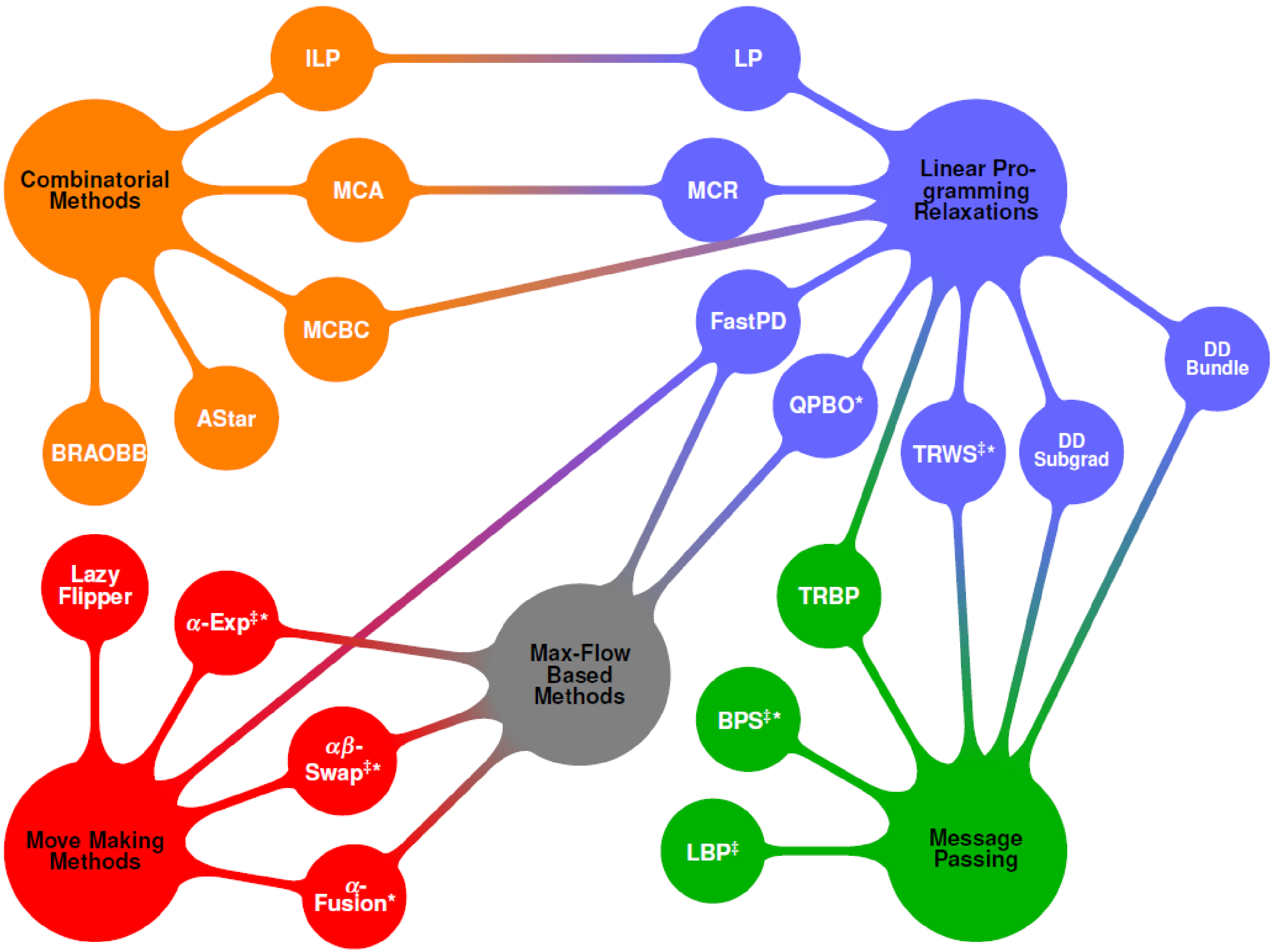
Image Segmentation with Markov Random Fields (Part 1)

Carsten Rother

Roadmap

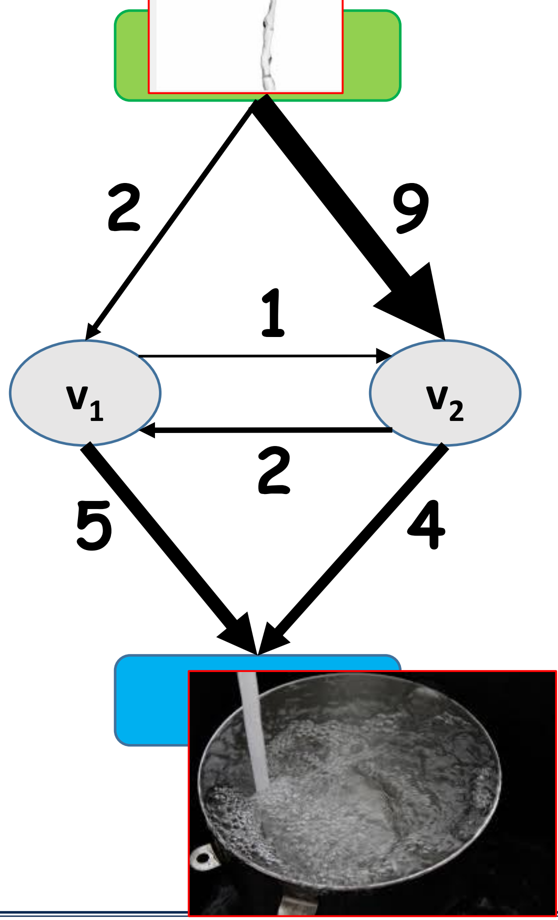
- Recap
- Higher-Order Models in Computer Vision
- Image Segmentation with Markov Random Fields

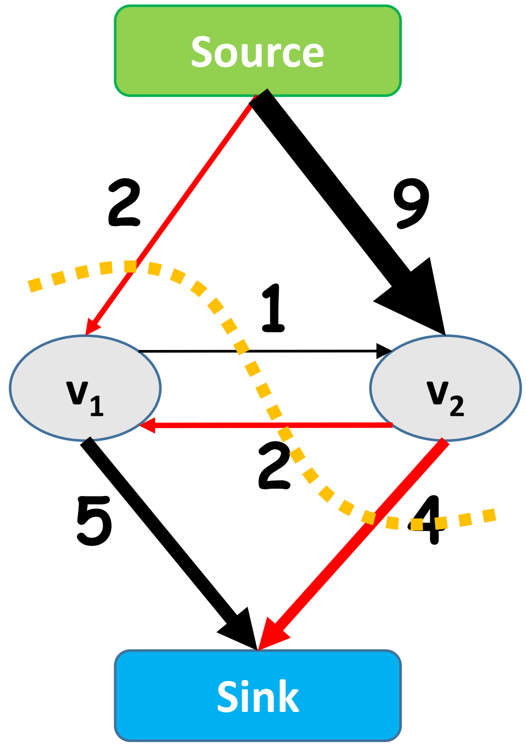
Recap: Optimization in Markov Random Fields



Recap: Visualization & cut

How much water can you push through?





The minimum cut is defined by the saturated edges of the maximum flow.

Recap: Alpha-Expansion: visually

- Variables take label α or retain current label

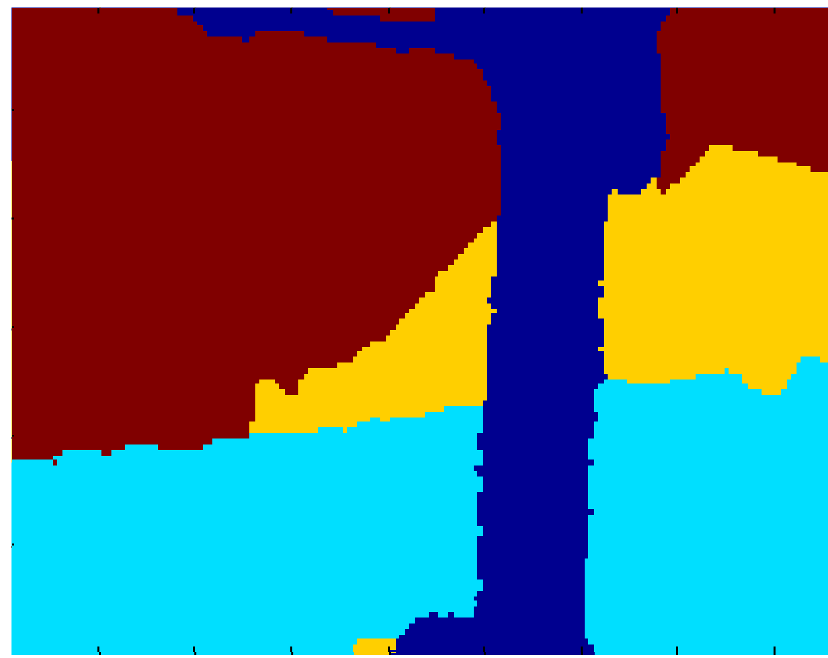
Tree

Ground

House

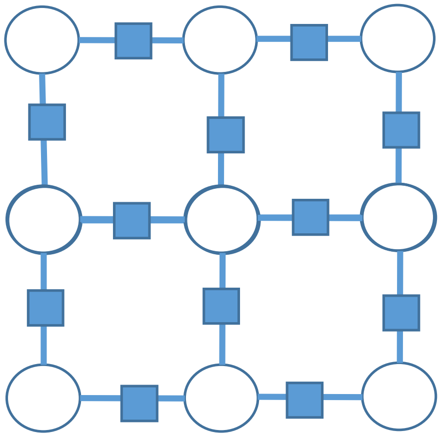
Sky

Status: Initialize with Tree Expand Ground Expand House Expand Sky



[Boykov , Veksler and Zabih 2001]

Recap: Examples: Order

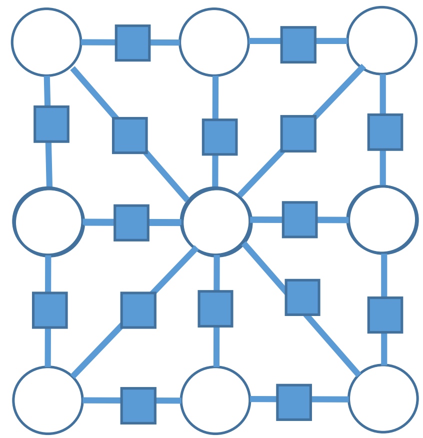


4-connected; pairwise MRF

$$E(\boldsymbol{x}) = \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j)$$

Order 2

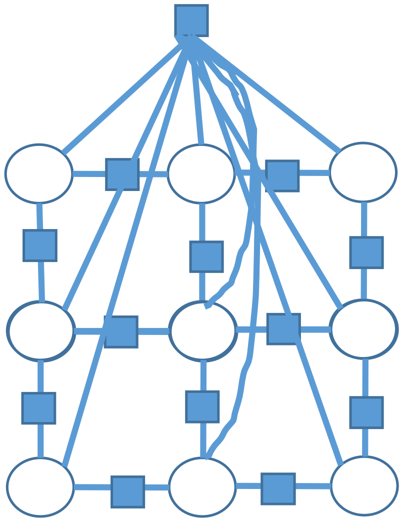
"Pairwise energy"



higher(8)-connected; pairwise MRF

$$E(\boldsymbol{x}) = \sum_{i,j \in N_8} \theta_{ij}(x_i, x_j)$$

Order 2



Higher-order RF

$$E(\boldsymbol{x}) = \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j) + \theta(x_1, \ldots, x_n)$$

Order n

"higher-order energy"

Recap: Higher-Order Optimization

<u>Usage:</u>

- "Window-based"
  - Image Restoration (de-noising, de-convolution) (better local model for texture and images)
  - Depth from Stereo, curvature model for surfaces and segmentation
  - Semantic Segmentation (P^n Potts, Curvature)

- "Image-wide"
  - Connectivity of a segmentation or surface
  - Image Restoration (de-noising, de-convolution) (better global model texture and images)
  - Semantic Segmentation (co-occurance statistic)

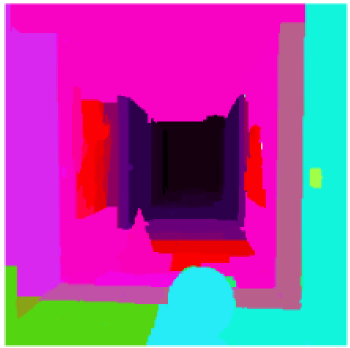
More likely a tennis ball than a lemon

<u>Optimization strategies:</u>

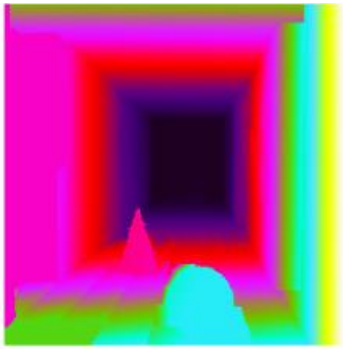
- Re-write higher-order energy as a pairwise energy
- Higher-order Message Passing
- Problem de-composition
- Etc.

Recap: “Window-based”: Depth from Stereo

Left stereo image

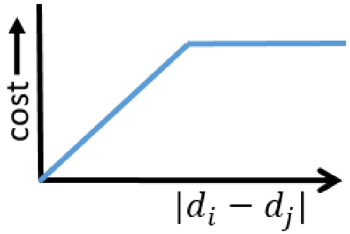


Depth map (color coded) using pairwise prior

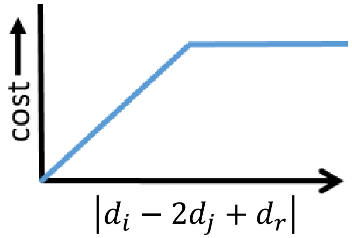


Depth map (color coded) using 3-pixel prior

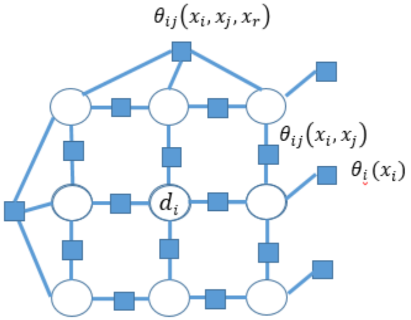
$$\theta(d_i, d_j) = \min(|d_i - d_j|, \tau)$$



$$\theta_{ijr}(d_i, d_j, d_r) = \min(|d_i - 2d_j + d_r|, \tau)$$



Robust curvature measure



$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j) + \sum_{i,j,r \in N} \theta_{ijr}(x_i, x_j, x_r)$$

$x_i \in \{1, \ldots, D\}$

[Woodford, PAMI et al. 2009]

Recap: Optimization (binary case)

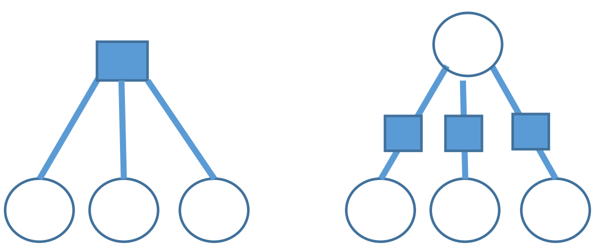
- In general we cannot re-write

  $\theta(x_i, x_j, x_r)$ as $\theta(x_i, x_j) + \theta(x_i, x_r) + \theta(x_j, x_r)$
  such that they are the same for all values of (x_i, x_j, x_r)

- Let us write:

$$\theta(x_i, x_j, x_r) = \theta_{111} x_i x_j x_r + \theta_{110} x_i x_j (1 - x_r) + \theta_{100} x_i (1 - x_j)(1 - x_r) + \ \ldots$$
$$= \ a x_i x_j x_r + \underbrace{b x_i x_j + c x_i x_r + \cdots + d}_{\text{Quadratic polynomial are standard pairwise terms}}$$

- The idea is to transform the 3rd order into many pairwise terms (there are many possible methods, we discuss one)



Recap: Optimization (binary case)

Transformation by "substitution"

$f(x_1,x_2,x_3) = ax_1x_2x_3 + bx_1x_2 + cx_2x_3 + \ldots$

$x_i \in \{0,1\}$

a,b,c are constant

Define auxiliary function:

$D(x_1,x_2,z) = x_1x_2 - 2x_1z - 2x_2z + 3z \qquad z \in \{0,1\}$

It is (check yourself)

$D(x_1,x_2,z) = 0$ if $x_1x_2 = z$

$D(x_1,x_2,z) > 0$ if $x_1x_2 \neq z$

Apply Substitution:

$$f(x_1,x_2,x_3) = \min_z g(x_1,x_2,x_3,z) = azx_3 + bz + cx_2x_3 + \ldots + K\,D(x_1,x_2,z)$$

when K is very large then $x_1x_2 = z$

Optimization problem:

$$\min_{x_1,x_2,x_3} f(x_1,x_2,x_3) = \min_{x_1,x_2,x_3,z} g(x_1,x_2,x_3,z)$$

Problems:

- Does not work well in practice (see [Ishikawa CVPR '09])
- Function D is **non-submodular** and "K enforces this strongly"

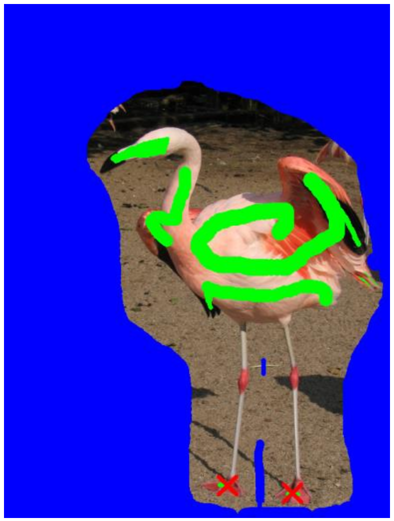
[Rosenberg '75, Boros and Hammer '02, Ali et al. ECCV '08]

Roadmap

- Recap
- Higher-Order Models in Computer Vision
- Image Segmentation with Markov Random Fields

“Image-wide”: Connectivity of Segmentation

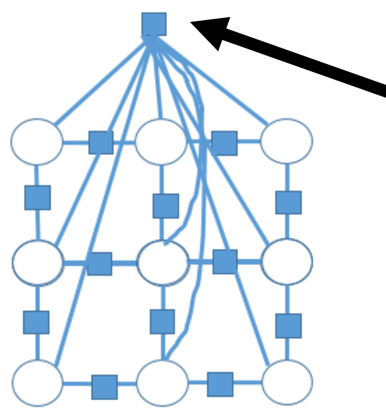
Foreground object must be connected:



User input

Standard MRF

with connectivity



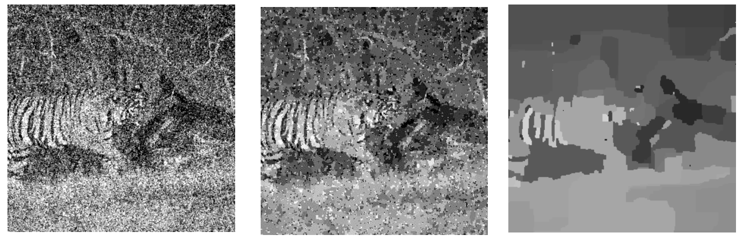
Check if the segmentation is connected?

[Vicente et al. ’08]

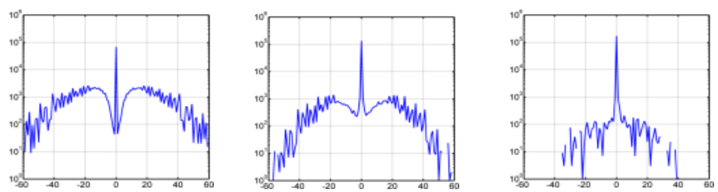
Global-Image Prior

Ground truth

Noisy input



Results: increased pairwise strength



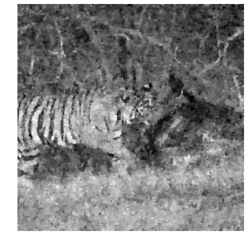
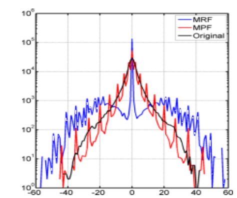
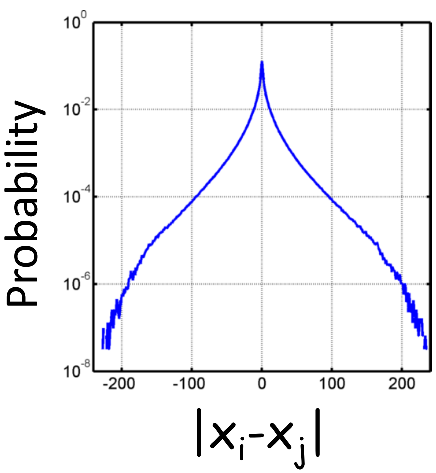
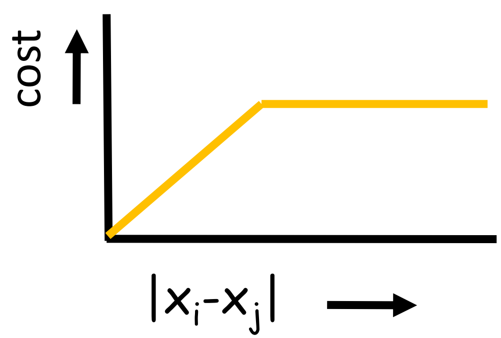


Image has
red curve statistics





From Ground truth images



Introduce a global term,
which controls the
global statistic for $|x_i - x_j|$

[Woodford et. al. ICCV '09]

P^n Potts Model

Example: Image Segmentation

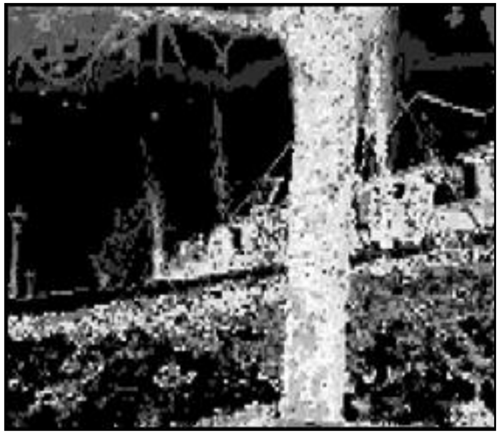
$$E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j|$$

n = number of pixels

$E: \{0,1\}^n \rightarrow R$

$0 \rightarrow fg, \; 1 \rightarrow bg$

Image



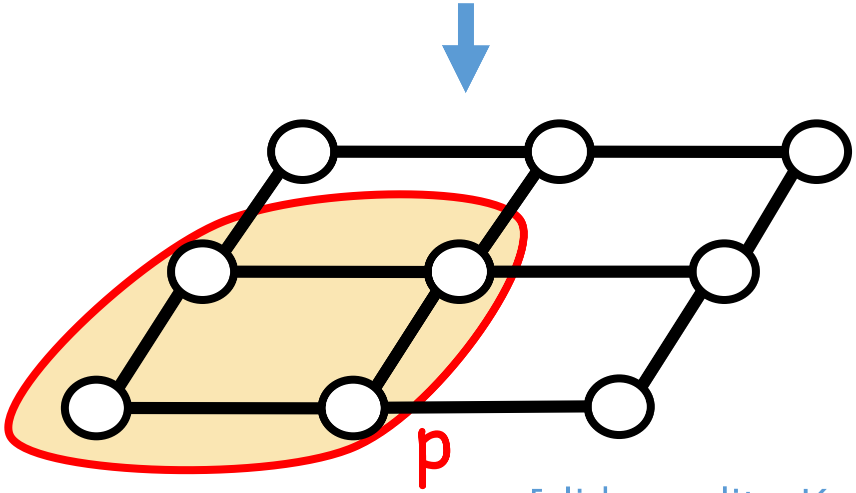
Unary Cost

Segmentation

[Boykov and Jolly ' 01] [Blake et al. '04] [Rother et al.`04]

P^n Potts Model

Patch Dictionary (Tree)



$$h(X_p) = \begin{cases} 0 & \text{if } x_i = 0,\ i \in p \\ C_{max} & \text{otherwise} \end{cases}$$

$$C_{max} \geq 0$$

[slide credits: Kohli]

P^n Potts Model

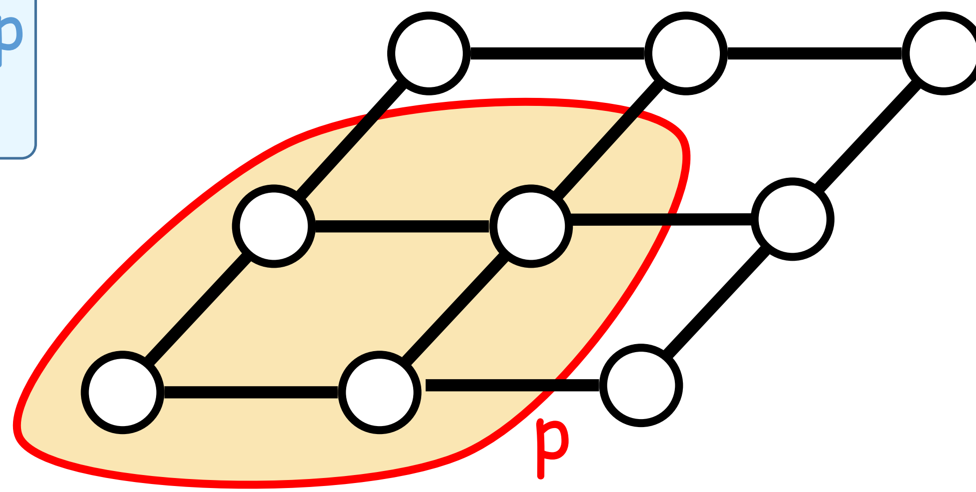
n = number of pixels

$E: \{0,1\}^n \rightarrow R$

$0 \rightarrow fg, \; 1 \rightarrow bg$

$$E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| + \sum_p h_p(X_p)$$

$$h(X_p) = \begin{cases} 0 & \text{if } x_i = 0, i \in p \\ C_{max} & \text{otherwise} \end{cases}$$



[slide credits: Kohli]

P^n Potts Model

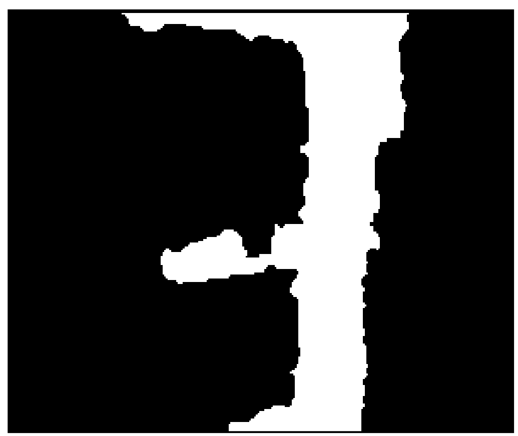
n = number of pixels

$E: \{0,1\}^n \rightarrow R$

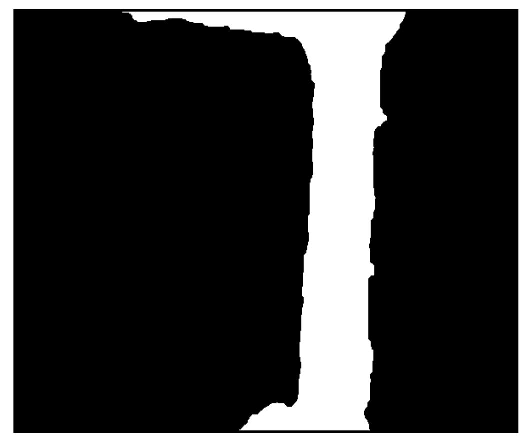
$0 \rightarrow fg, \ 1 \rightarrow bg$

$$E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| + \sum_p h_p(X_p)$$

Image



Pairwise Segmentation



Final Segmentation

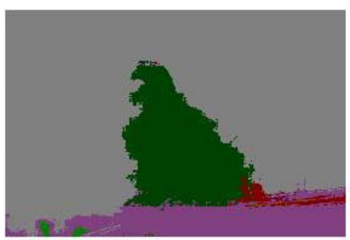
[slide credits: Kohli]

P^n Potts Model - Application

Image

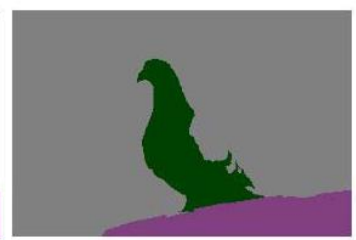
One super-pixelization

another super-pixelization



Unaries only from TextonBoost [Shotton et al. ‘06]

Pairwise CRF only [Shotton et al. ‘06]

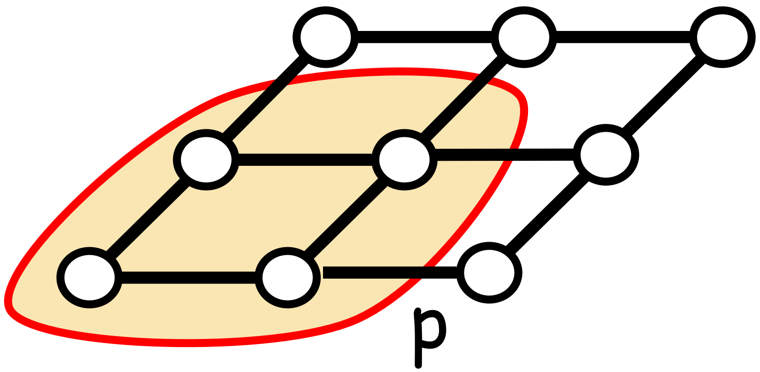


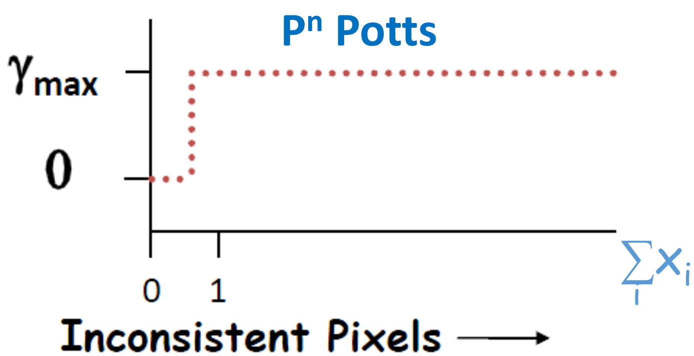
P^n Potts

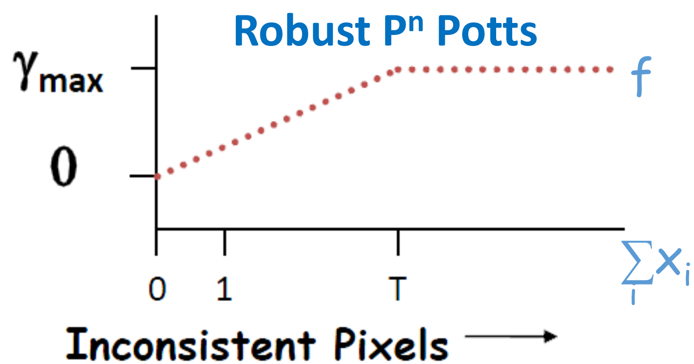
[slide credits: Kohli]

Robust P^n Potts Model

$$h(x_p) = \begin{cases} 0 & \text{if } x_i = 0,\ i \in p \\ f(\sum_i x_i) & \text{otherwise} \end{cases}$$







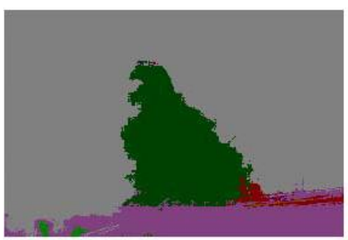
[slide credits: Kohli]

Robust P^n Potts Model - Application

Image

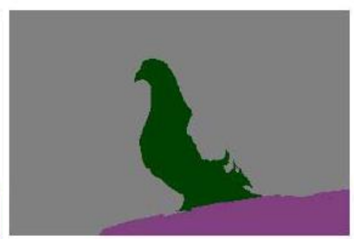
One super-pixelization

another super-pixelization

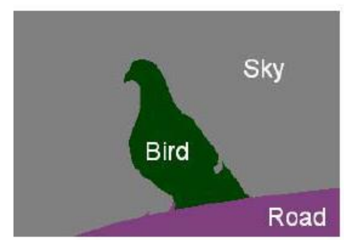


Unaries only
TextonBoost
[Shotton et al. '06]

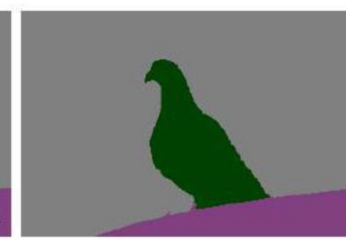
Pairwise CRF only
[Shotton et al. '06]



P^n Potts



robust P^n Potts



robust P^n Potts
(different f)

[slide credits: Kohli]

Robust P^n Potts Model - Application

One input image

Ground truth depth

Stereo without robust P^n Potts

Stereo with robust P^n Potts

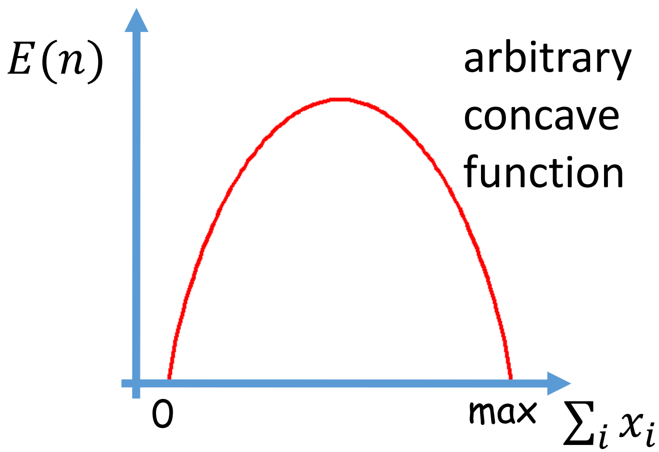
Very good result for e.g. Middlebury Teddy Image

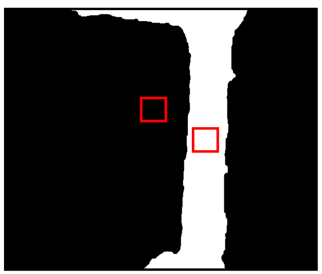
[Bleyer at al. CVPR '10]

P^n Potts Model: Optimization

Lets define: $n = \sum_i x_i$

Optimize: $\min_{\boldsymbol{x}} E'(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j) + E(n)$





For a binary segmentation we can enforce that in a window all pixels are more likely "all 0" or "all 1", but less likely 50% 0 and 50% 1.

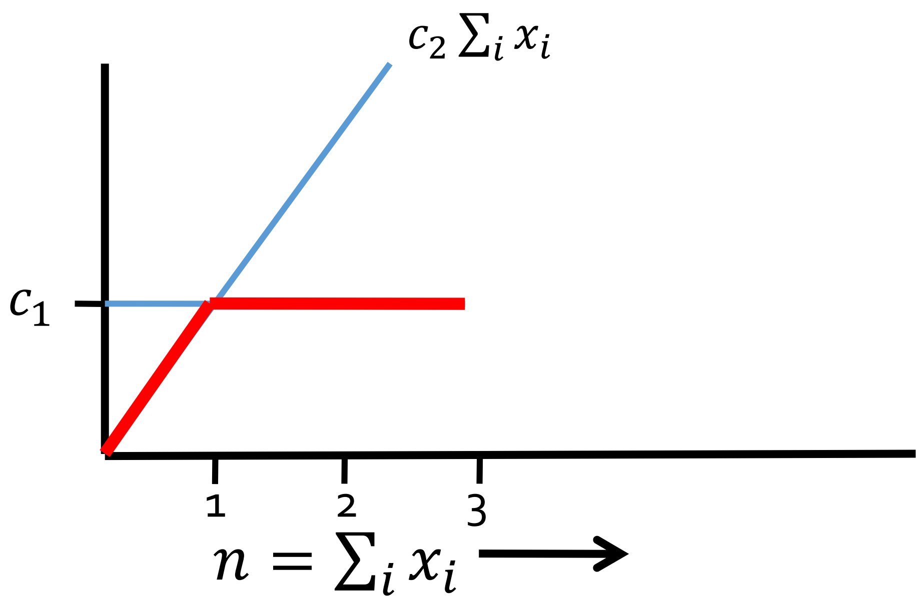
Goal: convert this higher-order function into a pairwise function

P^n Potts Model: Optimization

Higher-order Energy:

$$\min_{\boldsymbol{x}} E(n) = \min_{\boldsymbol{x}} \min(c_1, c_2 n)$$

$n = \sum_i x_i$

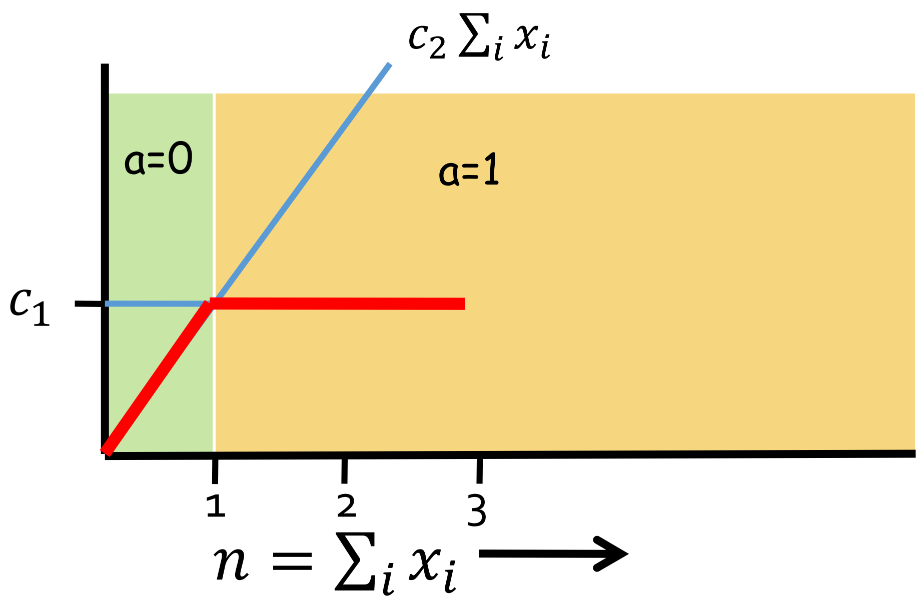


P^n Potts Model: Optimization

Higher-order Energy:

$$n = \sum_i x_i$$
$$a \in \{0,1\}, c_2 \geq 0$$

$$\min_{\boldsymbol{x}} E(n) = \min_{\boldsymbol{x}} \min(c_1, c_2 n)$$
$$= \min_{\boldsymbol{x},\boldsymbol{a}} \underbrace{a c_1 + (1-a)(c_2 n)}_{E'(n,a)}$$



Question

What is the function
$E'(n, a) = ac_1 + (1-a)(c_2 n)$?

$$n = \sum_i x_i$$
$$a \in \{0,1\}, c_2 \geq 0$$

1) Submodular
2) Not submodular
3) Sometimes submodular
4) All of the above
5) I don't know

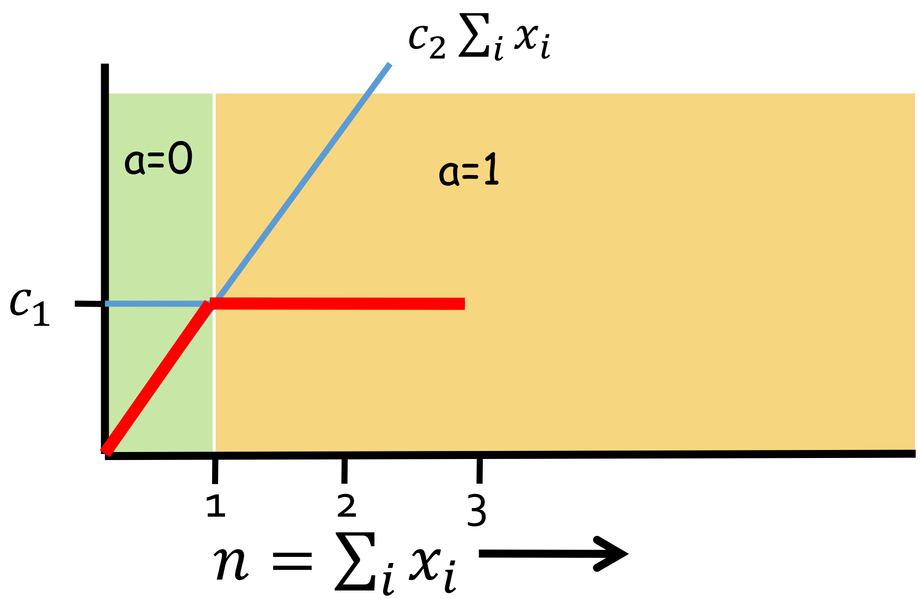
P^n Potts Model: Optimization

Higher-order Energy:

$$n = \sum_i x_i$$
$$a \in \{0,1\}, c_2 \geq 0$$

$$\min_{\boldsymbol{x}} E(n) = \min_{\boldsymbol{x}} \min(c_1, c_2 n)$$
$$= \min_{\boldsymbol{x},\boldsymbol{a}} \; a c_1 + (1-a)(c_2 n)$$
$$= \min_{\boldsymbol{x},\boldsymbol{a}} \; a c_1 + c_2 \sum_i x_i + \sum_i -c_2 x_i a$$

function is pairwise and submodular since $-c_2 x_i a \leq 0$



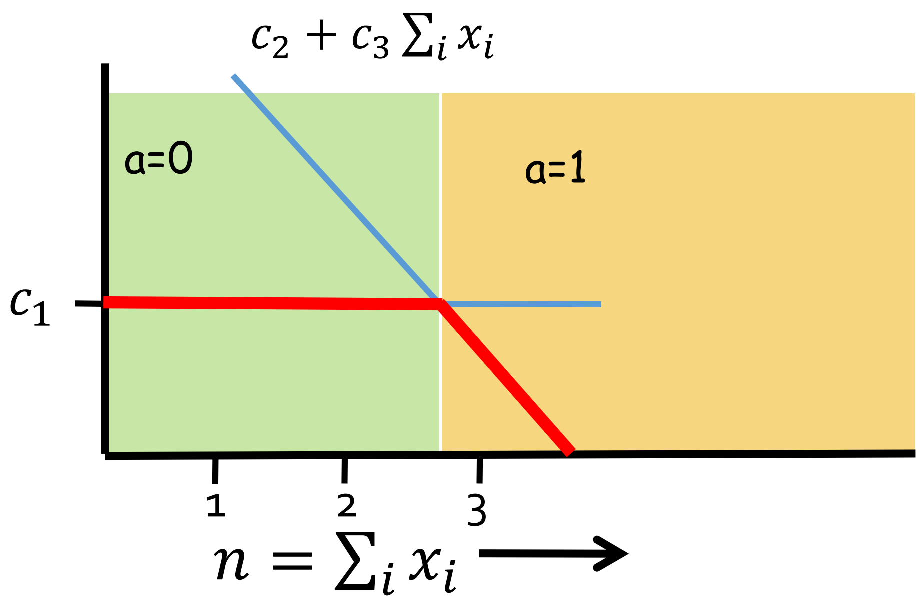
P^n Potts Model: Optimization

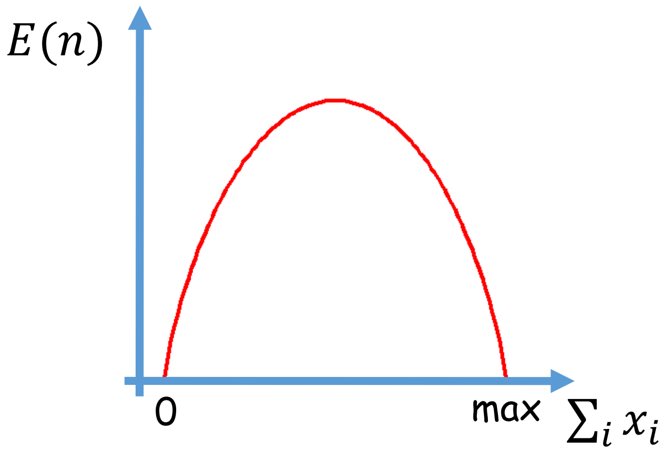
Higher-order Energy:

$$n = \sum_i x_i$$
$$a \in \{0,1\}, c_3 \leq 0$$

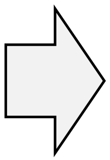
$$\min_{\boldsymbol{x}} E(n) = \min_{\boldsymbol{x}} \min(c_1, c_2 + c_3 n)$$
$$= \min_{\boldsymbol{x},\boldsymbol{a}} \; (1-a)c_1 + a(c_2+c_3 n)$$
$$= \min_{\boldsymbol{x},\boldsymbol{a}} \; c_1 - ac_1 + ac_2 + \sum_i c_3 x_i a$$

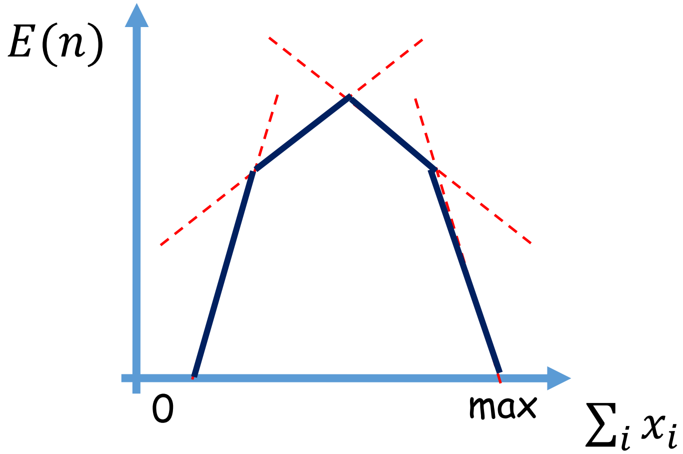
function is pairwise and submodular since $c_3 x_i a \leq 0$



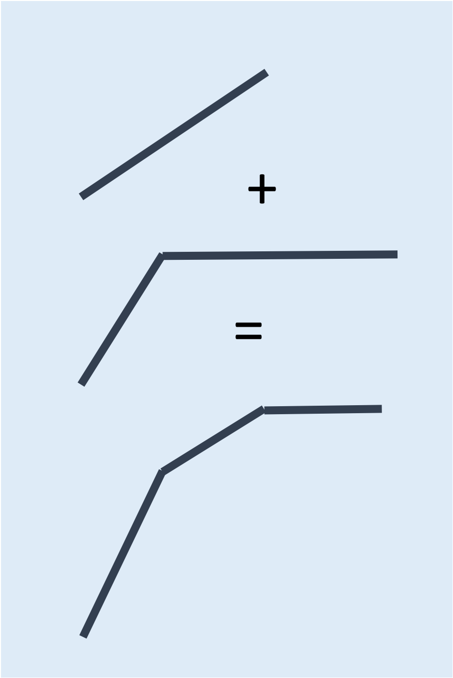


Arbitrary concave function





Approximate with lower envelop of linear functions



Roadmap

- Recap
- Higher-Order Models in Computer Vision
- Image Segmentation with Markov Random Fields

Demo

Demo

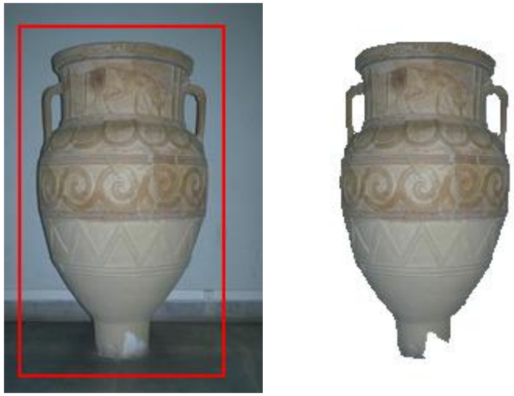
What is Segmentation?

- Figure-Ground Segmentation (Binary Segmentation)

  (often done with user interaction)

Brush input

[Lazy Snapping;
Li et al. SIGGRAPH 2004)



Bounding Box input

GrabCut (Rother et al. Siggraph 2004)

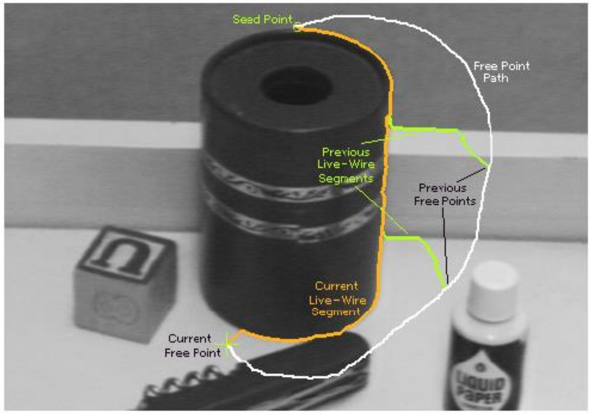


Figure 2: *Image demonstrating how the live-wire segment adapts and snaps to an object boundary as the free point moves (via cursor movement). The path of the free point is shown in white. Live-wire segments from previous free point positions (t_0, t_1, and t_2) are shown in green.*

Tracing the boundary

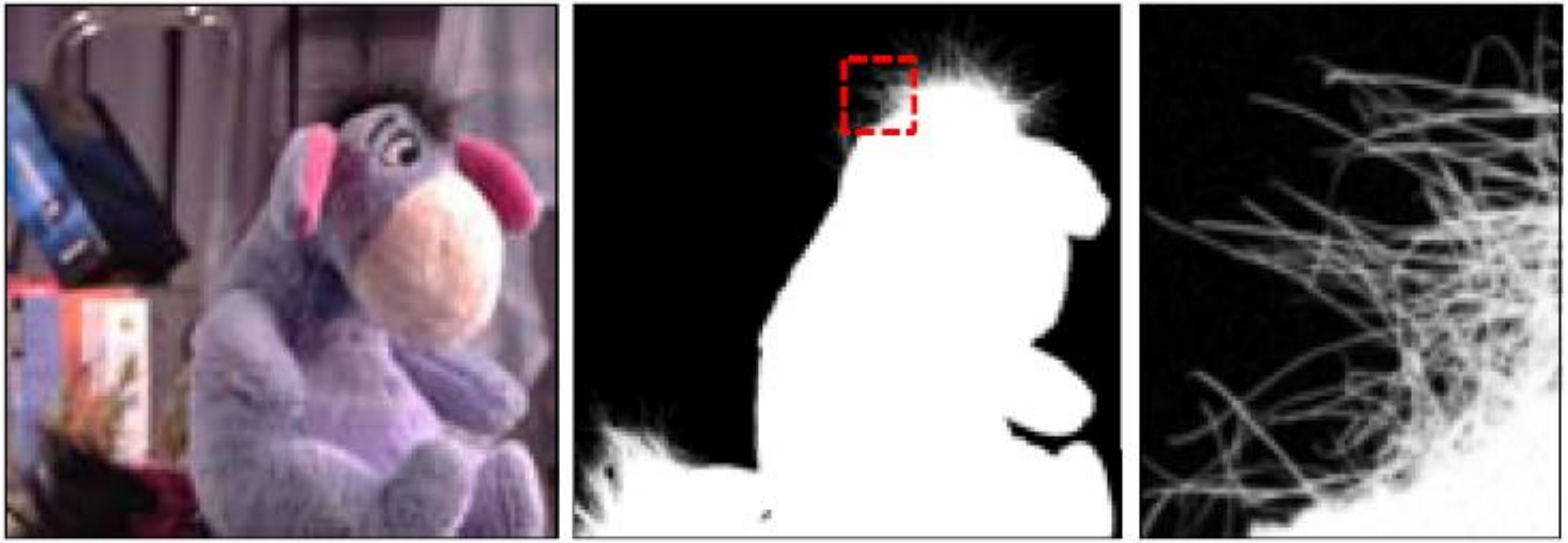
[Mortensen & Barrett,
Siggraph 1995, CVPR 1999]

<u>In this lecture series:</u>

- Graph-cut based segmentation
- Bounding Box segmentation
- Joint estimation of appearance and segmentation
- Gaussian MRF (Random Walk versus Graph Cut)
- (Variational methods)

What is Segmentation?

- Image Matting: Going from binary values to fractional values

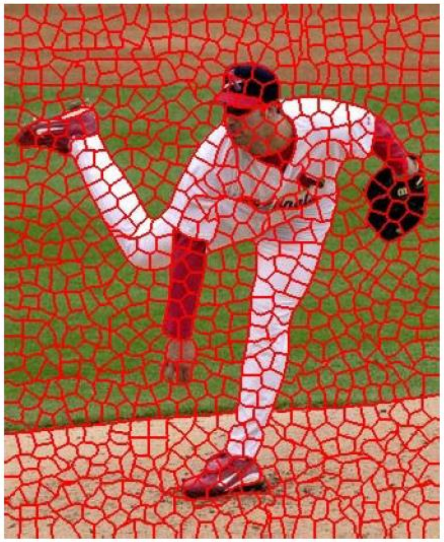


[Online Benchmark for matting; Rhemann, Rother et al. CVPR 2009]

In this lecture series:
- Matting Laplacian (Gaussian MRF)

What is Segmentation?

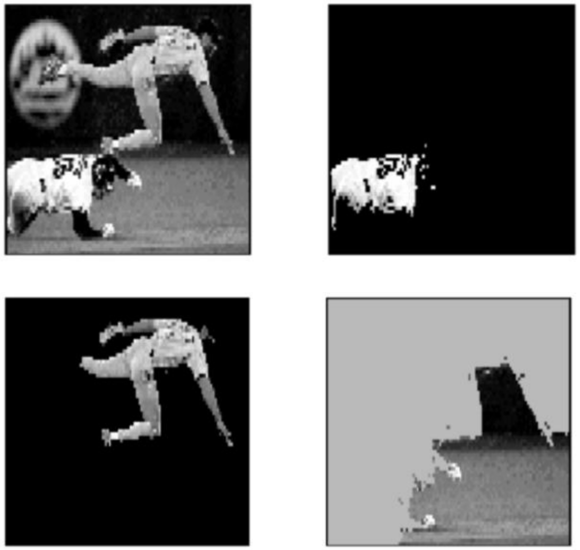
- Image Partitioning



Superpixel segmentation

Color Quantization with KMeans Clustering

Color-based segmentation



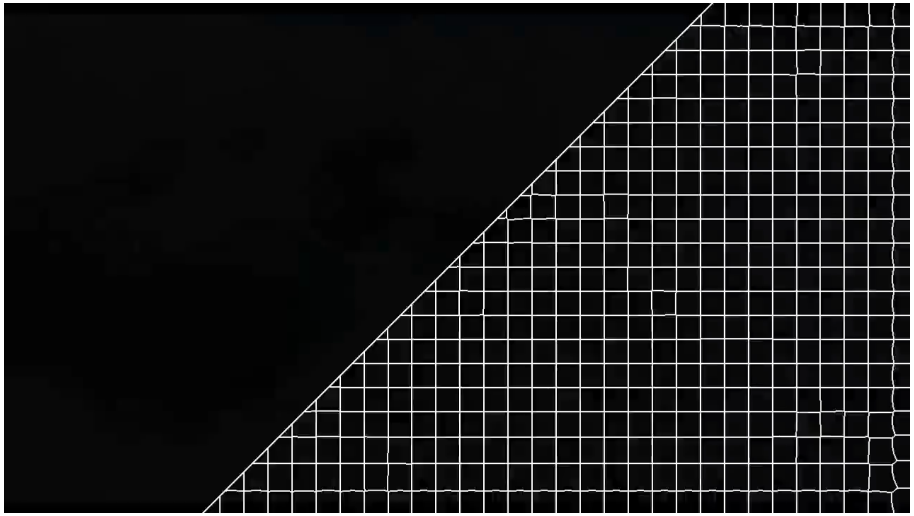
Object-level partitioning

<u>In this lecture series:</u>
- K-means clustering
- Mean-shift
- Normalized cuts

What is Segmentation?

- All of the above exists also for Video

Video Binary segmentation

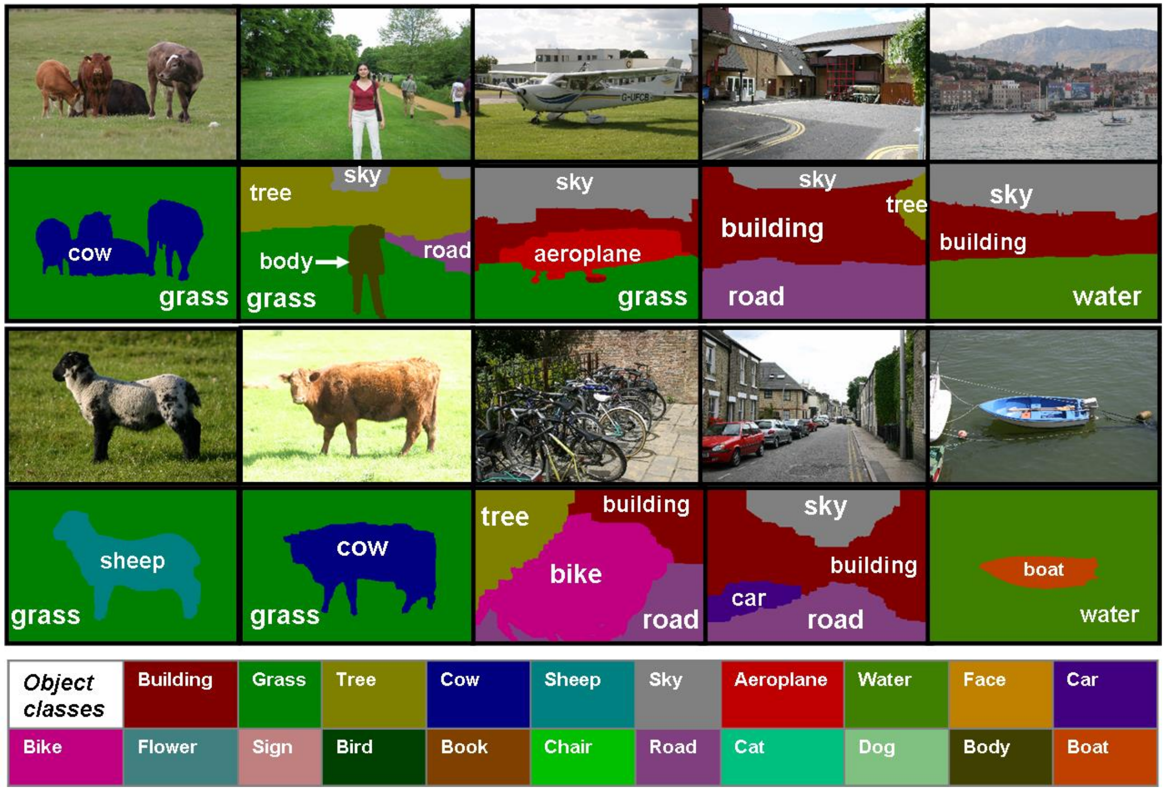


Video Superpixels

(this topic is not covered in this lecture)

What is Segmentation?

- Semantic Image segmentation [TextonBoost; Shotton et al, '06]



Label each pixel with one out of 21 classes

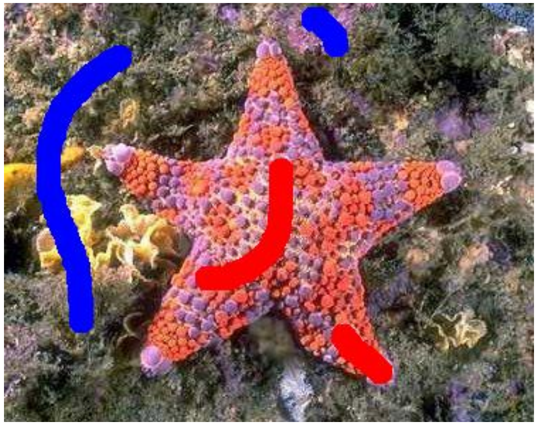
In this lecture series:
- Covered very briefly later

What is it useful for?

- Image/Video Editing (Adobe Photoshop, ect)
- Film Industry (The Foundry)

- Semantic Segmentation is relevant for many areas:
  - Autonomous Driving
  - Augmented Reality
  - Robotics

Interactive Segmentation



$$\boldsymbol{z} = (R, G, B)^n$$

Goal

$$\boldsymbol{x} = \{0,1\}^n$$

(user-specified pixels are not optimized for)

<u>Two-step approach:</u>

1. **Modelling**: Write down the model in form of an energy function
2. **Optimization**: find the minimal configuration of the energy

Comment on Feature for segmentation

- Image (or Video) Segmentation (Partitioning) use different types of features:
  - Position in the image (e.g. sky more likely art top of image)
  - Color
  - Texture
  - Motion (e.g. moving car)
  - Size and Orientation of super-pixels
  - Etc.

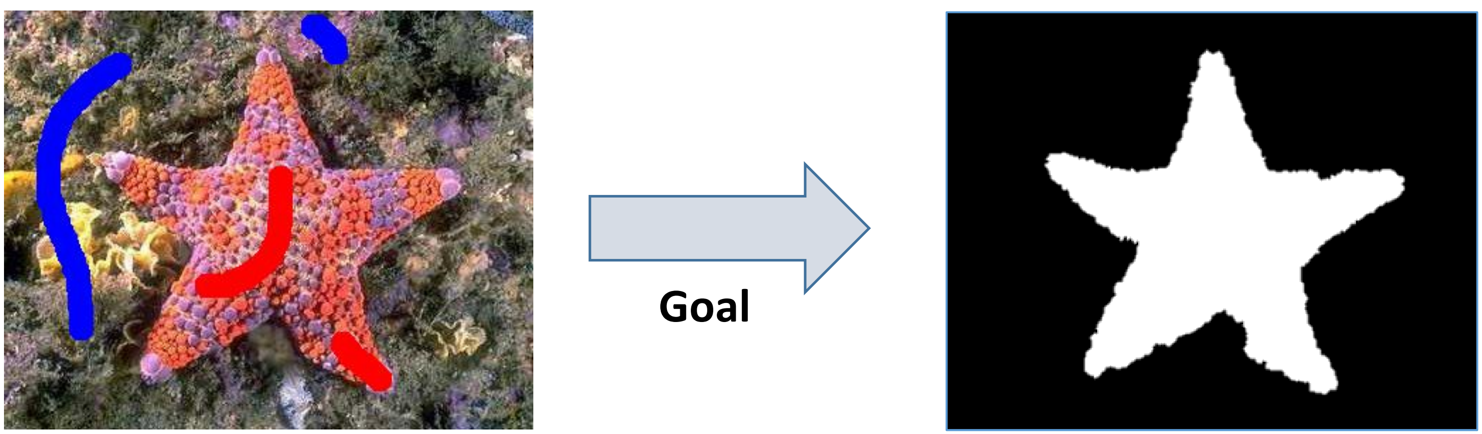
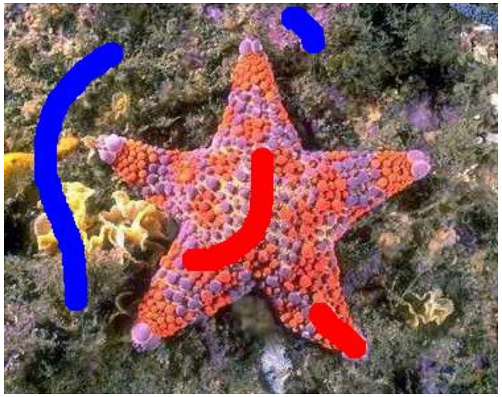
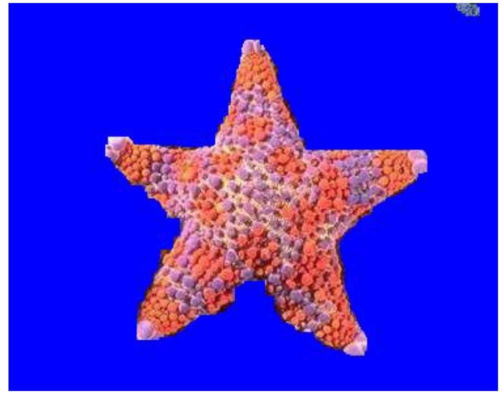


Image Segmentation



Input Image
with user brush strokes
(blue-background; red-foreground)



Desired binary output labeling

<u>We will use the following energy:</u>

$$E(\boldsymbol{x}) = \sum_{i \in N} \underbrace{\theta_i(x_i)}_{\text{Unary term}} + \sum_{i,j \in N_4} \underbrace{\theta_{ij}(x_i, x_j)}_{\text{Pairwise term}}$$

Binary Label: $x_i \in \{0,1\}$

N_4 is the set of all neighboring pixels

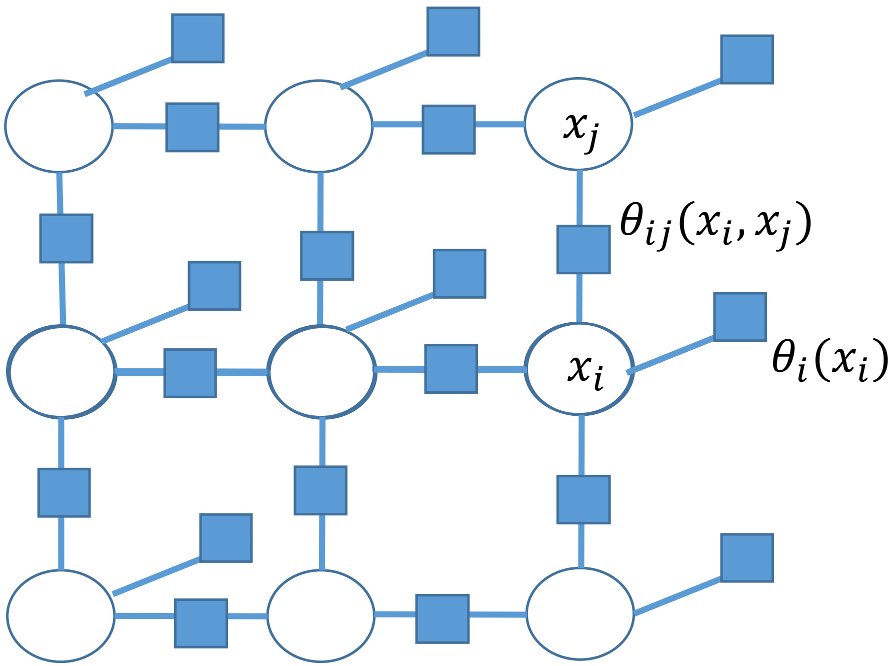
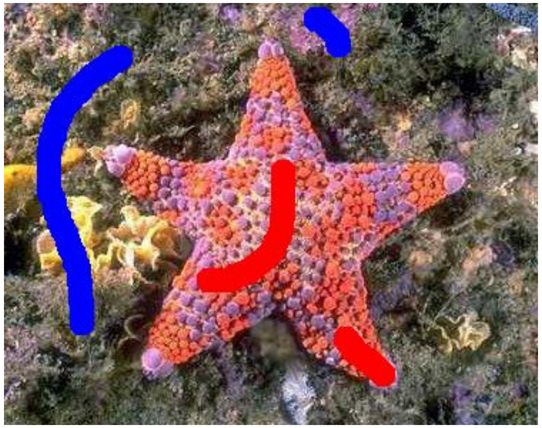


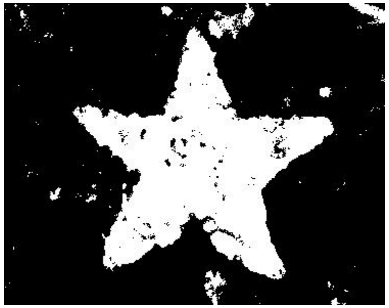
Image Segmentation: Energy



Goal: formulate $E(\boldsymbol{x})$ such that

$E(\boldsymbol{x}) = 0.01$

$E(\boldsymbol{x}) = 0.05$



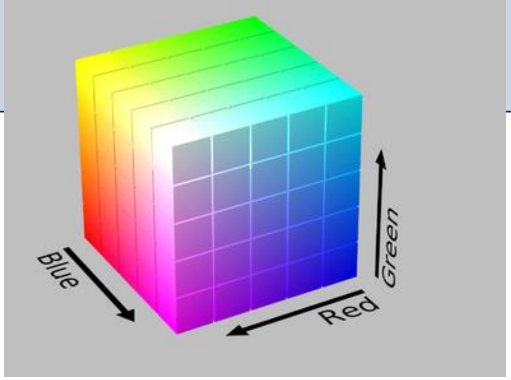
$E(\boldsymbol{x}) = 0.05$

$E(\boldsymbol{x}) = 0.1$

(numbers may not represent true numbers)

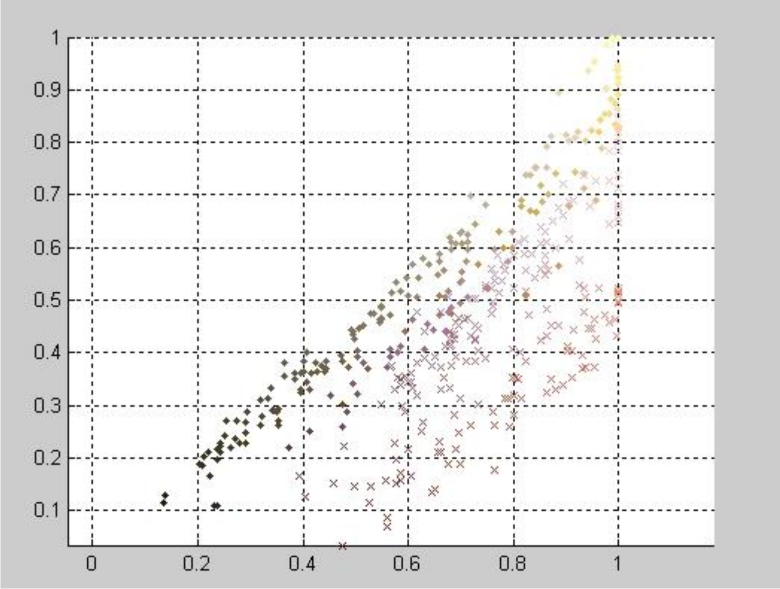
Solution: $\boldsymbol{x}^* = argmin_x \, E(\boldsymbol{x})$

Unary term



Gaussian Mixture Model fit

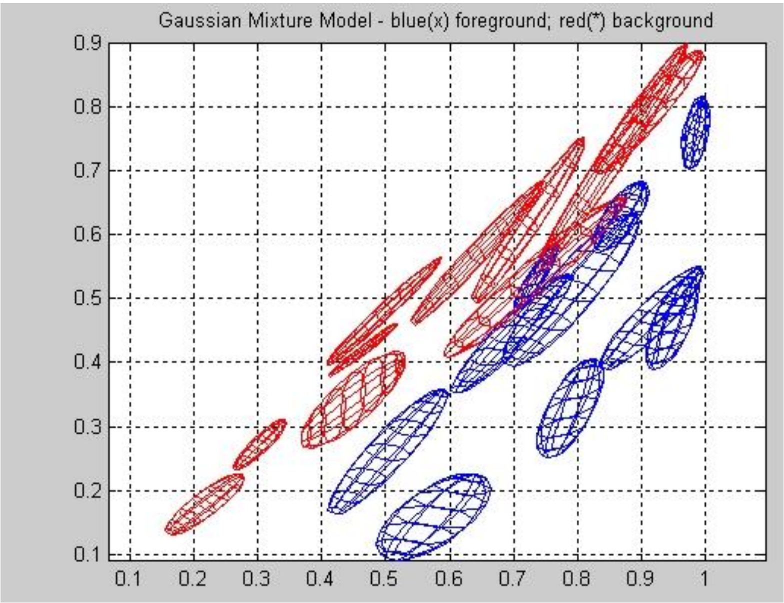
Green



Red

user labelled pixels
(cross foreground; dot background)

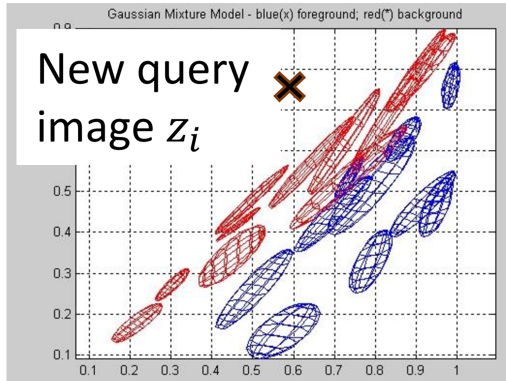
Green



Red

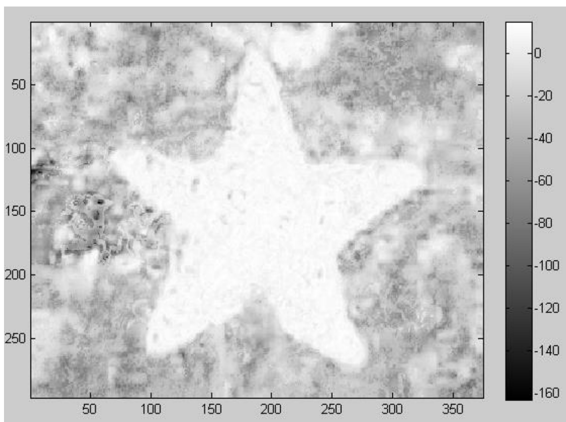
Foreground model is blue
Background model is red

Unary term



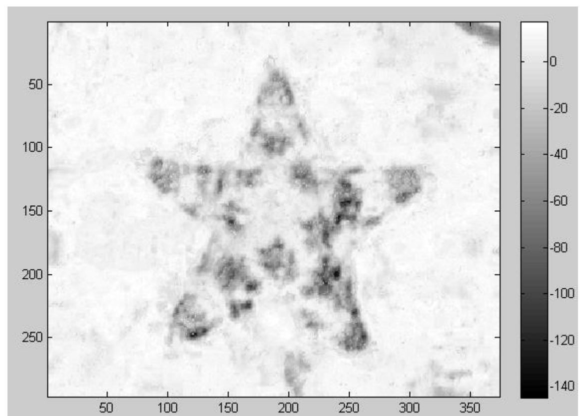
$\theta_i(x_i = 0) =$
$-\log P^{red}(z_i|x_i = 0)$

$\theta_i(x_i = 1) =$
$-\log P^{blue}(z_i|x_i = 1)$



$\theta_i(x_i = 0)$

Dark means likely background



$\theta_i(x_i = 1)$

Dark means likely foreground

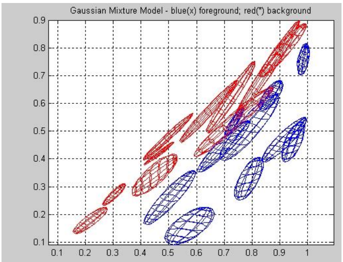
Optimum with unary terms only

$\boldsymbol{x}^* = argmin_x\, E(\boldsymbol{x})$

$E(\boldsymbol{x}) = \sum_i \theta_i(x_i)$

Gaussian Mixture Model (GMM)

- Mixture Model: $p(z) = \sum_{k=1}^{K} p(k)\ p(z|k)$



K = 8 for each fore- and background

- "k" is a latent variable we are not interested in

- $k \in \{1, \dots, K\}$ represents the K mixtures.

- Each mixture k is a 3D Gaussian distribution $N_k(z; \mu_k, \Sigma_k)$ where μ_k is a 3D vector and Σ_{k} a 3×3 matrix (positive-semidefinite), called covariance matrices:

$$N(z, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}\ |\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}\ (z-\mu)^T \Sigma^{-1} (z-\mu)\}$$

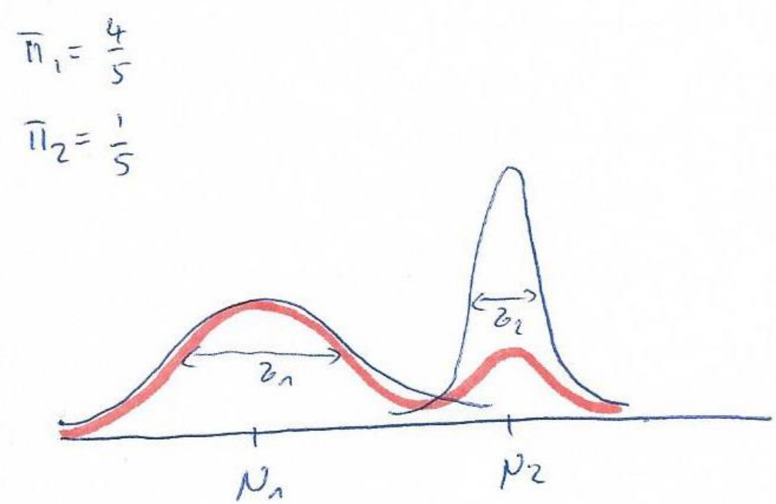
- $p(z) = \sum_{k=1}^{K} \pi_k\ N_k(z; \mu_k, \Sigma_k)$

Mixture coefficient

Gaussian Mixture Model (GMM)

- GMM probability $p(z) = \sum_{k=1}^{K} \pi_k \; N_k(z; \mu_k, \Sigma_k)$
- Reminder $\underbrace{\int}_{\text{"RGB cube"}} p(z) = 1$
- Unknown parameters: $\Theta = (\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$

Example:



GMM : $p(x) = \frac{4}{5} N_1(x; \mu_1, \Sigma_1) + \frac{1}{5} N_2(x; \mu_2, \Sigma_2)$

Note, $\int(\frac{4}{5} N_1(x) + \frac{1}{5} N_2(x)) = \frac{4}{5} \int N_1(x) + \frac{1}{5} \int N_2(x) = \frac{4}{5} + \frac{1}{5} = 1$

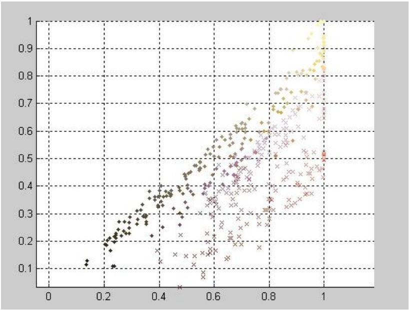
Fitting/Learning Gaussian Mixture Model (GMM)

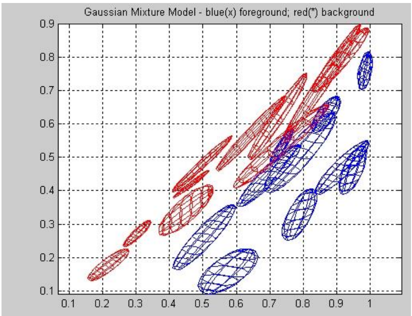
- GMM probability $p(z) = \sum_{k=1}^{K} \pi_k \ N_k(z; \mu_k, \Sigma_k)$
- Unknown parameters: $\Theta = (\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$
- How to learn Θ given data $\{z_1, \dots, z_n\}$:
  - Maximum Likelihood Learning objective:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \prod_{i=1}^{n} p_\Theta \left(z_i\right)$$

    where z_i are all pixels to which the GMM is fitted to

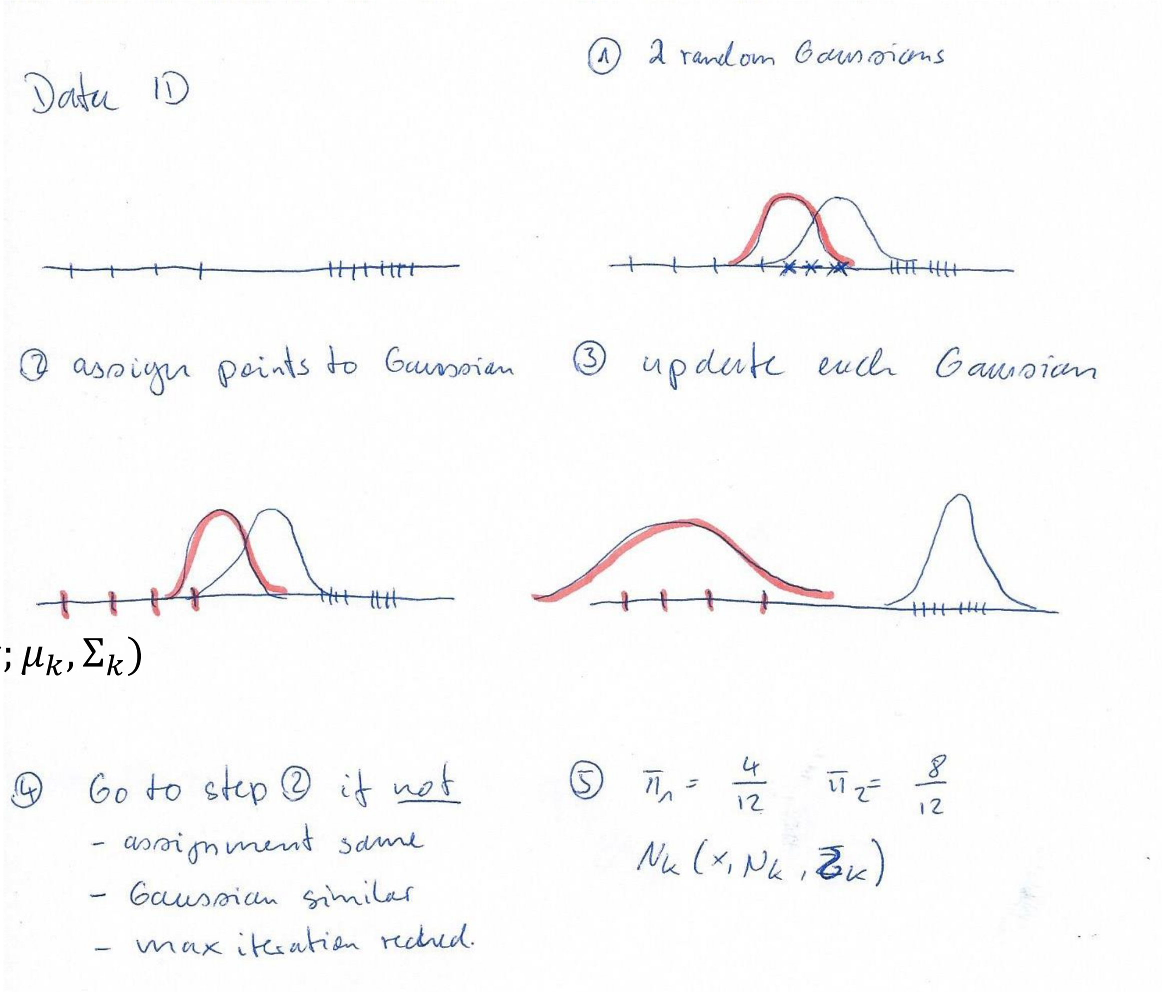
  - Full learning procedure: EM (see machine learning lecture ML 1)
  - Next is a simplified version





A simple procedure for GMM learning /fitting

Let us introduce an assignment variable for each data point (pixel) to which Gaussian it belongs to: $k_1, \ldots k_n$ where $k_i \in \{1, \ldots K\}$



$k_i = argmax_k N_k(z; \mu_k, \Sigma_k)$

Extensions

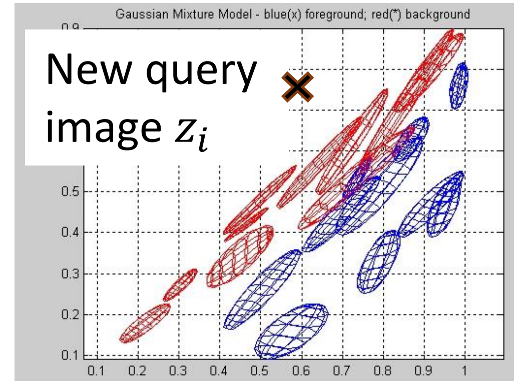
- Choose K automatically

- Go to probabilistic version using Expectation Maximization (EM). Now k_i are "soft assignments" to all Gaussian

- Faster versions:
  - Fit GMM to all data points (fore- and background) and then only change the mixture coefficients
  - Use Histograms instead of GMMs

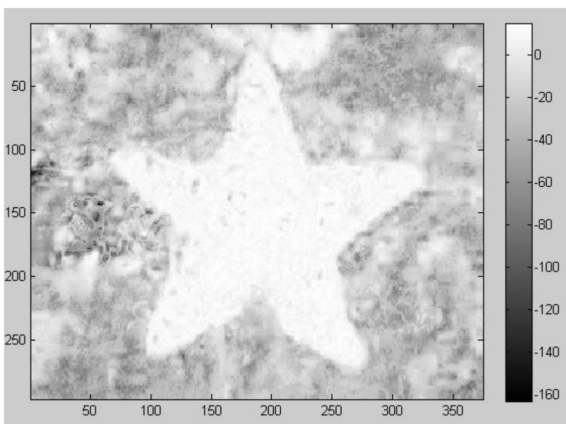
So far: Unary term

$$E(\boldsymbol{x}) = \sum_{i\in N} \underbrace{\theta_i(x_i)}_{\text{Unary term}} + \sum_{i,j\in N_4} \underbrace{\theta_{ij}(x_i,x_j)}_{\text{Pairwise term}}$$



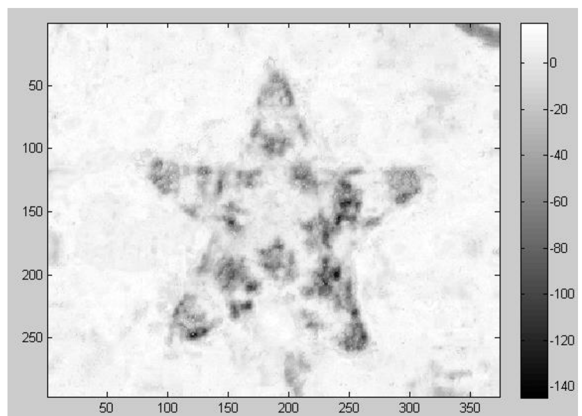
$$\theta_i(x_i = 1) = -\log P^{blue}(z_i | x_i = 1)$$

$$\theta_i(x_i = 0) = -\log P^{red}(z_i | x_i = 0)$$



$\theta_i(x_i = 0)$

Dark means likely background



$\theta_i(x_i = 1)$

Dark means likely foreground

Optimum with unary terms only

$$\boldsymbol{x}^* = argmin_x\, E(\boldsymbol{x})$$

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i)$$

Pairwise term

- We choose a so-called Ising Prior:

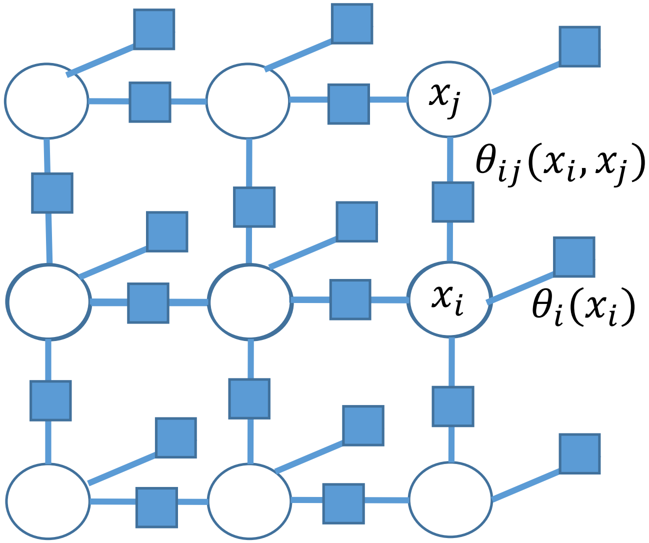
$$\theta_{ij}(x_i, x_j) = |x_i - x_j|$$

- Full Energy

$$E(\boldsymbol{x}) = \sum_{i \in N} \underbrace{\theta_i(x_i)}_{\text{Unary term}} + \sum_{i,j \in N_4} \underbrace{|x_i - x_j|}_{\text{Pairwise term}}$$

$$\theta_i(x_i = 1) = -\log P^{blue}(z_i | x_i = 1)$$

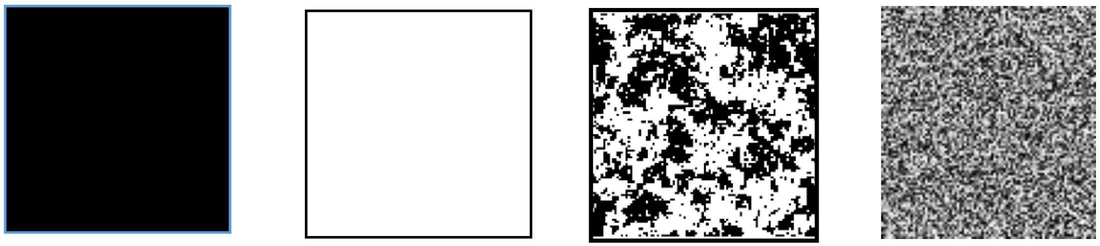
$$\theta_i(x_i = 0) = -\log P^{red}(z_i | x_i = 0)$$



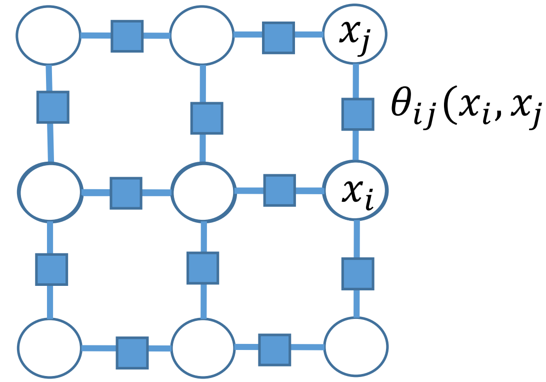
Question

Question: Given the energy $E(\boldsymbol{x}) = \sum_{i,j \in N_4} |x_i - x_j|$ with $x \in \{0,1\}$

Which labelling has lowest energy?



Solution A Solution B Solution C Solution D



Possible Answers:

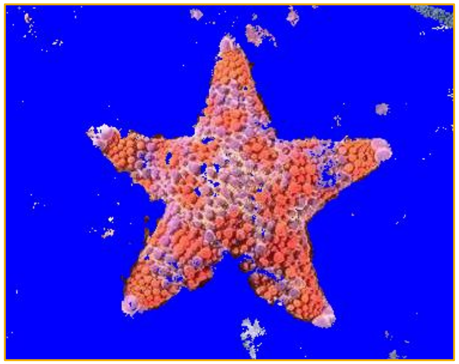
1) Solution A
2) Solution B
3) Solution A and B
4) Solution C
5) Solution D
6) I don't know

This models makes the assumption that the object is spatially coherent

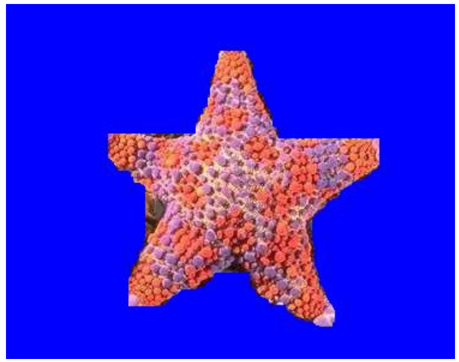
Question

<u>Question:</u> Given the energy $E(\boldsymbol{x}) = \sum_i \theta_i(x_i) \; + \omega \; \sum_{i,j \in N_4} |x_i - x_j| \, ; x \in \{0,1\}$

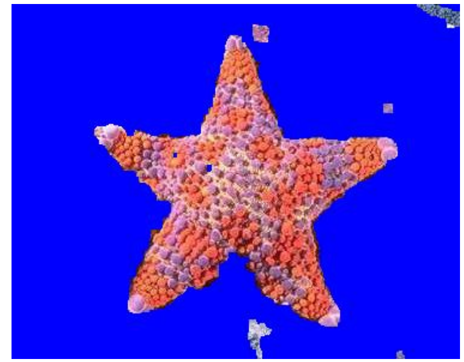
Please guess what ω_1, ω_2 could be?



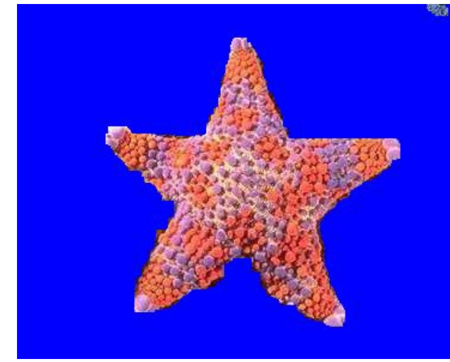
$\omega \; = 0$



$\omega \; = 200$



$\omega_1 \; = ?$



$\omega_2 \; = ?$

<u>Possible Answers:</u>

1) $\omega_1 = 10; \; \omega_2 = 40$
2) $\omega_1 = \omega_2$
3) $\omega_1 = 30; \; \omega_2 = 20$
4) I don't know

Adding Unary and Pairwise term

$\omega \ = 0$

$\omega \ = 10$

$\omega \ = 40$

$\omega \ = 200$

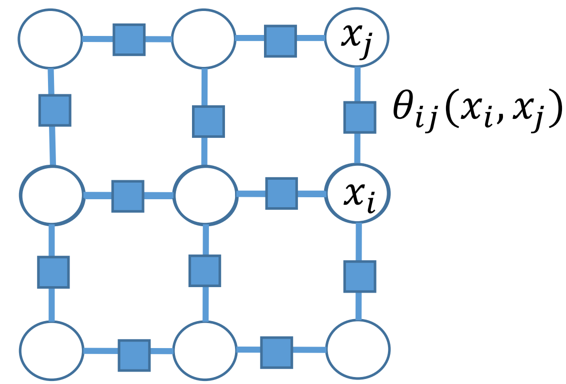
Energy: $E(\boldsymbol{x}) = \sum_i \theta_i(x_i) \ + \omega \ \sum_{i,j \in N_4} |x_i - x_j|$

Question: Given the energy $E(\boldsymbol{x}) = \sum_{i,j \in N_4} |x_i - x_j|$ with $x \in \{0,1\}$.
Given the 5x5 pixel image below, where several pixels have been assigned a labelling and others not. You have to fill in the remaining labels.
How many corners has the labelling with minimal energy?

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

1	0
1	0

Example: segmentation with 4 corners



Possible Answers:

1) 4 corners
2) 6 corners
3) 10 corners
4) There is no unique answer
5) I don't know

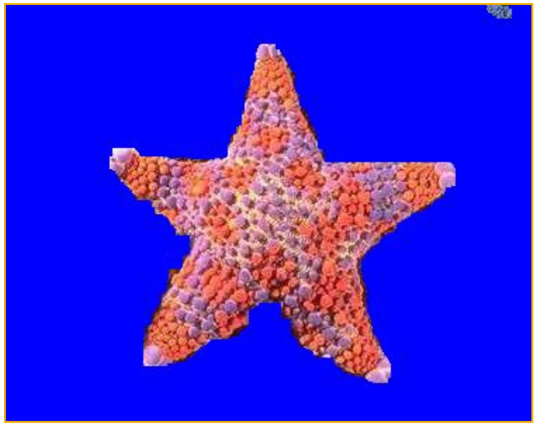
0	0	0	0	0
1	1	1	1	0
1	1	1	1	0
1	1	1	1	0
1	1	1	1	0

4 corners, 8 edges cut

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

10 corners, 8 edges cut

Is it the best we can do?



4-connected segmentation

zoom

zoom

Zoom-in on image

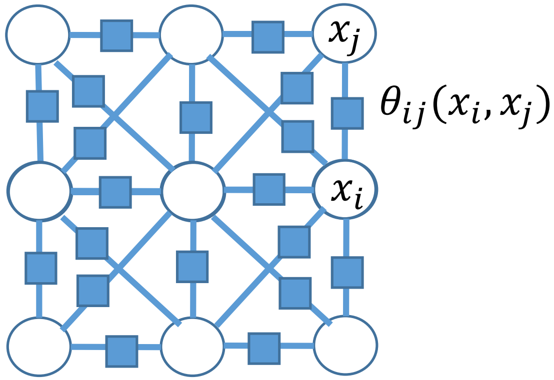
Energy: $E(\boldsymbol{x}) = \sum_i \theta_i(x_i) \; + \sum_{i,j \in N_4} |x_i - x_j|$

<u>Question:</u> Given the energy $E(\boldsymbol{x}) = \sum_{i,j \in N_8} |x_i - x_j|$ with $x \in \{0,1\}$.
Given the 5x5 pixel image below, where several pixels have been assigned a labelling and others not. You have to fill in the remaining labels.
How many corners has the labelling with minimal energy?

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

1	0
1	0

Example: segmentation with 4 corners



<u>Possible Answers:</u>

1) 4 corners
2) 6 corners
3) 10 corners
4) There is no unique answer
5) I don't know

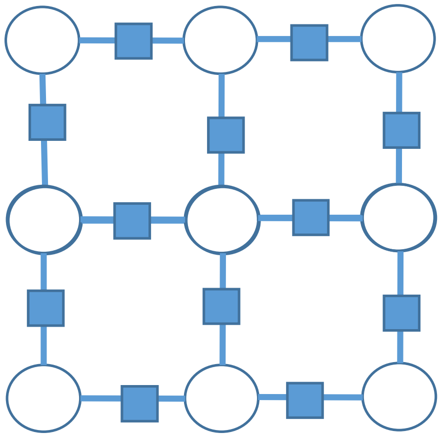
0	0	0	0	0
1	1	1	1	0
1	1	1	1	0
1	1	1	1	0
1	1	1	1	0

4 corners, 8+13 edges cut

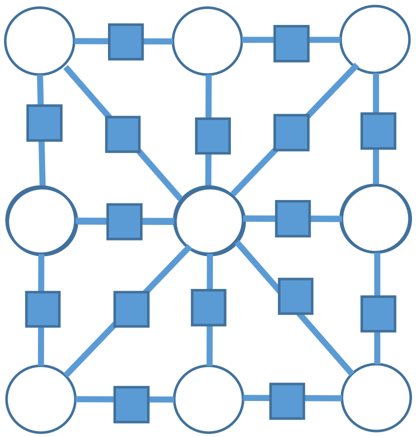
0	0	0	0	0
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0

10 corners, 8+7 edges cut

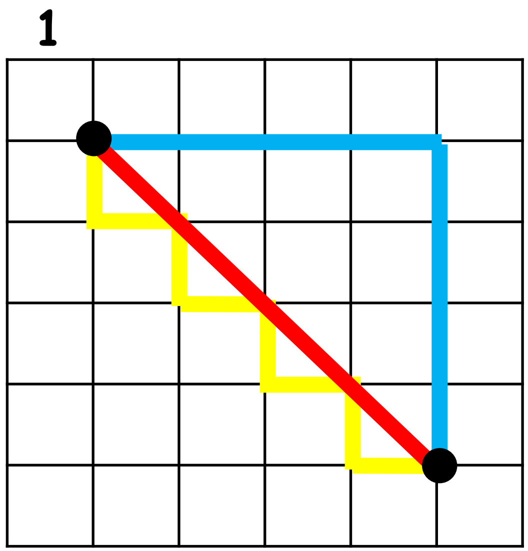
From 4-connected to 8-connected Factor Graph



4-connected



8-connected



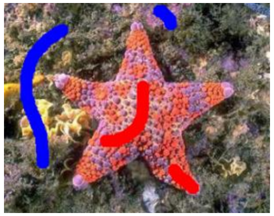
Length of the paths:

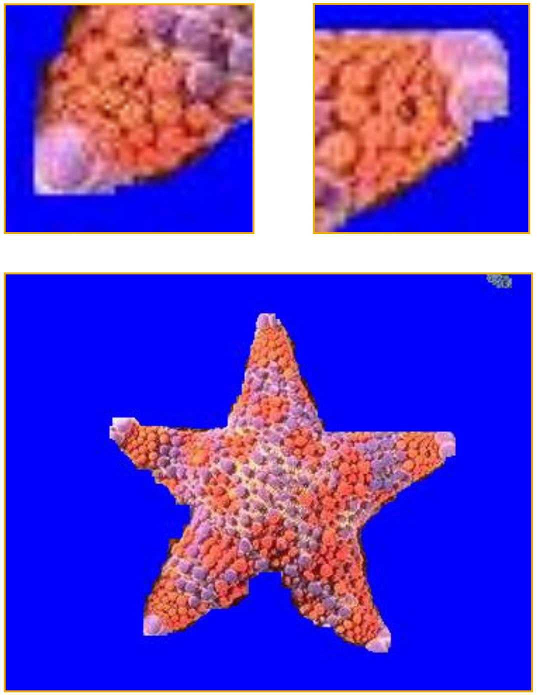
	Eucl.	4-con.	8-con.
	5.65	6.28	5.08
	8	6.28	6.75

Larger connectivity can model true Euclidean length (also other metric possible)

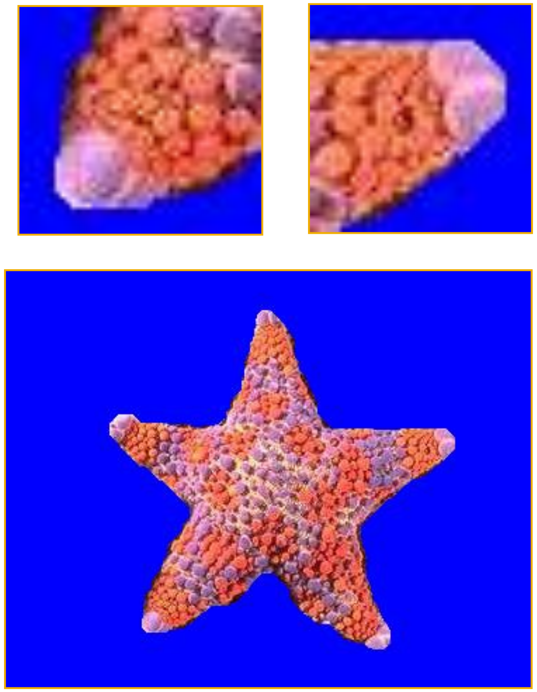
[Boykov et al. '03; '05]

Going to 8-connectivty





4-connected
Euclidean



8-connected
Euclidean (MRF)

Zoom-in image

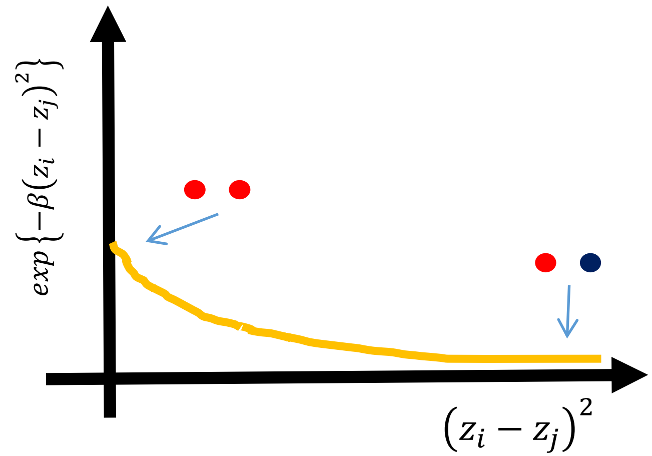
Is it the best we can do?

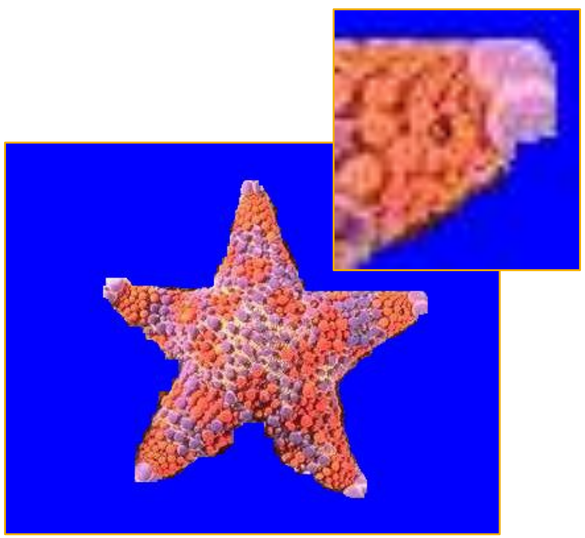
Adapt the pairwise term

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j)$$

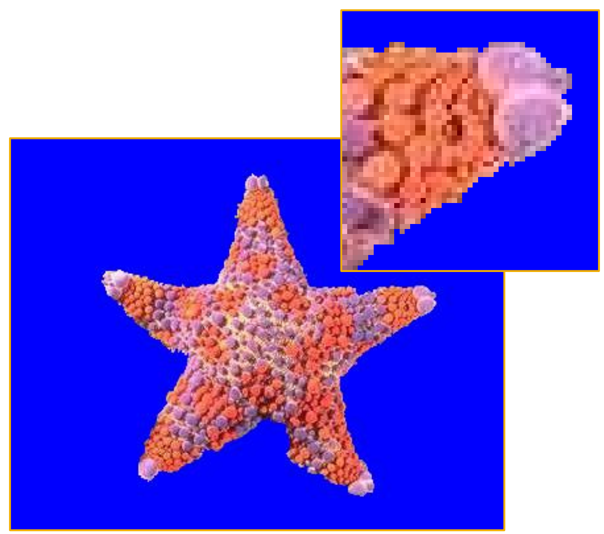
$$\theta_{ij}(x_i, x_j) = |x_i - x_j| \left(exp\left\{-\beta\left(z_i - z_j\right)^2\right\}\right)$$

where $\beta \geq 0$ is a constant

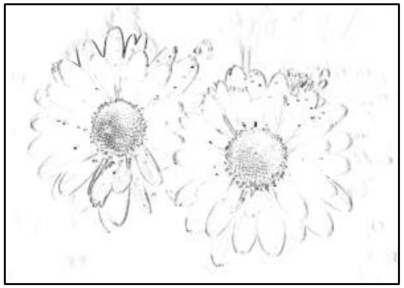




Standard 4-connected



Edge-dependent 4-connected



Optimization

- The defined Energy can be solved globally optimally with graph cut since submodularity condition is satisfied

Energy: $E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j)$

$$\theta_{ij}(x_i, x_j) = |x_i - x_j|(exp\left\{-\beta(z_i - z_j)^2\right\})$$

where $\beta \geq 0$ is a constant

- Submodularity condition:

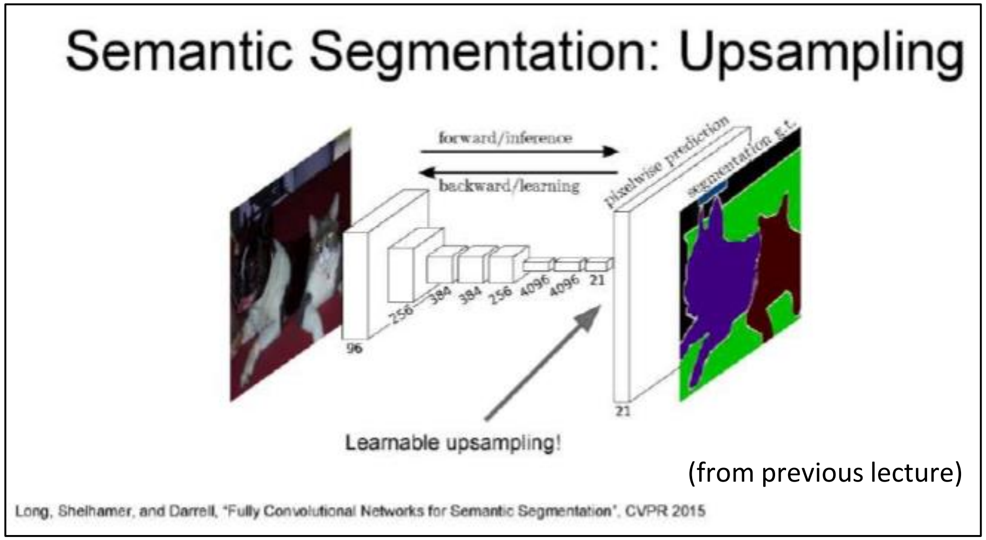
  for all i, j it is: $\theta_{ij}(1{,}0) + \theta_{ij}(0{,}1) \geq \theta_{ij}(0{,}0) + \theta_{ij}(1{,}1)$

A simple semantic segmentation system

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i,j \in N_4} |x_i - x_j| \left(exp\{-\beta (z_i - z_j)^2\} \right)$$

where $\beta \geq 0$ is a constant

Unaries are from a fully convolutional Neural Network:



Optimization is done with alpha expansion:

Image | Dense CNN | Dense CNN with MRF

Extension: Fully connected CRFs

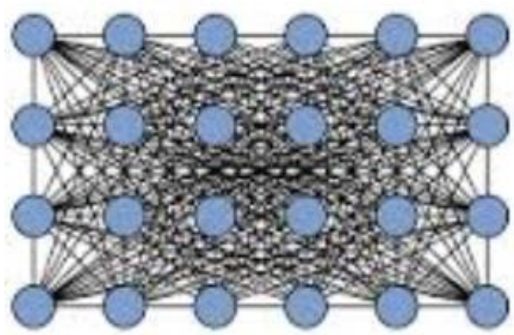
[Krähenbühl, Koltun, NIPS 2011]

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{i<j} w_{ij}\, |x_i - x_j|$$

$$w_{ij} = \exp\left\{-\left|p_i - p_j\right|^2/\lambda_1\right\} + \exp\left\{-\left|p_i - p_j\right|^2/\lambda_2 - \left|I_i - I_j\right|^2/\lambda_3\right\}$$

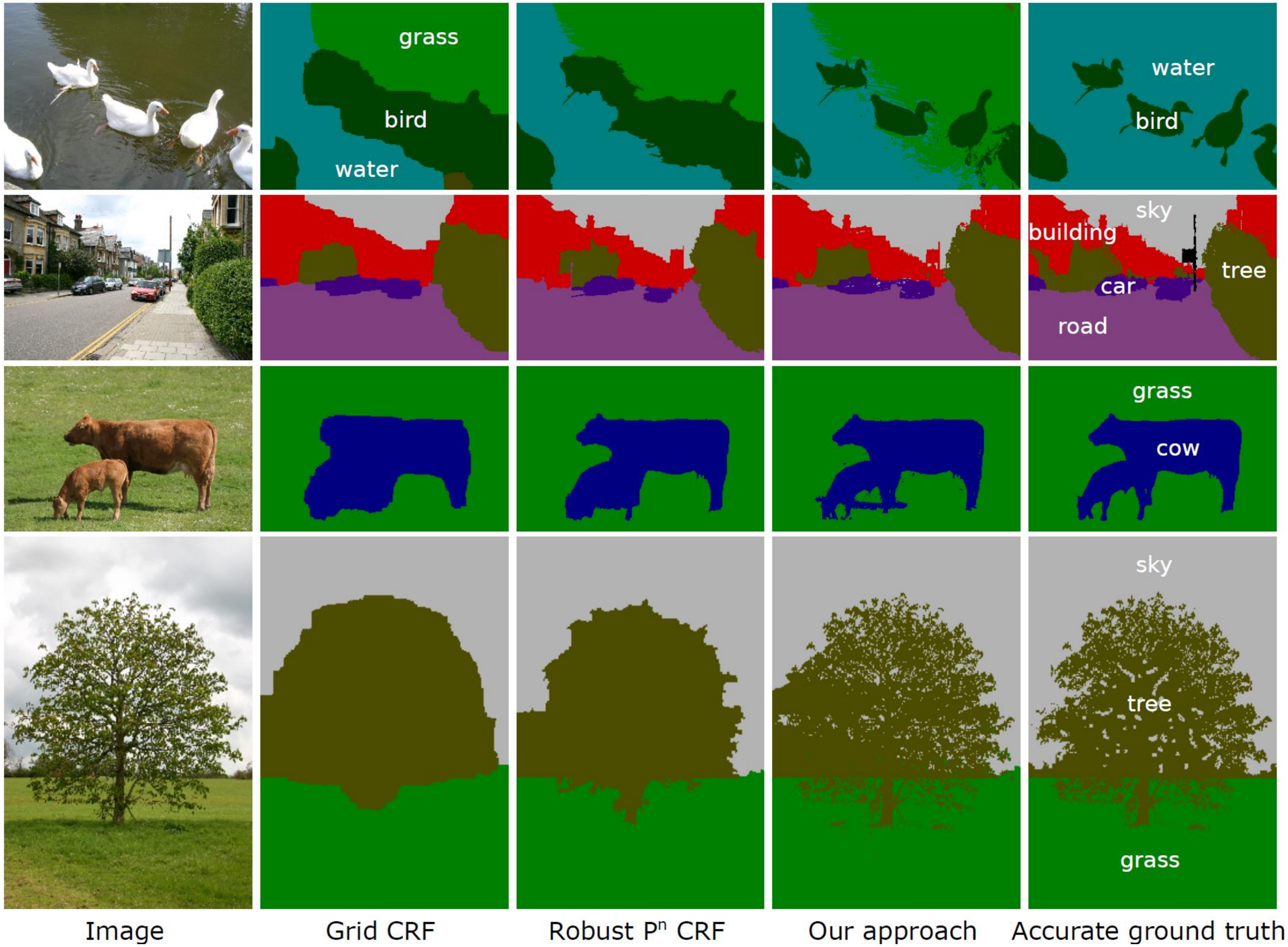
Spatial distance contrast-dependent

All pixels are connected:



Comment: Fully connected CRFs

[Krähenbühl, Koltun, NIPS 2011]



- Optimization is only approximate
- The fully connected CRF can also be written as "unrolled inference" at the end of an fully convolutional neural network