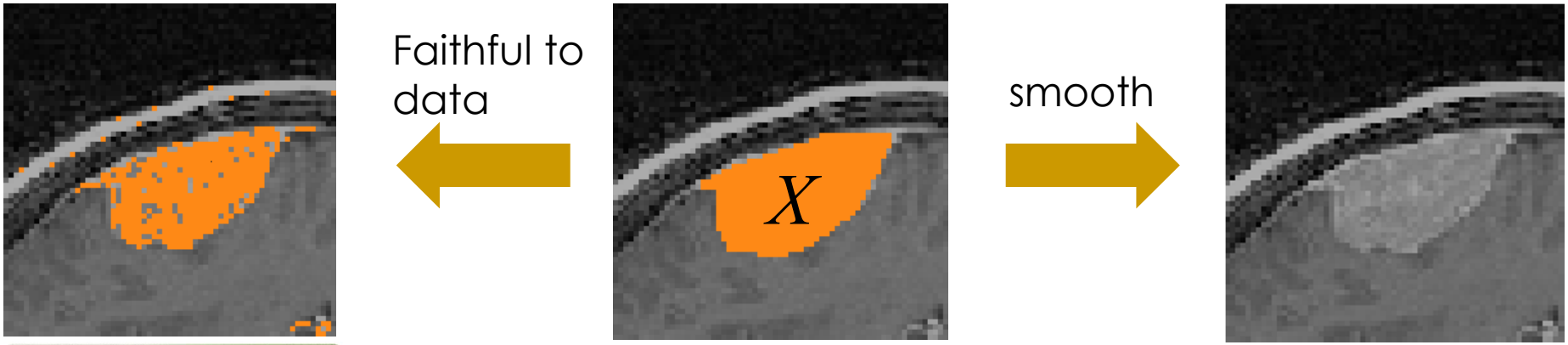
Proposal Selection in Higher-order Graph Cuts

Hiroshi Ishikawa
Department of Computer Science & Engineering
Waseda University

Example: Segmentation

- Local model
  - ex.: Models of pixel values for each kind of tissue
- Prior model / regularization
  - Assume smoothness
- Express the tradeoff by an energy $E(X)$
  - Faithful to the data and model and smooth



Example: Segmentation

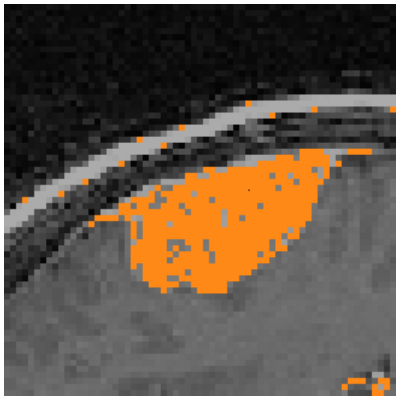
- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

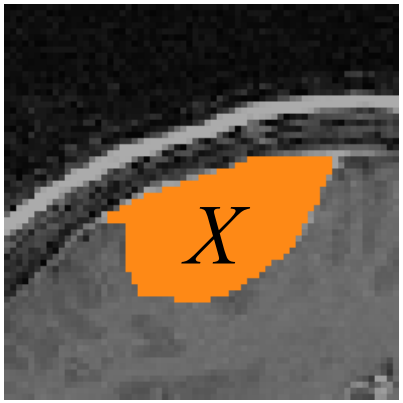
$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} \kappa |X_u - X_v|$$

All pixels

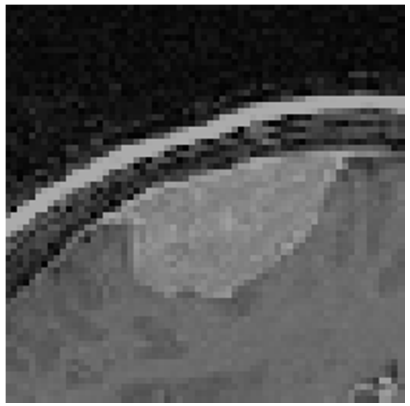
Neighboring pairs of pixels



Faithful to data



smooth



Example: Segmentation

- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

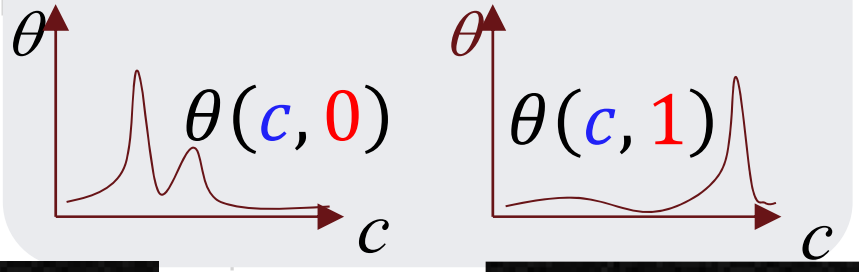
Data term
Based on the data and the image formation model

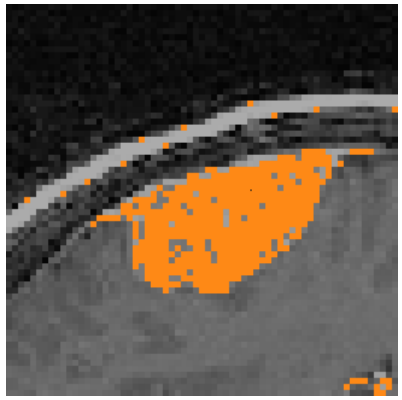
$$\sum_{v \in V} g_v(X_v)$$

All pixels

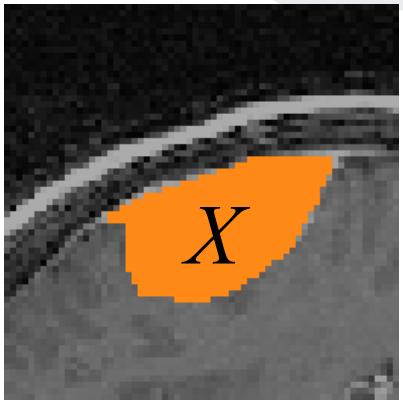
Data

$$g_v(l) = -\log \theta(I_v, l), l = 0,1$$

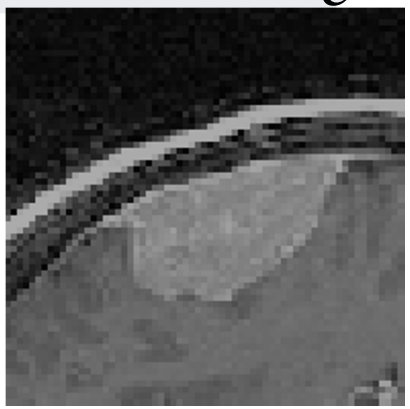




Faithful to data



smooth



Example: Segmentation

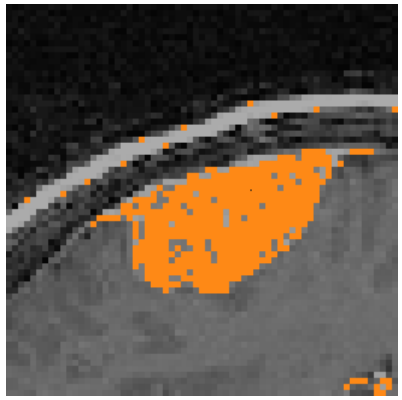
- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

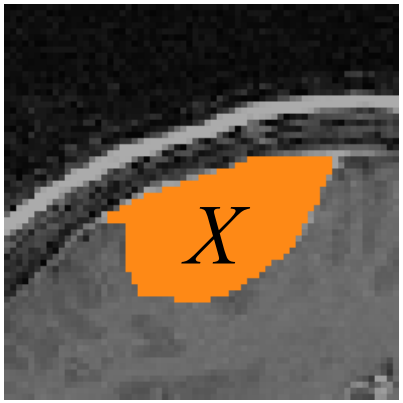
0 if neghiboring labels coincide
$\kappa > 0$ if they differ

$$\sum_{(u,v)\in E} \kappa |X_u - X_v|$$

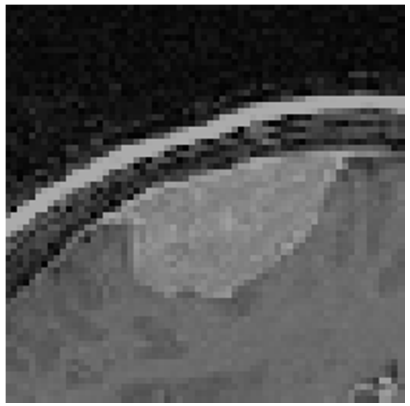
Neighboring pairs of pixels



Faithful to data



smooth



Energy Minimization

- Consider the energy of the form

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

Data term Smoothing term

where V is the set of locations (sites)

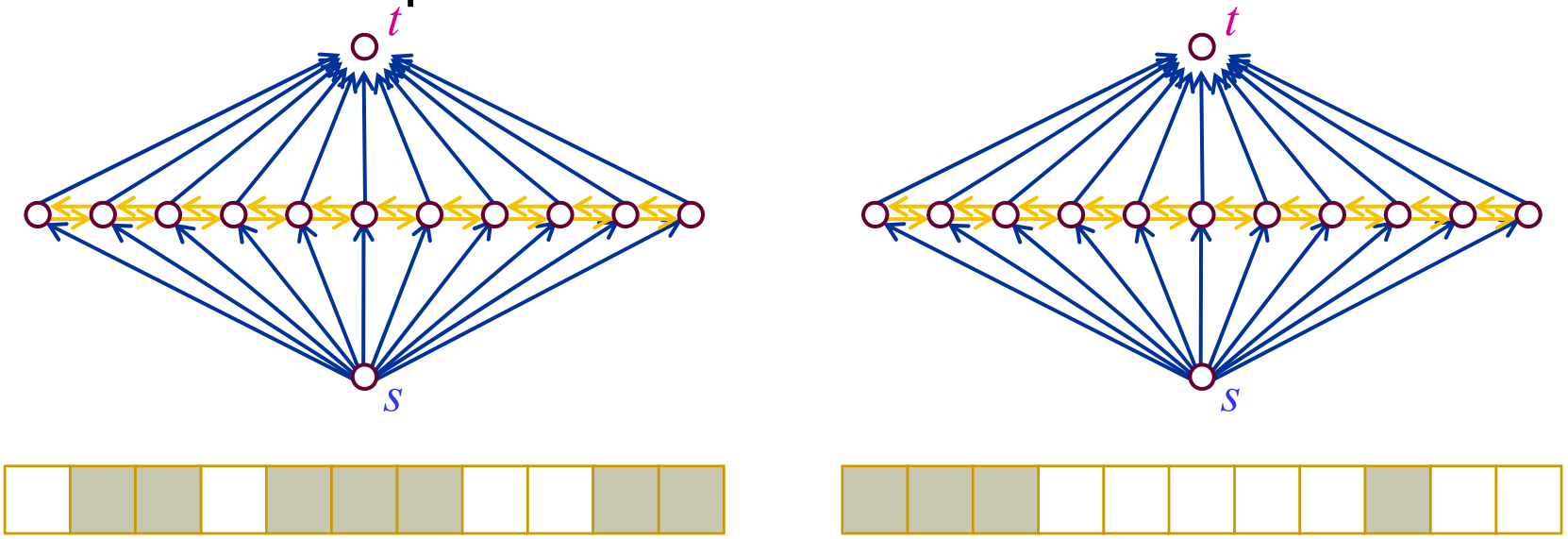
E is the set of neighboring pairs of sites

X assigns a label to each site in V

- 1st order Markov Random Field (MRF)
- Problem: Find the X that minimizes $E(X)$

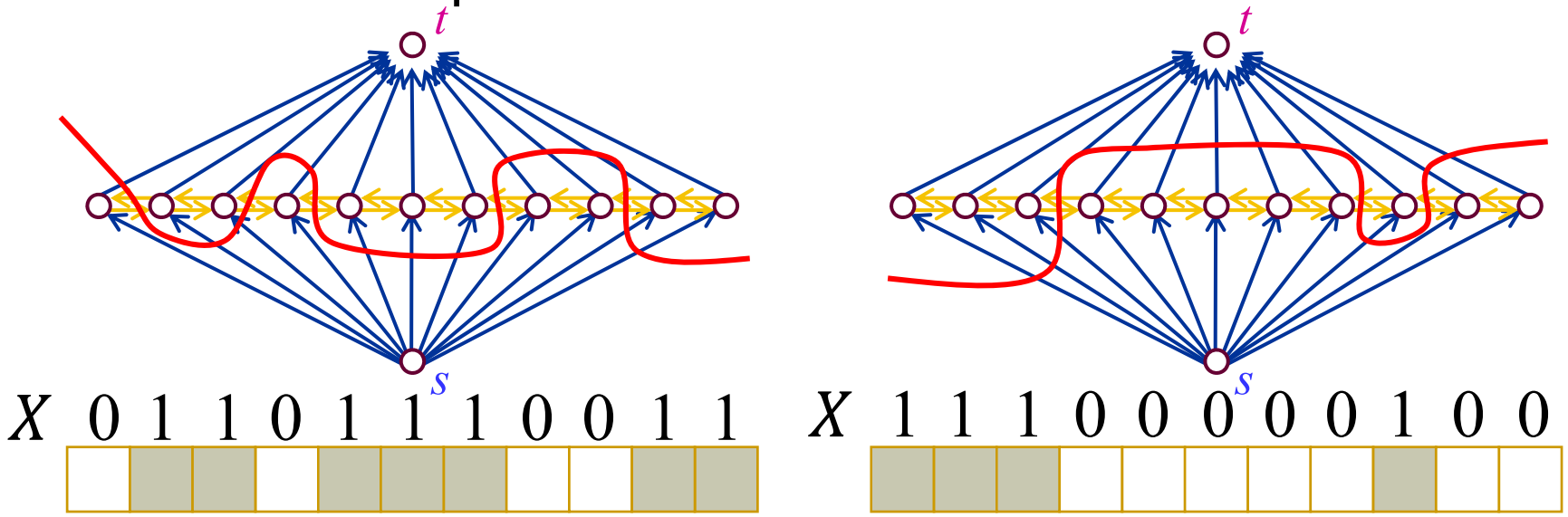
Energy minimization by minimum cuts (binary case)

- 1:1 correspondence between X and cuts



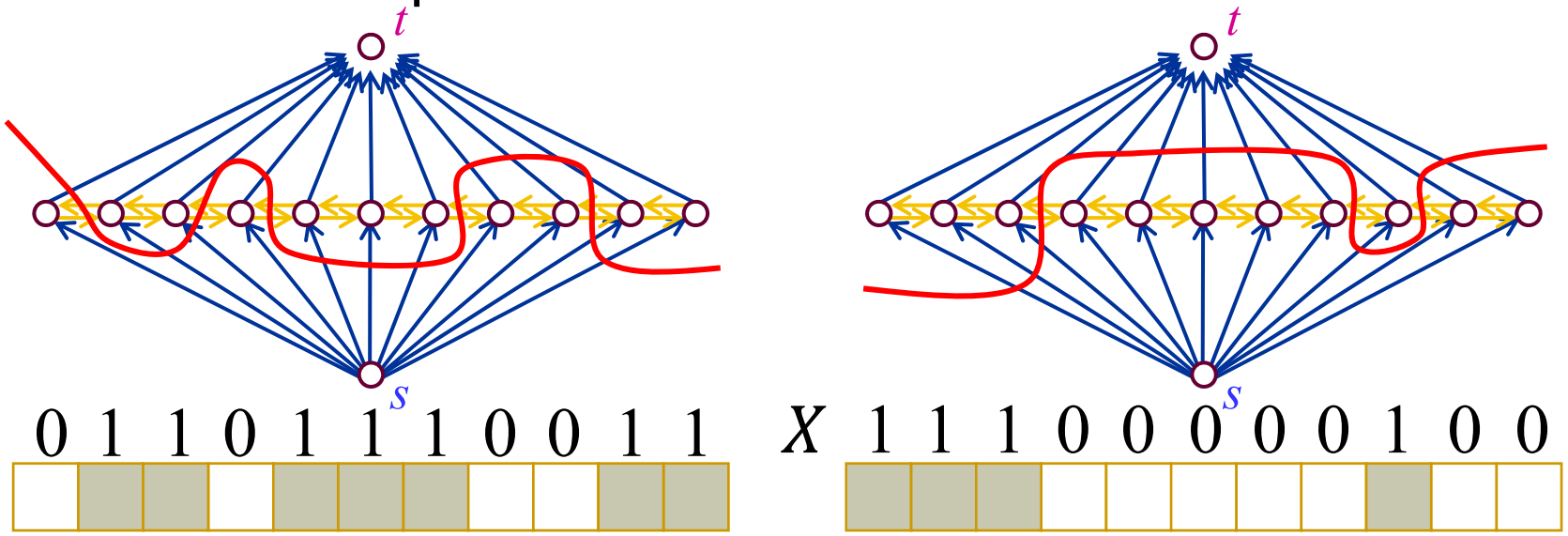
Energy minimization by minimum cuts (binary case)

- 1:1 correspondence between X and cuts



Energy minimization by minimum cuts (binary case)

- 1:1 correspondence between X and cuts



- Energy = Cut cost
- Mincut → Global Energy minimization

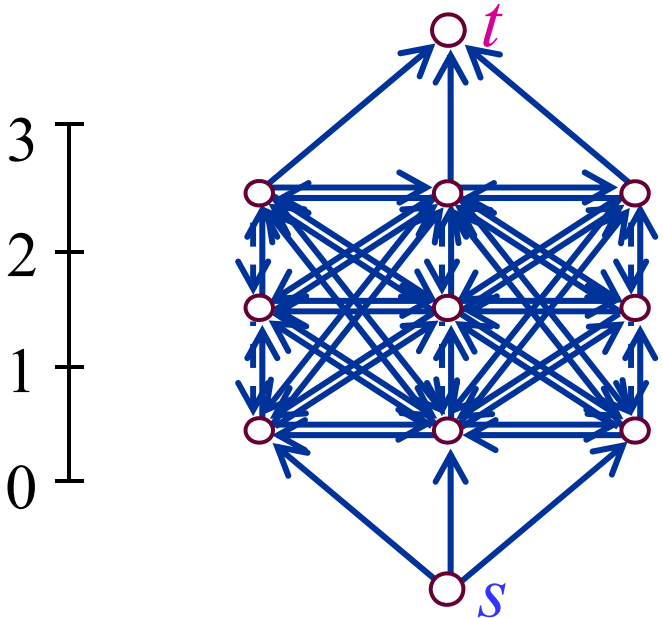
Submodularity Kolmogorov & Zabih IEEE TPAMI2004

$$h_{uv}(0,0) + h_{uv}(1,1) \leq h_{uv}(0,1) + h_{uv}(1,0)$$

Energy minimization by minimum cuts (≥ 3 labels)

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

- If the L has linear order $L = \{l_0, l_1, \ldots, l_k\}$
  - Globally minimizeable $\Leftrightarrow$ $h_{uv}(l_i, l_j)$is a convex function of $i - j$



Ishikawa IEEE TPAMI 2003

≥ 3 labels, approximation

Move-making algorithms

- Iterative approximation algorithms
- In each iteration, finds the globally optimal move using binary graph cuts

≥ 3 labels, approximation

Move-making algorithms

- Iterative approximation algorithms
- In each iteration, finds the globally optimal move using binary graph cuts

Move

- $\alpha\beta$ swap
  - Allows label changes $\alpha \rightarrow \beta$, $\beta \rightarrow \alpha$ only
- α-expansion
  - Allows changing to α only

Boykov, Veksler & Zabih IEEE TPAMI2001

α-expansion

In each iteration, the α area expands

Initial

-expansion

-expansion

-expansion

-expansion

-expansion

-expansion

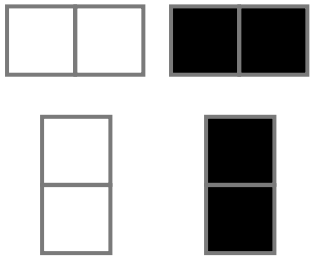
-expansion

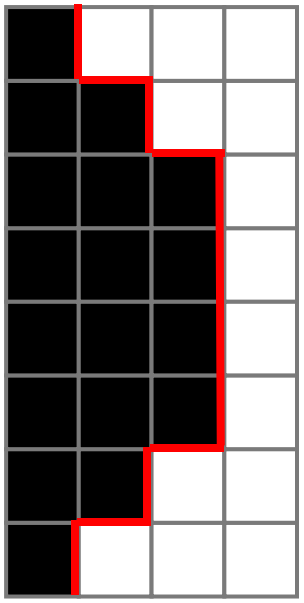
Choose the move that minimizes best : binary optimization

Courtesy Yuri Boykov

First-order energy

Good (Low Energy)

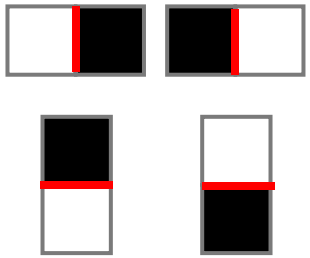


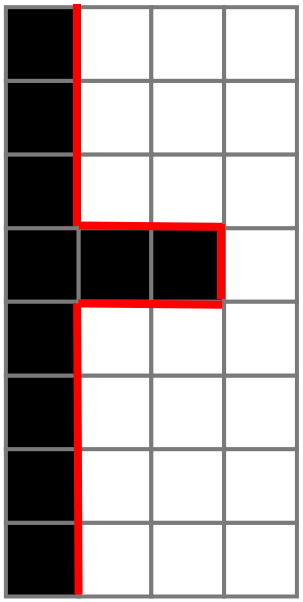


12 Bad

40 Good

Bad (High Energy)



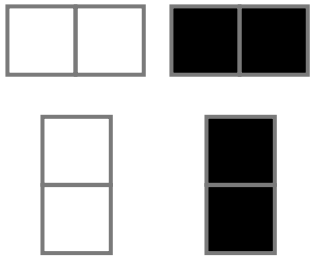


12 Bad

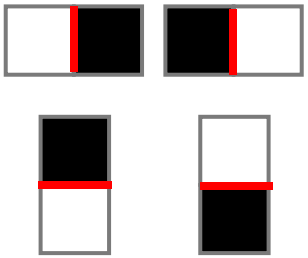
40 Good

Higher-order energy

Good (Low Energy)

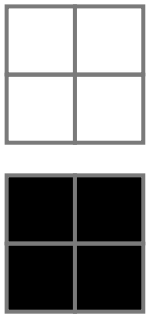


Bad (High Energy)

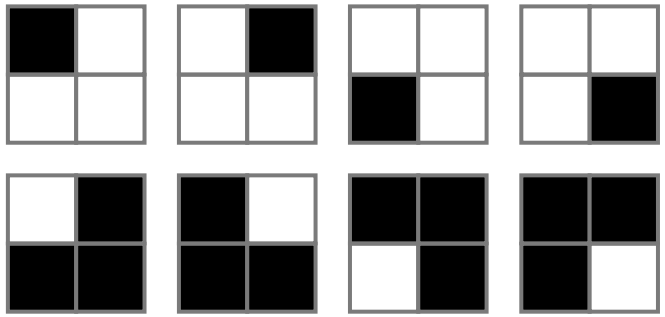


Better (Lower Energy) ⟷ Worse (Higher Energy)

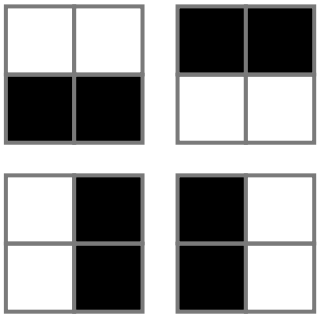
A



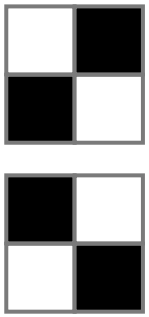
B



C



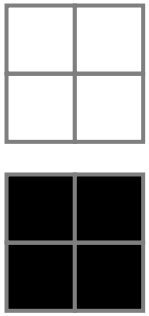
D



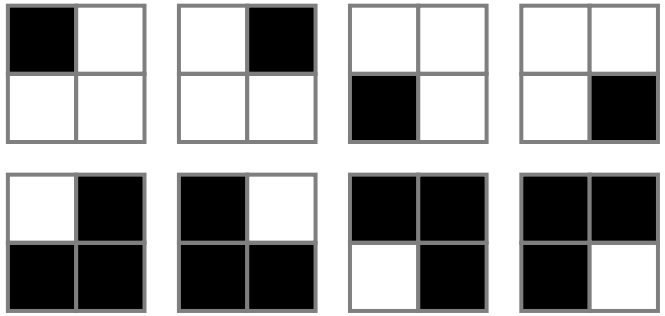
Higher-order energy

Better (Lower Energy) ⟷ Worse (Higher Energy)

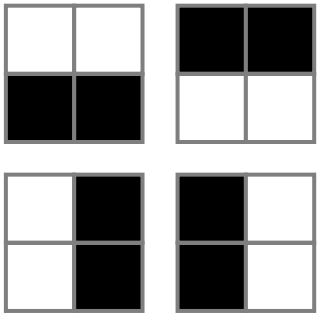
A



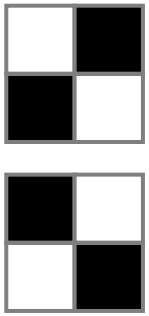
B

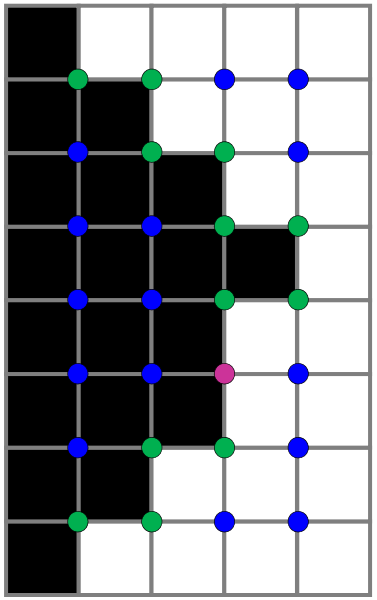


C



D



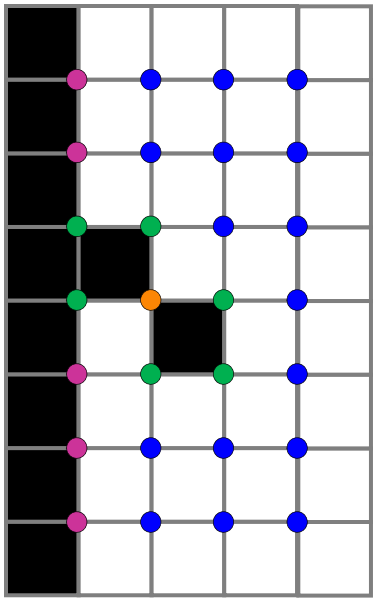


A: 15

B: 12

C: 1

D: 0



A: 16

B: 6

C: 5

D: 1

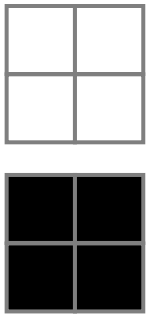
Higher-order energy

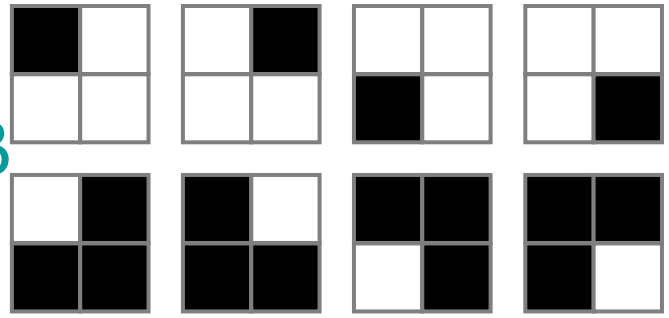
$$E(X) = \sum_{C \in \mathscr{C}} f_C(X_C)$$

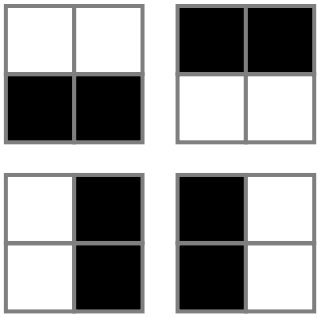
$$= \sum_{v \in V} g_v(X_v) + \sum_{(u,v)} h_{uv}(X_u, X_v)$$

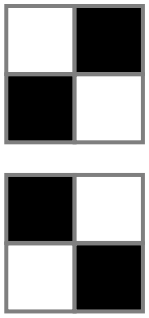
$$+ \sum_{(u,v,s,t)} k_{uvst}(X_u, X_v, X_s, X_t)$$

Better (Lower Energy) ⟷ Worse (Higher Energy)

A 

B 

C 

D 

Higher-order energy

Ishikawa CVPR2009, PAMI2011; Gallagher et al. CVPR2011
Fix et al. ICCV2011; Kahl & Strandmark ICCV2011

- Transform arbitrary higher-order binary energy

$$E(X) = E(X_1, \cdots, X_n) = \sum_{C \in \mathscr{C}} f_C(X_C)$$

into an equivalent first-order energy

$$\tilde{E}(\tilde{X}) = \tilde{E}(X_1, \cdots, X_n, \cdots, X_m) = \sum g_v(X_v) + \sum h_{uv}(X_u, X_v)$$

- Adds variables

Higher-order energy

Ishikawa CVPR2009, PAMI2011; Gallagher et al. CVPR2011
Fix et al. ICCV2011; Kahl & Strandmark ICCV2011

- Transform arbitrary higher-order binary energy

$$E(X) = E(X_1, \cdots, X_n) = \sum_{C \in \mathscr{C}} f_C(X_C)$$

into an equivalent first-order energy

$$\tilde{E}(\tilde{X}) = \tilde{E}(X_1, \cdots, X_n, \cdots, X_m) = \sum g_v(X_v) + \sum h_{uv}(X_u, X_v)$$

- Adds variables
- More than 2 labels → Fusion moves

Multiple labels: Fusion Move

Assume labels $L = \{l_1, \cdots, l_N\}$

Labeling Y assigns a label Y_v to each v

Fusion Move

Lempitsky et al. ICCV2007

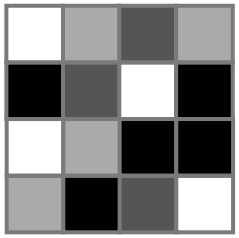
Iteratively update Y:

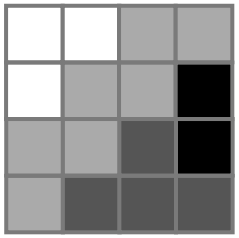
1. Generate a proposed labeling P
2. Merge Y and P

The merge defines a binary problem:

"For each v, change Y_v to P_v or not"

Multiple labels: Fusion Move





0	1	0	0
1	0	1	0
0	1	1	0
1	1	0	0

Fusion Move

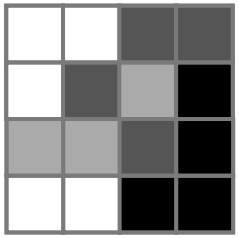
Iteratively update Y:

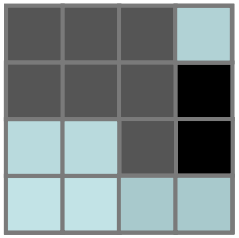
1. Generate a proposed labeling P
2. Merge Y and P

The merge defines a binary problem:

"For each v, change Y_v to P_v or not"

Multiple labels: Fusion Move





0	0	0	1
0	0	0	0
1	1	0	0
1	1	1	1

Fusion Move

Iteratively update Y:

1. Generate a proposed labeling P
2. Merge Y and P

The merge defines a binary problem:

"For each v, change Y_v to P_v or not"

Fusion Move with Roof Dual

Roof duality

Hammer et al. 1984, Boros et al. 1991, 2006

Minimizes submodular E globally

For non-submodular E, assigns each pixel

0, 1, or unlabeled

With fusion move, **by keeping unlabeled pixels unchanged, E doesn't increase**

Fusion Move with Roof Dual

Roof duality

Hammer et al. 1984, Boros et al. 1991, 2006

Minimizes submodular E globally

For non-submodular E, assigns each pixel

0, 1, or unlabeled

With fusion move, **by keeping unlabeled pixels unchanged, E doesn't increase**

Example: Denoising by FoE

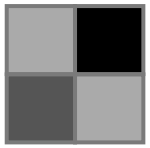
FoE (Fields of Experts) Roth & Black CVPR2005

A higher-order prior for natural images

$$E(Y) = \sum_{C \in \mathscr{C}} f_C(Y_C)$$

$\mathscr{C}$: a set of cliques

$$Y_C = (Y_v)_{v \in C}$$



$$f_C(Y_C) = \sum_{i=1}^{K} \alpha_i \log\left(1 + \frac{1}{2}(J_i \cdot Y_C)^2\right)$$

$$f_{\{v\}}(Y_{\{v\}}) = \frac{(N_v - Y_v)^2}{2\sigma^2}$$

Example: Denoising by FoE

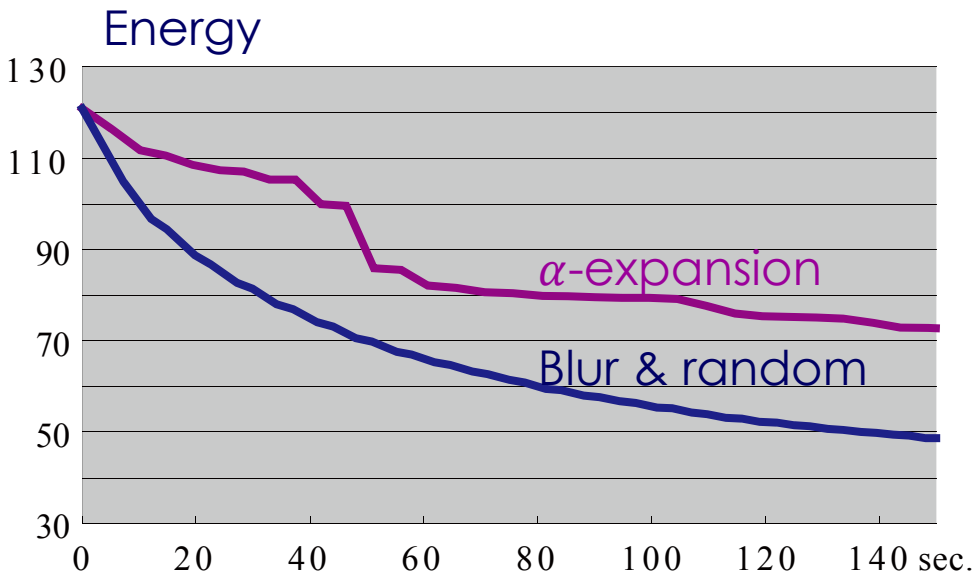
Original

Noise-added

3rd order

Proposal Selection

- α-expansion works well with the Potts energy because the proposal (constant) is in the null space of the prior
- With higher-order energy, selection of the proposal is more subtle



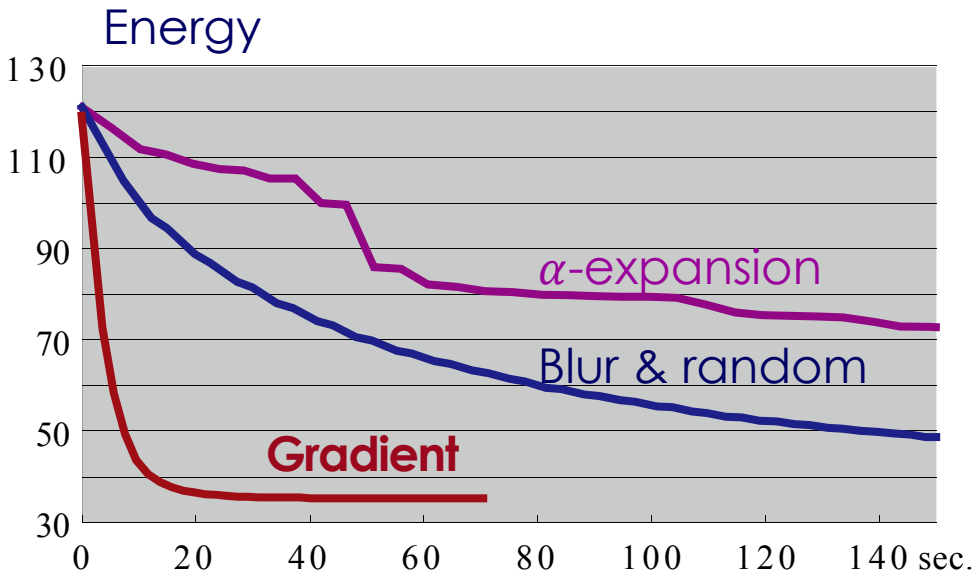
α-expansion

Blur & random

"Higher-order Gradient Descent"

- Move the current labeling X by the gradient of E to generate the proposal P

$$P = X - \eta \,\mathrm{grad}E(X) \qquad P_v = X_v - \eta \frac{\partial E}{\partial X_v}$$



Gradient

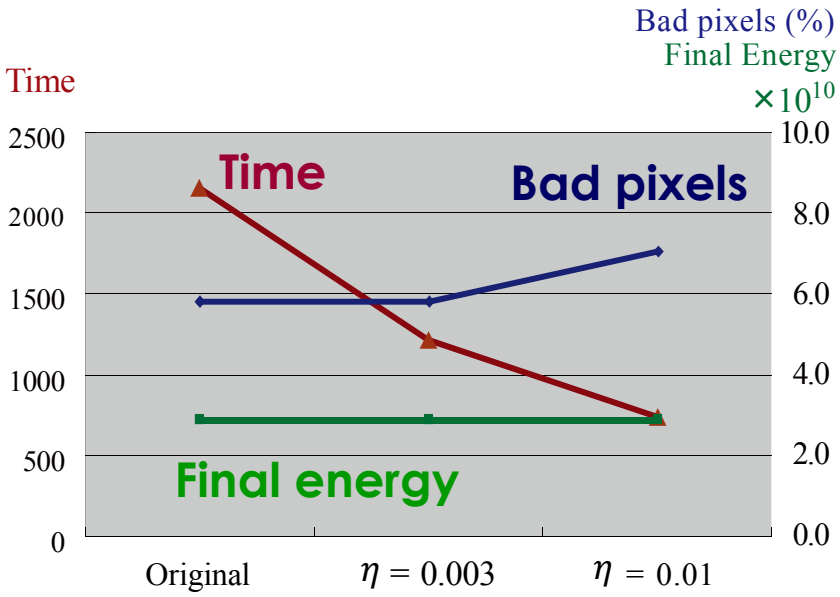
Blur & random

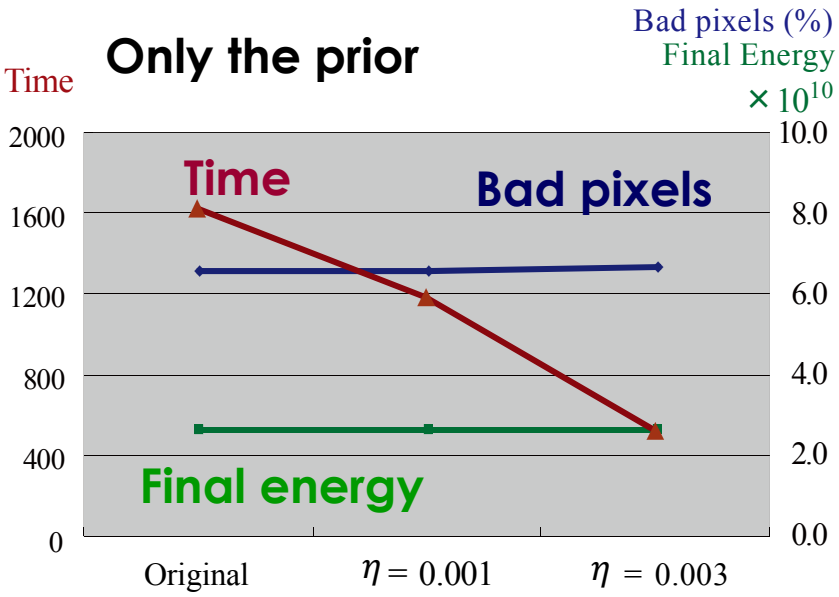
"Higher-order Gradient Descent"

- Not quite gradient descent
  - The gradient of even a part of the energy (e.g. only the higher-order, prior part) can be used
  - The descent step (η) can safely be made quite large
- These can safely be done because the actual move is guarded against increasing the energy by graph cut

“Higher-order Gradient Descent”

Second-order stereo (Woodford et al. CVPR2008)





Summary

- Labeling problem
- Graph Cut
  - Binary
  - Multiple label
- Higher-order energy
- Fusion move
- Higher-order Gradient descent