Belief propagation and MRF's

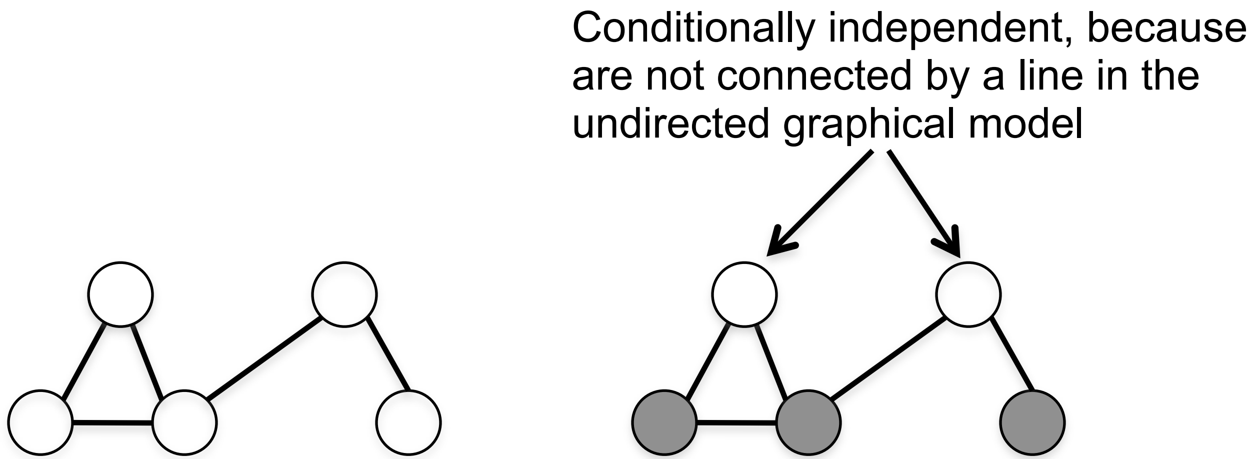
Bill Freeman

6.869

March 7, 2011

Undirected graphical models

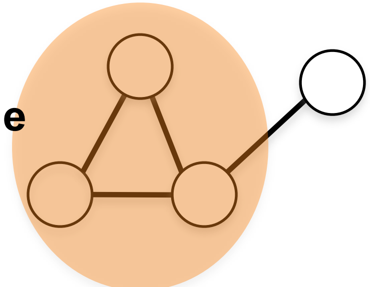
- **A set of nodes joined by undirected edges.**
- **The graph makes conditional independencies explicit: If two nodes are not linked, and we condition on every other node in the graph, then those two nodes are conditionally independent.**

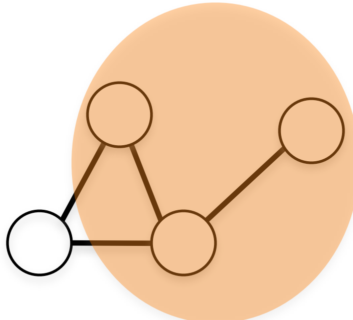


Undirected graphical models: cliques

- **Clique: a fully connected set of nodes**

clique

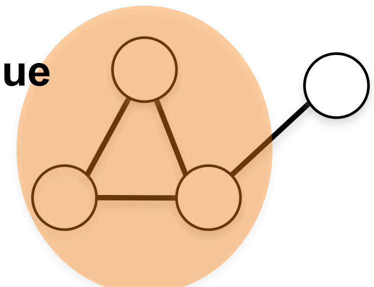




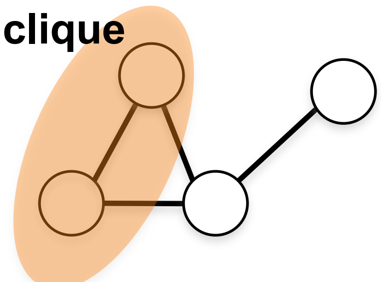
not a clique

- **A maximal clique is a clique that can’t include more nodes of the graph w/o losing the clique property.**

Maximal clique



Non-maximal clique



Undirected graphical models: probability factorization

- **Hammersley-Clifford theorem addresses the pdf factorization implied by a graph: A distribution has the Markov structure implied by an undirected graph iff it can be represented in the factored form**

$$P_x = \frac{1}{Z}\prod_{c\in\xi}\Psi_{x_c}$$

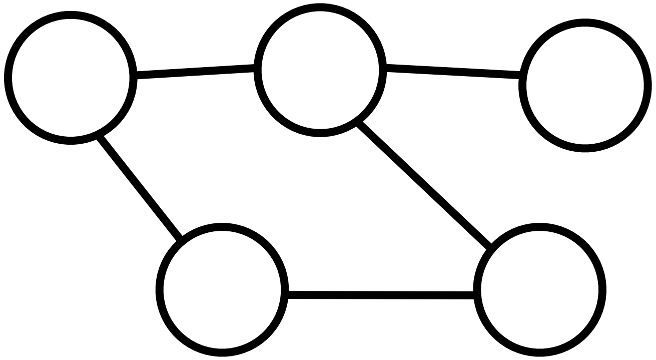
Normalizing constant

set of maximal cliques

Potential functions of states of variables in maximal clique

Graphical Models

Markov Random Fields

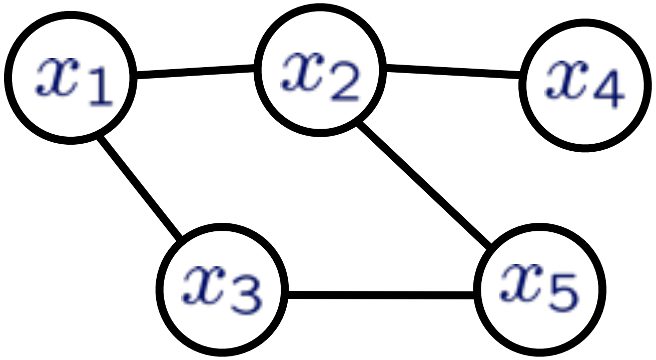


$\mathcal{V}$ ⟶ **set of** N **nodes**

$\mathcal{E}$ ⟶ **edges** (i,j) **connecting nodes** $i, j \in \mathcal{V}$

Graphical Models

Markov Random Fields



$\mathcal{V}$ → **set of** N nodes

$\mathcal{E}$ → edges (i,j) **connecting nodes** $i, j \in \mathcal{V}$

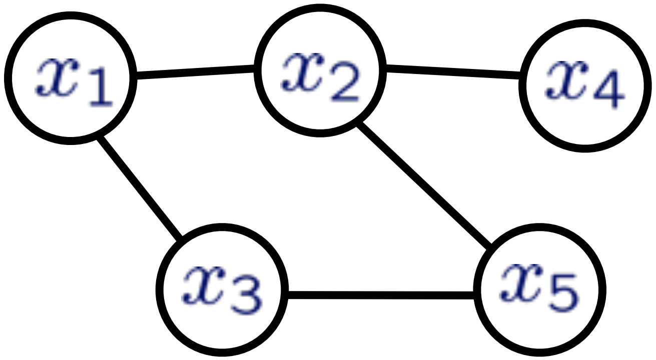
Nodes $i \in \mathcal{V}$ **are associated with** hidden variables x_i

$$p(x \mid y) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y)$$

Potential functions may depend on observations y

Graphical Models

Markov Random Fields



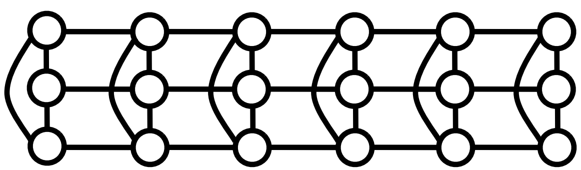
$\mathcal{V} \longrightarrow$ **set of N nodes**

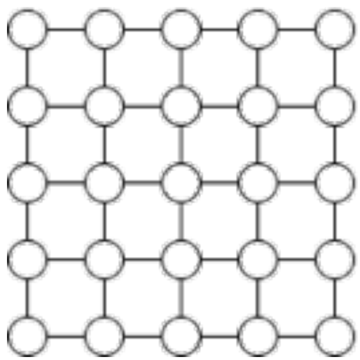
$\mathcal{E} \longrightarrow$ **edges (i,j) connecting nodes $i, j \in \mathcal{V}$**

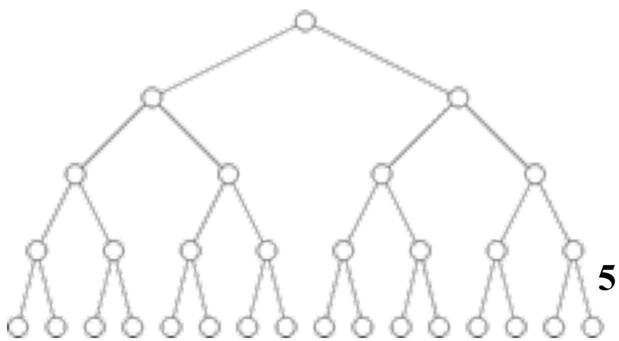
Nodes $i \in \mathcal{V}$ are associated with hidden variables x_i

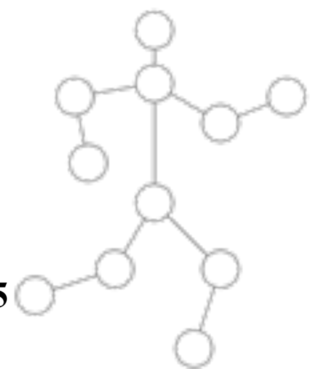
$$p(x \mid y) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y)$$

Potential functions may depend on observations y



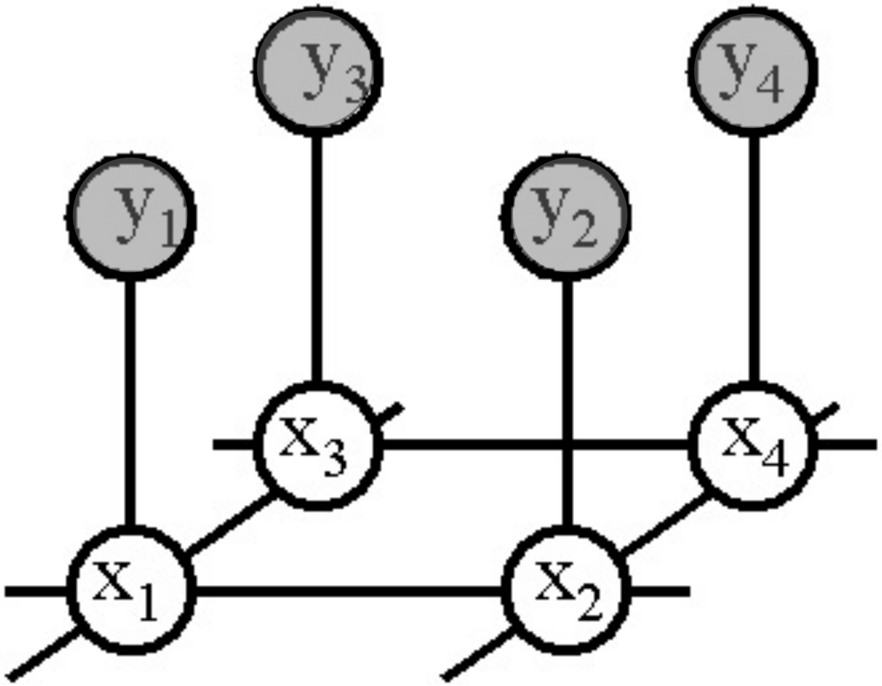






Markov Random Fields (MRF)

- Oftentimes MRF's have a regular, grid-like structure (but they don't need to).



MRF nodes as pixels

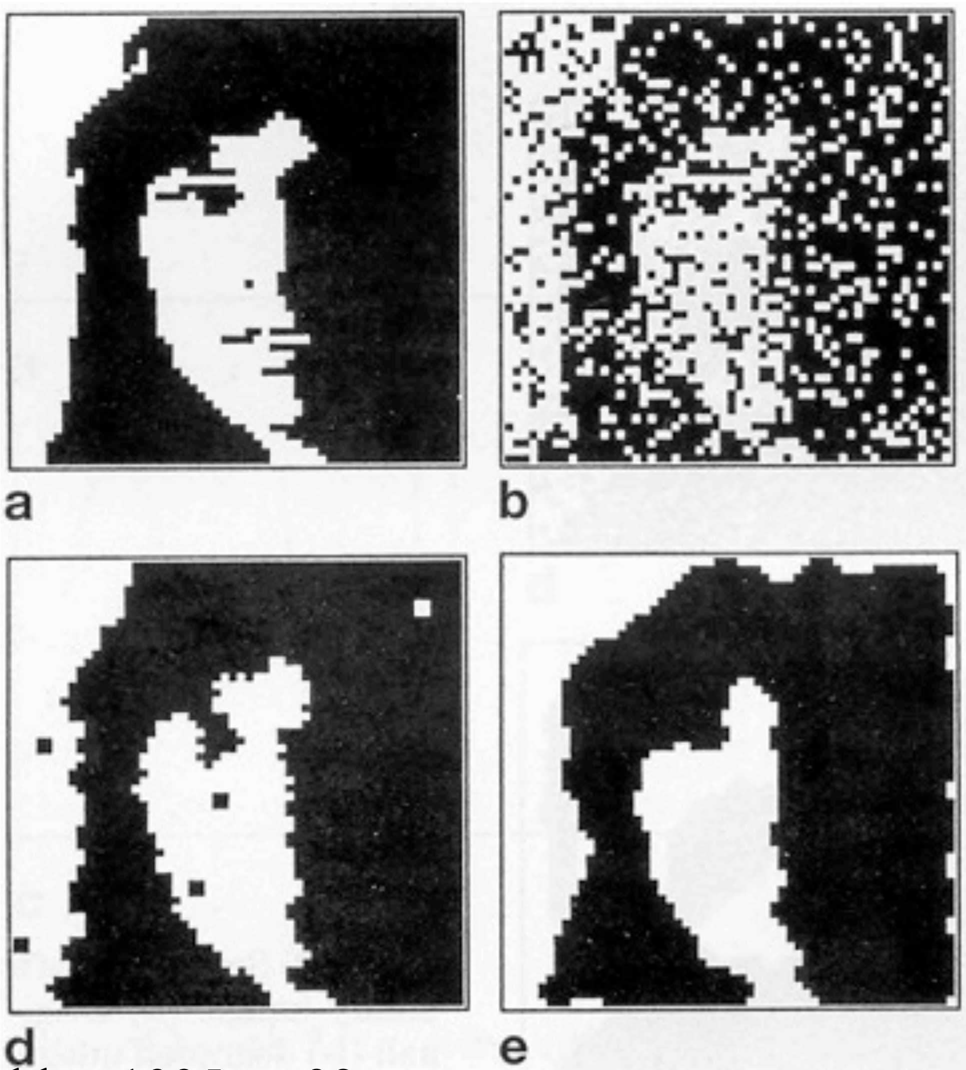
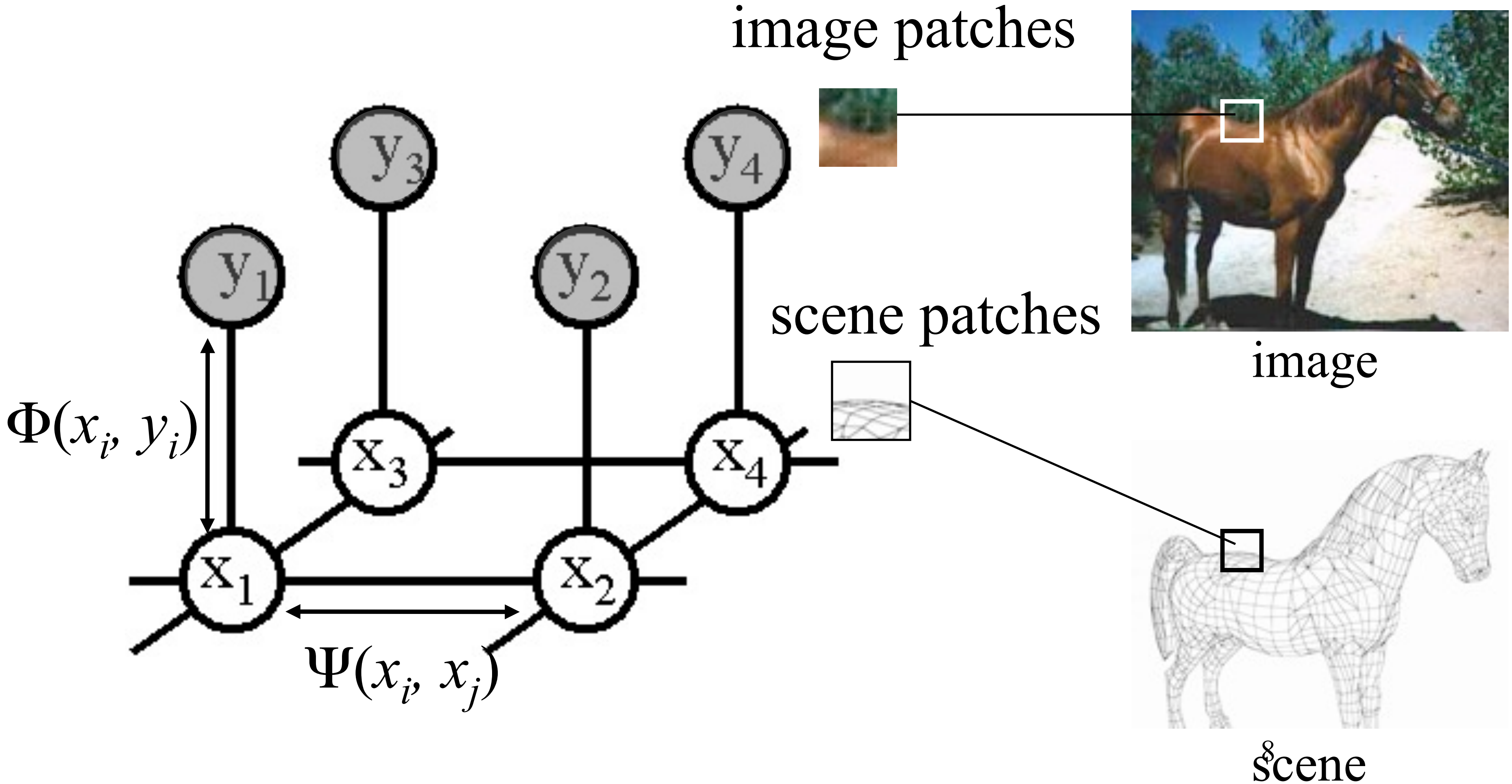


Fig. 2.3. Smoothing with the wrong prior. (**a**) Original, (**b**) degraded image, (**c**) MAP estimate $\beta = 1$, (**d**) MAP estimate $\beta = 0.3$, (**e**) median filter

Winkler, 1995, p. 32

MRF nodes as patches



Network joint probability

$$P(x,y) = \frac{1}{Z} \prod_{i,j} \Psi(\underbrace{x_i , x_j}_{\text{neighboring scene nodes}}) \prod_{i} \Phi(x_i , y_i)$$

scene

image

Scene-scene compatibility function

neighboring scene nodes

Image-scene compatibility function

local observations

Energy formulation

$$\mathrm{E}(x,y) = k + \sum_{(i,j)} \beta(x_i, \underbrace{x_i, x_j}_{\text{neighboring scene nodes}}) + \sum \alpha(x_i, y_i)$$

scene

image

Scene-scene compatibility function

neighboring scene nodes

Image-scene compatibility function

local observations

In order to use MRFs:

11 22

In order to use MRFs:

- Given observations y, and the parameters of the MRF, how <u>infer</u> the hidden variables, x?

In order to use MRFs:

- Given observations y, and the parameters of the MRF, how <u>infer</u> the hidden variables, x?
- How <u>learn</u> the parameters of the MRF?

Markov Random Fields (MRF's)

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
  - Variational methods
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
  - Variational methods
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Gibbs Sampling

- Gibbs sampling:
  - A way to generate random samples from a (potentially very complicated) probability distribution.
  - Fix all dimensions except one. Draw from the resulting 1-d conditional distribution. Repeat for all dimensions, and repeat many times.
  - Take an average of a subset of those samples to estimate the posterior mean. (Wait for a “burn in” period before including samples. And then subsample Gibbs sampler outputs to find independent draws of the joint probability).

Reference: Geman and Geman, IEEE PAMI 1984.

Sampling from a 1-d function

Sampling from a 1-d function

1. Discretize the density function

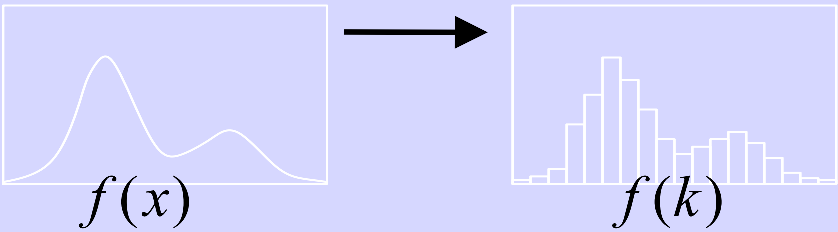
Sampling from a 1-d function

1. Discretize the density function

$f(x)$

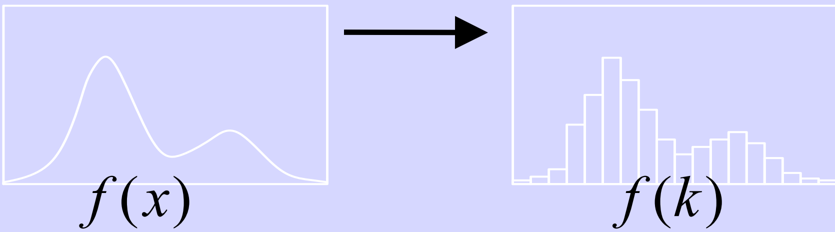
Sampling from a 1-d function

1. Discretize the density function



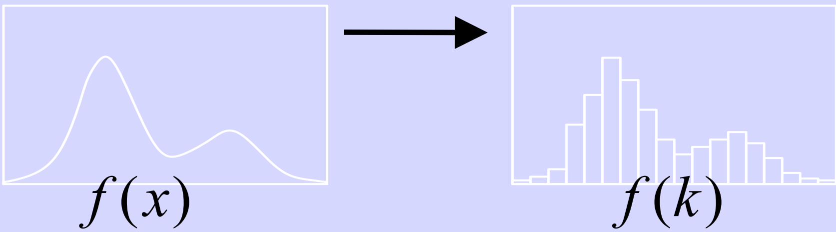
Sampling from a 1-d function

1. Discretize the density function



Sampling from a 1-d function

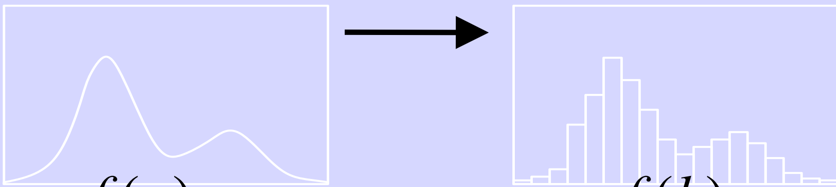
1. Discretize the density function



2. Compute distribution function from density function

Sampling from a 1-d function

1. Discretize the density function



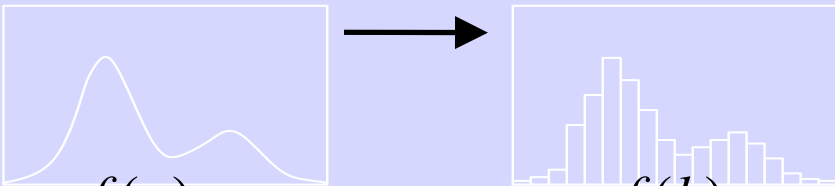
$f(x)$ $\qquad$ $f(k)$

$f(k)$

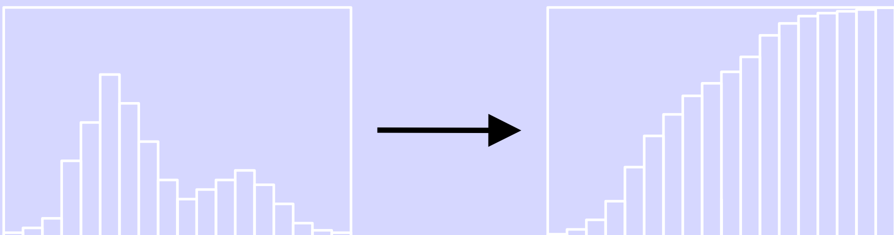
2. Compute distribution function from density function

Sampling from a 1-d function

1. Discretize the density function



$f(x)$ $f(k)$

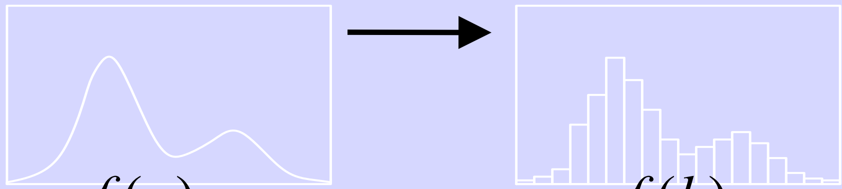


$f(k)$ $F(k)$

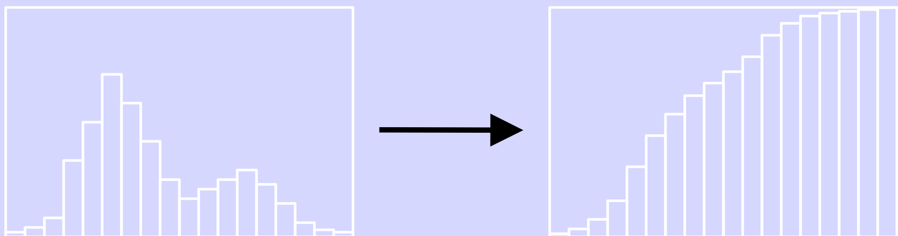
2. Compute distribution function from density function

Sampling from a 1-d function

1. Discretize the density function



$f(x)$ $f(k)$



$f(k)$ $F(k)$

2. Compute distribution function from density function

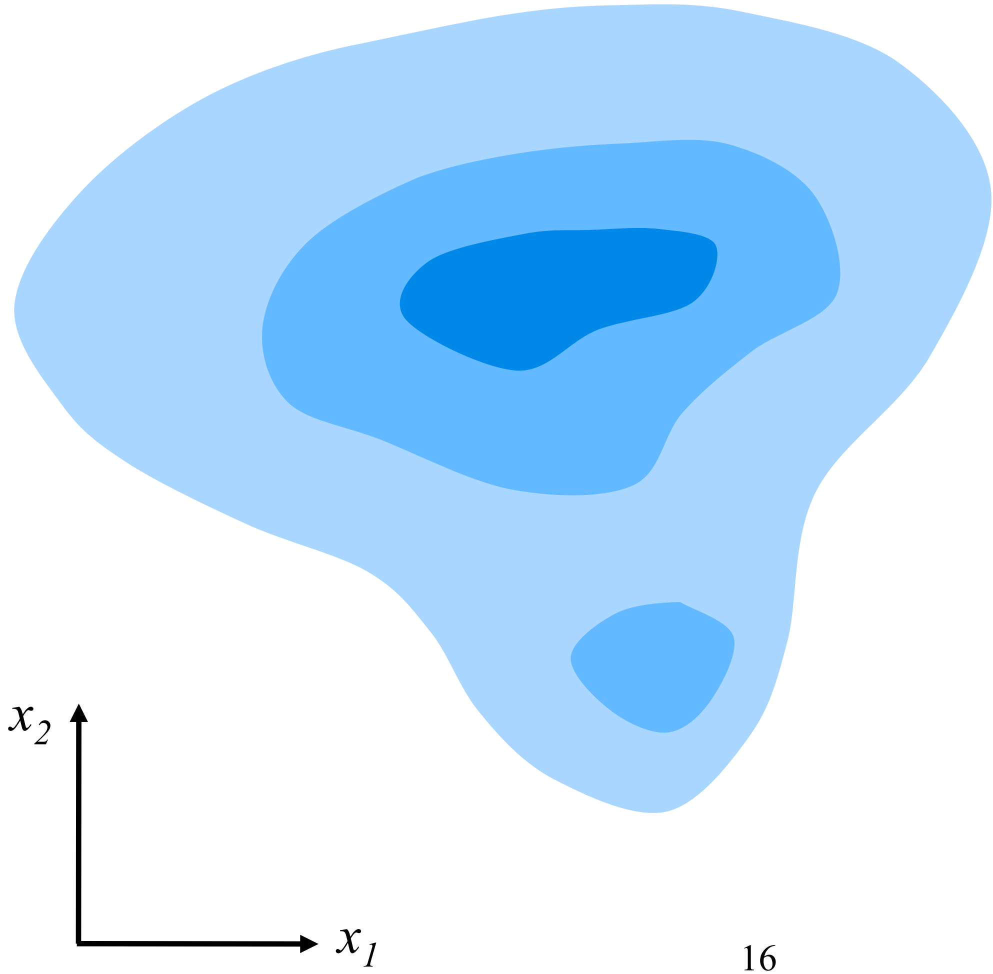
3. Sampling

draw $\alpha \sim U(0,1)$;
for $k = 1$ to n
  if $F(k) \geq \alpha$
    break;
$x = x_0 + k\tau$;

Gibbs Sampling

Slide by Ce Liu

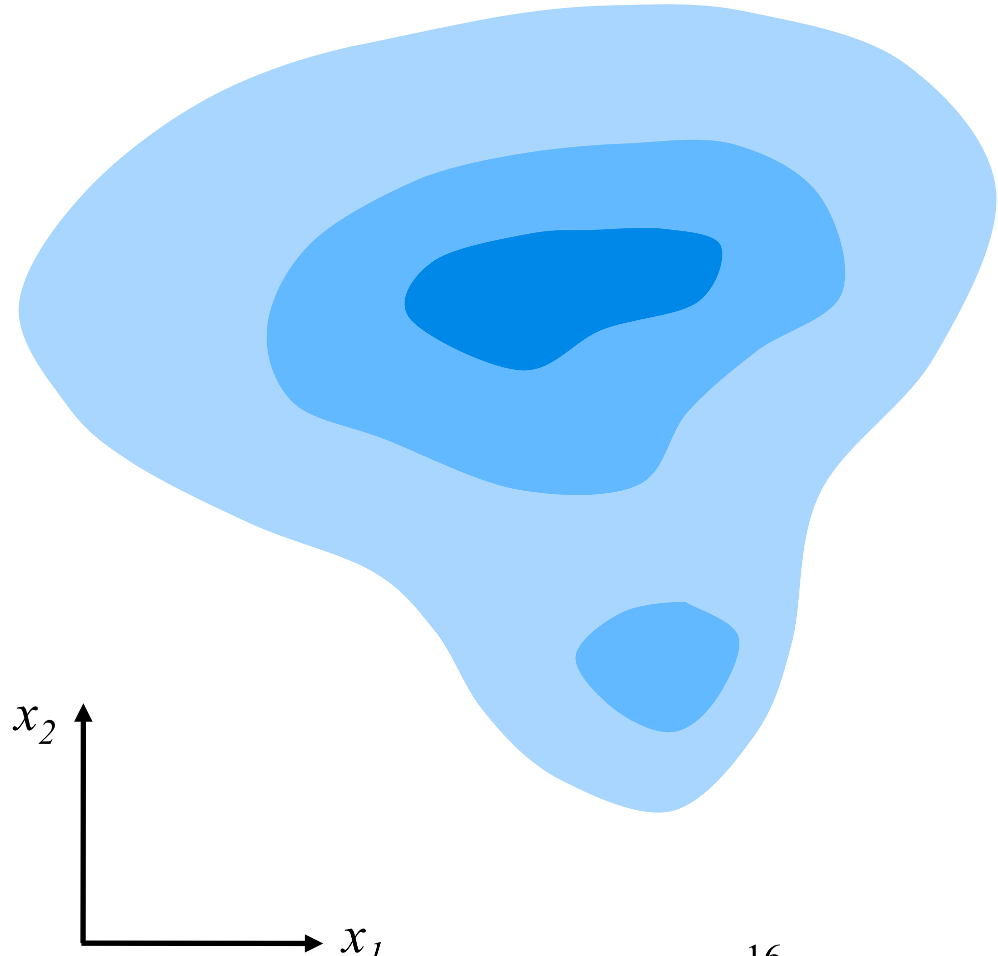
Gibbs Sampling



Slide by Ce Liu

Gibbs Sampling

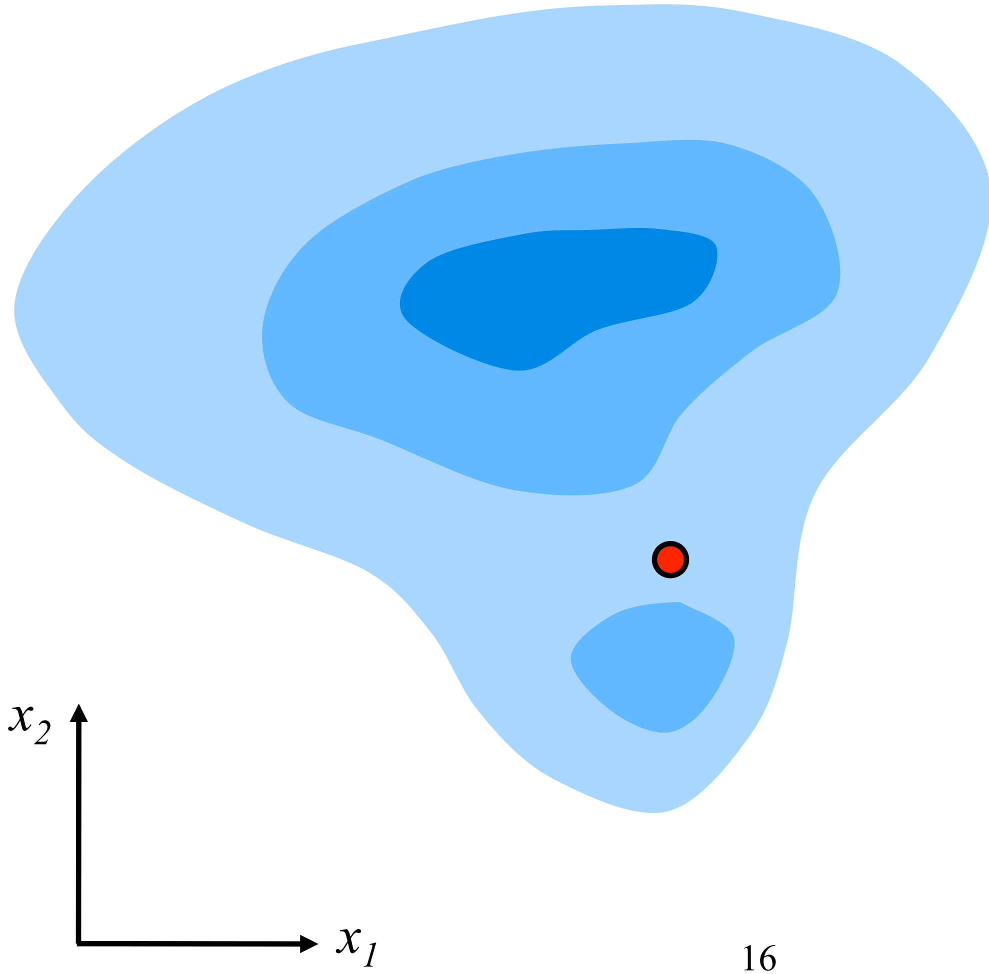
$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$



Slide by Ce Liu

Gibbs Sampling

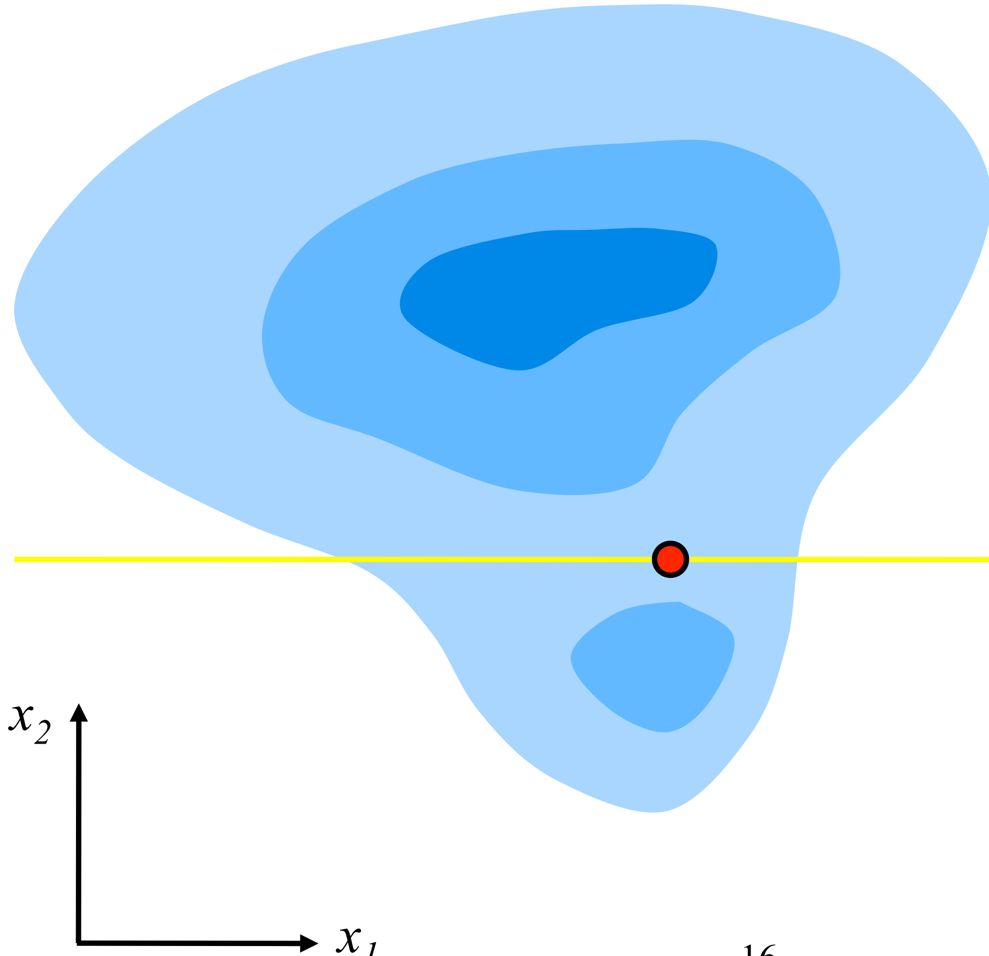
$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$



Slide by Ce Liu

Gibbs Sampling

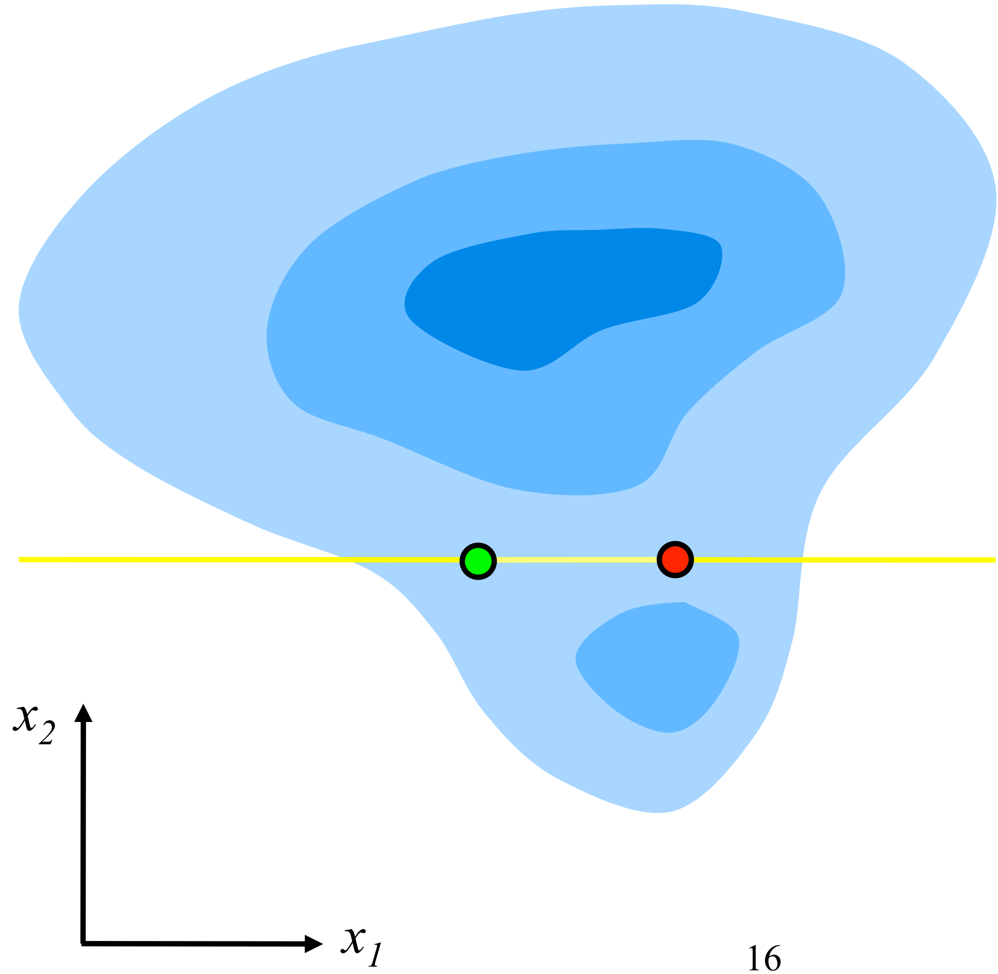
$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$



Slide by Ce Liu

Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

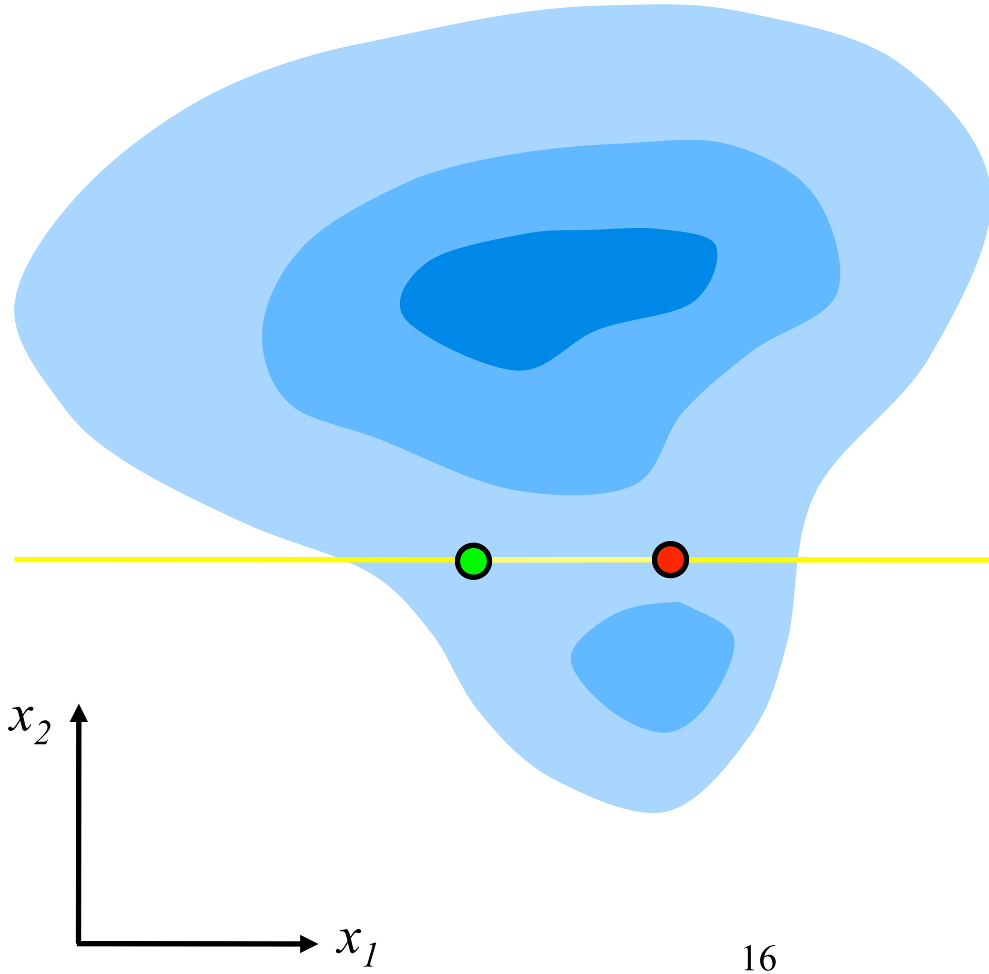


Slide by Ce Liu

Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

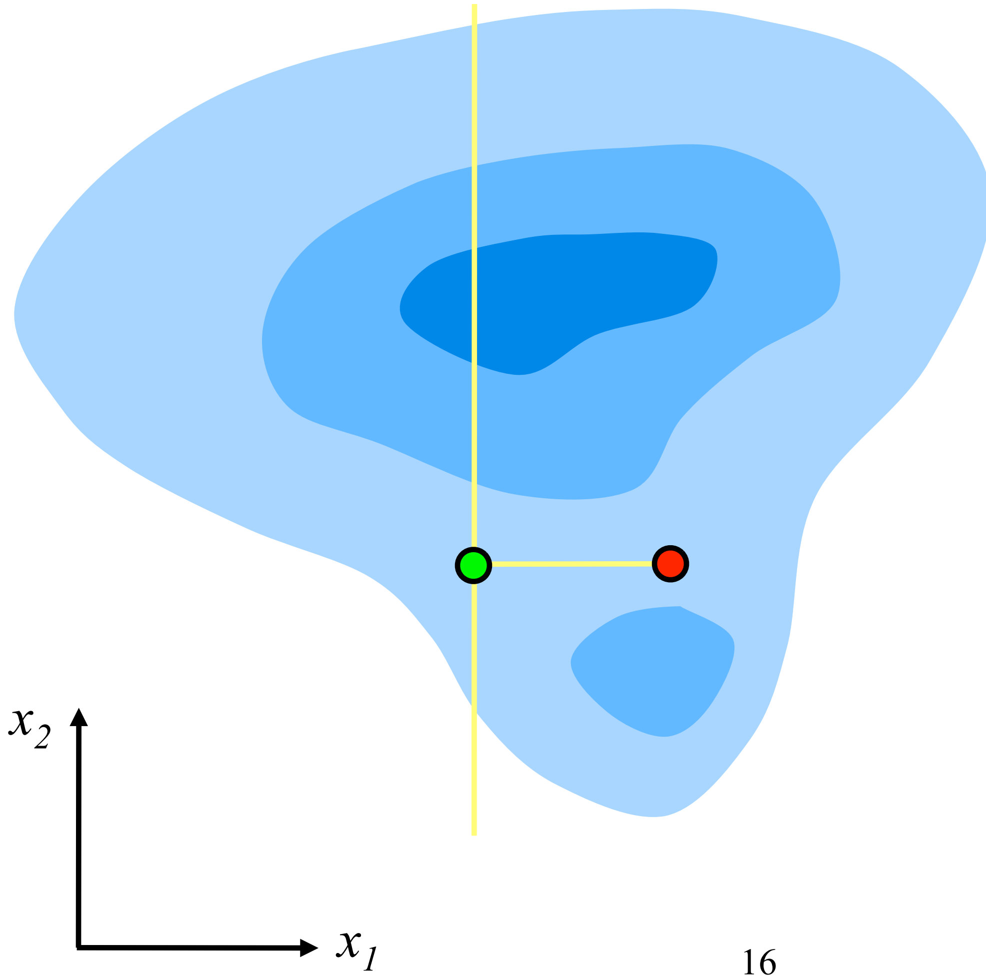


Slide by Ce Liu

Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

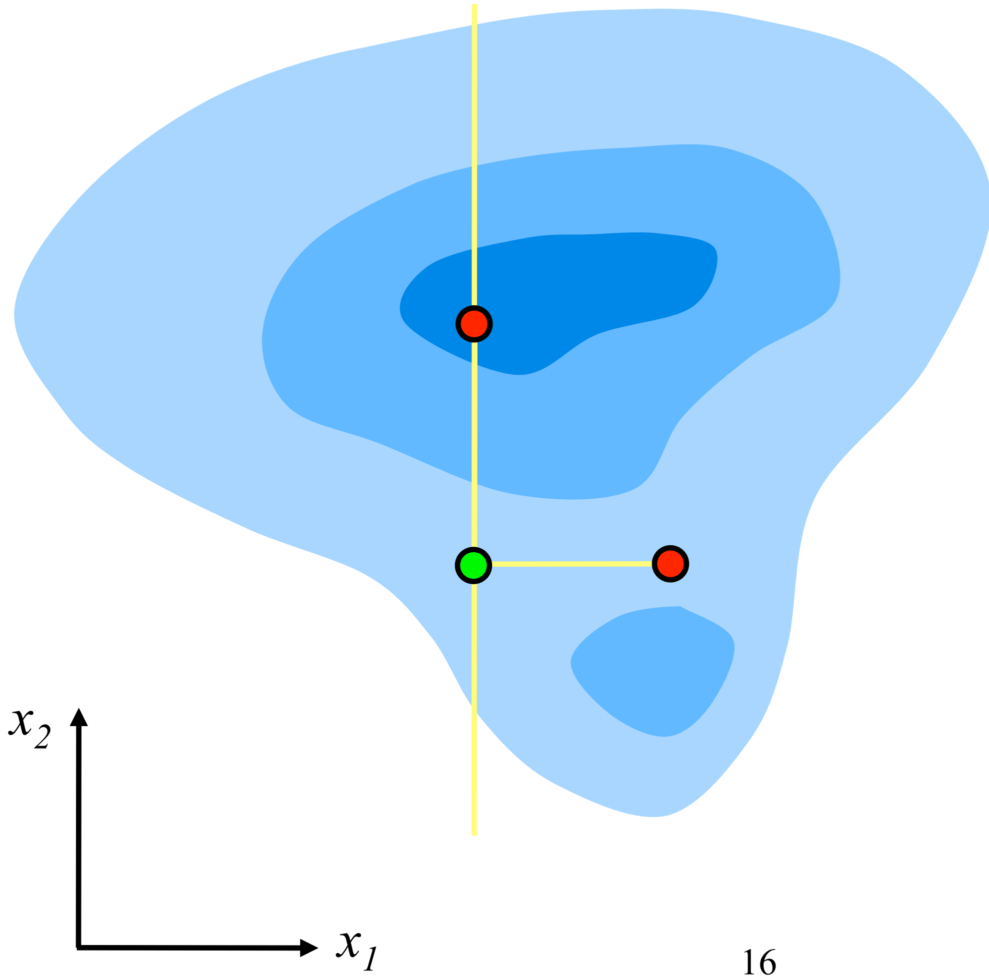


Slide by Ce Liu

Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

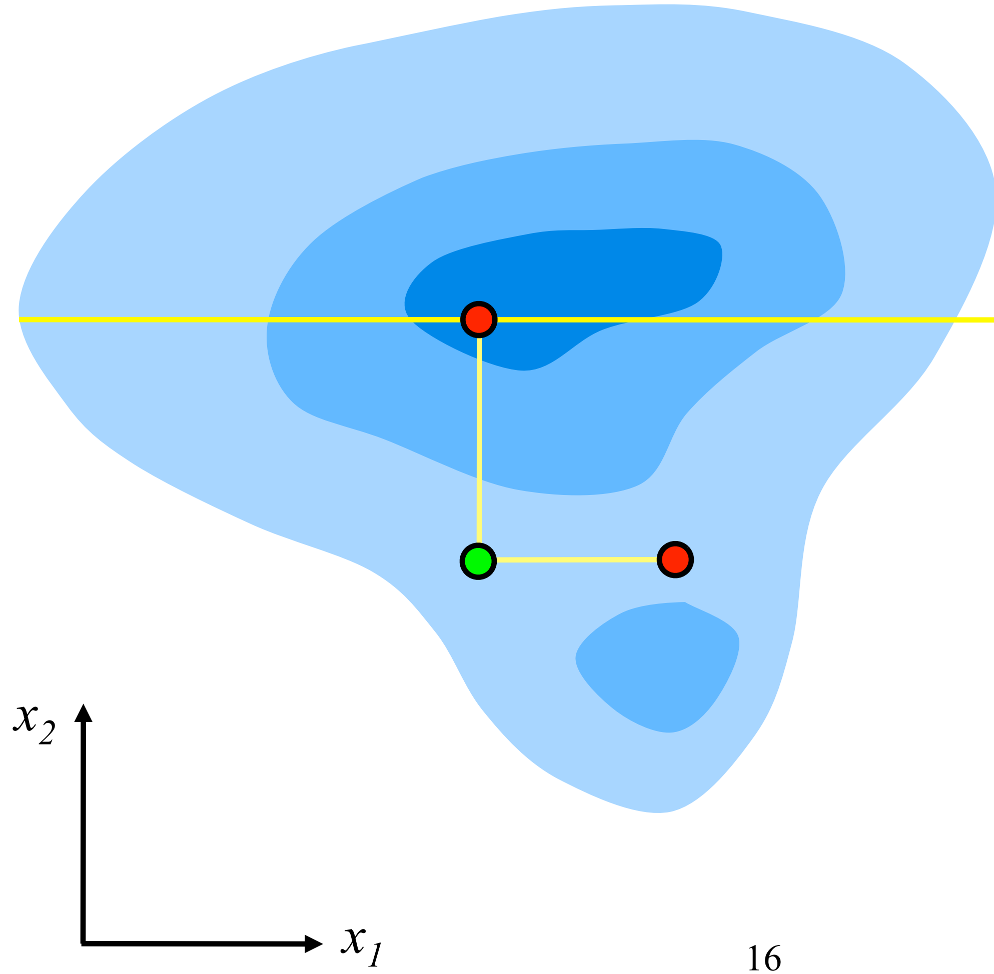


Slide by Ce Liu

Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$



Slide by Ce Liu

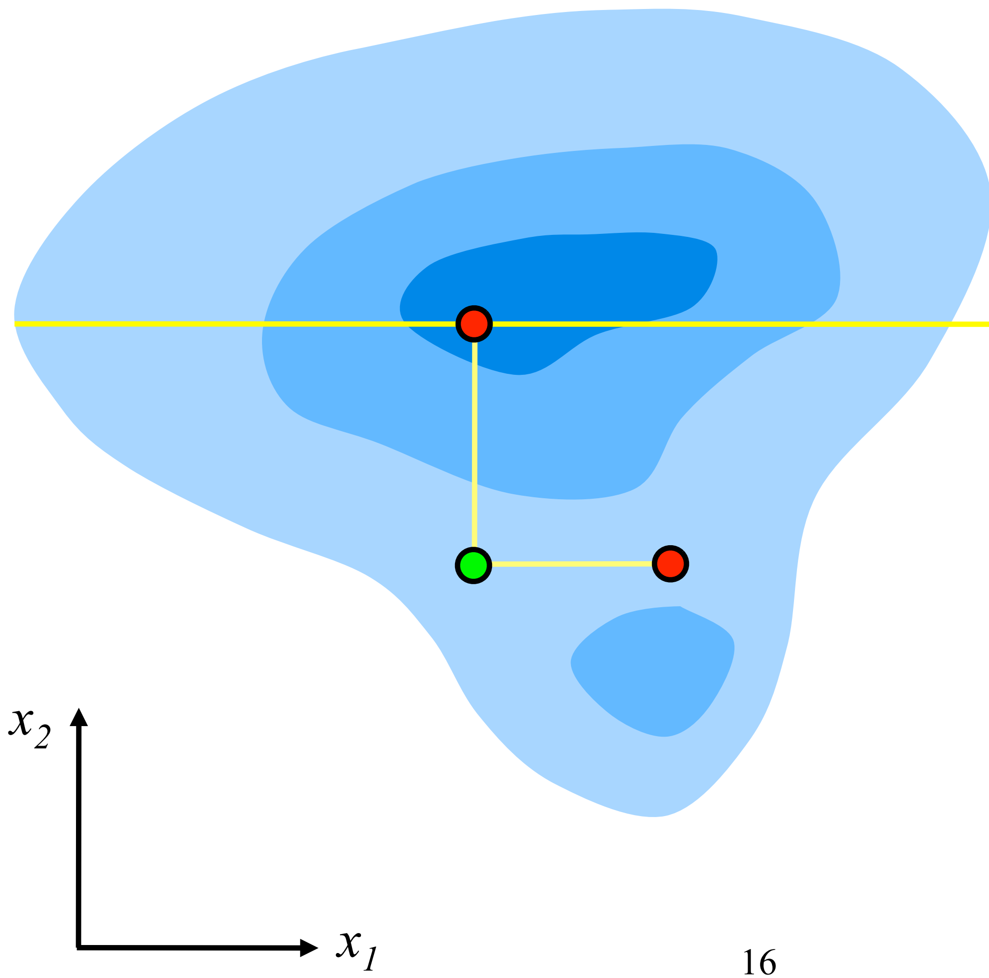
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

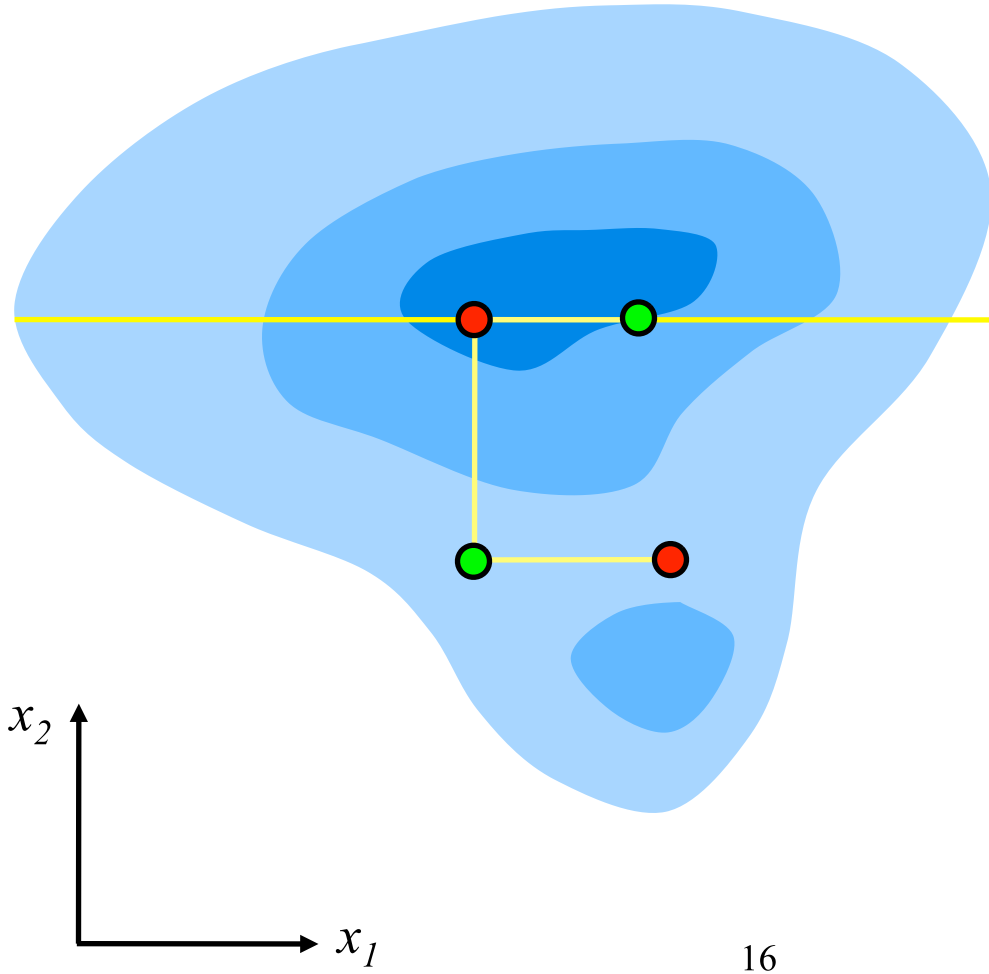
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

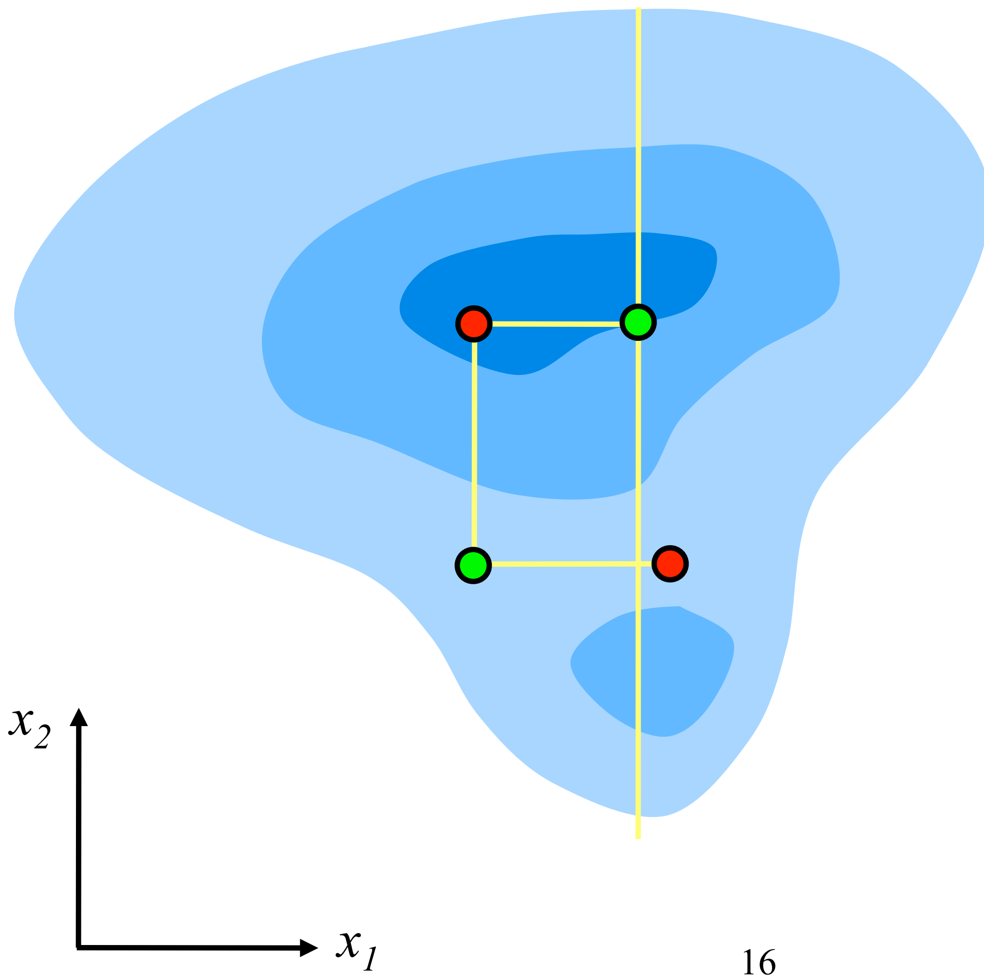
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

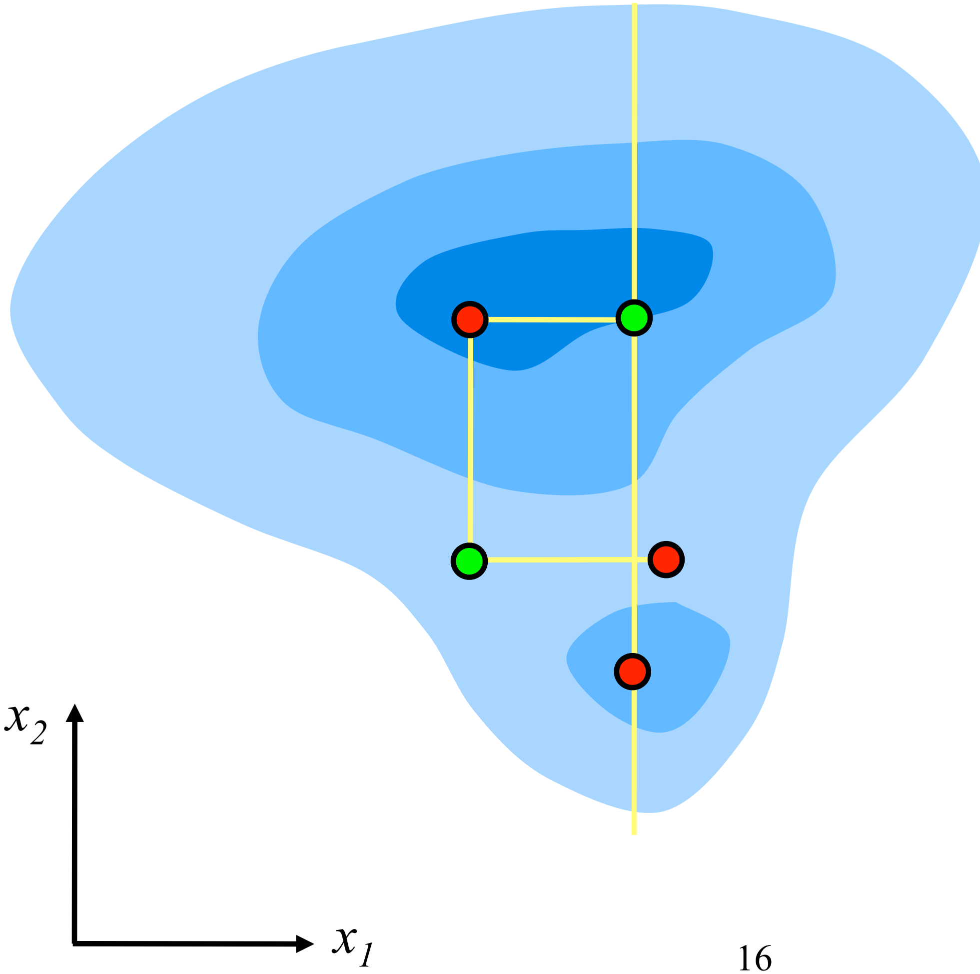
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

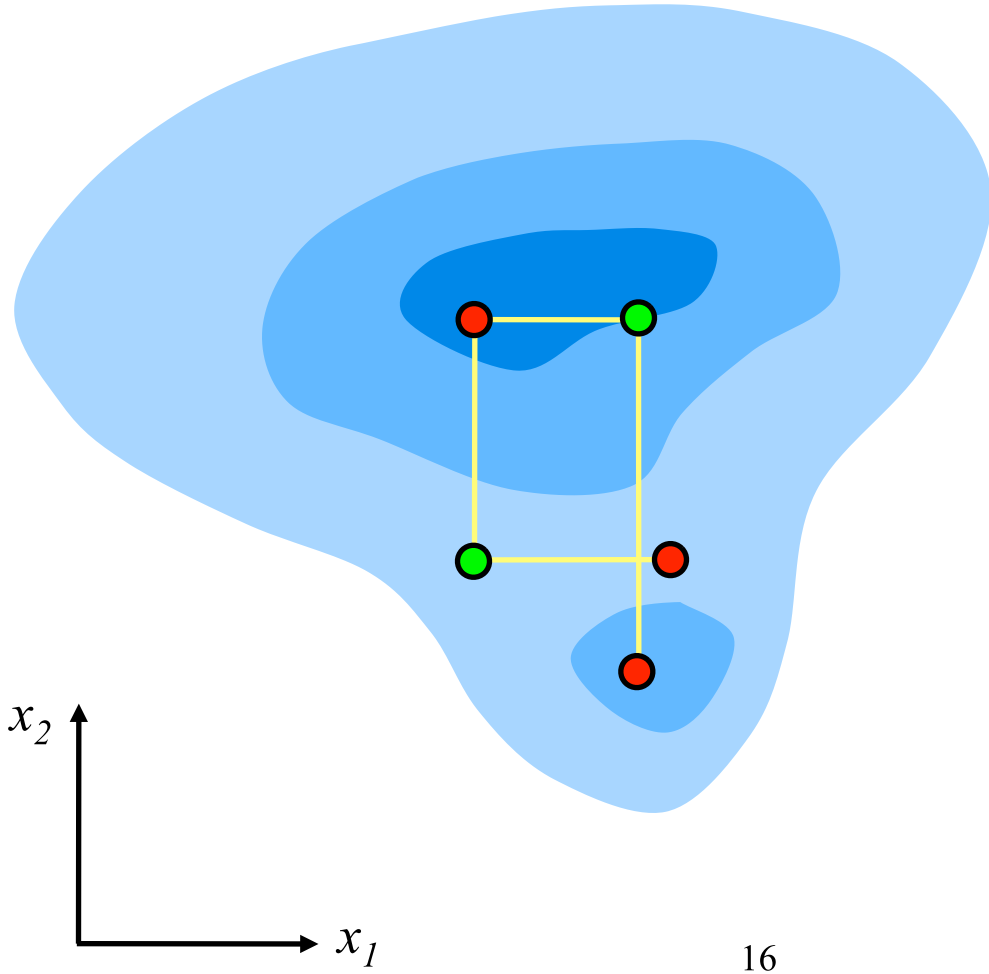
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \partial(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

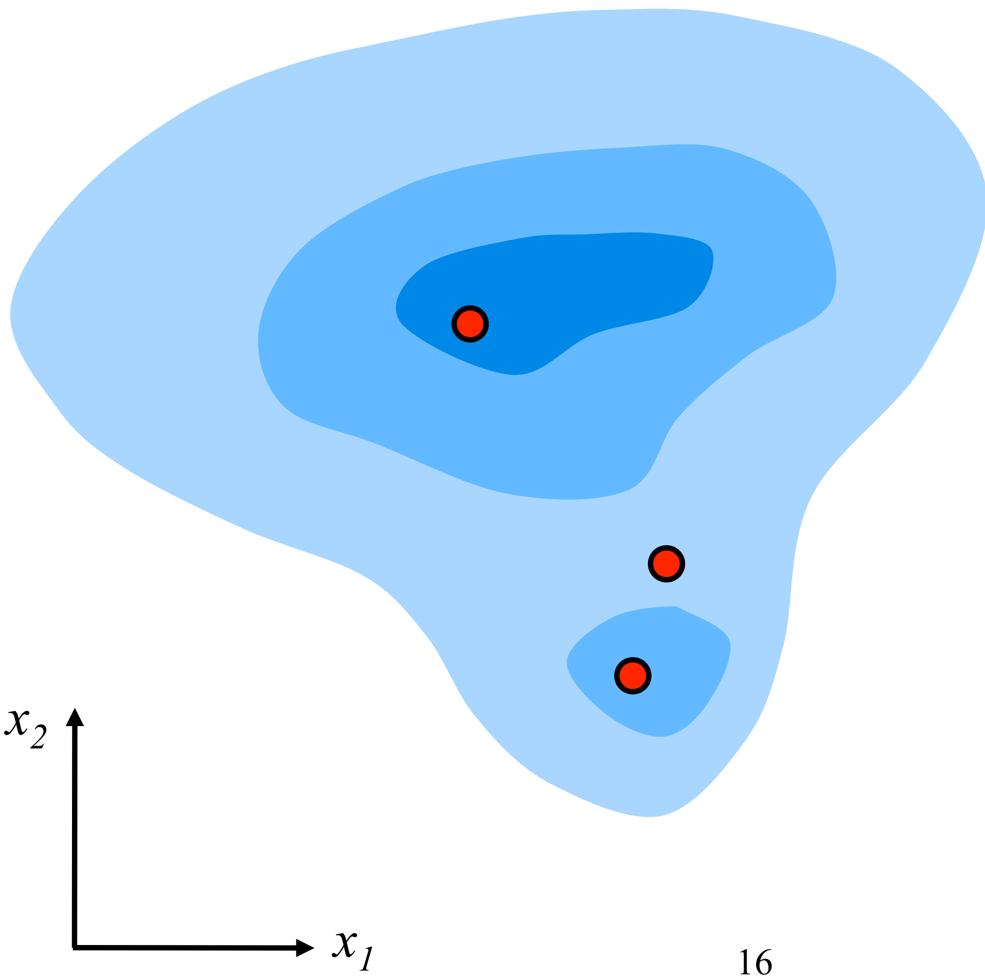
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \partial(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

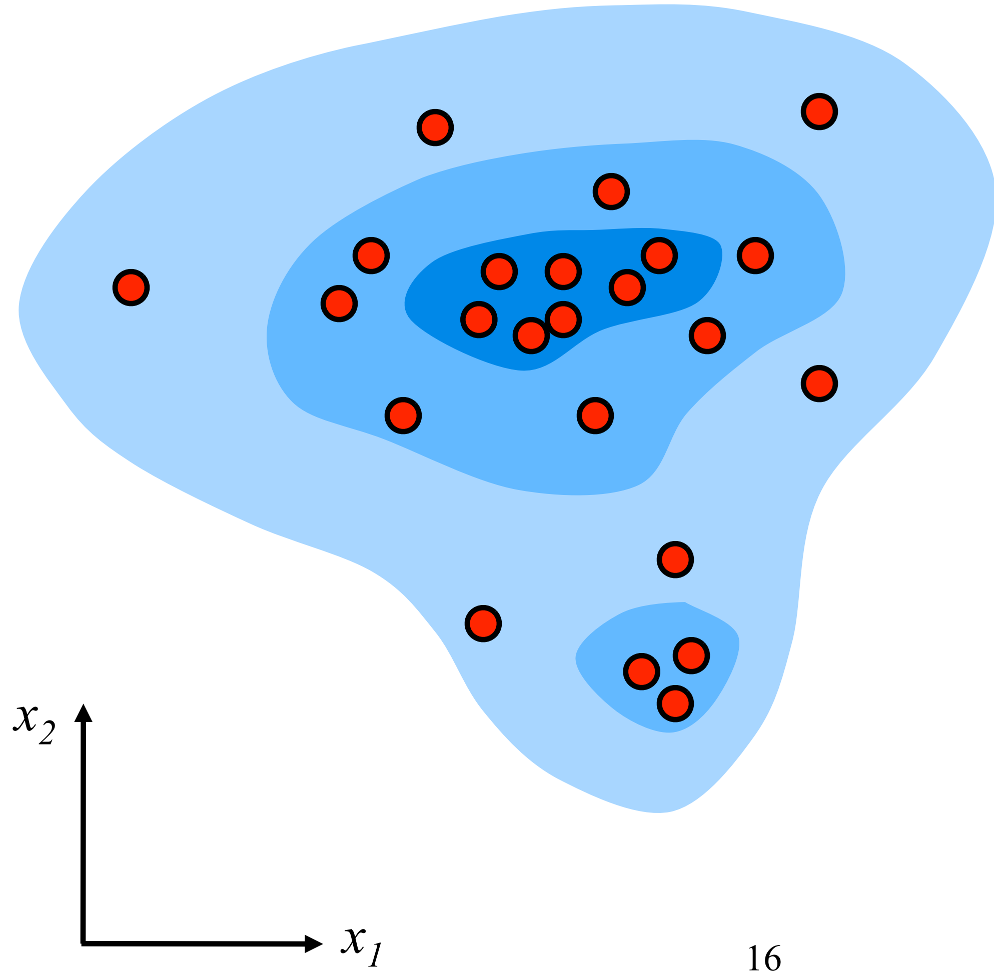
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

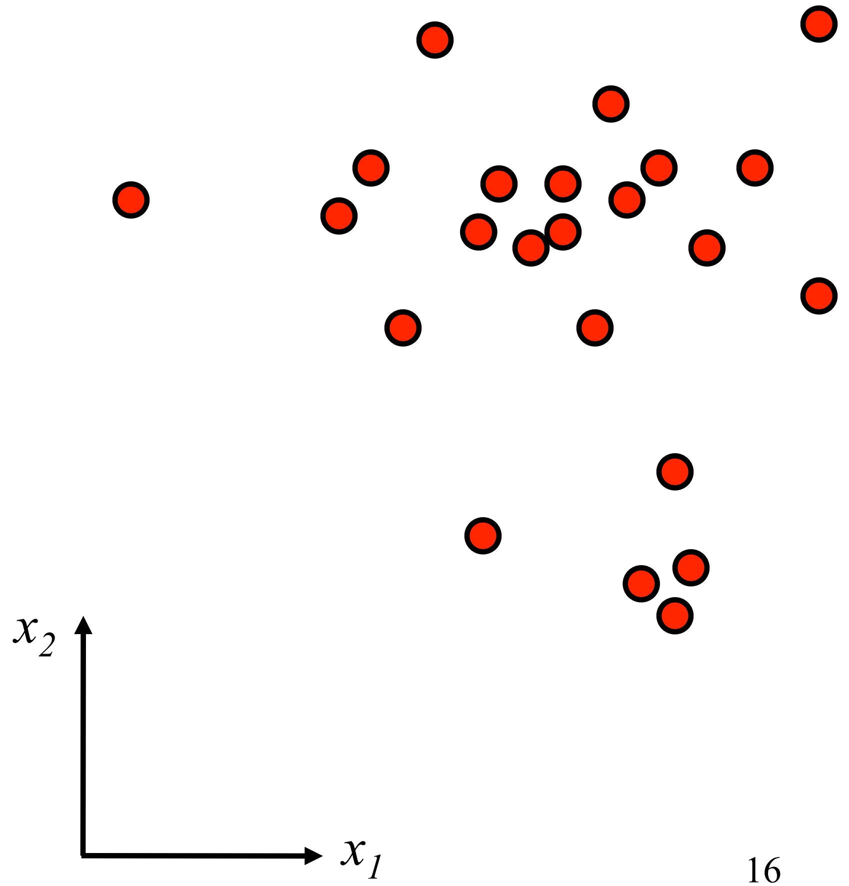
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \eth(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Slide by Ce Liu

Gibbs Sampling and Simulated Annealing

- Gibbs sampling:
  - A way to generate random samples from a (potentially very complicated) probability distribution.
  - Fix all dimensions except one. Draw from the resulting 1-d conditional distribution. Repeat for all dimensions, and repeat many times
- Simulated annealing:
  - A schedule for modifying the probability distribution so that, at "zero temperature", you draw samples only from the MAP solution.

$$P(x) = \frac{1}{Z} \exp(-E(x)/kT)$$

Reference: Geman and Geman, IEEE PAMI 1984.

Gibbs sampling and simulated annealing

Gibbs sampling and simulated annealing

Simulated annealing as you gradually lower the "temperature" of the probability distribution ultimately giving zero probability to all but the MAP estimate.

Gibbs sampling and simulated annealing

Simulated annealing as you gradually lower the "temperature" of the probability distribution ultimately giving zero probability to all but the MAP estimate.

What's good about it: finds global MAP solution.

Gibbs sampling and simulated annealing

Simulated annealing as you gradually lower the "temperature" of the probability distribution ultimately giving zero probability to all but the MAP estimate.

<u>What's good about it</u>: finds global MAP solution.

<u>What's bad about it</u>: takes forever. Gibbs sampling is in the inner loop…

Gibbs sampling and simulated annealing

Gibbs sampling and simulated annealing

So you can find the mean value (MMSE estimate) of a variable by doing Gibbs sampling and averaging over the values that come out of your sampler.

Gibbs sampling and simulated annealing

So you can find the mean value (MMSE estimate) of a variable by doing Gibbs sampling and averaging over the values that come out of your sampler.

You can find the MAP estimate of a variable by doing Gibbs sampling and gradually lowering the temperature parameter to zero.

Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
    - Application example--stereo
  - Variational methods
    - Application example—blind deconvolution
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Iterated conditional modes

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors
  - Find the mode

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors
  - Find the mode
  - Repeat.

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors
  - Find the mode
  - Repeat.
- Compare with Gibbs sampling…

Described in: Winkler, 1995. Introduced by Besag in 1986.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors
  - Find the mode
  - Repeat.
- Compare with Gibbs sampling…
- Very small region over which it's a local maximum

Described in: Winkler, 1995. Introduced by Besag in 1986.

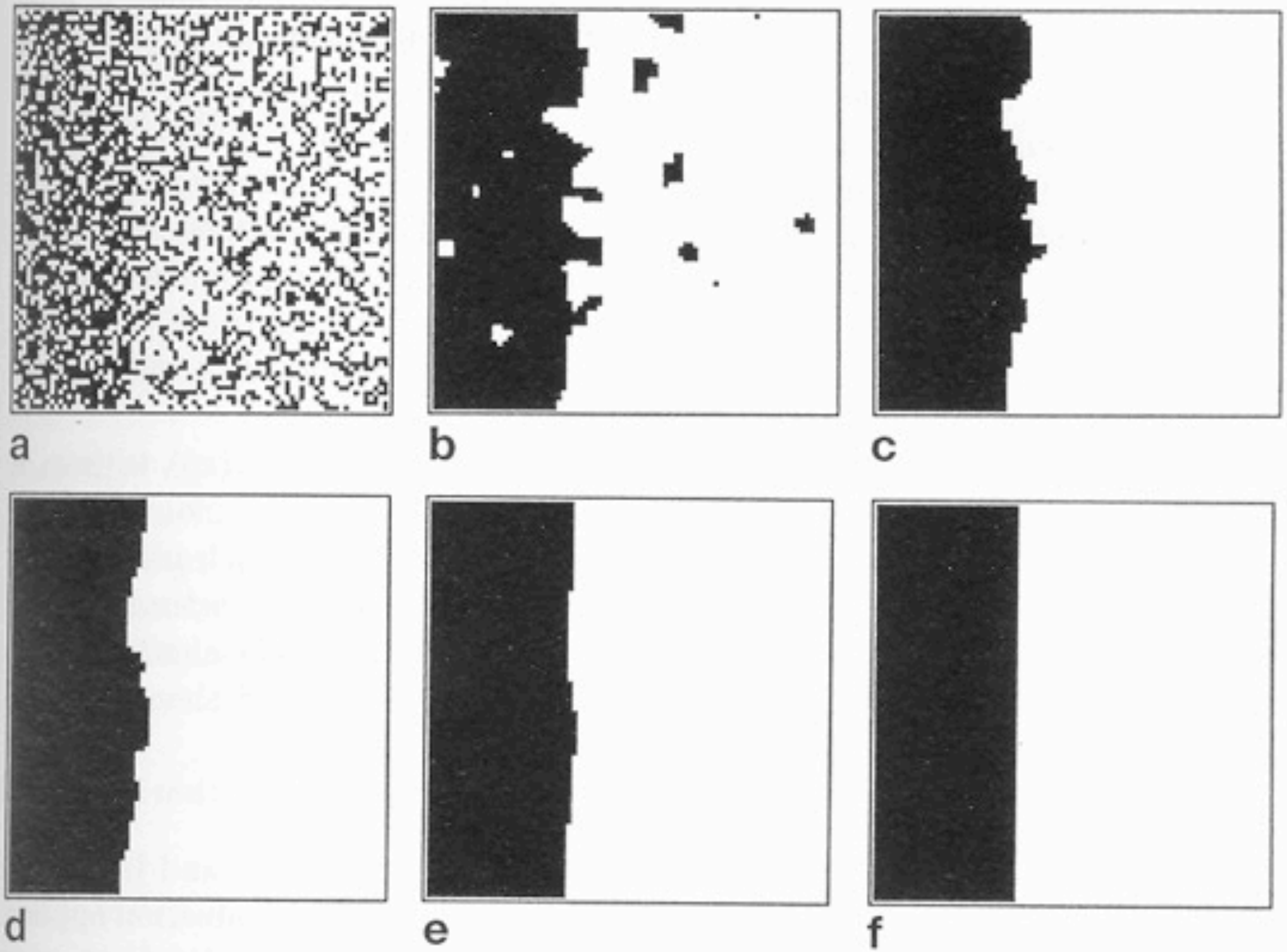


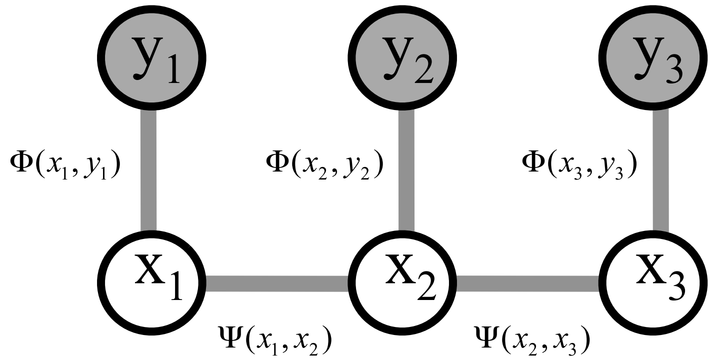
Fig. 6.2. Various steps of ICM

Winkler, 1995

Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Loopy belief propagation
    - Application example—super-resolution
  - Graph cuts
  - Variational methods
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Derivation of belief propagation



$$P(x_1 \mid y_1, y_2, y_3) = k \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

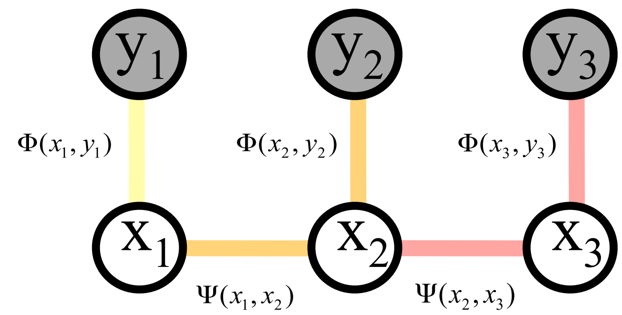
The posterior factorizes

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$= \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1, y_1)$$

$$\Phi(x_2, y_2)\ \Psi(x_1, x_2)$$

$$\Phi(x_3, y_3)\ \Psi(x_2, x_3)$$



Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; \Phi(x_1, y_1)$$

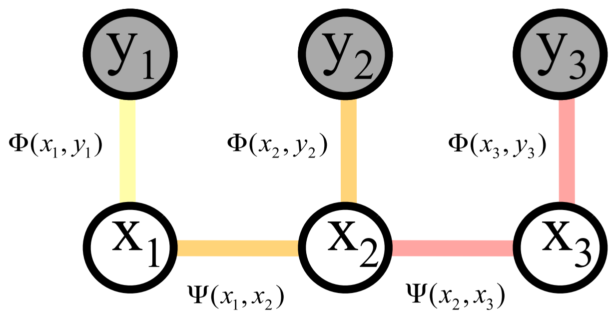
$$\Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \Phi(x_1, y_1)$$

$$\underset{x_2}{\text{sum}} \; \Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\underset{x_3}{\text{sum}} \; \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$



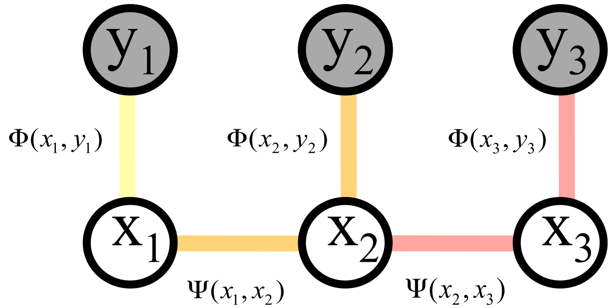
Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\, \Phi(x_1, y_1)$$

$$\underset{x_2}{\text{sum}}\, \Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\underset{x_3}{\text{sum}}\, \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

$$M_1^2(x_1) = \underset{x_2}{\text{sum}}\; \Psi(x_1, x_2)\; \Phi(x_2, y_2)\; M_2^3(x_2)$$



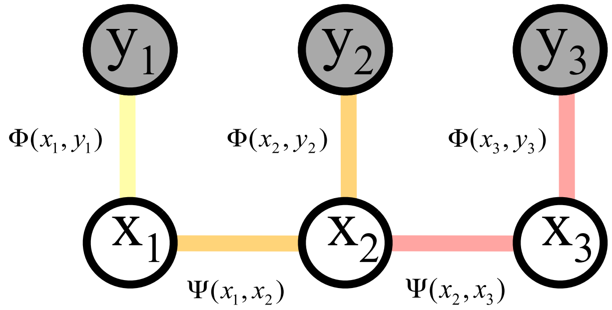
Propagation rules

$$x_{1MMSE} = \operatorname*{mean}_{x_1} \Phi(x_1, y_1)$$

$$\operatorname*{sum}_{x_2} \Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\operatorname*{sum}_{x_3} \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

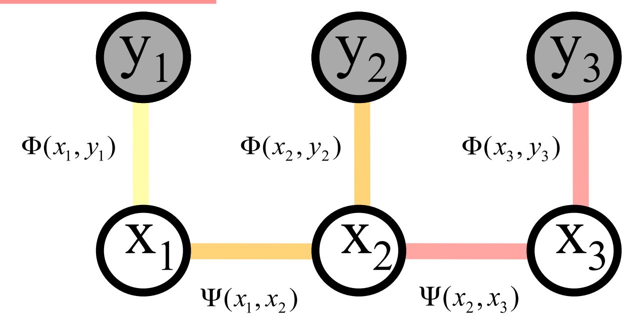
$$M_1^2(x_1) = \operatorname*{sum}_{x_2} \Psi(x_1, x_2)\ \Phi(x_2, y_2)\ M_2^3(x_2)$$



Propagation rules

$$x_{1MMSE} = \operatorname*{mean}_{x_1} \Phi(x_1, y_1) \operatorname*{sum}_{x_2} \Phi(x_2, y_2)\, \Psi(x_1, x_2) \boxed{\operatorname*{sum}_{x_3} \Phi(x_3, y_3)\, \Psi(x_2, x_3)}$$

$$M_1^2(x_1) = \operatorname*{sum}_{x_2} \Psi(x_1, x_2)\; \Phi(x_2, y_2) \boxed{M_2^3(x_2)}$$



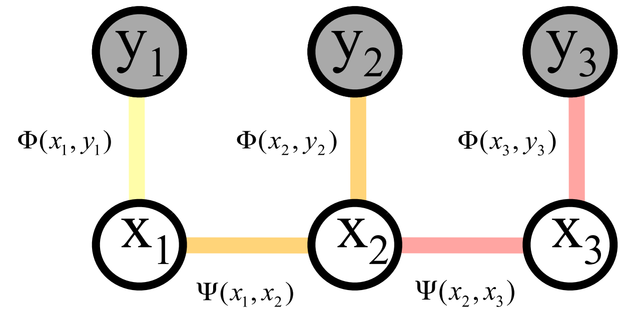
Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\, \Phi(x_1, y_1)$$

$$\underset{x_2}{\text{sum}}\, \Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\underset{x_3}{\text{sum}}\, \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

$$\boxed{M_1^2(x_1)} = \underset{x_2}{\text{sum}}\; \Psi(x_1, x_2)\; \Phi(x_2, y_2)\, \boxed{M_2^3(x_2)}$$

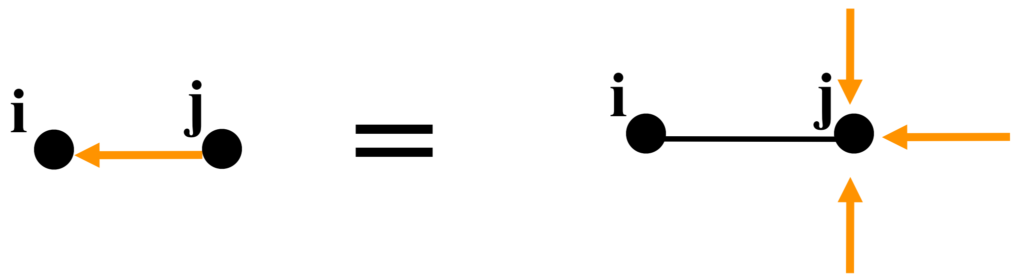


Belief propagation messages

<u>A message</u>: can be thought of as a set of weights on each of your possible states

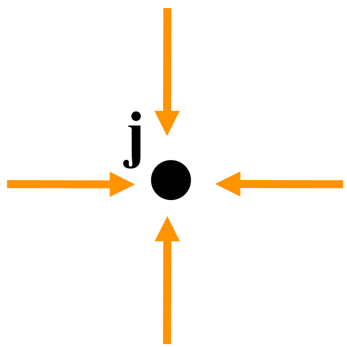
<u>To send a message</u>: Multiply together all the incoming messages, except from the node you're sending to, then multiply by the compatibility matrix and marginalize over the sender's states.

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$



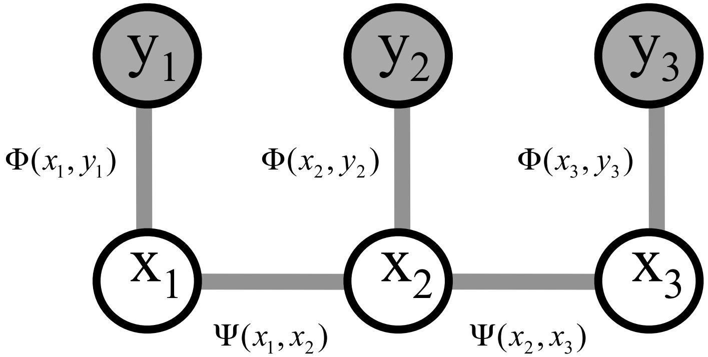
Beliefs

<u>To find a node's beliefs</u>: Multiply together all the messages coming in to that node.



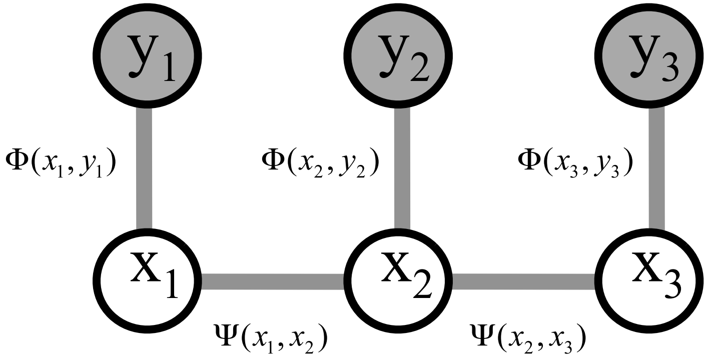
$$b_j(x_j) = \prod_{k \in N(j)} M_j^k(x_j)$$

Max-product belief propagation



$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

Max-product belief propagation

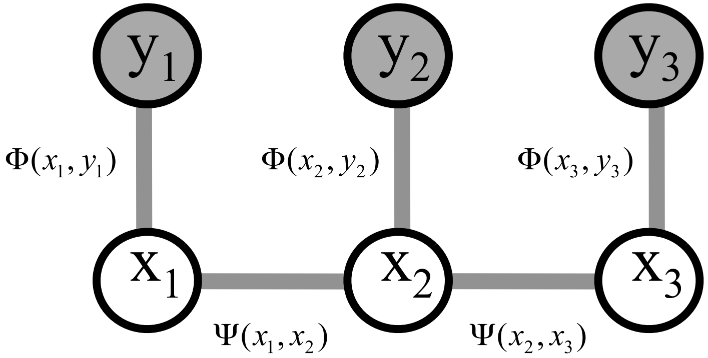


$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

message update equation for sum-product algo

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$

Max-product belief propagation



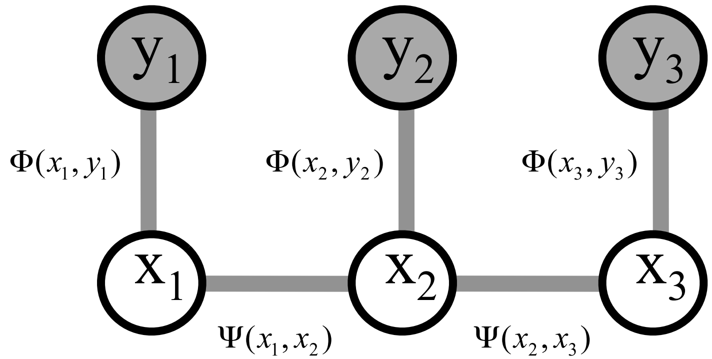
$$x_{1MMSE} = \underset{x_1}{\mathrm{mean}} \; \underset{x_2}{\mathrm{sum}} \; \underset{x_3}{\mathrm{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

message update equation for sum-product algo

$$M_i^j(x_i) = \sum_{x_j} \psi_{\mathrm{ij}}(x_i, x_j) \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$

$$x_{1MAP} = \underset{\mathrm{x}_1}{\mathrm{argmax}} \; \underset{x_2}{\max} \; \underset{x_2}{\max} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

Max-product belief propagation



$$x_{1MMSE} = \underset{x_1}{\text{mean}} \ \underset{x_2}{\text{sum}} \ \underset{x_3}{\text{sum}} \ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

message update equation for sum-product algo

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$

$$x_{1MAP} = \underset{x_1}{\text{argmax}} \ \underset{x_2}{\max} \ \underset{x_2}{\max} \ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

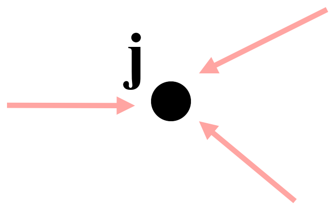
message update equation for max-product algo.

$$M_i^j(x_i) = \max_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$

Optimal solution in a chain or tree: Belief Propagation

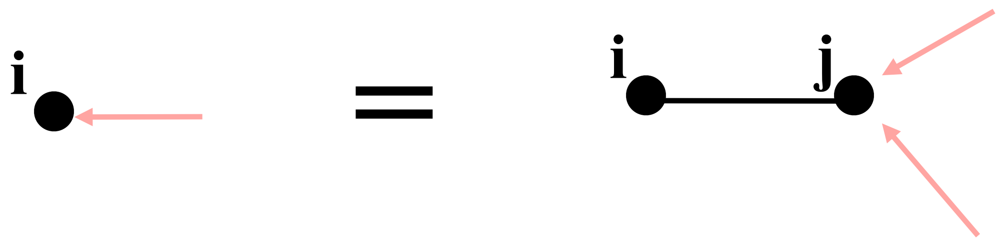
- "Do the right thing" Bayesian algorithm.
- For Gaussian random variables over time: Kalman filter.
- For hidden Markov models: forward/backward algorithm (and MAP variant is Viterbi).

Belief, and message update rules are just local operations, and can be run whether or not the network has loops



$$b_j(x_j) = \prod_{k \in N(j)} M_j^k(x_j)$$

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \setminus i} M_j^k(x_j)$$



Justification for running belief propagation in networks with loops

- Experimental results:
  - Comparison of methods Szeliski et al. 2008 http://vision.middlebury.edu/MRF/
  - Error-correcting codes Kschischang and Frey, 1998; McEliece et al., 1998
  - Vision applications Freeman and Pasztor, 1999; Frey, 2000
- Theoretical results:
  - For Gaussian processes, means are correct. Weiss and Freeman, 1999
  - Large neighborhood local maximum for MAP. Weiss and Freeman, 2000
  - Equivalent to Bethe approx. in statistical physics. Yedidia, Freeman, and Weiss, 2000
  - Tree-weighted reparameterization Wainwright, Willsky, Jaakkola, 2001

Show program comparing some methods on a simple MRF

testMRF.m

Outline of MRF section

- Inference in MRF’s.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
  - Variational methods
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Super-resolution

- Image: low resolution image
- Scene: high resolution image

ultimate goal...

Polygon-based graphics images are resolution independent

Pixel-based images are not resolution independent

Polygon-based graphics images are resolution independent

Pixel-based images
are not resolution
independent

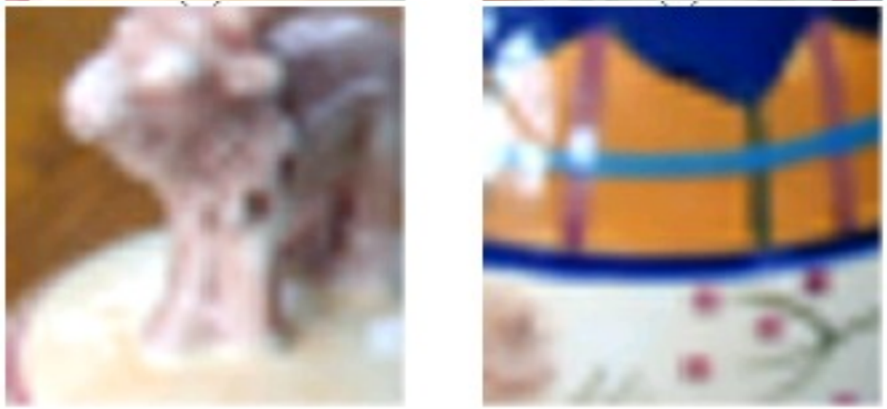
Pixel replication

Polygon-based
graphics
images are
resolution
independent

Pixel-based images are not resolution independent

Pixel replication

Cubic spline

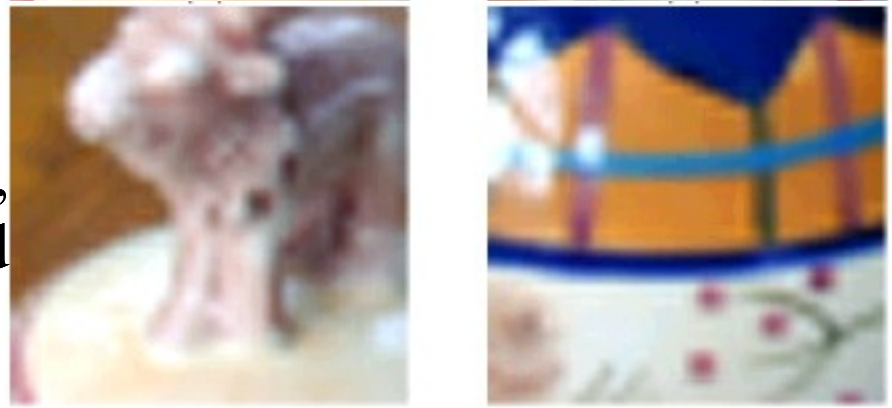


Polygon-based graphics images are resolution independent

Pixel-based images are not resolution independent

Pixel replication

Cubic spline, sharpened



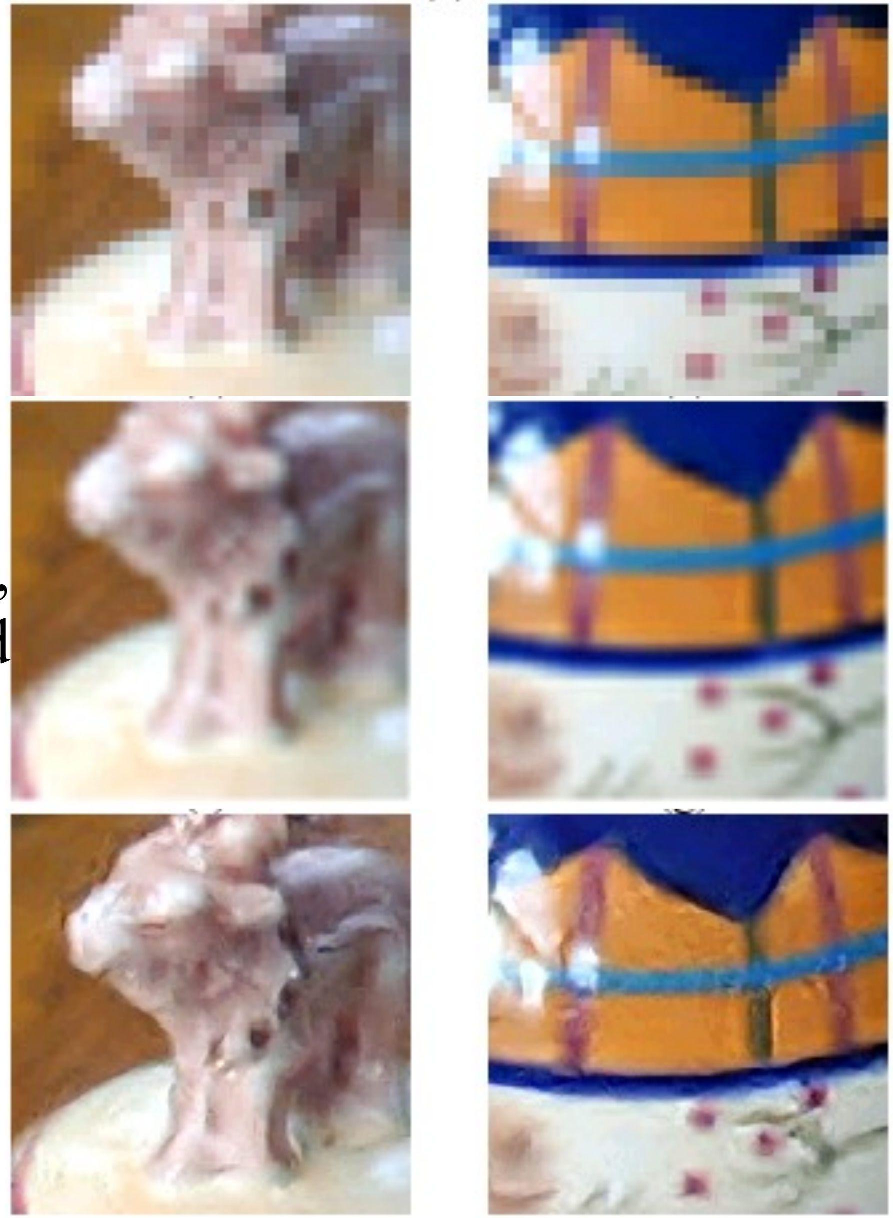
Polygon-based graphics images are resolution independent

Pixel-based images are not resolution independent

Pixel replication

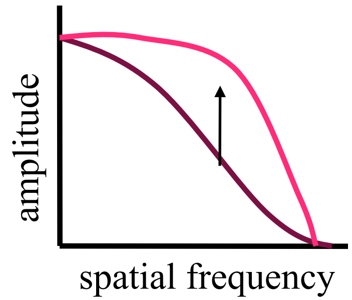
Cubic spline, sharpened

Training-based super-resolution



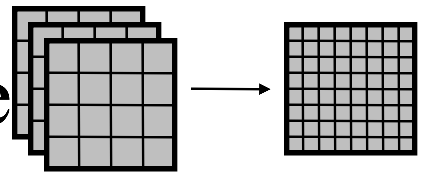
Polygon-based graphics images are resolution independent

3 approaches to perceptual sharpening

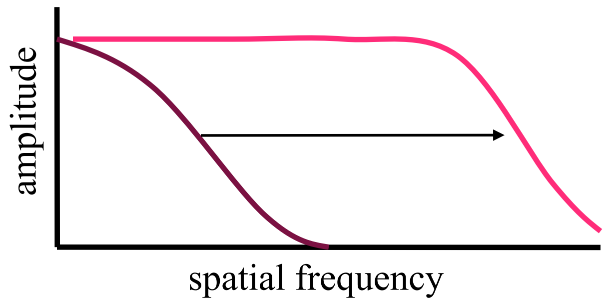


(1) Sharpening; boost existing high frequencies.

(2) Use multiple frames to obtain higher sampling rate in a still frame



(3) Estimate high frequencies not present in image, although implicitly defined.



In this talk, we focus on (3), which we'll call "super-resolution".

Super-resolution: other approaches

- Schultz and Stevenson, 1994
- Pentland and Horowitz, 1993
- fractal image compression (Polvere, 1998; Iterated Systems)
- astronomical image processing (eg. Gull and Daniell, 1978; "pixons" http://casswww.ucsd.edu/puetter.html)
- Follow-on: Jianchao Yang, John Wright, Thomas S. Huang, Yi Ma: Image super-resolution as sparse representation of raw image patches. CVPR 2008

Training images, **~100,000 image/scene patch pairs**

Images from two Corel database categories:
"giraffes" and "urban skyline".

Do a first interpolation

Zoomed low-resolution

Low-resolution

Zoomed low-resolution

Full frequency original

Low-resolution

Representation

Zoomed low-freq.

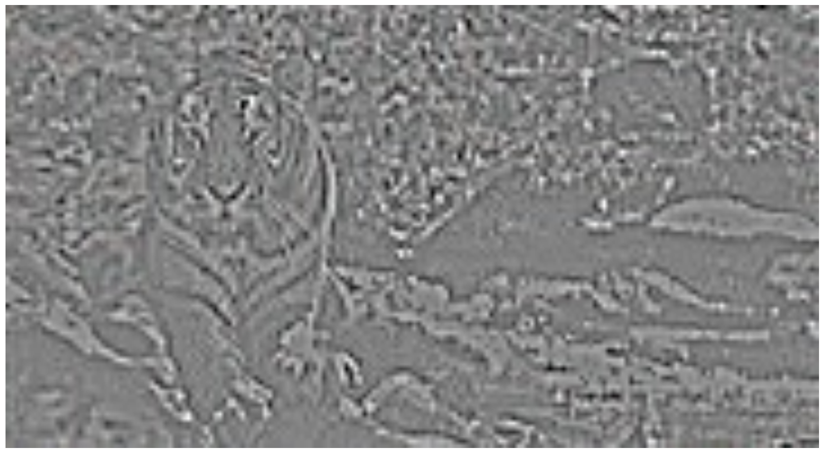
Full freq. original

Representation

Zoomed low-freq.

Full freq. original

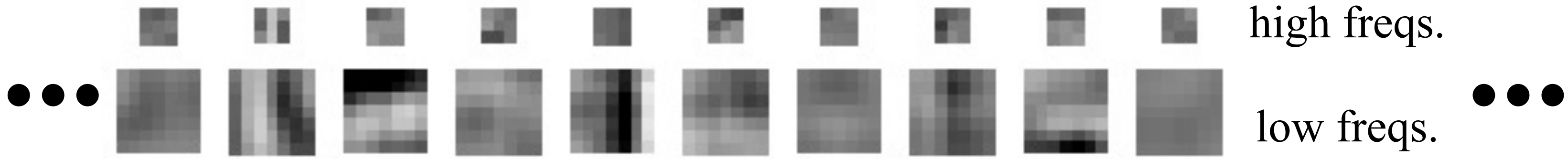
Low-band input
(contrast normalized,
PCA fitted)



True high freqs

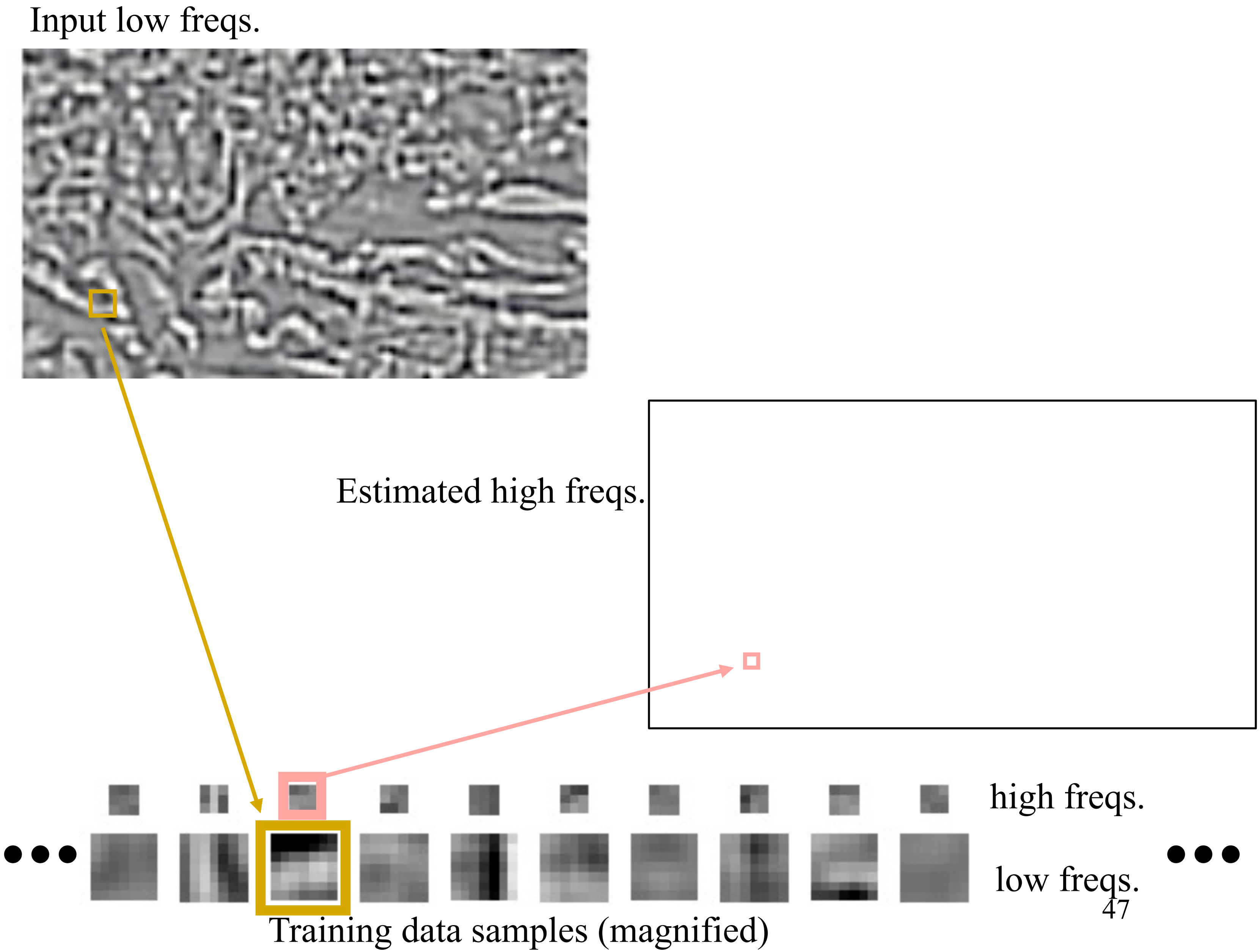
(to minimize the complexity of the relationships we have to learn,
we remove the lowest frequencies from the input image,
and normalize the local contrast level).

Gather ~100,000 patches

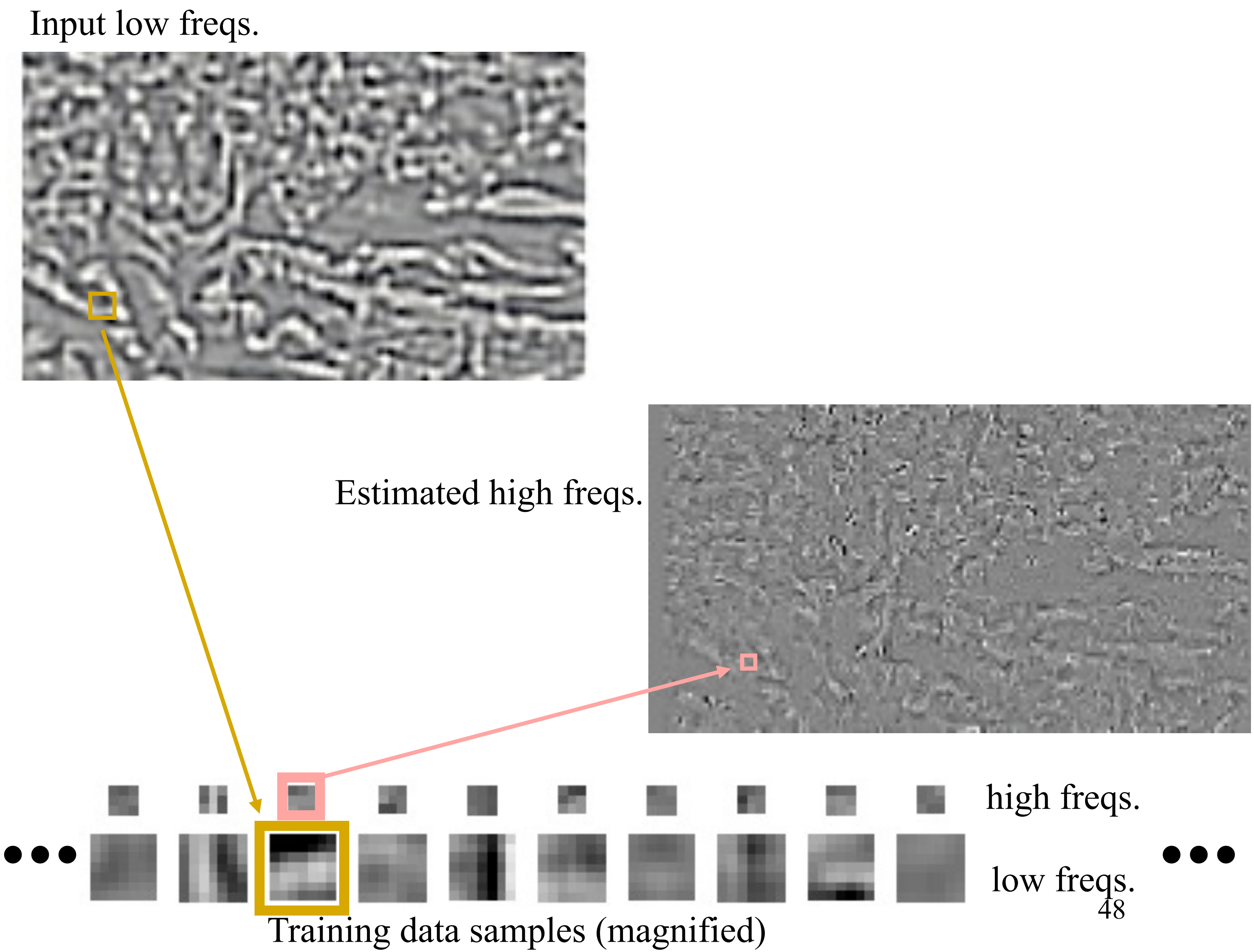


Training data samples (magnified)

Nearest neighbor estimate

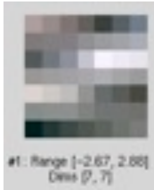


Nearest neighbor estimate

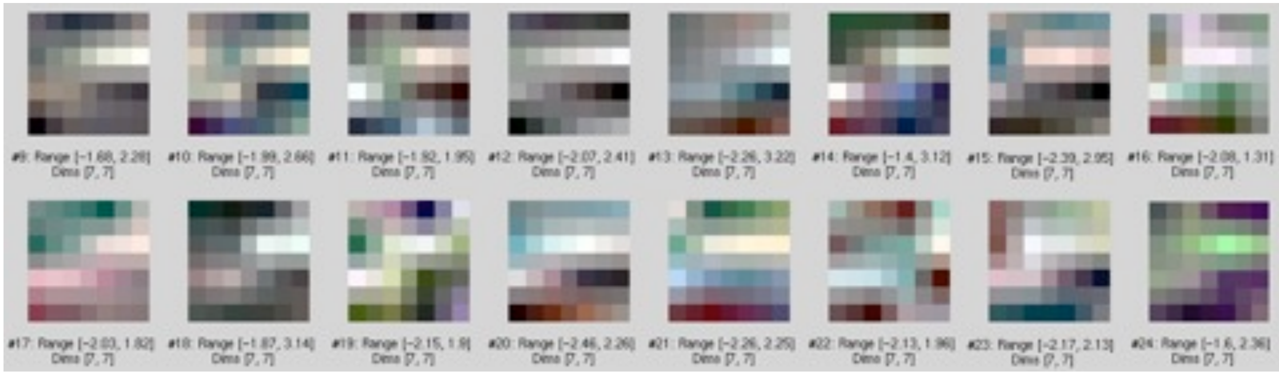


Example: input image patch, and closest matches from database

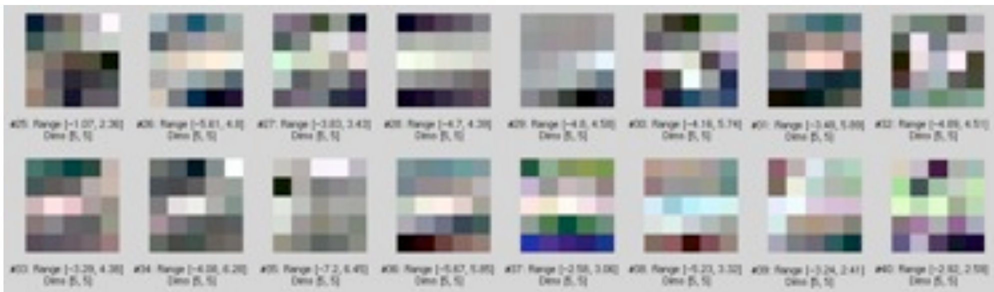
Input patch



Closest image patches from database



Corresponding high-resolution patches from database



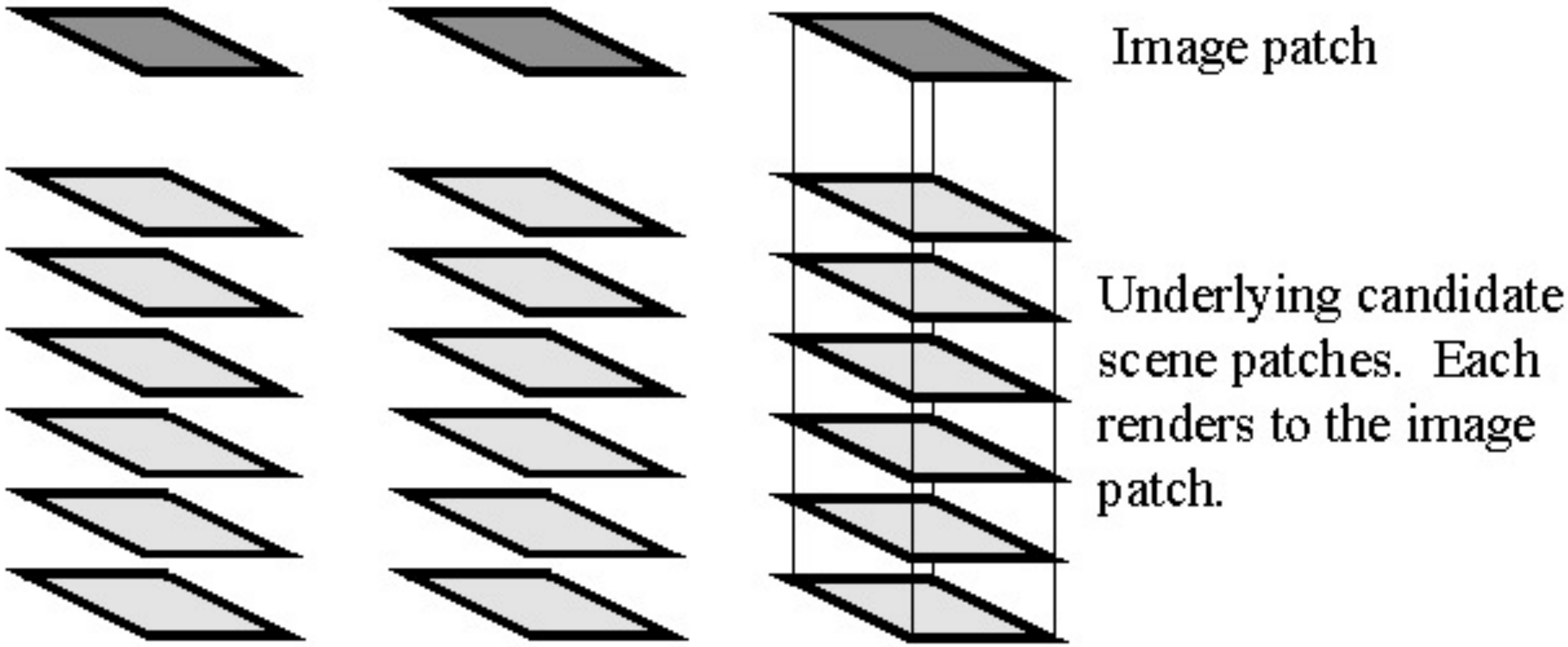
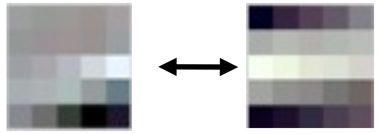


Image patch
Underlying candidate scene patches. Each renders to the image patch.

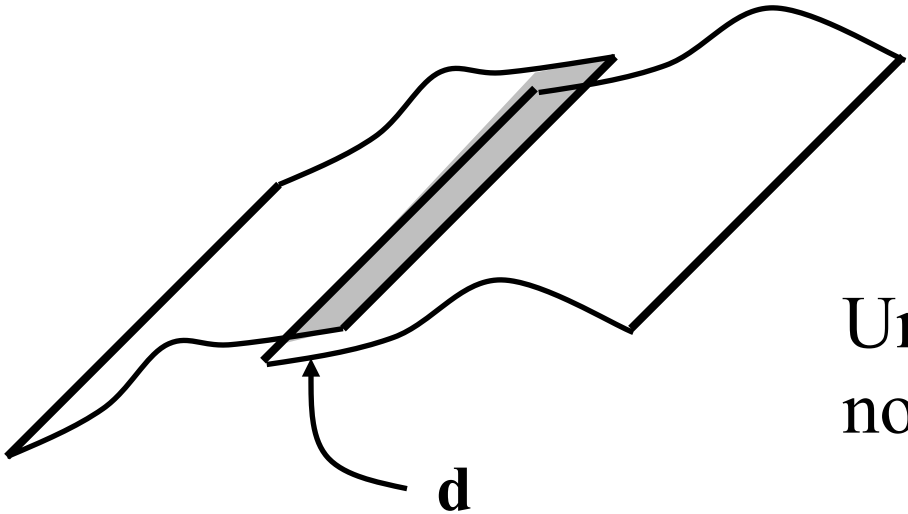
Scene-scene compatibility function, Ψ

(x_i, x_j)



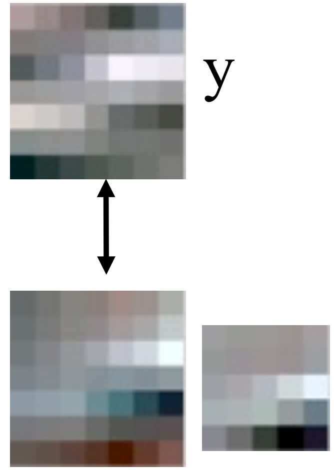
Assume overlapped regions, d, of hi-res.
patches differ by Gaussian observation noise:

$$\Psi(x_i, x_j) = \exp^{-|d_i - d_j|^2 / 2\sigma^2}$$



Uniqueness constraint,
not smoothness.

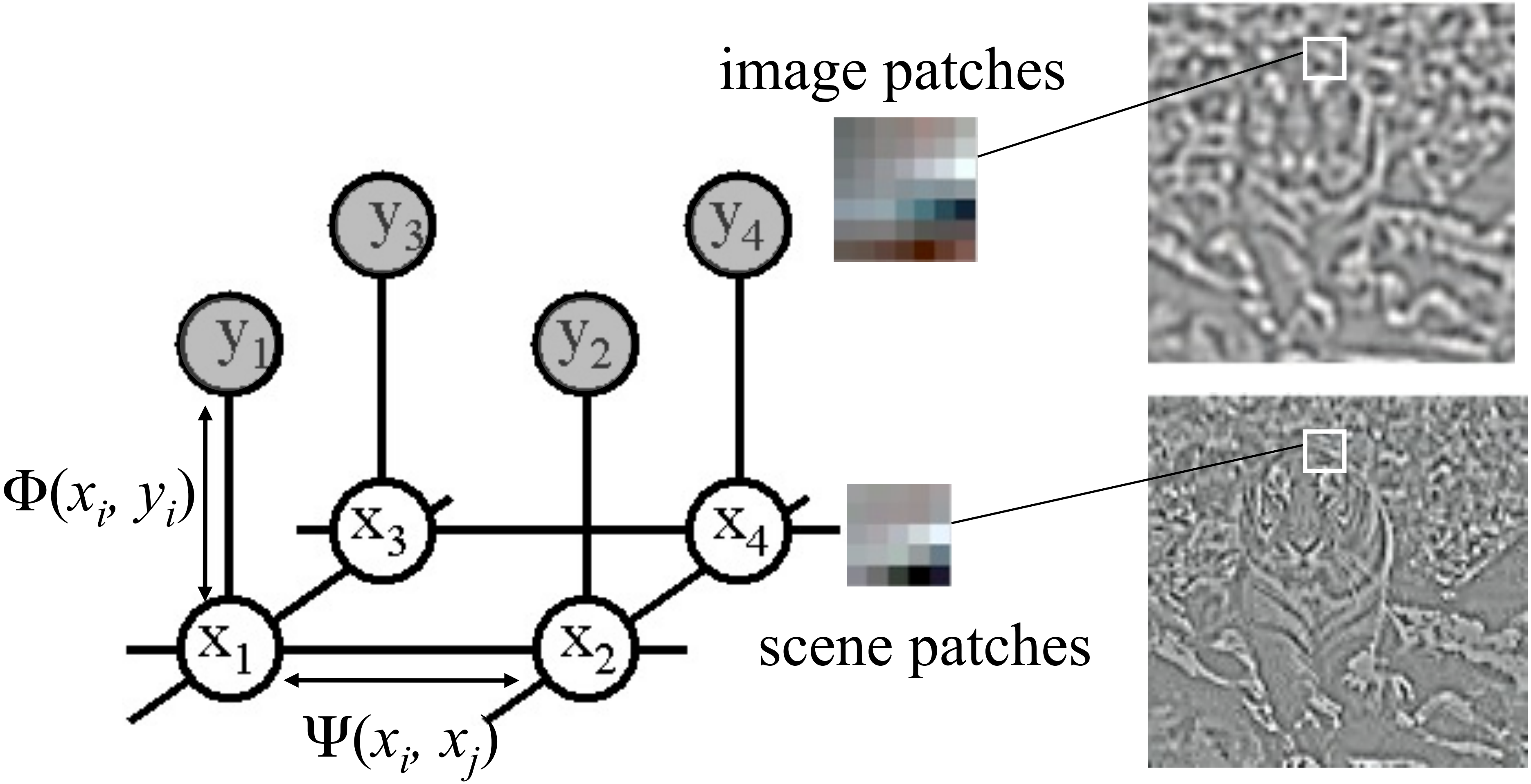
Image-scene compatibility function, $\Phi(x_i, y_i)$



Assume Gaussian noise takes you from observed image patch to synthetic sample:

$$\Phi(x_i, y_i) = \exp^{-|y_i - y(x_i)|^2 / 2\sigma^2}$$

Markov network



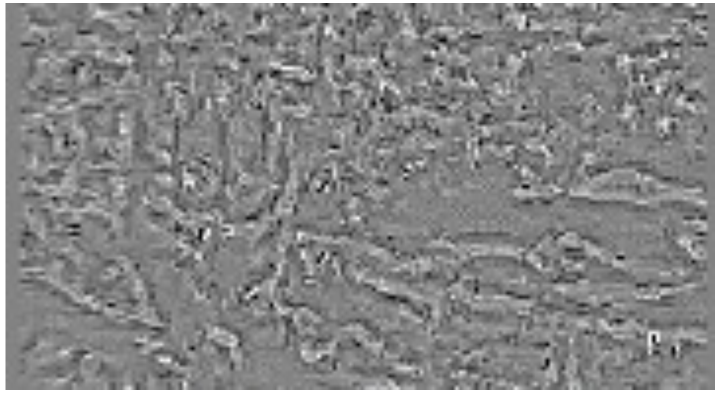
Belief Propagation

Input

Belief Propagation

After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

Input

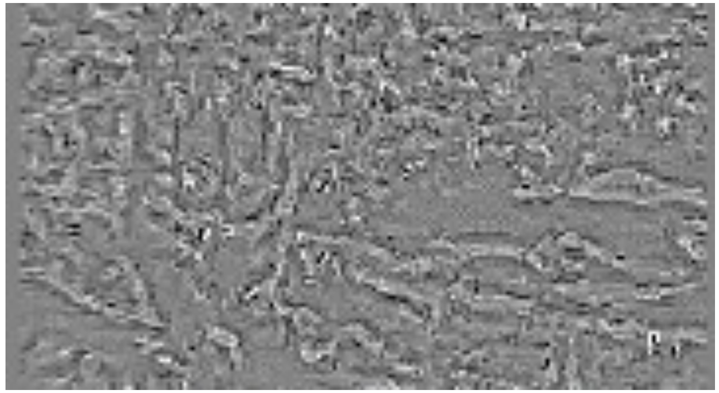


Iter. 0

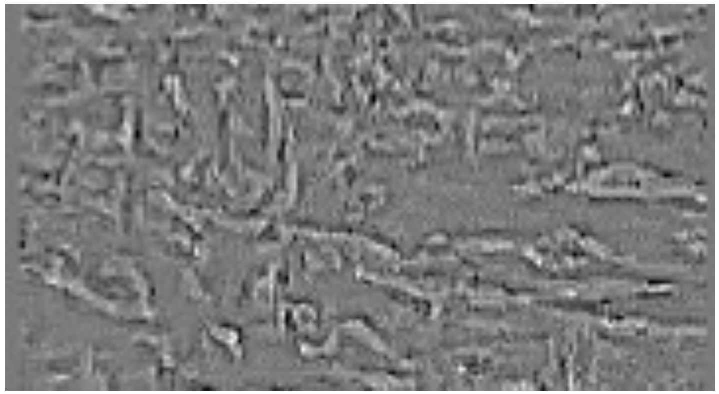
Belief Propagation

After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

Input



Iter. 0

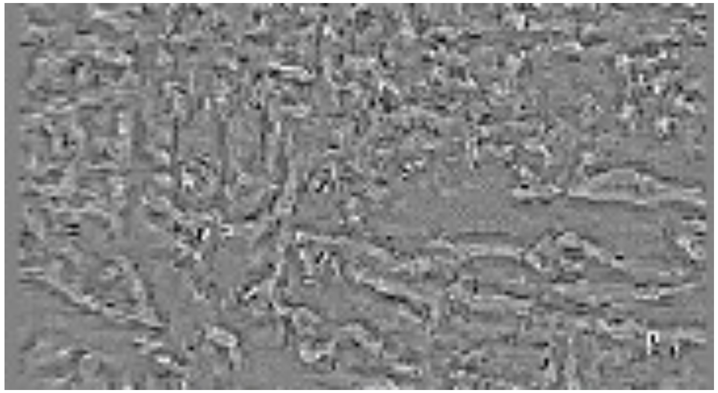


Iter. 1

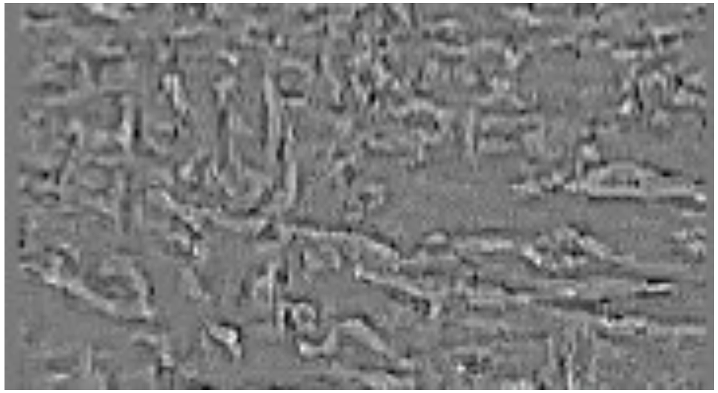
Belief Propagation

After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

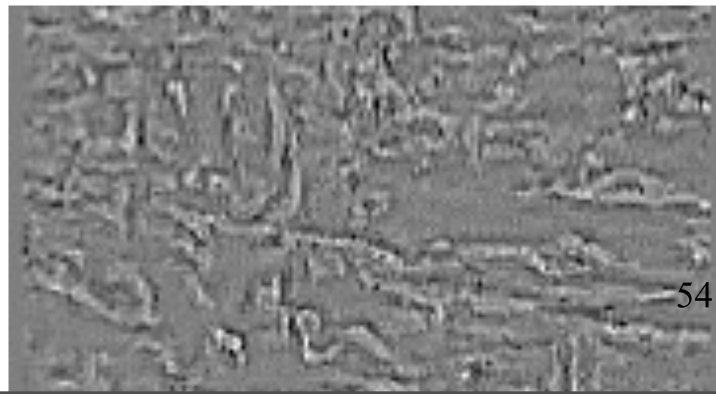
Input



Iter. 0



Iter. 1



Iter. 3

Zooming 2 octaves

We apply the super-resolution algorithm recursively, zooming up 2 powers of 2, or a factor of 4 in each dimension.

85 x 51 input

Zooming 2 octaves

We apply the super-resolution algorithm recursively, zooming up 2 powers of 2, or a factor of 4 in each dimension.

85 x 51 input

Cubic spline zoom to 340x204

Zooming 2 octaves

We apply the super-resolution algorithm recursively, zooming up 2 powers of 2, or a factor of 4 in each dimension.

85 x 51 input

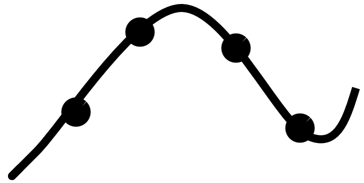
Cubic spline zoom to 340x204

Max. likelihood zoom to 340x204

Now we examine the effect of the prior assumptions made about images on the high resolution reconstruction.
First, cubic spline interpolation.

Original
50x58

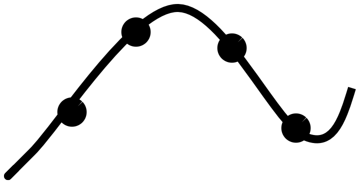
(cubic spline implies thin plate prior)



True
200x232

Original
50x58

(cubic spline implies
thin plate prior)

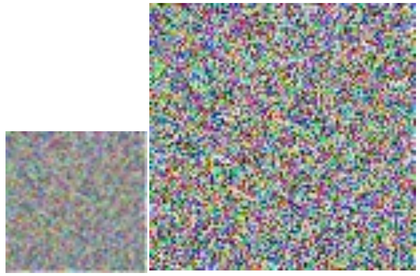


Cubic spline

True
200x232

Next, train the Markov network algorithm on a world of random noise images.

Original
50x58

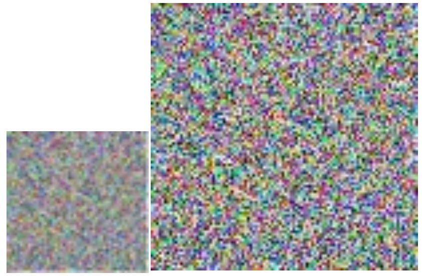


Training images

True

The algorithm learns that, in such a world, we add random noise when zoom to a higher resolution.

Original 50x58



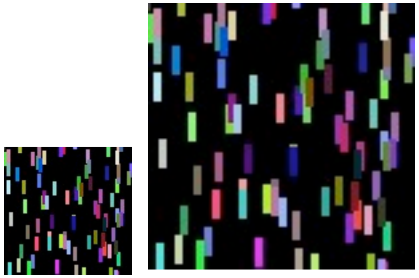
Training images

Markov network

True

Next, train on a world of vertically oriented rectangles.

Original
50x58

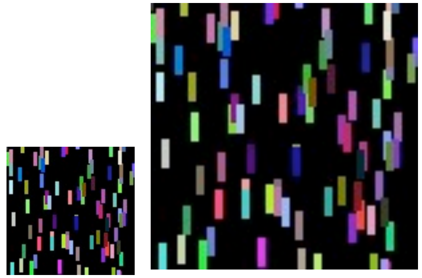


Training images

True

The Markov network algorithm hallucinates those vertical rectangles that it was trained on.

Original 50x58



Training images

Markov network

True

Now train on a generic collection of images.

Original
50x58

Training images

True

The algorithm makes a reasonable guess at the high resolution image, based on its training images.

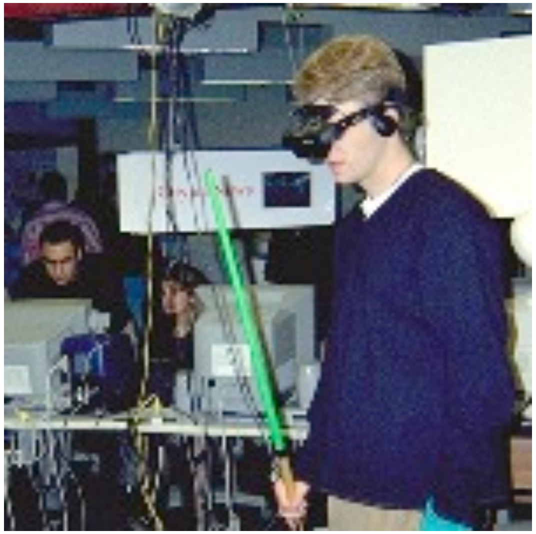
Original
50x58

Training images

Markov
network

True

Generic training images

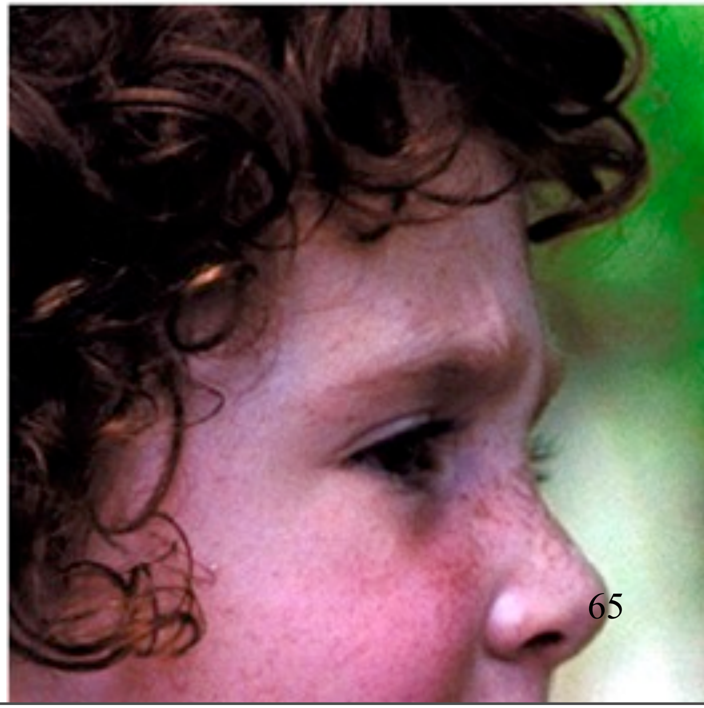


Next, train on a generic set of training images. Using the same camera as for the test image, but a random collection of photographs.

Original
70x70

Cubic
Spline

Markov
net,
training:
generic



True
280x280

Kodak Imaging Science Technology Lab test.

3 test images, 640x480, to be zoomed up by 4 in each dimension.

8 judges, making 2-alternative, forced-choice comparisons.

Algorithms compared

- Bicubic Interpolation
- Mitra's Directional Filter
- Fuzzy Logic Filter
- Vector Quantization
- VISTA

Bicubic spline | Altamira | VISTA

Bicubic spline Altamira VISTA

User preference test results

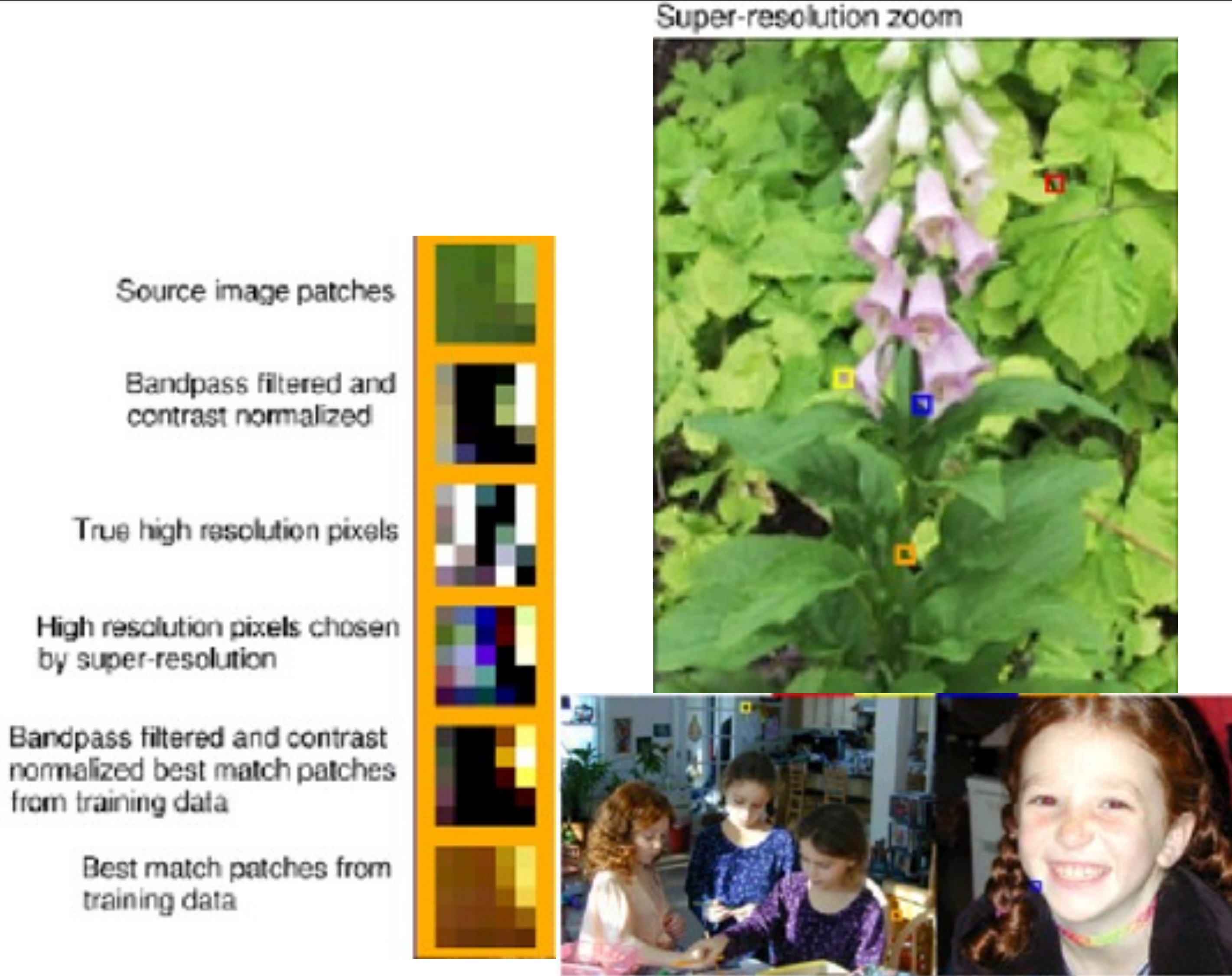
“The observer data indicates that six of the observers ranked Freeman’s algorithm as the most preferred of the five tested algorithms. However the other two observers rank Freeman’s algorithm as the least preferred of all the algorithms….

Freeman’s algorithm produces prints which are by far the sharpest out of the five algorithms. However, this sharpness comes at a price of artifacts (spurious detail that is not present in the original scene). Apparently the two observers who did not prefer Freeman’s algorithm had strong objections to the artifacts. The other observers apparently placed high priority on the high level of sharpness in the images created by Freeman’s algorithm.”

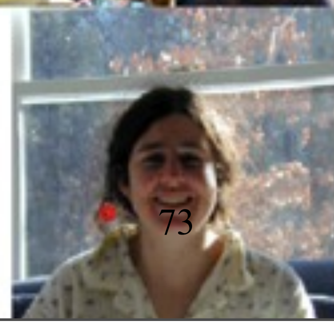

Input
Cubic spline zoom
Super-resolution zoom
True high-resolution image

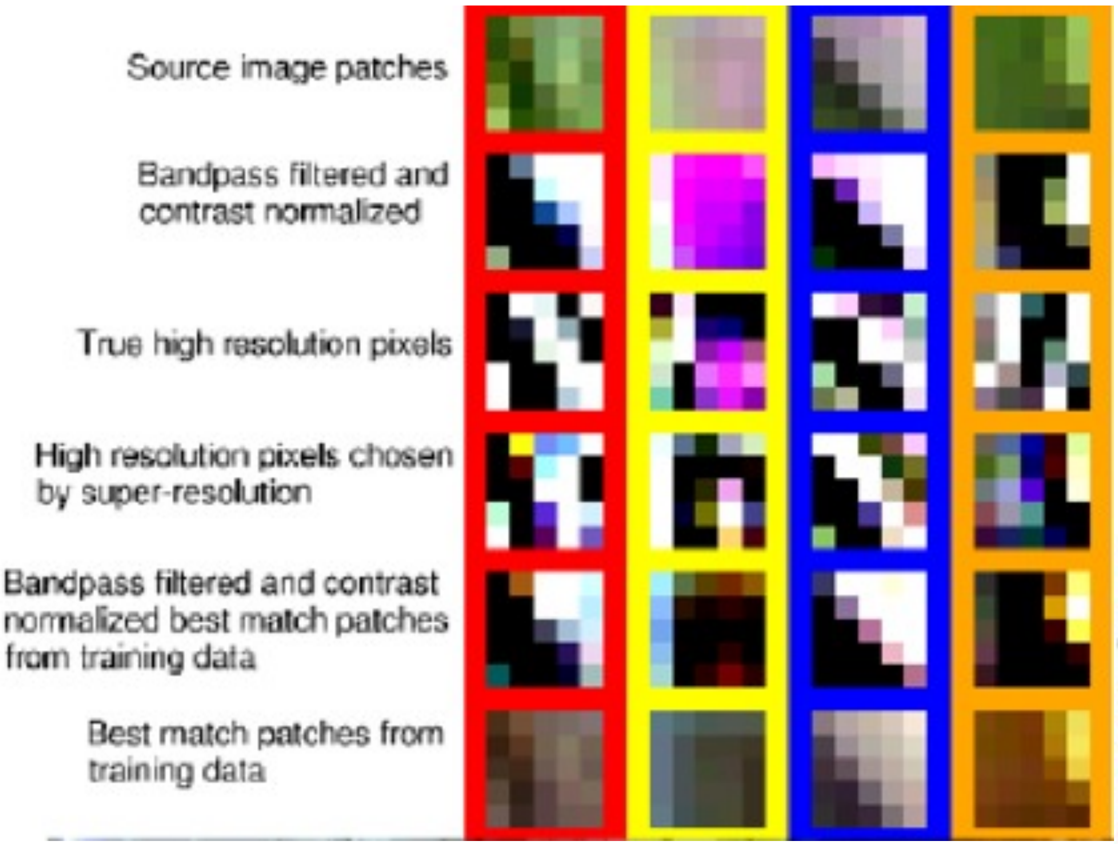
Super-resolution zoom

Training images

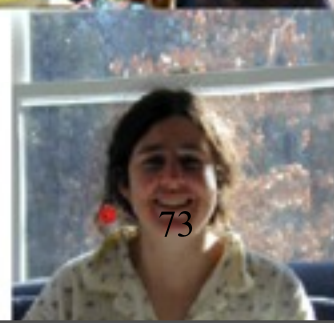


Training images

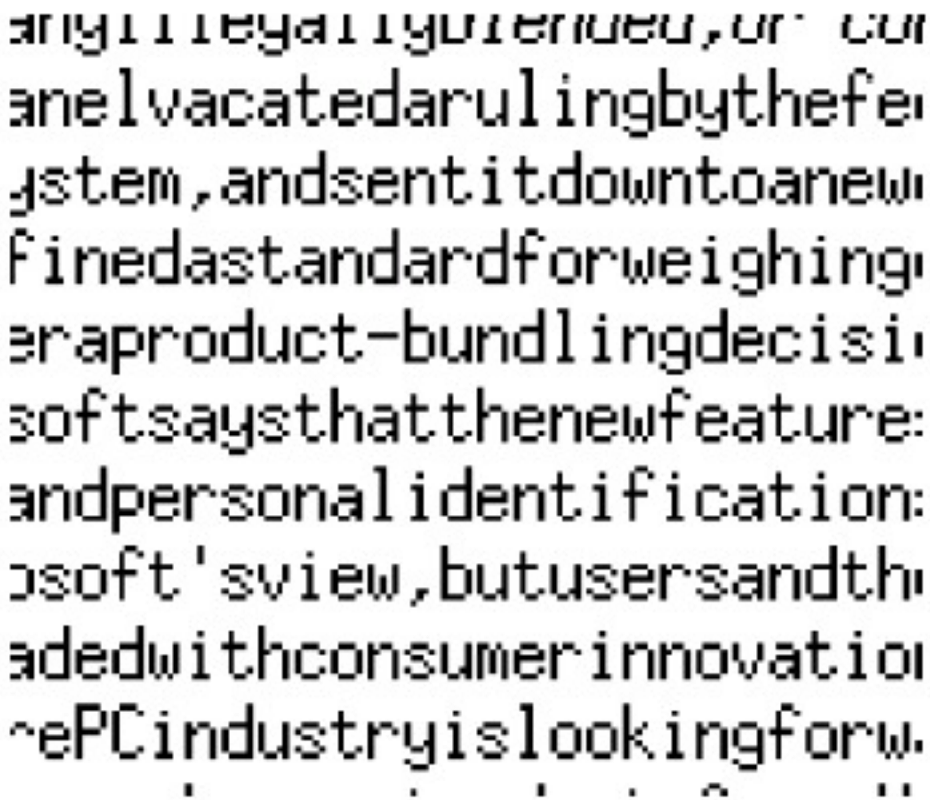




Training images



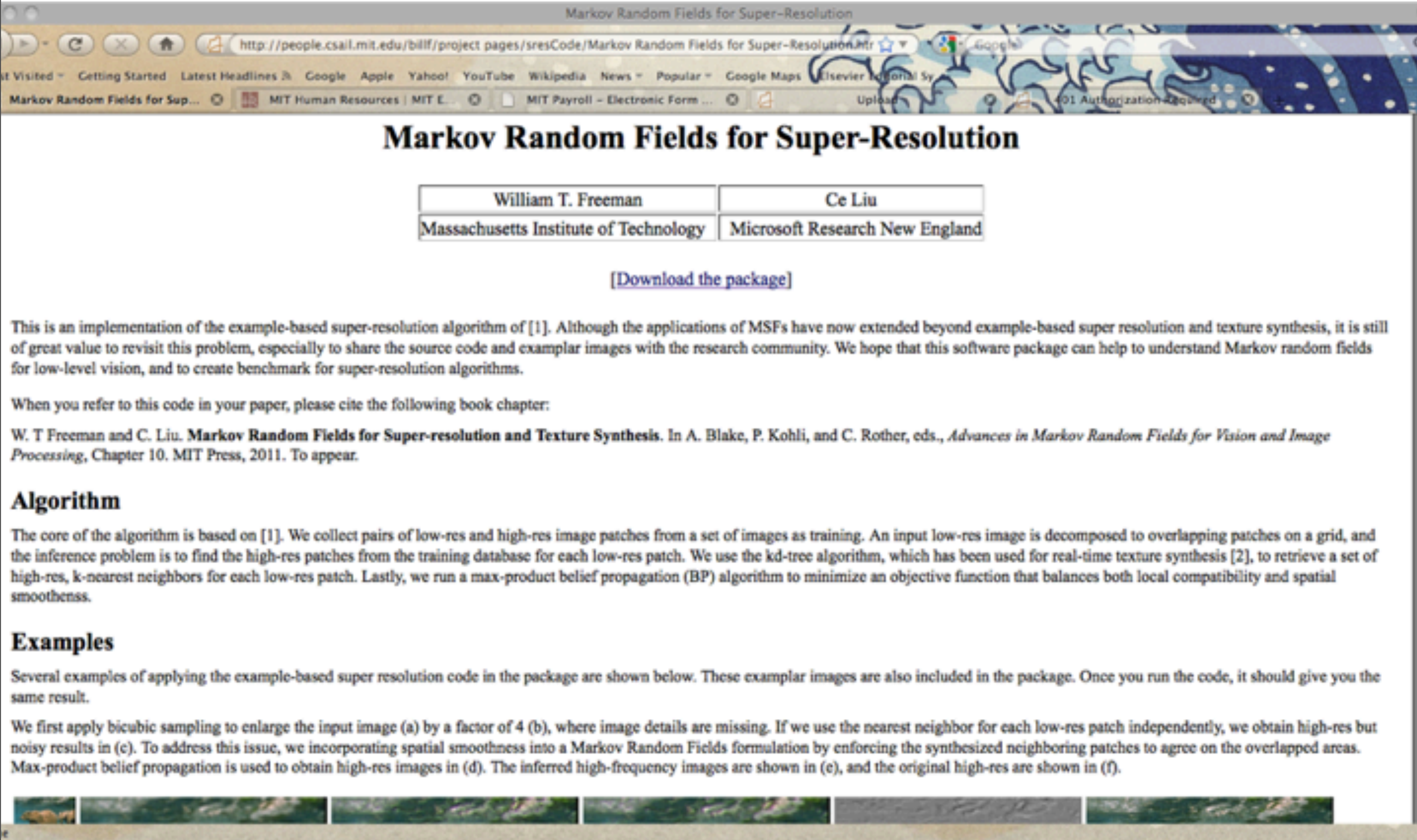
Training image



Processed image

code available online

http://people.csail.mit.edu/billf/project%20pages/sresCode/Markov%20Random%20Fields%20for%20Super-Resolution.html



Markov Random Fields for Super-Resolution

William T. Freeman	Ce Liu
Massachusetts Institute of Technology	Microsoft Research New England

[Download the package]

This is an implementation of the example-based super-resolution algorithm of [1]. Although the applications of MSFs have now extended beyond example-based super resolution and texture synthesis, it is still of great value to revisit this problem, especially to share the source code and examplar images with the research community. We hope that this software package can help to understand Markov random fields for low-level vision, and to create benchmark for super-resolution algorithms.

When you refer to this code in your paper, please cite the following book chapter:

W. T Freeman and C. Liu. **Markov Random Fields for Super-resolution and Texture Synthesis**. In A. Blake, P. Kohli, and C. Rother, eds., *Advances in Markov Random Fields for Vision and Image Processing*, Chapter 10. MIT Press, 2011. To appear.

Algorithm

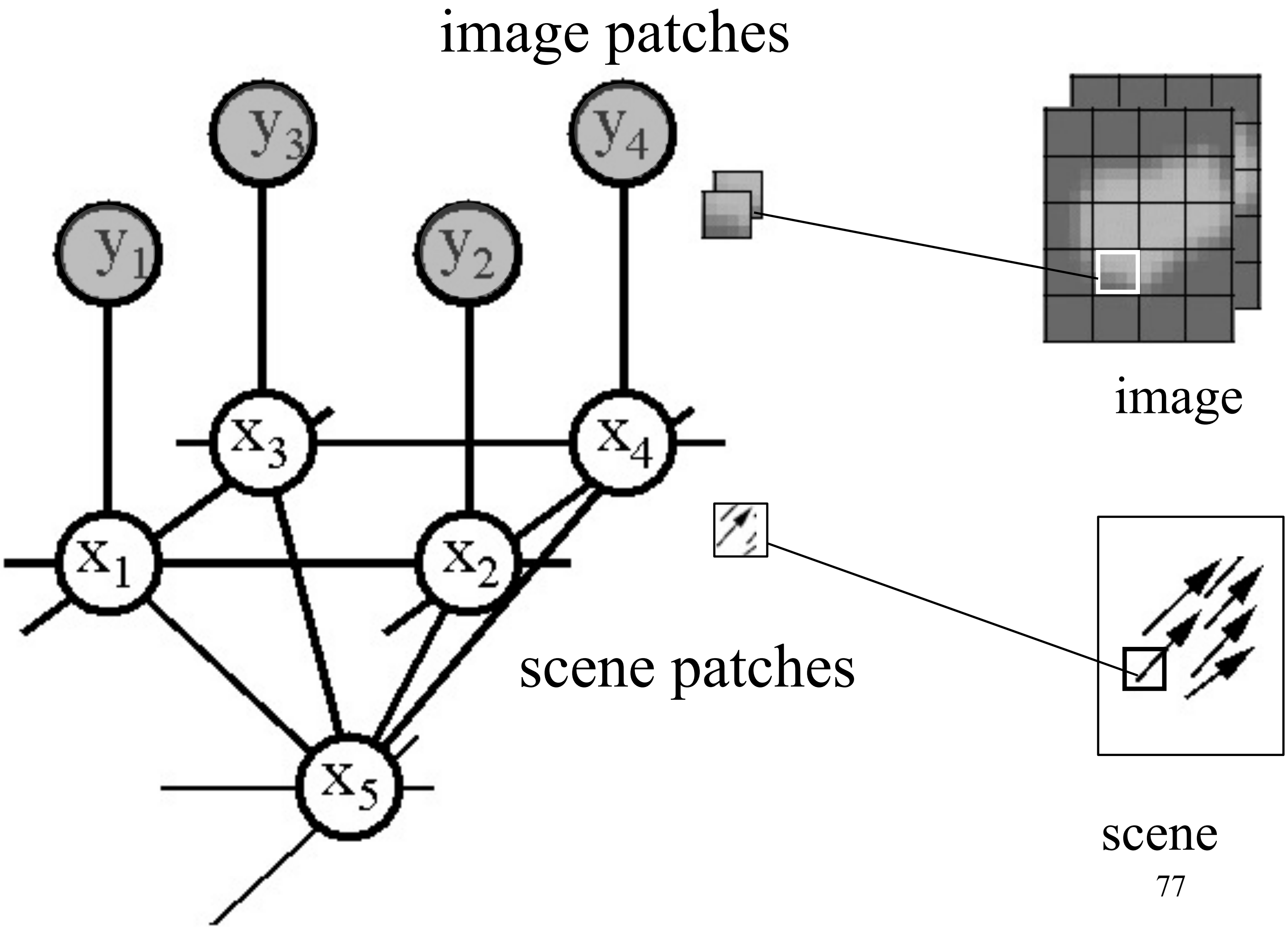
The core of the algorithm is based on [1]. We collect pairs of low-res and high-res image patches from a set of images as training. An input low-res image is decomposed to overlapping patches on a grid, and the inference problem is to find the high-res patches from the training database for each low-res patch. We use the kd-tree algorithm, which has been used for real-time texture synthesis [2], to retrieve a set of high-res, k-nearest neighbors for each low-res patch. Lastly, we run a max-product belief propagation (BP) algorithm to minimize an objective function that balances both local compatibility and spatial smoothenss.

Examples

Several examples of applying the example-based super resolution code in the package are shown below. These examplar images are also included in the package. Once you run the code, it should give you the same result.

We first apply bicubic sampling to enlarge the input image (a) by a factor of 4 (b), where image details are missing. If we use the nearest neighbor for each low-res patch independently, we obtain high-res but noisy results in (c). To address this issue, we incorporate spatial smoothness into a Markov Random Fields formulation by enforcing the synthesized neighboring patches to agree on the overlapped areas. Max-product belief propagation is used to obtain high-res images in (d). The inferred high-frequency images are shown in (e), and the original high-res are shown in (f).

Motion application



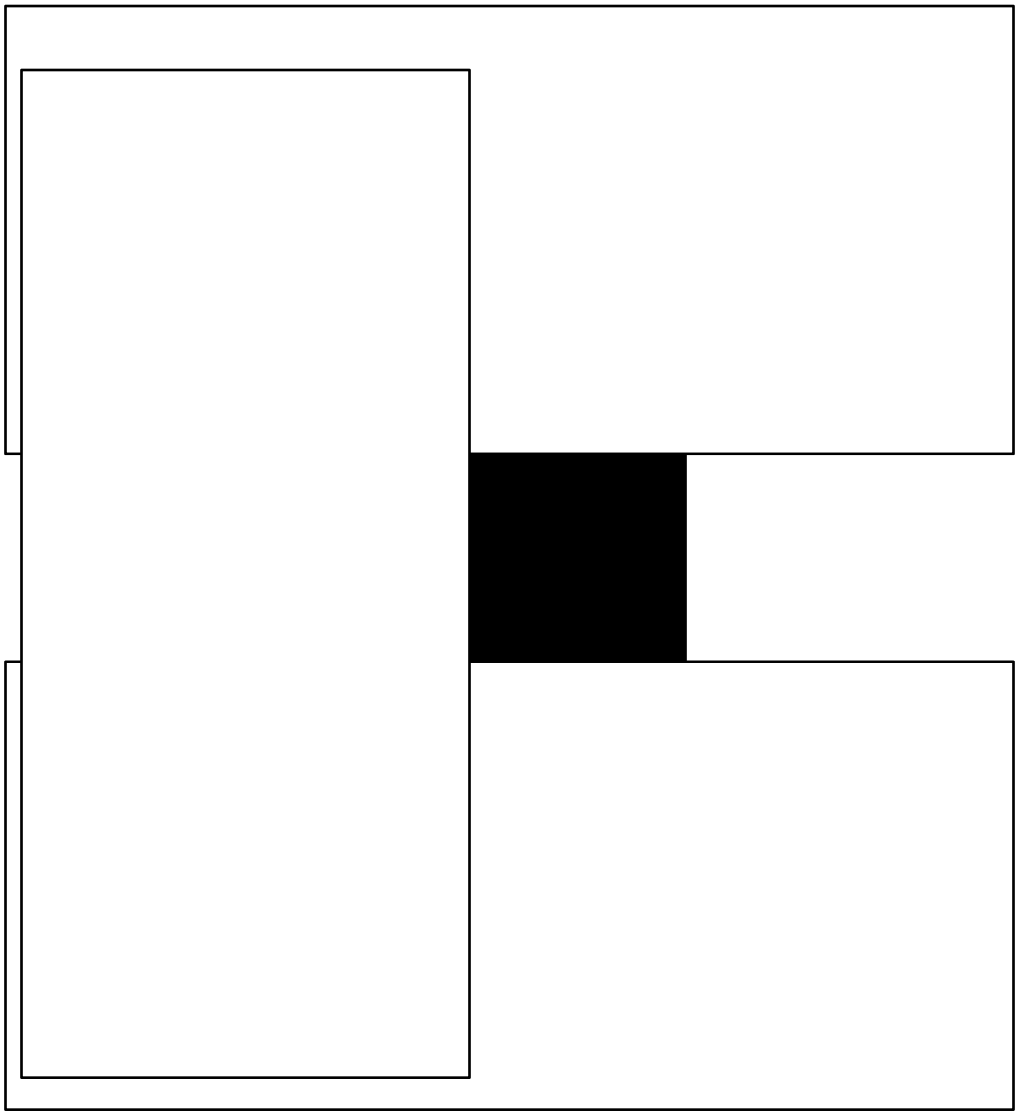
77

What behavior should we see in a motion algorithm?

- Aperture problem
- Resolution through propagation of information
- Figure/ground discrimination

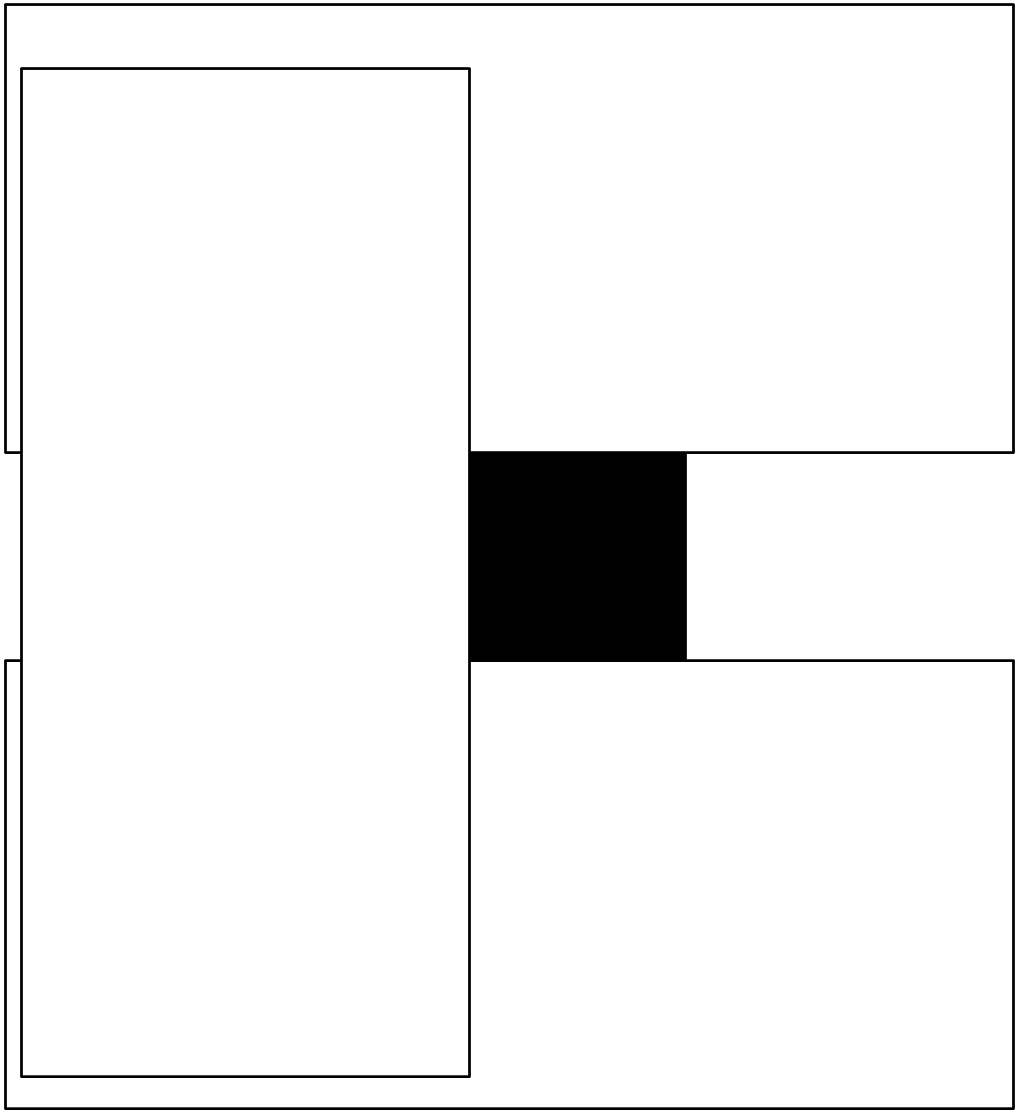
The aperture problem

http://web.mit.edu/persci/demos/Motion&Form/demos/one-square/one-square.html

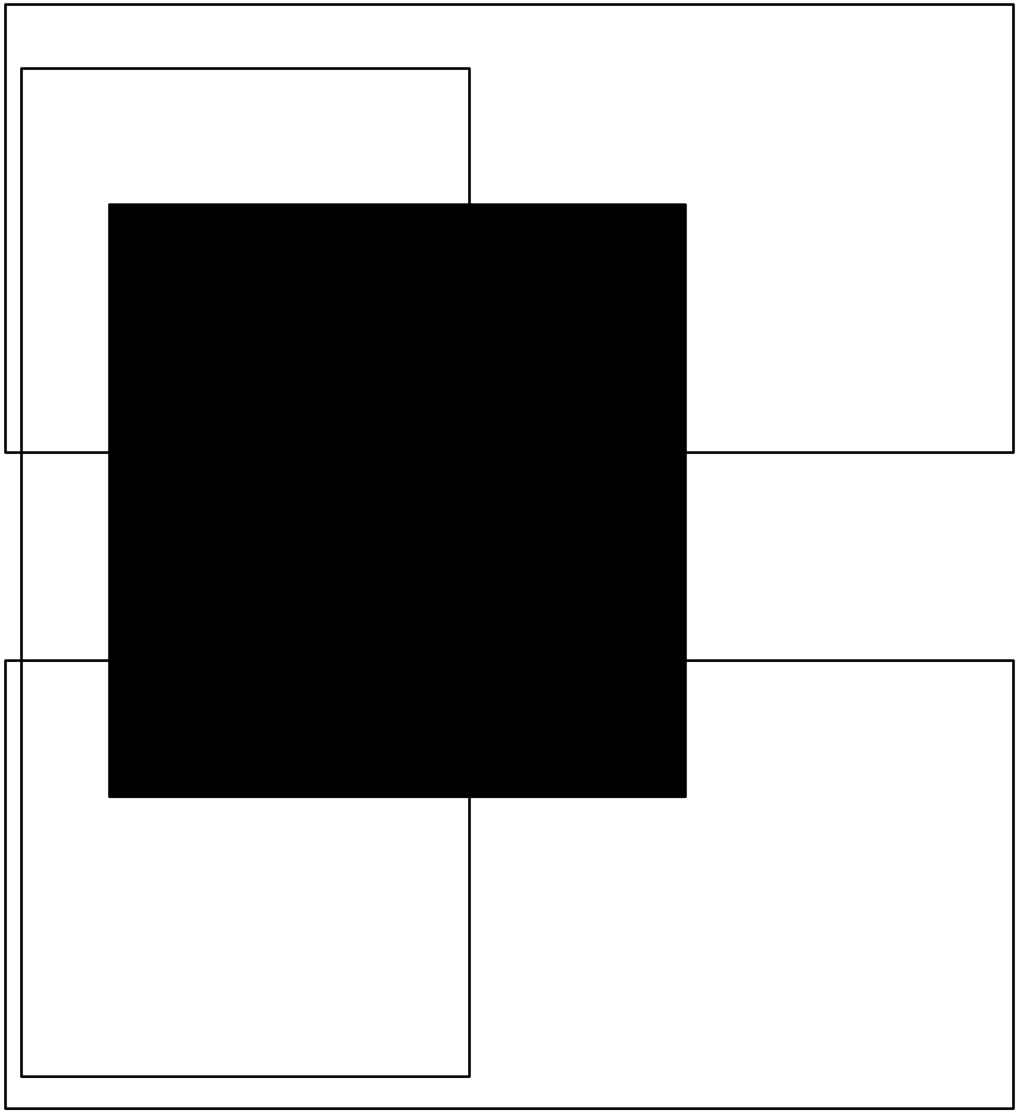


The aperture problem

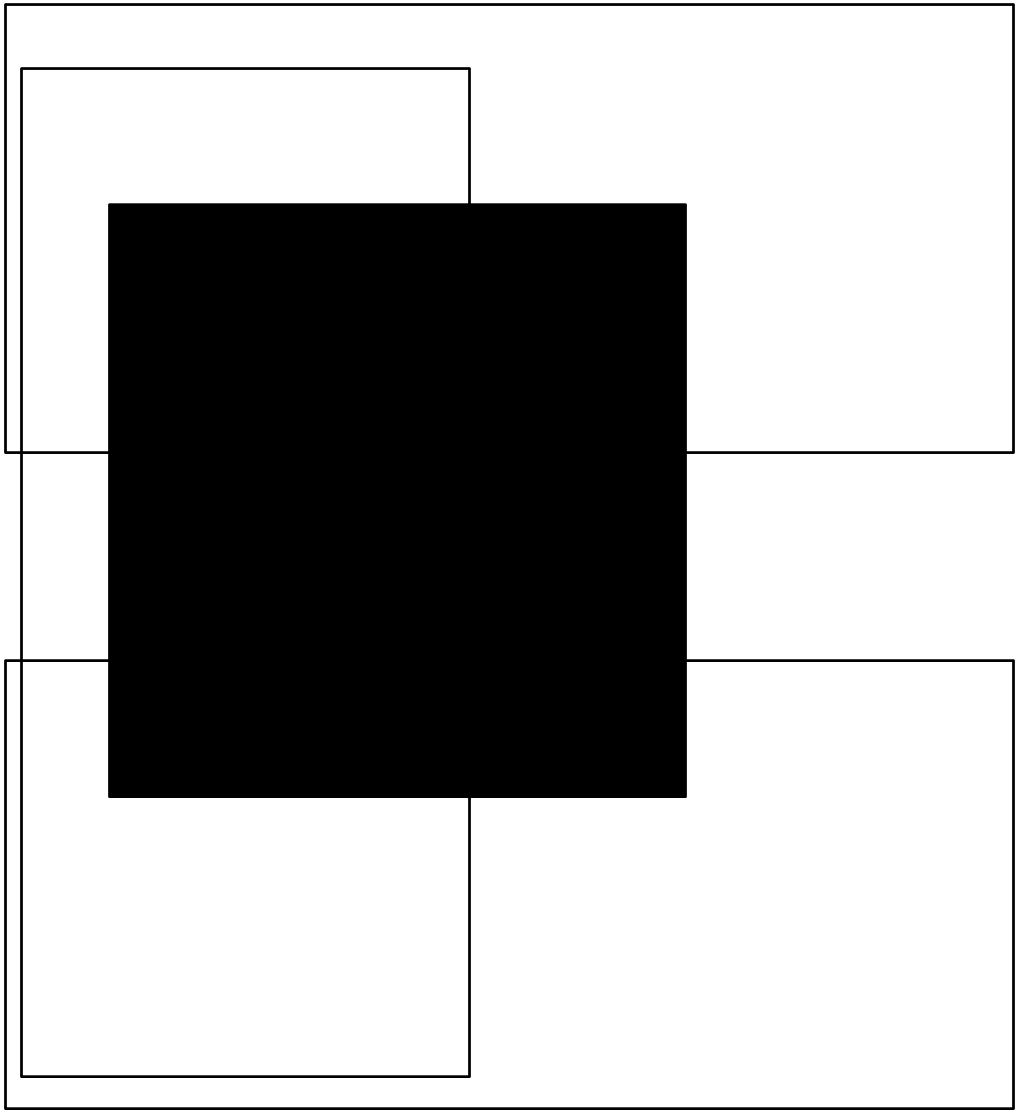
http://web.mit.edu/persci/demos/Motion&Form/demos/one-square/one-square.html



The aperture problem



The aperture problem



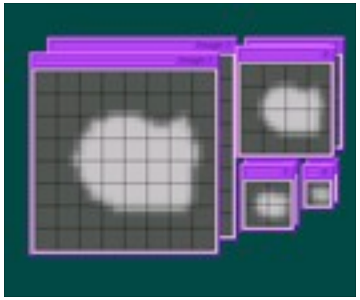
motion program demo

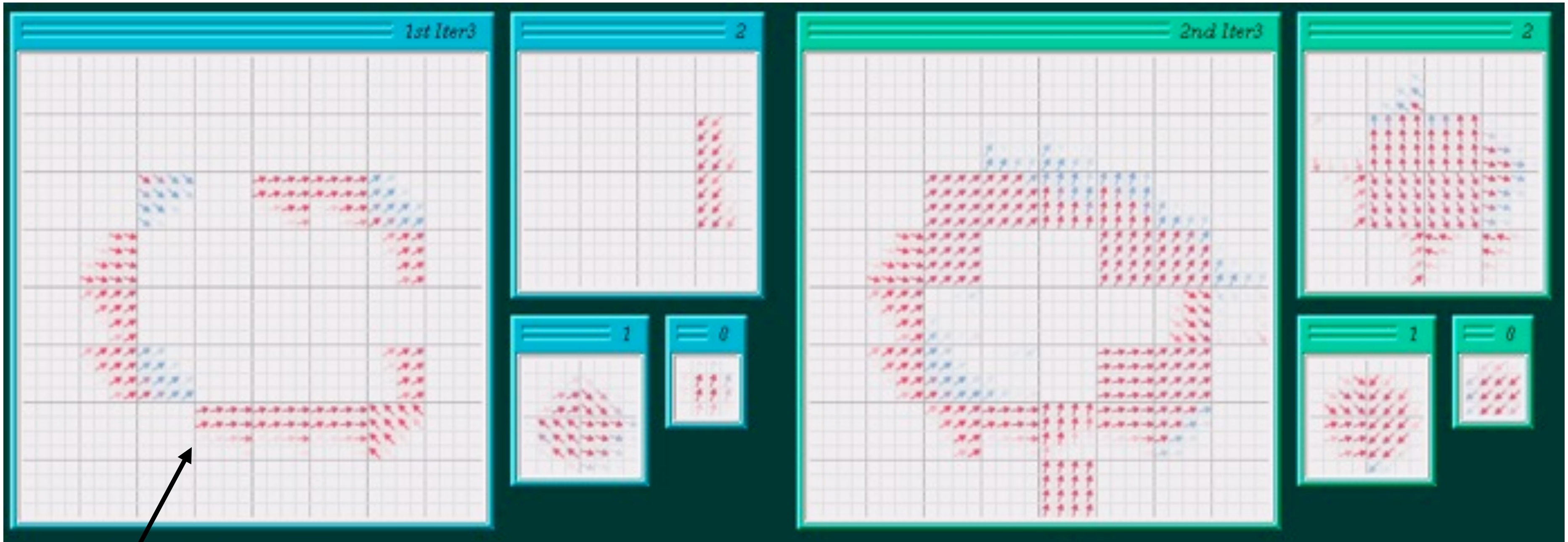
Inference:

Motion estimation results

(maxima of scene probability distributions displayed)

Image data



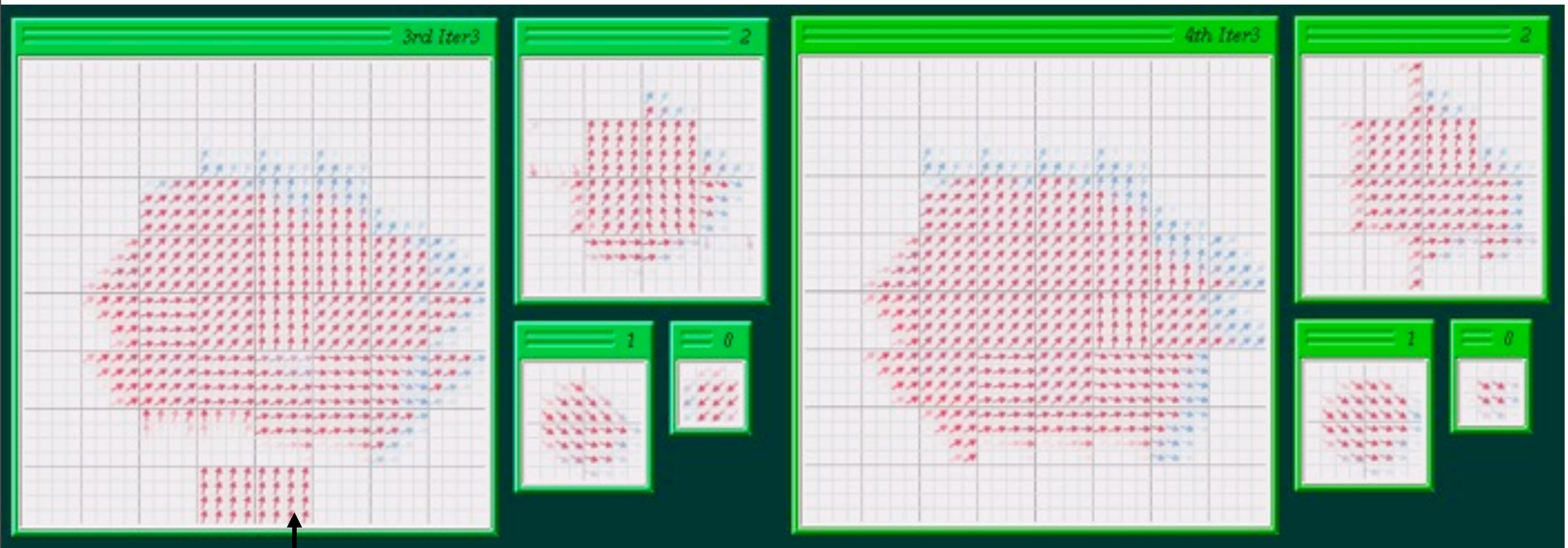


Iterations 0 and 1

Initial guesses only
show motion at edges.

Motion estimation results

(maxima of scene probability distributions displayed)

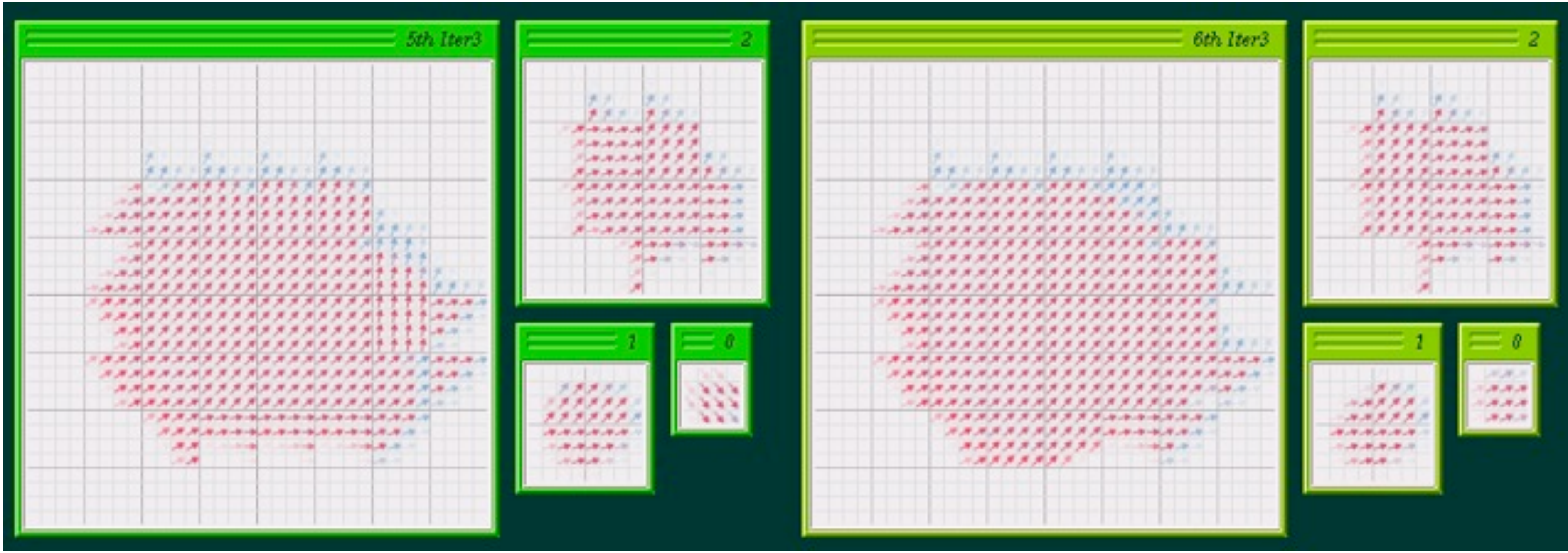


Iterations 2 and 3

Figure/ground still unresolved here.

Motion estimation results

(maxima of scene probability distributions displayed)



Iterations 4 and 5

Final result compares well with vector quantized true (uniform) velocities.

Vision applications of MRF’s

- Stereo
- Motion estimation
- Labelling shading and reflectance
- Segmentation
- Many others…

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

$$\underbrace{p(h \mid I, \theta)}_{\textbf{Posterior}}$$

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

$$\underbrace{p(h \mid I, \theta)}_{\textbf{Posterior}} \propto \underbrace{p(I, h \mid \theta)}_{\textbf{Joint}} = \underbrace{p(I \mid h, \theta)}_{\textbf{Likelihood}} \underbrace{p(h \mid \theta)}_{\textbf{Prior}}$$

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

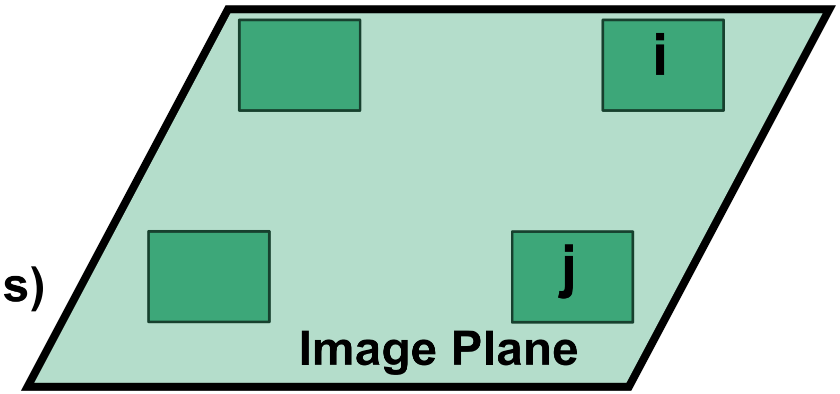
$$\underbrace{p(h \mid I, \theta)}_{\textbf{Posterior}} \propto \underbrace{p(I, h \mid \theta)}_{\textbf{Joint}} = \underbrace{p(I \mid h, \theta)}_{\textbf{Likelihood}} \underbrace{p(h \mid \theta)}_{\textbf{Prior}}$$

1. Generative approach models joint
→ Markov random field (MRF)

2. Discriminative approach models posterior directly
→ Conditional random field (CRF)

Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\boxed{p(h \mid \theta)}$$



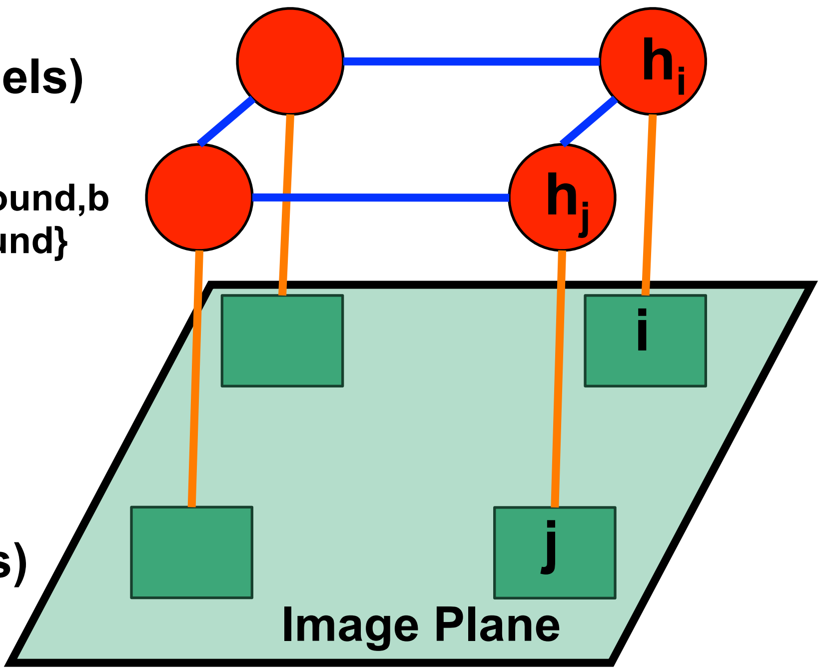
I **(pixels)**

Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$

h **(labels)**

$\in$ **{foreground,background}**



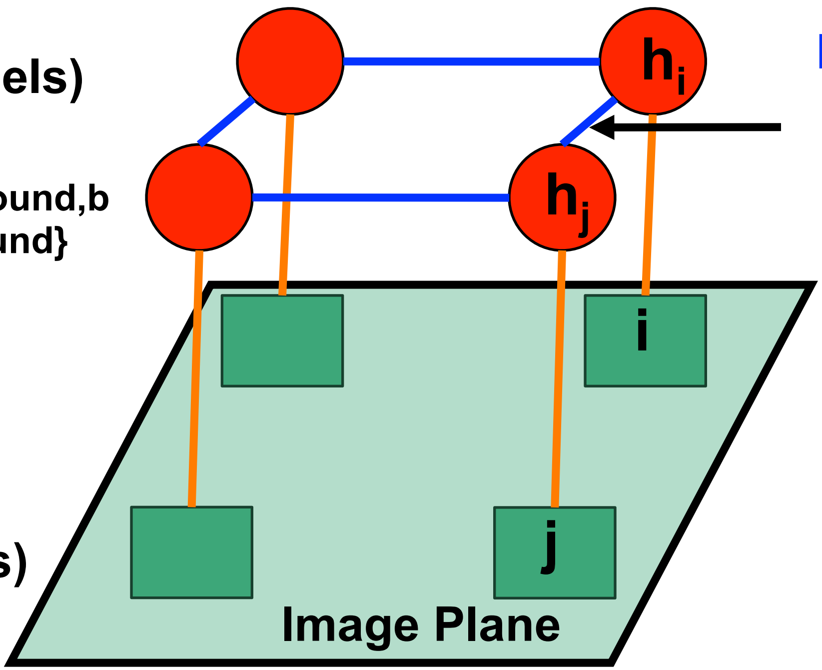
I **(pixels)**

Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$

h **(labels)**

$\in$ **{foreground,background}**

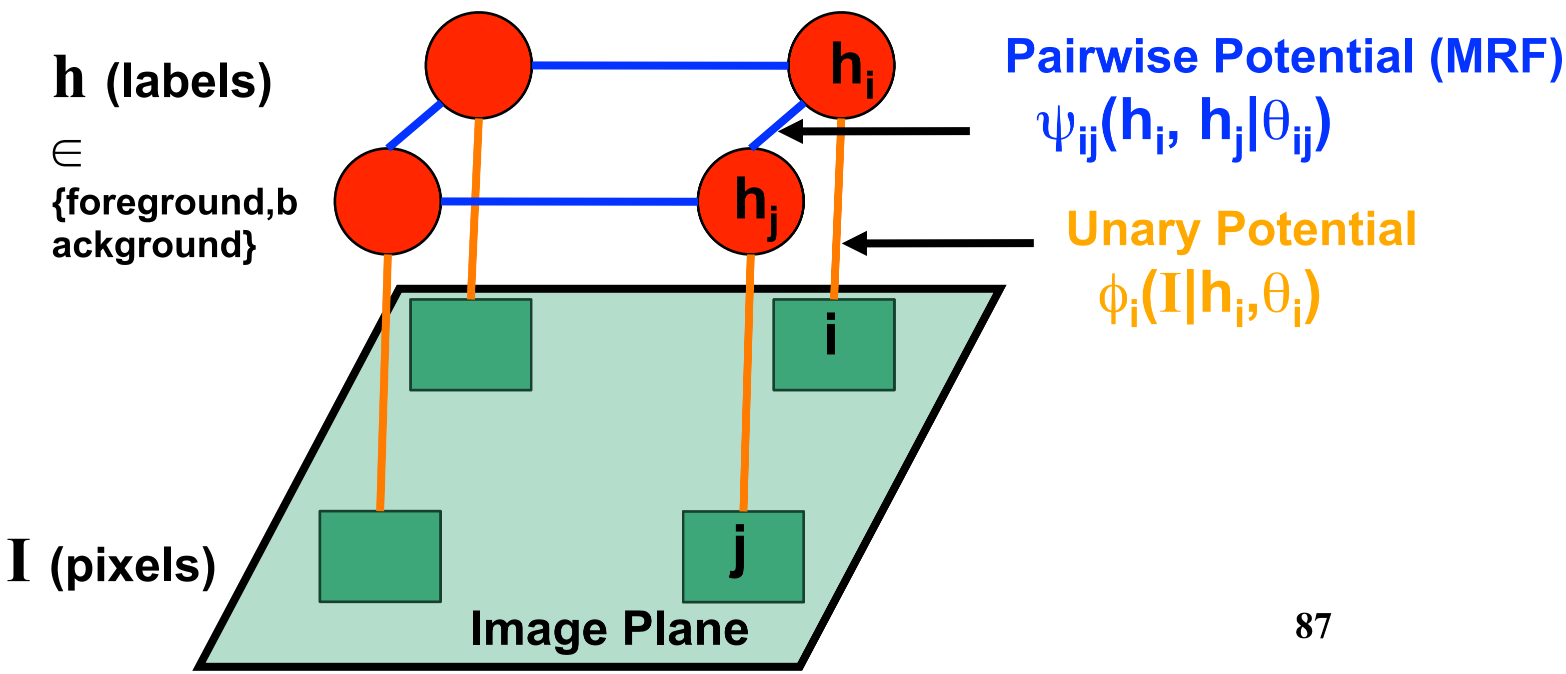


I **(pixels)**

Pairwise Potential (MRF)
$\psi_{ij}(h_i, h_j \mid \theta_{ij})$

Generative Markov Random Field

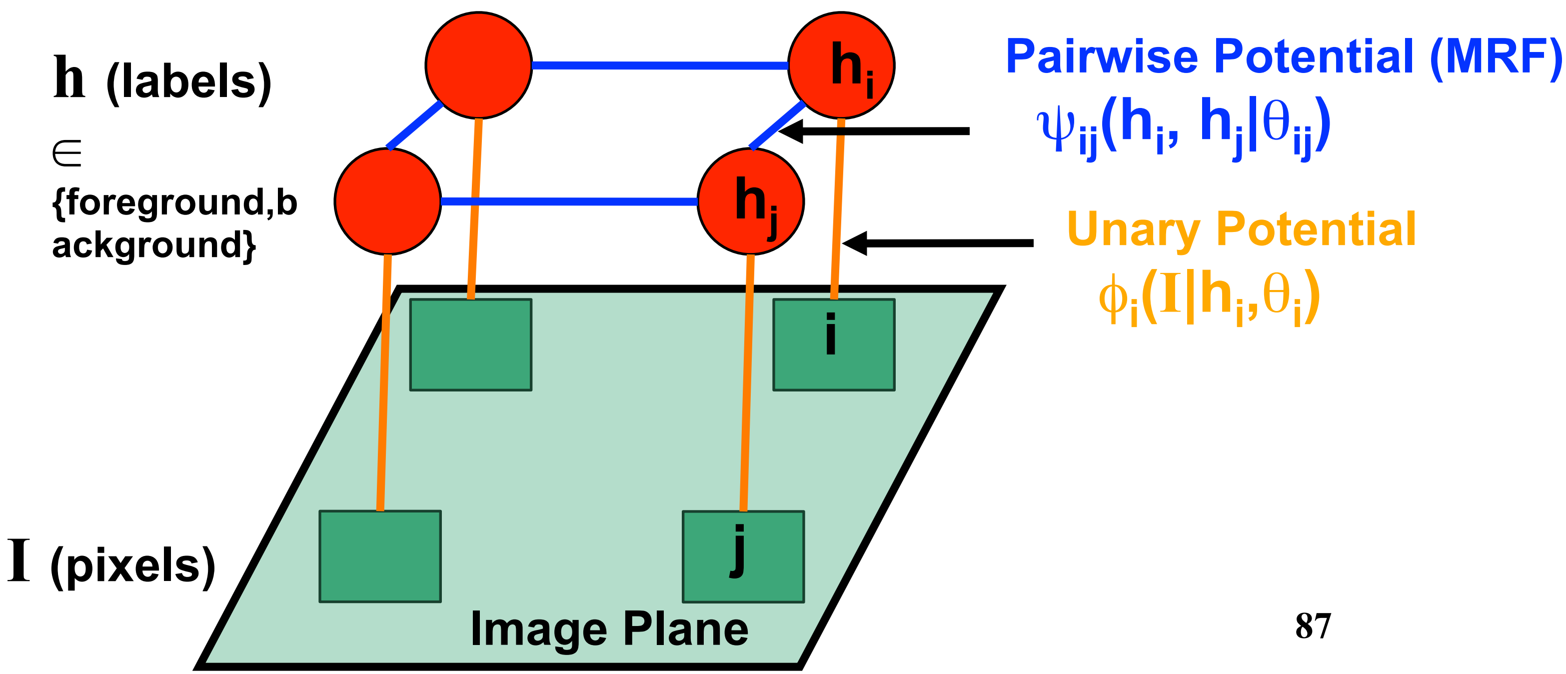
$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$



Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$

$$= \frac{1}{Z(\theta)}\left[\prod_i \underbrace{\phi_i(I \mid h_i, \theta_i)}_{\text{Likelihood}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j \mid \theta_{ij})}_{\text{MRF Prior}}\right]$$



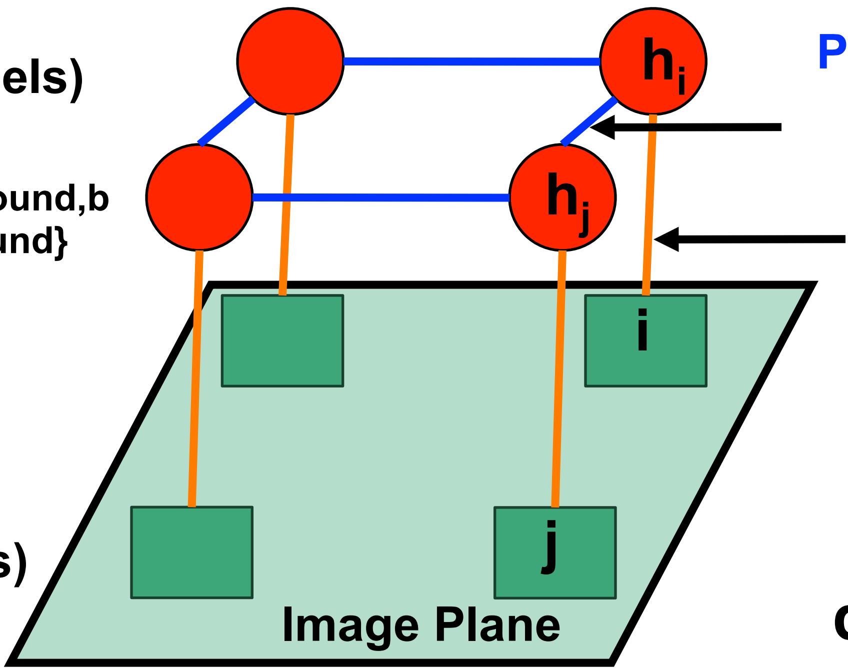
Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$

$$= \frac{1}{Z(\theta)}\left[\underbrace{\prod_i \phi_i(I \mid h_i, \theta_i)}_{\text{Likelihood}}\;\prod_{ij}\underbrace{\psi_{ij}(h_i, h_j \mid \theta_{ij})}_{\text{MRF Prior}}\right]$$

h (labels)

$\in$ {foreground,background}



I (pixels)

Pairwise Potential (MRF)
$\psi_{ij}(h_i, h_j|\theta_{ij})$

Unary Potential
$\phi_i(I|h_i,\theta_i)$

Prior has no dependency on I

Conditional Random Field

Discriminative approach

Lafferty, McCallum and Pereira 2001

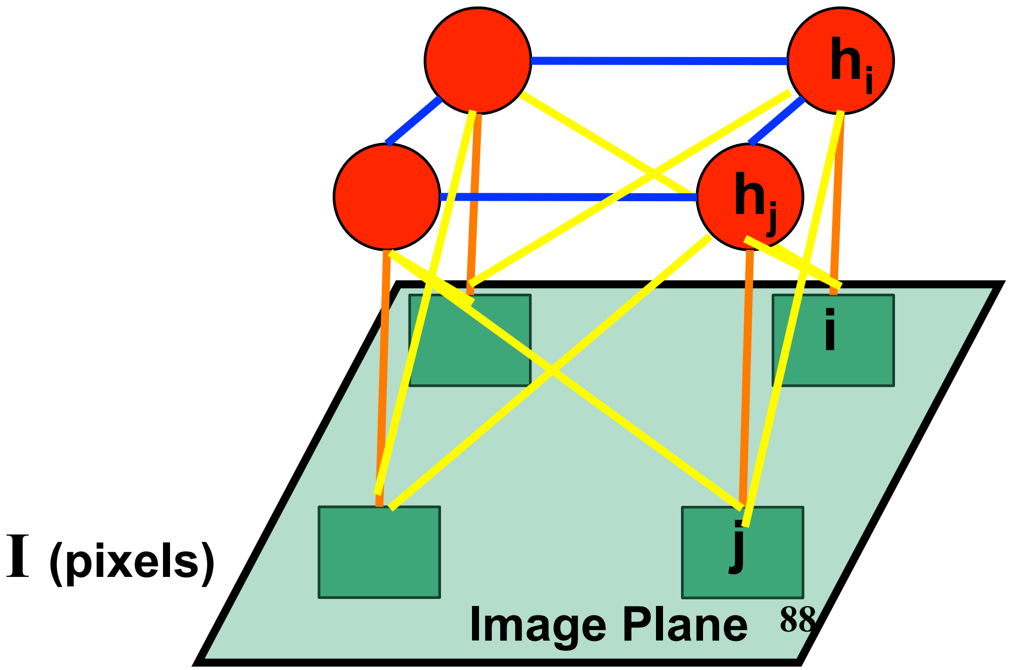
$$p(h \mid I,\theta) = \frac{1}{Z(I,\theta)}\left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\textbf{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\textbf{Pairwise}}\right]$$

Conditional Random Field

Discriminative approach

Lafferty, McCallum and Pereira 2001

$$p(h \mid I, \theta) = \frac{1}{Z(I,\theta)} \left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\textbf{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\textbf{Pairwise}} \right]$$



Conditional Random Field

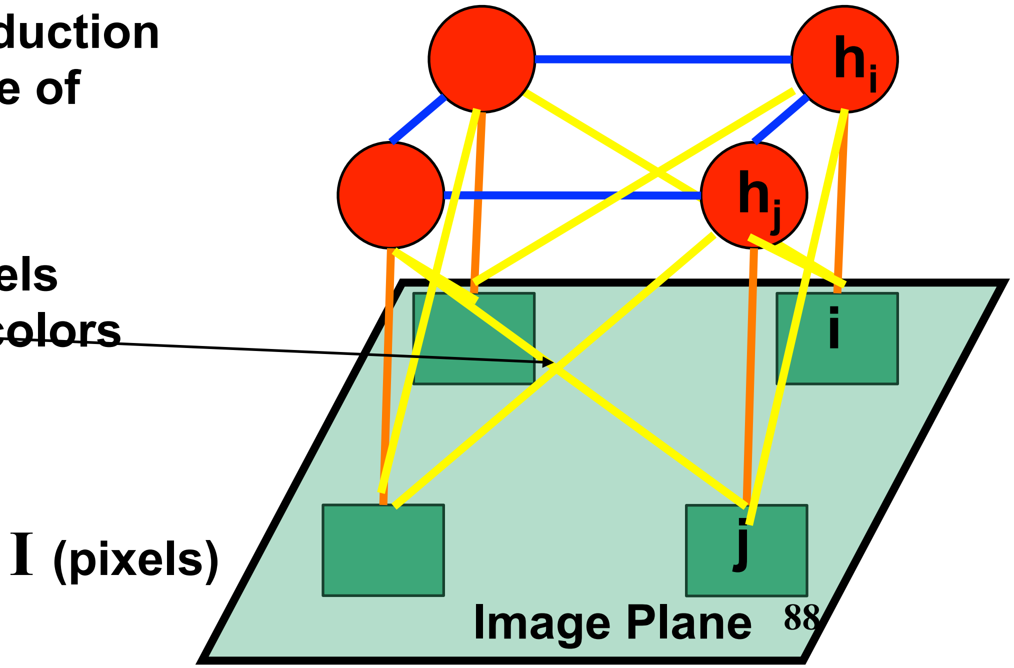
Discriminative approach

Lafferty, McCallum and Pereira 2001

$$p(h \mid I, \theta) = \frac{1}{Z(I,\theta)} \left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\textbf{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\textbf{Pairwise}} \right]$$

- **Dependency on I allows introduction of pairwise terms that make use of image.**

- **For example, neighboring labels should be similar only if pixel colors are similar → Contrast term**



Conditional Random Field

Discriminative approach

Lafferty, McCallum and Pereira 2001

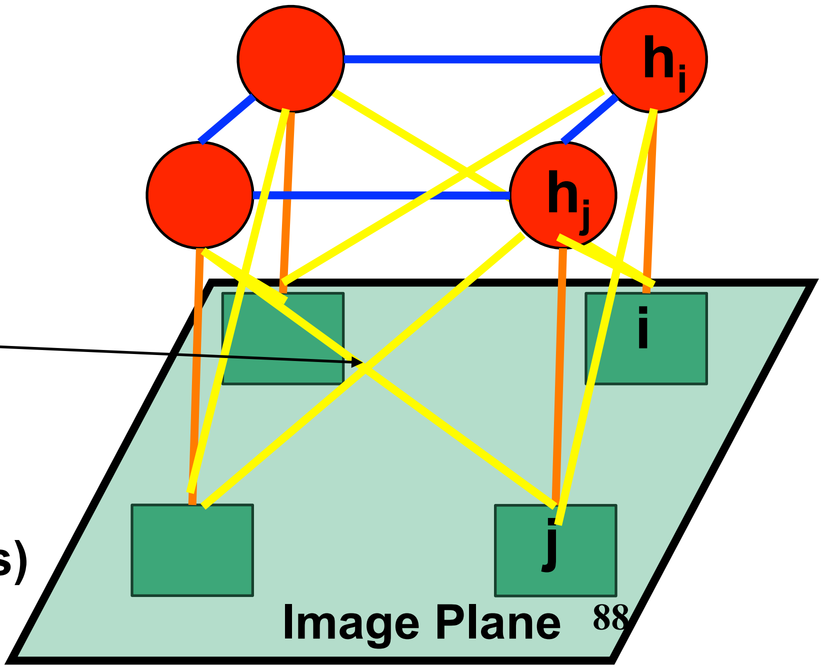
$$p(h \mid I,\theta) = \frac{1}{Z(I,\theta)}\left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\text{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\text{Pairwise}}\right]$$

- **Dependency on I allows introduction of pairwise terms that make use of image.**

- **For example, neighboring labels should be similar only if pixel colors are similar → Contrast term**

e.g Kumar and Hebert 2003

I (pixels)



OBJCUT

Kumar, Torr & Zisserman 2005

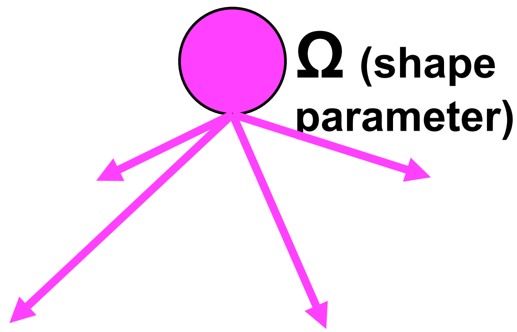
$$p(h \mid \Omega, I, \theta) \propto \left[\prod_i \overbrace{\underbrace{\phi_i^1(I \mid h_i, \theta_i)}_{\text{Color Likelihood}} \times \underbrace{\phi_i^2(h_i \mid \Omega)}_{\text{Distance from } \Omega}}^{\text{Unary}} \prod_{ij} \overbrace{\underbrace{\psi_{ij}^1(h_i, h_j \mid \theta_{ij})}_{\text{Label smoothness}} \times \underbrace{\psi_{ij}^2(I \mid h_i, h_j, \theta_{ij})}_{\text{Contrast}}}^{\text{Pairwise}} \right]$$

- **Ω is a shape prior on the labels from a Layered Pictorial Structure (LPS) model**

- **Segmentation by:**

    - Match LPS model to image (get number of samples, each with a different pose

    -Marginalize over the samples using a single graph cut [Boykov & Jolly, 2001]



OBJCUT

Kumar, Torr & Zisserman 2005

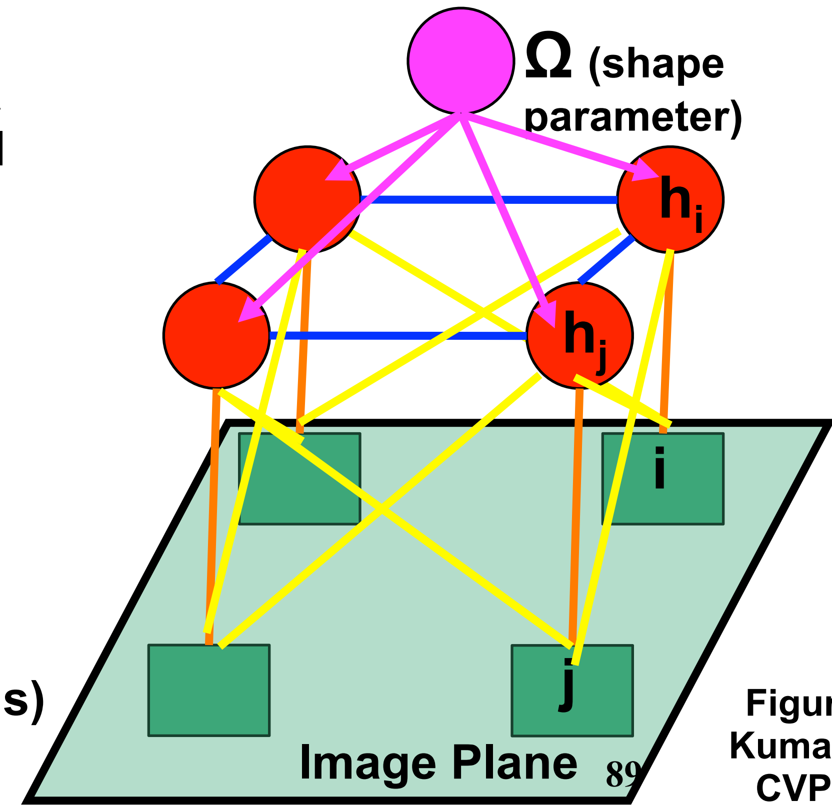
$$p(h\,|\,\Omega, I, \theta) \propto \left[\prod_i \overbrace{\underbrace{\phi_i^1(I\,|\,h_i,\theta_i)}_{\text{Color Likelihood}} \times \underbrace{\phi_i^2(h_i\,|\,\Omega)}_{\text{Distance from }\Omega}}^{\text{Unary}} \prod_{ij} \overbrace{\underbrace{\psi_{ij}^1(h_i,h_j\,|\,\theta_{ij})}_{\text{Label smoothness}} \times \underbrace{\psi_{ij}^2(I\,|\,h_i,h_j,\theta_{ij})}_{\text{Contrast}}}^{\text{Pairwise}} \right]$$

- **Ω is a shape prior on the labels from a Layered Pictorial Structure (LPS) model**

- **Segmentation by:**

  - Match LPS model to image (get number of samples, each with a different pose

  -Marginalize over the samples using a single graph cut [Boykov & Jolly, 2001]

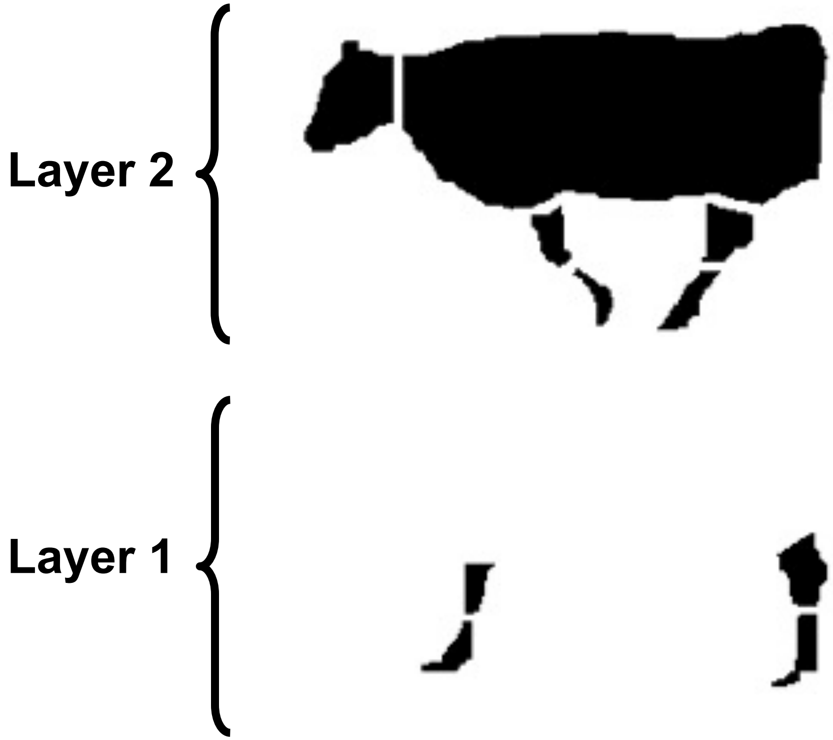


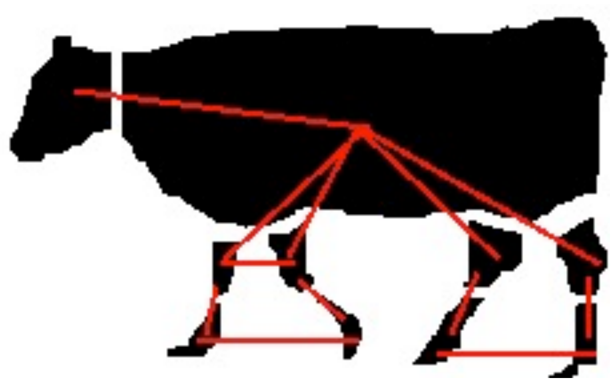
I (pixels)

Figure from Kumar et al., CVPR 2005

OBJCUT:
Shape prior - **Ω** - Layered Pictorial Structures (LPS)

- Generative model
- Composition of parts + spatial layout





Spatial Layout
(Pairwise Configuration)

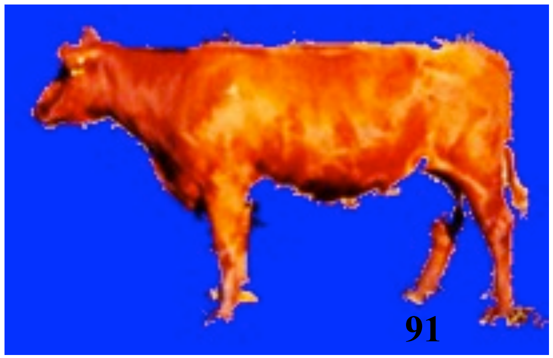
Parts in Layer 2 can occlude parts in Layer 1

Kumar, et al. 2004,

OBJCUT: Results

Using LPS Model for Cow

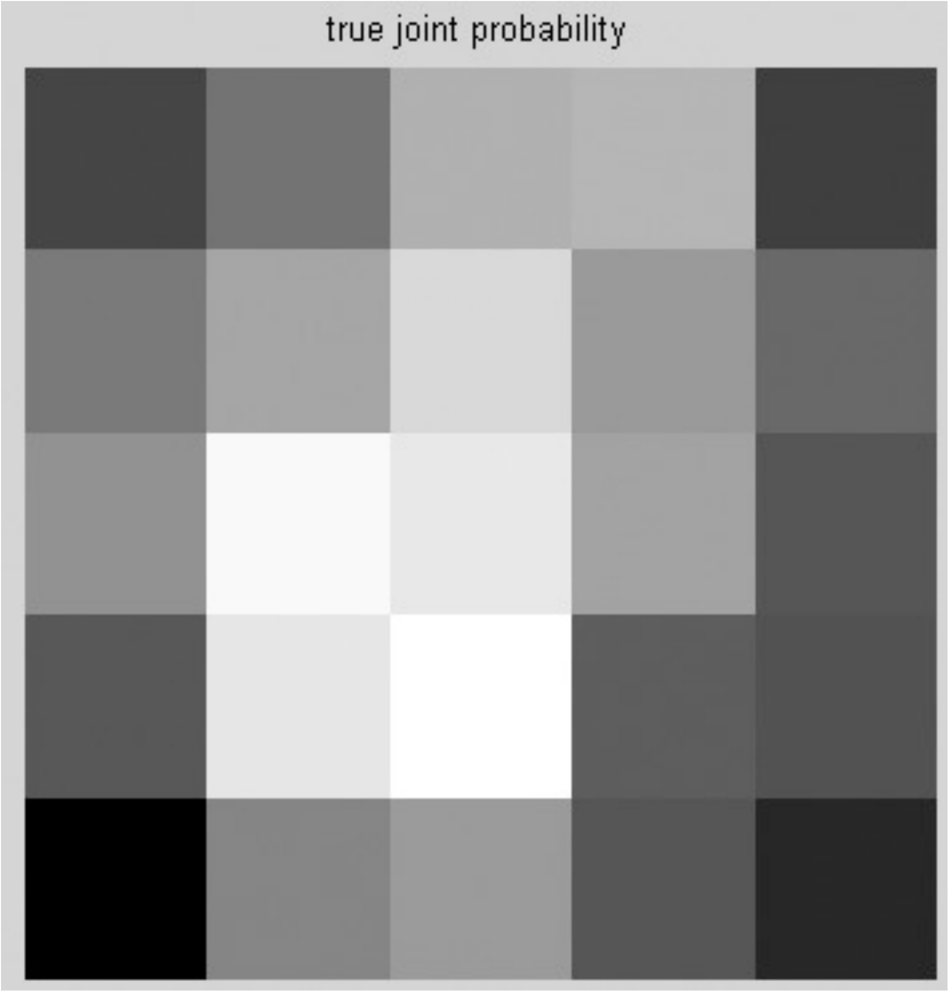
In the absence of a clear boundary between object and background

Image	Segmentation
	
	

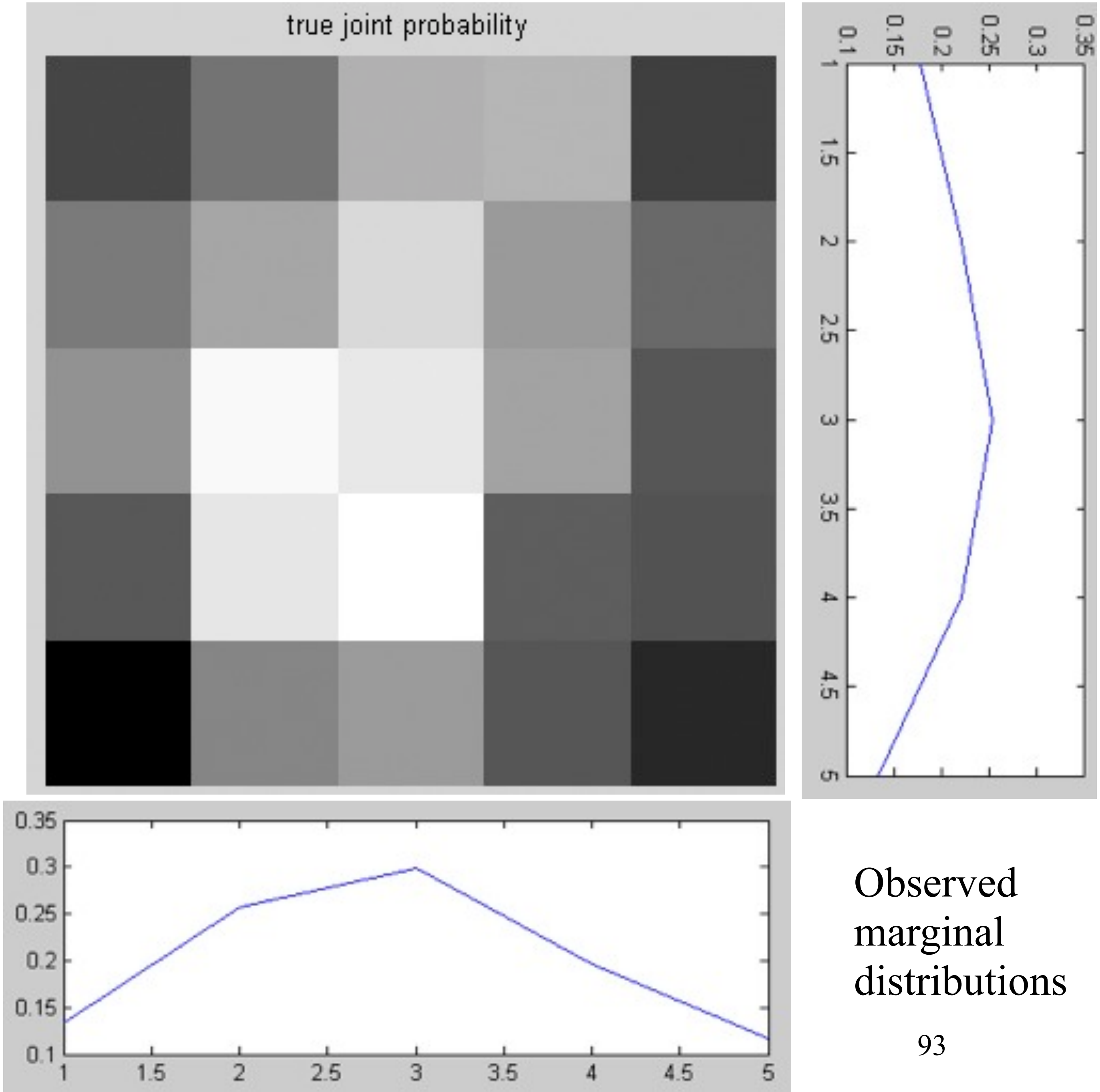
Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
  - Variational methods
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

True joint probability

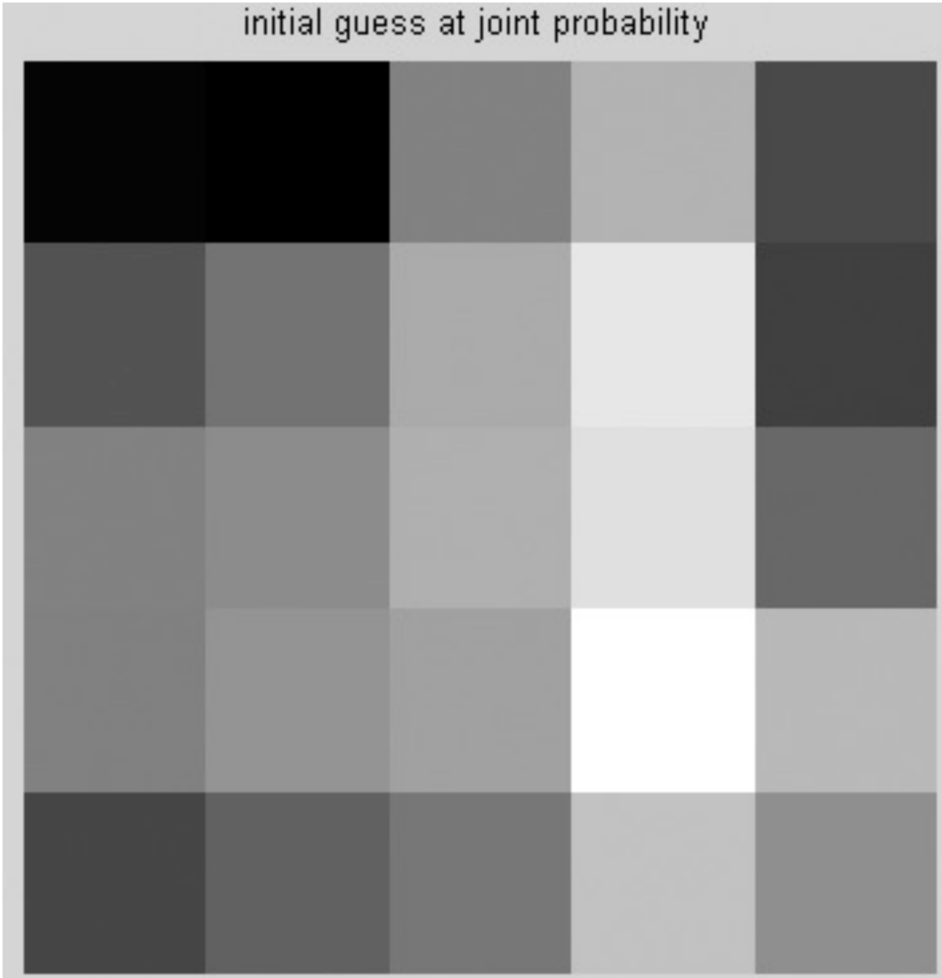


True joint probability



Observed marginal distributions

Initial guess at joint probability



IPF update equation, for maximum likelihood estimate of clique potentials

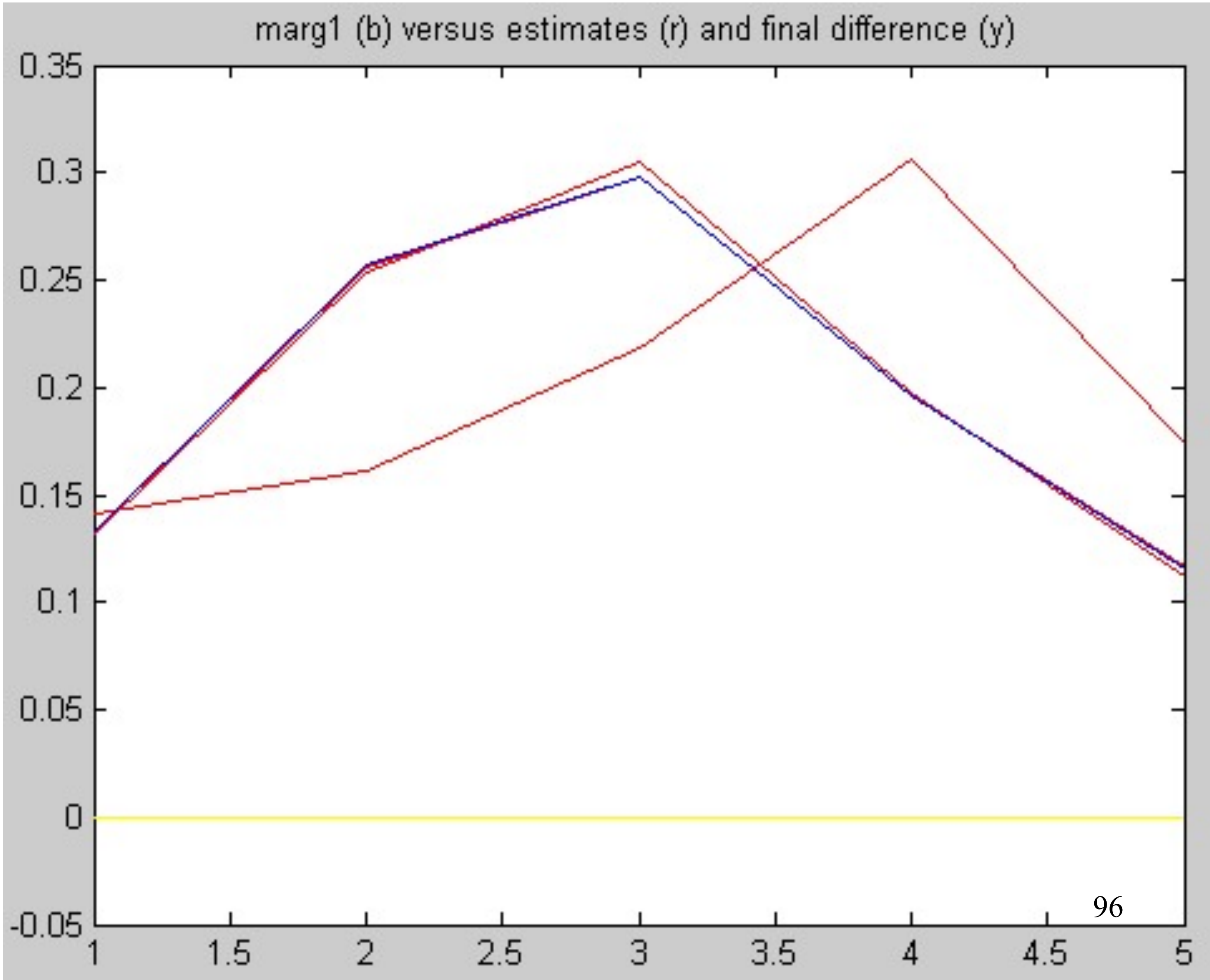
$$P(x_1, x_2, \ldots, x_d)^{(t+1)} = P(x_1, x_2, \ldots, x_d)^{(t)} \frac{P(x_i)^{\text{observed}}}{P(x_i)^{(t)}}$$

Scale the previous iteration's estimate for the joint probability by the ratio of the true to the predicted marginals.

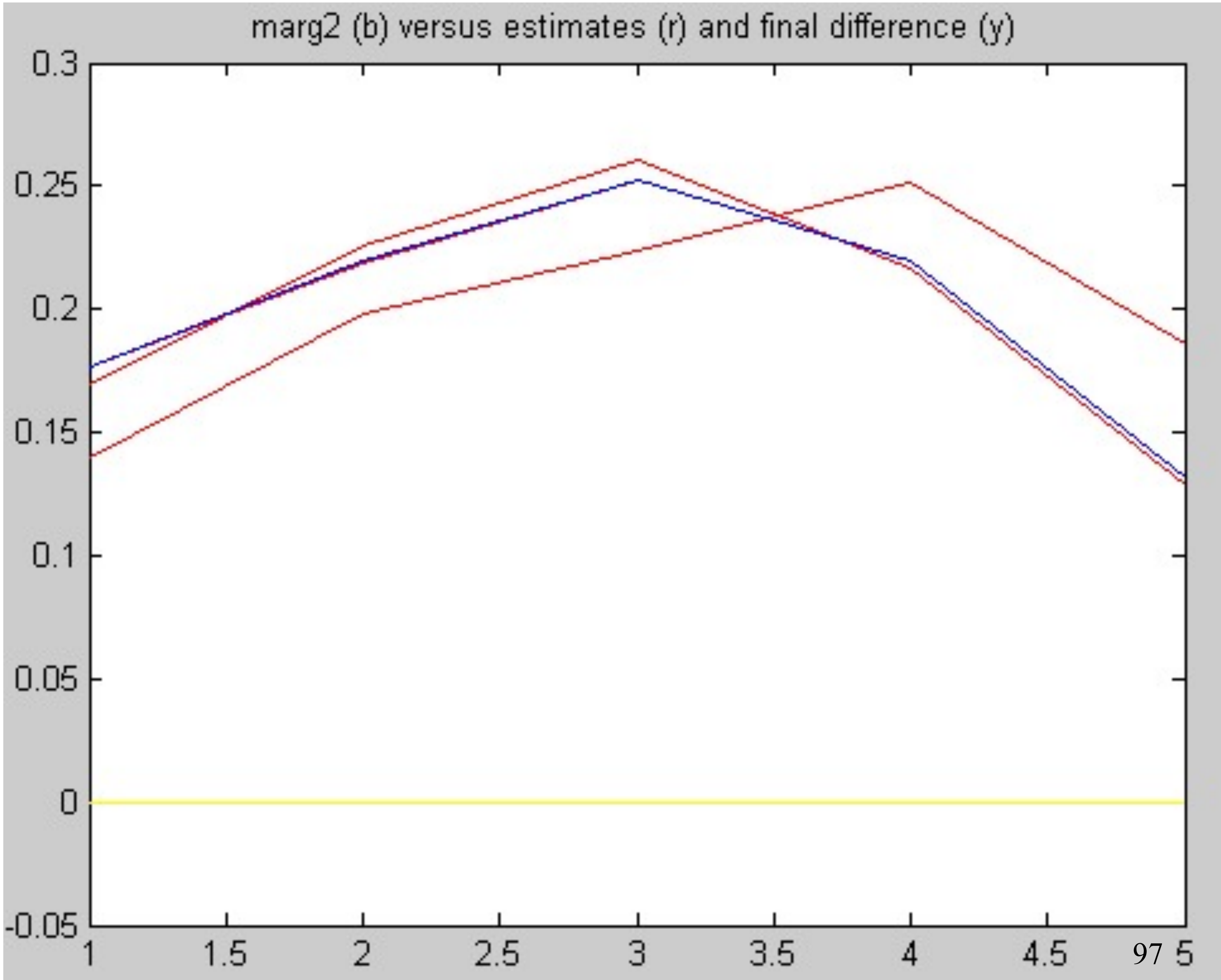
Gives gradient ascent in the likelihood of the joint probability, given the observations of the marginals.

See: Michael Jordan's book on graphical models

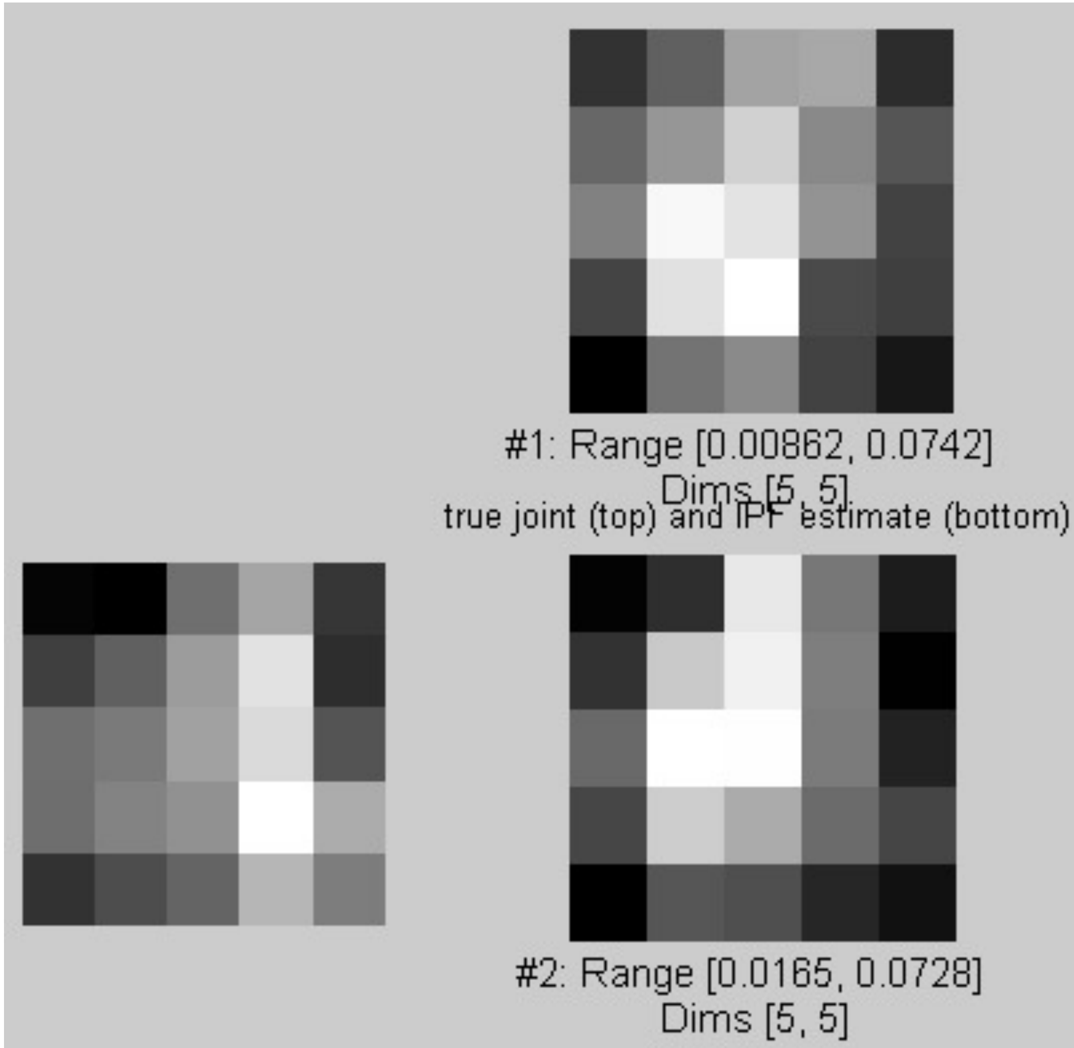
Convergence to correct marginals by IPF algorithm



Convergence of to correct marginals by IPF algorithm



IPF results for this example: comparison of joint probabilities



True joint probability

Initial guess

Final maximum entropy estimate

Application to MRF parameter estimation

- Can show that for the ML estimate of the clique potentials, $\phi_c(x_c)$, the empirical marginals equal the model marginals,

$$\tilde{p}(x_c) = p(x_c)$$

- Because the model marginals are proportional to the clique potentials, we have the IPF update rule for $\phi_c(x_c)$, which scales the model marginals to equal the observed marginals:

$$\phi_C^{(t+1)}(x_c) = \phi_c^{(t)}(x_c)\frac{\tilde{p}(x_c)}{p^{(t)}(x_c)}$$

- Performs coordinate ascent in the likelihood of the MRF parameters, given the observed data.

Reference: unpublished notes by Michael Jordan, and by Roweis: http://www.cs.toronto.edu/~roweis/csc412-2004/notes/lec11x.pdf

Learning MRF parameters, labeled data

Iterative proportional fitting lets you make a maximum likelihood estimate a joint distribution from observations of various marginal distributions.

Applied to learning MRF clique potentials:

(1) measure the pairwise marginal statistics (histogram state co-occurrences in labeled training data).

(2) guess the clique potentials (use measured marginals to start).

(3) do inference to calculate the model's marginals for every node pair.

(4) scale each clique potential for each state pair by the empirical over model marginal