ECSE-626
Statistical Computer Vision

Markov Random Fields Lecture #2

Examples of MRFs: Auto-Logistic (Ising) Model

Contextual constraints on two labels are the lowest order constraints capable of conveying contextual relationships. They are widely used because of the simple form and low computational cost.

They are encoded in the Gibbs energy as pair site clique potentials. With clique potentials of up to two sites, we have:

$$U(f) = \sum_{i \in S} V_1(f_i) + \sum_{i \in S} \sum_{i' \in N_i} V_2(f_i, f_{i'})$$

Examples of MRFs: Auto-Logistic (Ising) Model

When $V_1(f_i) = f_i G_i(f_i)$ and $V_2(f_i, f_{i'}) = \beta_{i,i'} f_i f_{i'}$, where $G_i(\cdot)$ are arbitrary functions, and $\beta_{i,i'}$ are constants, then the energy is:

$$U(f) = \sum_{\{i\} \in c_1} f_i G_i(f_i) + \sum_{\{i,i'\} \in c_2} \beta_{i,i'} f_i f_{i'}$$

The above model for the energy is called an auto-model.

Examples of MRFs: Auto-Logistic (Ising) Model

An auto-model is said to be an auto-logistic model if the f_i's take on values in the discrete label set $L = \{0,1\}$ or $L = \{-1,+1\}$.

In this case, the energy is:

$$U(f) = \sum_{\{i\} \in c_1} \alpha_i f_i + \sum_{\{i,i'\} \in c_2} \beta_{i,i'} f_i f_{i'}$$

Examples of MRFs: Auto-Logistic (Ising) Model

When N is the nearest neighborhood system on a lattice (e.g. 4-nearest neighbours in 2D or 2-nearest neighbours in 1D), the auto-logistic model reduces to the Ising model.

This model has application in condensed matter physics, where it was first used, in 1925, to model formation of magnetic field domains in ferromagnetic materials.

Examples of MRFs: Auto-Logistic (Ising) Model

If we take all of the $\beta_{i,i'}$'s to be the same, then the conditional distribution at site i for the {-1, +1} Ising model is straightforwardly computed to be:

$$P(f_i \mid f_{N_i}) = \frac{\exp(\beta f_i \sum_{j \in N_i} f_j)}{2\cosh(\beta \sum_{j \in N_i} f_j)}$$

A positive value for β statistically favors the identity of field values at neighboring sites, while a negative value favors the field values to take opposite values.

Examples of MRFs: Auto-Logistic (Ising) Model

The Ising model is useful in vision problems where binary variables (such as edges) are to be found.

Examples of MRFs: Multi-Level Logistic Model

The auto-logistic model can be generalized to the multi-level logistic model (MLL), by increasing the number of possible labels to be more than 2.

This model is often used in performing image segmentation, where the number of labels is equal to the number of distinct regions to be found.

Examples of MRFs: Multi-Level Logistic Model

In the MLL model, the clique potentials are defined as follows:

$$V_c(f) = \begin{cases} L_c & \text{if all sites on c have the same label} \\ -L_c & \text{otherwise} \end{cases}$$

L_c is a potential value for cliques of type-c.

Examples of MRFs: Multi-Level Logistic Model

For second-order models, the clique potentials are non-zero only for single-site and pair-site cliques.

For the 4-neighborhood system there are four types of pair-site cliques, and so there are 4 associated parameters (clique potentials).

Examples of MRFs: Multi-Level Logistic Model

When the model is isotropic, all 4 of the pair-site clique potentials are taken to be the same.

Such isotropic models describe 'blob-like' regions, and have the following conditional probability:

$$P(f_i = I \mid f_{N_i}) = \frac{\exp(-\alpha_I - \beta n_i(I))}{\sum_{I=1}^{M} \exp(-\alpha_I - \beta n_i(I))}$$

(NOTE: I in the denominator not same as I in the numerator – I is an index in the denominator)

where $n_i(I)$ is the number of sites in N_i which have the label I and α_I is the (single-site clique) potential for label I.

Examples of MRFs: Auto-Normal (Gauss-Markov) Model

An auto-model is said to be an auto-normal model, or Gauss-Markov Random Field, if the label set is the real line and the joint distribution is a multivariate Gaussian:

$$p(f) = \frac{\sqrt{\det(B)}}{\sqrt{(2\pi\sigma^2)^m}} \exp(-\frac{(f-\mu)^T B(f-\mu)}{2\sigma^2})$$

Examples of MRFs: Auto-Normal (Gauss-Markov) Model

The conditional density is:

$$p(f_i \mid f_{N_i}) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp(-\frac{1}{2\sigma^2}[(f_i - \mu_i)^2 - (f_i - \mu_i)\sum_{i' \in N_i} \beta_{i,i'}(f_{i'} - \mu_{i'})])$$

B is a very sparse "interaction matrix", whose diagonal elements are all equal to 1 and whose off diagonal elements (i, i') are equal to $-\beta_{i,i'}$

Examples of MRFs: Auto-Normal (Gauss-Markov) Model

Recall that the conditional density for the second-order case is given by

$$P(f_i \mid f_{N_i}) = \frac{\exp(-[V_1(f_i) + \sum_{i' \in N_i} V_2(f_i, f_{i'})])}{\sum_{f_i \in L} \exp(-[V_1(f_i) + \sum_{i' \in N_i} V_2(f_i, f_{i'})])}$$

From the formula for the conditional density on the previous slide we can identify the clique potentials:

$$V_1(f_i) = (f_i - \mu_i)^2 / 2\sigma^2$$

$$V_2(f_i, f_{i'}) = \beta_{i,i'}(f_i - \mu_i)(f_{i'} - \mu_{i'}) / 2\sigma^2$$

Sampling Markov Random Fields

In order to compute expectations (e.g. to determine a MAP estimate) we must either know the posterior distribution, or some approximation to it.

An approximation can be made by representing the distribution by a sum of impulse functions, corresponding to samples taken from the distribution.

Sampling Markov Random Fields

Samples from the distribution can be useful in generating instantiations of the field, for example for texture synthesis.

Sampling Markov Random Fields

- The **Monte-Carlo method** of sampling takes samples uniformly over the space of possible configurations to get statistics (e.g. Expectations).
- Assumes samples i.i.d.
- Approximates expectations with a sample average of samples.

- However, these samples will typically be of configurations where the probability is very low.

Sampling Markov Random Fields

- Picking a good sampling distribution becomes hard in high dimensions
- Major contributions can be hidden in small areas
- Danger of missing those. Need to search for high values of density function
- Thus, other sampling schemes need to be used…

Markov Chain Monte Carlo Sampling

- An alternative is to generate **dependent samples**
- We sample from another, known, easy-to-sample, ***proposal*** distribution.
- (But, how would this work? Isn't this a bit like looking for your lost keys under the streetlamp because the light is better?)
- We maintain record of current state, and following proposal distribution depends on current proposal, so sample sequence forms Markov chain

The stationary distribution should be the target density

MCMC Sampling

- MCMC methods involve a Markov process in which a sequence of states is generated: Each sample has a probability distribution that depends on the previous value.
- Since successive samples are *dependent*, the Markov chain may have to run for a considerable time in order to generate samples that are effectively **independent samples** of the probability density function of interest.

MCMC Sampling

Let’s look at two (older) MCMC methods - the *Metropolis algorithm* and the *Gibbs Sampler*.

Sampling Markov Random Fields

The Metropolis and Gibbs samplers both work by producing a sequence of samples, $\{f^0, f^1, \ldots, f^k\}$, taken from a sequence of distributions, whose limiting, or equilibrium, distribution is the Gibbs distribution of interest.

After equilibrium is reached the subsequent samples in the sequence can then be used, for example, to compute expectations.

Sampling Markov Random Fields

For example, the mean of a function of the field can be approximated by:

$$< g(F) >= \frac{1}{R}\sum_{r=1}^{R} g(F^{r})$$

where the F^{r} are realizations of the MRF once equilibrium has been reached.

The Metropolis Sampler

The Metropolis sampler is also known as 'Stochastic Relaxation'.

The algorithm starts by randomly selecting some initial sample value f^0.
This starting point could be of a very unlikely configuration.

We then proceed with the following iteration:

The Metropolis Sampler

- **Basic strategy**:
  - Start from arbitrary f^0
  - Tweak f^0 a little to get w
  - If $P(w) \geq P(f^0)$ then move to w, $f^1 = w$
  - If $P(w) << P(f^0)$ then stay at f^0
  - In intermediate cases, randomize

How do we tweak f^0 a little to get w?

Encoded in one-step proposal distribution $q(w|f^0)$

The Metropolis Sampler

1. Given f^{k} generate a new sample, w, through some transition process with a probability $q(w \mid f^{k})$. This transition process must be symmetric.

2. Compute $\Delta E(w) = U(w) - U(f^{k})$.

3. If $\Delta E(w) < 0$ then we 'accept' the new sample and set $f^{k+1} = w$, we increment $k \to k+1$, and return to step 1.

The Metropolis Sampler

If $\Delta E(w) \geq 0$ then generate a random number $\mathrm{R} \in [0,1]$.

If $\mathrm{R} \leq \exp(-\Delta E(w))$, we 'accept' the new sample with a probability $\exp(-\Delta E(w))$ and increment $k \rightarrow k+1$.

Otherwise, we 'reject' the new sample, and return to step 1 without incrementing the iteration counter.

The Metropolis Sampler

The symmetric transition process results in the sequence of samples being a reversible Markov chain (in time).

For example, we could have $q(w|f^k) = N(0,\sigma)$,
i.e. a Gaussian random walk, and where σ could change over time (become smaller) in a process known as 'annealing'.

The Metropolis Sampler

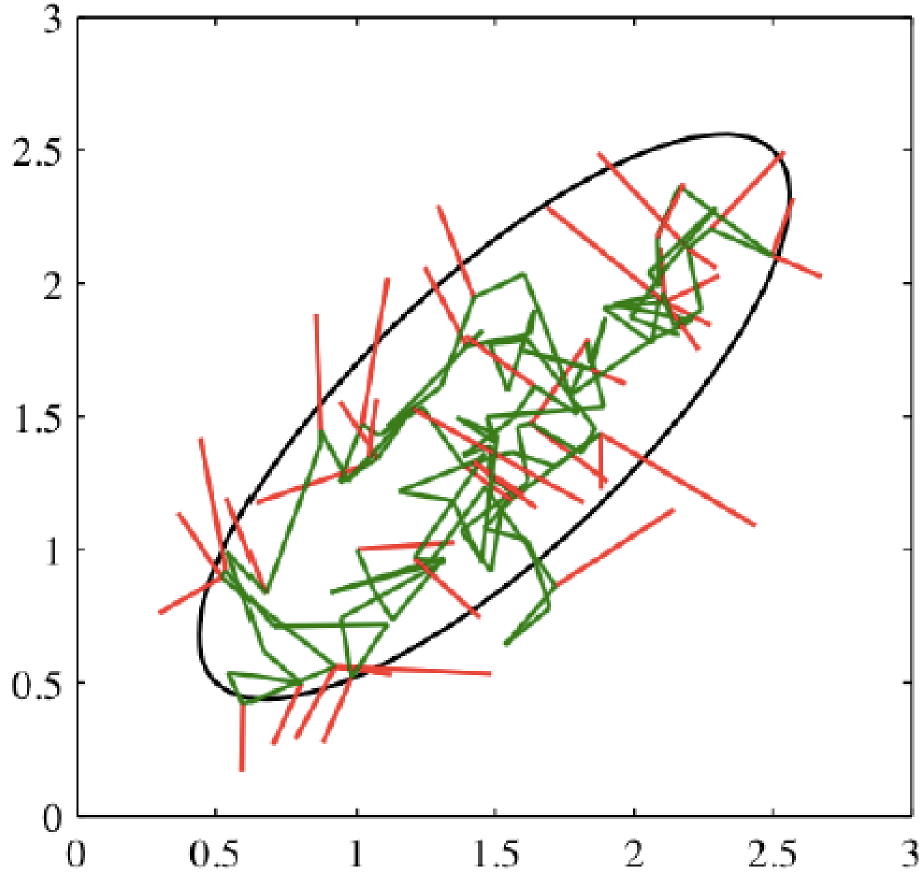
It can be shown that in the limit as $k \to \infty$,

$$\lim_{k\to\infty} P(f^k = f) = \frac{1}{Z}\exp[-U(f)]$$

In other words, the distribution of f^k approaches that of a Gibbs distribution.

The Metropolis algorithm can be slow if there are many rejections, and is constrained to use a symmetric transition function $q(w \mid f^k)$.

Example: approximate distribution whose one std deviation contour given by an ellipse, $Q(x'|x)$, $\mathcal{N}(x,\sigma^2)$



http://www.cs.toronto.edu/~jessebett/CSC412/content/week8/07-sampling.pdf

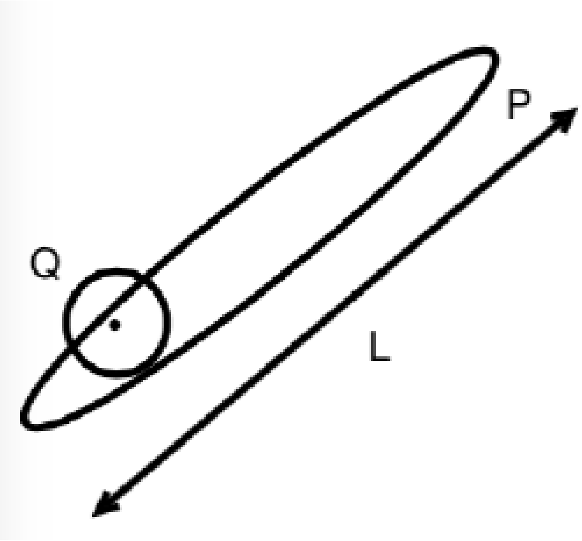
Proposal distribution has significant effect on performance

Common proposal is Gaussian centered on current state

$$Q(x'|x) = N(x,\sigma^2)$$

- Small σ : slow random walk with long correlation times

  ~ $(L/\sigma)^2$ iterations
- Large σ : many rejections



http://www.cs.toronto.edu/~jessebett/CSC412/content/week8/07-sampling.pdf

The Gibbs Sampler

(Geman, S. and Geman, D., "Stochastic relaxation, Gibbs distributions, and Bayesian restoration", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 6, November, 1984, pp 721-741)

Like the Metropolis algorithm, the Gibbs sampler produces a Markov chain of sample values, whose equilibrium distribution is the Gibbs distribution of interest.

The transition probabilities, however, are non-stationary, and the chain is not reversible.

This complicates the proof (and detection) of convergence.

The Gibbs Sampler

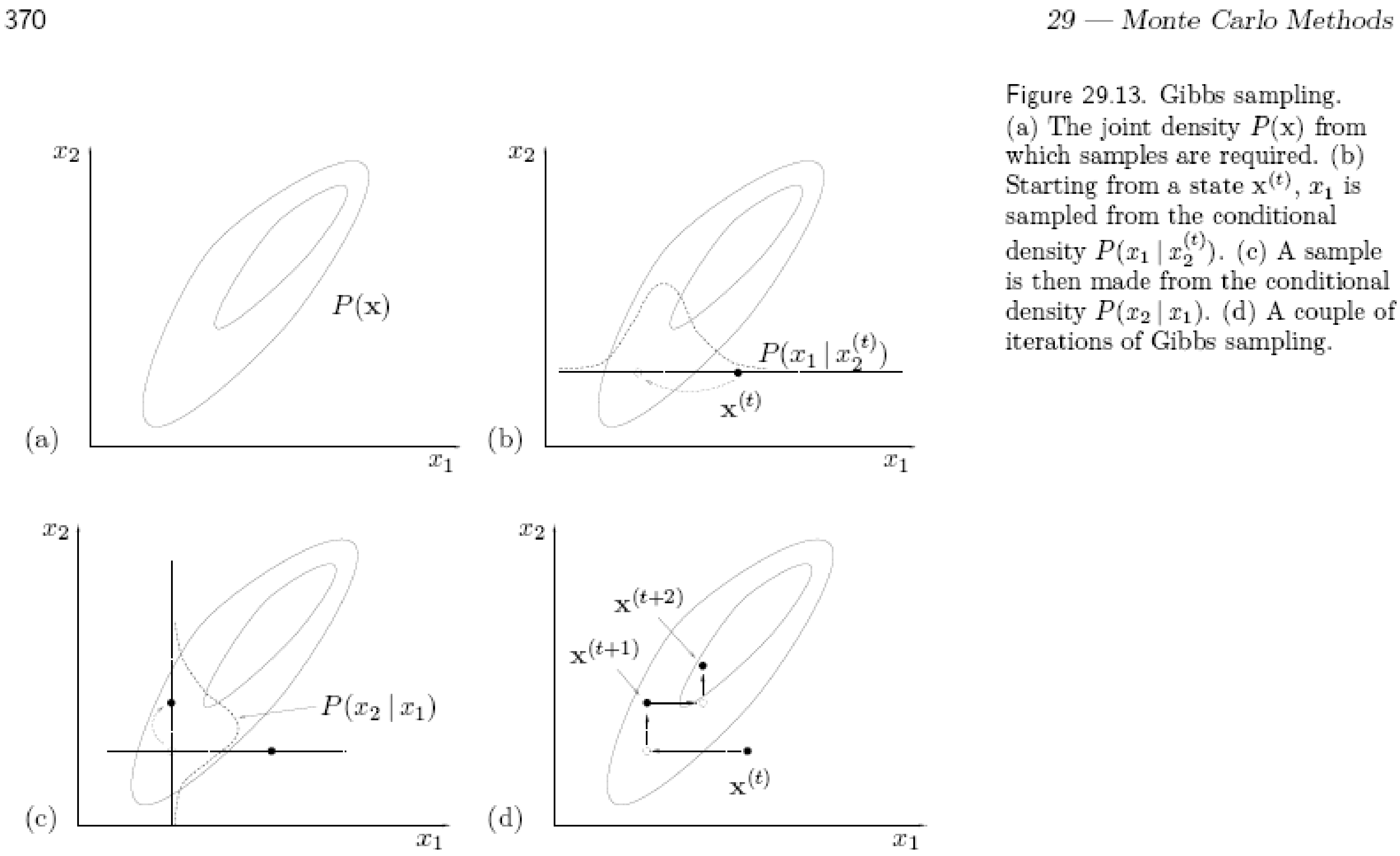
The Gibbs sampler is initialized just as in the Metropolis algorithm to some random field value, f^0.

The iteration in the Gibbs sampler involves updating the field one site at a time, according to the current conditional probabilities.

The Gibbs Sampler

The sites in S are visited according to some schedule (e.g. randomly, or perhaps via a 'raster-scan') and the value of the field at the site being visited is replaced by a sample from the conditional distribution.

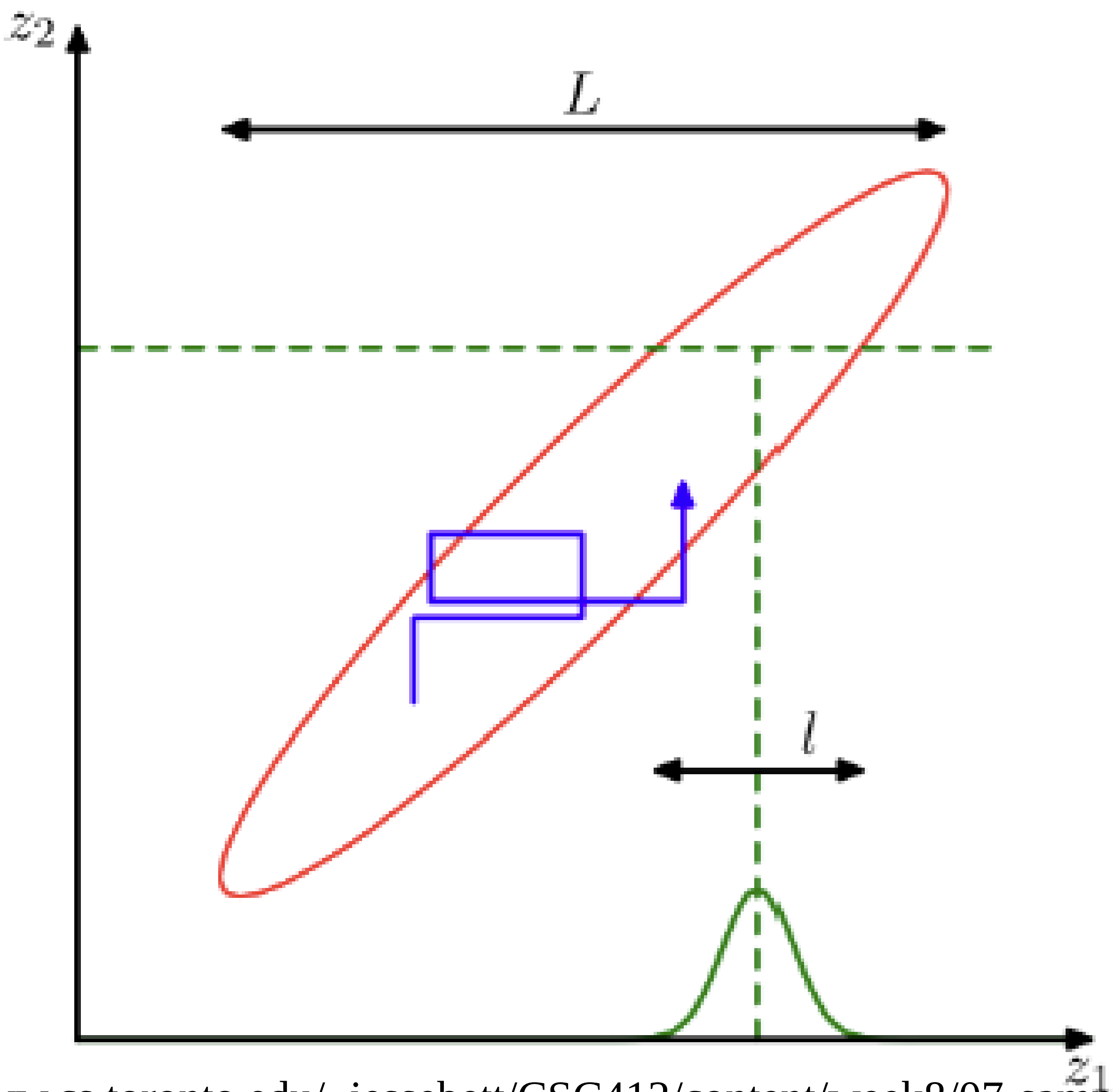
Thus, $f_i^k \rightarrow f_i^{k+1}$ with probability $P(f_i^{k+1} \mid f_{N_i})$.



370 *29 — Monte Carlo Methods*

Figure 29.13. Gibbs sampling. (a) The joint density $P(\mathbf{x})$ from which samples are required. (b) Starting from a state $\mathbf{x}^{(t)}$, x_1 is sampled from the conditional density $P(x_1 \mid x_2^{(t)})$. (c) A sample is then made from the conditional density $P(x_2 \mid x_1)$. (d) A couple of iterations of Gibbs sampling.

Mackay, p. 382



http://www.cs.toronto.edu/~jessebett/CSC412/content/week8/07-sampling.pdf

Application: Texture Synthesis

A common application of Markov Random Fields is in Texture Synthesis, where the goal is to produce images that look like some naturally occuring textures, such as wood, grass, sand, etc.

The Gibbs sampler or the Metropolis sampler can be used for this purpose by providing samples of a suitable MRF.

Application: Texture Synthesis

By altering the MRF that is used, as well as the parameters of the MRF, different textures can be obtained.

As the sampling process is stochastic, different (but qualitatively similar) textures can be obtained from the same model and model parameters.

Example of the Gibbs Sampler - Ising Model Samples

To apply the Gibbs sampler to the Ising model, we first take some random assignment of pixels to the values (0,1) (e.g. a uniform distribution).

We run the Gibbs sampler iteration for some number of iterations to obtain statistical equilibrium (How many is this? Good question!).

Example of the Gibbs Sampler - Ising Model Samples

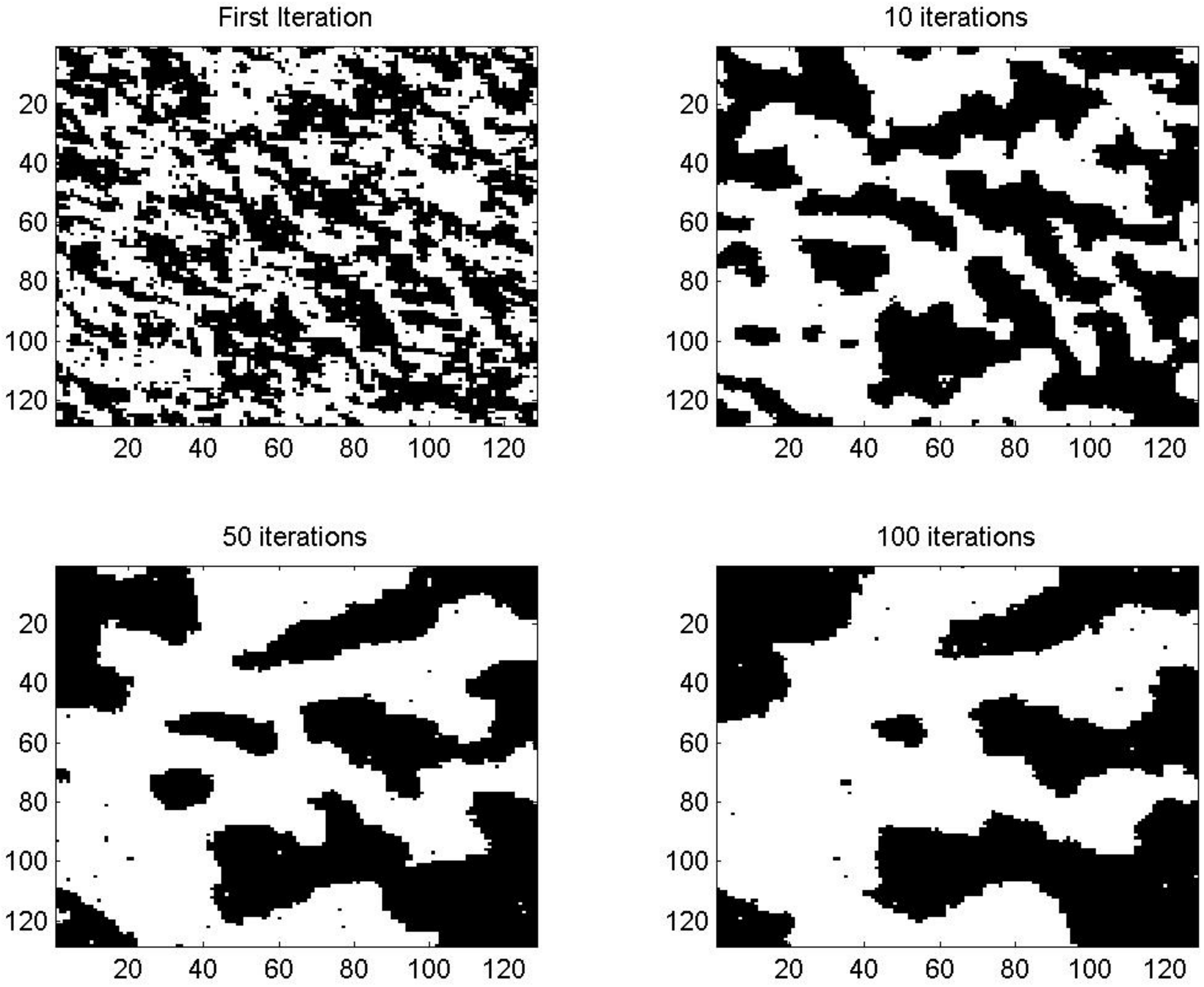
We use the conditional probability function for the Ising model as seen earlier.

For the isotropic case, recall that this is:

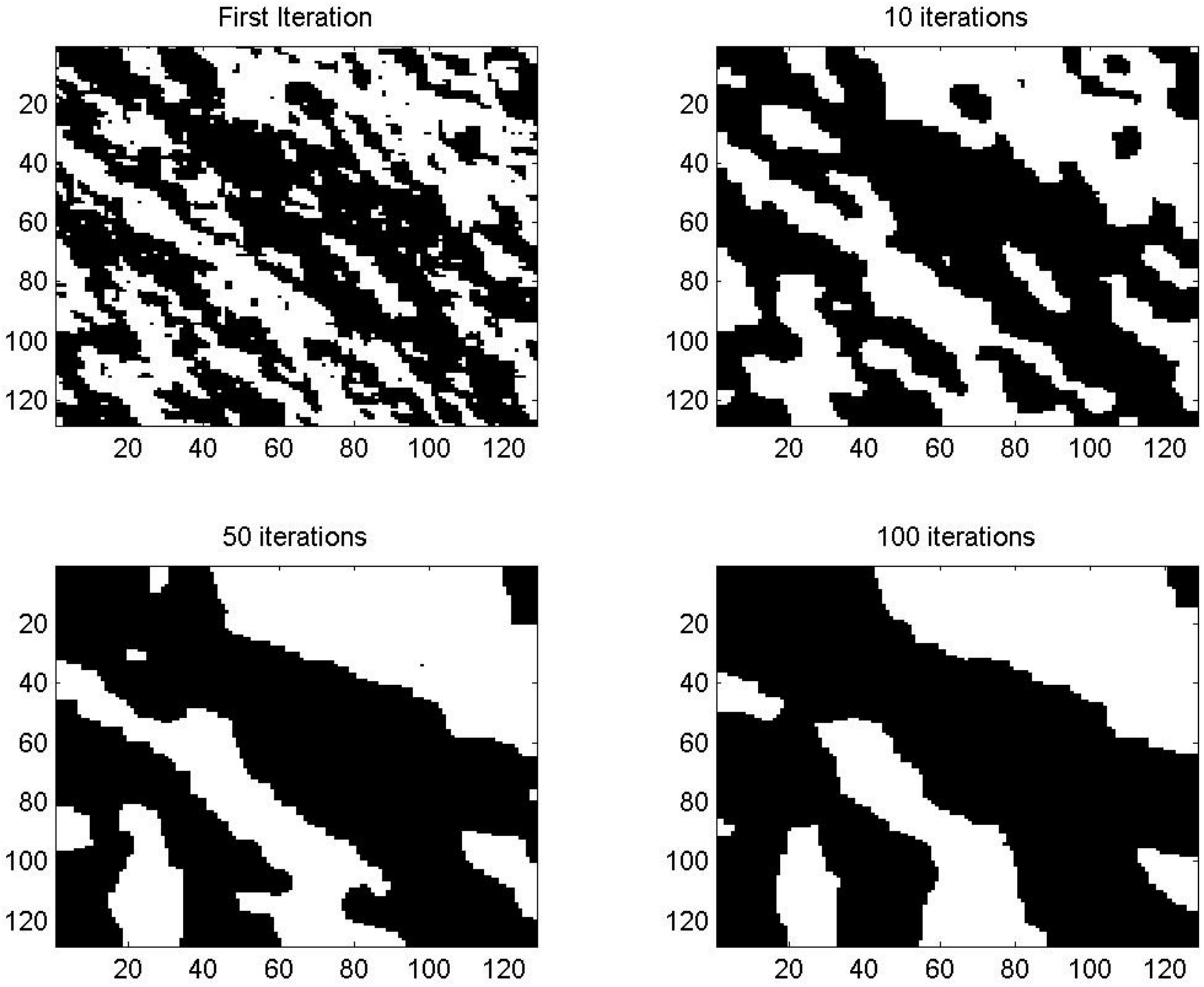
$$P(f_i \mid f_{N_i}) = \frac{\exp(\beta f_i \sum_{j \in N_i} f_j)}{2\cosh(\beta \sum_{j \in N_i} f_j)}$$

A Matlab program implementing Ising Textures:

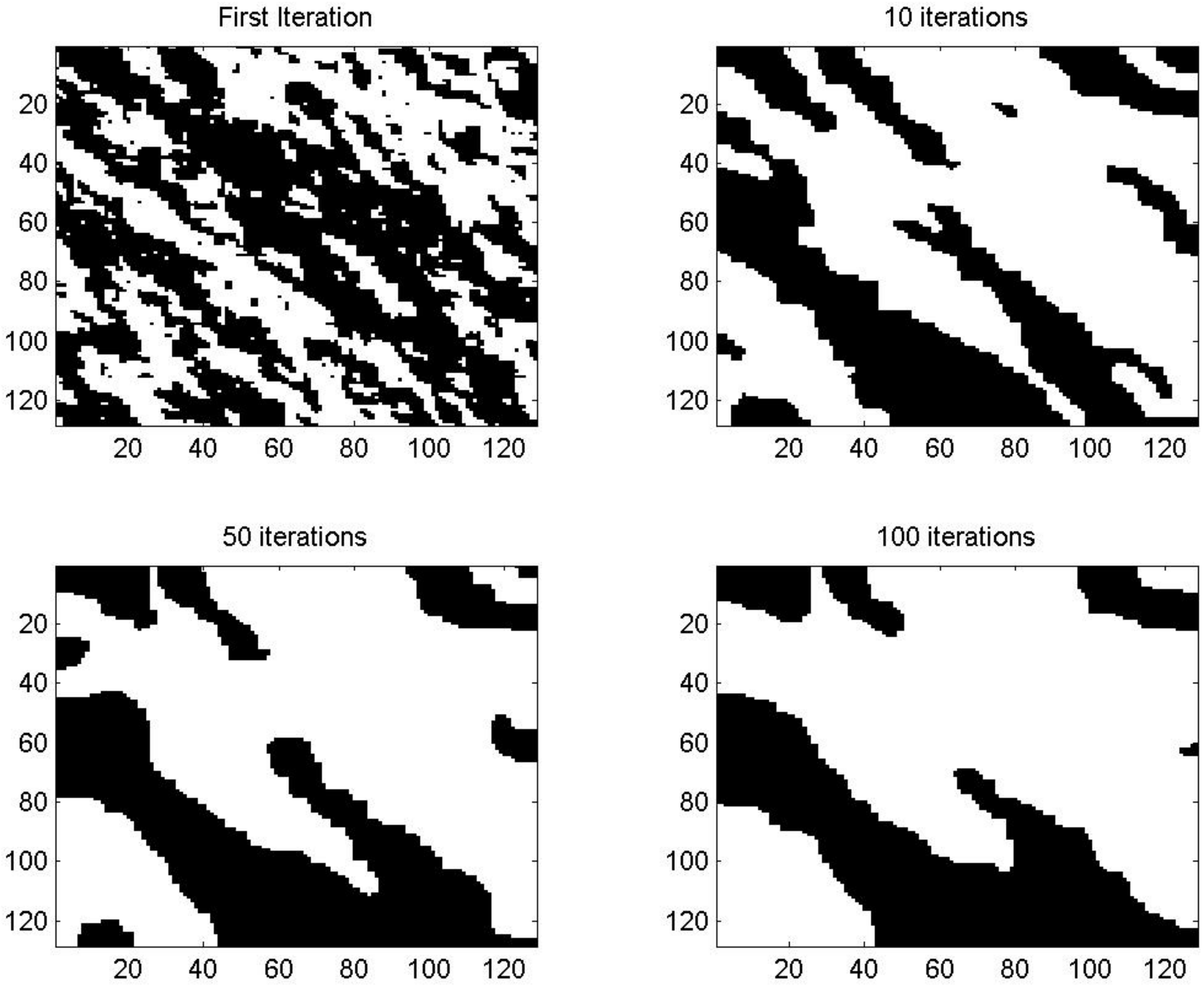
```
function im1=Isotropic_Ising_Gibbs_Sampler(im0,beta)
%
% assumes a 4-neighborhood isotropic model
% use a Raster-scan approach
%
[n,m]=size(im0);
im1=im0;
for row=1:n,
    for col=1:m,
        nbrs = 0;
        if(row > 1) nbrs = nbrs + im1(row-1,col);
        end
        if(row < n) nbrs = nbrs + 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(beta*nbrs)/(2.0*cosh(beta*nbrs)); % cond prob of f=+1
        im1(row,col)=-1;
        if(rand < cond) % rand generates a uniform variate from 0-1.
            im1(row,col) = 1;
        end
    end
end
```



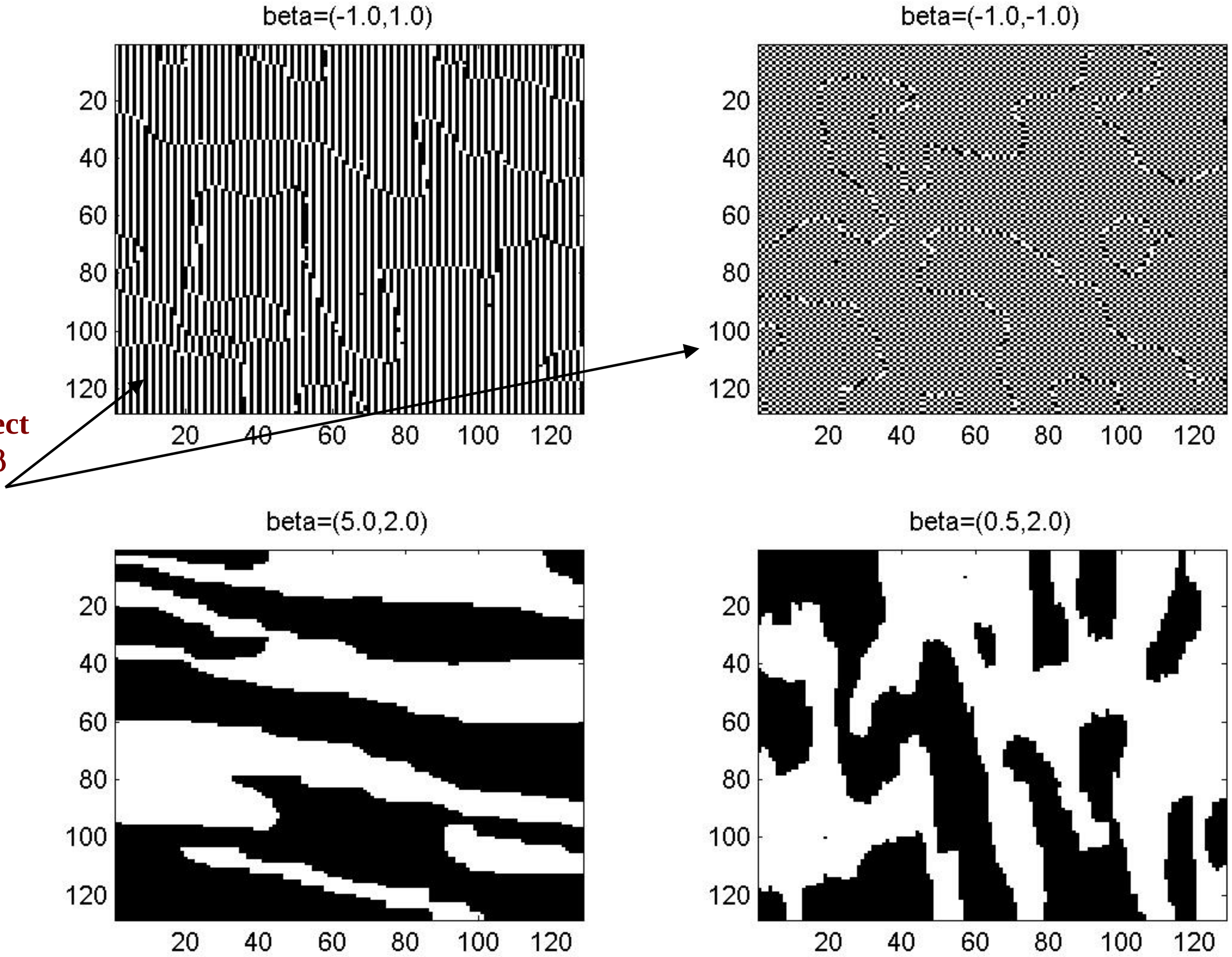
Ising model sample fields from the Gibbs sampler for β=0.8



Ising model sample fields from the Gibbs sampler for β=1.5

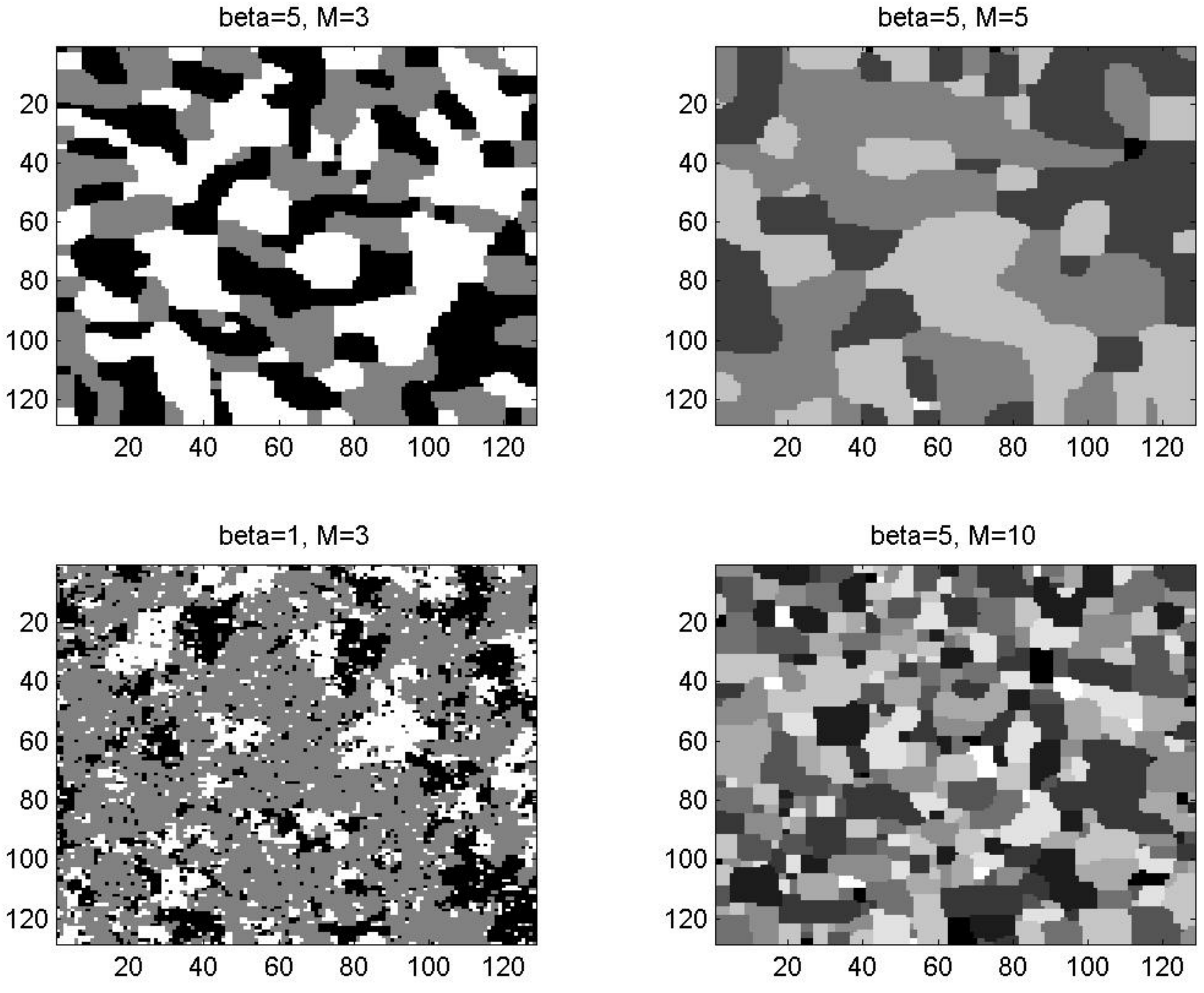


Ising model sample fields from the Gibbs sampler for β=4.0



Note the effect of negative β values!

Anisotropic Ising model sample fields (after 50 iterations)



Textures generated with a Multi-Level Logistic model MRF

Generation of Textures using Gaussian MRFs

For Gaussian MRFs with the second-order neighborhood (8-connected nbhd) we can use the following energy:

$$U(x) = \beta_1 \sum_{horiz} (x_i - x_j)^2 + \beta_2 \sum_{vert} (x_i - x_j)^2 + \beta_3 \sum_{LL-UR} (x_i - x_j)^2 + \beta_4 \sum_{UL-LR} (x_i - x_j)^2 + \varepsilon \sum_i x_i^2$$

where $horiz, vert, LL-UR, UL-LR$ denote the 4 different types of 2-neighbor cliques.

Generation of Textures using Gaussian MRFs

In this case, the conditional density is also Gaussian, with the following mean and variance:

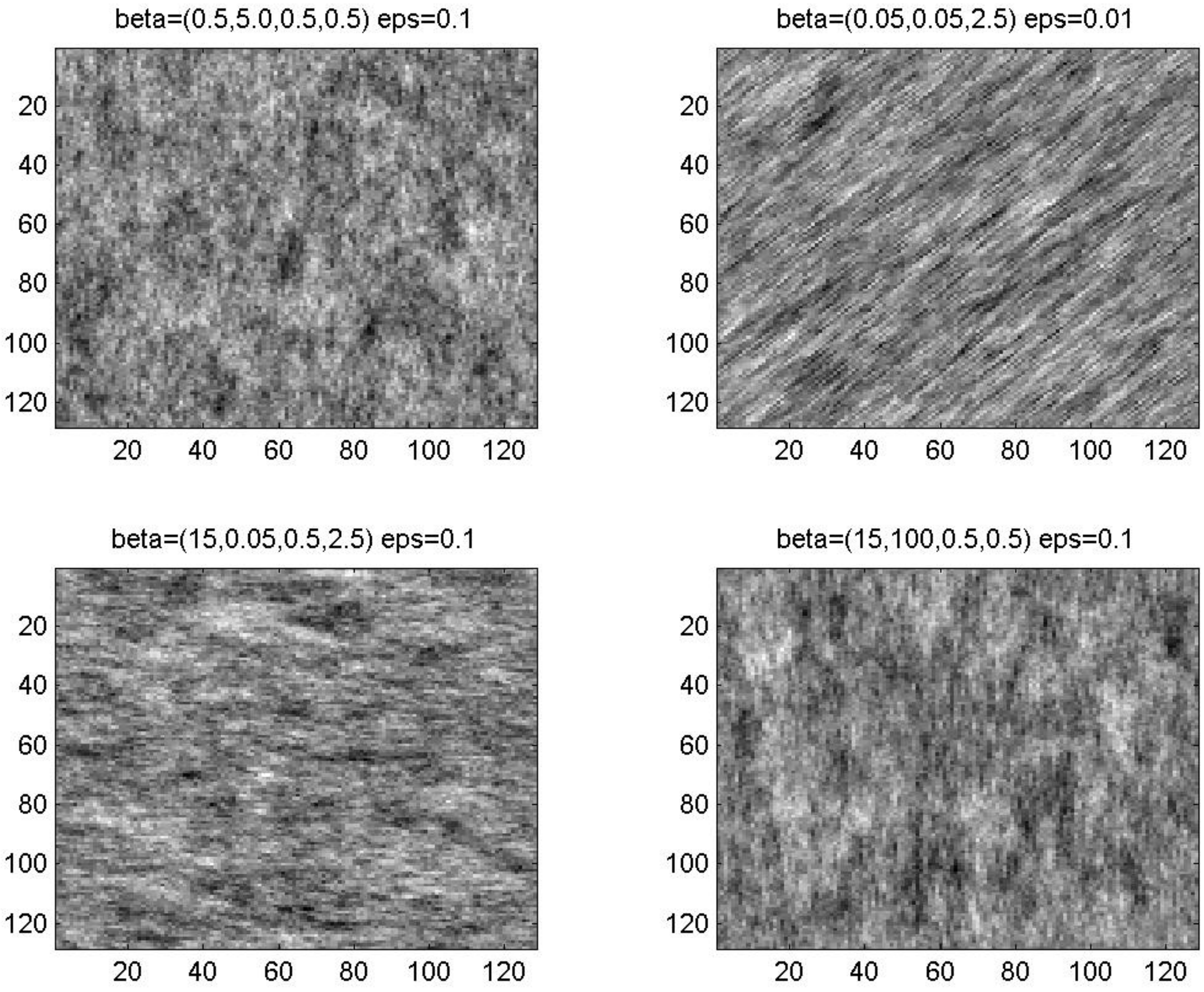
$$\mu = \frac{\sum_{j \in N(i)} \beta_{ij} x_j}{\varepsilon + 8\bar{\beta}}, \qquad \sigma = \frac{1}{2\varepsilon + 16\bar{\beta}}$$

where $\bar{\beta} = (\beta_1 + \beta_2 + \beta_3 + \beta_4)/4$ and $\beta_{ij} = \beta_1, \beta_2, \beta_3,$ or β_4 depending on the clique orientation.

A Matlab program implementing Gaussian textures:

```
function im1=Gaussian_Gibbs_Sampler(im0,beta1,beta2,beta3,beta4,eps)
%
% assumes a 8-neighborhood anisotropic model
% use a Raster-scan approach
%
[n,m]=size(im0);
im1=im0;
avg_beta=(beta1+beta2+beta3+beta4)/4.0;
sigma=1.0/(2.0*eps+16.0*avg_beta);
deviation = sqrt(sigma);
for row=1:n,
    for col=1:m,
        h_nbrs = 0; v_nbrs = 0; llur_nbrs = 0; ullr_nbrs = 0;
        if(row > 1) v_nbrs = v_nbrs + beta2*im1(row-1,col); end
        if(row < n) v_nbrs = v_nbrs + beta2*im1(row+1,col); end
        if(col < m) h_nbrs = h_nbrs + beta1*im1(row,col+1); end
        if(col > 1) h_nbrs = h_nbrs + beta1*im1(row,col-1); end

        if(row > 1 & col > 1) ullr_nbrs = ullr_nbrs + beta4*im1(row-1,col-1); end
        if(row < n & col < m) ullr_nbrs = ullr_nbrs + beta4*im1(row+1,col+1); end
        if(col < m & row > 1) llur_nbrs = llur_nbrs + beta3*im1(row-1,col+1); end
        if(col > 1 & row < n) llur_nbrs = llur_nbrs + beta3*im1(row+1,col-1); end
        mn = (h_nbrs+v_nbrs+llur_nbrs+ullr_nbrs)*2.0*sigma;
        im1(row,col)=(randn*deviation)+mn;
    end
end
```



beta=(0.5,5.0,0.5,0.5) eps=0.1

beta=(0.05,0.05,2.5) eps=0.01

beta=(15,0.05,0.5,2.5) eps=0.1

beta=(15,100,0.5,0.5) eps=0.1

Examples of Gaussian MRF textures created with different values of β