
%
[n,m]=size(im0); s22 = 1/(2.0*variance);
f=im0.*im0; % take the initial field to be the square of the measurements.
for k=1:niters,
    Tinv = log(1+k)/3.0;
    for row=randperm(n),
        for col=randperm(m),
            H = 0.0; nn = 0;
            if(row > 1) H = H + f(row-1,col); nn=nn+1; end
            if(row < n) H = H + f(row+1,col); nn=nn+1; end
            if(col > 1) H = H + f(row,col-1); nn=nn+1; end
            if(col < m) H = H + f(row,col+1); nn=nn+1; end
            H = H*0.5; % coefficients of the blurring kernel.
            nn = 0.5*nn+1;
            for fi=1:nlevels, % compute the cond prob for each possible level
                nbrs = 0;
                if(row > 1) nbrs = nbrs + (f(row-1,col)==fi); end
                if(row < n) nbrs = nbrs + (f(row+1,col)==fi); end
                if(col > 1) nbrs = nbrs + (f(row,col-1)==fi); end
                if(col < m) nbrs = nbrs + (f(row,col+1)==fi); end
                cond(fi) = exp(-Tinv*(alpha*fi+beta*nbrs+s22*(1-im0(row,col)/sqrt((H+fi)/nn))^2));
            end
            Z = sum(cond); % compute the partition function
            cumulative(1)=cond(1)/Z;
            for IK=2:nlevels,
                cumulative(IK)=cumulative(IK-1)+(cond(IK)/Z);
            end
            dice_roll=rand; % generate a uniform variate in [0,1]
            for IM=1:nlevels,
                if(dice_roll < cumulative(IM))
                    f(row,col)=IM;
                    break;
                end
            end
        end
    end
end
```



Restoration results for a blurred, distorted, noisy (multiplicative) MLL MRF texture

Marginal Posterior Mode (MPM) Estimation

Instead of looking for the estimate which globally maximizes the posterior distribution (MAP) estimate, we can try to find at each site in the field, the value that is most probable given all of the data, that is:

$$\hat{f}_i = \arg\max_{f_i \in L} P(f_i \mid u),$$ where L is the space of possible field values.

Marginal Posterior Mode (MPM) Estimation

The conditional probability $P(f_i \mid u)$ is a "marginal" obtained from the full posterior $P(f \mid u)$:

$$P(f_i \mid u) = \sum_{f_1} \sum_{f_2} \cdots \sum_{f_{i-1}} \sum_{f_{i+1}} \cdots \sum_{f_N} P(f_0, f_1, \ldots, f_i, \ldots, f_N \mid u)$$

The estimate $\{\hat{f}_i\}$ so obtained is known as the "Marginal Posterior Mode".

The difficulty with this approach lies in the amount of computation required to compute the modes.

Marginal Posterior Mode (MPM) Estimation

We can't directly use Simulated Annealing to find statistics of the distribution other than the mode, such as the Marginal Posterior Mode.

The MRF sampling techniques that we described earlier can be used to solve the MPM problems.

Marginal Posterior Mode (MPM) Estimation

The idea is to use sampling (such as the Gibbs or Metropolis samplers) to generate a set of many samples from the Gibbs distribution of interest (i.e. that of the posterior distribution).

We can then used these samples in a Monte-Carlo like procedure to compute the required statistic.

Marginal Posterior Mode (MPM) Estimation

Let us have M samples obtained from a sampling process. We can estimate the marginal distributions as follows:

$$P(f_i \,|\, u) \approx \frac{\#\{m \in [1, M]: f_i^m = f_i\}}{M}$$

That is, for each site in the field, we count how many times the field at that site takes on a given value (f_i) and divide the count by the total number of samples.

Marginal Posterior Mode (MPM) Estimation

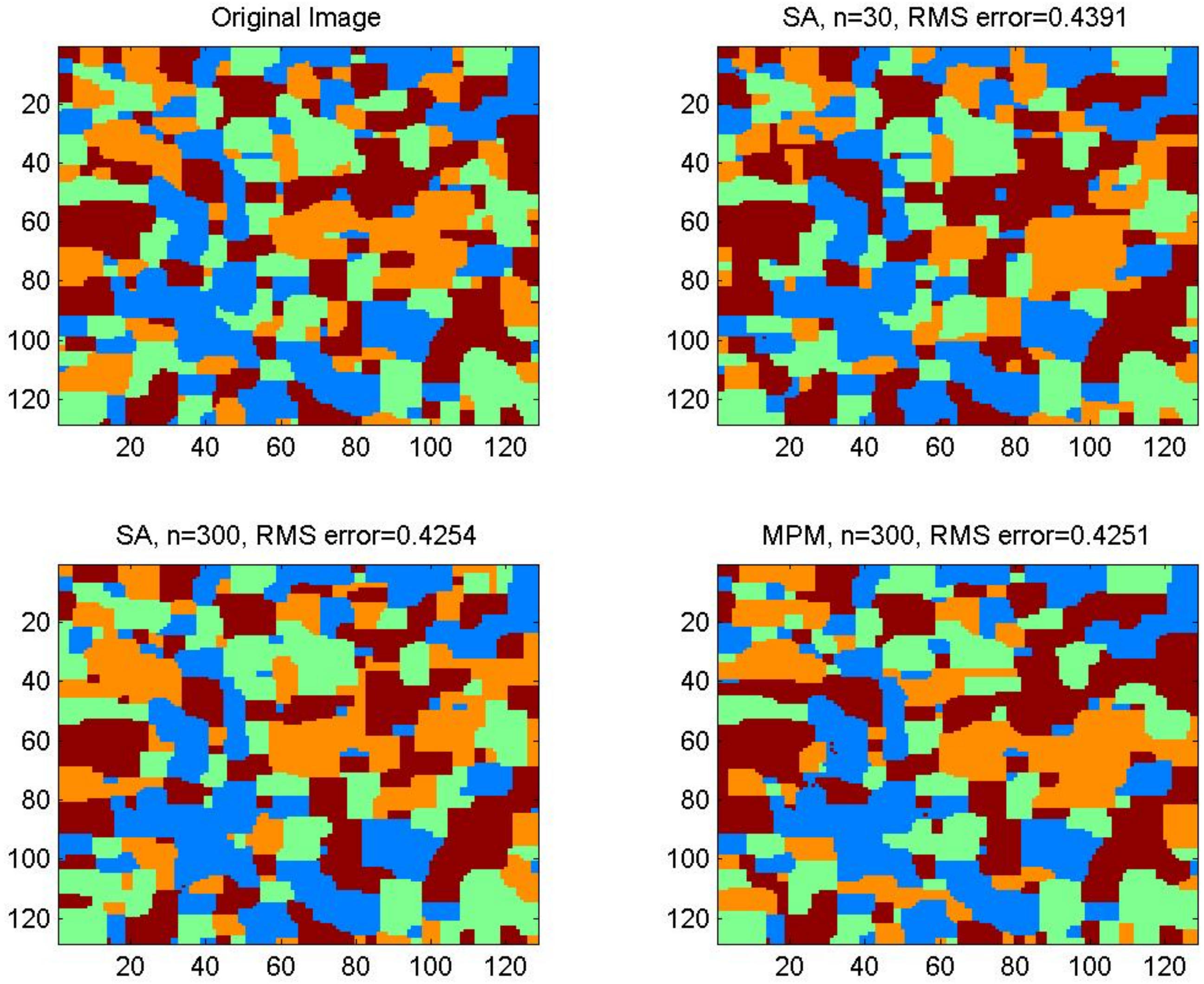
This gives us a "histogram" of field values, as it were.

The MPM solution is then obtained simply by finding the mode of this histogram (i.e. the field value that shows up the most times).

Marginal Posterior Mode (MPM) Estimation

The main **advantage** of the MPM approach over the MAP with simulated annealing approach is that a convergent annealing schedule does not have to be determined, or for that matter, used.
Thus convergence to a given accuracy may be faster.

The main **disadvantage** of the MPM approach is that it can use a ***large amount of memory*** if the label set is large, since we have to store the label histograms (posterior marginals) for each pixel.


Original Image
SA, n=30, RMS error=0.4391
SA, n=300, RMS error=0.4254
MPM, n=300, RMS error=0.4251
20 40 60 80 100 120
20 40 60 80 100 120

MCMC

Techniques, such as the ones we have just described, which use a series of samples of a MRF to compute some expectation are often referred to as ***Markov-Chain-Monte-Carlo*** (**MCMC**) methods.

ICM

A technique related to MPM estimation is the ***Iterated Conditional Modes*** (**ICM**) estimation technique.

In ICM, instead of doing a stochastic search to find the Mode of the Marginal Conditional, a deterministic, iterative, search for the configuration that minimizes the local energy is done.

ICM

In the point-wise measurement case (i.e. where the measurement sites are the same as the MRF sites) the ICM estimates at a given time step *m* are:

$$f_i^{m+1} = \arg\min_{f_i} \sum_{j \in N(i)} V_c(f_i, f_{N_i}^m) + V_i(f_i, u)$$

Gradient descent can be used to find this value at each time step.

Sometimes the minimization can be done analytically.