THE AUSTRALIAN NATIONAL UNIVERSITY

MRF Optimization

OXFORD BROOKES UNIVERSITY

Western

Thanks to

Phil Torr,

Oxford Brookes University,

Yuri Boykov

University of Western Ontario

also

Manish Jethwa, Pushmeet Kohli, Pawan Kumar, Yuri Boykov, George Vogliatzsis, Chria Bishop, Bill Freeman, Peter Carr and many others.

THE AUSTRALIAN NATIONAL UNIVERSITY

NICTA

The task of segmentation

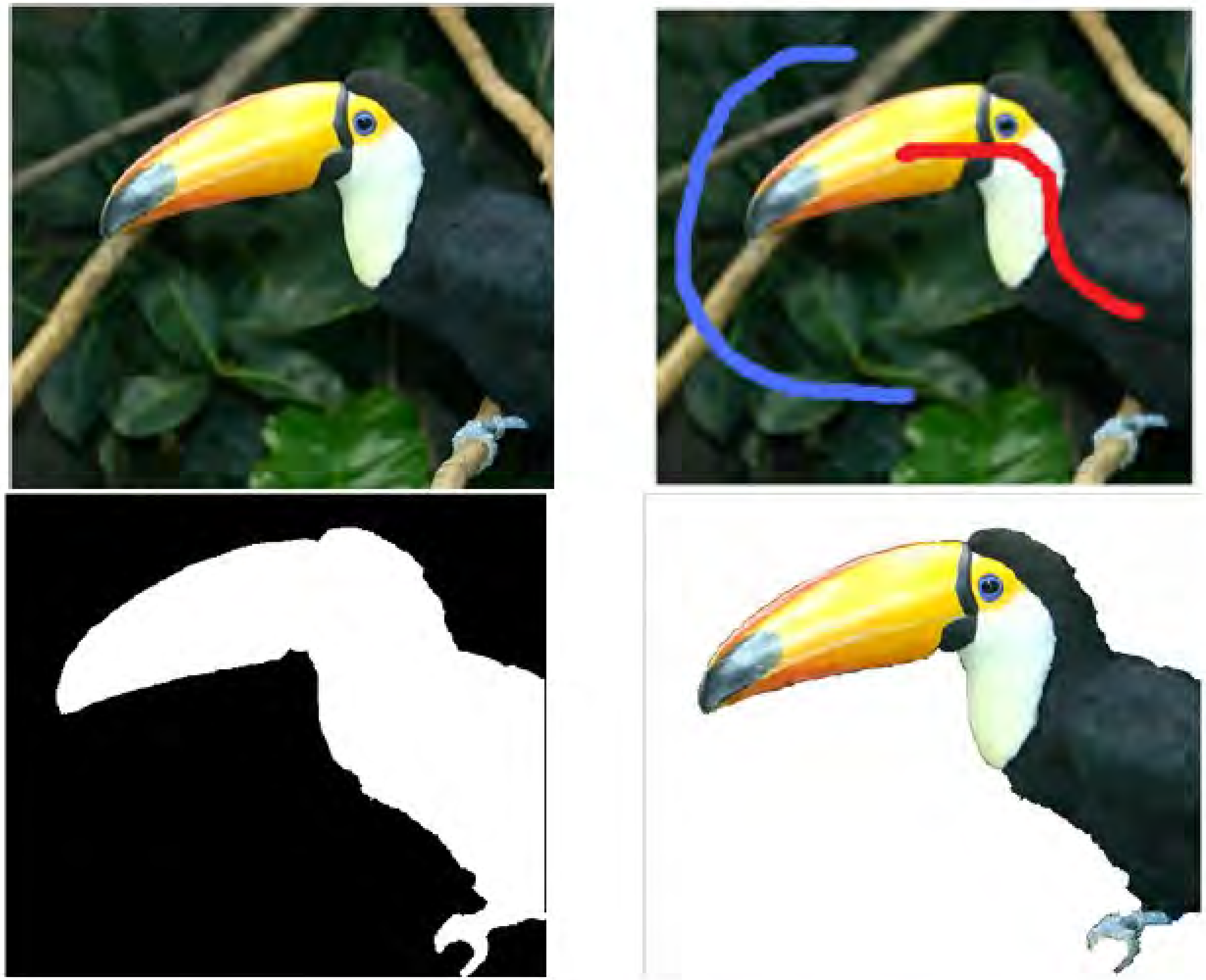


Fig. 1.3. *Here we wish to distinguish a foreground object (the toucan) from the background (the jungle). From top left to bottom right: original image; a set of user provided scribbles; the matte automatically generated by the algorithm of Boykov and Jolly and the resulting extracted object of interest: the toucan.*

Select points in background and foreground – gives sample points in RGB space, belonging to two classes.

Classify points according to probability they belong to given class.

Data (**D**)

Unary likelihood

Results will not be great

Regularization by locality.

- Close-by points should belong to the same class
- Segmentation edges usually occur where there is an intensity edge.

- Add a cost to prevent noisy segmentation – edge terms.

$$E_{\mathbf{I}}(\mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(x_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(x_i, x_j) \ . \qquad (1.6)$$

The first term $E_i(x_i)$, the unary term, simply corresponds to $P(I_i \mid x_i)$ in this case. The second term, $E_{ij}(x_i, x_j)$, the binary term, was given the form

$$E_{ij}(x_i, x_j) = \begin{cases} \beta_{ij} & \text{if } x_i \neq x_j \\ 0 & \text{otherwise} \ , \end{cases}$$

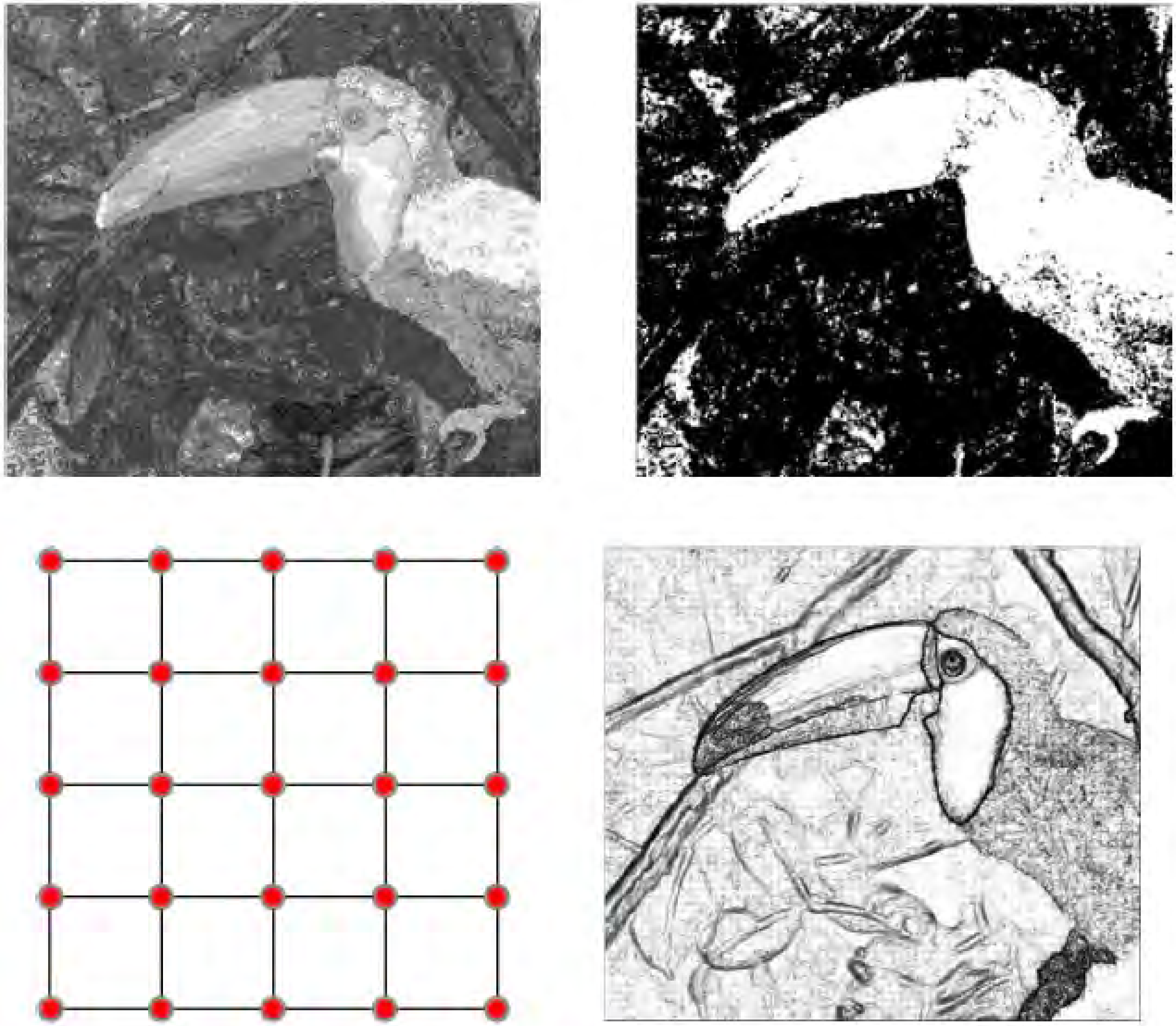
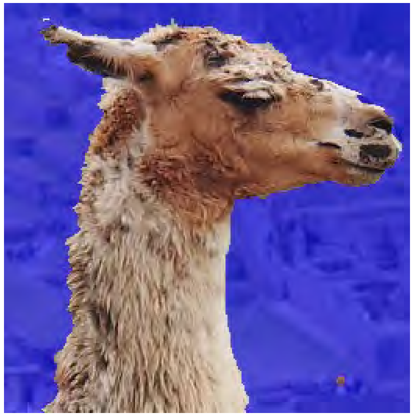


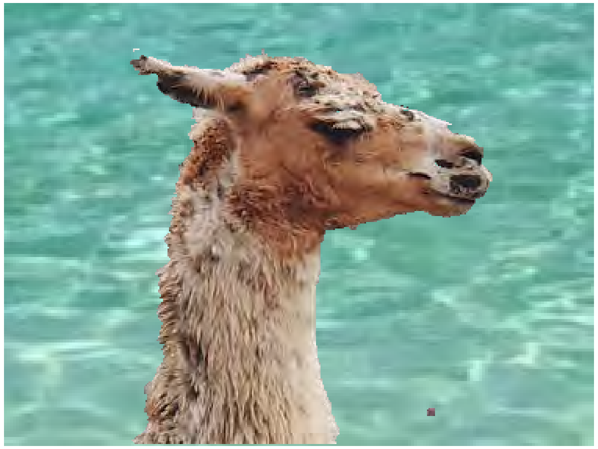
Fig. 1.4. *From top left to bottom: the likelihood of foreground assuming independence of labels as in 1.5; the resulting maximum likelihood segmentation; a nearest neighbour grid graph; data-driven edge weights.*

Fig. 1.11. *Example images from the Pascal VOC segmentation challenge.* **Left** *Original Image.* **Middle** *Object instance segmentation.* **Right** *Class segmentation. Note that black indicates no label. One problem with this data set is that typically only a few object classes, and that much of each of the images is unlabelled.*

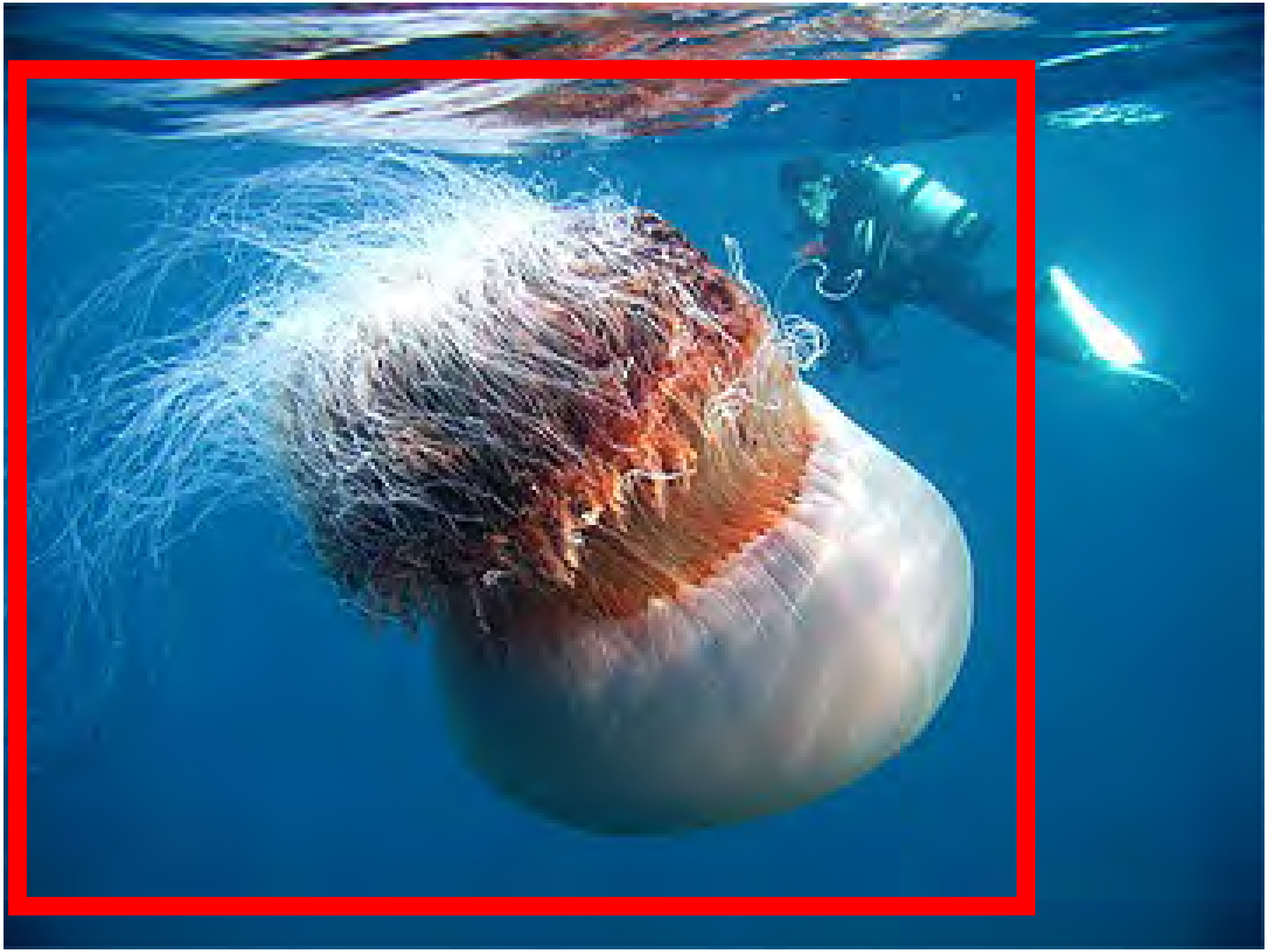
Segmentation with Alpha Matting

Grabcut demo

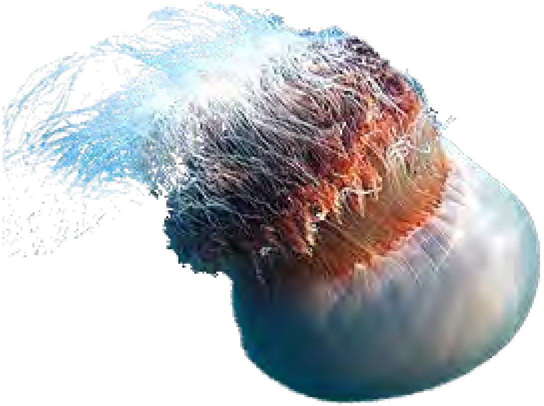




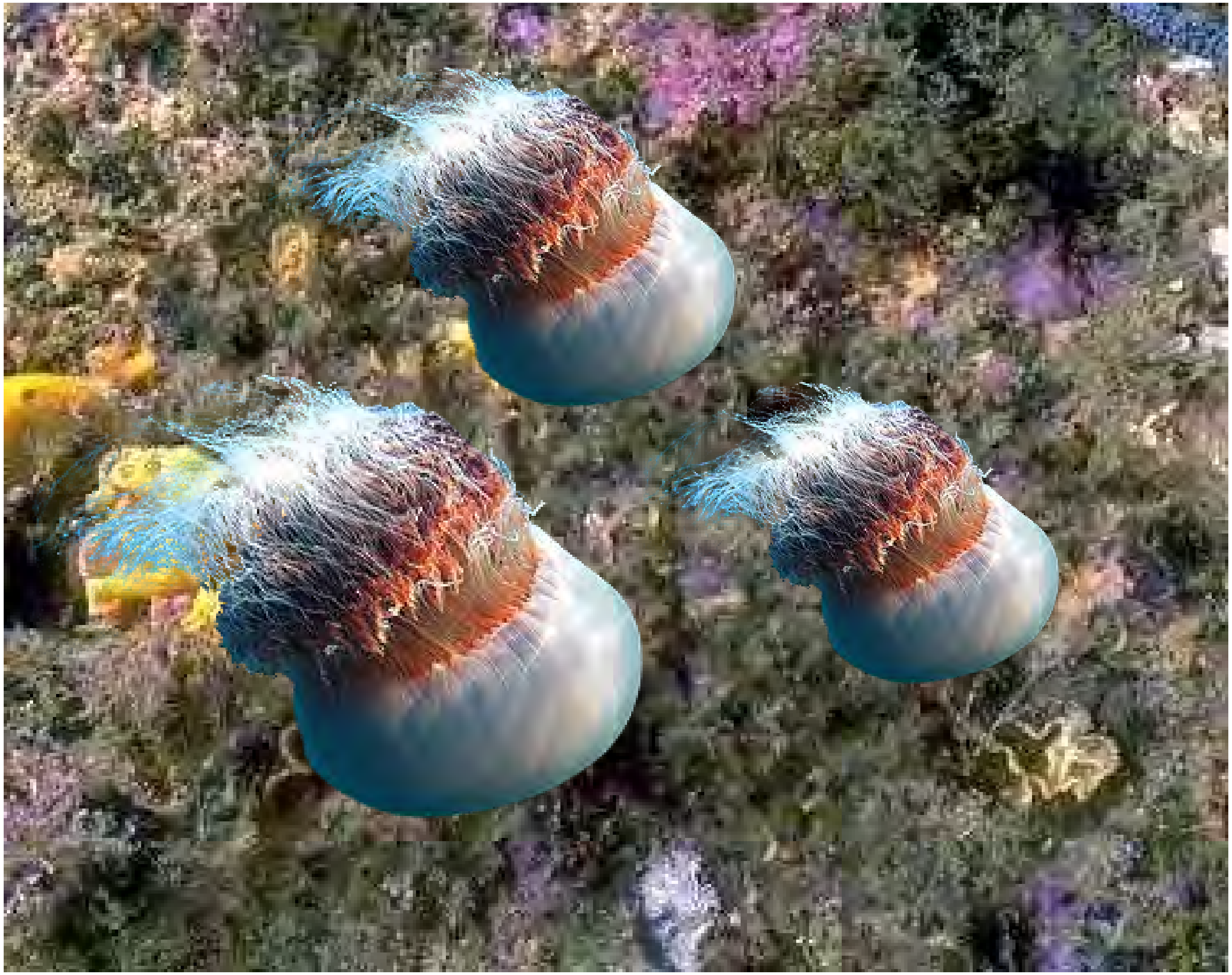
First place rectangle round object



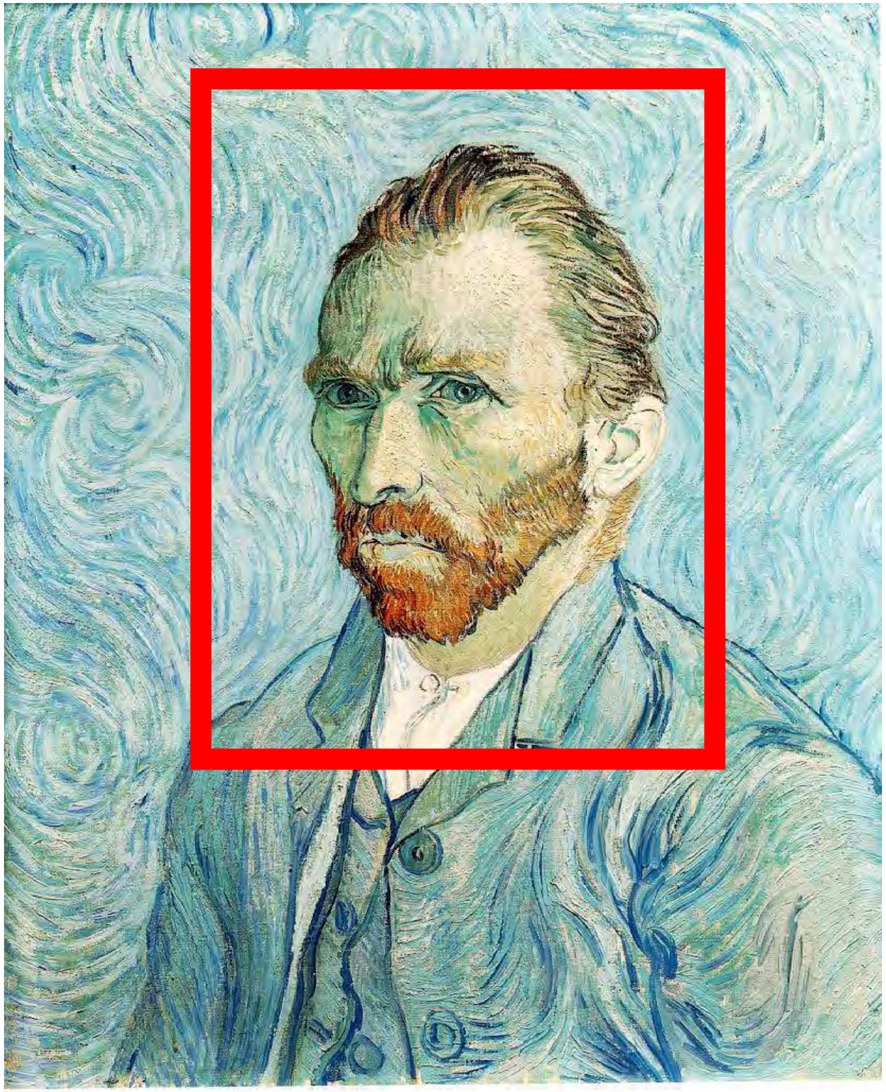
Object plus alpha mask segmented by new secret method.

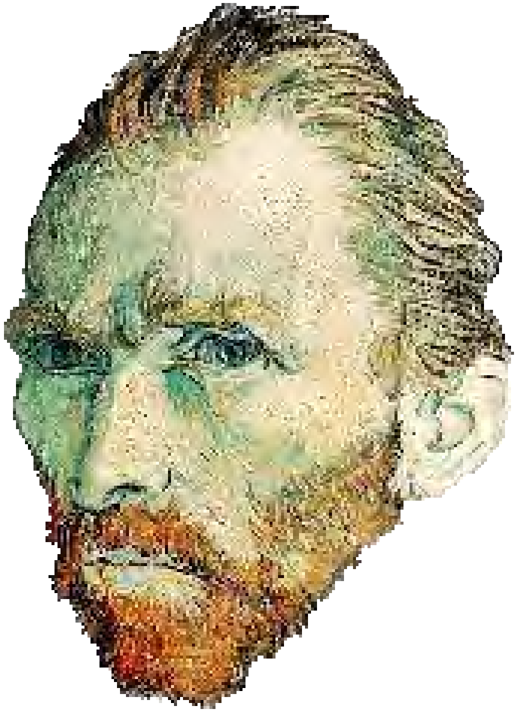


Watch out they are coming!!…note alpha mask

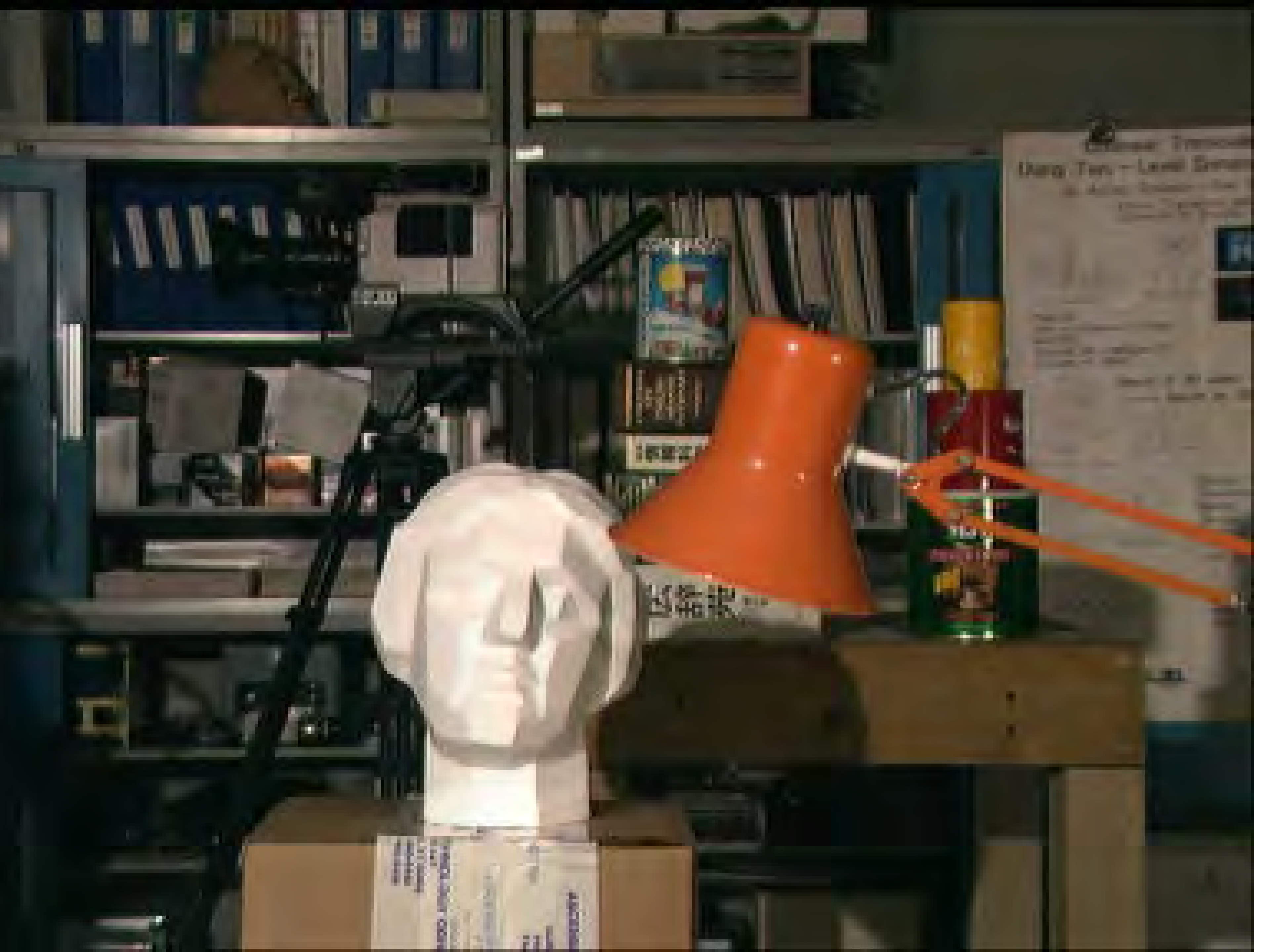


Chop off Van Gogh's head with one click!!





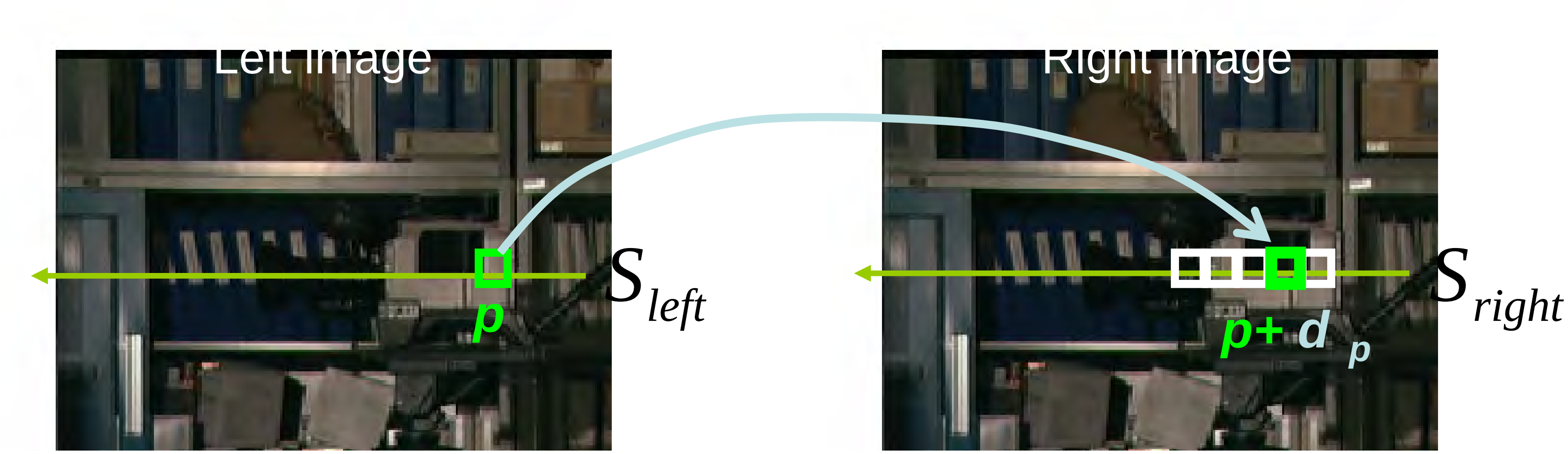
Stereo Matching



Scan-line stereo

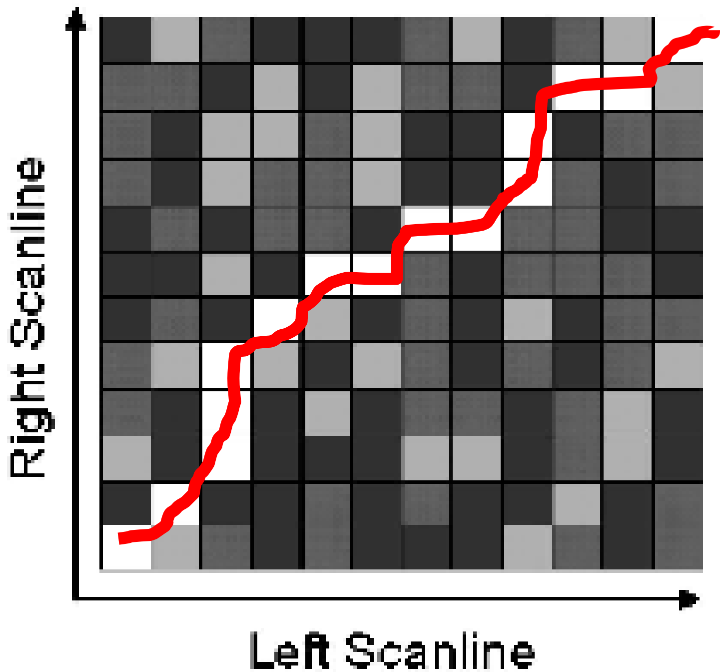
- •Baker & Binford 1981
- •Ohta & Kanade, 1985
- •Geiger at.al. 1992
- •Belhumeur & Mumford 1992
- •Cox at.al. 1996
- •Scharstein & Szelisky 2001

Example:

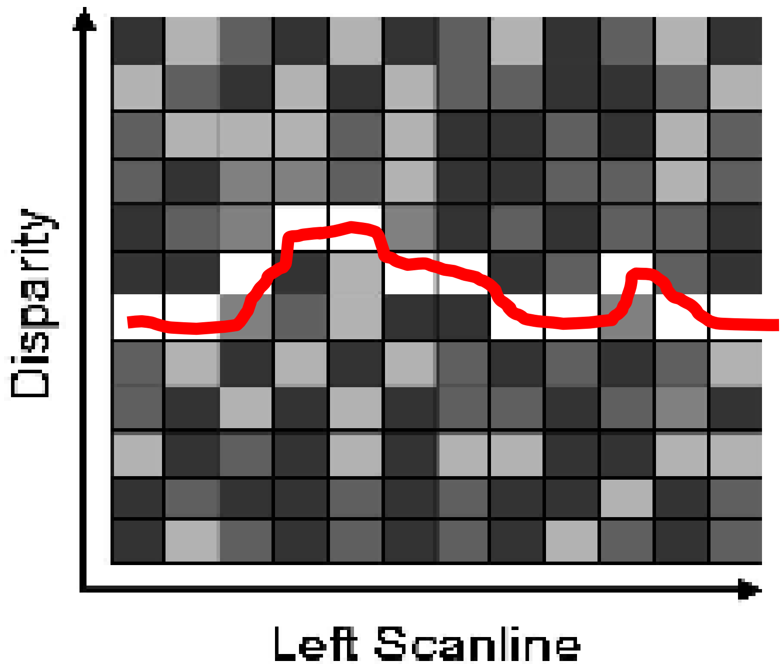


$$E(d_1, d_2, \ldots, d_n) = \sum_{p \in S_{left}} E_p(d_p, d_{p+1})$$

Disparities of pixels in the scan line

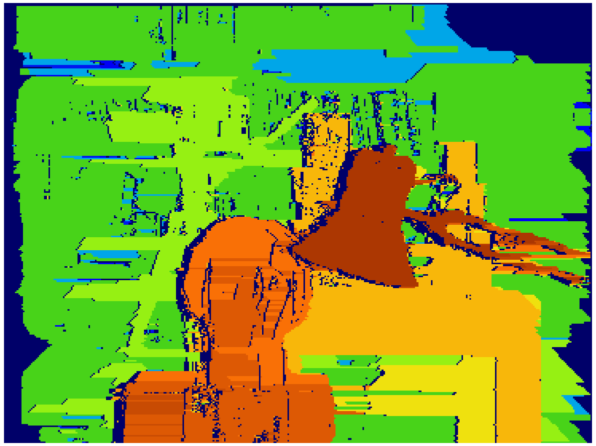


Right Scanline
Left Scanline



Disparity
Left Scanline

Stereo example



Independent scan-lines (via DP)

Multi-scan line (via Graph Cuts)

Ground truth

Graph Cut Textures

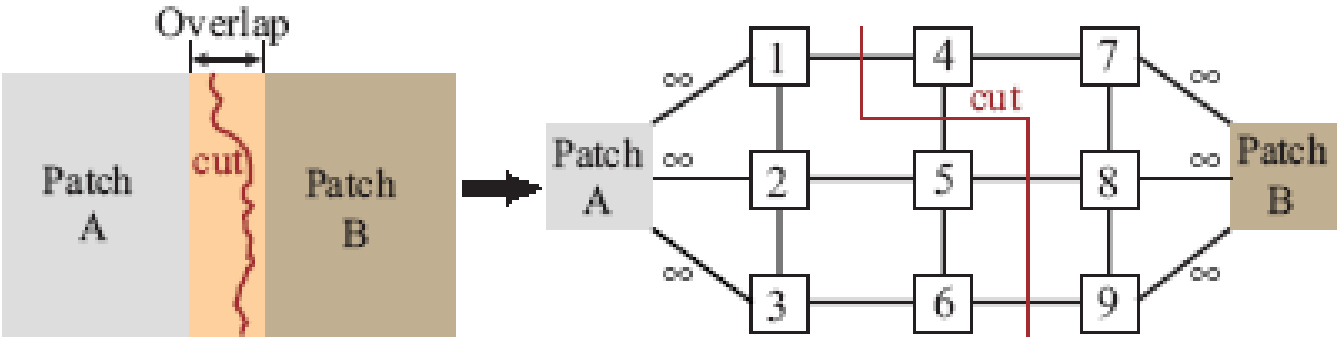
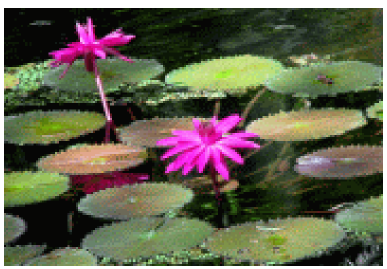


Figure 2: (Left) Schematic showing the overlapping region between two patches. (Right) Graph formulation of the seam finding problem, with the red line showing the minimum cost cut.

Multi-way graph cuts

Graph-cut textures

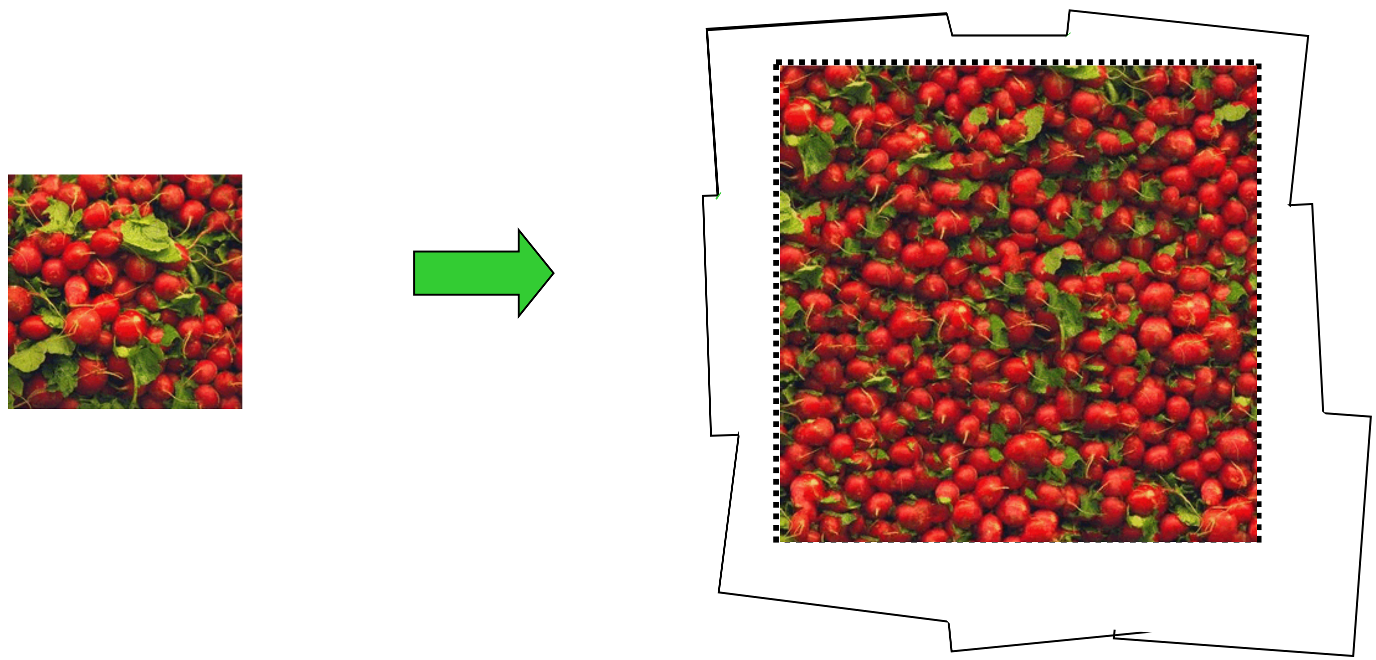
(Kwatra, Schodl, Essa, Bobick 2003)



Shortest paths: Texture synthesis

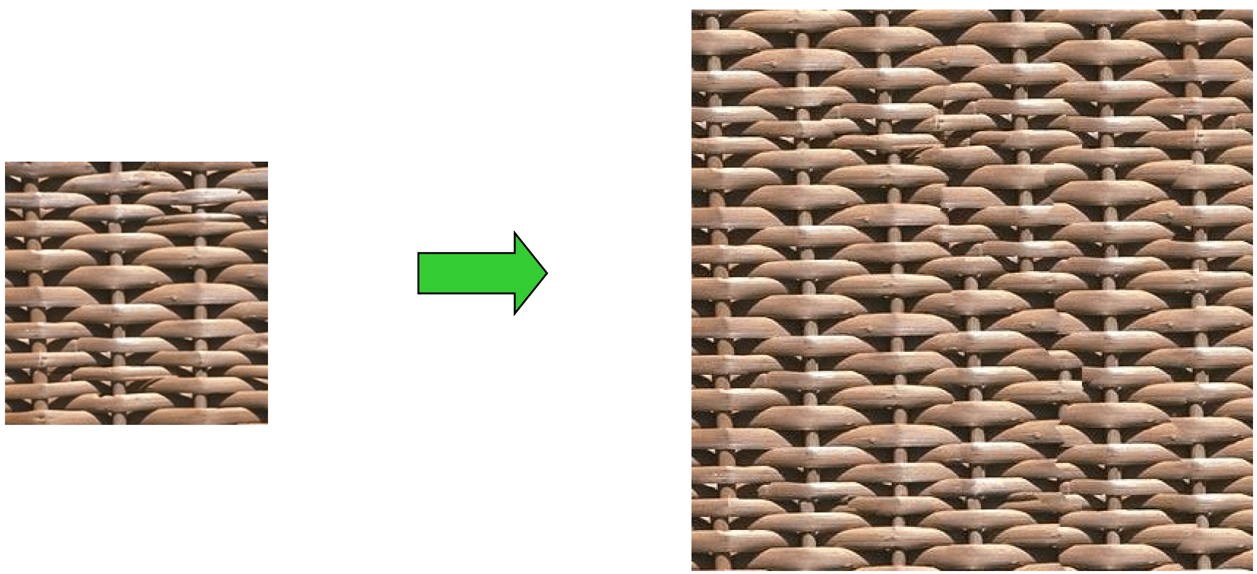
“Image quilting”

Efros & Freeman, 2001



Shortest paths:
Texture synthesis

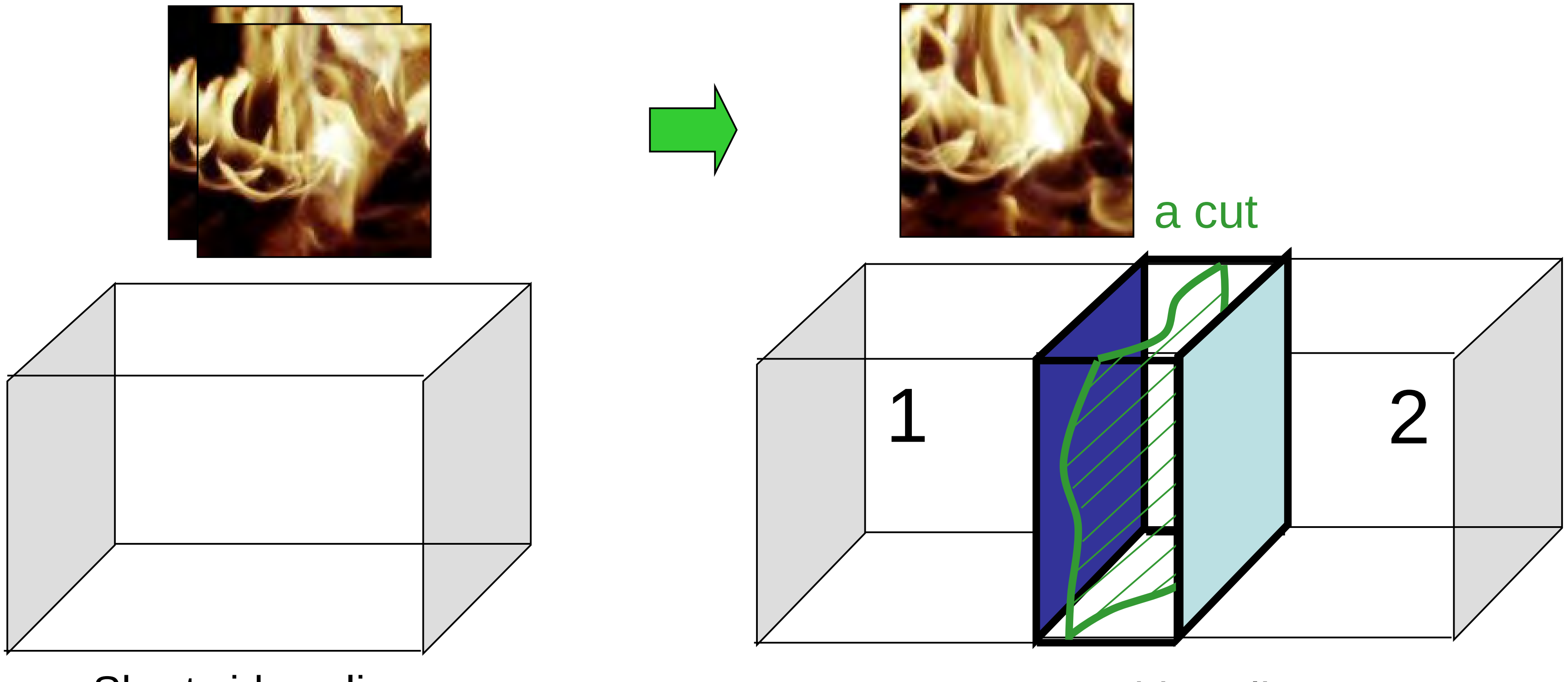
“Image quilting”
Efros & Freeman, 2001



s-t graph cuts for video textures

Graph-cuts video textures

(Kwatra, Schodl, Essa, Bobick 2003)



Short video clip

Long video clip

3D generalization of "**image-quilting**" (Efros & Freeman, 2001)

s-t graph cuts for video textures

Graph-cuts video textures
(Kwatra, Schodl, Essa, Bobick 2003)

original short clip

synthetic infinite texture

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)

Basic Problem of Image Labelling (or Segmentation)

1. How to find the most likely labelling of the graph.
2. Equivalently, how to label the graph with the lowest cost assignment.
3. In 2-label problem, how to separate the nodes into two sets (0 and 1) with the least cost cut of edges in the graph.

MRFs and graphs

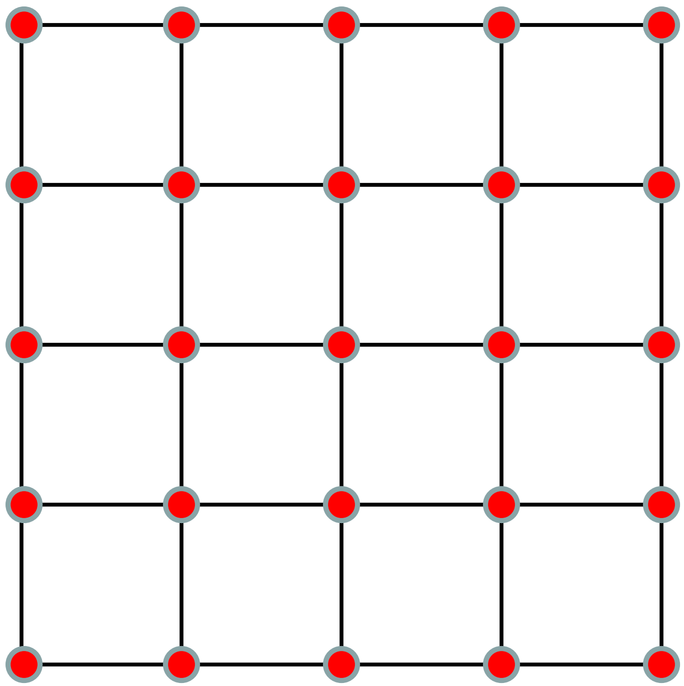
Define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that

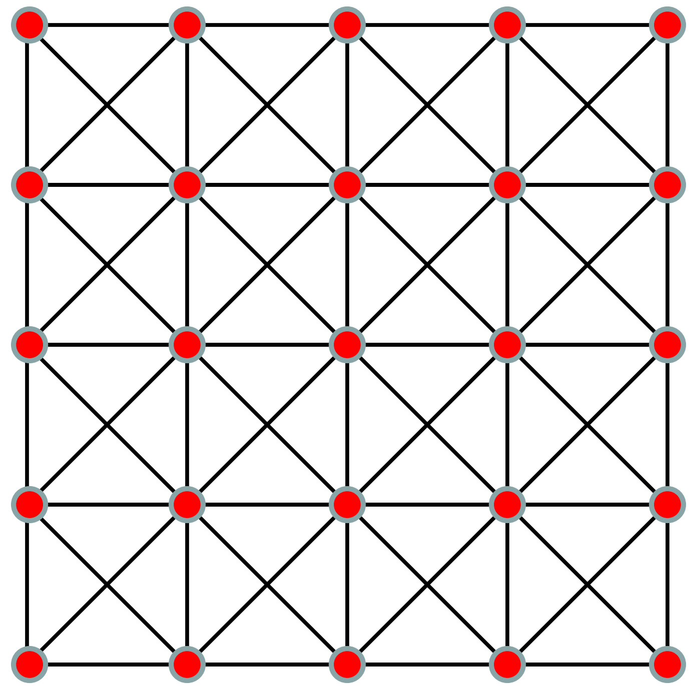
1. The vertices $\mathcal{V}$ are in one-to-one correspondence with the random variables X_i. (In fact we will refer to the vertices as X_i.)
2. There is an edge from X_i to X_j if and only if $i \in \mathcal{N}_j$.

Example: Image graph (4-connected) has cliques of size 1 and 2.

- size 1 (vertices)
- size 2 (pairs of vertices joined by an edge).

6-connected and 8-connected graphs have cliques of size 3 and 4 respectively.





4-connected graph

- Each vertex connected to 4 neighbors.
- Cliques of size 1 (vertices) and 2 (edges).

8-connected graph

- Each vertex connected to 8 neighbors.
- Cliques of size 1, 2, 3 and 4 .
- Energy function might still only use cliques of size 1 and 2.

MRFs and Energy Functions

Practical considerations

Two basic problems.

- How to choose the right energy function.
- If we have the right energy functions, can we minimize it?
- The general energy minimization problem is NP-hard.

$$E_{\mathrm{I}}(\mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(x_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(x_i, x_j) \ . \tag{1.6}$$

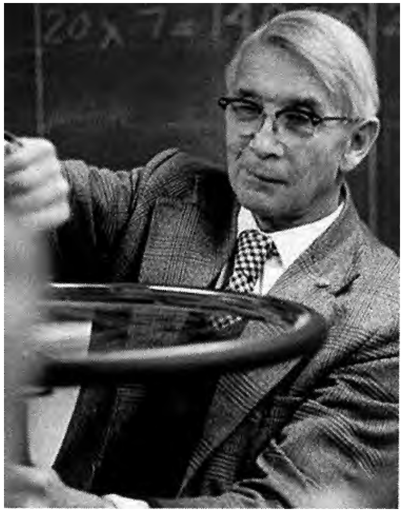
The first term $E_i(x_i)$, the unary term, simply corresponds to $P(I_i \mid x_i)$ in this case. The second term, $E_{ij}(x_i, x_j)$, the binary term, was given the form

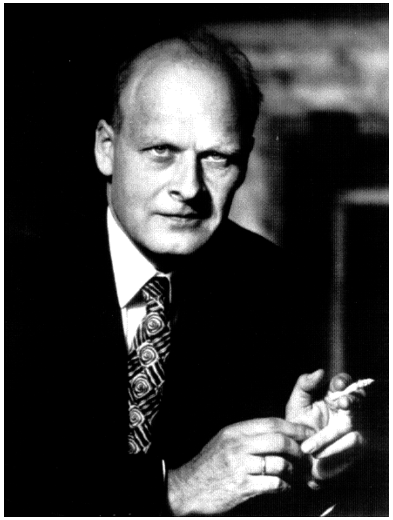
$$E_{ij}(x_i, x_j) = \begin{cases} \beta_{ij} & \text{if } x_i \neq x_j \\ 0 & \text{otherwise} \ , \end{cases}$$

The Ising Model

1. The Ising model is the simplest model of two-label MRF.
2. Used in Physics to model magnetism.
3. Edge weight $E_{ij}(x_i, x_j) = k$ if $x_i = x_j$, and 0 otherwise. Model interaction potential of two neigboring particles.
4. Vertex weights are interpreted as action of external magnetic field.
5. 2D Zero field case solved by Lars Onsager (Nobel prize for Chemistry 1968).
6. Unsolved in closed form in the 3D case – intractible.

Ising

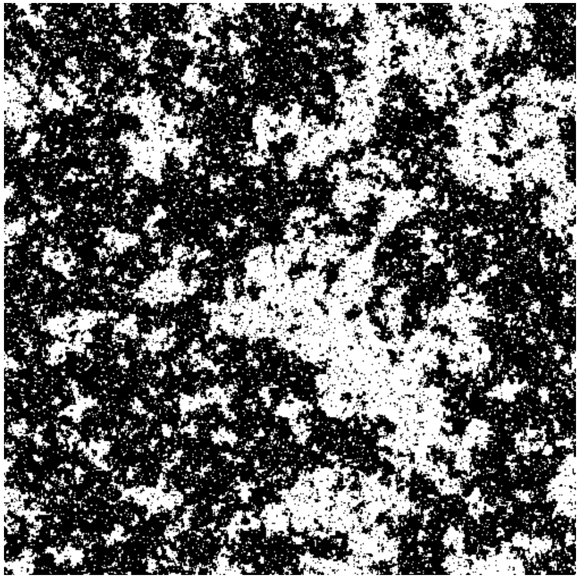


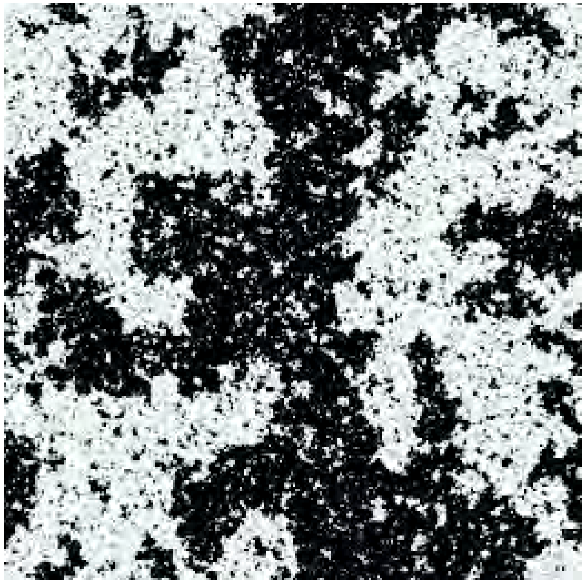


Onsager

Partition Function for the 2D Ising Model

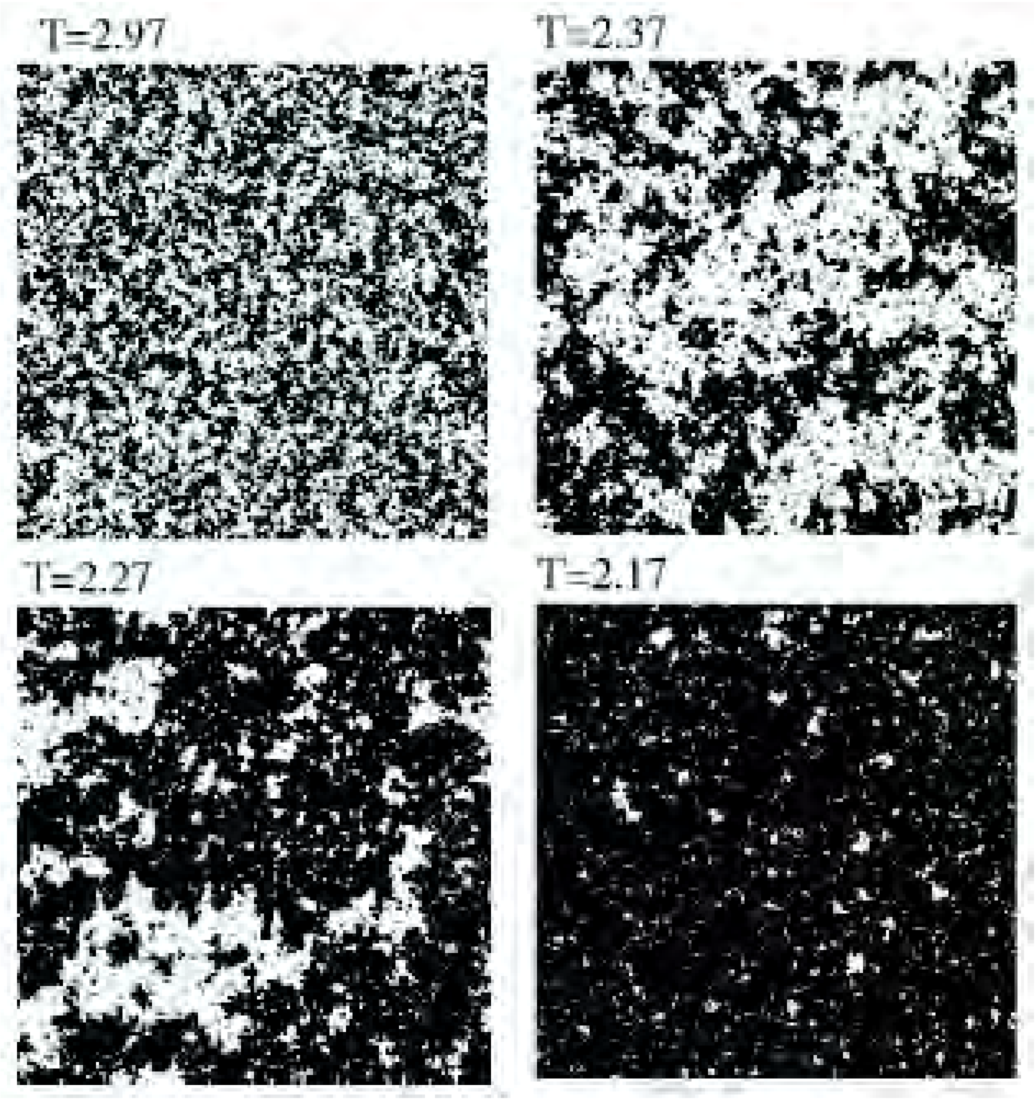
- Recall
$$P(\mathbf{x}) = \frac{1}{Z}\, e^{-E(\mathbf{x})}$$
- Z is called the partition function.
- The zero-field solution partition function is given by
$$\frac{-\log Z}{N} = -\log 2 - \frac{1}{8\pi^2}\int_0^{2\pi}\int_0^{2\pi} \log\left[\cosh^2 2k - \sinh 2k(\cos\eta + \cos\zeta)\right] d\eta d\zeta$$
where k is the edge weight, N the number of nodes.
- **Critical temperature ($T_C = 2.269$).** There is a "critical" value of the weight
$$k_C = \log(1+\sqrt{2})/2 = 0.440686$$
- For smaller values of k, a sample is expected to have equal numbers of 1 and 0.
- For larger values of k, samples are expected to be unbalanced (expected non-zero magnetization).





Samples of zero-field Ising model.

Is this a good model for natural images?



Binary (pseudo-boolean) functions

Definition of pseudo-boolean function.

Define $\mathcal{B} = \{0, 1\}$. A pseudo-boolean function is a mapping

$$f : \mathcal{B}^n \rightarrow \mathbb{R} \ .$$

Variables: x_i. The set of variables will be denoted $X = \{x_i; i = 1, \ldots, n\}$.

Literals: Literals are $x_i, \bar{x}_i$, where $\bar{x}_i = 1 - x_i$. Use u to represent a literal. The set of literals is denoted by $L = \{x_i, \bar{x}_i; i = 1, \ldots, n\}$.

Graph-representable functions

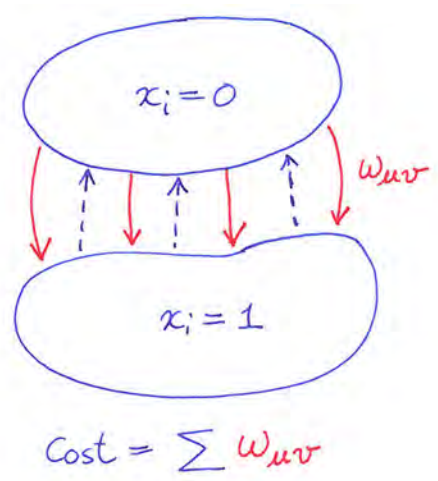
The **cost** of a partition $(\mathcal{V}_0, \mathcal{V}_1)$ is the sum of weights of all edges going from $\mathcal{V}_0$ to $\mathcal{V}_1$. Formally,

$$\text{Cost}(\mathcal{V}_0, \mathcal{V}_1) = \sum_{u \in \mathcal{V}_0, v \in \mathcal{V}_1} w(u, v)$$

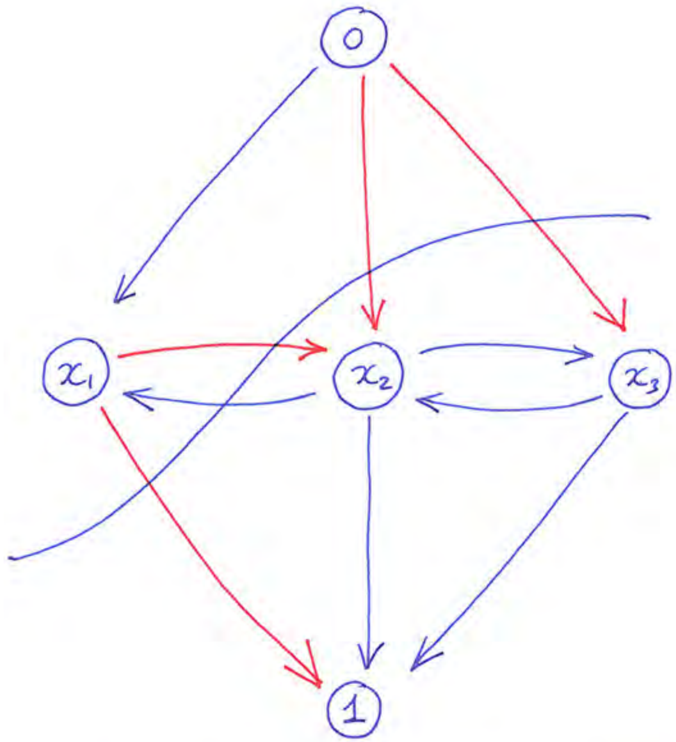
where $w(u,v)$ is the weight of the edge from vertex u to v.

Definition The function $f(\mathbf{x})$ is *simple graph representable* if there is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{W})$, such that for all x,

$$f(\mathbf{x}) = \text{Cost}(\mathcal{V}_0(\mathbf{x}), \mathcal{V}_1(\mathbf{x})) \ .$$



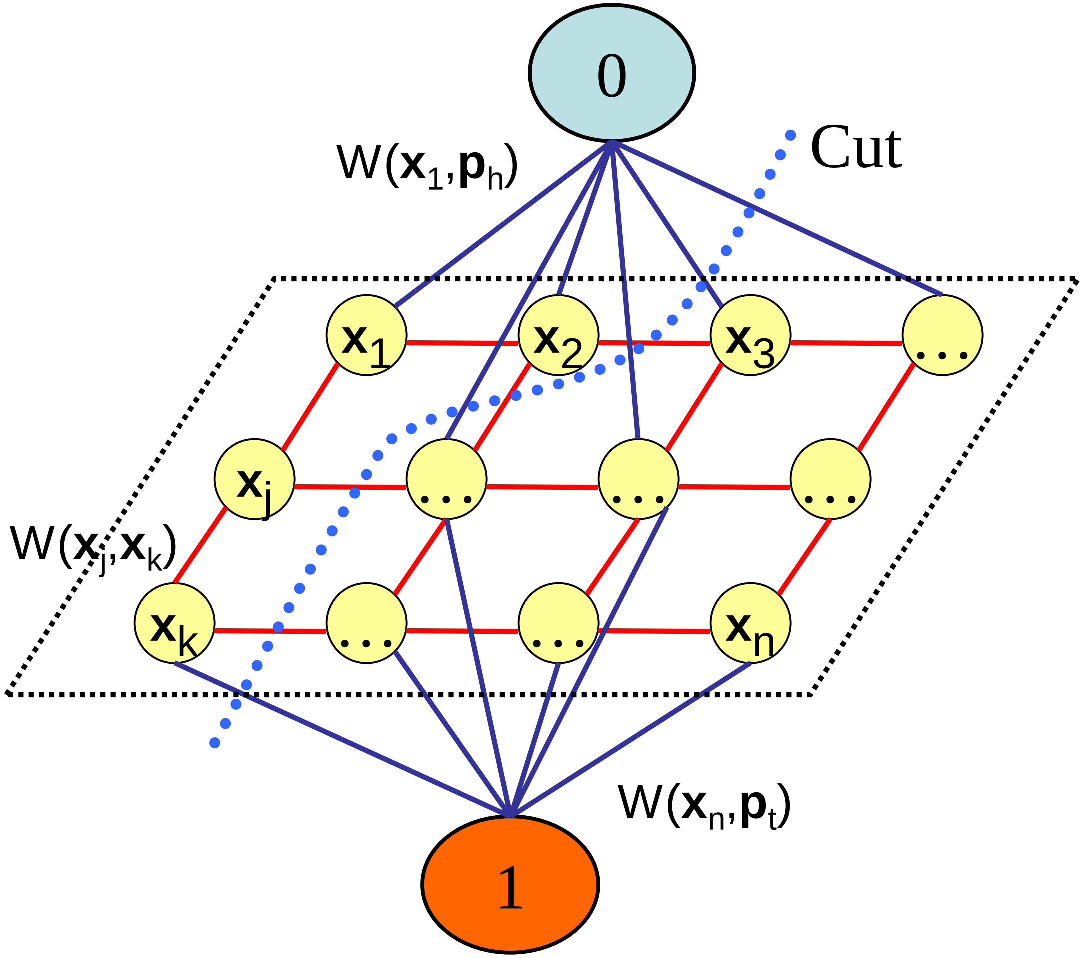
Cost of a partition is the sum of weights of edges passing from $\mathcal{V}_0$ to $\mathcal{V}_1$.



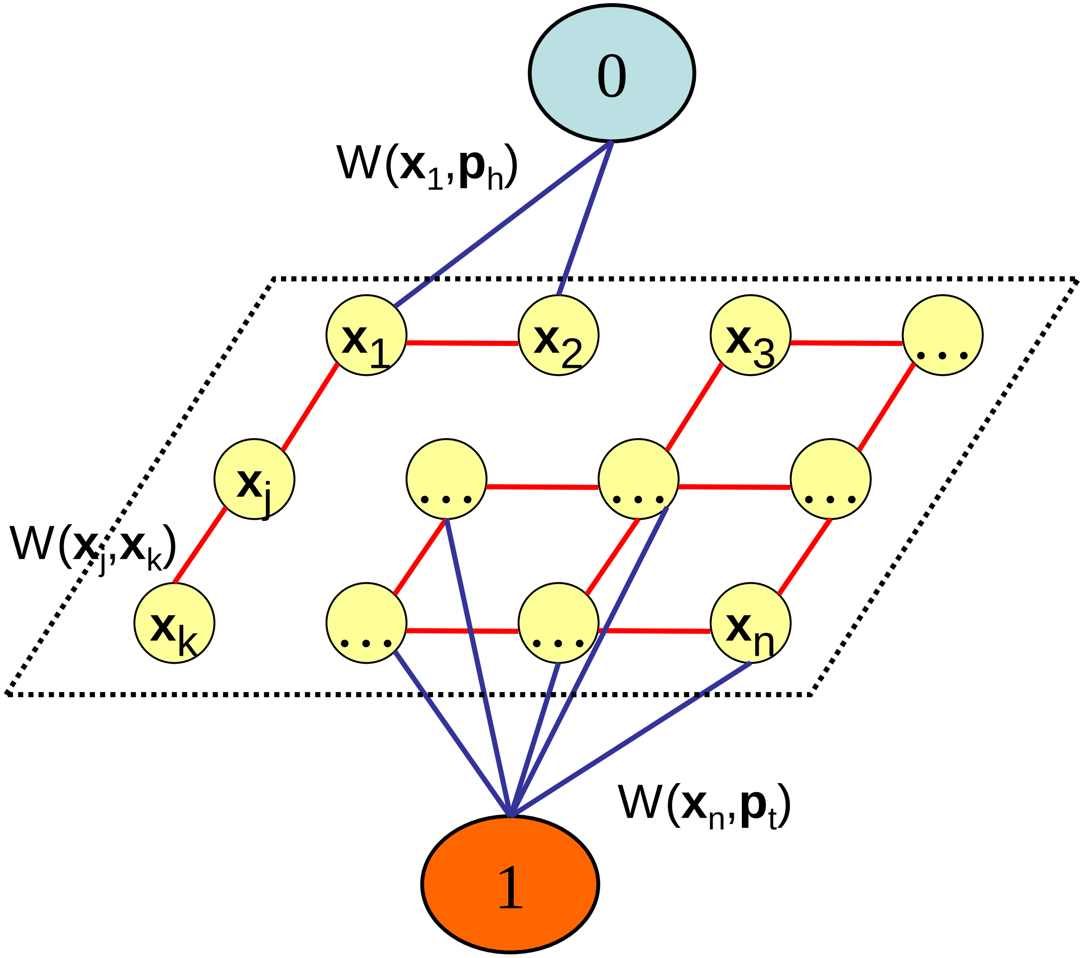
$$\mathrm{Cost}\left(x_1 = 0,\ x_2 = x_3 = 1\right) = \sum \text{red edges}.$$

Graph Cuts

Consider the case of two segments.



Graph Cuts

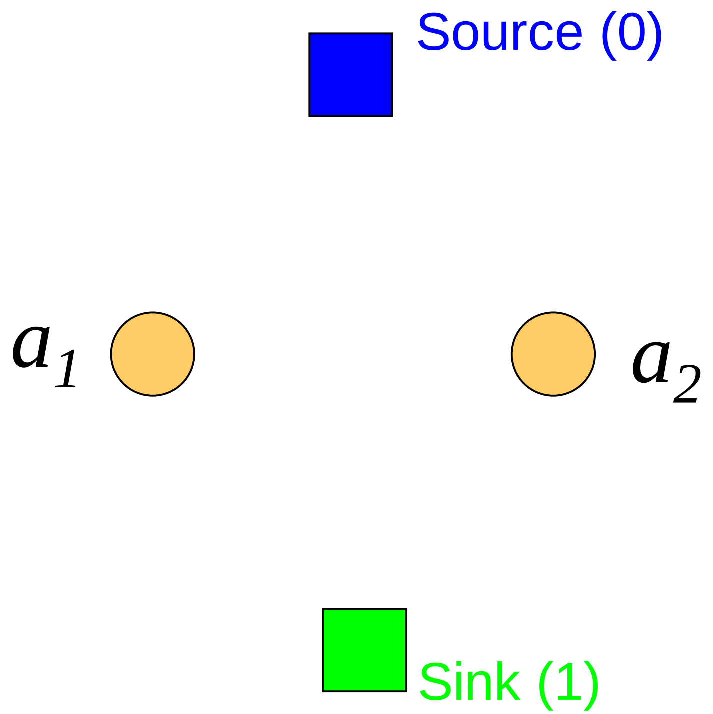


0
W($\mathbf{x}_1$,$\mathbf{p}_h$)
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
…
$\mathbf{x}_j$
…
…
…
W($\mathbf{x}_j$,$\mathbf{x}_k$)
$\mathbf{x}_k$
…
…
$\mathbf{x}_n$
W($\mathbf{x}_n$,$\mathbf{p}_t$)
1

Energy Minimization using Graph cuts

What really happens? Building the graph

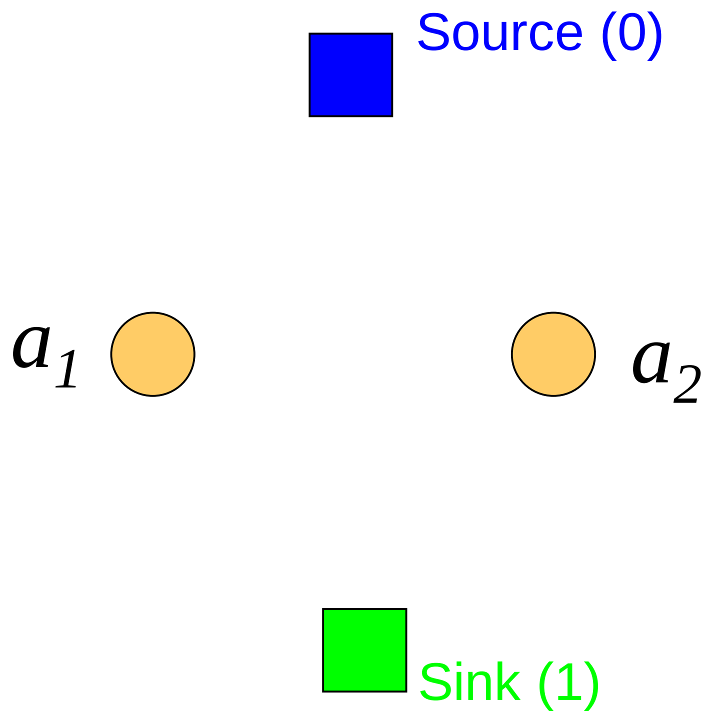
$E_{MRF}(a_1,a_2)$



Energy Minimization using Graph cuts

What really happens? Building the graph

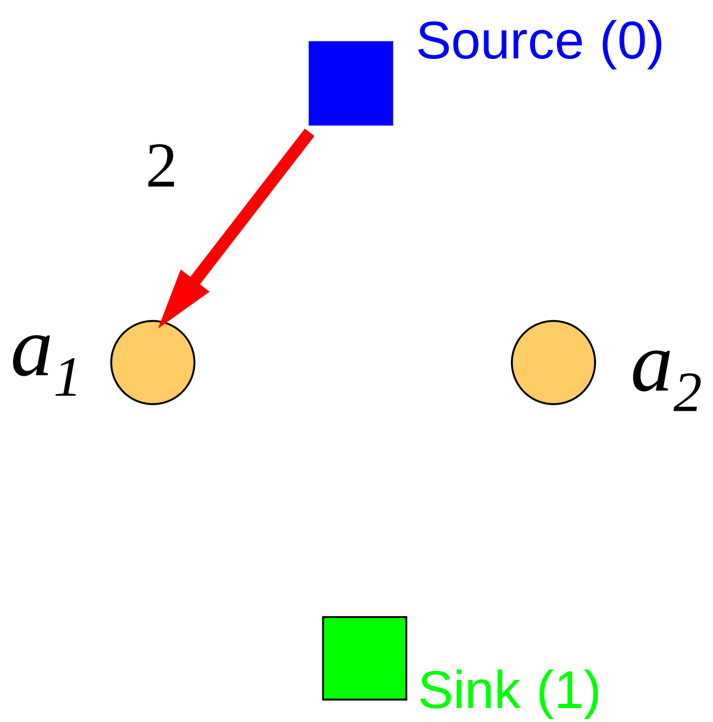
If we cut a link from 0 to 1, incur the cost of that edge



Energy Minimization using Graph cuts

What really happens? Building the graph

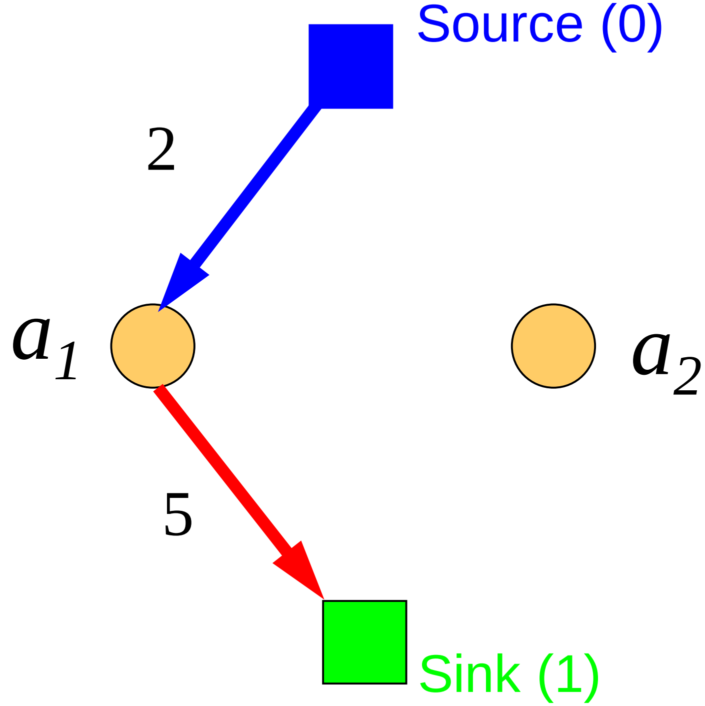
$E_{\text{MRF}}(a_1,a_2) = 2a_1$



Energy Minimization using Graph cuts

What really happens? Building the graph

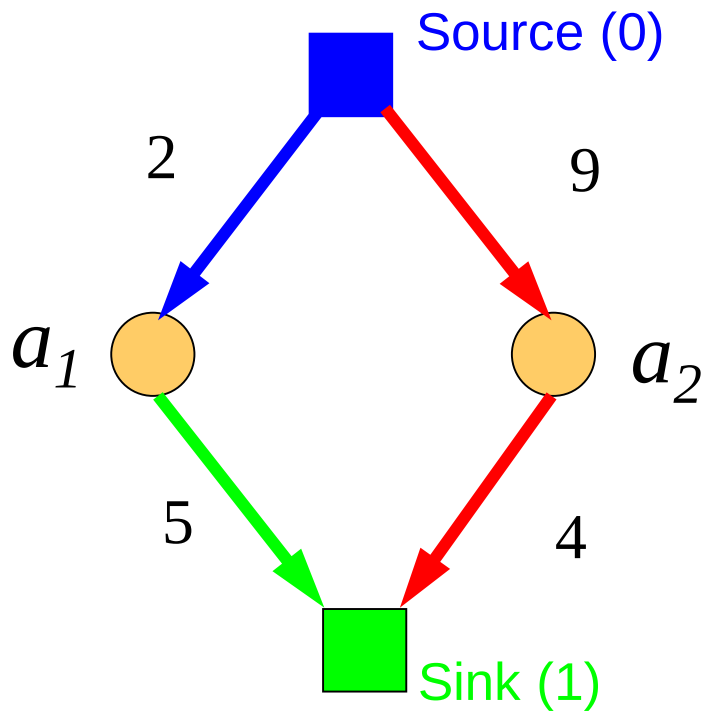
$$E_{\text{MRF}}(a_1,a_2) = 2a_1 + 5\bar{a}_1$$



Energy Minimization using Graph cuts

What really happens? Building the graph

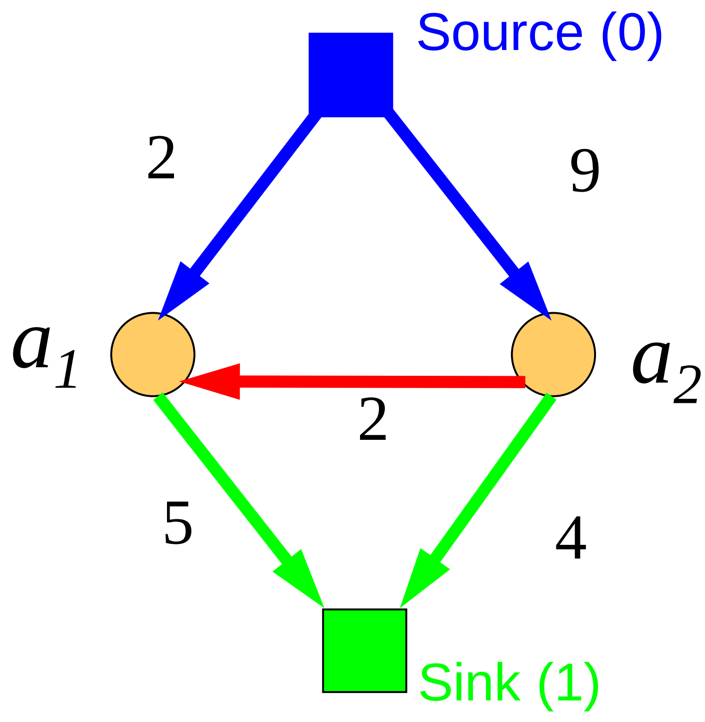
$$E_{\mathrm{MRF}}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2$$



Energy Minimization using Graph cuts

What really happens? Building the graph

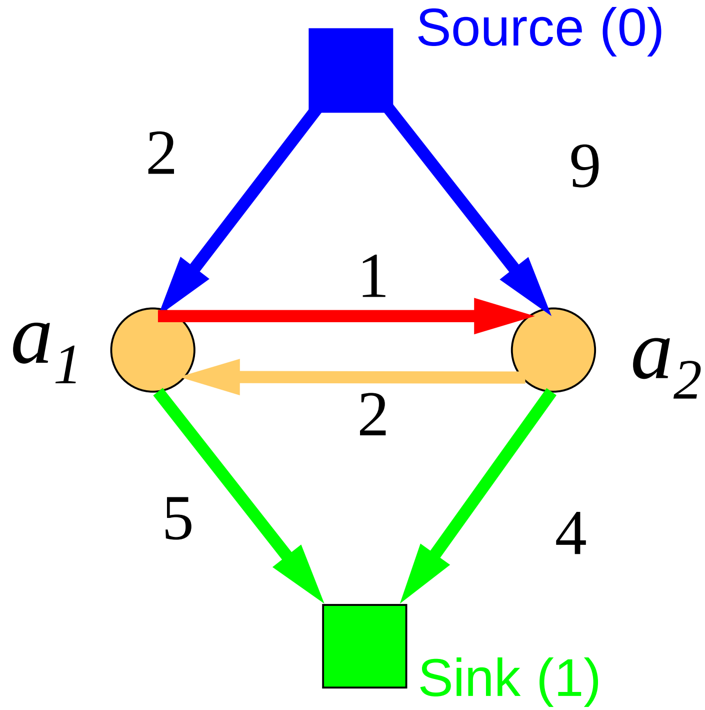
$$E_{\text{MRF}}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2$$



Energy Minimization using Graph cuts

What really happens? Building the graph

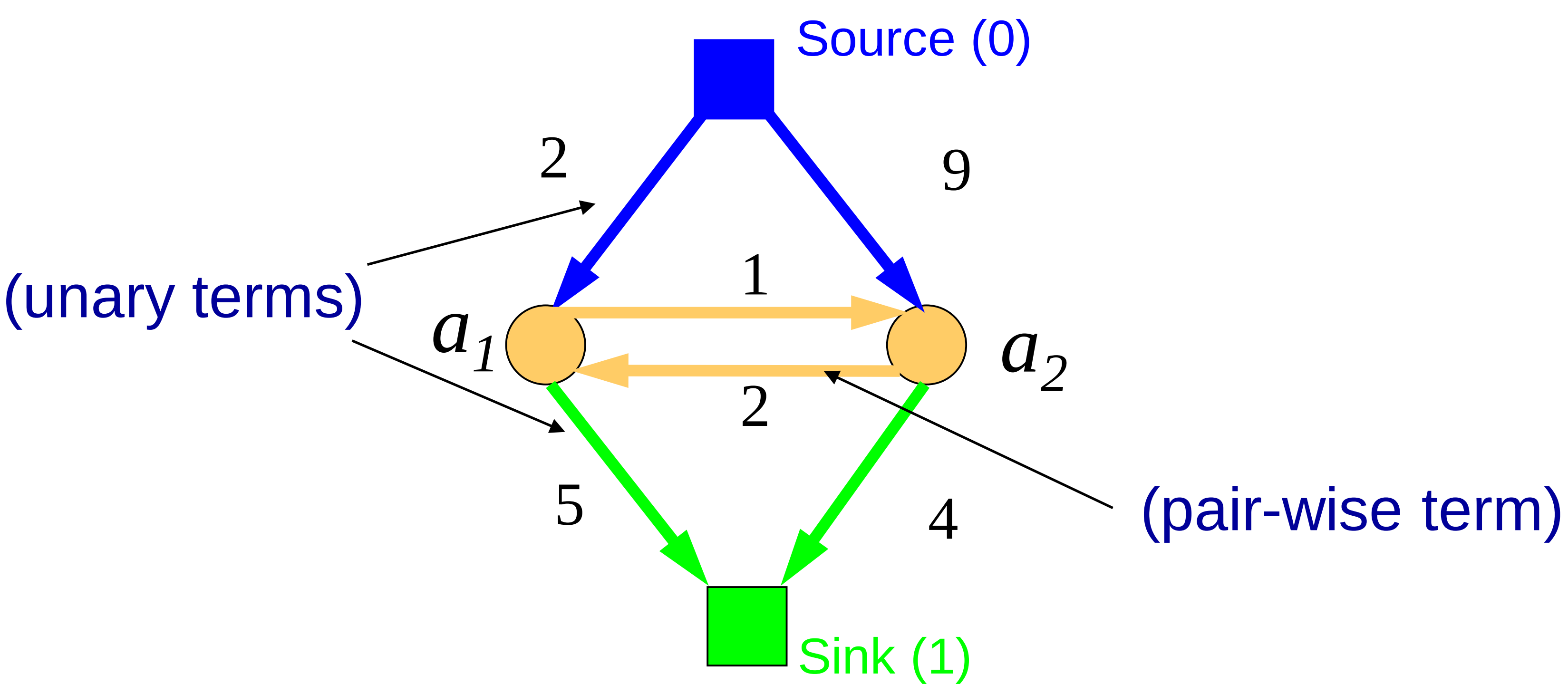
$$E_{MRF}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



Energy Minimization using Graph cuts

What really happens? Building the graph

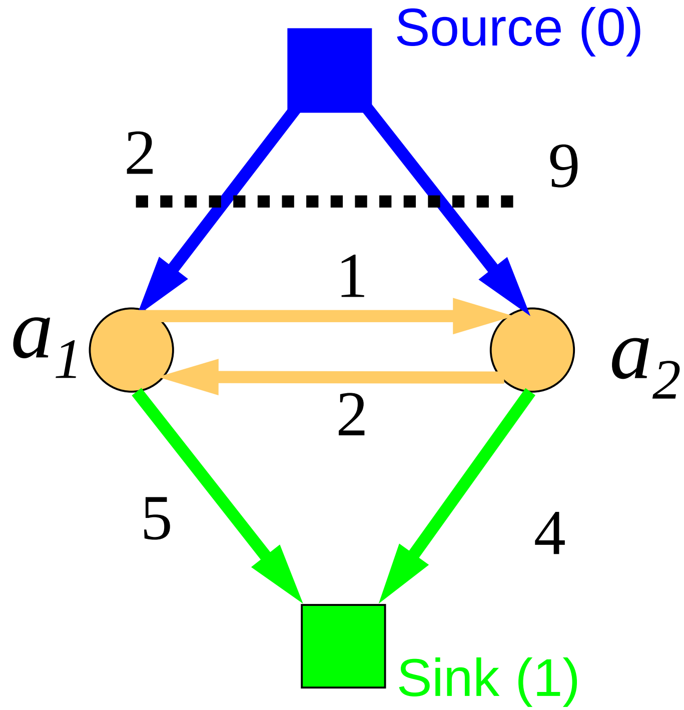
$$E_{\text{MRF}}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



Energy Minimization using Graph cuts

What really happens? Building the graph

$$E_{MRF}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



$a_1 = 1 \quad a_2 = 1$

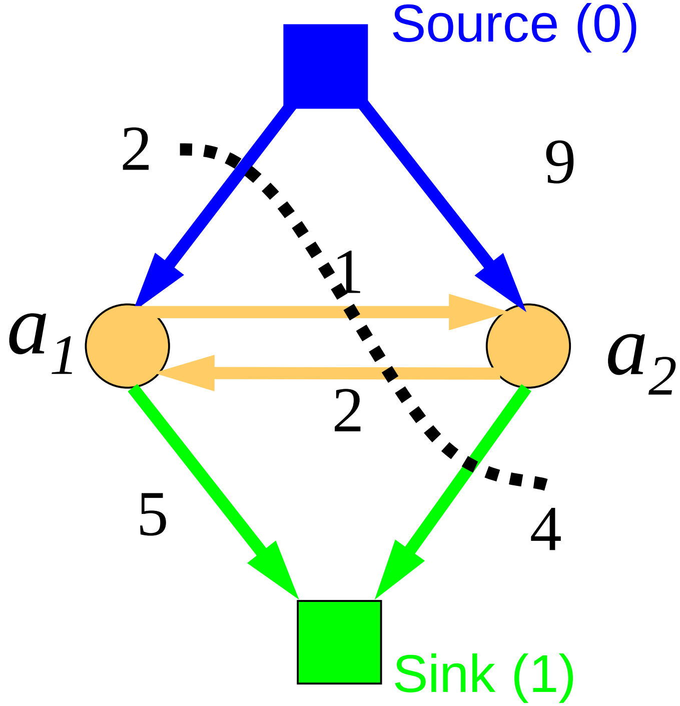
Cost of st-cut = 11

$E_{MRF}(1,1) = 11$

Energy Minimization using Graph cuts

What really happens? Building the graph

$$E_{MRF}(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



$a_1 = 1 \quad a_2 = 0$

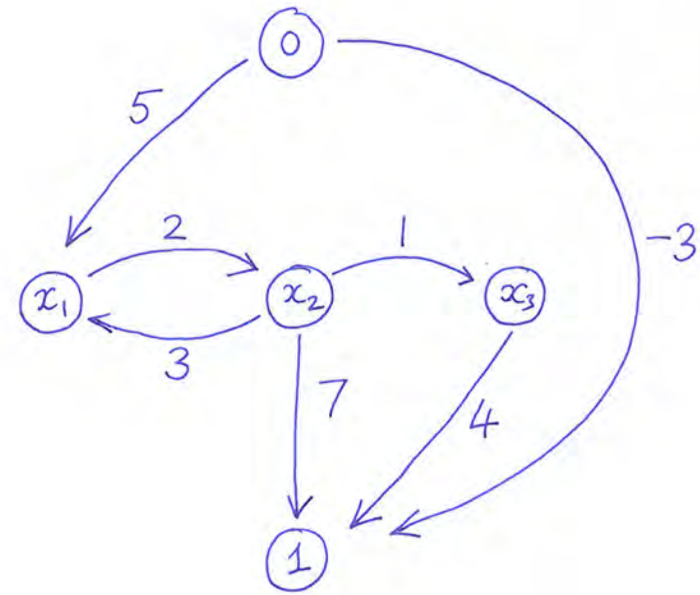
Cost of st-cut = 8

$E_{MRF}(1,0) = 8$

Notes:

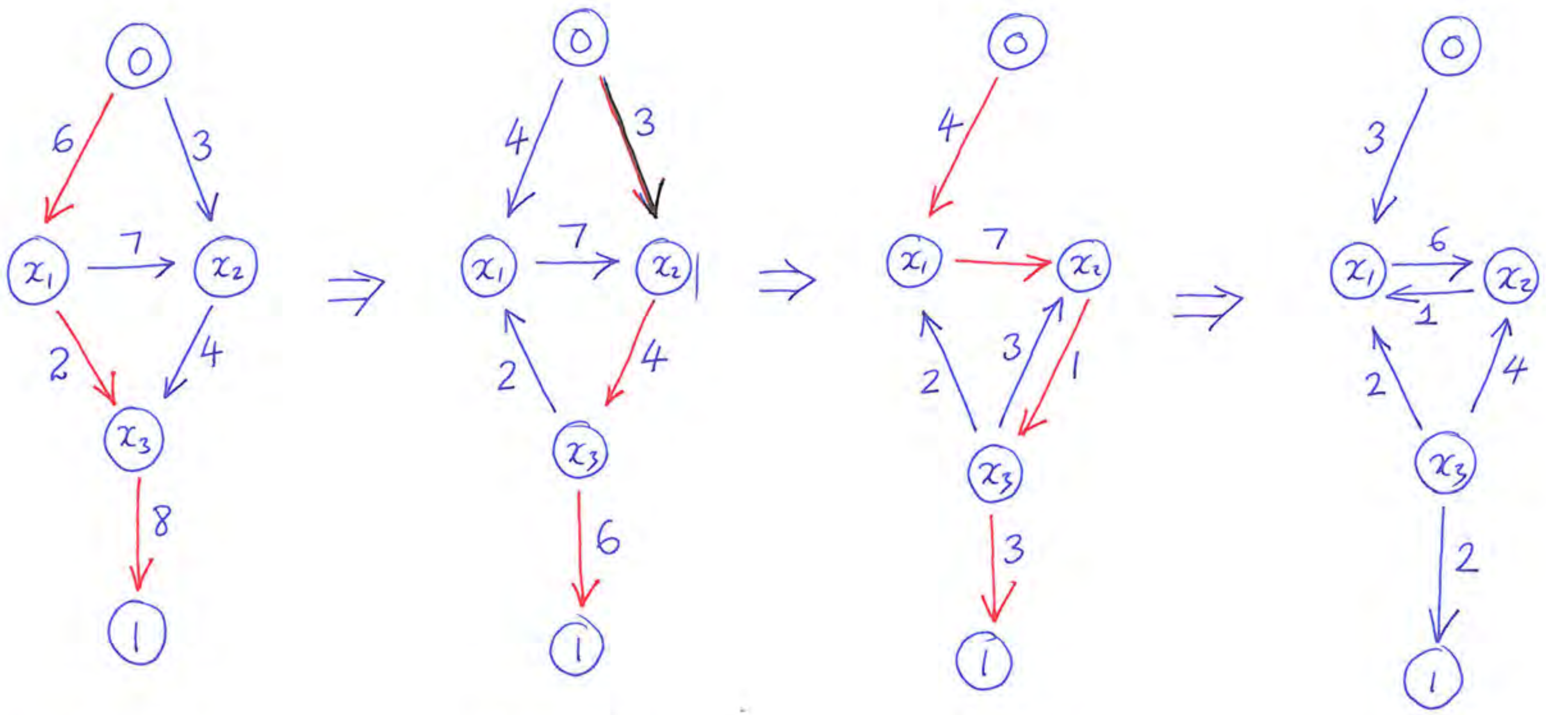
1. An edge from vertex u to vertex v with weight a represents the term $a\bar{u}v$, where $\bar{0} = 1$.
2. It is unnecessary to have both terms x_i and $\bar{x}_i$ occuring. Furthermore, non-zero linear terms can be of the form $a_i x_i$ with $a_i > 0$, or $a_{\bar{i}}\bar{x}_i$ with $a_{\bar{i}} > 0$.
3. It is unnecessary to have both $\bar{x}_i x_j$ and $\bar{x}_j x_i$ occuring. Thus, the function can be written as

$$f(\mathbf{x}) = L + \sum_{1 \le i < j \le n} a_{\bar{i}j}\bar{x}_i x_j \ .$$



$$-3 + 5x_1 + 7\bar{x}_2 + 4\bar{x}_3$$
$$+ 2\bar{x}_1 x_2 + 3\bar{x}_2 x_1 + \bar{x}_2 x_3$$

Max flow algorithm



Cost = 2   Cost = 3   Cost = 1 $x_1 = x_2 = 0$

$x_3 = 1$

Total Cost = 6

Summary.

1. All quadratic pseudo-boolean functions $f(\mathbf{x})$ can be expressed as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{W})$ such that the function value is equal to the cost of the corresponding partition:

$$f(\mathbf{x}) = \text{Cost}(\mathcal{V}_0(\mathbf{x}), \mathcal{V}_1(\mathbf{x}))$$

2. If all weights are positive, then the min-cut on the graph (minimum of the function) is equal to the maximal permissible flow.

3. A graph with weights w_{uv} can be reparametrized to a graph with non-negative weights if and only if

$$w(x_i, x_j) + w(x_j, x_i) \geq 0$$

for all pairs of variables x_i, x_j.

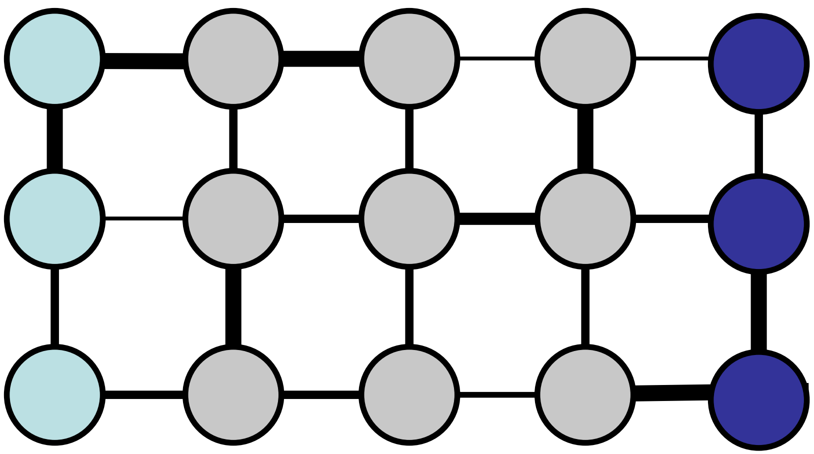
4. If all the weights are non-negative, then

$$\text{mincut} = \min_{\mathbf{x}} f(\mathbf{x}) = \text{maxflow}$$

and the minimization problem can be solved using a max-flow algorithm in polynomial time.

Graph Cuts Basics
(simple 2D example)

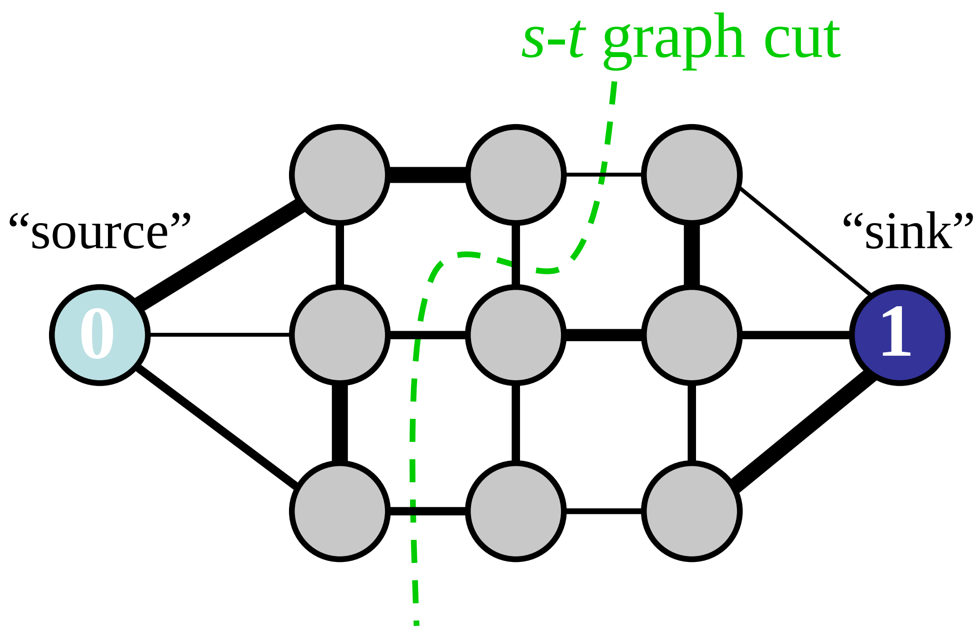
Goal: divide the graph into two parts separating red and blue nodes



Red/blue nodes can be "identified" into two super nodes (terminals)

Graph Cuts Basics
(simple 2D example)

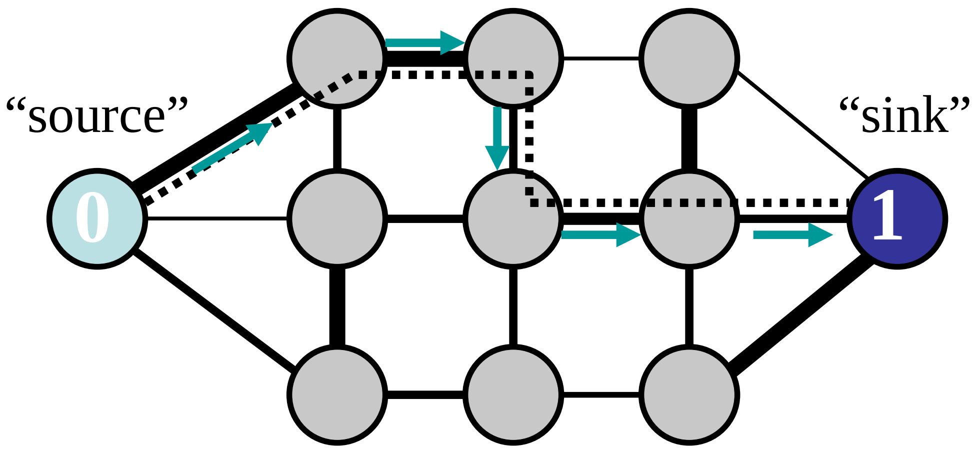
Goal: divide the graph into two parts separating red and blue nodes



A graph with two terminals *0* and *1*

- Cut cost is a sum of severed edge weights from 0 to 1
- **Minimum cost *s-t* cut can be found in polynomial time**

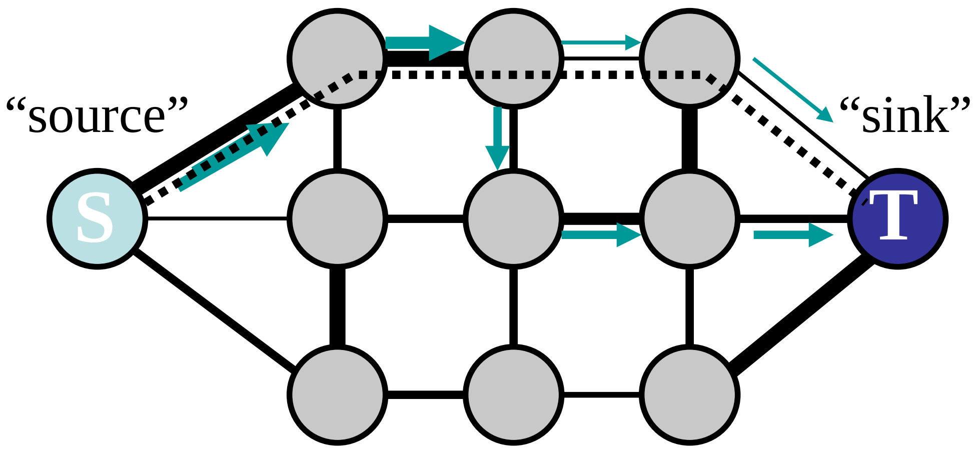
"Augmenting Paths"



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

"Augmenting Paths"



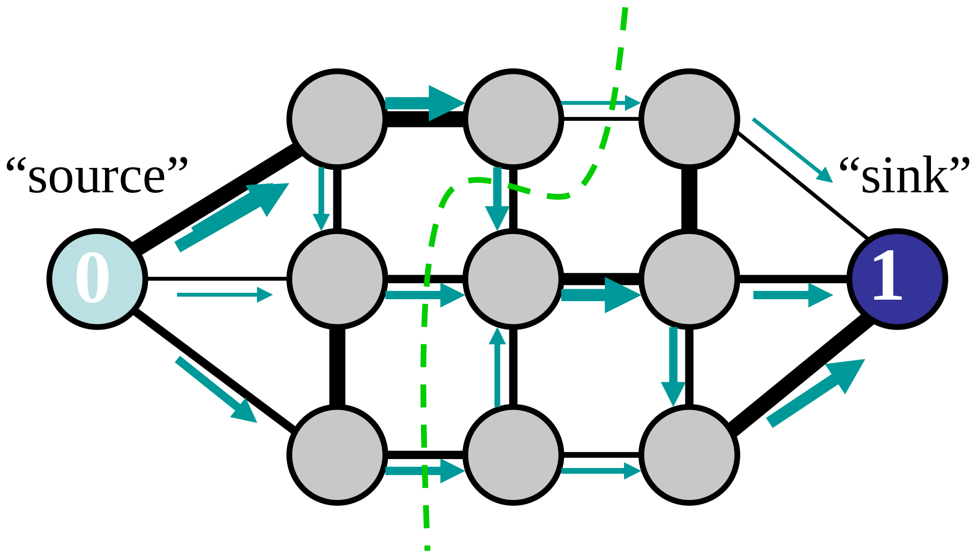
A graph with two terminals

- Find a path from 0 to 1 along non-saturated edges

- Increase flow along this path until some edge saturates

- Find next path…
- Increase flow…

"Augmenting Paths"



A graph with two terminals

MAX FLOW = MIN CUT

- Find a path from 0 to 1 along non-saturated edges
- Increase flow along this path until some edge saturates

Iterate until … all paths from 0 to 1 have at least one saturated edge

Multi-label Optimization

Multi Label Problems

- So far we have considered generic cuts or 2 label problems
- Now we consider multi label problems

s-t graph-cuts for multi-label problems

- Multi-scan-line stereo
  - Roy & Cox 1998, 1999
  - Ishikawa & Geiger 1999 (occlusion handling)

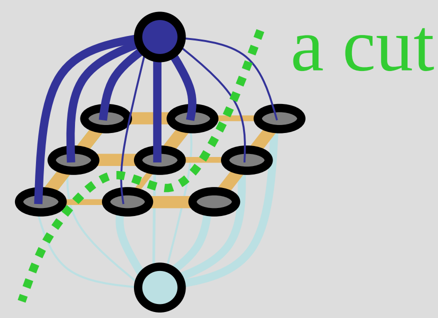
- “Linear” interaction energy
  - Ishikawa & Geiger 1998
  - BVZ 1998

- Convex interaction energy
  - Ishikawa 2000, 2003

Graph cuts and Potts energy minimization

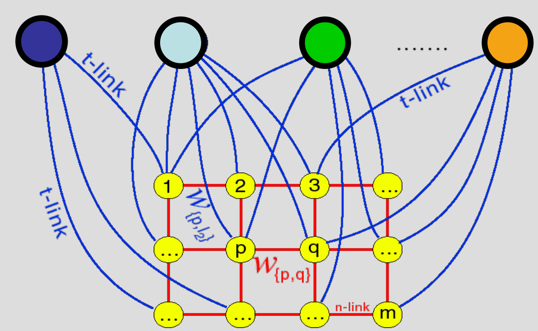
Binary Potts energy (Ising model)

_s-t_ graph cuts
(Greig at.al. 1989)

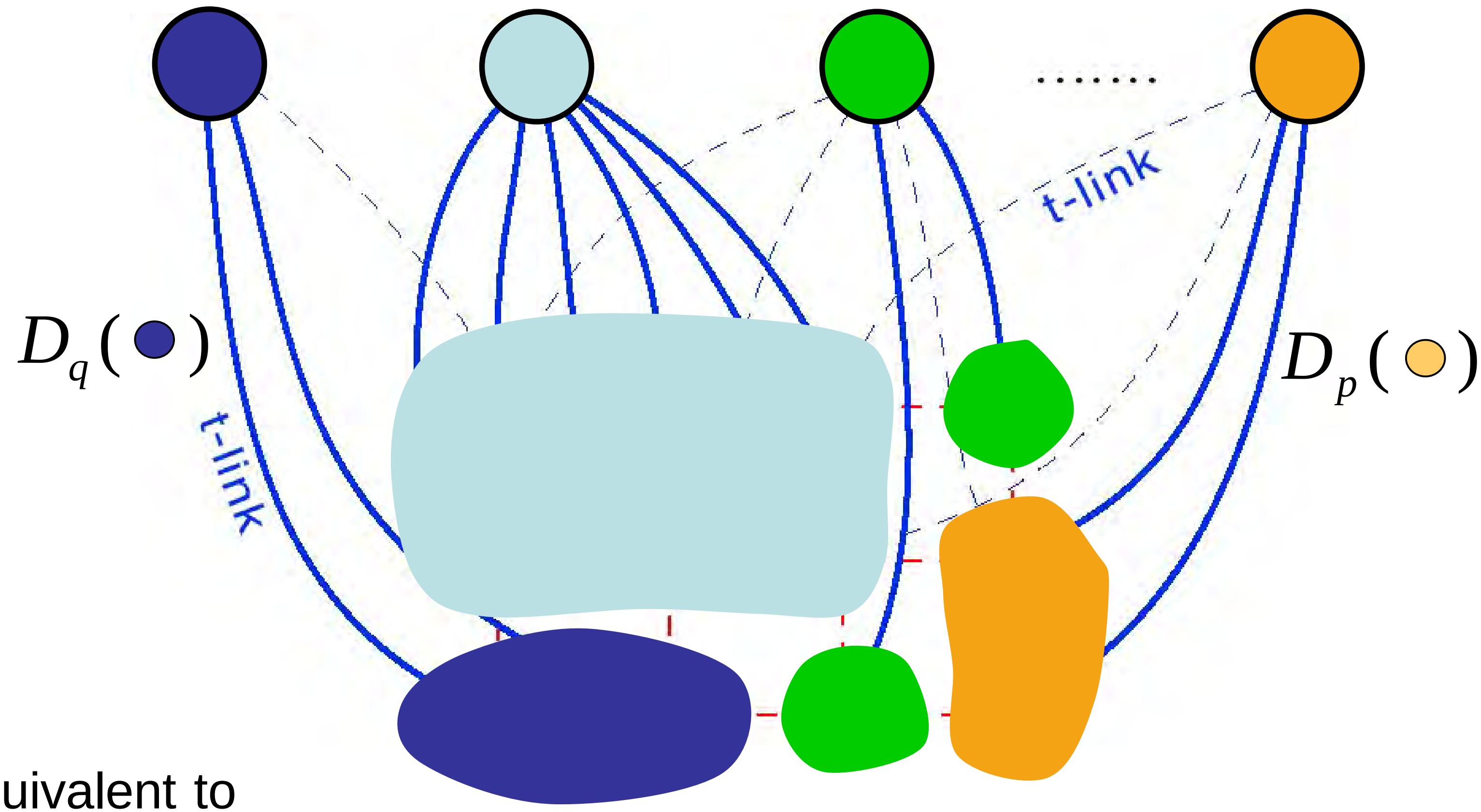


Multi-label Potts energy

Multi-way graph cuts
(BVZ 1998,2001)



Multi-way graph cuts

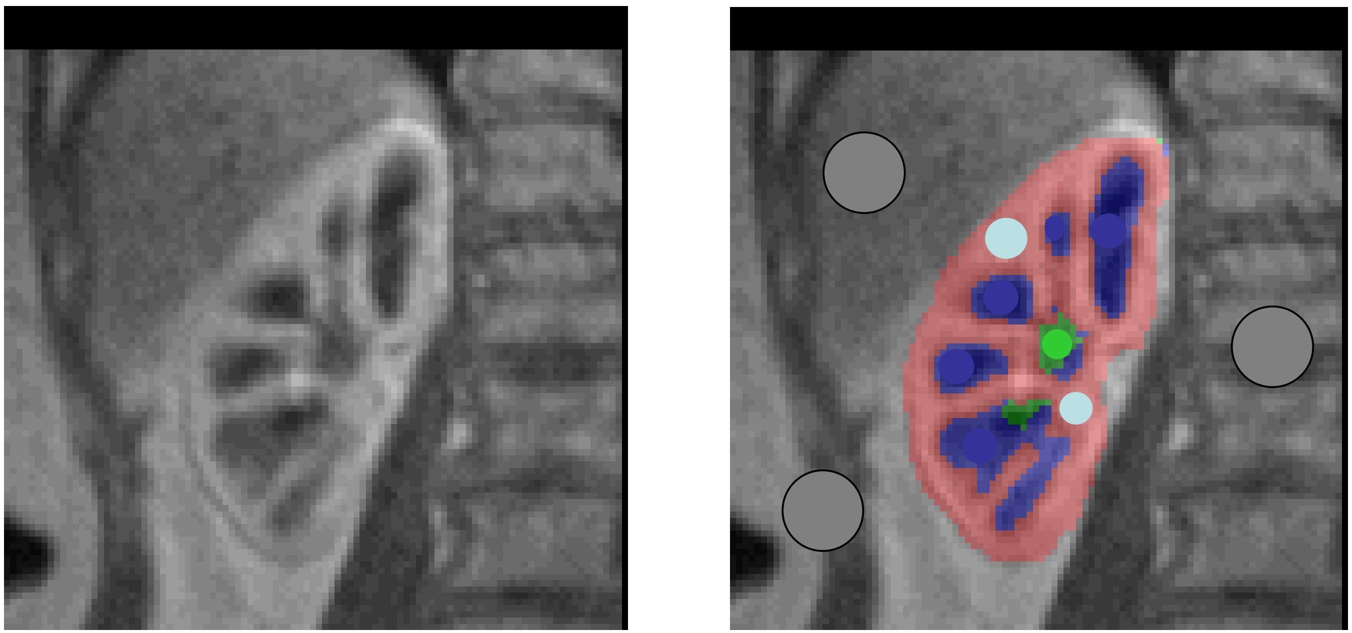


Equivalent to minimization of **the Potts energy** of labeling *L*

$$E(L) = \underbrace{\sum_{p} -D_p(L_p)}^{\text{t-links}} + \underbrace{\sum_{pq\in N} w_{pq}\cdot\delta_{Lp\neq Lq}}^{\text{n-links}}$$

Multi-way graph cuts

Multi-object Extraction



Obvious generalization of binary object extraction technique
(Boykov, Jolly, Funkalea 2004)

Multi-way graph cuts

stereo vision

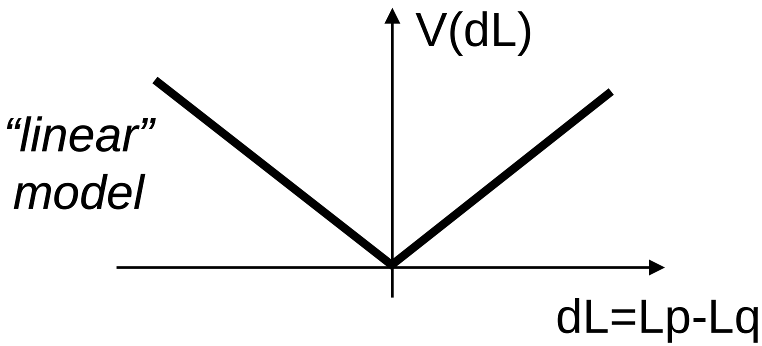
original pair of “stereo” images

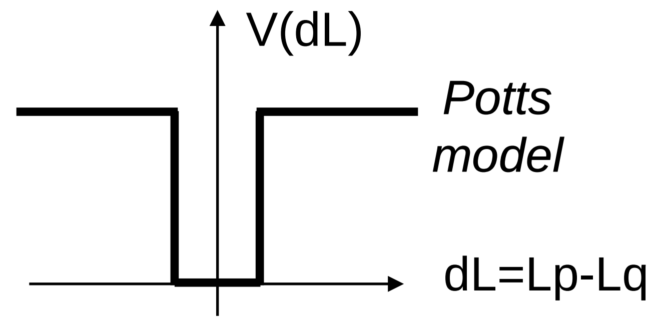
depth map

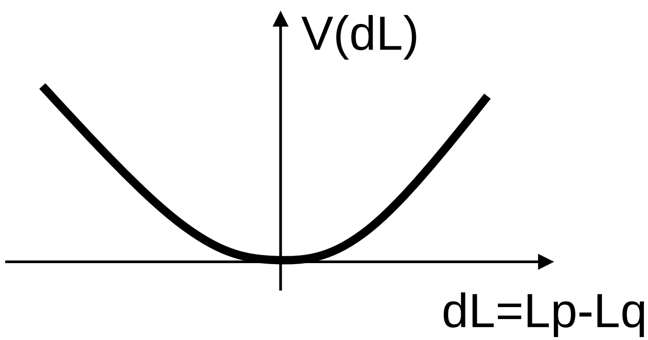
Common edge functions

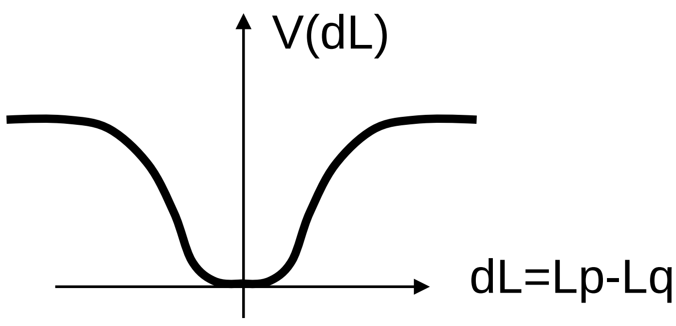
Pixel interactions:
“*convex*” vs. “*discontinuity-preserving*”

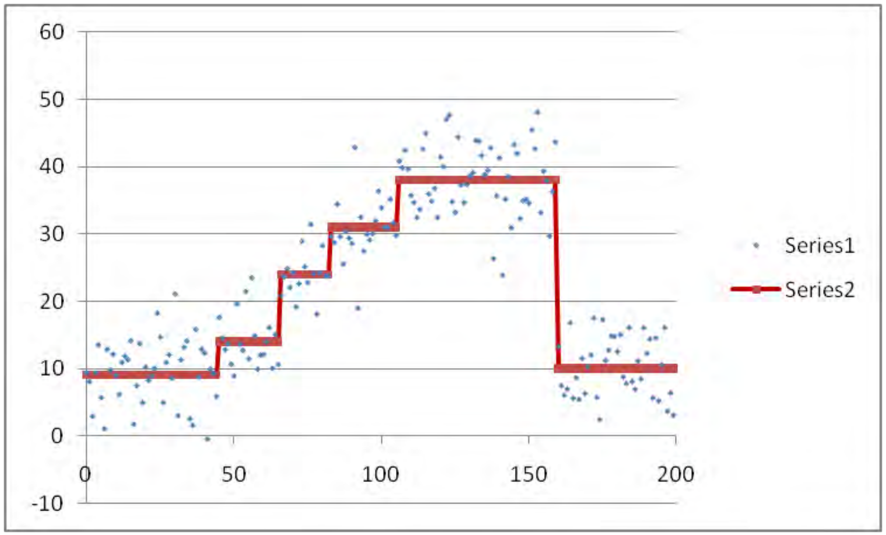
“Convex” interactions	Robust “discontinuity preserving” interactions



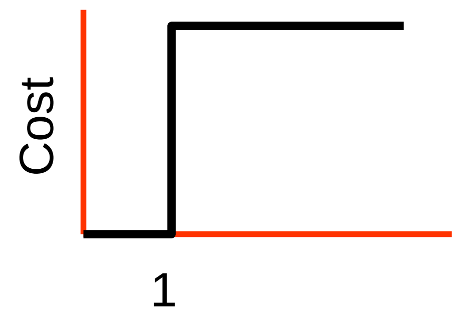


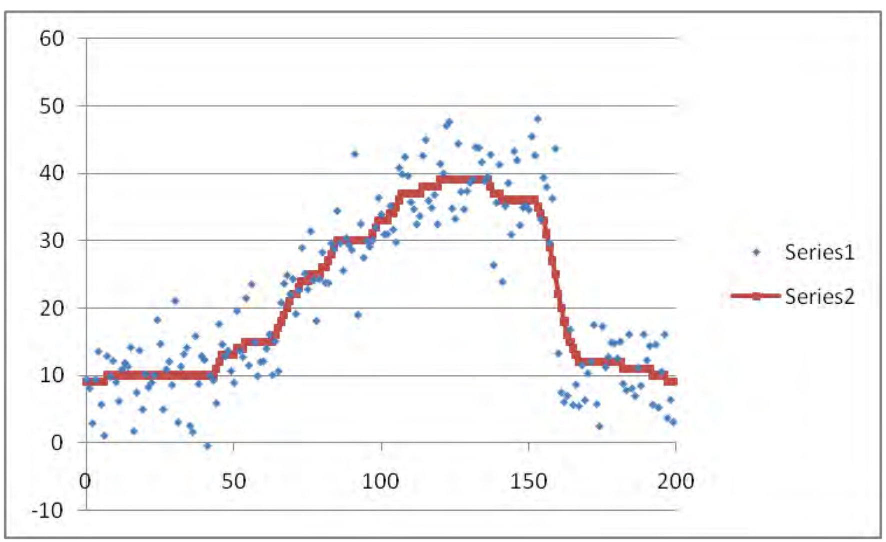




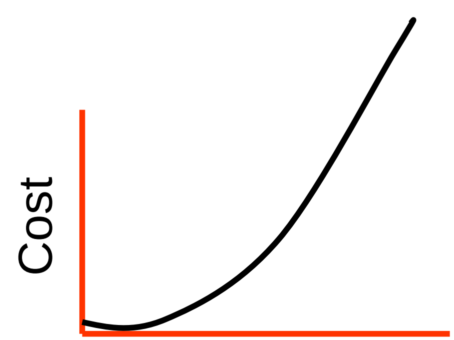


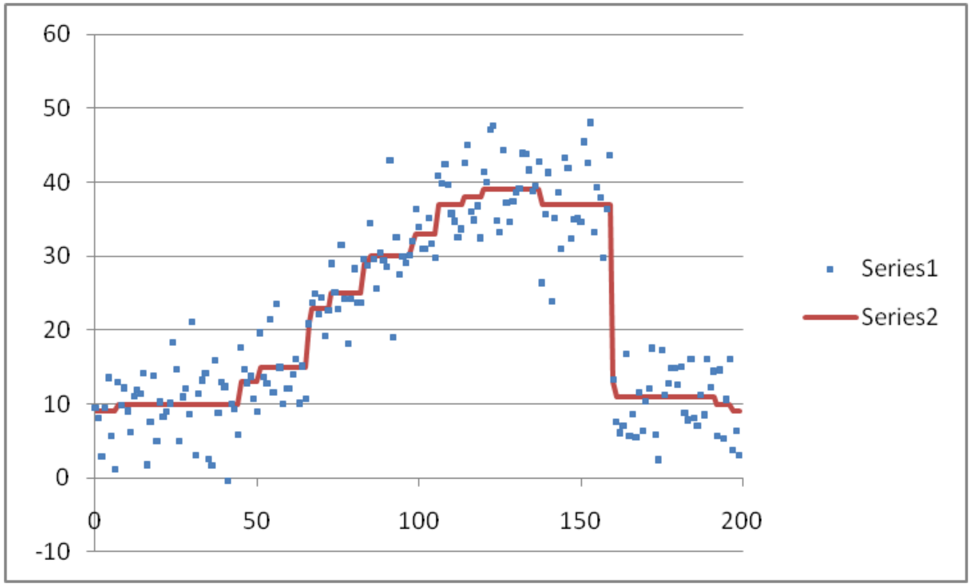
Potts Model



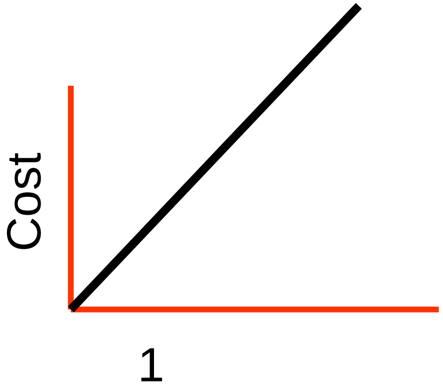


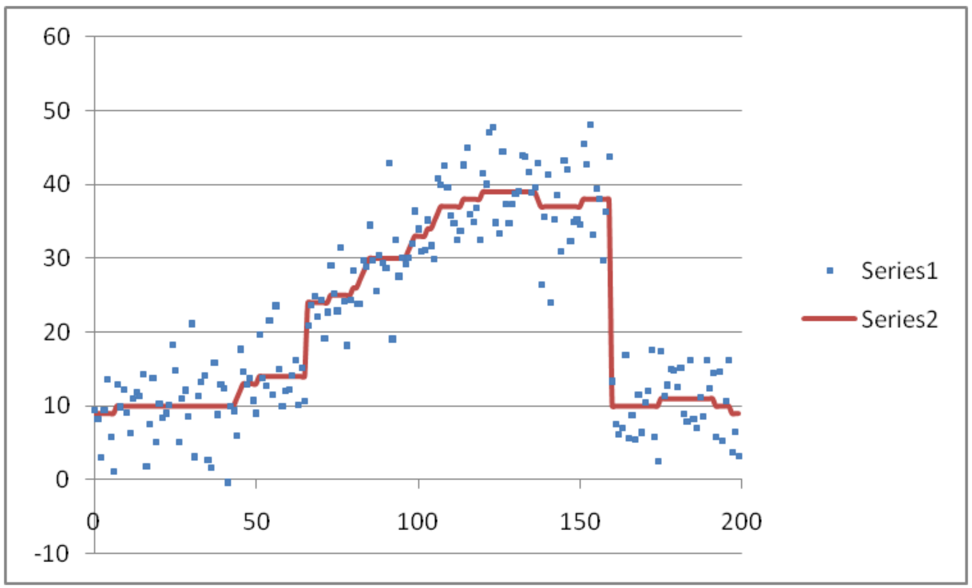
Quadratic cost



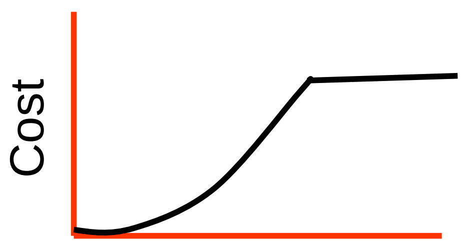


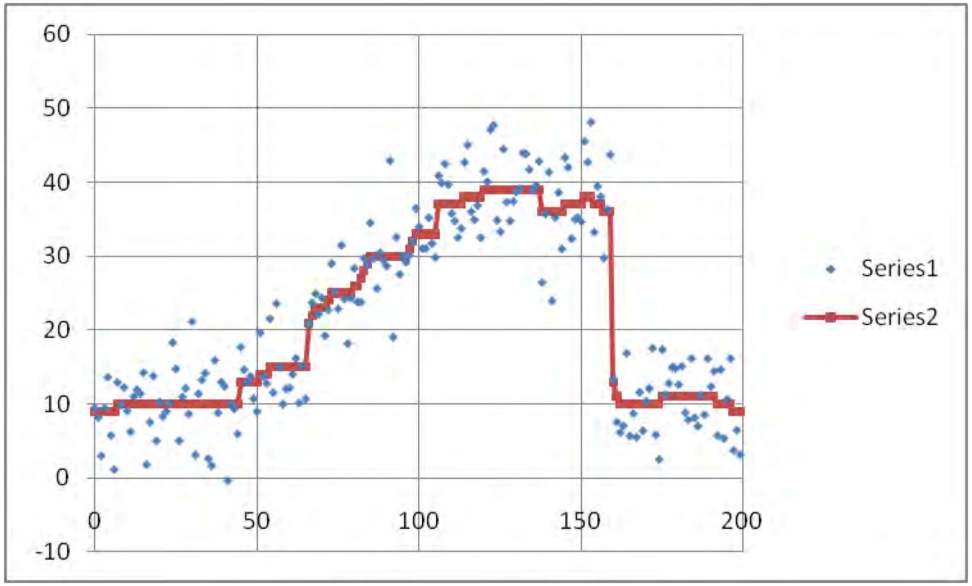
Linear

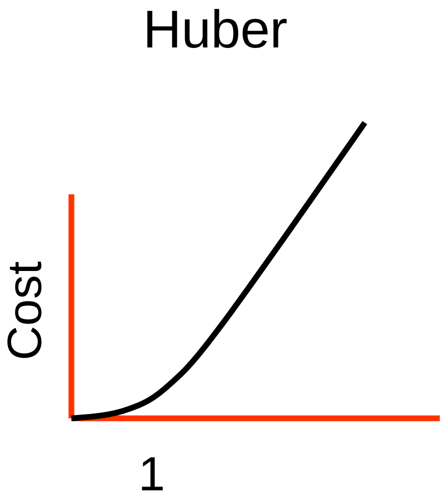


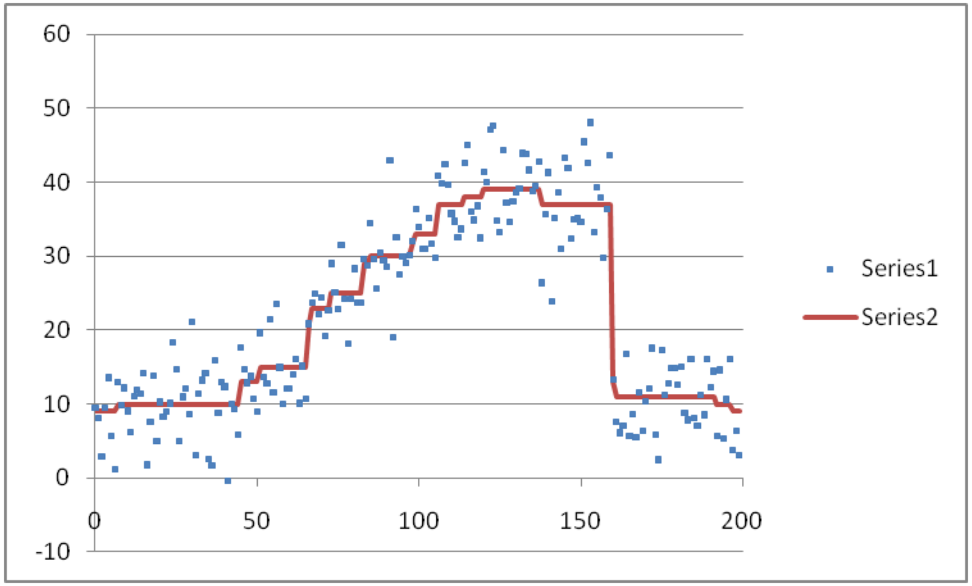


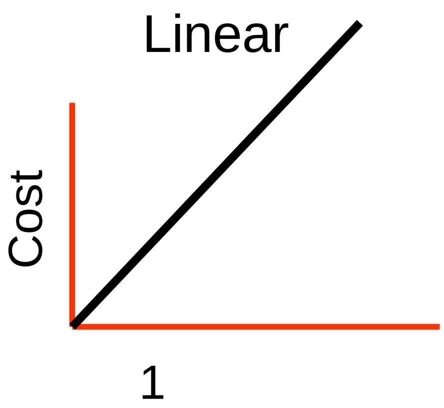
Truncated quadratic

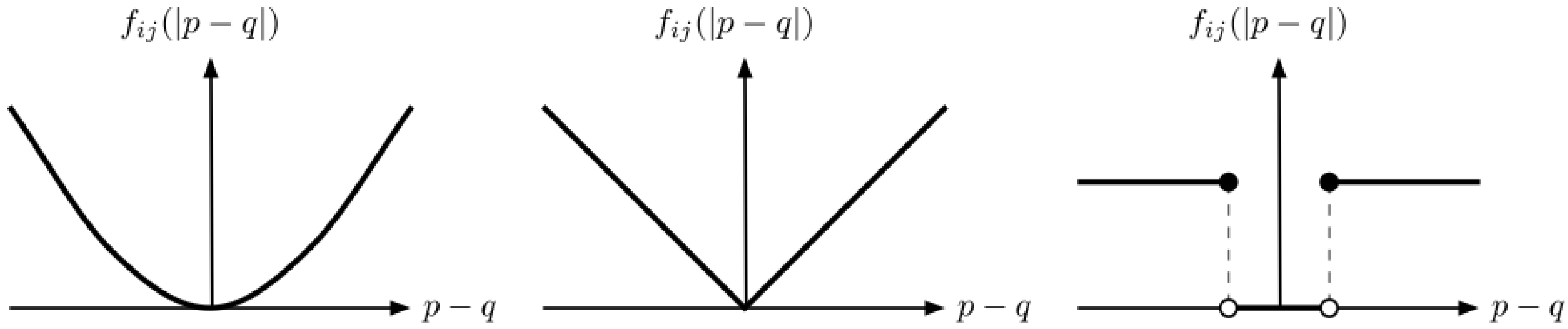



60
50
40
30
20
10
0
-10
0
50
100
150
200
Series1
Series2


Huber
Cost
1


60
50
40
30
20
10
0
-10
0
50
100
150
200
Series1
Series2


Linear
Cost
1



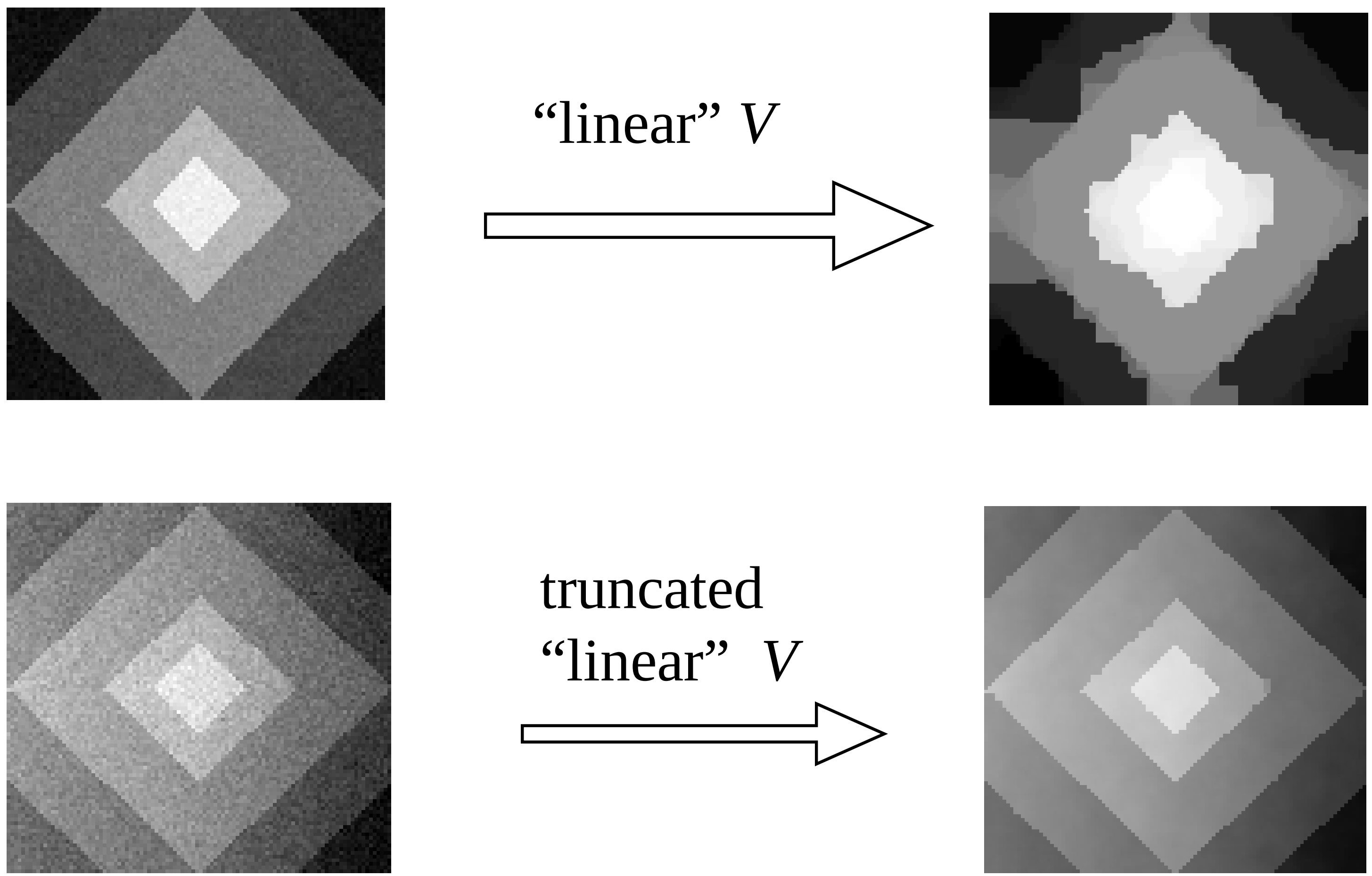
Alpha-Expansion

✓	✓	✓

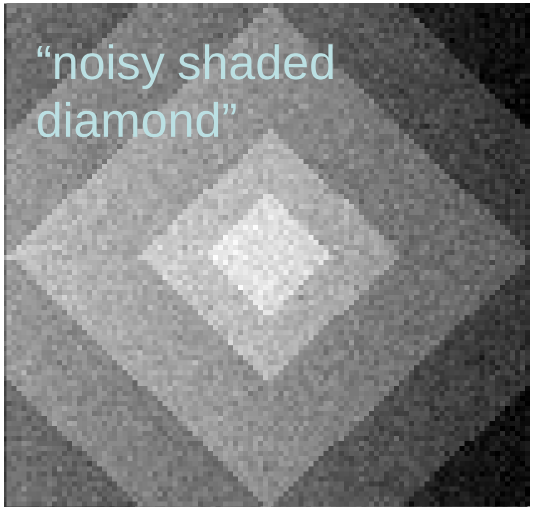
Ishikawa’s Construction

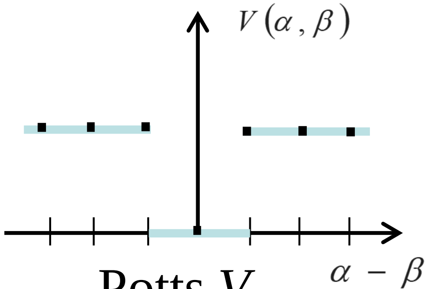
✓	✓	✗

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

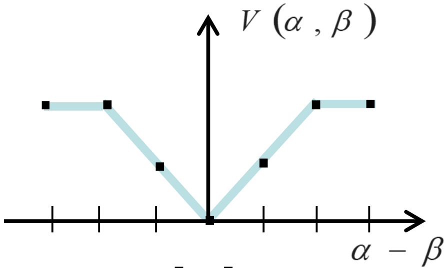


α-expansions: examples of *metric* interactions

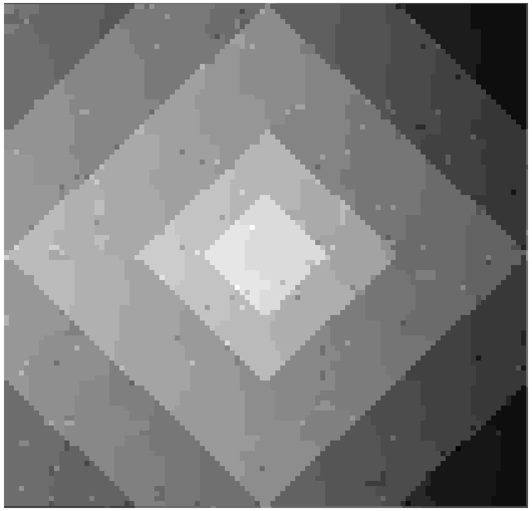


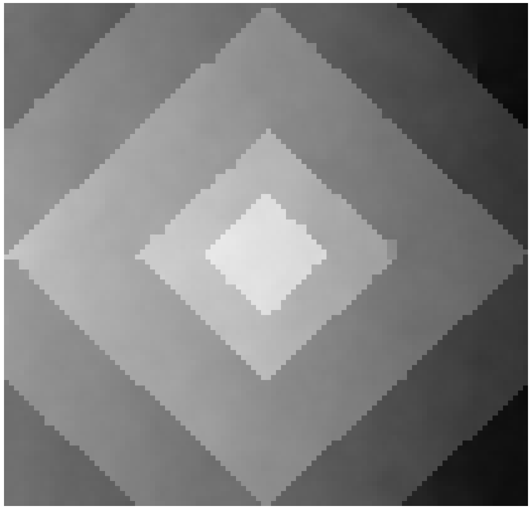


Potts V

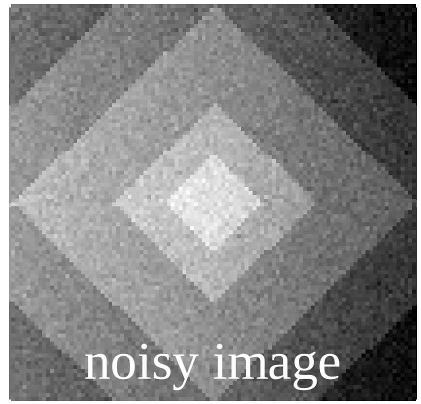


Truncated “linear” V



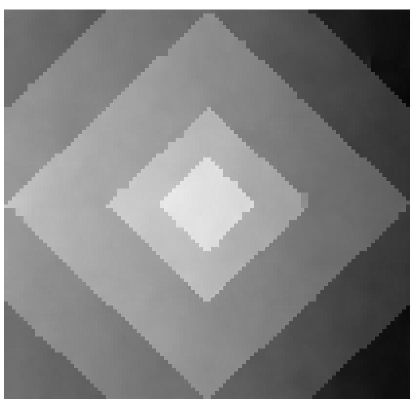


a-expansion move vs. “standard” moves

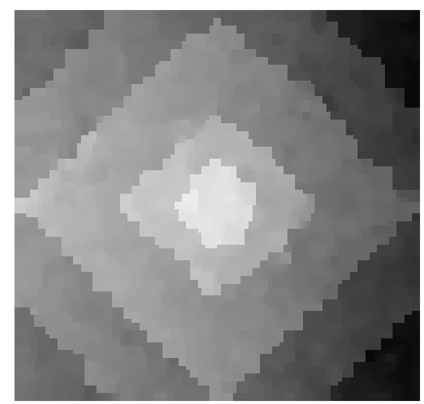


noisy image

Potts energy minimization



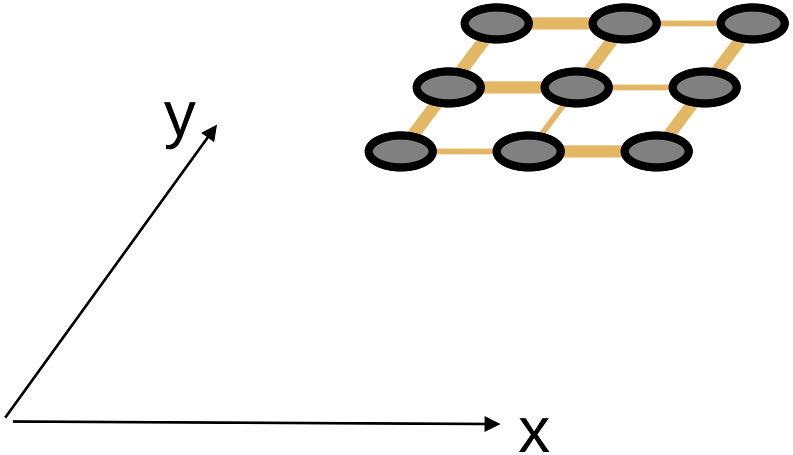
a local minimum
w.r.t. expansion moves



a local minimum
w.r.t. “one-pixel” moves

Ishikawa's Method – convex functions

Multi-scan-line stereo with *s-t* graph cuts (Roy&Cox'98)

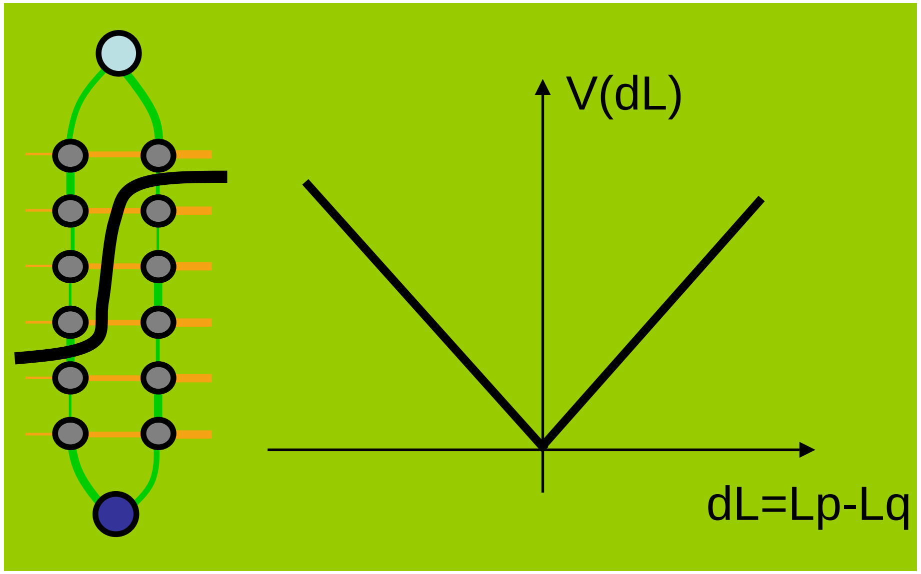


s-t graph-cuts for multi-label energy minimization

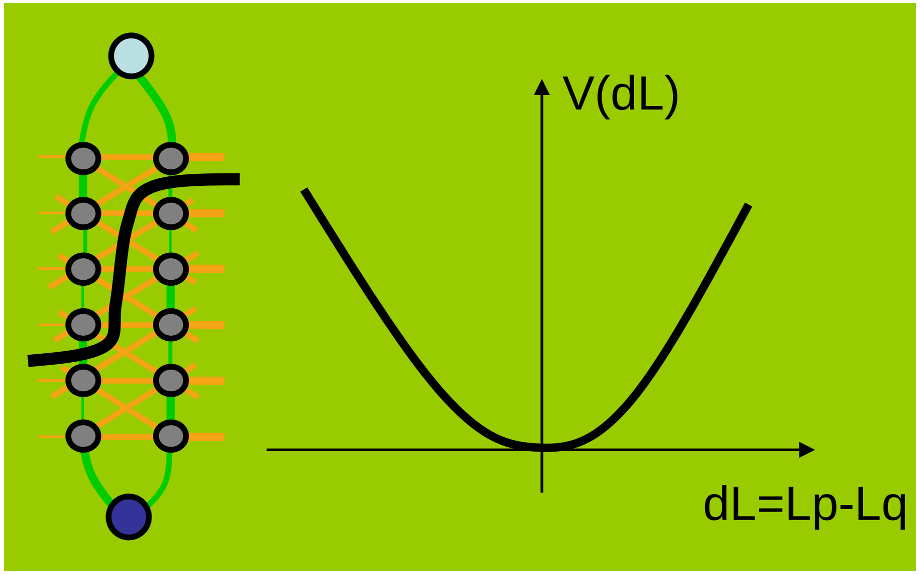
- Ishikawa 1998, 2000, 2003
- Modification of construction by Roy&Cox 1998

$$\boxed{E(L) = \sum_p -D_p(L_p) + \sum_{pq \in N} V(L_p, L_q)} \qquad L_p \in R^1$$

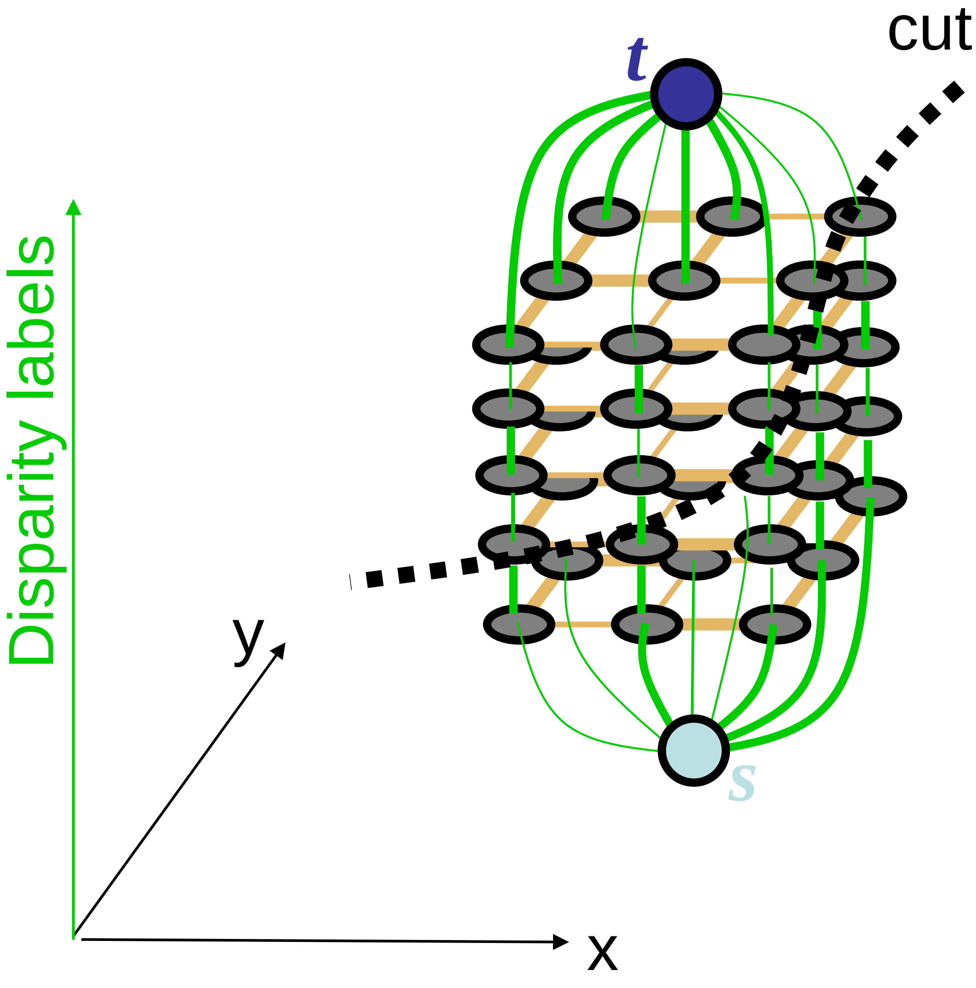
Linear interactions

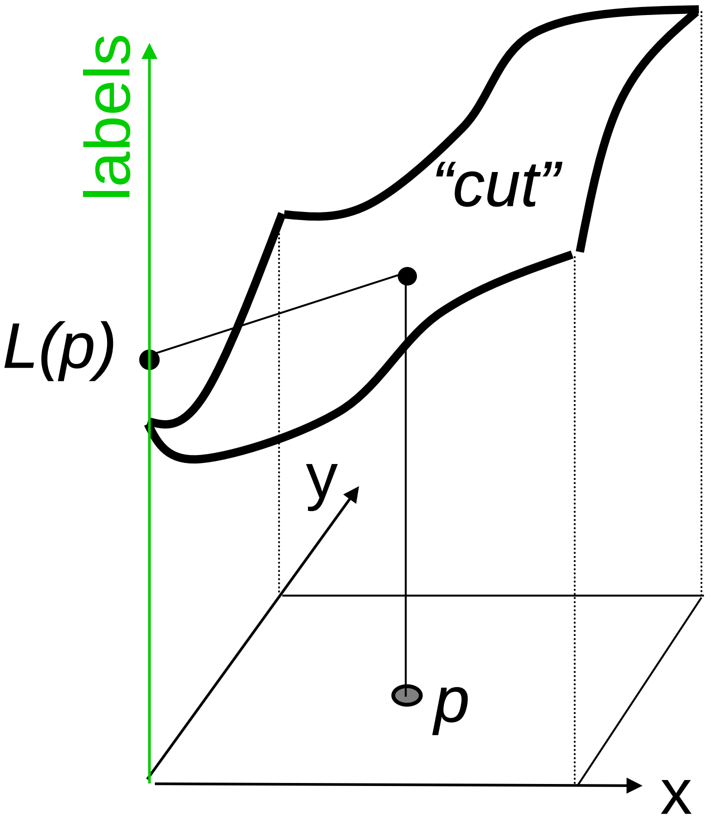


"Convex" interactions

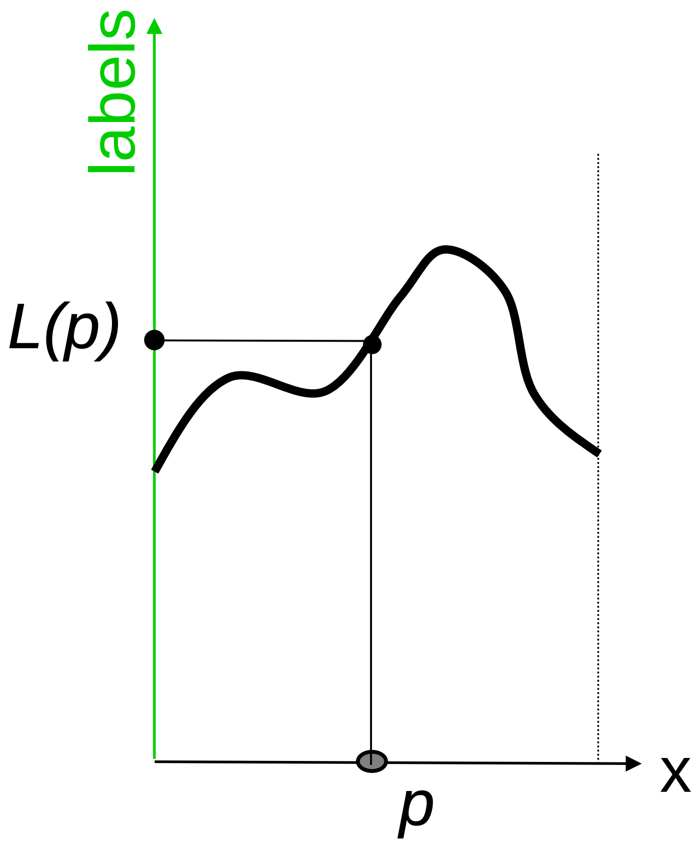


Multi-scan-line stereo
with *s-t* graph cuts (Roy&Cox'98)

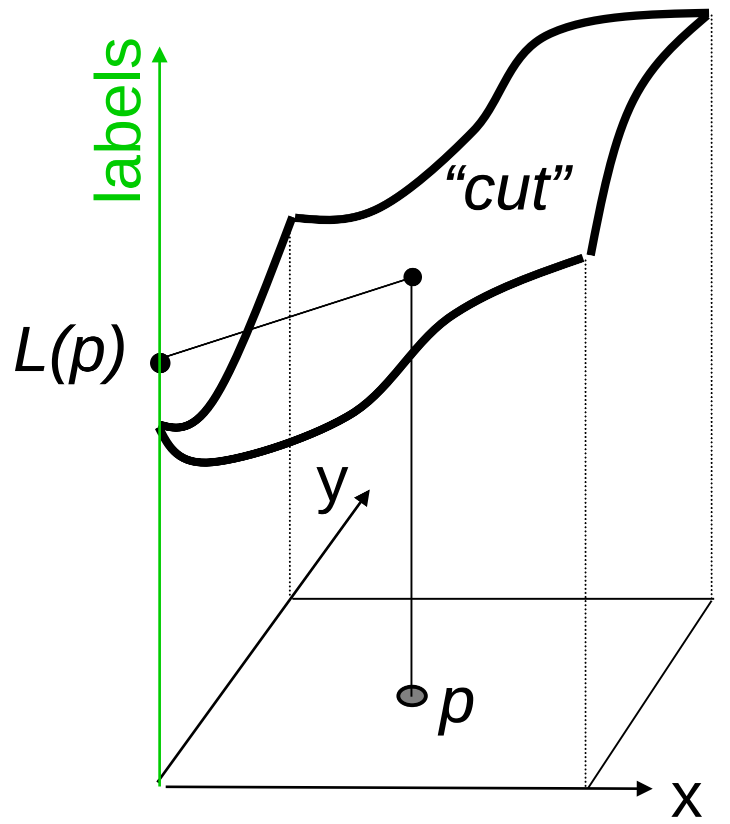




Scan-line stereo vs. Multi-scan-line stereo

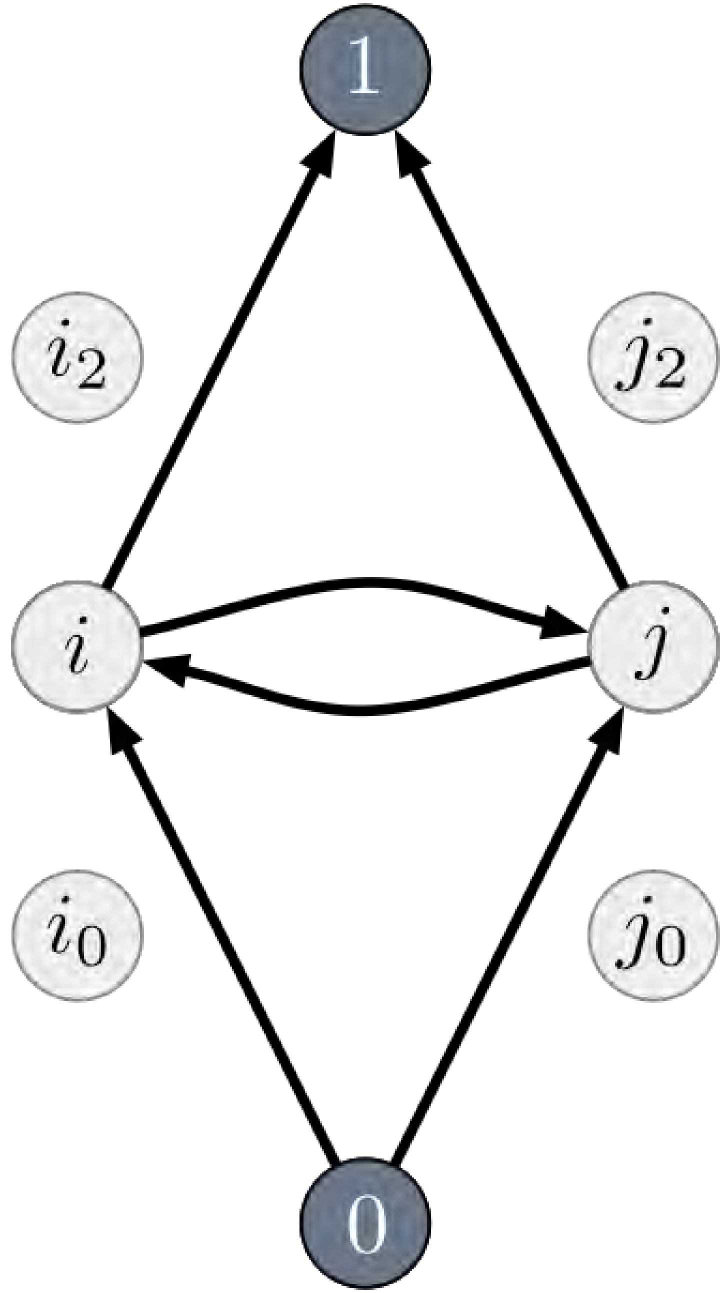


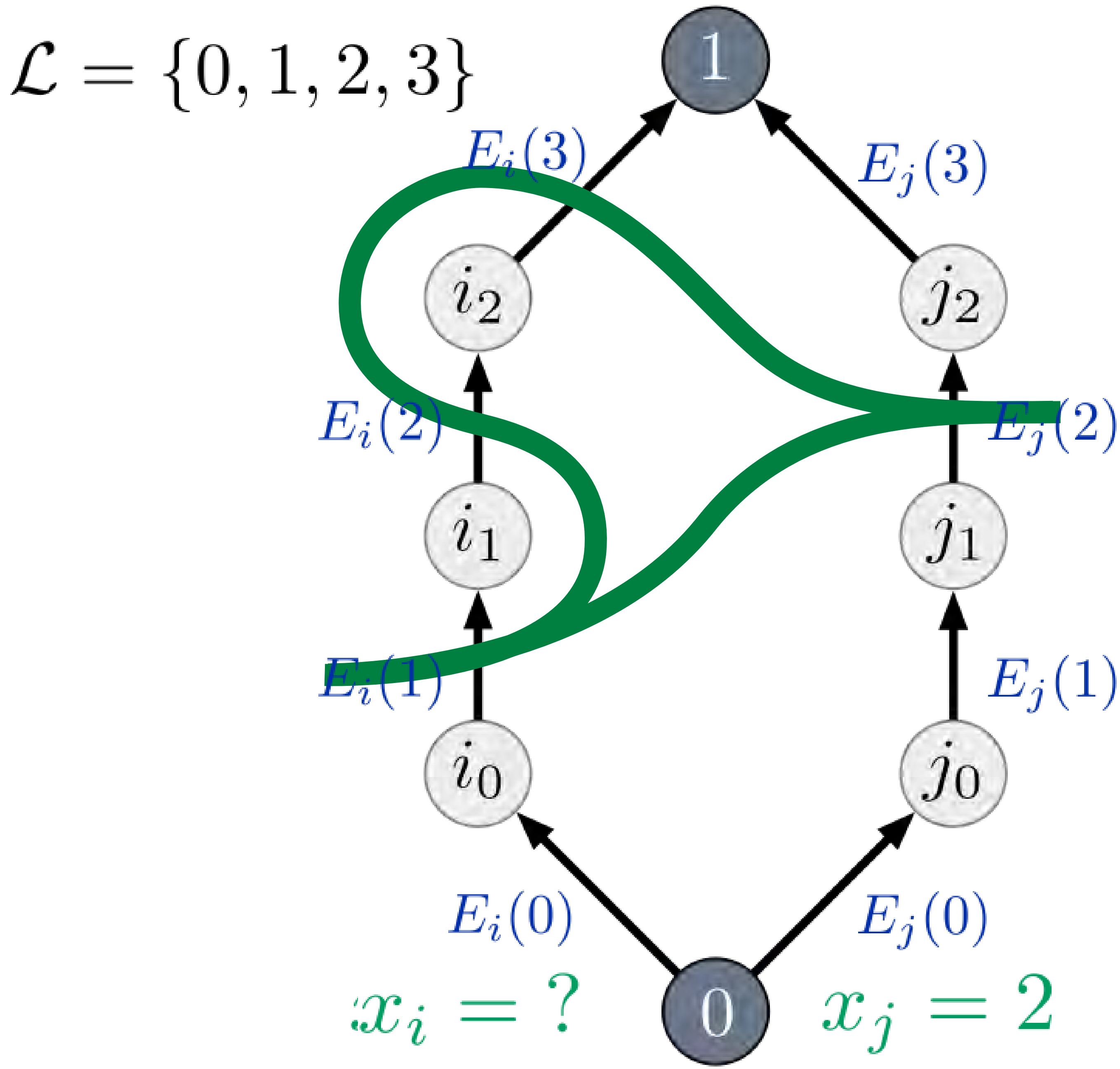
Dynamic Programming
(single scan line optimization)



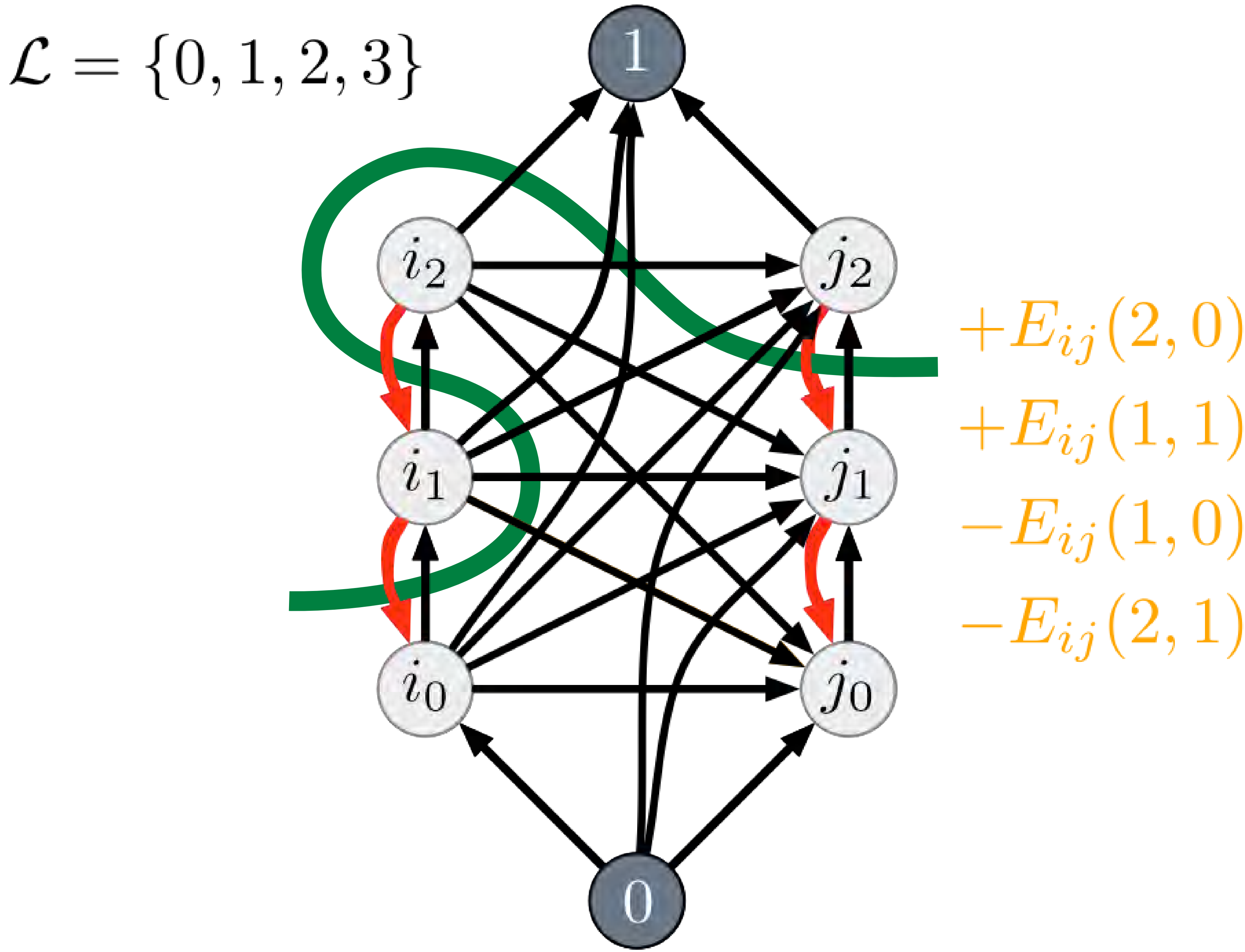
s-t Graph Cuts
(multi-scan-line optimization)

$\mathcal{L} = \{0, 1\}$





$\mathcal{L} = \{0, 1, 2, 3\}$
1
$E_i(3)$
$E_j(3)$
i_2
j_2
$E_i(2)$
$E_j(2)$
i_1
j_1
$E_i(1)$
$E_j(1)$
i_0
j_0
$E_i(0)$
$E_j(0)$
$x_i = ?$
0
$x_j = 2$



$\mathcal{L} = \{0, 1, 2, 3\}$
1
i_2
j_2
i_1
j_1
i_0
j_0
0
$+E_{ij}(2,0)$
$+E_{ij}(1,1)$
$-E_{ij}(1,0)$
$-E_{ij}(2,1)$

Label-swap algorithms

Alpha Expansion

α-expansion algorithm

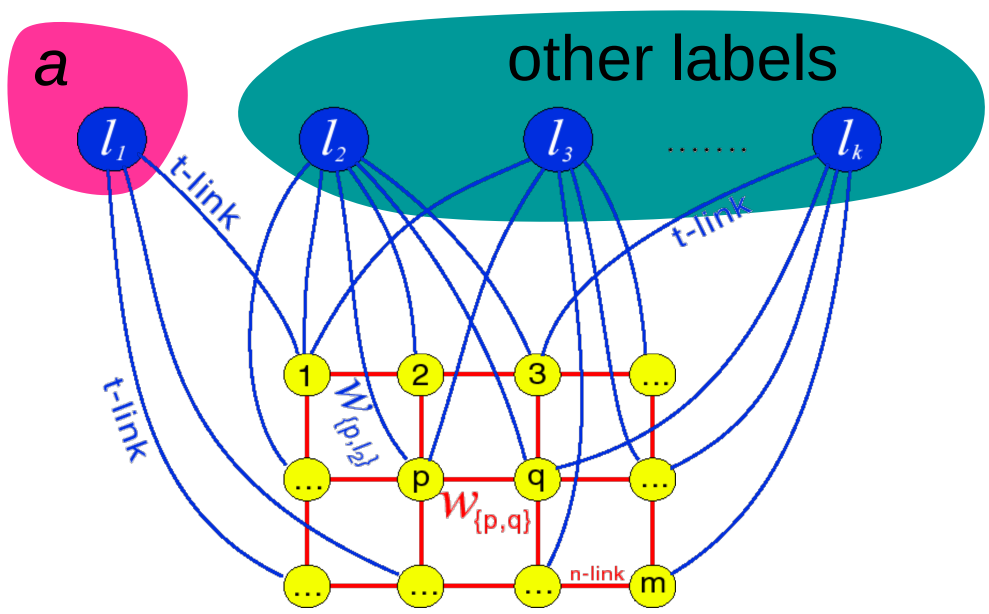
1. Start with any initial solution
2. For each label "α" in any (e.g. random) order

    1. *Compute optimal α-expansion move (s-t graph cuts)*
    2. *Decline the move if there is no energy decrease*

3. *Stop when no expansion move would decrease energy*

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

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**

Example: α-expansion

Status:

Expand Ground

Initialize with Tree

Expand House

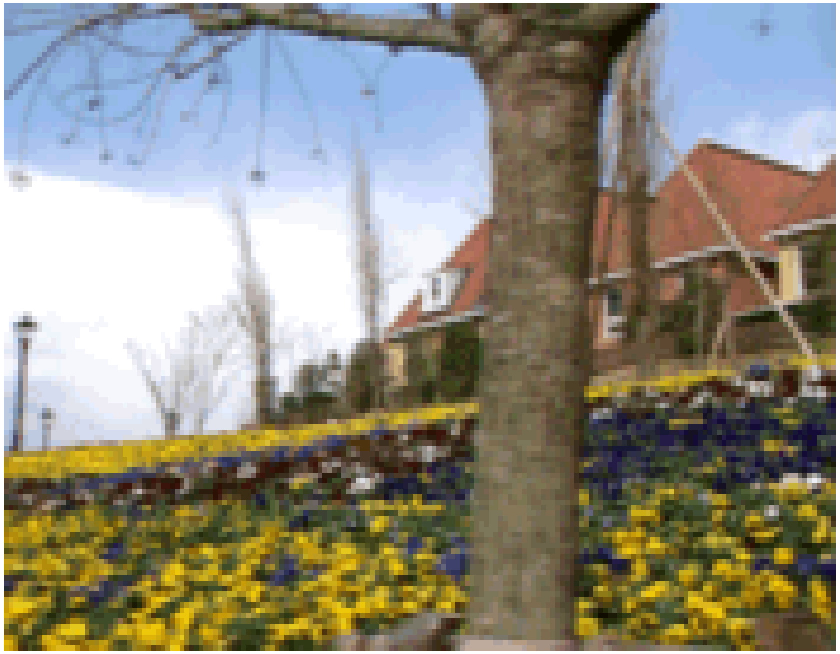
Expand Sky

Tree

Ground

House

Sky



Courtesy: Pushmeet Kohli

α-expansion

Let $\mathbf{u} \in \mathcal{B}^n$ be a binary labelling of the nodes. The new labelling $\mathbf{x}'$ is then defined by

$$\begin{aligned} x'_i &= x_i \;\; \text{if} \;\; u_i = 0 \\ &= \alpha \;\; \text{if} \;\; u_i = 1 \end{aligned}$$

Cost function is

$$E(\mathbf{x}') = \sum_{i=1}^{N} A_i(x'_i) + \sum_{1 \le i < j \le N} B_{ij}(x'_i, x'_j) \ .$$

Binary cost function $E^\alpha(\mathbf{u})$ will have the form

$$E^\alpha(\mathbf{u}) = \sum_{i=1}^{N} A_i^\alpha(u_i) + \sum_{1 \le i < j \le N} B_{ij}^\alpha(u_i, u_j) \ .$$

When is it submodular

The cost function $E^\alpha(\mathbf{u})$ will be submodular if for each such quadratic term we have

$$\Big(B_{ij}^\alpha(0,1) + B_{ij}^\alpha(1,0)\Big) - \Big(B_{ij}^\alpha(0,0) + B_{ij}^\alpha(1,1)\Big) \geq 0 \ .$$

This gives the required condition

$$\Big(B_{ij}(x_i,\alpha) + B_{ij}(\alpha,x_j)\Big) - \Big(B_{ij}(\alpha,\alpha) + B_{ij}(x_i,x_j)\Big) \geq 0$$

When is alpha expansion possible

1. Each variable has the option to switch to a given value α.
2. Problem is encoded as a binary optimization problem.

α-expansions for energies with *metric* interactions

- α-expansion algorithm applies to pair-wise interactions that are *metrics* on the space of labels (BVZ, PAMI'01)

$$V(a,a) = 0$$

$$V(a,b) \geq 0$$

$$V(a,b) \leq V(a,c) + V(c,b)$$

Triangular inequality

- *Example*: any truncated metric is also a metric

- => *Metric* case includes many **robust interactions**

a-expansions for energies with *submodular* interactions

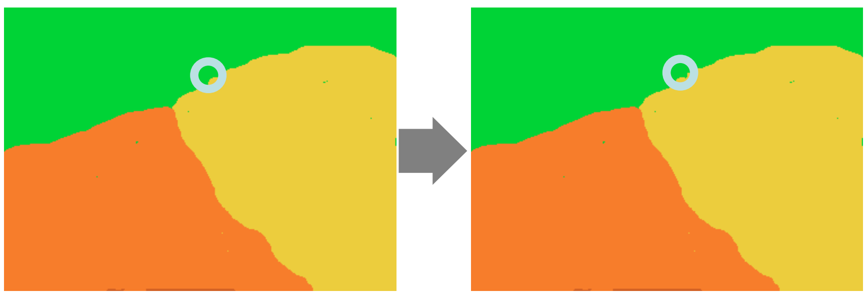
- *a*-expansions algorithm further generalizes to *submodular* pair-wise interactions

$$V(c,c)+V(a,b)\leq V(a,c)+V(c,b)$$

- follows from complete characterization of binary energies that can be minimized via *s-t* graph cuts (KZ 2002, 2004)

a-expansion algorithm vs. standard discrete energy minimization techniques

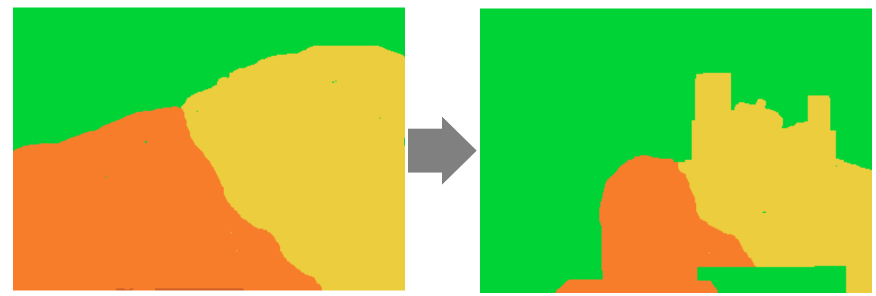
single “one-pixel” move
(simulated annealing, ICM,…)



- Only one pixel can change its label at a time

- Finding an optimal move is computationally trivial

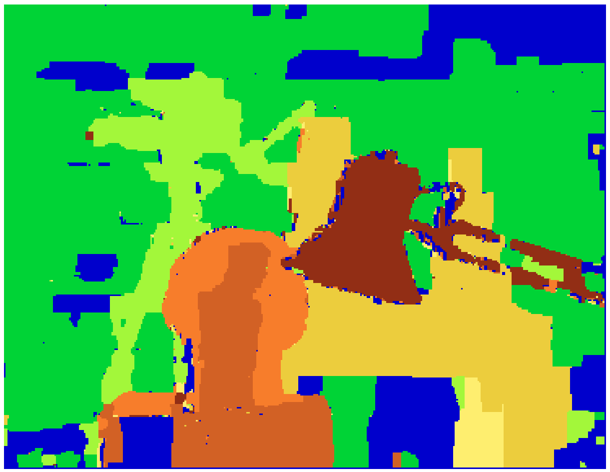
single *a*-expansion move



- Large number of pixels can change their labels simultaneously

- Finding an optimal move is computationally intensive O(2^n)
(*s*-*t* cuts)

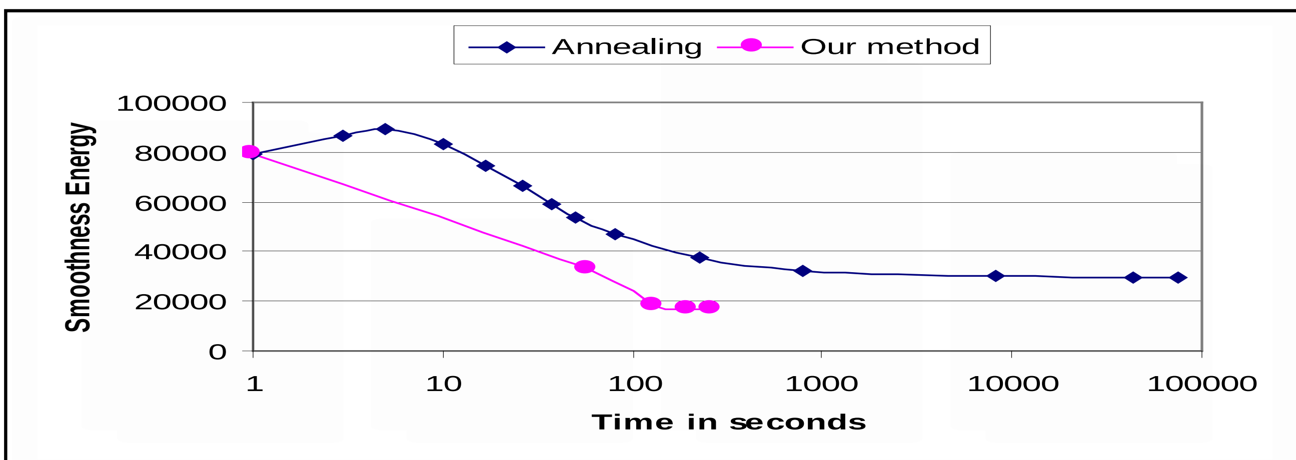
a-expansions vs. simulated annealing



normalized correlation,
start for annealing, 24.7% err

simulated annealing,
19 hours, 20.3% err

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



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 in practice
• solution is within the factor of 2 from the global minima	• Theoretically, solution reaches the global minimum.
• 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

Convex alpha expansion

1. Each variable has the option to switch to a given value α.
2. Problem is encoded as a binary optimization problem.

Alpha-beta Swap

Experimental Results

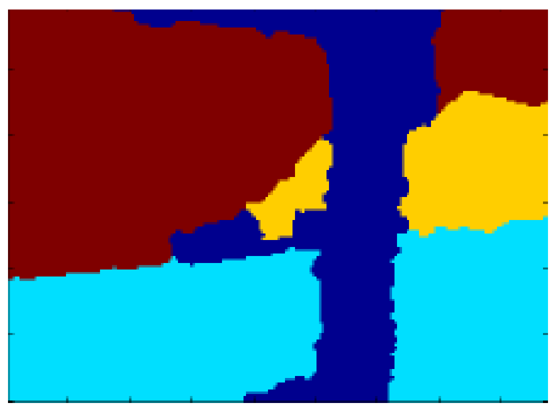
Pairwise | **Higher Order**

Original

Swap (4.7 sec)

Swap (5.0 sec)

Ground Truth



Expansion (3.7sec)

Expansion (4.4 sec)