C280, Computer Vision

Prof. Trevor Darrell
trevor@eecs.berkeley.edu

Lecture 20: Markov Random Fields

Slide Credits

- Bill Freeman
- Yuri Boykov
- others, as noted…

Making probability distributions modular, and therefore tractable:

Probabilistic graphical models

Vision is a problem involving the interactions of many variables: things can seem hopelessly complex. Everything is made tractable, or at least, simpler, if we modularize the problem. That's what probabilistic graphical models do, and let's examine that.

Readings: Jordan and Weiss intro article—fantastic!
Kevin Murphy web page—comprehensive and with pointers to many advanced topics

A toy example

Suppose we have a system of 5 interacting variables, perhaps some are observed and some are not. There's some probabilistic relationship between the 5 variables, described by their joint probability,
P(x1, x2, x3, x4, x5).

If we want to find out what the likely state of variable x1 is (say, the position of the hand of some person we are observing), what can we do?

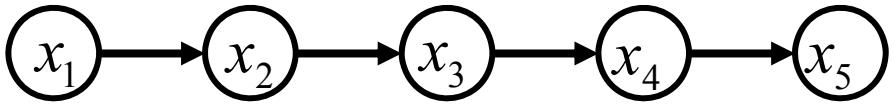
Two reasonable choices are: (a) find the value of x1 (and of all the other variables) that gives the maximum of P(x1, x2, x3, x4, x5); that's the MAP solution.
Or (b) marginalize over all the other variables and then take the mean or the maximum of the other variables. Marginalizing, then taking the mean, is equivalent to finding the MMSE solution. Marginalizing, then taking the max, is called the max marginal solution and sometimes a useful thing to do.

To find the marginal probability at x1, we have to take this sum:

$$\sum_{x_2, x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5)$$

If the system really is high dimensional, that will quickly become intractable. But if there is some modularity in $P(x_1, x_2, x_3, x_4, x_5)$ then things become tractable again.

Suppose the variables form a Markov chain: x1 causes x2 which causes x3, etc. We might draw out this relationship as follows:



P(a,b) = P(b|a) P(a)

By the chain rule, for any probability distribution, we have:

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2, x_3, x_4, x_5 \mid x_1)$$

$$= P(x_1)P(x_2 \mid x_1)P(x_3, x_4, x_5 \mid x_1, x_2)$$

$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4, x_5 \mid x_1, x_2, x_3)$$

$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4 \mid x_1, x_2, x_3)P(x_5 \mid x_1, x_2, x_3, x_4)$$

But if we exploit the assumed modularity of the probability distribution over the 5 variables (in this case, the assumed Markov chain structure), then that expression simplifies:

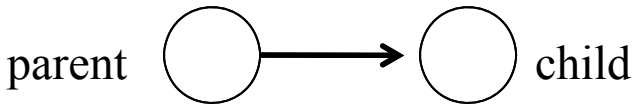
$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)P(x_4 \mid x_3)P(x_5 \mid x_4)$$

Now our marginalization summations distribute through those terms:

$$\sum_{x_2, x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5) = \quad P(x_1) \sum_{x_2} P(x_2 \mid x_1) \sum_{x_3} P(x_3 \mid x_2) \sum_{x_4} P(x_4 \mid x_3) \sum_{x_5} P(x_5 \mid x_4)$$

Directed graphical models

- A directed, acyclic graph.
- Nodes are random variables. Can be scalars or vectors, continuous or discrete.
- The direction of the edge tells the parent-child-relation:



- With every node i is associated a conditional pdf defined by all the parent nodes π_i of node i. That conditional probability is $P_{x_i | x_{\pi_i}}$

- The joint distribution depicted by the graph is the product of all those conditional probabilities:

$$P_{x_1 \ldots x_n} = \prod_{i=1}^{n} P_{x_i | x_{\pi_i}}$$

Another modular probabilistic structure, more common in vision problems, is an undirected graph:

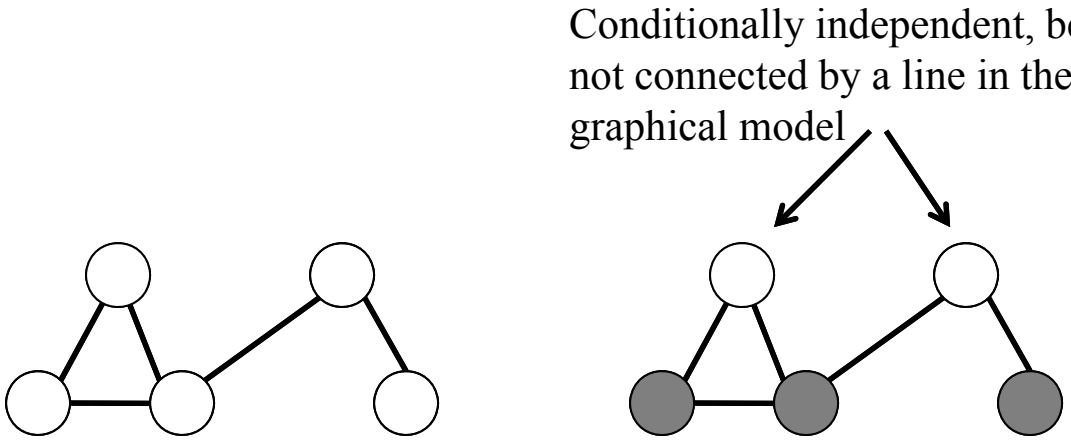
The joint probability for this graph is given by:

$$P(x_1, x_2, x_3, x_4, x_5) = \Phi(x_1, x_2)\Phi(x_2, x_3)\Phi(x_3, x_4)\Phi(x_4, x_5)$$

Where $\Phi(x_1, x_2)$ is called a "compatibility function". We can define compatibility functions we result in the same joint probability as for the directed graph described in the previous slides; for that example, we could use either form.

Undirected graphical models

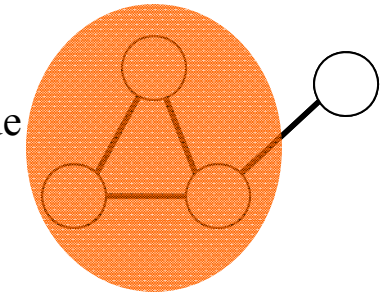
- A set of nodes joined by undirected edges.
- The graph makes conditional independencies explicit: If two nodes are not linked, and we condition on every other node in the graph, then those two nodes are conditionally independent.

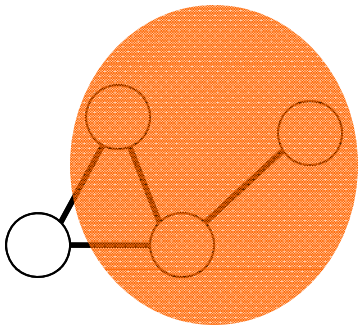


Undirected graphical models: cliques

- Clique: a fully connected set of nodes

clique

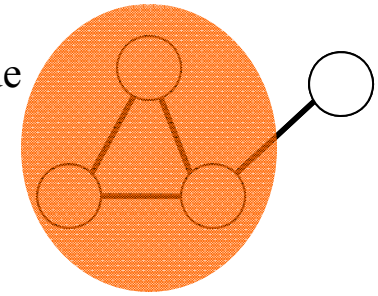




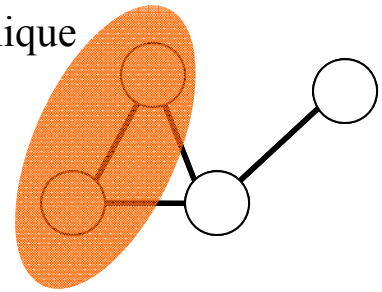
not a clique

- 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

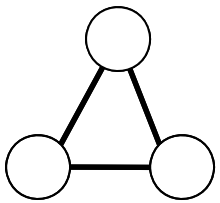
$$P_x = \frac{1}{Z} \prod_{c \in \xi} \Psi_{x_c}$$

Potential functions of states of variables in maximal clique

Normalizing constant

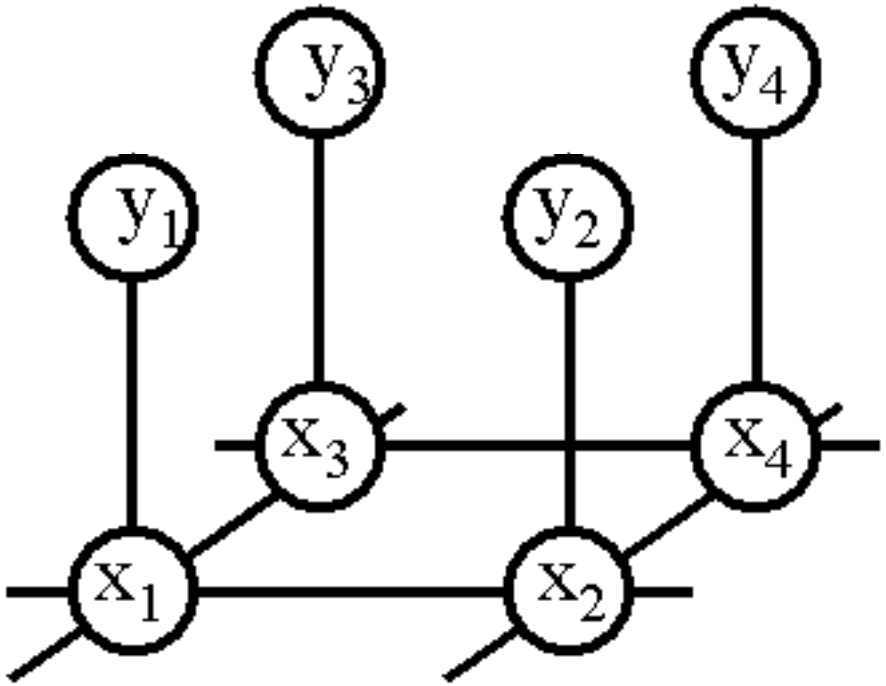
set of maximal cliques

- So for this graph, the factorization is ambiguous:. (something called "factor graphs" makes it explicit).



Markov Random Fields

- Allows rich probabilistic models for images.
- But built in a local, modular way. Learn local relationships, get global effects out.



MRF nodes as pixels

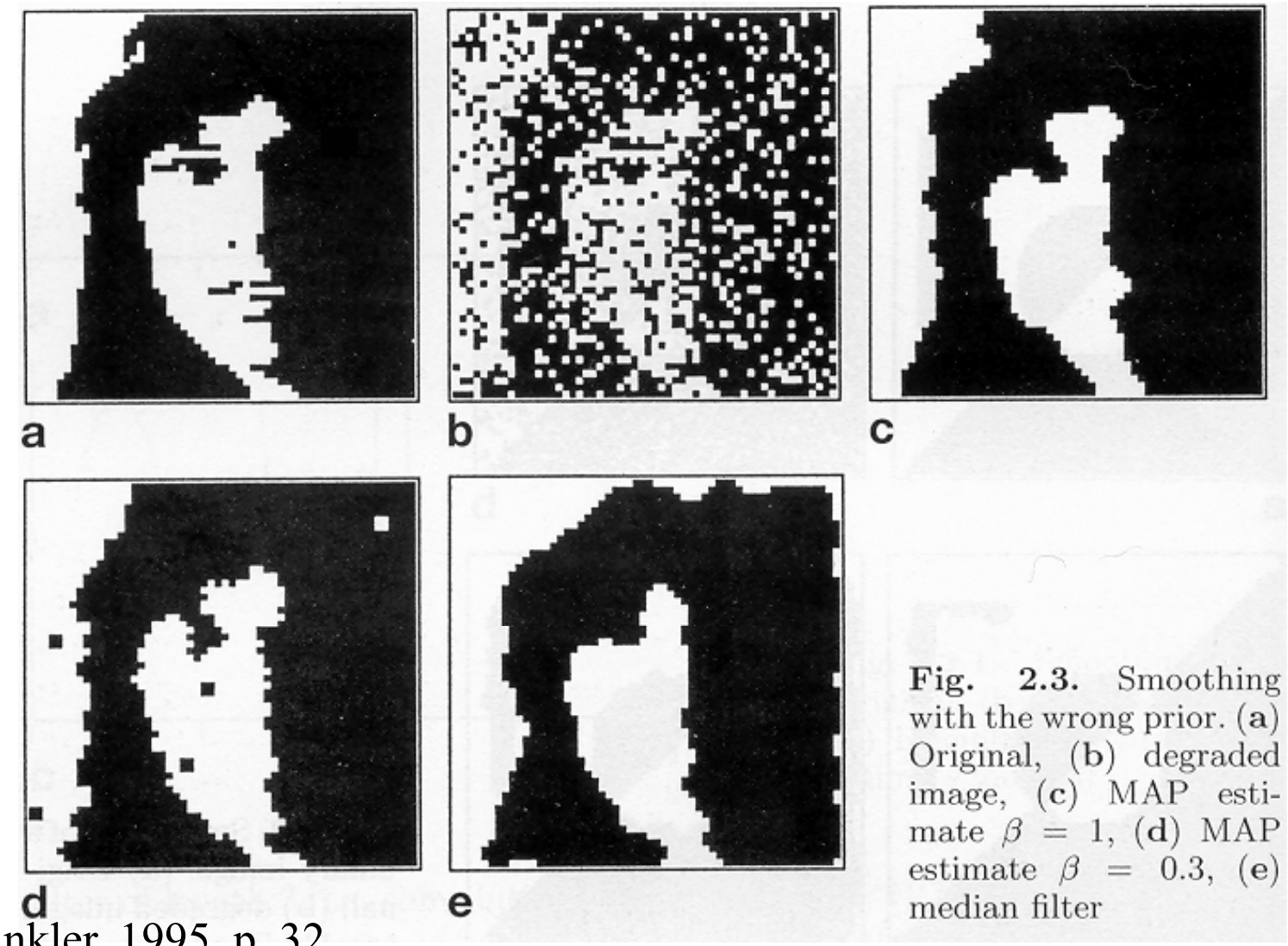
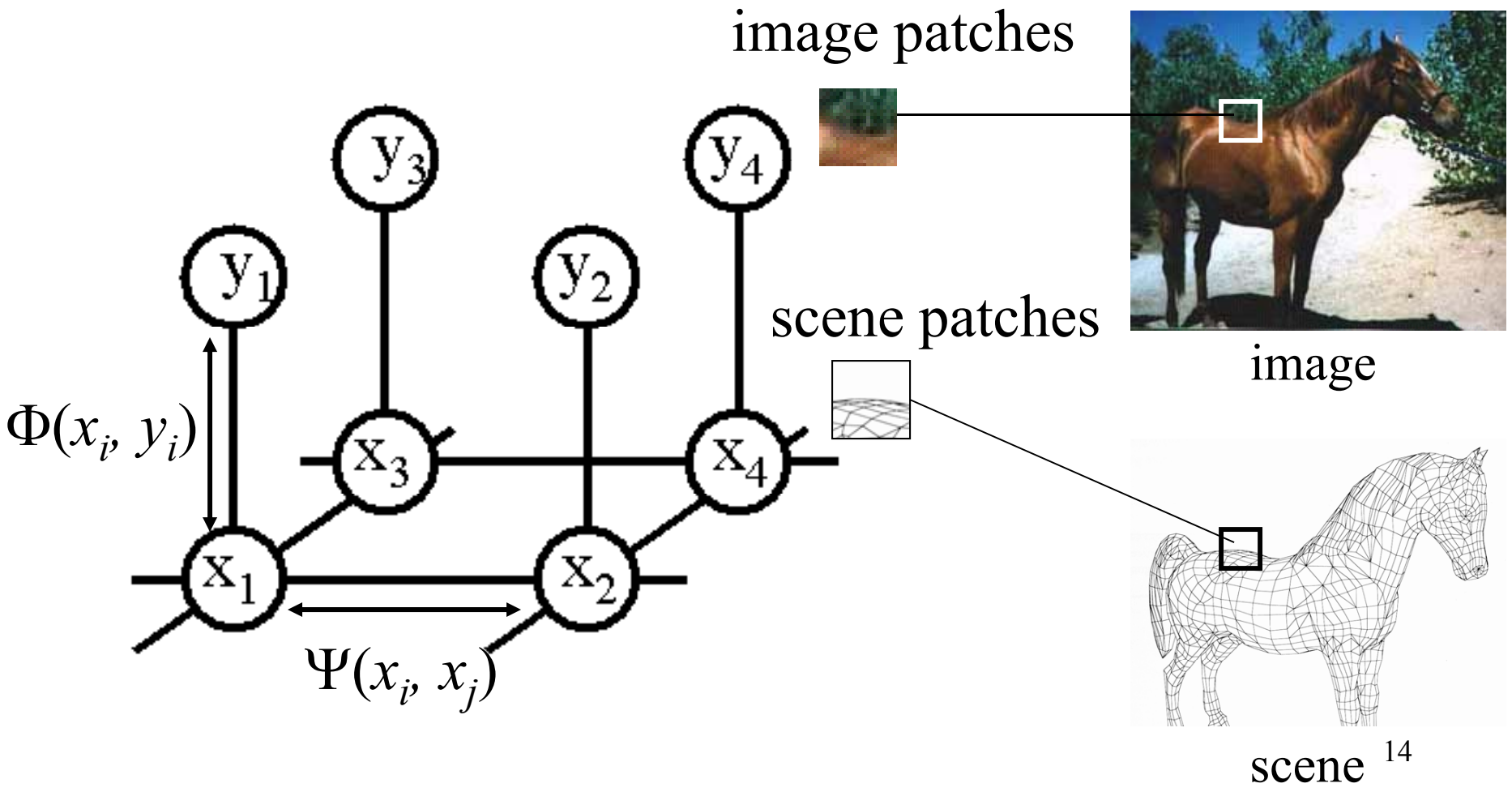


Fig. 2.3. Smoothing with the wrong prior. (**a**) Original, (**b**) degraded image, (**c**) MAP estimate $\beta = 1$, (**d**) MAP estimate $\beta = 0.3$, (**e**) median filter

Winkler, 1995, p. 32

MRF nodes as patches



Network joint probability

$$P(x, y) = \frac{1}{Z} \prod_{i,j} \Psi(x_i, x_j) \prod_{i} \Phi(x_i, y_i)$$

scene

image

Scene-scene compatibility function

neighboring scene nodes

Image-scene compatibility function

local observations

In order to use MRFs:

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

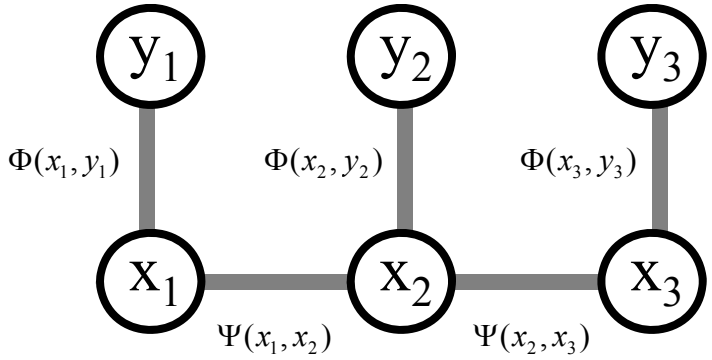
Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated condtional modes (ICM)
  - Variational methods
  - Belief propagation
  - Graph cuts
- Vision applications of inference in MRF's.
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated condtional modes (ICM)
  - Variational methods
  - Belief propagation
  - Graph cuts
- Vision applications of inference in MRF's.
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Derivation of belief propagation



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

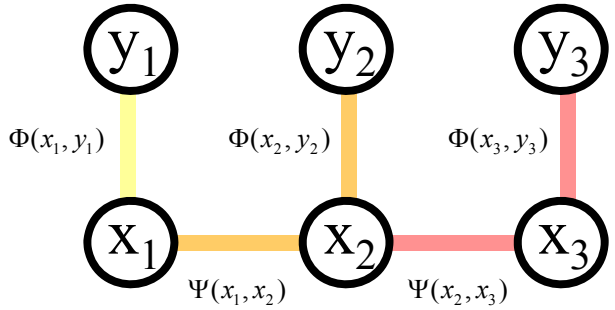
The posterior factorizes

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

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

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

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



Propagation rules

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

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

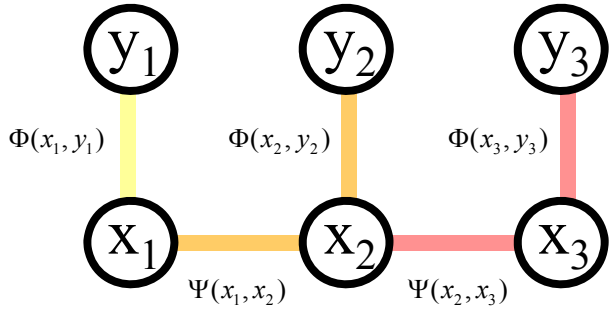
$$\Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

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

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

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

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



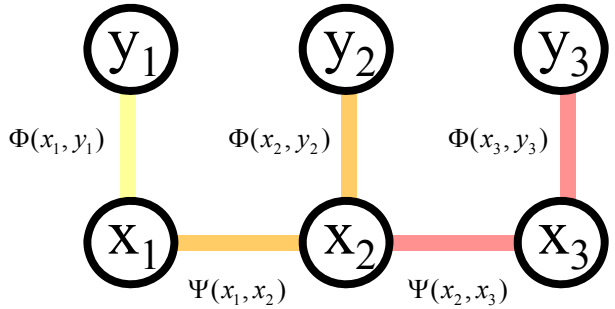
Propagation rules

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

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

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

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



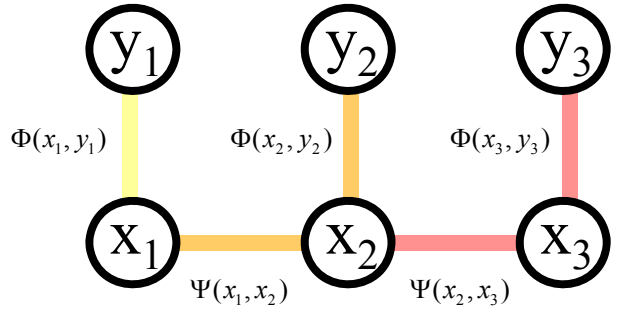
Propagation rules

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

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

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

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



Belief propagation: the nosey neighbor rule

"Given everything that I know, here's what I think you should think"

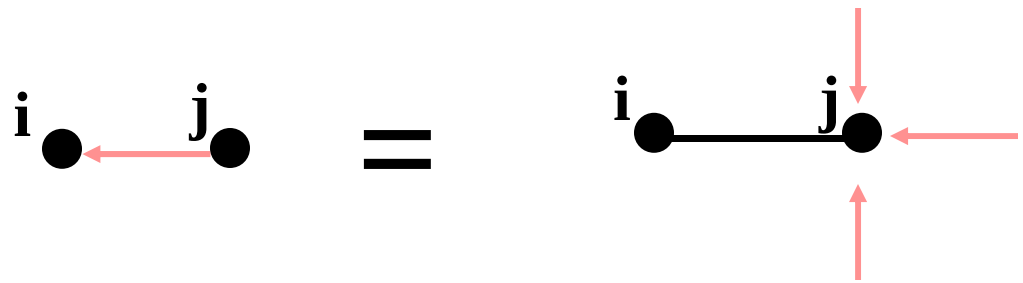
(Given the probabilities of my being in different states, and how my states relate to your states, here's what I think the probabilities of your states should be)

Belief propagation messages

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

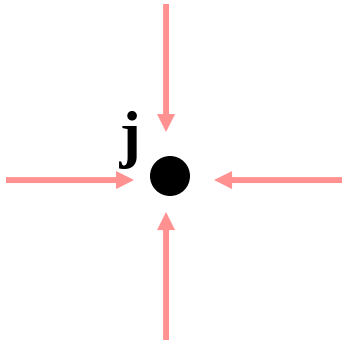
To send a message: Multiply together all the incoming messages, except from the node you're sending to, then multiply by the compatibility matrix and marginalize over the sender's states.

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



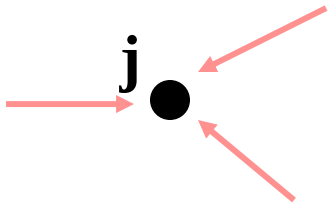
Beliefs

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



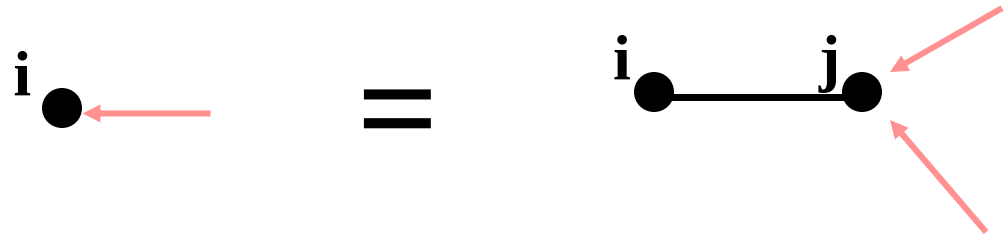
$$b_j(x_j) = \prod_{k \in N(j)} M_j^k(x_j)$$

Belief, and message updates



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

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



Optimal solution in a chain or tree: Belief Propagation

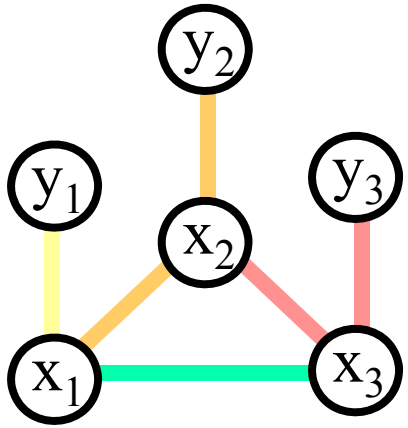
- "Do the right thing" Bayesian algorithm.
- For Gaussian random variables over time: Kalman filter.
- For hidden Markov models: forward/backward algorithm (and MAP variant is Viterbi).

No factorization with loops!

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

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

$$\underset{x_3}{\text{sum}}\ \Phi(x_3, y_3)\ \Psi(x_2, x_3)\ \ \Psi(x_1, x_3)$$



Justification for running belief propagation in networks with loops

- Experimental results:
  - 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

Results from Bethe free energy analysis

- Fixed point of belief propagation equations iff. Bethe approximation stationary point.
- Belief propagation always has a fixed point.
- Connection with variational methods for inference: both minimize approximations to Free Energy,
  - **variational**: usually use primal variables.
  - **belief propagation**: fixed pt. equs. for dual variables.
- Kikuchi approximations lead to more accurate belief propagation algorithms.
- Other Bethe free energy minimization algorithms—Yuille, Welling, etc.

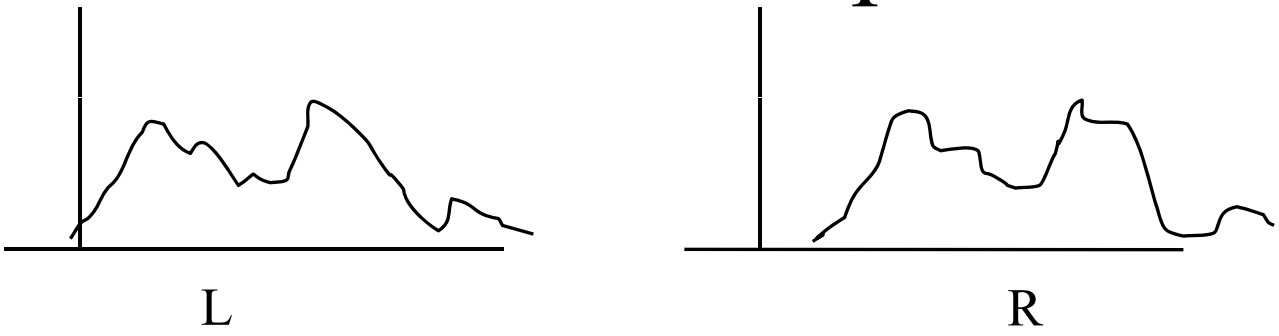
References on BP and GBP

- J. Pearl, 1985
  - classic
- Y. Weiss, NIPS 1998
  - Inspires application of BP to vision
- W. Freeman et al learning low-level vision, IJCV 1999
  - Applications in super-resolution, motion, shading/paint discrimination
- H. Shum et al, ECCV 2002
  - Application to stereo
- M. Wainwright, T. Jaakkola, A. Willsky
  - Reparameterization version
- J. Yedidia, AAAI 2000
  - The clearest place to read about BP and GBP.

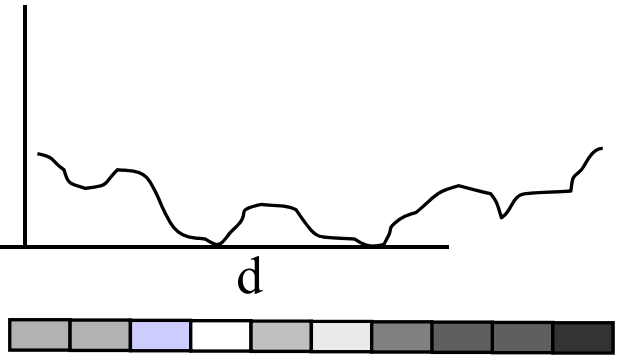
Outline of MRF section

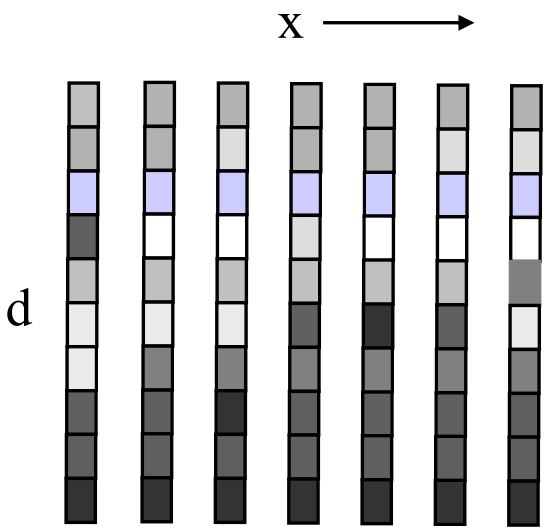
- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Belief propagation
    - Application example—super-resolution
  - Graph cuts
    - Application example--stereo
  - Variational methods
    - Application example—blind deconvolution
- Learning MRF parameters.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

Stereo problem



Squared difference, $(L[x] - R[x-d])^2$, for some x.





Showing local disparity evidence vectors for a set of neighboring positions, x.

MRF for stereo

Disparity estimate at each pixel

Local evidence for each disparity based on intensity match to neighboring frame

$$P(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) = \prod_{(i,j)} \Psi(x_i, x_j) \prod_{p} \Phi(x_p, y_p) \quad (1)$$

where N is the number of nodes, (i, j) represent a pair of neighboring nodes, x_n is the variable at location n, and y_n is the variable representing the intensity differences. The y

Show Middlebury stereo evaluation web pages.

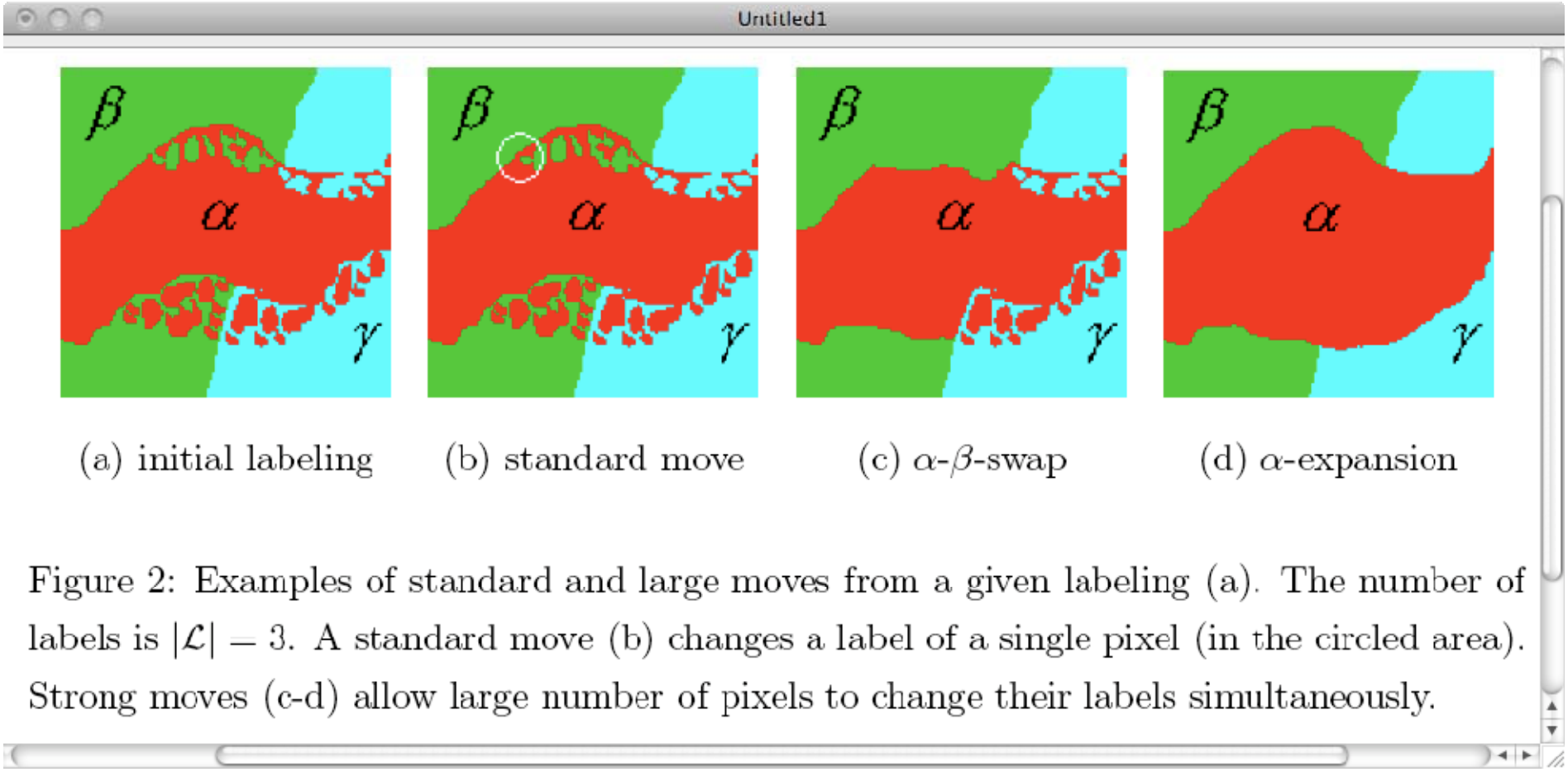
Log probability, or energy function, to be optimized for MAP estimation in MRF's

$$E = E_{data} + E_{smooth}$$

$$= \sum_{p} D_p(f_p) + \sum_{\{p,q\} \in N} V_{p,q}(f_p, f_q)$$

Graph cuts

- Algorithm: uses node label swaps or expansions as moves in the algorithm to reduce the energy. Swaps many labels at once, not just one at a time, as with ICM.
- Find which pixel labels to swap using min cut/max flow algorithms from network theory.
- Can offer bounds on optimality.
- See Boykov, Veksler, Zabih, IEEE PAMI 23 (11) Nov. 2001 (available on web).



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Figure 2: Examples of standard and large moves from a given labeling (a). The number of labels is $|\mathcal{L}| = 3$. A standard move (b) changes a label of a single pixel (in the circled area). Strong moves (c-d) allow large number of pixels to change their labels simultaneously.

Two moves: swap and expansion

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f (Section 3)
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

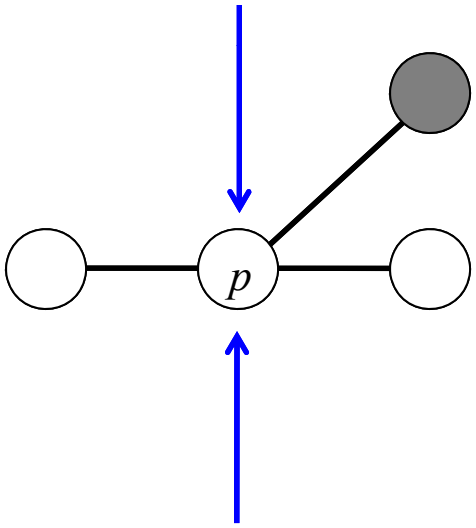
---

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f (Section 4)
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

The cost of an alpha-beta assignment

for each node in $P_{\alpha\beta}$ labeled α:

$$D_p(\alpha) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\alpha, f_q)$$



not a member of $P_{\alpha\beta}$

a member of $P_{\alpha\beta}$

for each connected $P_{\alpha\beta}$ neighbor:

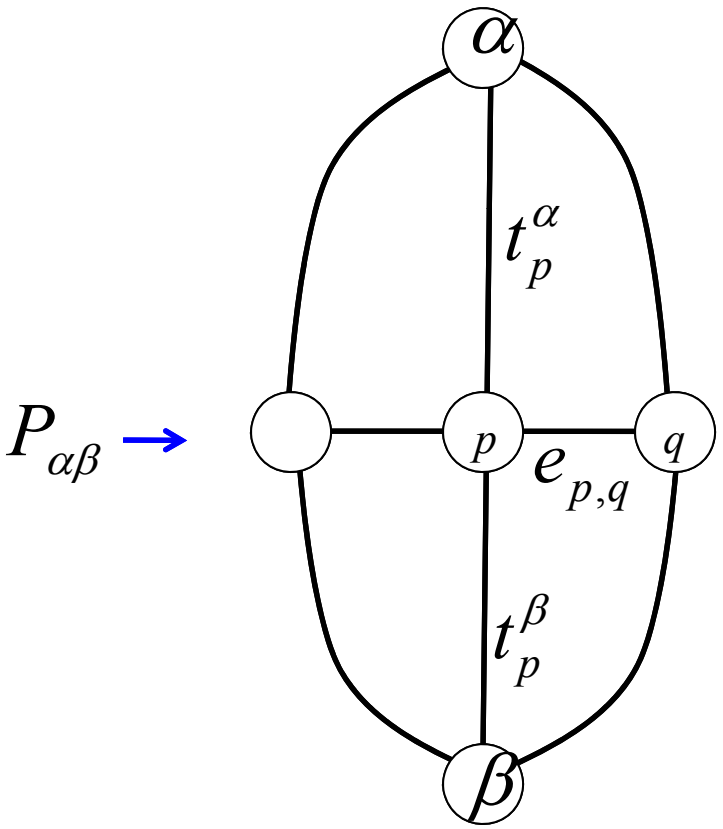
$$+V_{p,q}(\alpha,\beta)$$

$$D_p(\beta) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\beta, f_q)$$

for each node in $P_{\alpha\beta}$ labeled β:

Flash of inspiration here…

The cost of any possible alpha-beta assignment described as a cut in a graph



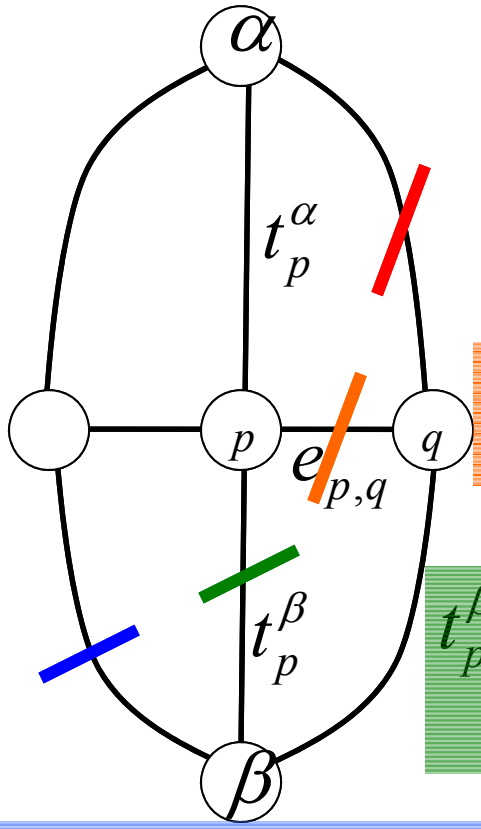
Definitions of graph edge weights:

$$t_p^{\alpha} = D_p(\alpha) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\alpha, f_q)$$

$$e_{p,q} = V_{p,q}(\alpha, \beta)$$

$$t_p^{\beta} = D_p(\beta) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\beta, f_q)$$

The cost of any possible alpha-beta assignment described as a cut in this graph, with edge-weights defined depending on the MRF energies.



$$t_q^\alpha = D_q(\alpha) + \sum_{r \in N_q, r \notin P_{\alpha\beta}} V_{q,r}(\alpha, f_r)$$

$$e_{p,q} = V_{p,q}(\alpha, \beta)$$

$$t_p^\beta = D_p(\beta) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\beta, f_q)$$

$$t_p^\beta = D_p(\beta) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} V_{p,q}(\beta, f_q)$$

next two sections.

3 Finding the optimal swap move

Given an input labeling f (partition $\mathbf{P}$) and a pair of labels α, β, we wish to find a labeling $\hat{f}$ that minimizes E over all labelings within one α-β swap of f. This is the critical step in the algorithm given at the top of Figure 1. Our technique is based on comput-

n-links). The weights assigned to the edges are

edge	weight	for
t_p^α	$D_p(\alpha) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V_{\{p,q\}}(\alpha, \beta)$	$\{p,q\} \in \mathcal{N}$ $p,q \in \mathcal{P}_{\alpha\beta}$

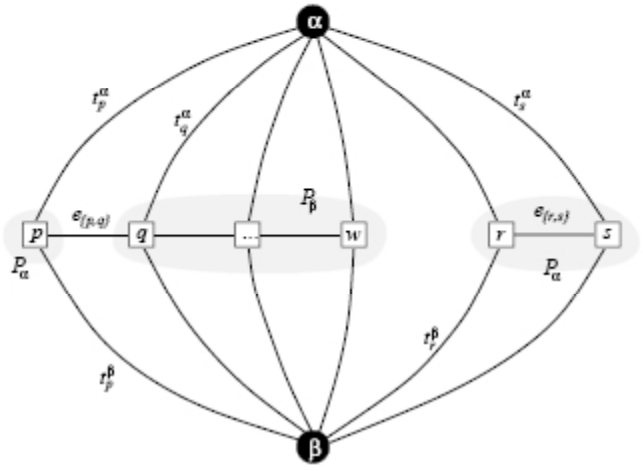


Figure 2: An example of the graph $\mathcal{G}_{\alpha\beta}$ for a 1D image. The set of pixels in the image is $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$ where $\mathcal{P}_\alpha = \{p, r, s\}$ and $\mathcal{P}_\beta = \{q, \ldots, w\}$.

Corollary 1 *The optimal α-β swap from f is $\hat{f} = f^{\mathcal{C}}$ where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_{\alpha\beta}$.*

4 Finding the optimal expansion move

Given an input labeling f (partition $\mathbf{P}$) and a label α, we wish to find a labeling $\hat{f}$ that minimizes E over all labelings within one α-expansion of f. This is the critical step in the algorithm given at the bottom of Figure 1. In this section we describe a technique that solves the problem assuming that each $V_{\{p,q\}}$ is a metric, and thus satisfies the triangle inequality (2). Some important examples of metrics are given in the introduction. Our technique is based on computing a labeling corresponding to a minimum cut on a graph $\mathcal{G}_\alpha = \langle \mathcal{V}_\alpha, \mathcal{E}_\alpha \rangle$. The structure of this graph is determined

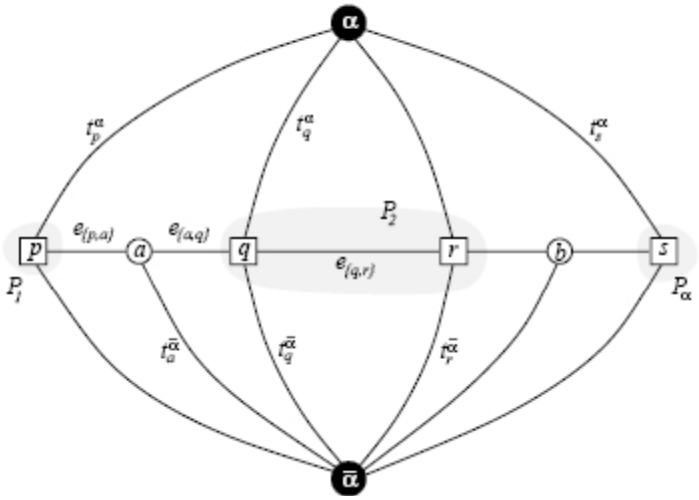


Figure 4: An example of $\mathcal{G}_\alpha$ for a 1D image. The set of pixels in the image is $\mathcal{P} = \{p, q, r, s\}$ and the current partition is $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_\alpha\}$ where $\mathcal{P}_1 = \{p\}$, $\mathcal{P}_2 = \{q, r\}$, and $\mathcal{P}_\alpha = \{s\}$. Two auxiliary nodes $a = a_{\{p,q\}}$, $b = a_{\{r,s\}}$ are introduced between neighboring pixels separated in the current partition. Auxiliary nodes are added at the boundary of sets $\mathcal{P}_l$.

The weights assigned to the edges are

edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$D_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$D_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$e_{\{a,q\}}$	$V_{\{p,q\}}(\alpha, f_q)$	
$t_a^{\bar{\alpha}}$	$V_{\{p,q\}}(f_p, f_q)$	
$e_{\{p,q\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

Corollary 2 *The optimal α expansion from f is $\hat{f} = f^{\mathcal{C}}$ where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_\alpha$.*

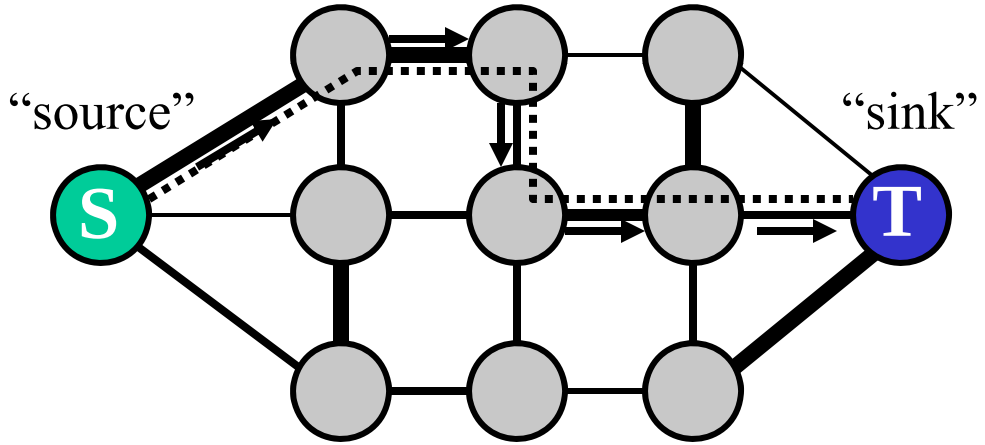
Slide by Yuri Boykov

Minimum *s*-*t* cuts algorithms

- Augmenting paths [Ford & Fulkerson, 1962]
- Push-relabel [Goldberg-Tarjan, 1986]

Slide by Yuri Boykov

"Augmenting Paths"

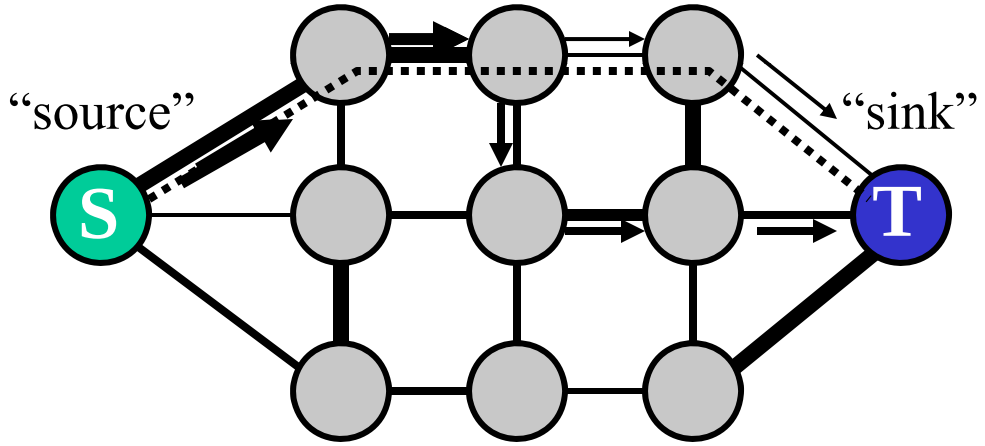


A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

Slide by Yuri Boykov

"Augmenting Paths"

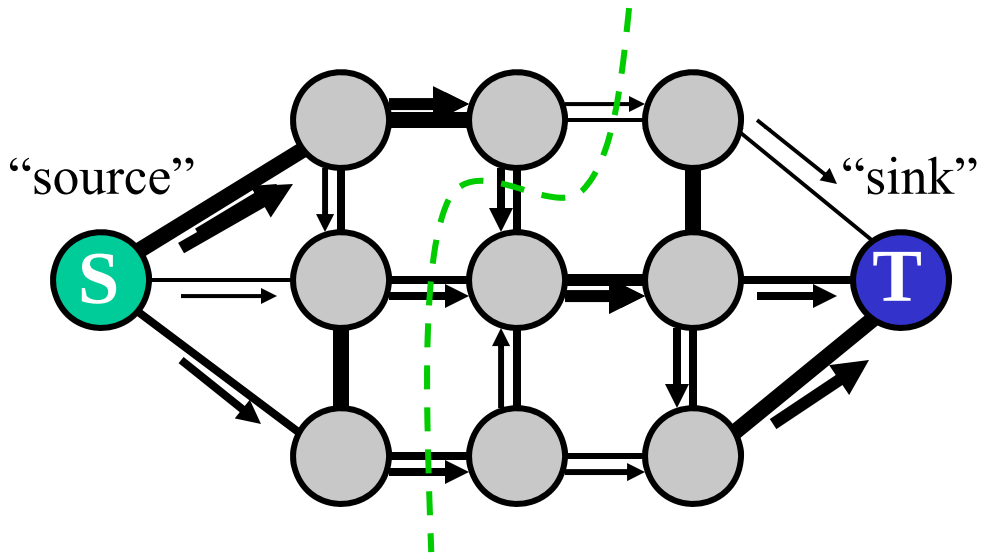


A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates
- Find next path…
- Increase flow…

Slide by Yuri Boykov

"Augmenting Paths"



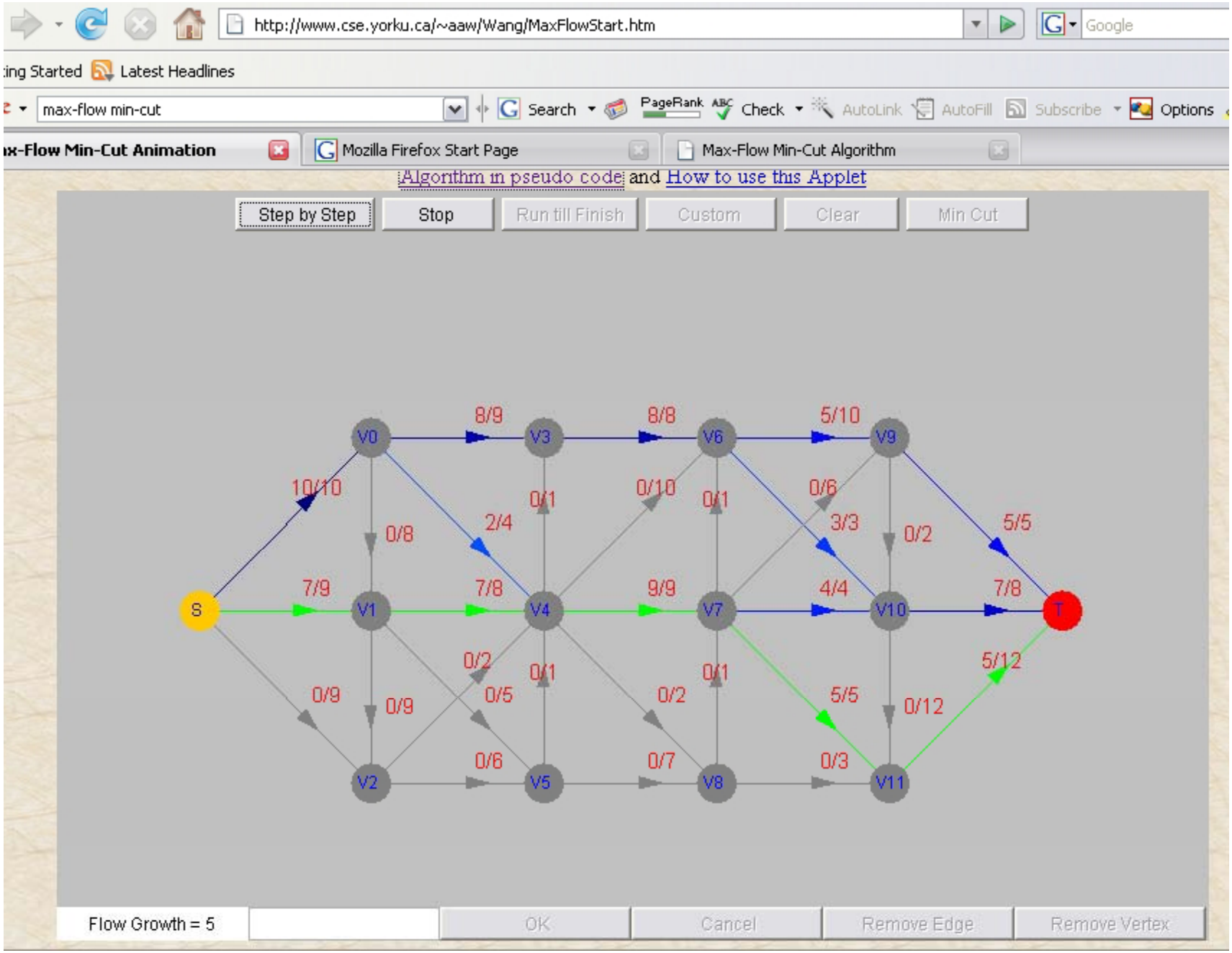
A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

Iterate until … all paths from S to T have at least one saturated edge

MAX FLOW ⇔ MIN CUT

Break between the two sets always cutting connections that are at capacity


http://www.cse.yorku.ca/~aaw/Wang/MaxFlowStart.htm
Google
ing Started
Latest Headlines
max-flow min-cut
Search
PageRank
Check
AutoLink
AutoFill
Subscribe
Options
x-Flow Min-Cut Animation
Mozilla Firefox Start Page
Max-Flow Min-Cut Algorithm
Algorithm in pseudo code and How to use this Applet
Step by Step
Stop
Run till Finish
Custom
Clear
Min Cut
S
V0
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
T
8/9
8/8
5/10
10/10
0/1
0/10
0/1
0/6
0/8
2/4
3/3
0/2
5/5
7/9
7/8
9/9
4/4
7/8
0/2
0/1
0/1
5/12
0/9
0/9
0/5
0/2
5/5
0/12
0/6
0/7
0/3
Flow Growth = 5
OK
Cancel
Remove Edge
Remove Vertex

ICCV 2001

Fast Approximate Energy Minimization via Graph Cuts

Yuri Boykov Olga Veksler Ramin Zabih
Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract

In this paper we address the problem of minimizing large class of energy functions that occur in early

piecewise smooth, while E_{data} measures the disagree ment between f and the observed data. Many differ ent energy functions have been proposed in the liter

http://citeseer.ist.psu.edu/rd/0%2C251059%2C1%2C0.25%2CDownload/http://coblitz.codeen.org:3125/citeseer.ist.psu.edu/cache/papers/cs/11281/http:zSzzSzwww.cs.cornell.eduzSzrdzzSzPaperszSzBVZ-iccv99.pdf/boykov99fast.pdf

These vision problems can be naturally formulated in terms of energy minimization. In this framework, one seeks the labeling f that minimizes the energy

$$E(f) \quad = \quad E_{smooth}(f) \quad + \quad E_{data}(f).$$

piecewise smooth, while E_{data} measures the disagreement between f and the observed data. Many different energy functions have been proposed in the literature. The form of E_{data} is typically

$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p),$$

where D_p measures how appropriate a label is for the pixel p given the observed data. In image restoration, for example, $D_p(f_p)$ is typically $(f_p - i_p)^2$, where i_p is the observed intensity of the pixel p.

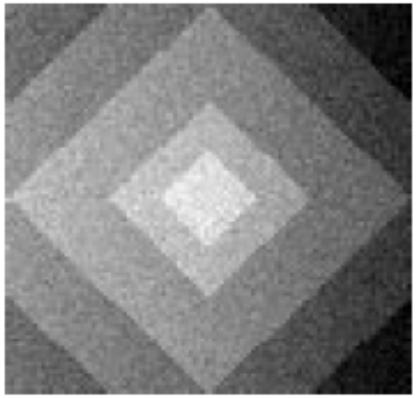
arbitrary, and consider smoothing terms of the form

$$E_{smooth} \quad = \quad \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q), \qquad (1)$$

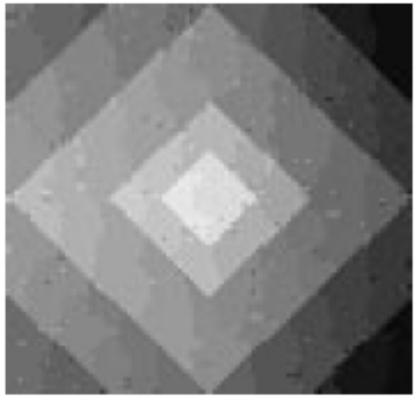
where $\mathcal{N}$ is the set of pairs of adjacent pixels. In spe-

	E			E_{smooth}		
	Our results	Annealing	Ratio	Our results	Annealing	Ratio
Diamond (image restoration, E_1)						
First cycle ($t = 36$)	1,577	55,892	**35.5**	637	9,658	**15.2**
Last cycle ($t = 389$)	1,472	15,215	**10.3**	576	8,475	**14.7**
Best annealing ($t = 417,317$)	—	1,458	—	—	571	—
Tree image (motion, E_1)						
First cycle ($t = 29$)	2,591	3,604	**1.4**	974	2,146	**2.2**
Last cycle ($t = 183$)	2,255	2,449	**1.1**	768	1,078	**1.4**
Best annealing ($t = 56,170$)	—	2,418	—	—	1,050	—
Rock image (stereo, E_3)						
First cycle ($t = 204$)	780	1,635	**2.1**	270	699	**2.6**
Last cycle ($t = 431$)	776	1,557	**2.0**	270	626	**2.3**
Best annealing ($t = 56,645$)	—	1,480	—	—	577	—

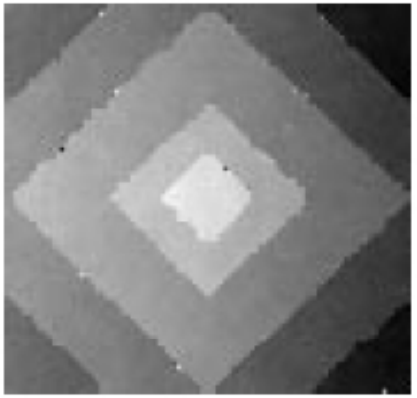
Figure 6: Comparative energy minimization results for our methods and simulated annealing



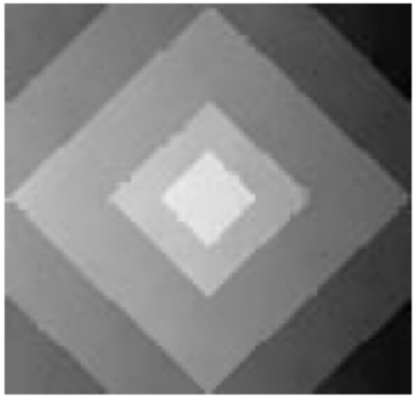
Diamond image (input)



Our method (E_2)



Annealing (E_1)



Our method (E_1)

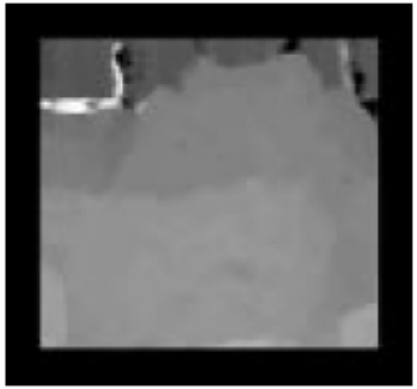
Tree image

Normalized correlation

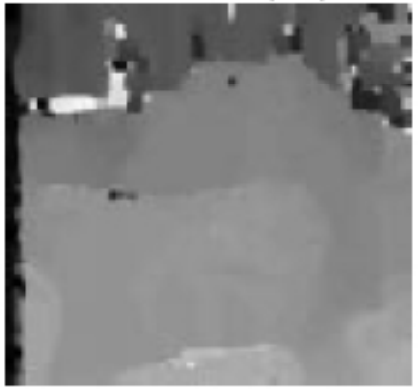
Annealing (E_1)

Our method (E_1)

Rock image



Normalized correlation



Annealing (E_3)

Our method (E_3)

To Boykov slides

http://palantir.swarthmore.edu/cvpr05/CVPR05_tutorial_yyb.ppt

And see:
http://www-static.cc.gatech.edu/gvu/perception/projects/graphcuttextures/#video
And
http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem
And
http://www.cse.yorku.ca/~aaw/Wang/MaxFlowStart.htm

Slide by Yuri Boykov

CVPR'05 Tutorial

Discrete Optimization Methods in Vision

Part III: **Graph Algorithms in Vision**

Yuri Boykov, University of Western Ontario
Pedro Felzenszwalb, University of Chicago
Ramin Zabih, Cornell University

Slide by Yuri Boykov

Outline

- Shortest path algorithm
- *s-t* Graph cuts
  - Hypersurface optimization
  - Binary energy minimization
- N-way cuts / Multi-label energy minimization
  - *a*-expansion algorithm

Applications in Computer Vision

Slide by Yuri Boykov

Shortest paths on graphs (examples)

- Texture Synthesis
  - Image quilting [Efros & Freeman, 2001]

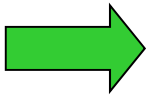
- Object extraction
  - live-wire [Falcao, Udupa, Samarasekera, Sharma 1998]
  - intelligent scissors [Mortensen, Barrett 1998]

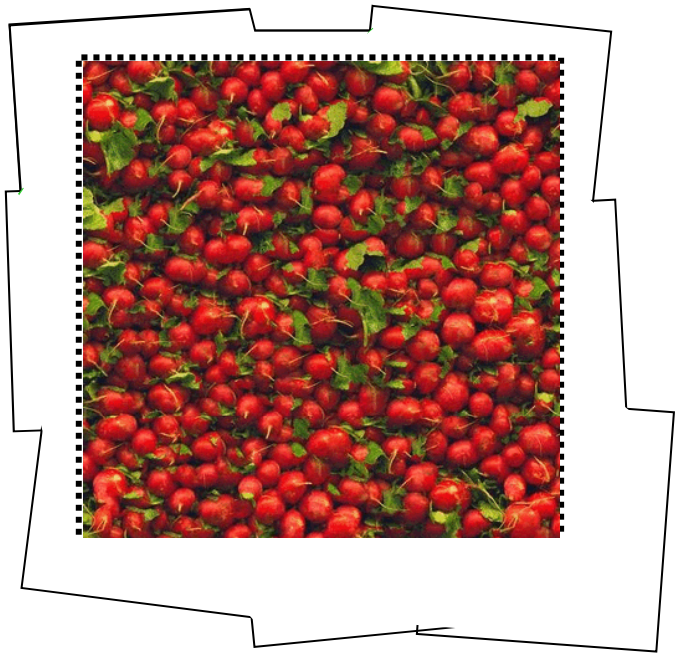
- Scan line stereo
  - Ohta & Kanade, 1985
  - Cox, Hingorani, Rao, 1996

Slide by Yuri Boykov

Shortest paths: Texture synthesis

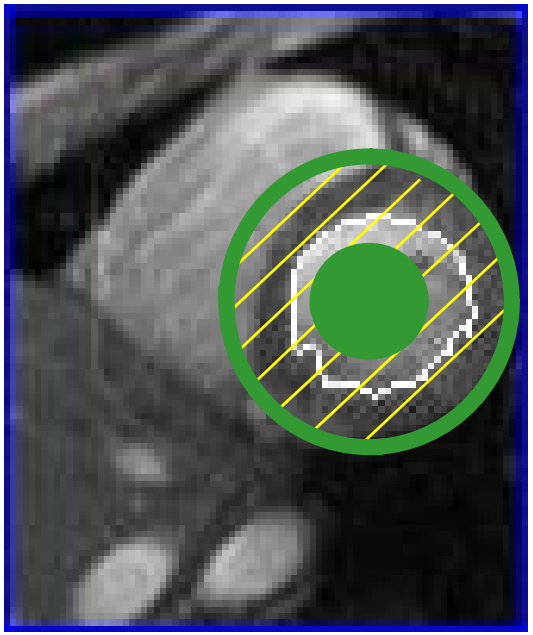
"Image quilting"



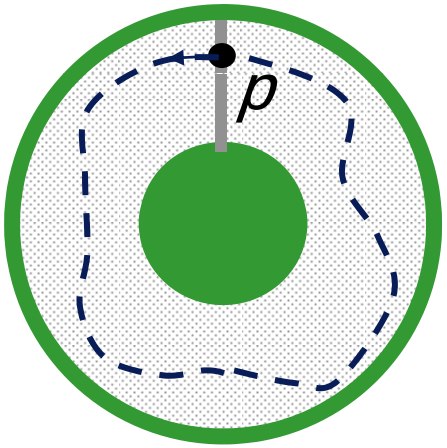


2D Graph cut ⇔ *shortest path on a graph*

Example:
find the shortest closed contour in a given domain of a graph

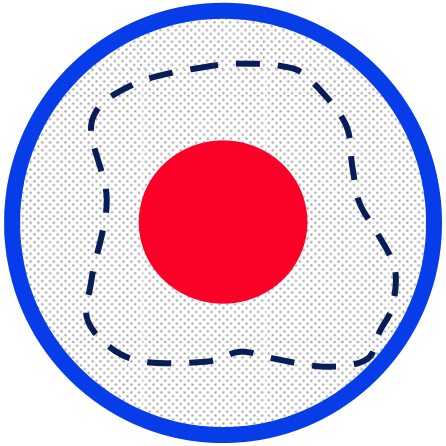


Shortest paths approach
(*live wire*, *intelligent scissors*)



Compute the *shortest path* *p* -> *p* for a point *p*. Repeat for all points on the gray line. Then choose the optimal contour.

Graph Cuts approach



Compute the *minimum cut* that separates red region from blue region

live-wire: [Falcao, Udupa, Samarasekera, Sharma 1998] intelligent scissors: [Mortensen, Barrett 1998]

Slide by Yuri Boykov

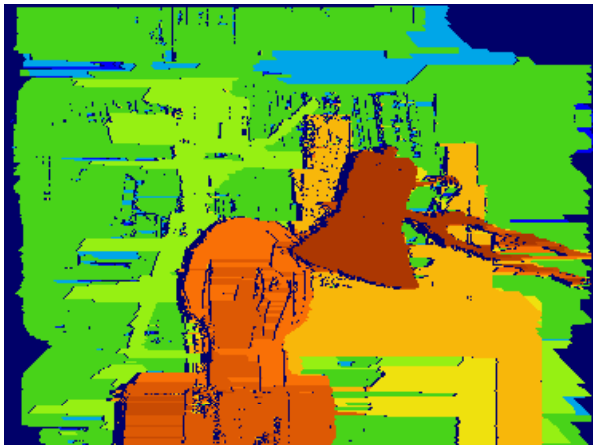
DP / Shortest-paths

- Efficient global optimization tools ☺
- Successes are primarily in 1D problems ☹
  - Can't deal with dependent scan-lines in stereo
  - Can't handle bubbles (3D snakes) or object boundaries in volumetric data

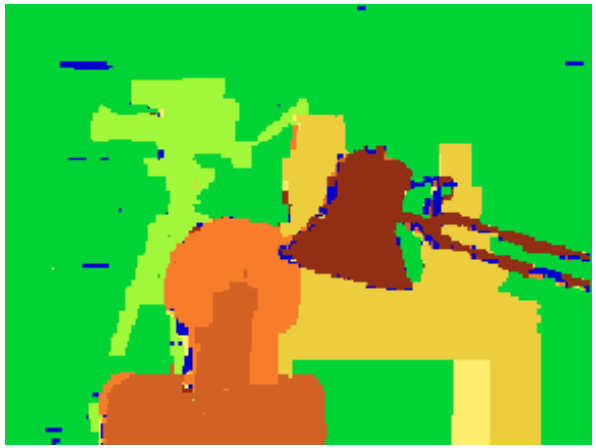
- Graph Cuts can be seen as a "generalization" to N-D

Slide by Yuri Boykov

Stereo example



Independent
scan-lines
(via DP)



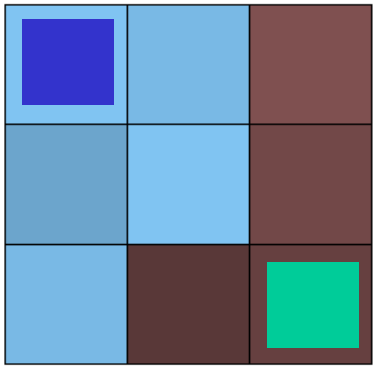
Multi-scan line
(via Graph Cuts)

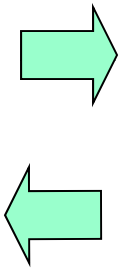
Ground truth

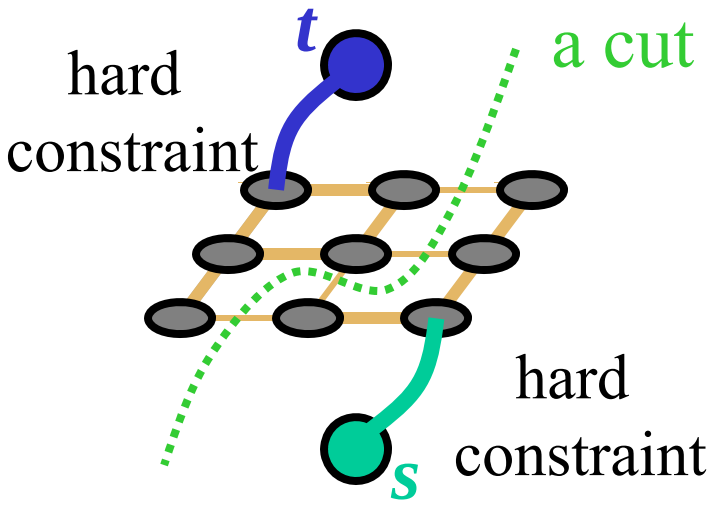
Slide by Yuri Boykov

Graph cuts for segmentation

(simple example à la Boykov&Jolly, ICCV'01)



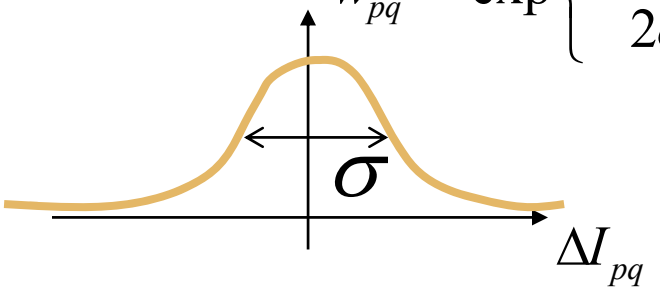




Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

$$w_{pq} = \exp\left\{-\frac{\Delta I_{pq}}{2\sigma^2}\right\}$$



Slide by Yuri Boykov

Examples of Graph Cuts in vision

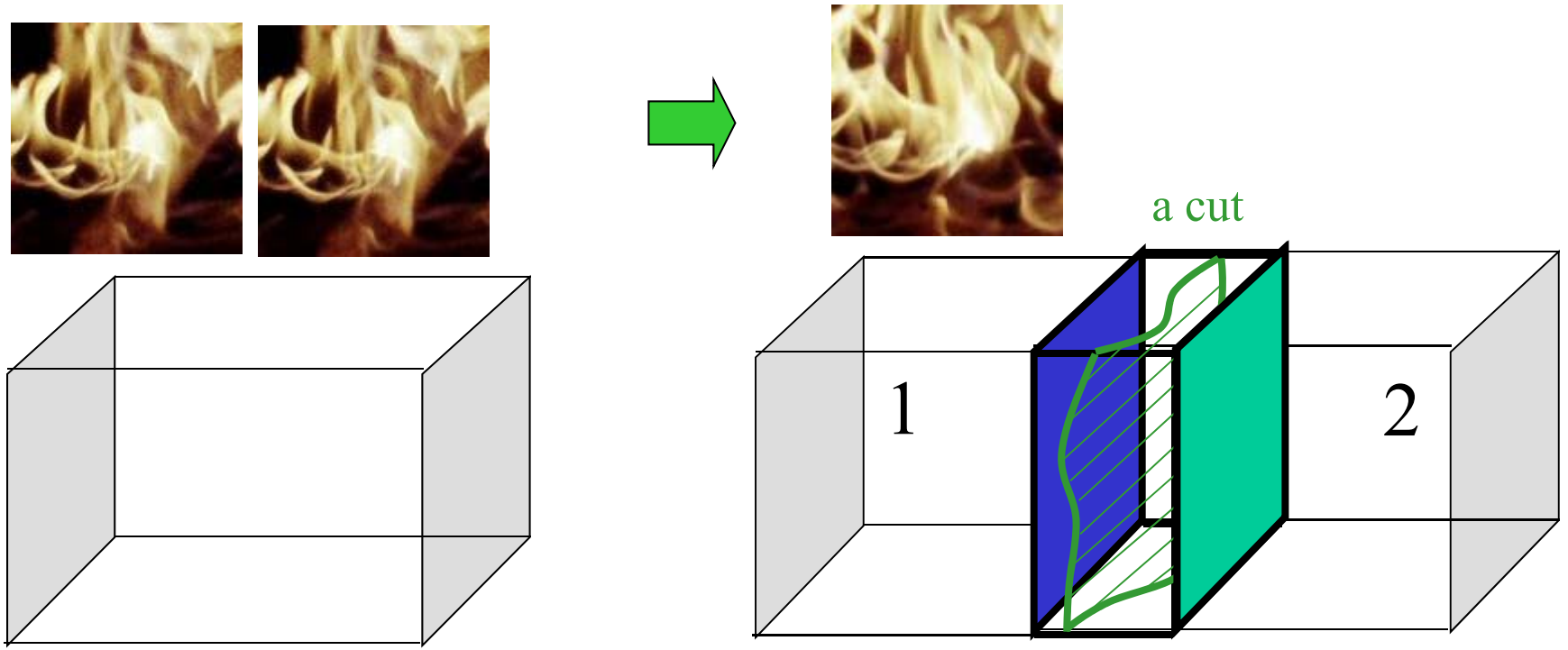
- Image Restoration (e.g. Greig at.al. 1989)
- Segmentation
  - Wu & Leahy 1993
  - Nested Cuts, Veksler 2000
- Multi-scan-line Stereo, Multi-camera stereo
  - Roy & Cox 1998, 1999
  - Ishikawa & Geiger 1998, 2003
  - Boykov, Veksler, Zabih 1998, 2001
  - Kolmogorov & Zabih 2002, 2004
- Object Matching/Recognition (Boykov & Huttenlocher 1999)
- N-D Object extraction (photo-video editing, medical imaging)
  - Boykov, Jolly, Funka-Lea 2000, 2001, 2004
  - Boykov & Kolmogorov 2003
  - Rother, Blake, Kolmogorov 2004
- Texture synthesis (Kwatra, Schodl, Essa, Bobick 2003)
- Shape reconstruction (Snow,Viola,Zabih 2000)
- Motion (e.g. Xiao, Shah 2004)

Slide by Yuri Boykov

s-t graph cuts for video textures

http://www-static.cc.gatech.edu/gvu/perception/projects/graphcuttextures/#video

Graph cuts video textures



Short video clip

Long video clip

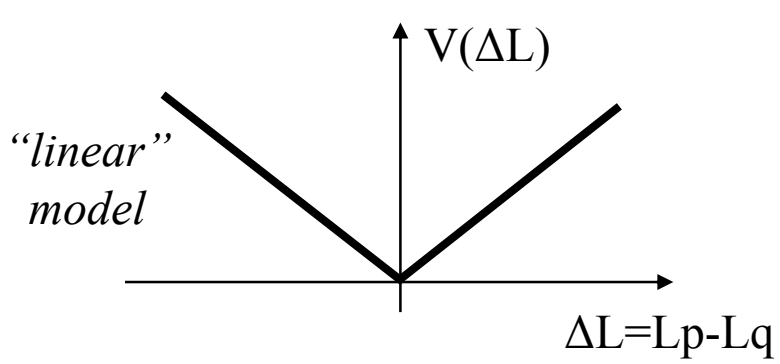
3D generalization of "**image-quilting**"

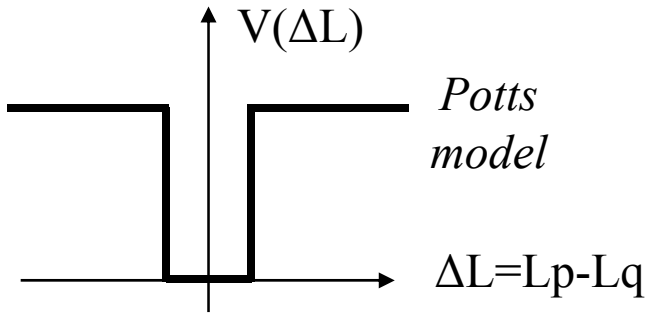
Slide by Yuri Boykov

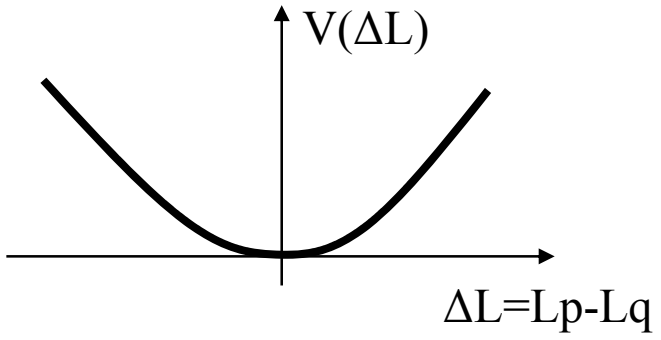
Pixel interactions: "*convex*" vs. "*discontinuity-preserving*"

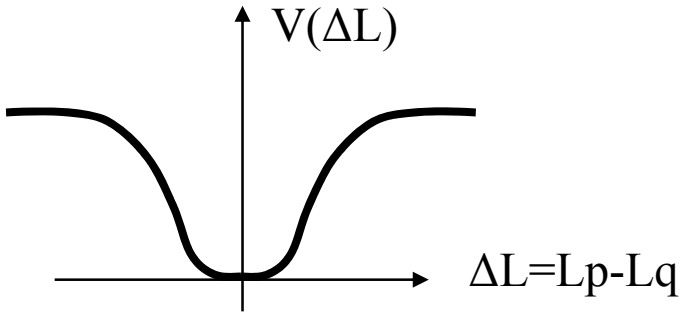
"Convex" interactions

Robust "discontinuity preserving" interactions



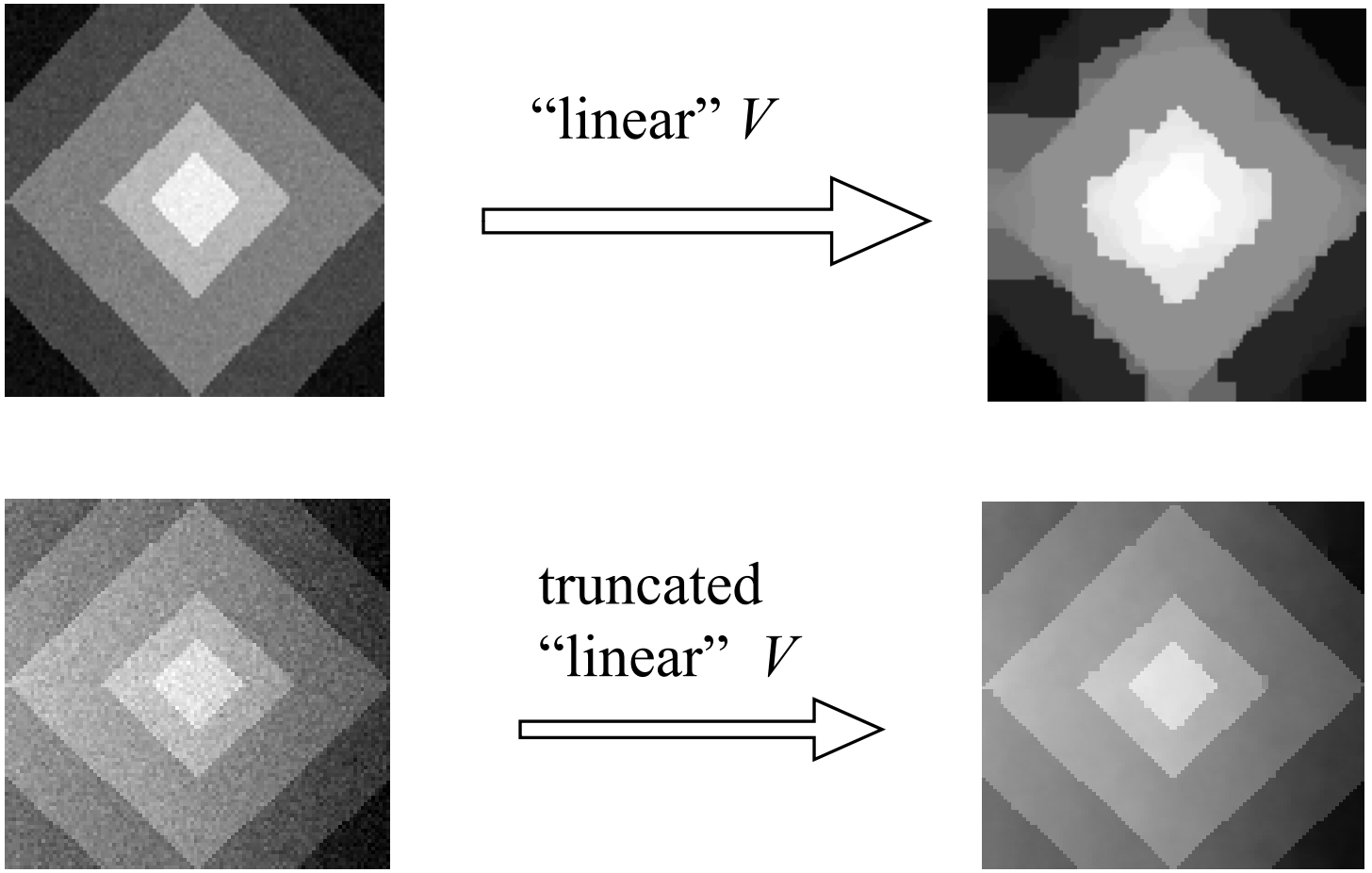






Slide by Yuri Boykov

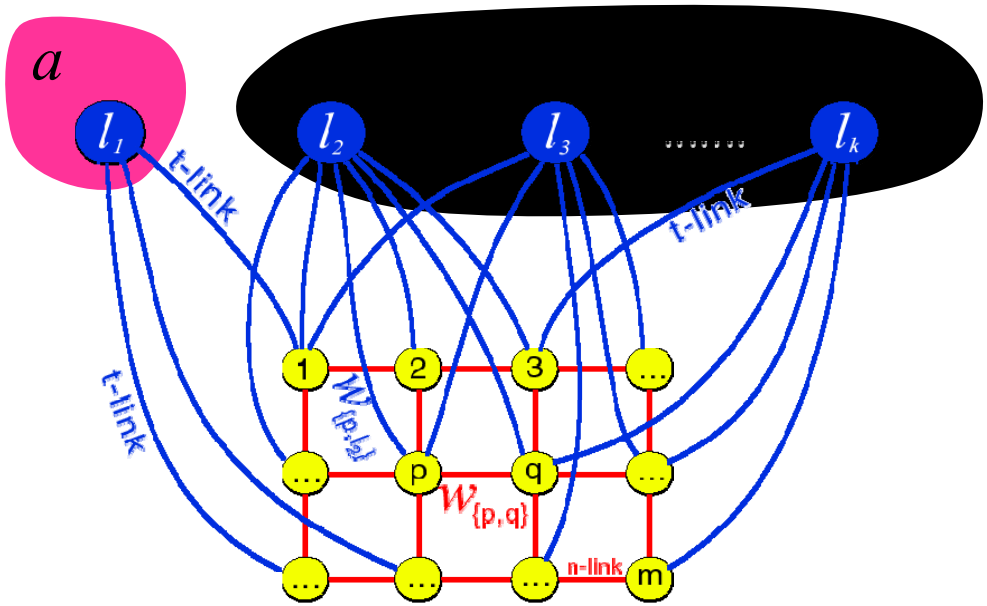
Pixel interactions: "*convex*" vs. "*discontinuity-preserving*"



Slide by Yuri Boykov

a-expansion move

Basic idea: break multi-way cut computation into a sequence of binary *s*-*t* cuts



Iteratively, each label competes with the other labels for space in the image

Slide by Yuri Boykov

a-expansion moves

In each *a*-expansion a given label "*a*" grabs space from other labels

initial solution

-expansion

-expansion

-expansion

-expansion

-expansion

-expansion

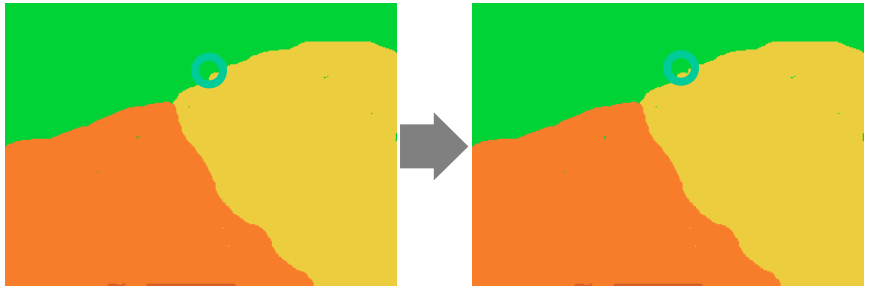
-expansion

For each move we choose expansion that gives the largest decrease in the energy:
binary optimization problem

Slide by Yuri Boykov

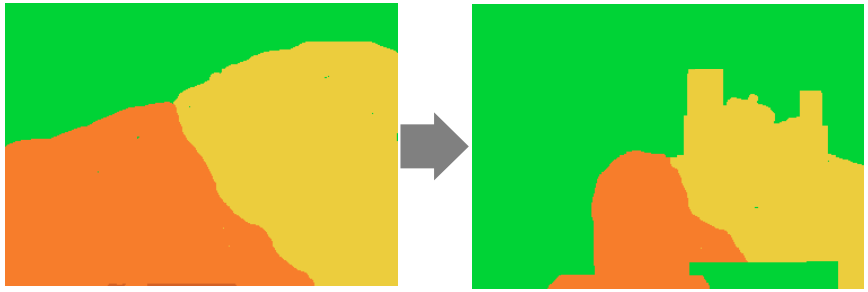
a-expansion algorithm vs. standard discrete energy minimization techniques

single "one-pixel" move
(simulated annealing, ICM,…)



- Only one pixel can change its label at a time
- Finding an optimal move is computationally trivial

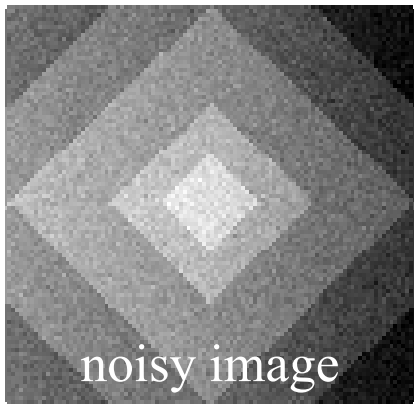
single *a*-expansion move



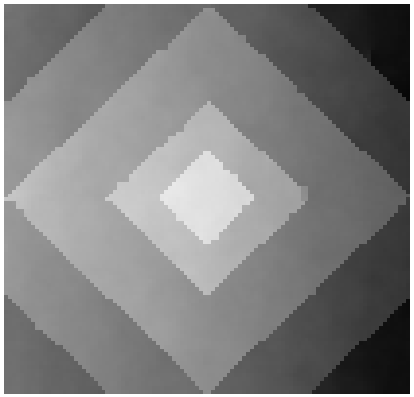
- Large number of pixels can change their labels simultaneously
- Finding an optimal move is computationally intensive O(2^n)
  (*s*-*t* cuts)

Slide by Yuri Boykov

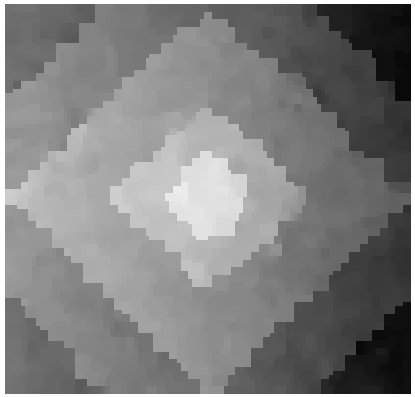
a-expansion move vs. "standard" moves



Potts energy minimization



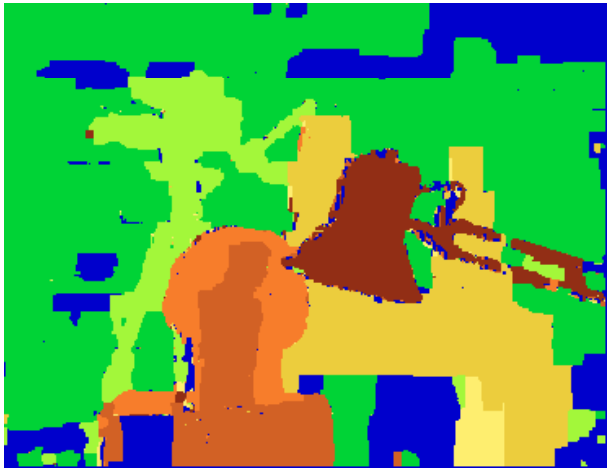
a local minimum
w.r.t. expansion moves



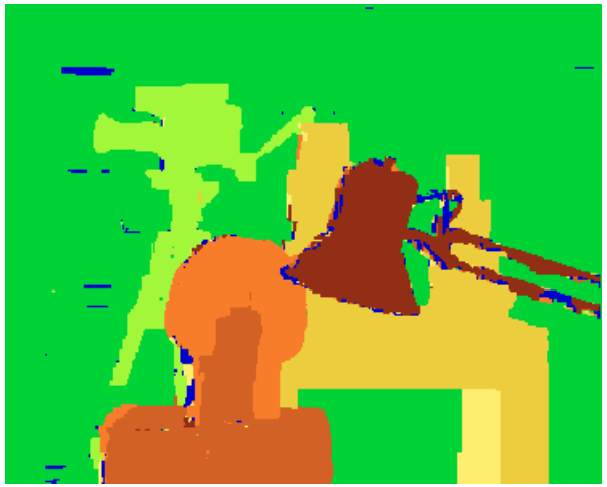
a local minimum
w.r.t. "one-pixel" moves

Slide by Yuri Boykov

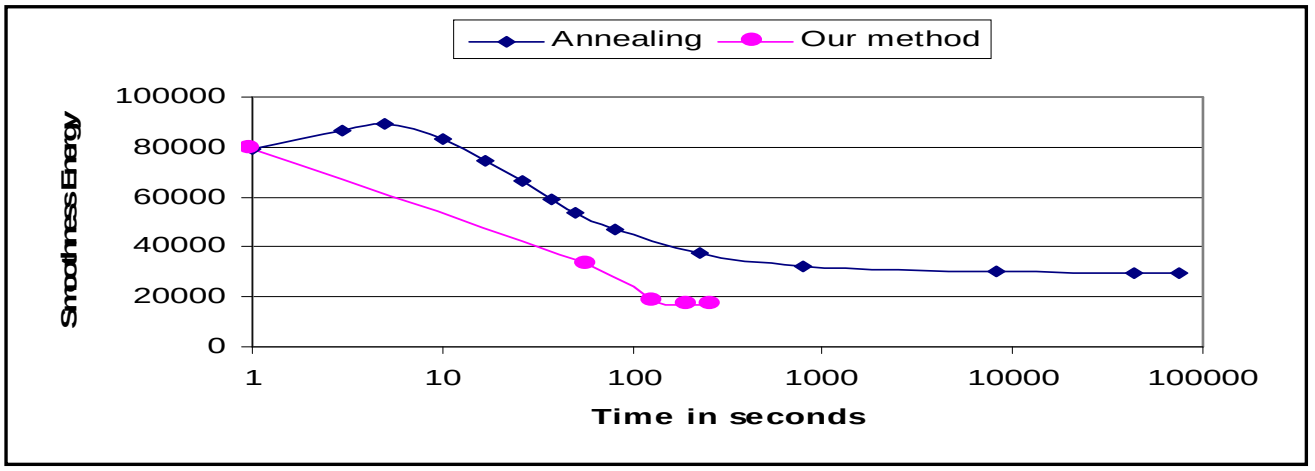
a-expansions vs. simulated annealing



simulated annealing,
start from scratch, 20 hours, 24.7% err



a-expansions (BVZ 89,01)
90 seconds, 5.8% err



Slide by Yuri Boykov

a-expansion algorithm vs. local-update algorithms (SA, ICM, ...)

a-expansions	simulated annealing, …
• Finds *local* minimum of energy with respect to very strong *moves*	• Finds *local* minimum of energy with respect to small "one-pixel" *moves*
• In practice, results do not depend on initialization	• Initialization is important
• solution is within the factor of 2 from the global minima	• solution could be arbitrarily far from the global minima
• In practice, one cycle through all labels gives sufficiently good results	• May not know when to stop. Practical complexity may be worse than exhaustive search
• Applies to a restricted class of energies	• Can be applied to anything

end of Boykov slides

Middlebury stereo web page

vision.middlebury.edu/stereo

http://vision.middlebury.edu/stereo/

Most Visited · Getting Started · Latest Headlines · Google · Apple · Yahoo! · YouTube · Wikipedia · News · Popular · Google Maps

vision.middlebury.edu

stereo · mview · MRF · flow

Stereo Evaluation · Datasets · Code · Submit

Daniel Scharstein • Richard Szeliski

Welcome to the Middlebury Stereo Vision Page, formerly located at www.middlebury.edu/stereo. This website accompanies our taxonomy and comparison of two-frame stereo correspondence algorithms [1]. It contains:

- An on-line evaluation of current algorithms
- Many stereo datasets with ground-truth disparities
- Our stereo correspondence software
- An on-line submission script that allows you to evaluate your stereo algorithm in our framework

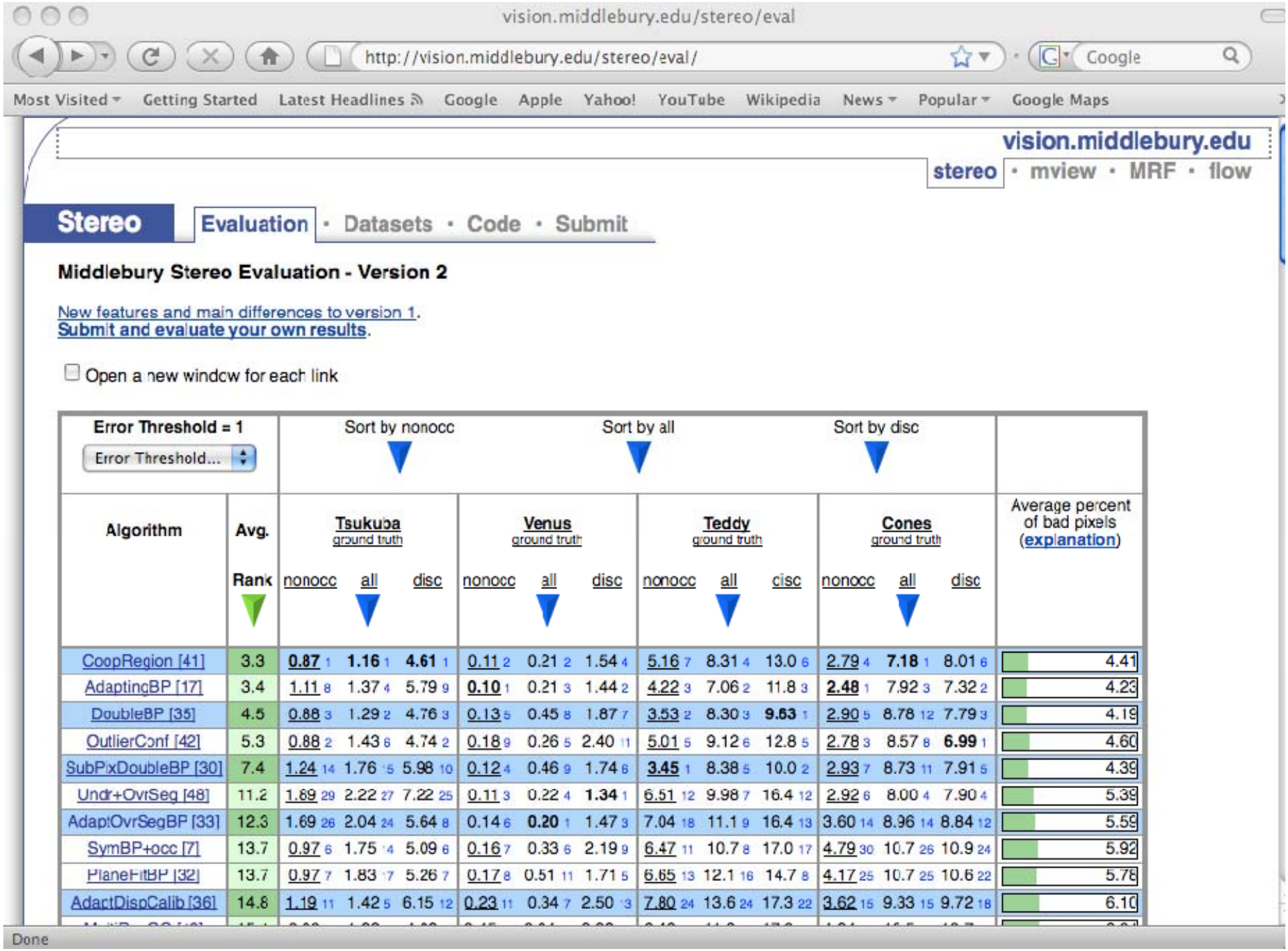
How to cite the materials on this website:
We grant permission to use and publish all images and numerical results on this website. If you report performance results, we request that you cite our paper [1]. Instructions on how to cite our datasets are listed on the datasets page. If you want to cite this website, please use the URL "**vision.middlebury.edu/stereo/**".

References:

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002.
Microsoft Research Technical Report MSR-TR-2001-81, November 2001.

Support for this work was provided in part by NSF CAREER grant 9984485 and NSF grant IIS-0413169. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Done

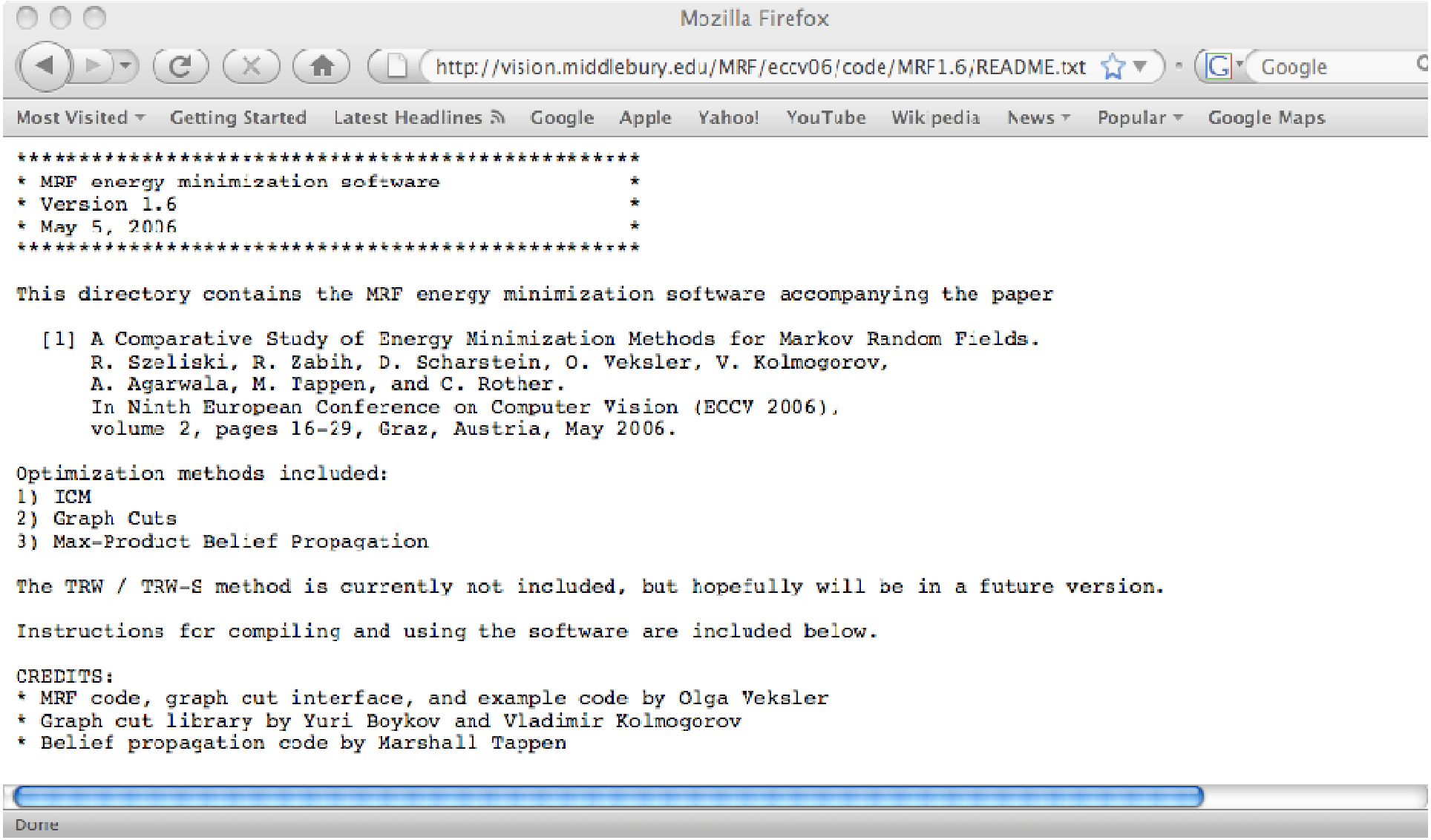

vision.middlebury.edu/stereo/eval
http://vision.middlebury.edu/stereo/eval/
Google
Most Visited
Getting Started
Latest Headlines
Google
Apple
Yahoo!
YouTube
Wikipedia
News
Popular
Google Maps
vision.middlebury.edu
stereo · mview · MRF · flow
Stereo
Evaluation · Datasets · Code · Submit
Middlebury Stereo Evaluation - Version 2
New features and main differences to version 1.
Submit and evaluate your own results.
Open a new window for each link
Error Threshold = 1
Error Threshold...
Sort by nonocc
Sort by all
Sort by disc
Algorithm
Avg. Rank
Tsukuba ground truth
Venus ground truth
Teddy ground truth
Cones ground truth
Average percent of bad pixels (explanation)
nonocc all disc
CoopRegion [41] 3.3 0.87 1 1.16 1 4.61 1 0.11 2 0.21 2 1.54 4 5.16 7 8.31 4 13.0 6 2.79 4 7.18 1 8.01 6 4.41
AdaptingBP [17] 3.4 1.11 8 1.37 4 5.79 9 0.10 1 0.21 3 1.44 2 4.22 3 7.06 2 11.8 3 2.48 1 7.92 3 7.32 2 4.23
DoubleBP [35] 4.5 0.88 3 1.29 2 4.76 3 0.13 5 0.45 8 1.87 7 3.53 2 8.30 3 9.63 1 2.90 5 8.78 12 7.79 3 4.19
OutlierConf [42] 5.3 0.88 2 1.43 6 4.74 2 0.18 9 0.26 5 2.40 11 5.01 5 9.12 6 12.8 5 2.78 3 8.57 8 6.99 1 4.60
SubPixDoubleBP [30] 7.4 1.24 14 1.76 15 5.98 10 0.12 4 0.46 9 1.74 6 3.45 1 8.38 5 10.0 2 2.93 7 8.73 11 7.91 5 4.39
Undr+OvrSeg [48] 11.2 1.69 29 2.22 27 7.22 25 0.11 3 0.22 4 1.34 1 6.51 12 9.98 7 16.4 12 2.92 6 8.00 4 7.90 4 5.39
AdaptOvrSegBP [33] 12.3 1.69 26 2.04 24 5.64 8 0.14 6 0.20 1 1.47 3 7.04 18 11.1 9 16.4 13 3.60 14 8.96 14 8.84 12 5.59
SymBP+occ [7] 13.7 0.97 6 1.75 14 5.09 6 0.16 7 0.33 6 2.19 9 6.47 11 10.7 8 17.0 17 4.79 30 10.7 26 10.9 24 5.92
PlaneFitBP [32] 13.7 0.97 7 1.83 17 5.26 7 0.17 8 0.51 11 1.71 5 6.65 13 12.1 16 14.7 8 4.17 25 10.7 25 10.6 22 5.78
AdaptDispCalib [36] 14.8 1.19 11 1.42 5 6.15 12 0.23 11 0.34 7 2.50 13 7.80 24 13.6 24 17.3 22 3.62 15 9.33 15 9.72 18 6.10
Done

A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors

Richard Szeliski, *Fellow*, *IEEE*, Ramin Zabih, *Senior Member*, *IEEE*, Daniel Scharstein, *Member*, *IEEE*, Olga Veksler, *Member*, *IEEE*, Vladimir Kolmogorov, *Member*, *IEEE*, Aseem Agarwala, *Member*, *IEEE*, Marshall Tappen, *Member*, *IEEE*, and Carsten Rother, *Member*, *IEEE*

Abstract—Among the most exciting advances in early vision has been the development of efficient energy minimization algorithms for pixel-labeling tasks such as depth or texture computation. It has been known for decades that such problems can be elegantly expressed as Markov random fields, yet the resulting energy minimization problems have been widely viewed as intractable. Recently, algorithms such as graph cuts and loopy belief propagation (LBP) have proven to be very powerful: For example, such methods form the basis for almost all the top-performing stereo methods. However, the trade-offs among different energy minimization algorithms are still not well understood. In this paper, we describe a set of energy minimization benchmarks and use them to compare the solution quality and runtime of several common energy minimization algorithms. We investigate three promising recent methods—graph cuts, LBP, and tree-reweighted message passing—in addition to the well-known older iterated conditional mode (ICM) algorithm. Our benchmark problems are drawn from published energy functions used for stereo, image stitching, interactive segmentation, and denoising. We also provide a general-purpose software interface that allows vision researchers to easily switch between optimization methods. The benchmarks, code, images, and results are available at http://vision.middlebury.edu/MRF/.

Index Terms—Performance evaluation, Markov random fields, global optimization, graph cuts, belief propagation.


Mozilla Firefox
http://vision.middlebury.edu/MRF/eccv06/code/MRF1.6/README.txt
Google
Most Visited
Getting Started
Latest Headlines
Google
Apple
Yahoo!
YouTube
Wikipedia
News
Popular
Google Maps
****
* MRF energy minimization software *
* Version 1.6 *
* May 5, 2006 *
****
This directory contains the MRF energy minimization software accompanying the paper
[1] A Comparative Study of Energy Minimization Methods for Markov Random Fields.
R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov,
A. Agarwala, M. Tappen, and C. Rother.
In Ninth European Conference on Computer Vision (ECCV 2006),
volume 2, pages 16-29, Graz, Austria, May 2006.
Optimization methods included:
1) ICM
2) Graph Cuts
3) Max-Product Belief Propagation
The TRW / TRW-S method is currently not included, but hopefully will be in a future version.
Instructions for compiling and using the software are included below.
CREDITS:
* MRF code, graph cut interface, and example code by Olga Veksler
* Graph cut library by Yuri Boykov and Vladimir Kolmogorov
* Belief propagation code by Marshall Tappen
Done

Comparison of graph cuts and belief propagation

Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters, ICCV 2003.
Marshall F. Tappen William T. Freeman

(a) Tsukuba Image

(b) Graph Cuts

(c) Synchronous BP

(d) Accelerated BP

Figure 3. Results produced by the three algorithms on the Tsukuba image. The parameters used to generate this field were $s = 50, T = 4, P = 2$. Again, Graph Cuts produces a much smoother solution. Belief Propagation does maintain some structures that are lost in the Graph Cuts solution, such as the camera and the face in the foreground.

Ground truth, graph cuts, and belief propagation disparity solution energies

Image	Energy of MRF Labelling Returned ($\times 10^3$)			% Energy from Occluded Matching Costs
	Ground-Truth	Graph Cuts	Synchronous Belief Prop	
Map	757	383	442	61%
Sawtooth	6591	1652	1713	79%
Tsukuba	1852	663	775	61%
Venus	5739	1442	1501	76%

Figure 2. Field Energies for the MRF labelled using ground-truth data compared to the energies for the fields labelled using Graph Cuts and Belief Propagation. Notice that the solutions returned by the algorithms consistently have a much lower energy than the labellings produced from the ground-truth, showing a mismatch between the MRF formulation and the ground-truth. The final column contains the percentage of each ground-truth solution's energy that comes from matching costs of occluded pixels.

Graph cuts versus belief propagation

- Graph cuts consistently gave slightly lower energy solutions for that stereo-problem MRF, although BP ran faster, although there is now a faster graph cuts implementation than what we used…
- Advantages of Belief Propagation:
  - Works for any compatibility functions, not a restricted set like graph cuts.
  - I find it very intuitive.
  - Extensions: sum-product algorithm computes MMSE, and Generalized Belief Propagation gives you very accurate solutions, at a cost of time.

MAP versus MMSE

(a) MAP Estimate

(b) MMSE Estimate

Figure 7. Comparison of MAP and MMSE estimates on a different MRF formulation. The MAP estimate chooses the most likely discrete disparity level for each point, resulting in a depth-map with stair-stepping effects. Using the MMSE estimate assigns sub-pixel disparities, resulting in a smooth depth map.

Outline of MRF section

- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated condtional modes (ICM)
  - Variational methods
  - Belief propagation
  - Graph cuts
- Vision applications of inference in MRF's.
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

Vision applications of MRF's

- Super-resolution
- Motion estimation
- Labelling shading and reflectance
- Many others…

Super-resolution

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

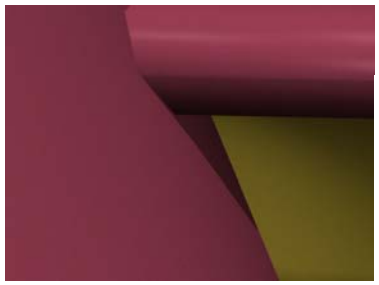
ultimate goal...

Pixel-based images are not resolution independent

Pixel replication

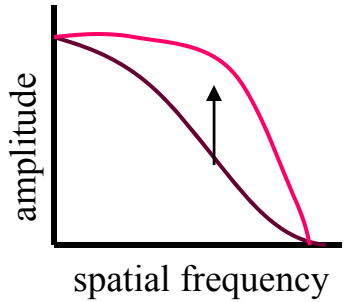
Cubic spline, sharpened

Training-based super-resolution



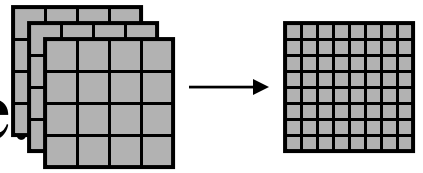
Polygon-based graphics images are resolution independent

3 approaches to perceptual sharpening

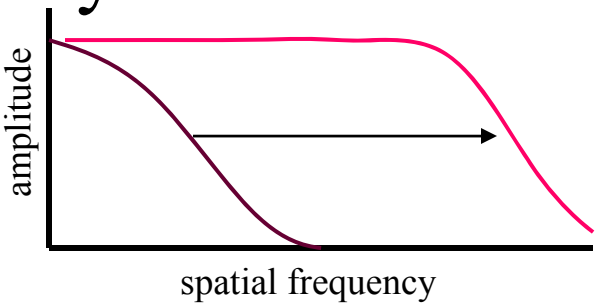


(1) Sharpening; boost existing high frequencies.

(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

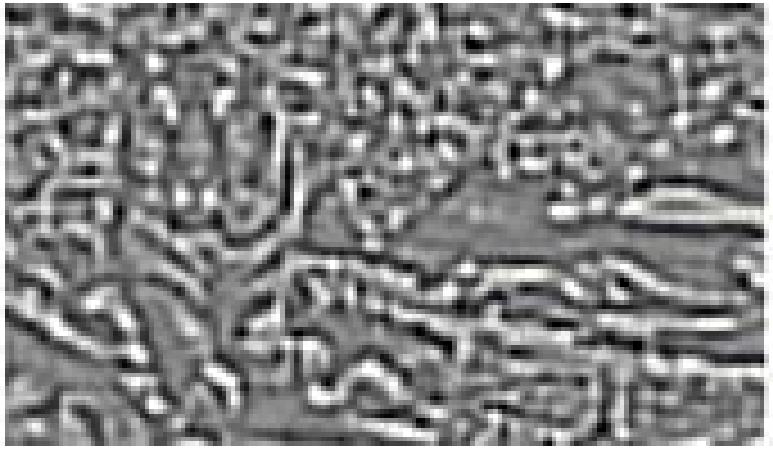
Zoomed low-freq.

Full freq. original

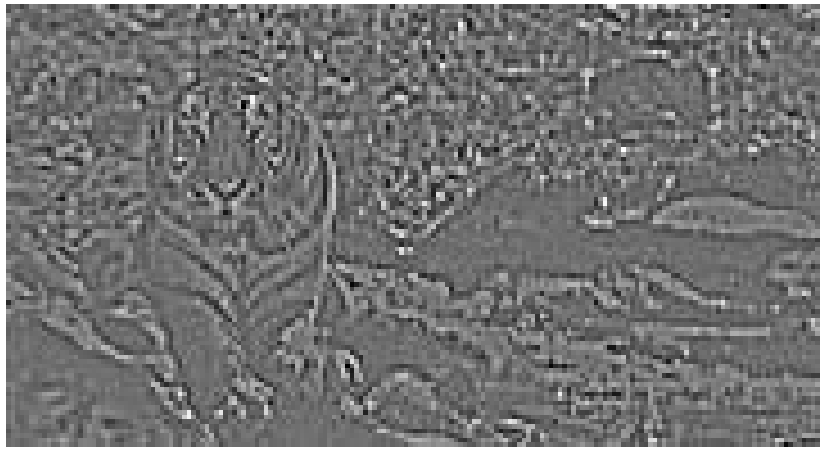
Representation

Zoomed low-freq.

Full freq. original



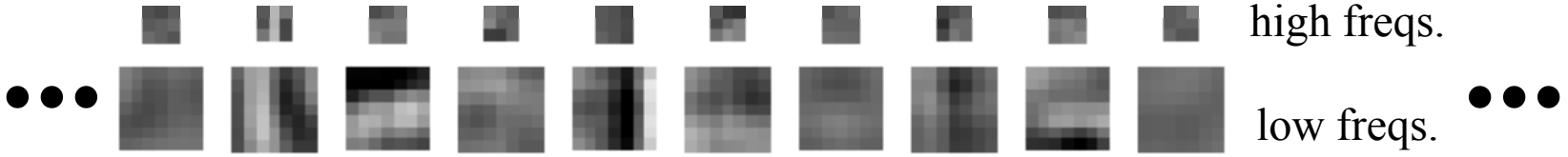
Low-band input
(contrast normalized,
PCA fitted)



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).

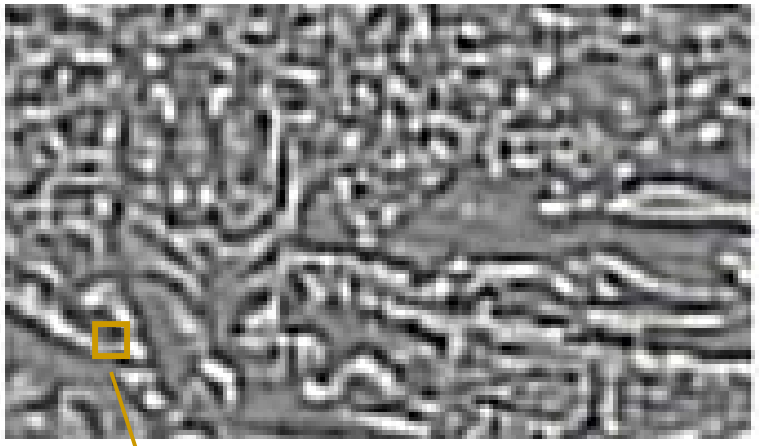
Gather ~100,000 patches



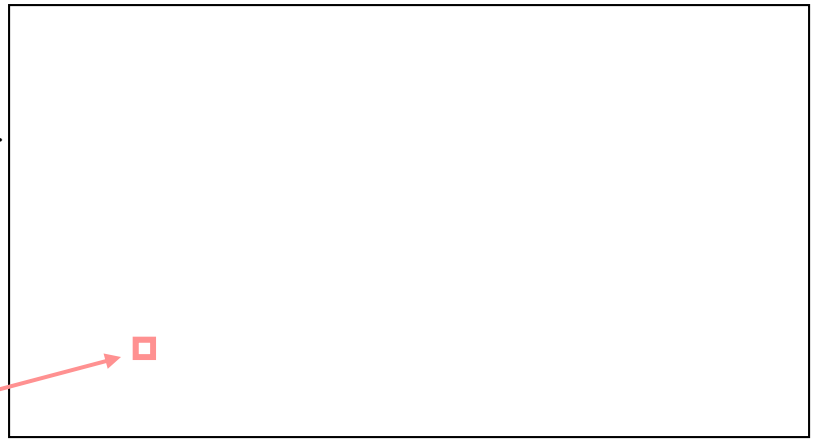
Training data samples (magnified)

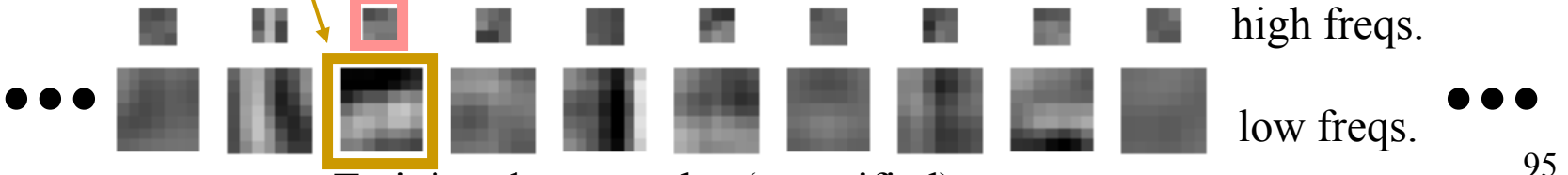
Nearest neighbor estimate

Input low freqs.



Estimated high freqs.

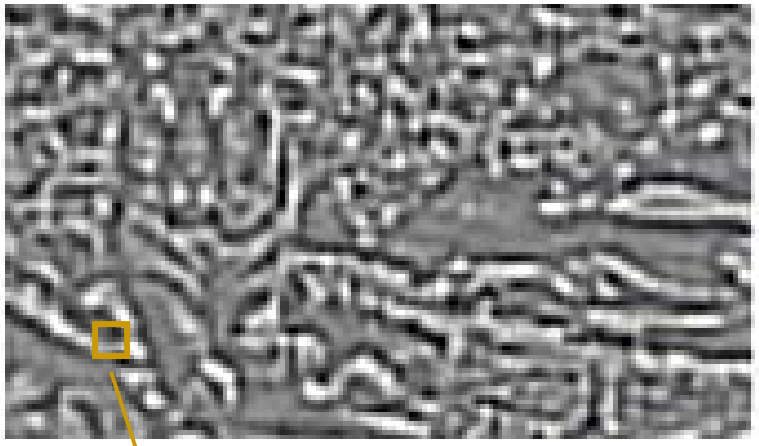




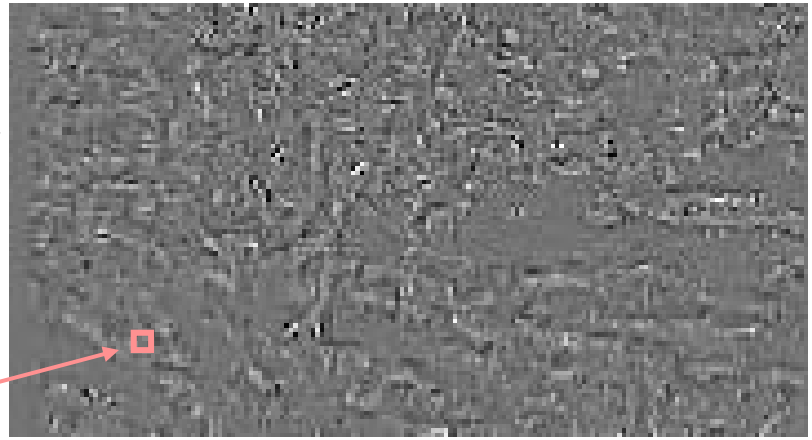
Training data samples (magnified)

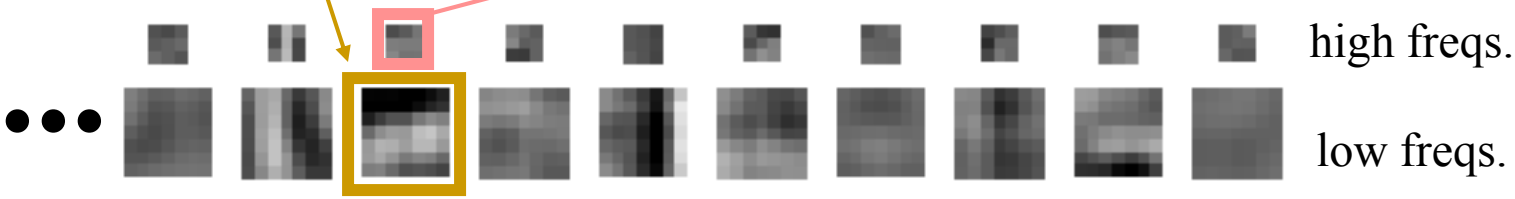
Nearest neighbor estimate

Input low freqs.



Estimated high freqs.

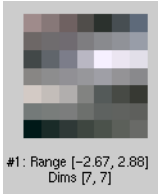




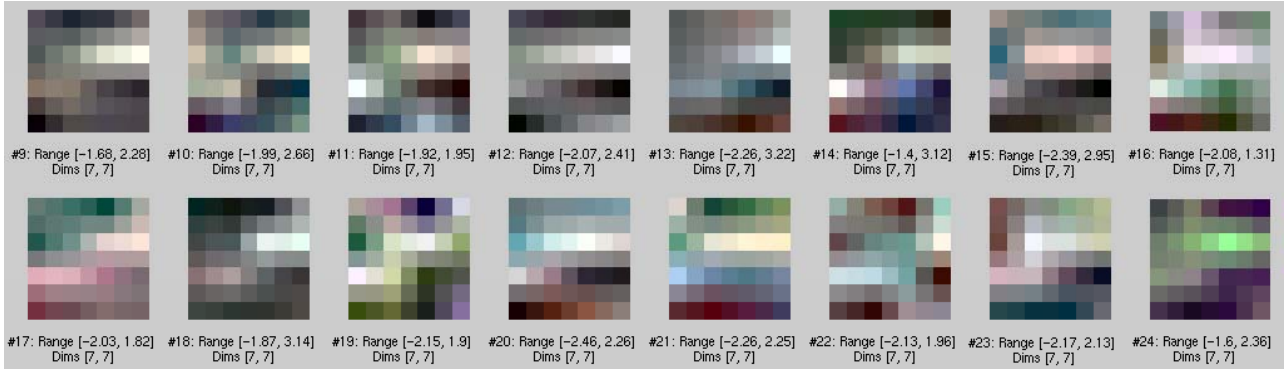
Training data samples (magnified)

Example: input image patch, and closest matches from database

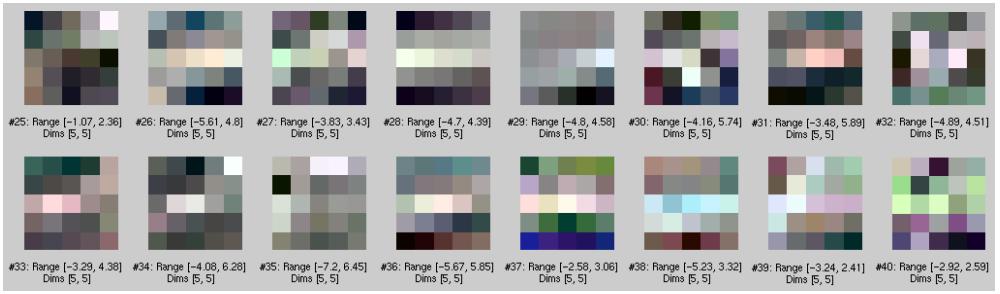
Input patch



Closest image patches from database



Corresponding high-resolution patches from database



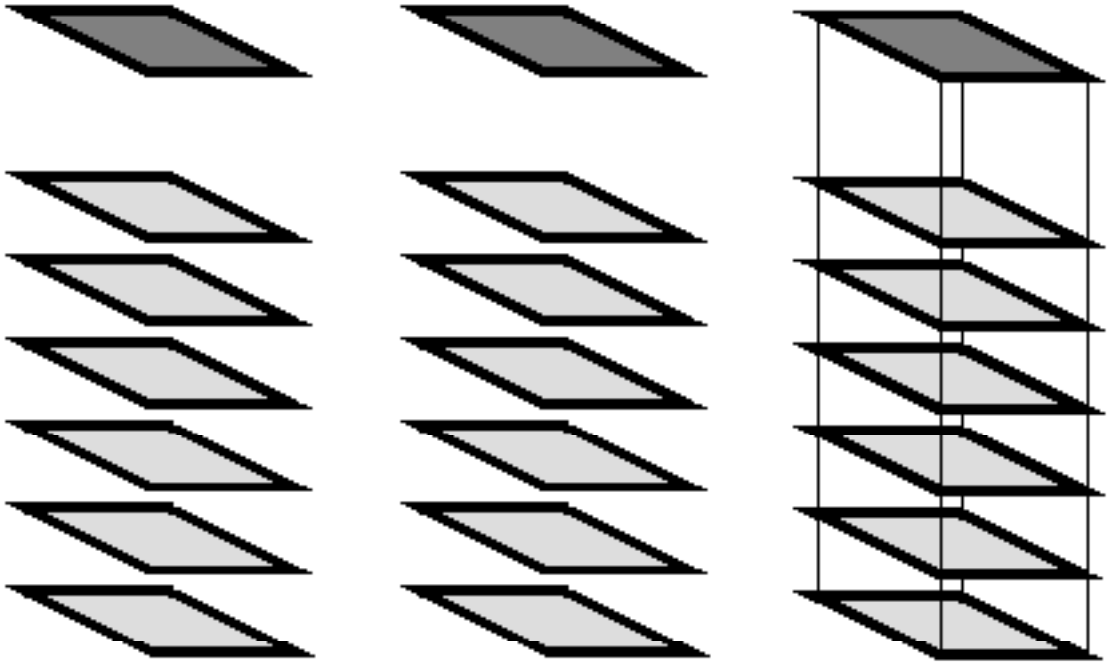
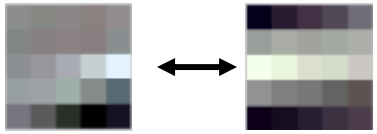


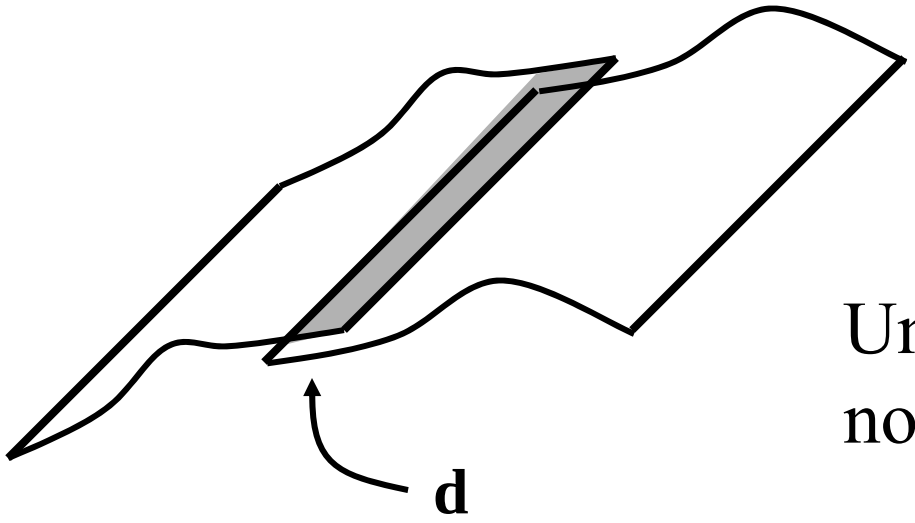
Image patch

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

Scene-scene compatibility function, $\Psi(x_i, x_j)$ 

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

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



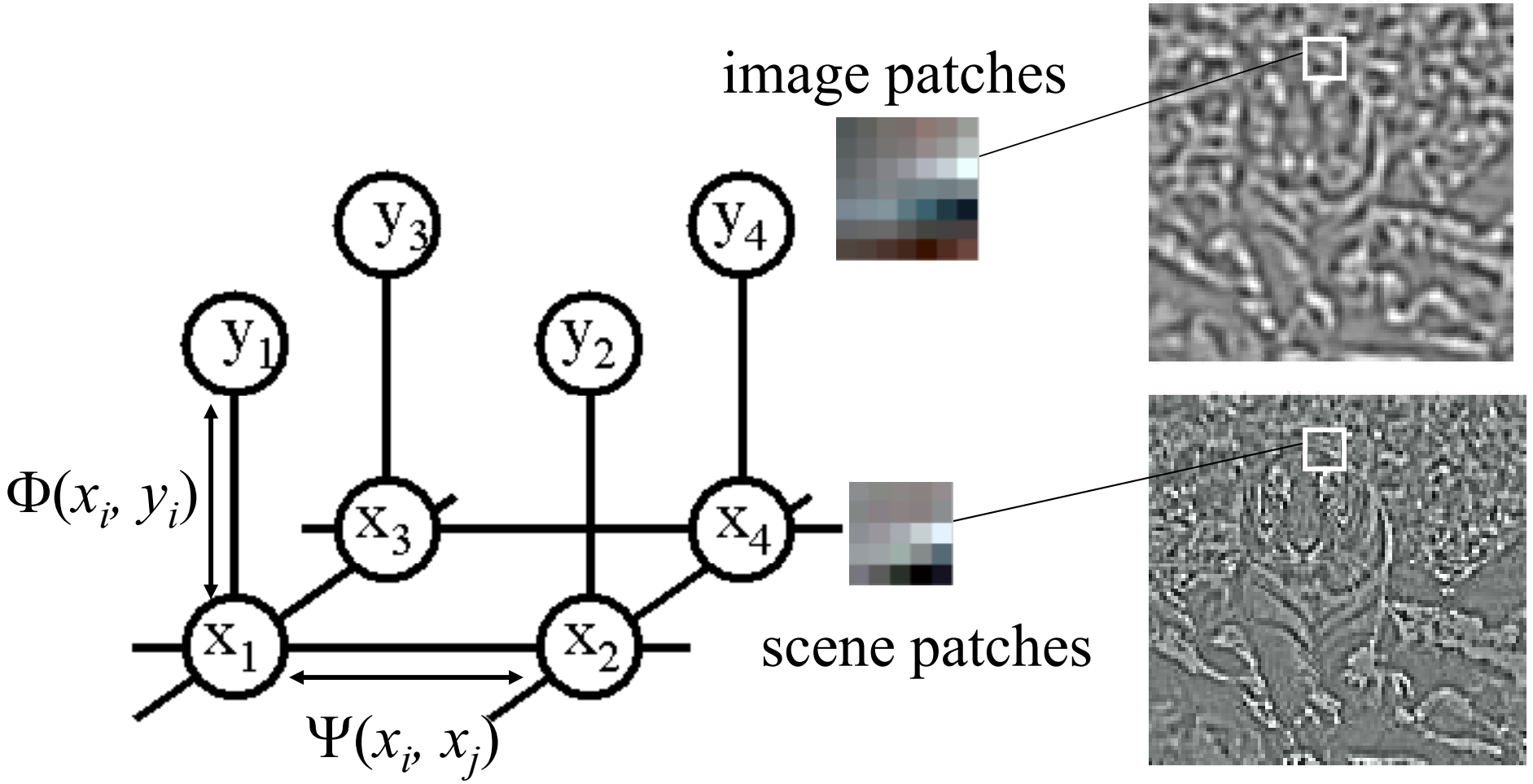
Uniqueness constraint, not smoothness.

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

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

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

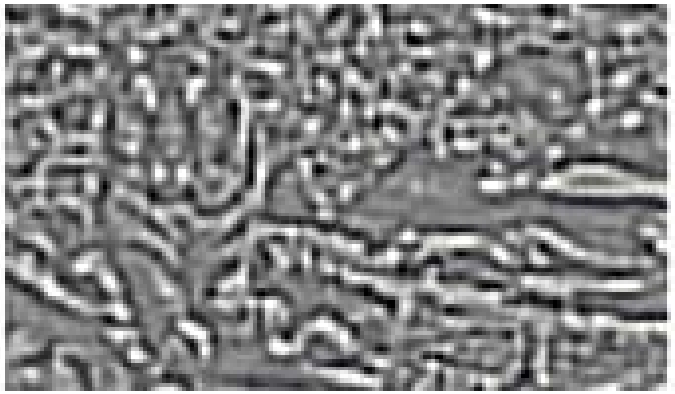
Markov network

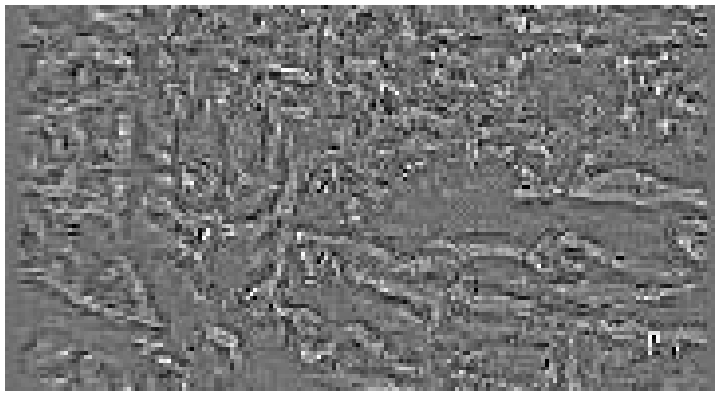


Belief Propagation

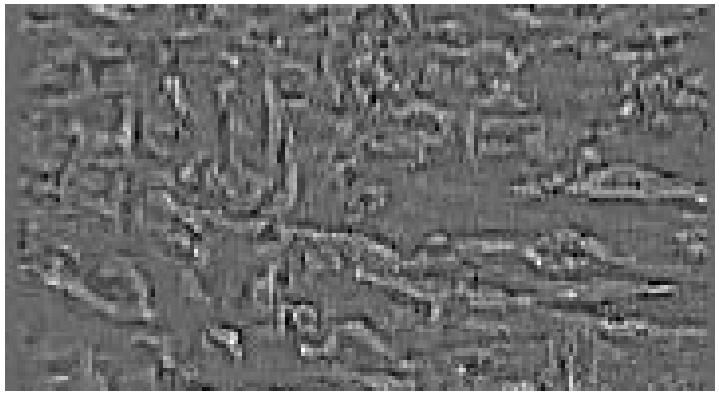
After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

Input

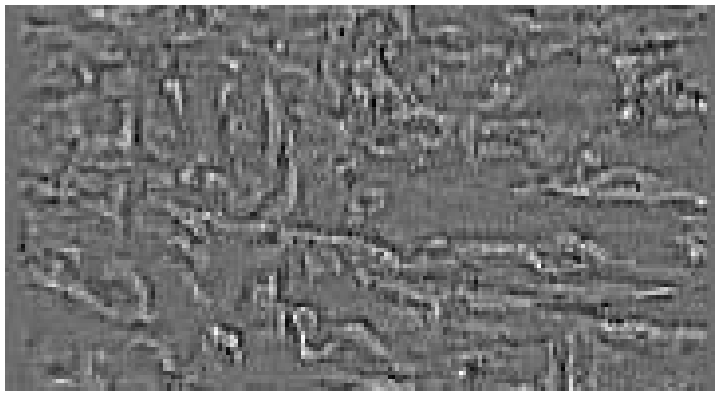




Iter. 0



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

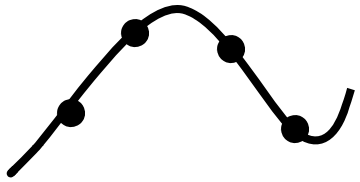
Cubic spline zoom to 340x204

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

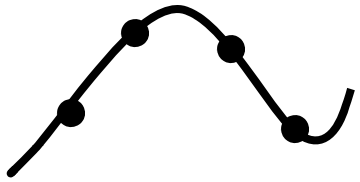
(cubic spline implies thin plate prior)



True
200x232

Original
50x58

(cubic spline implies thin plate prior)

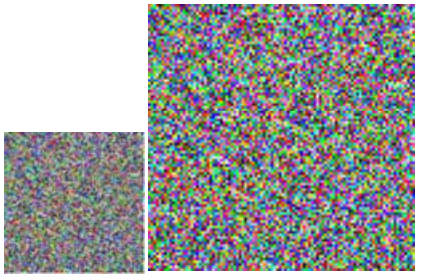


Cubic spline

True
200x232

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

Original
50x58

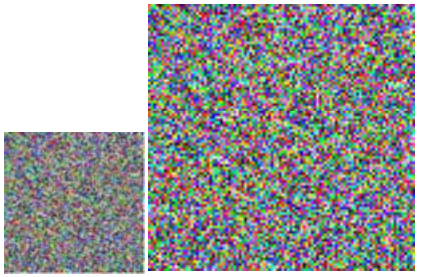


Training images

True

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

Original
50x58



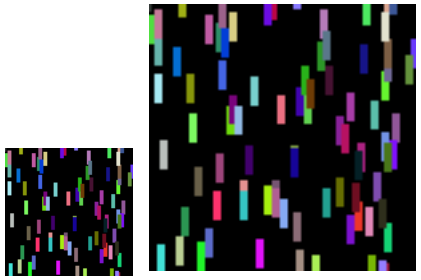
Training images

Markov
network

True

Next, train on a world of vertically oriented rectangles.

Original
50x58

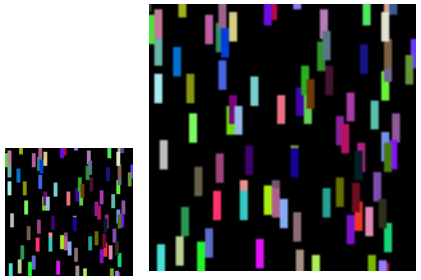


Training images

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

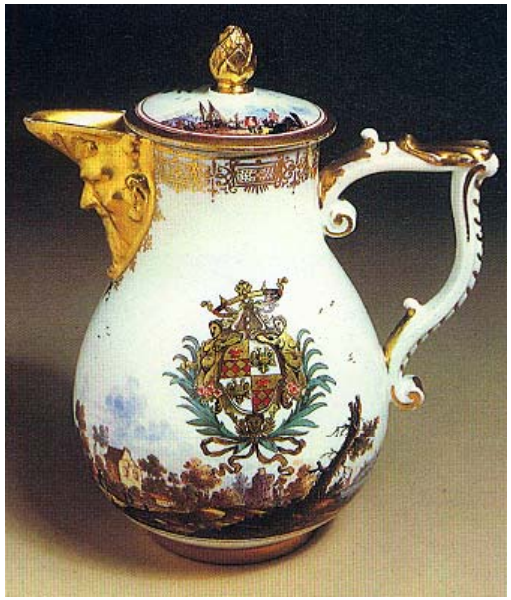
True

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

Original
50x58

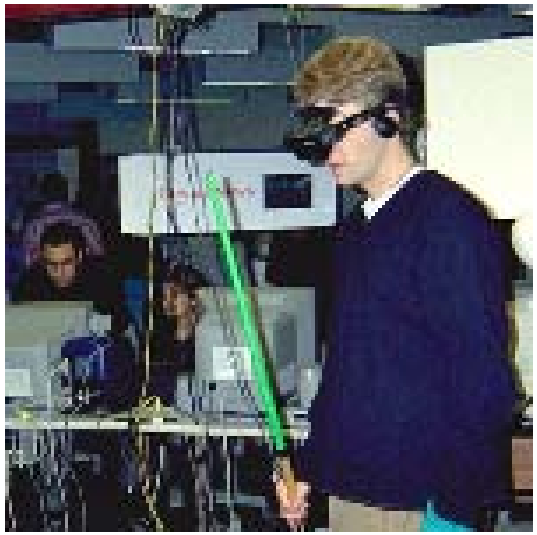
Training images

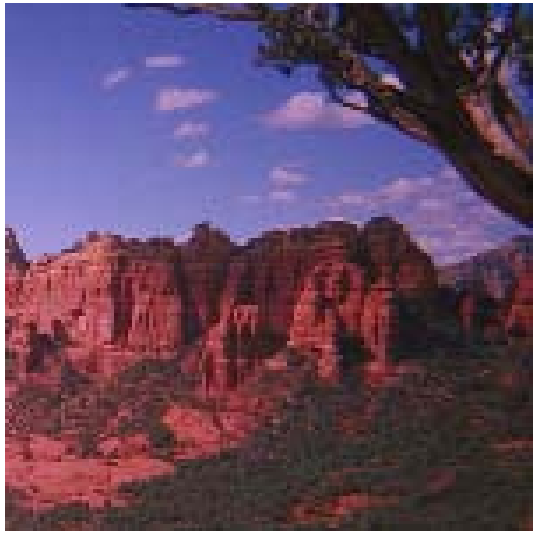
Markov
network



True

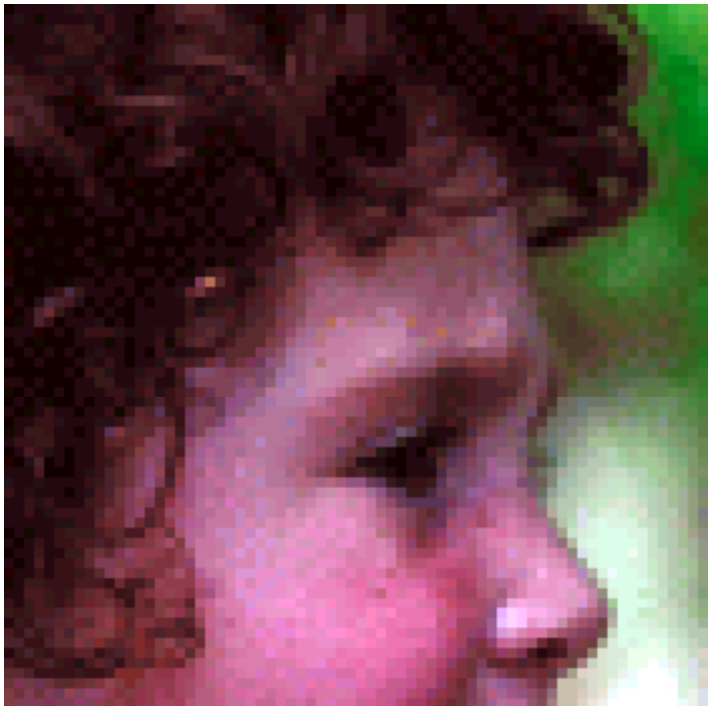
Generic training images





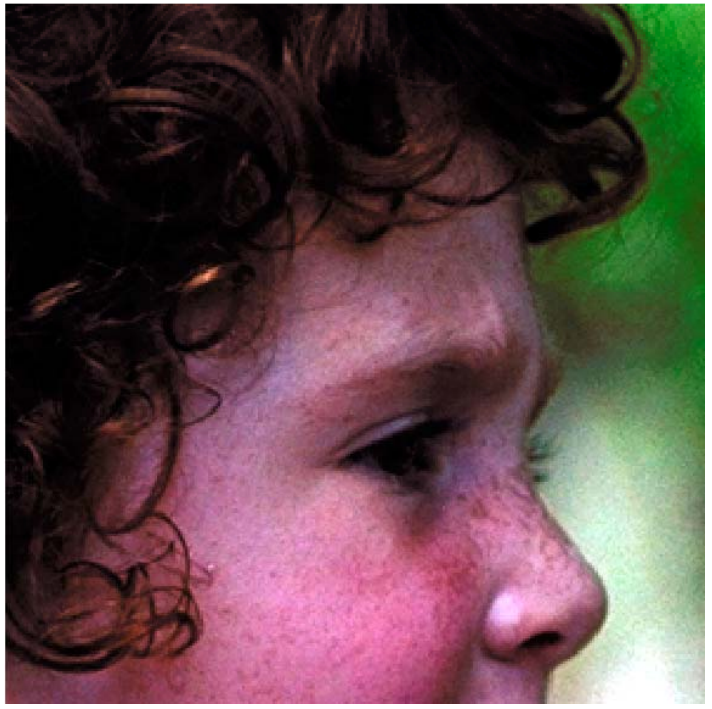
Next, train on a generic set of training images. Using the same camera as for the test image, but a random collection of photographs.

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.

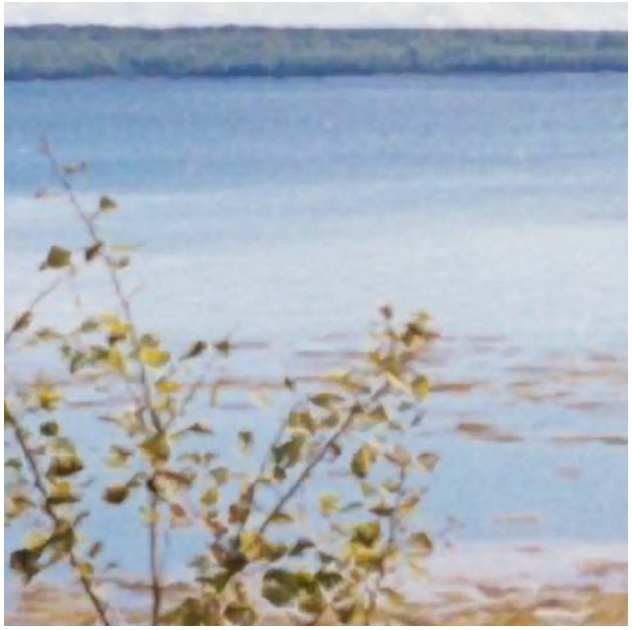
Algorithms compared

- 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."

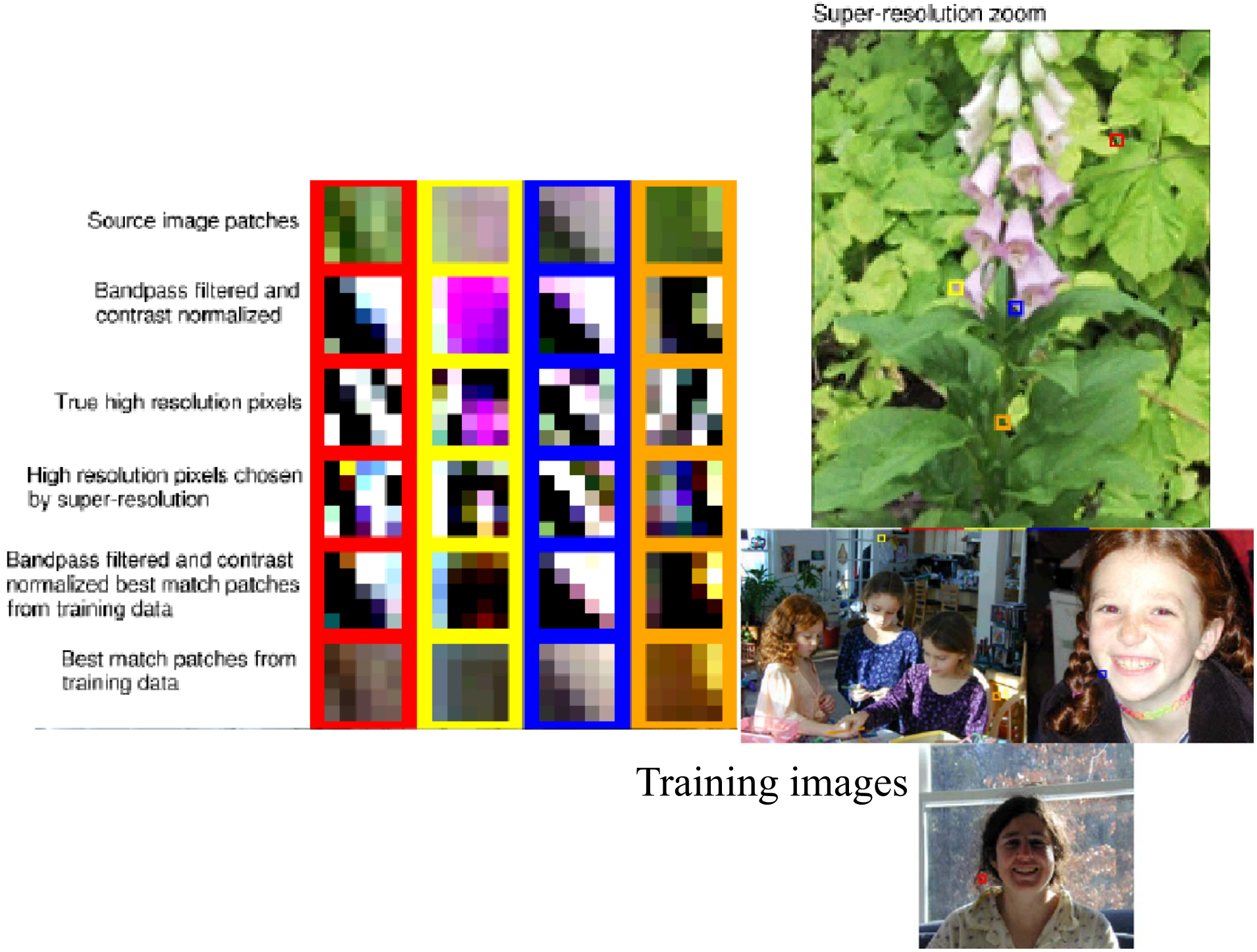
Input

Cubic spline zoom

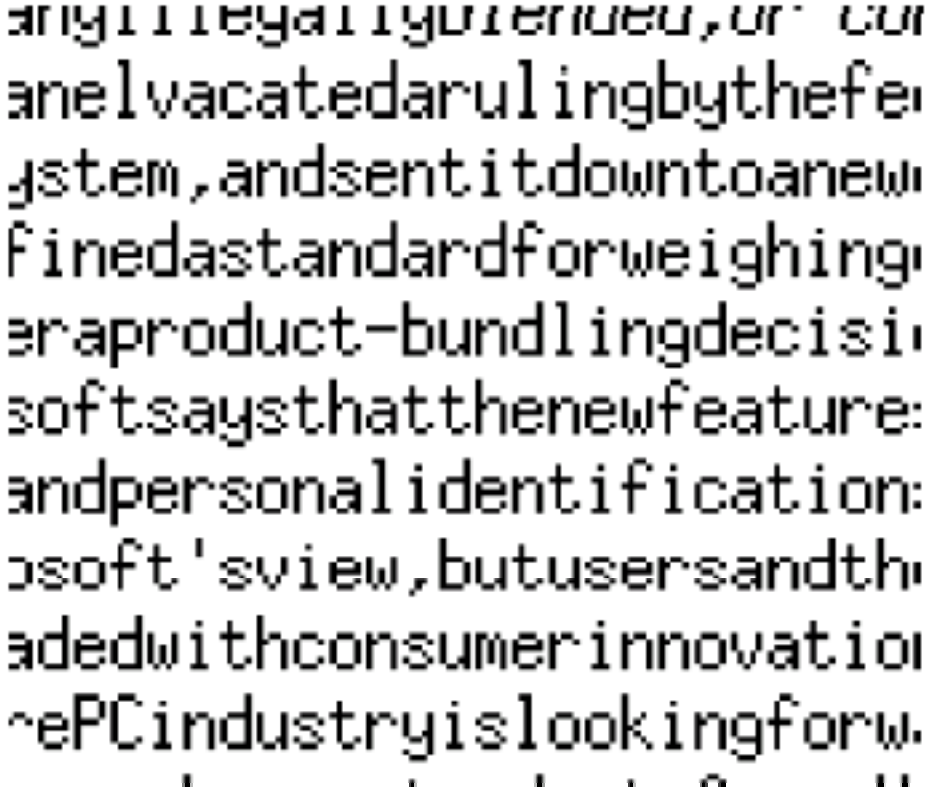
Super-resolution zoom

True high-resolution image




Super-resolution zoom
Source image patches
Bandpass filtered and contrast normalized
True high resolution pixels
High resolution pixels chosen by super-resolution
Bandpass filtered and contrast normalized best match patches from training data
Best match patches from training data
Training images

Training image

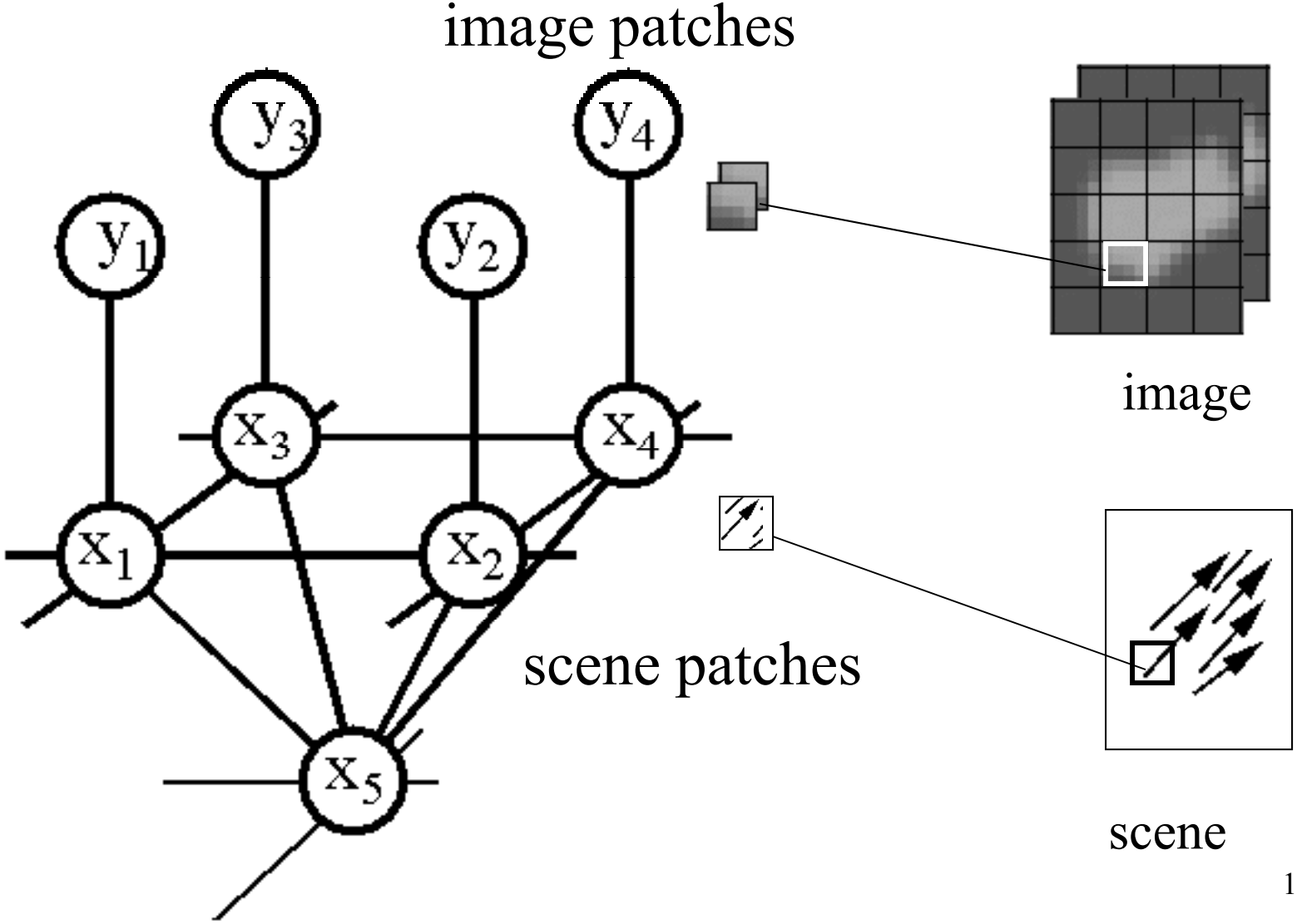


Processed image

Vision applications of MRF's

- Super-resolution
- Motion estimation
- Labelling shading and reflectance
- Many others…

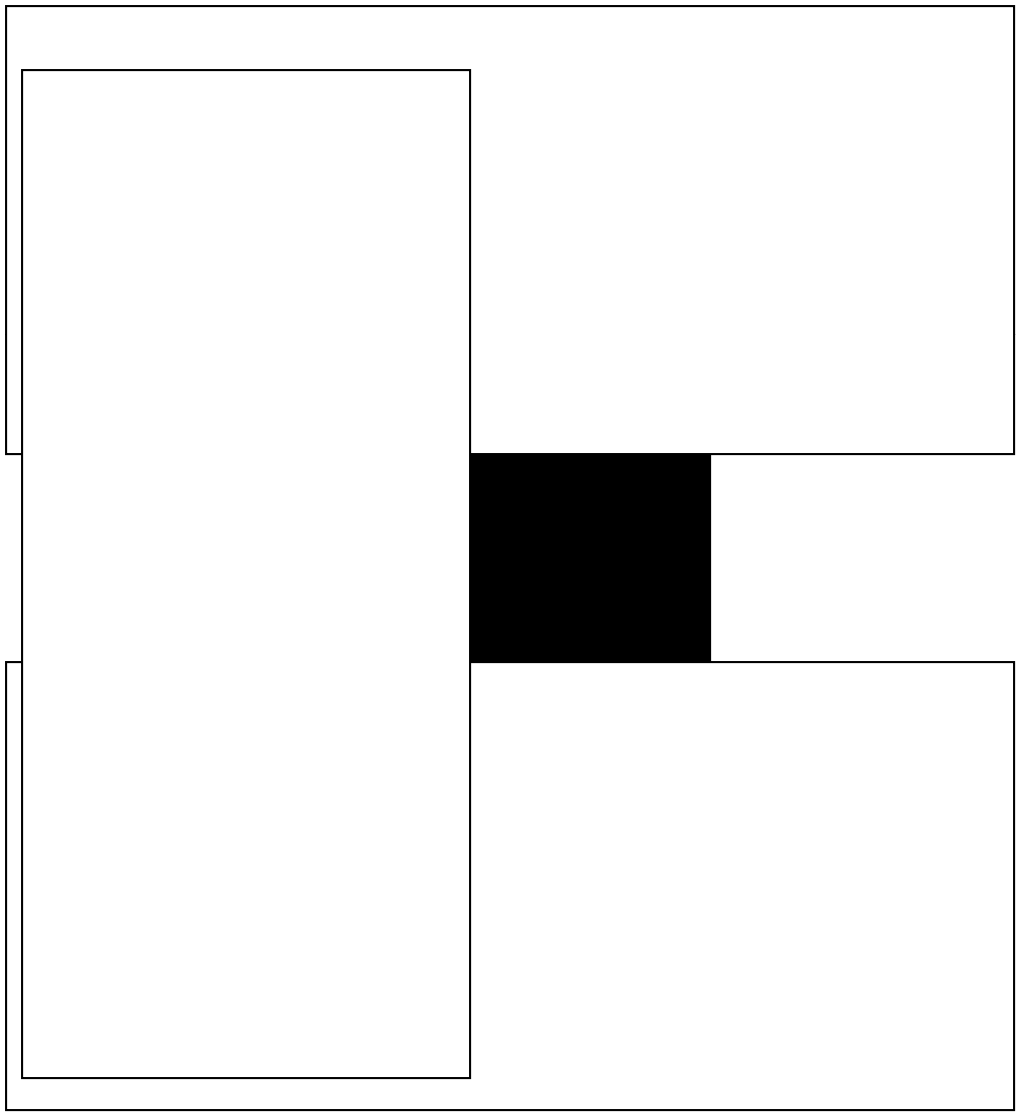
Motion application



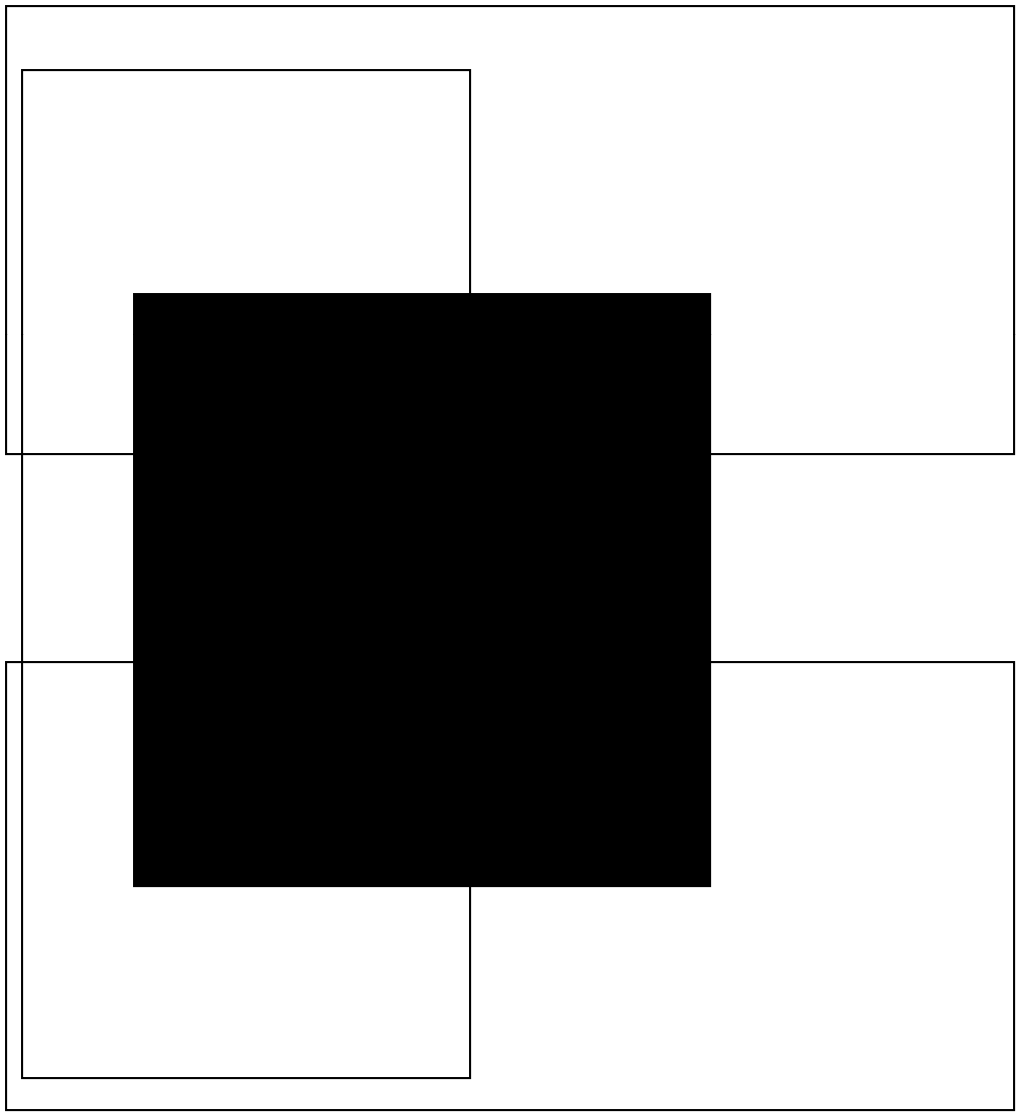
What behavior should we see in a motion algorithm?

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

The aperture problem



The aperture problem



Motion analysis: related work

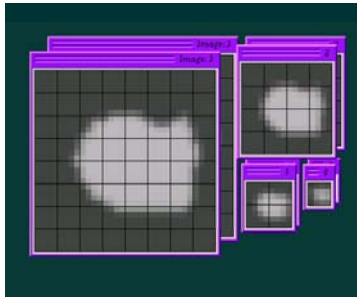
- **Markov network**
  - Luettgen, Karl, Willsky and collaborators.
- **Neural network or learning-based**
  - Nowlan & T. J. Senjowski; Sereno.
- **Optical flow analysis**
  - Weiss & Adelson; Darrell & Pentland; Ju, Black & Jepson; Simoncelli; Grzywacz & Yuille; Hildreth; Horn & Schunk; etc.

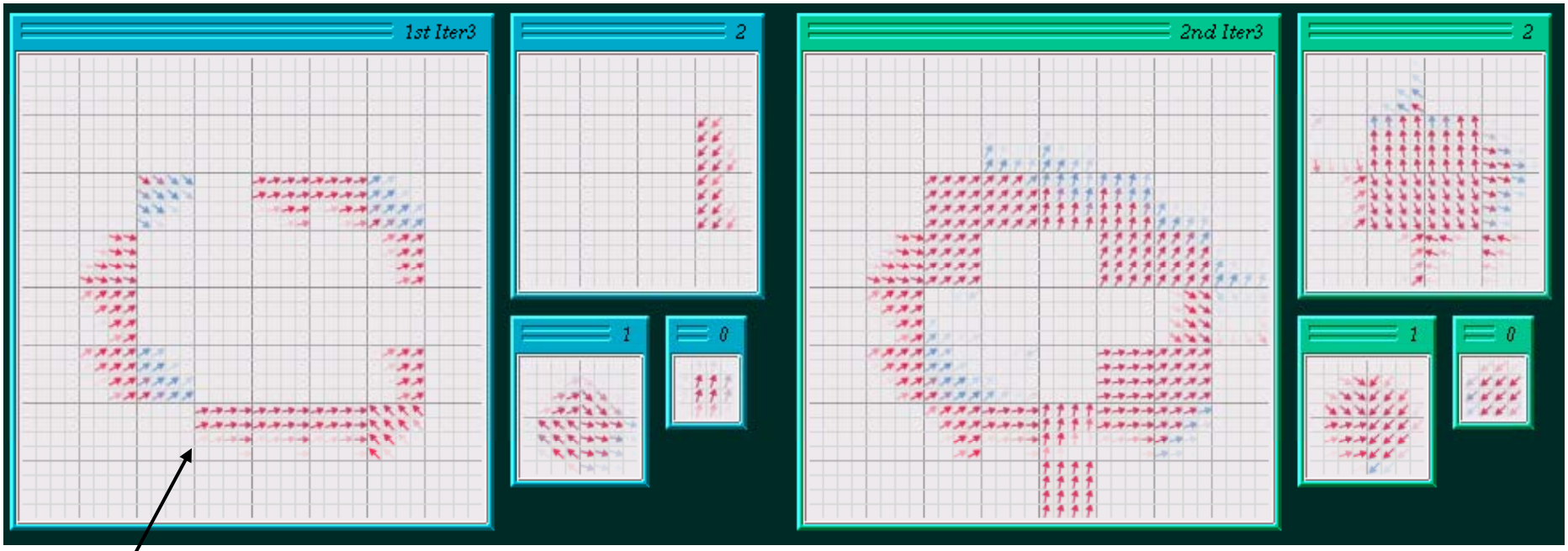
Inference:

Motion estimation results

(maxima of scene probability distributions displayed)

Image data



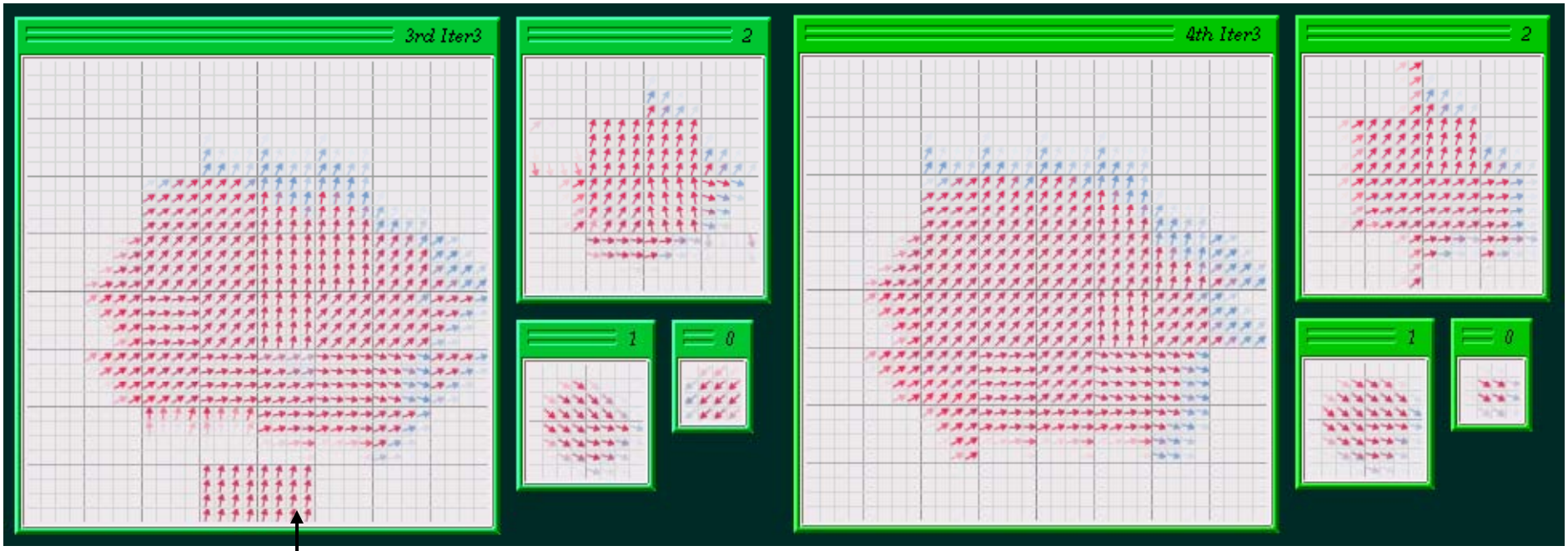


Iterations 0 and 1

Initial guesses only
show motion at edges.

Motion estimation results

(maxima of scene probability distributions displayed)

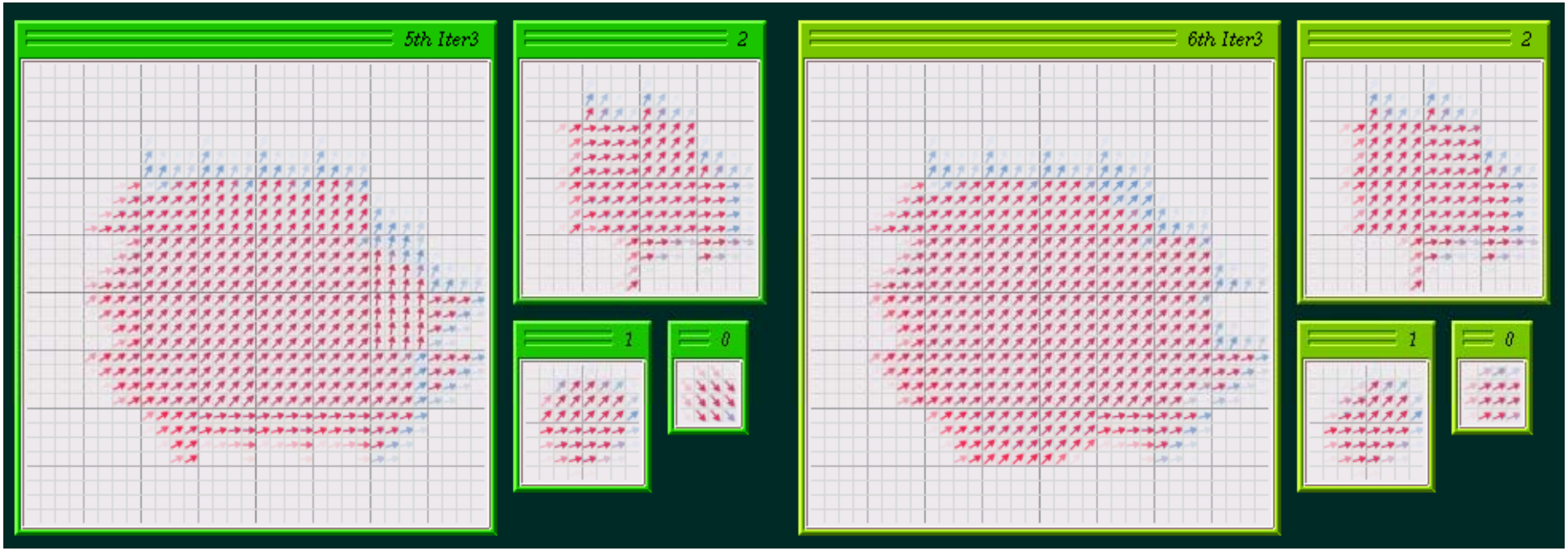


Iterations 2 and 3

Figure/ground still unresolved here.

Motion estimation results

(maxima of scene probability distributions displayed)



Iterations 4 and 5

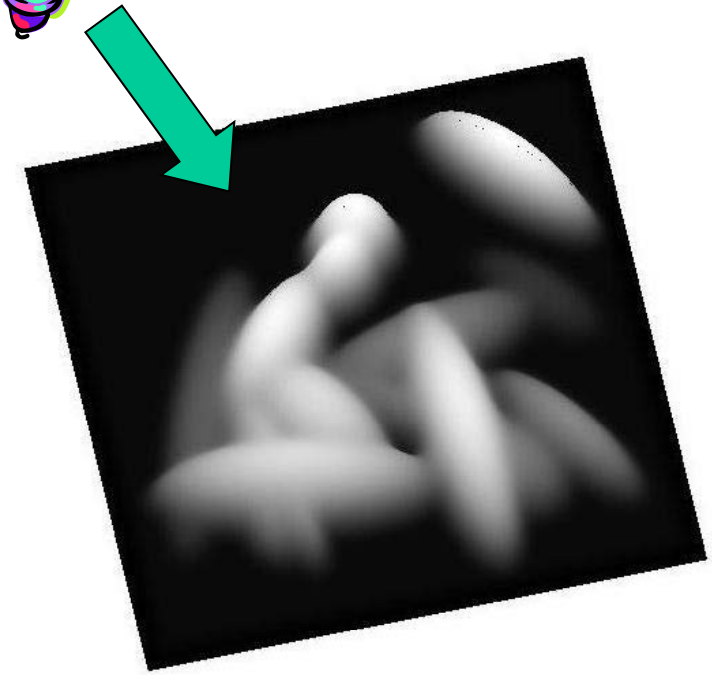
Final result compares well with vector quantized true (uniform) velocities.

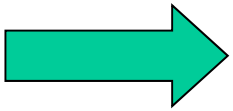
Vision applications of MRF's

- Super-resolution
- Motion estimation
- Labelling shading and reflectance
- Many others…

Forming an Image

Illuminate the surface to get:



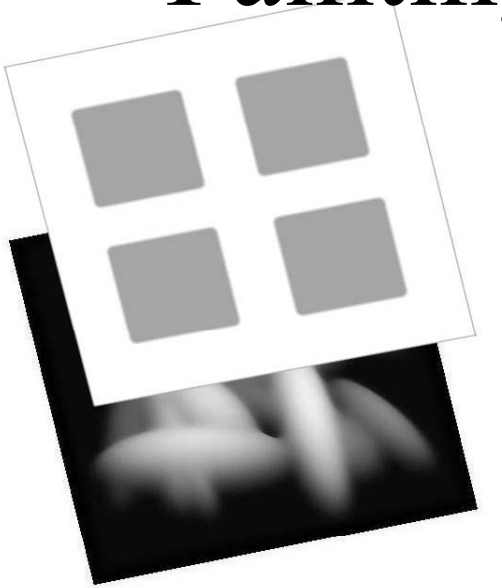


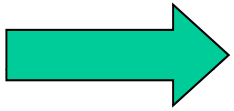
Surface (Height Map)

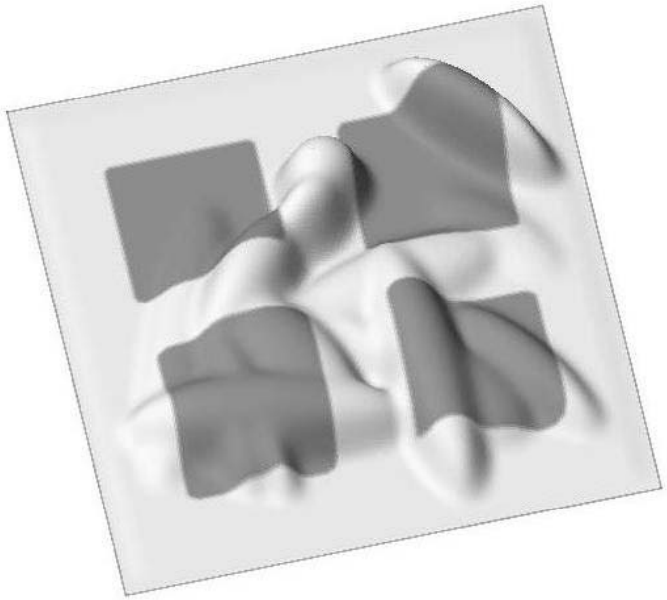
Shading Image

The shading image is the interaction of the shape of the surface and the illumination

Painting the Surface





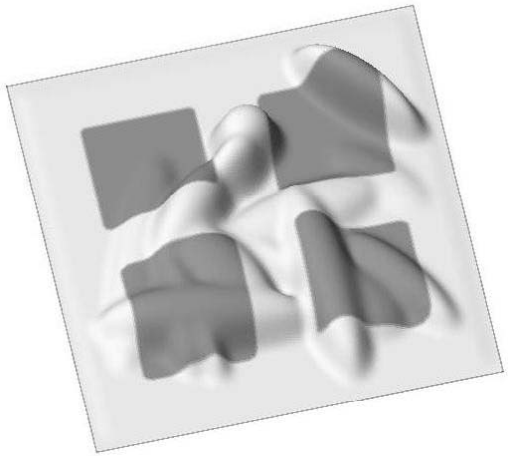


Scene

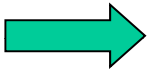
Image

Add a reflectance pattern to the surface. Points inside the squares should reflect less light

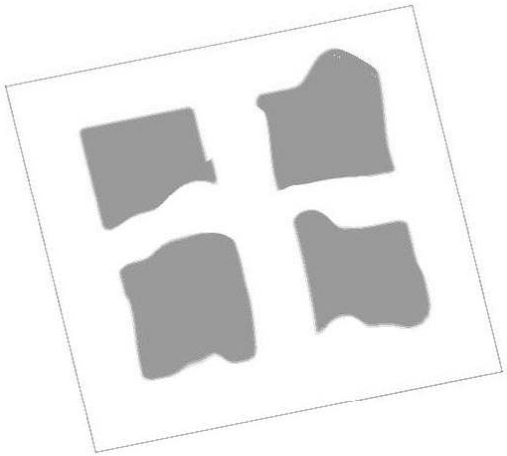
Goal



Image



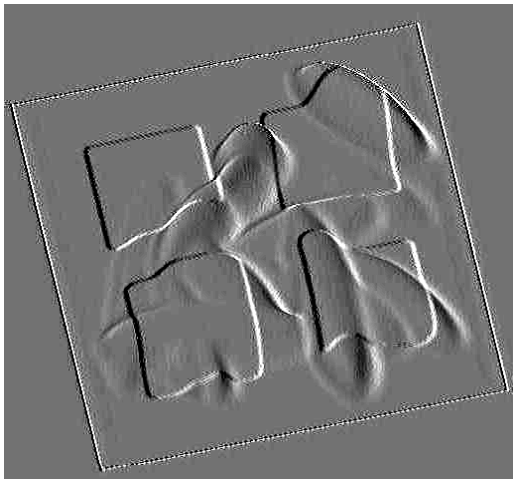
Shading Image



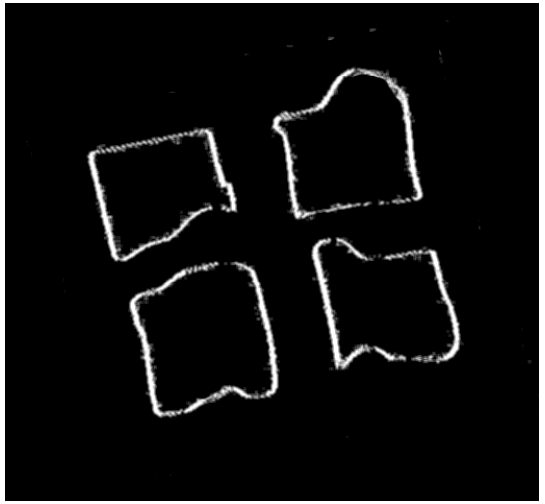
Reflectance Image

Basic Steps

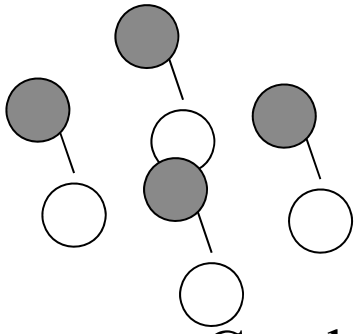
1. Compute the *x* and *y* image derivatives
2. Classify each derivative as being caused by *either* shading or a reflectance change
3. Set derivatives with the wrong label to zero.
4. Recover the intrinsic images by finding the least-squares solution of the derivatives.



Original *x* derivative image



Classify each derivative
(White is reflectance)

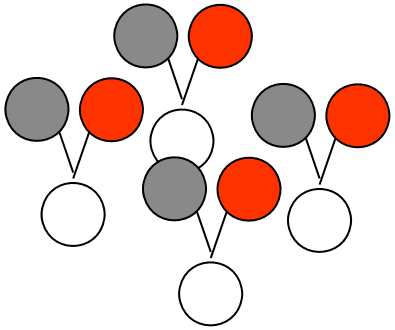


Learning the Classifiers

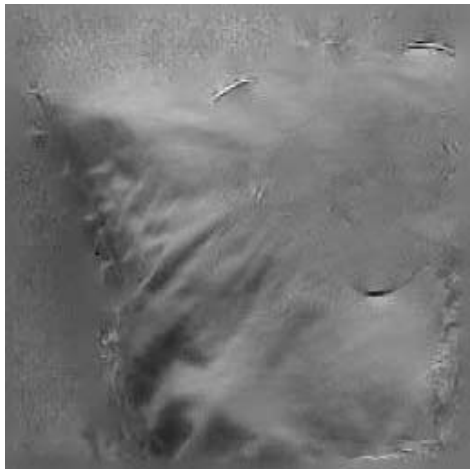
- Combine multiple classifiers into a strong classifier using AdaBoost (Freund and Schapire)
- Choose weak classifiers greedily similar to (Tieu and Viola 2000)
- Train on synthetic images
- Assume the light direction is from the right

Shading Training Set

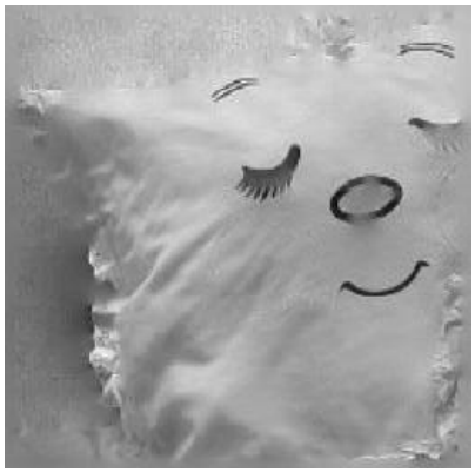
Reflectance Change Training Set



Using Both Color and Gray-Scale Information

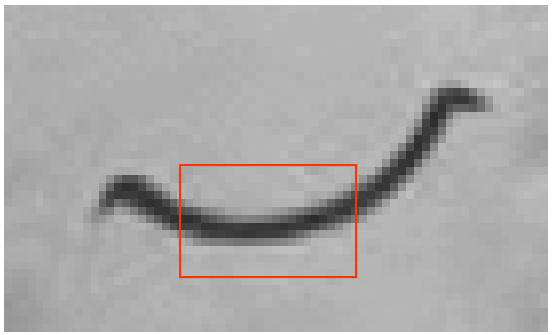


Results without considering gray-scale

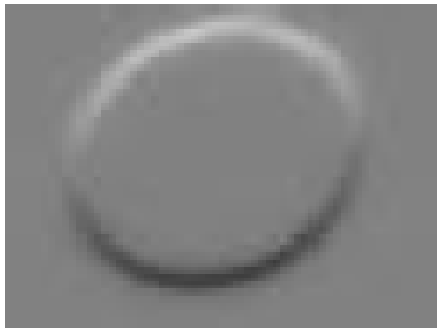


Some Areas of the Image Are Locally Ambiguous

Input

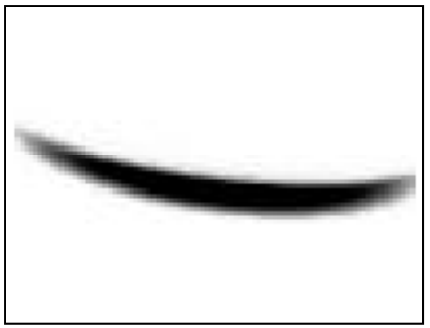


Is the change here better explained as



Shading

or

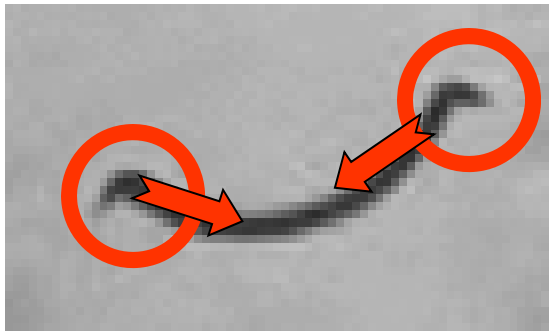


?

Reflectance

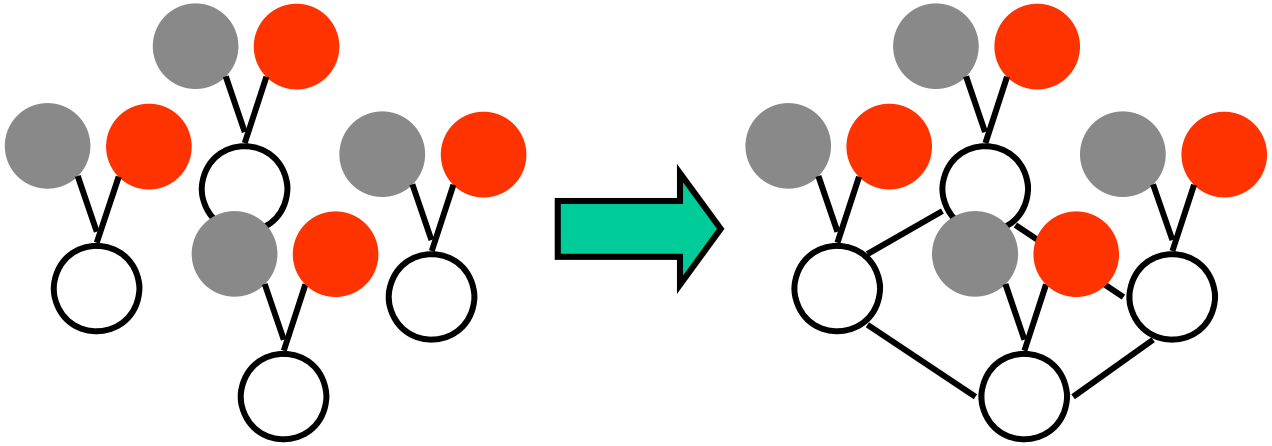
Propagating Information

- Can disambiguate areas by propagating information from reliable areas of the image into ambiguous areas of the image



Propagating Information

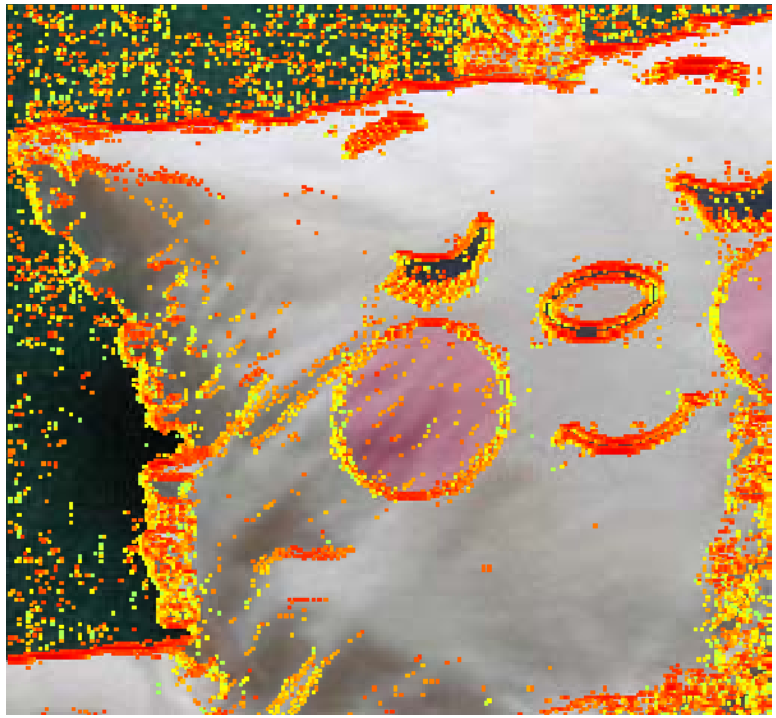
- Consider relationship between neighboring derivatives



- Use Generalized Belief Propagation to infer labels

Setting Compatibilities

- Set compatibilities according to image contours
  - All derivatives along a contour should have the same label
- Derivatives along an image contour strongly influence each other



$$\psi(x_i, x_j) = \begin{bmatrix} 1-\beta & \beta \\ \beta & 1-\beta \end{bmatrix}$$

Improvements Using Propagation

Input Image

Reflectance Image
Without Propagation

Reflectance Image
With Propagation

Combining: local evidence from shape and color, and GBP for propagation

(a) Original Image (b) Shading Image (c) Reflectance Image

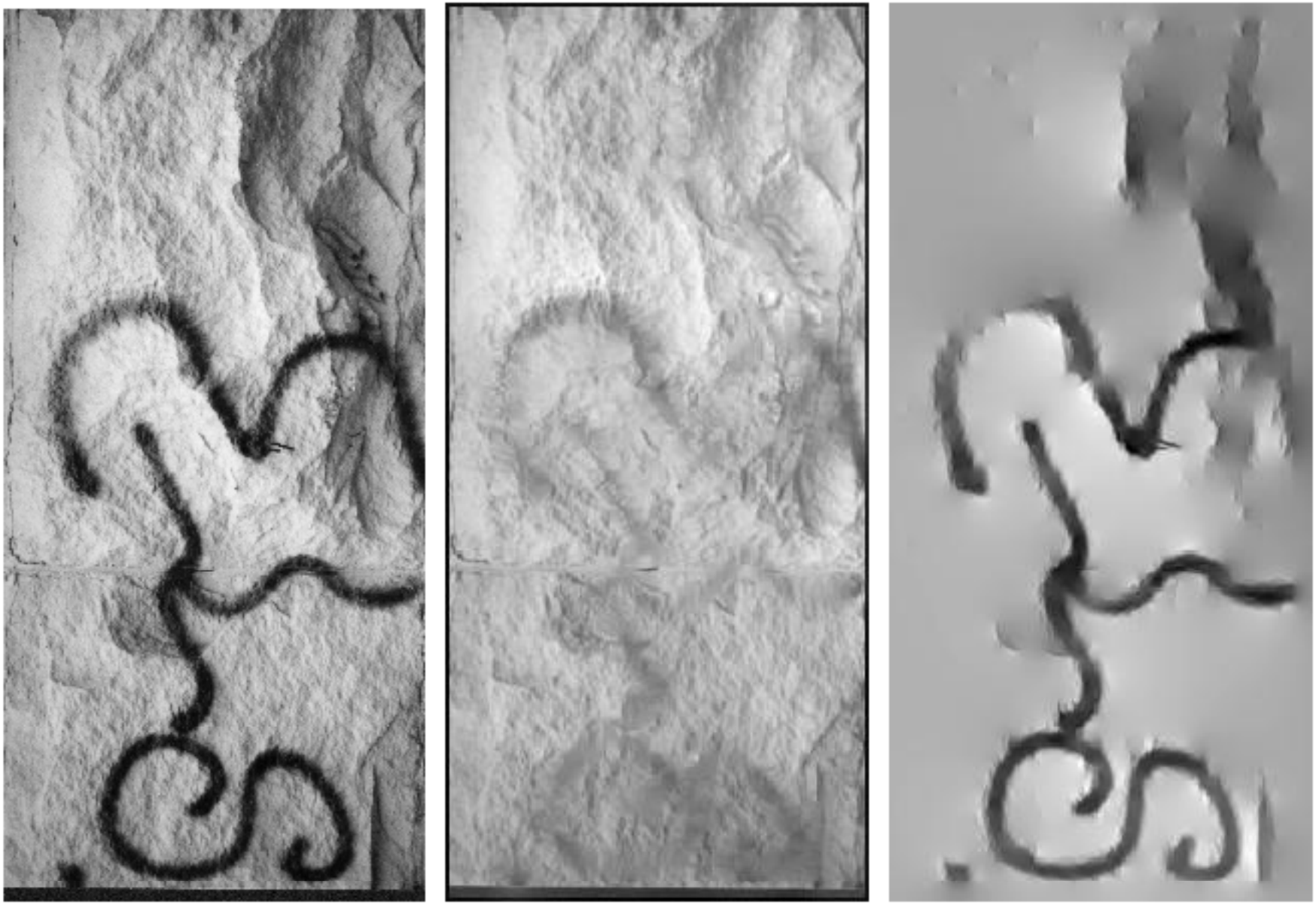
(More Results)

Input Image

Shading Image

Reflectance Image

(a) Original Image



(a) Original Image (b) Shape Image (c) Reflectance Image

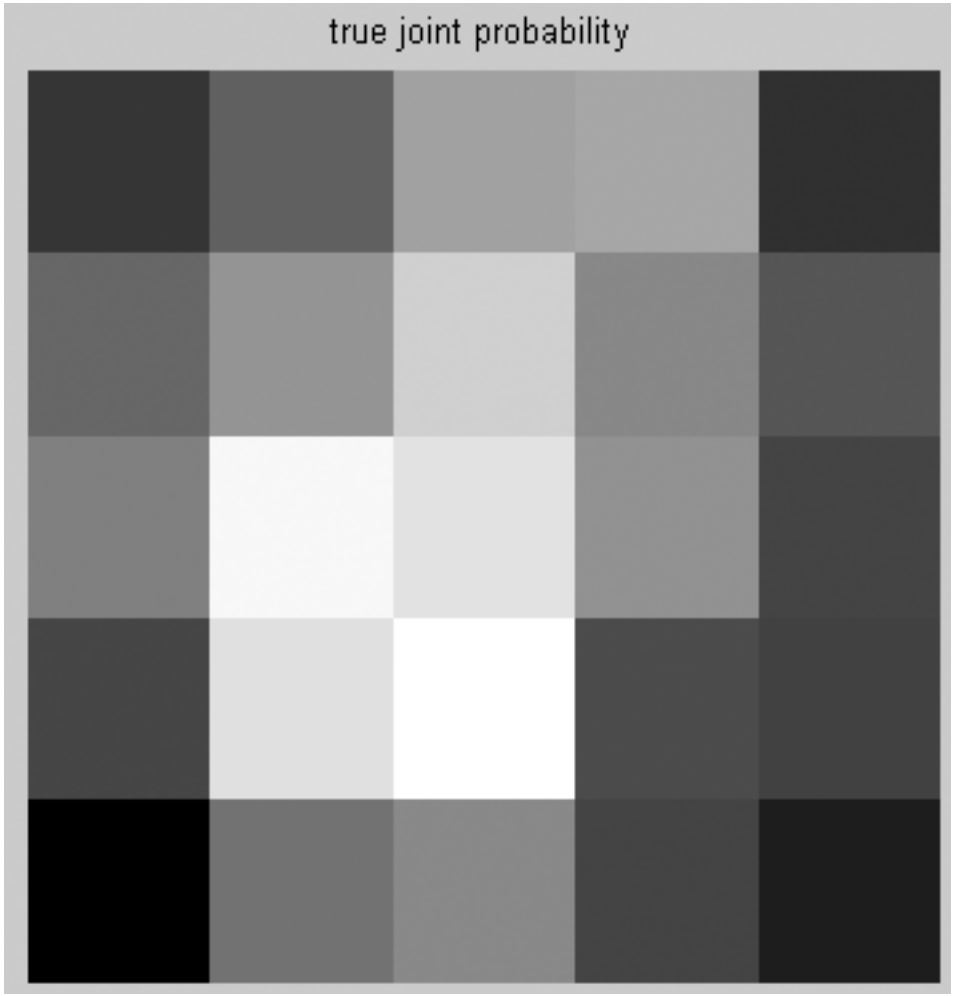
Outline of MRF section

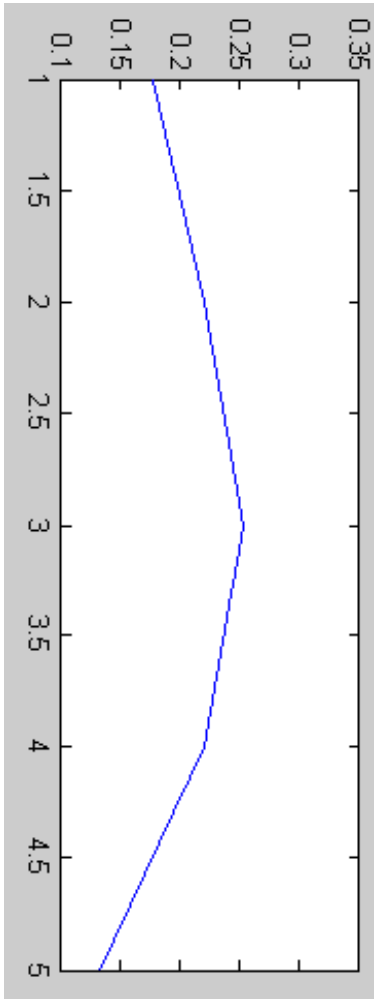
- Inference in MRF's.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Variational methods
  - Belief propagation
  - Graph cuts
- Vision applications of inference in MRF's.
- Learning MRF parameters.
  - Iterative proportional fitting (IPF)

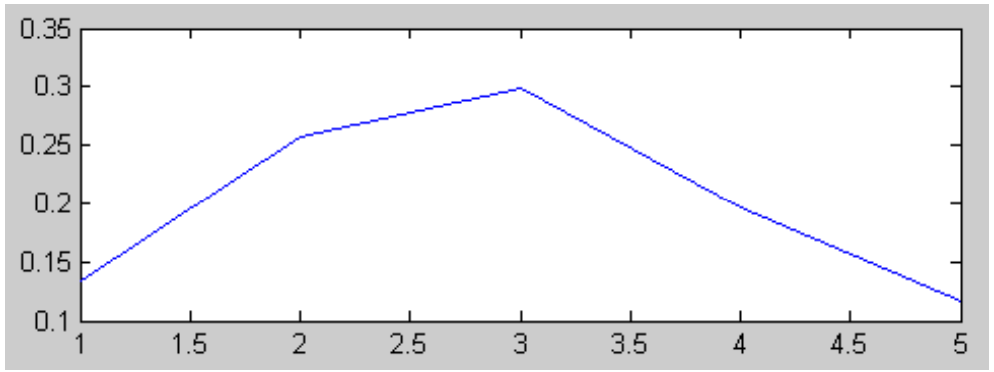
Learning MRF parameters, labeled data

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

True joint probability

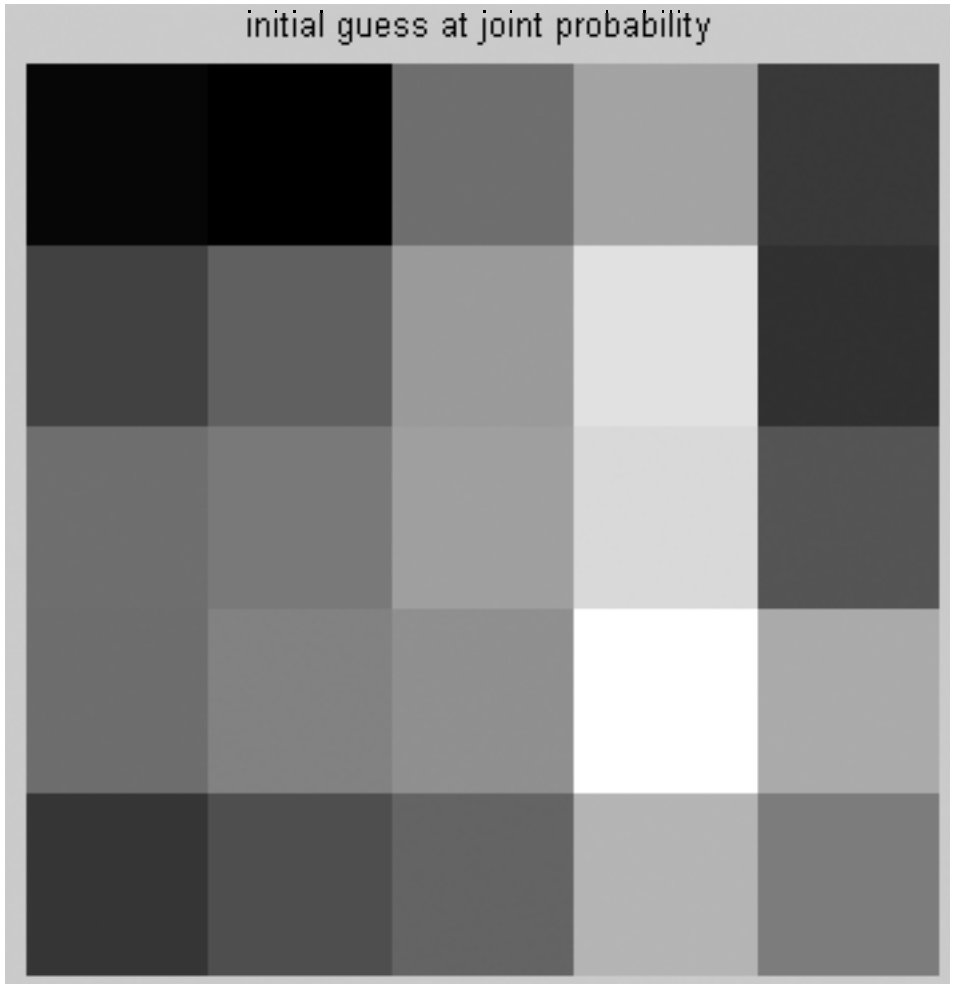






Observed marginal distributions

Initial guess at joint probability



IPF update equation

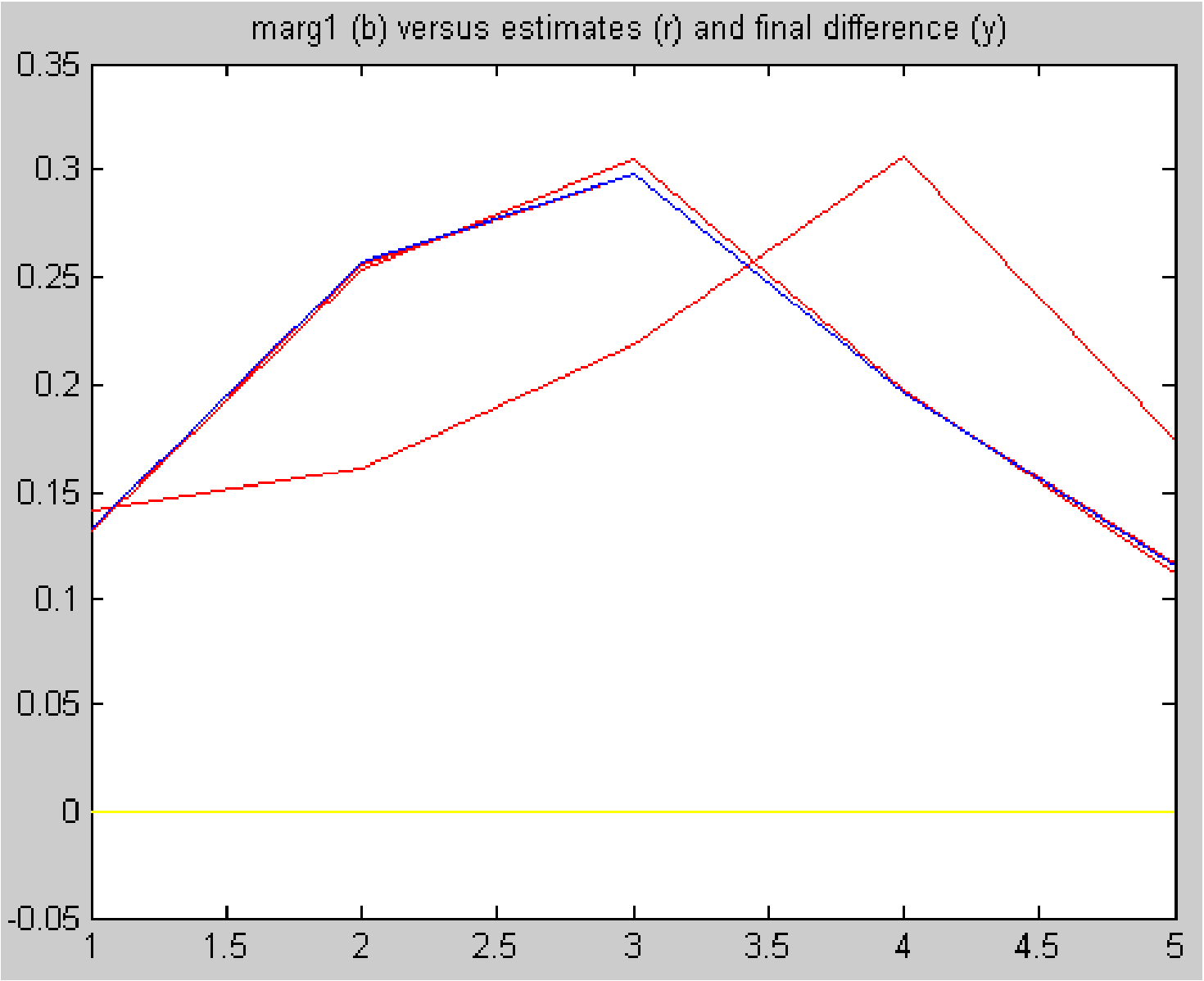
$$P(x_1, x_2, \ldots, x_d)^{(t+1)} = P(x_1, x_2, \ldots, x_d)^{(t)} \frac{P(x_i)^{\text{observed}}}{P(x_i)^{(t)}}$$

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

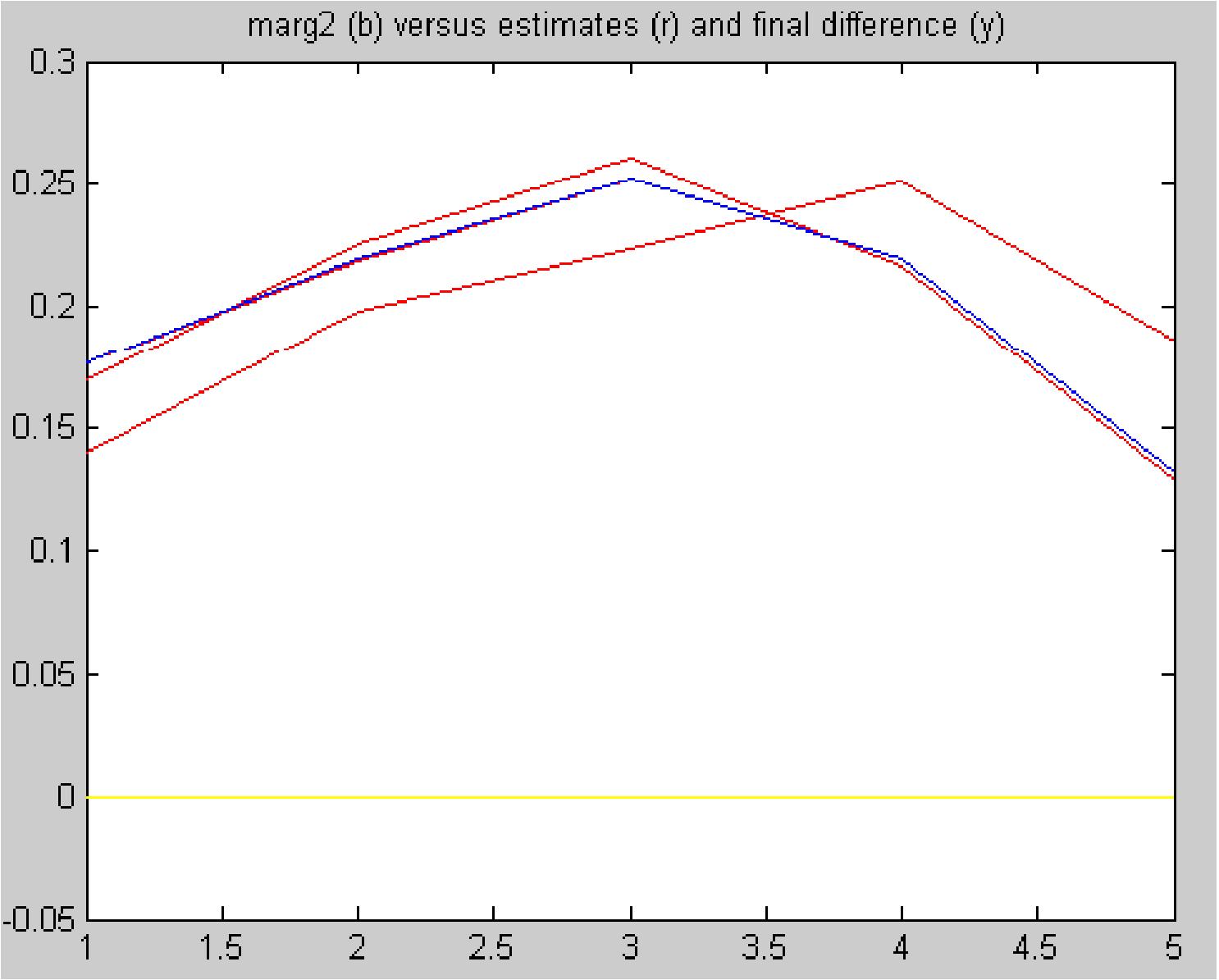
Gives gradient ascent in the likelihood of the joint probability, given the observations of the marginals.

See: Michael Jordan's book on graphical models

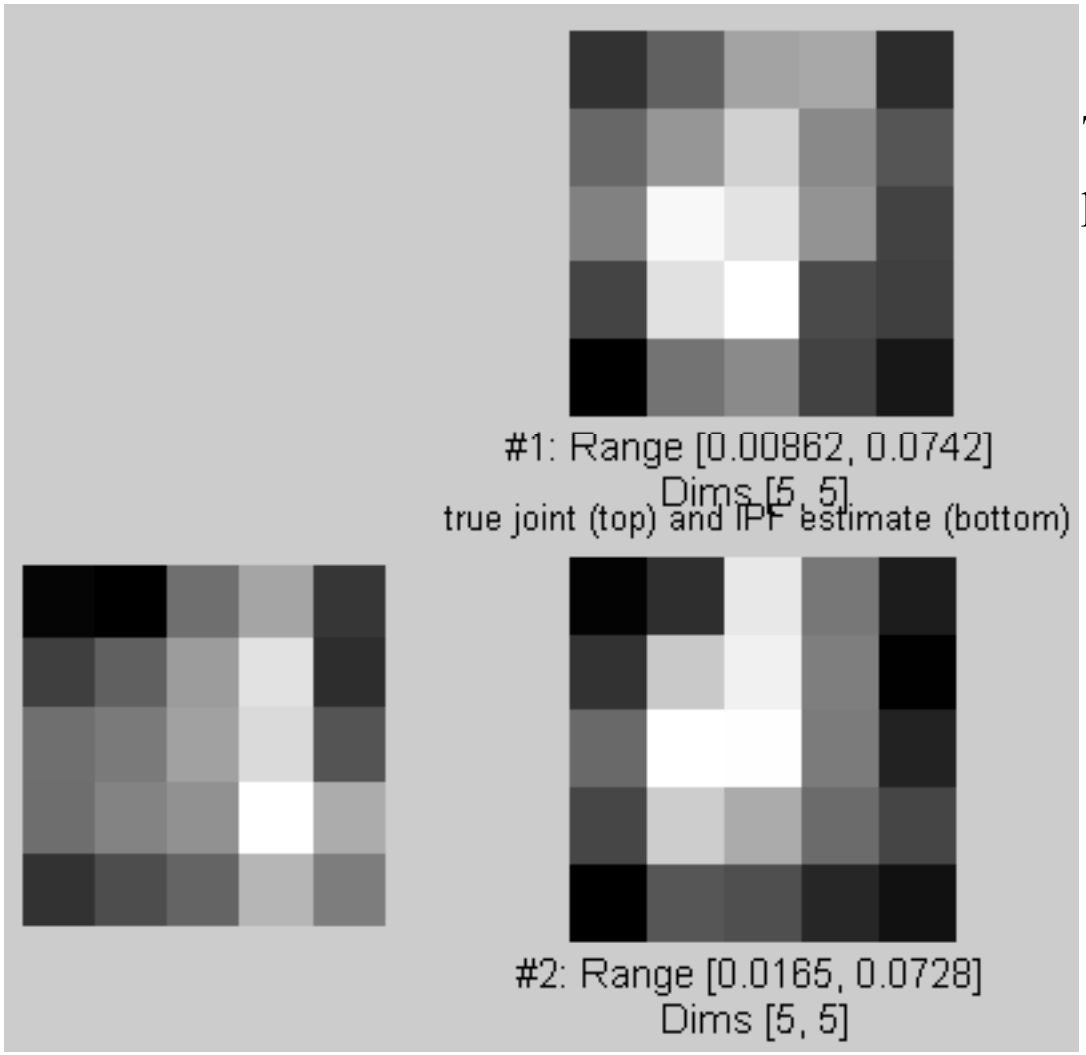
Convergence of to correct marginals by IPF algorithm



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,

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

- This leads to the IPF update rule for $\phi_c(x_c)$

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

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

Reference: unpublished notes by Michael Jordan

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

Slide Credits

- Bill Freeman
- Yuri Boykov
- others, as noted…