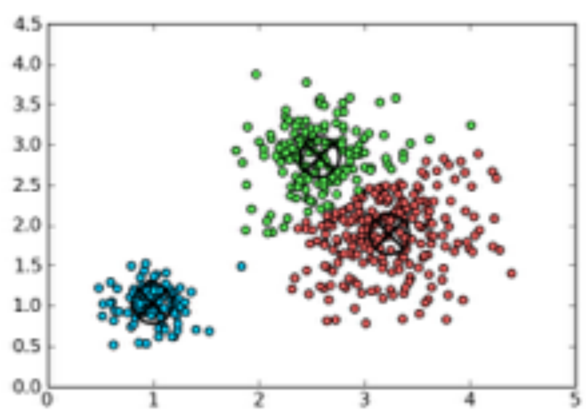
Grouping

Outline

- Gestault motivations
- Grouping as clustering (k-means, meanshift, agglomertive)
- Graph theoretic (graph cuts, normalized cuts)

Bird's eye view of grouping

Model-based clustering

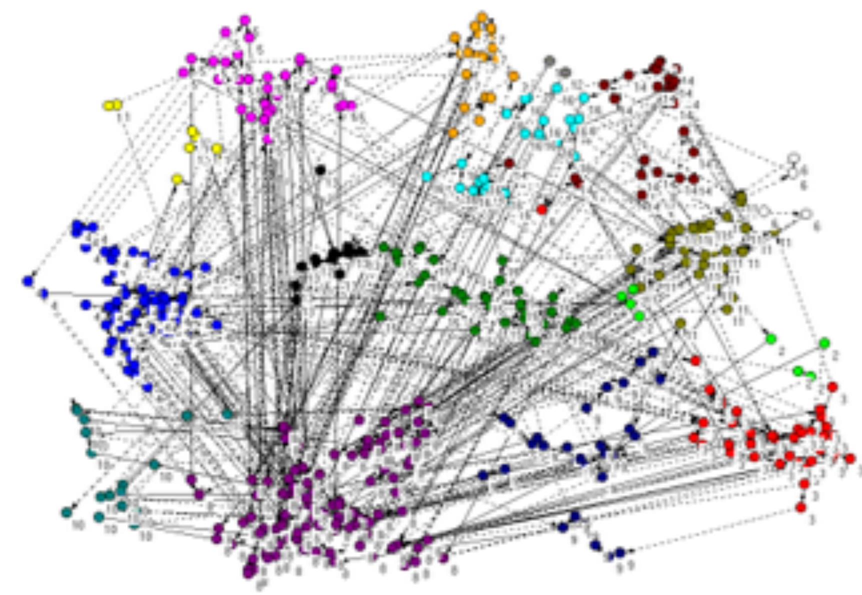


K-means
Sparse coding
Gaussian mixture models
....

Start by representing pixel as a vector

$$x_i \in R^N$$

Graph theoretic (pairwise)



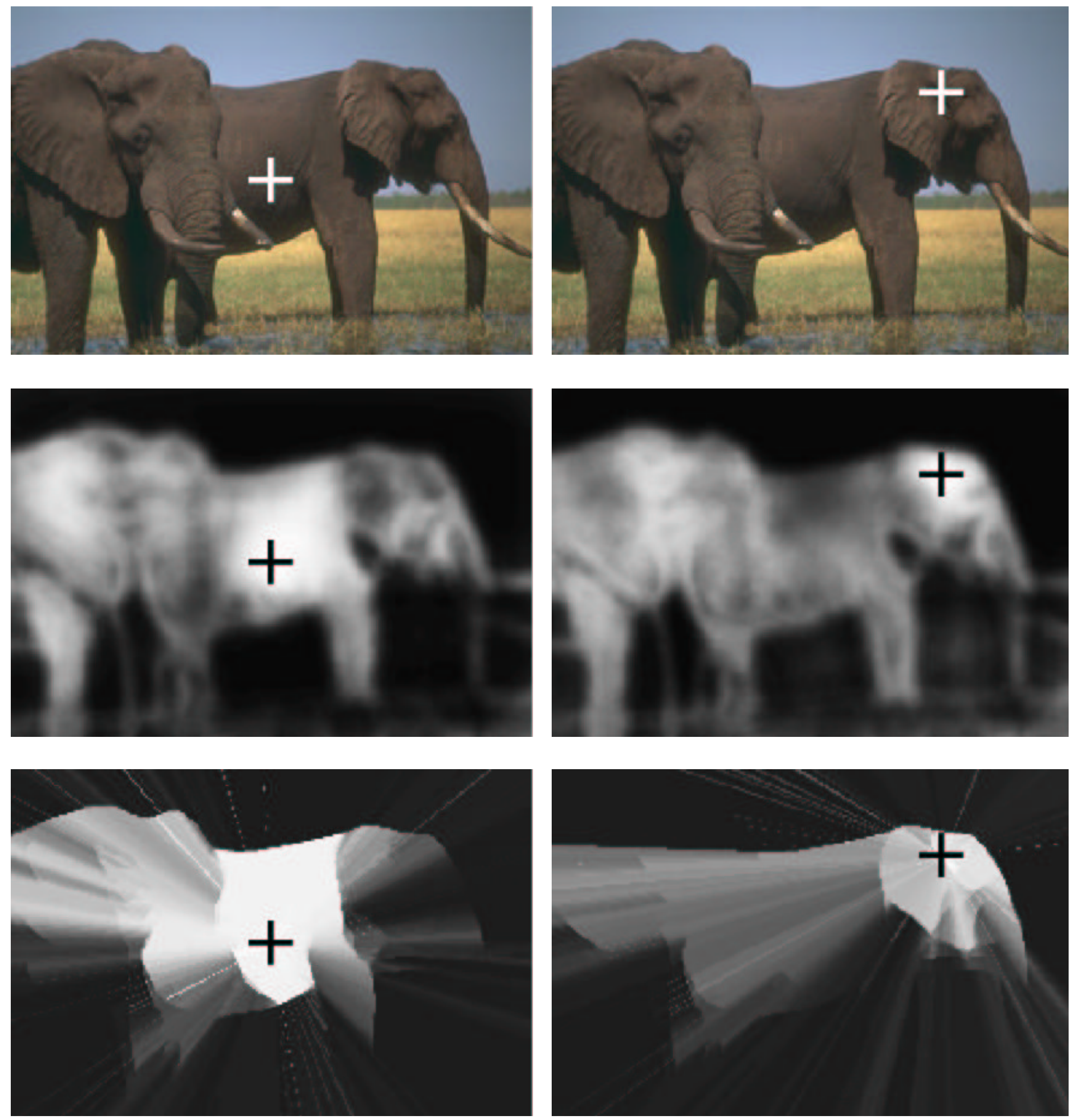
Graph cuts
Normalized cuts
Spectral clustering
....

Start by computing pairs of similarities/affinities between pixels

$$W_{ij} = e^{-||x_i - x_j||^2}$$

We want vector embedding or similarity to capture gestalt cues

Flexibility of a similarity matrix



“Intervening contour cue”:
two pixels are similar if there exists no strong edge between them

K-means

$$x_i = \begin{bmatrix} r_i \\ b_i \\ g_i \end{bmatrix}$$

K-means using color alone, 11 segments.

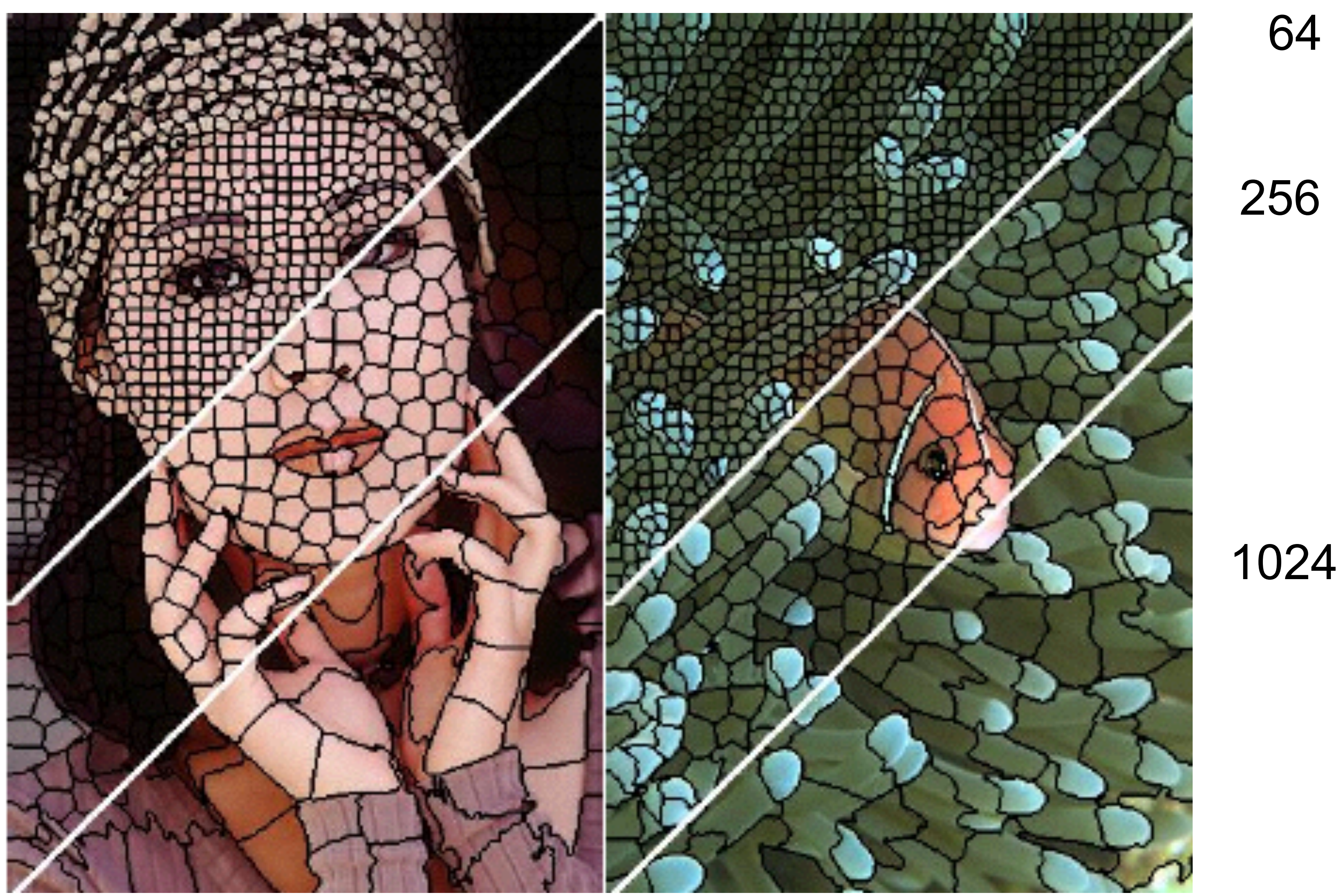
Aside: Given N points, what is complexity of clustering?

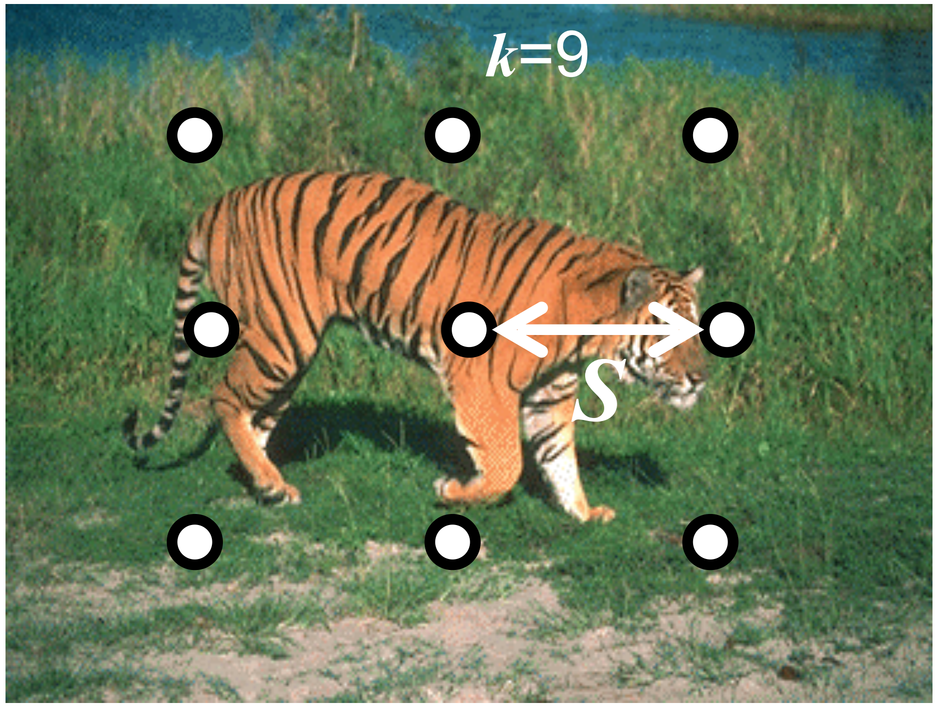



$$x_i = \begin{bmatrix} r_i \\ b_i \\ g_i \\ u_i \\ v_i \end{bmatrix}$$


SLIC superpixels

- Find k superpixels such that
  - Respect boundaries
  - Are spatially roughly equally sized





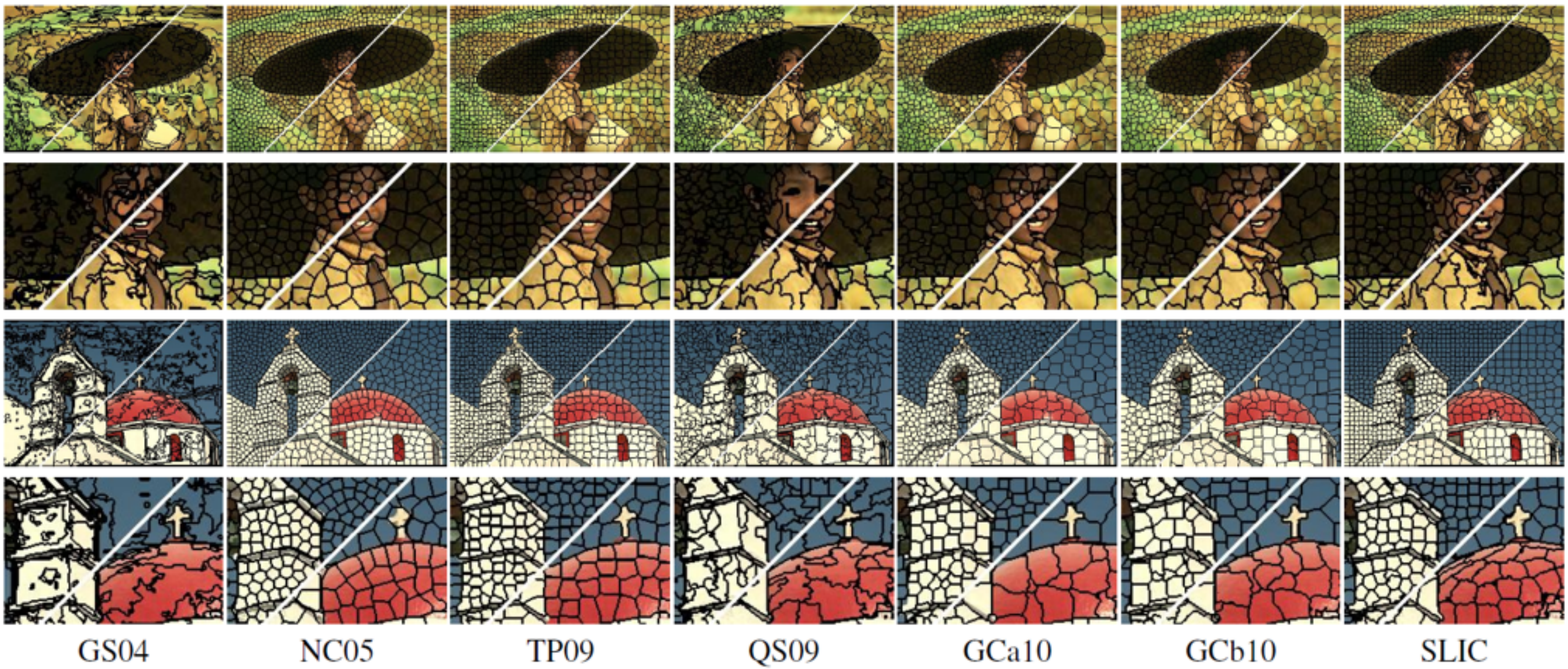
Select $\{\mathbf{c}_j\}$ on a regular grid of k centers

0. Initialize cluster centers $\{\mathbf{c}_j\}$

1. Given $\{\mathbf{c}_j\}$, assign each $\mathbf{x}_i$ to the closest j
2. Compute the centers $\mathbf{c}_j = 1/N \, \Sigma \, \mathbf{x}_i$

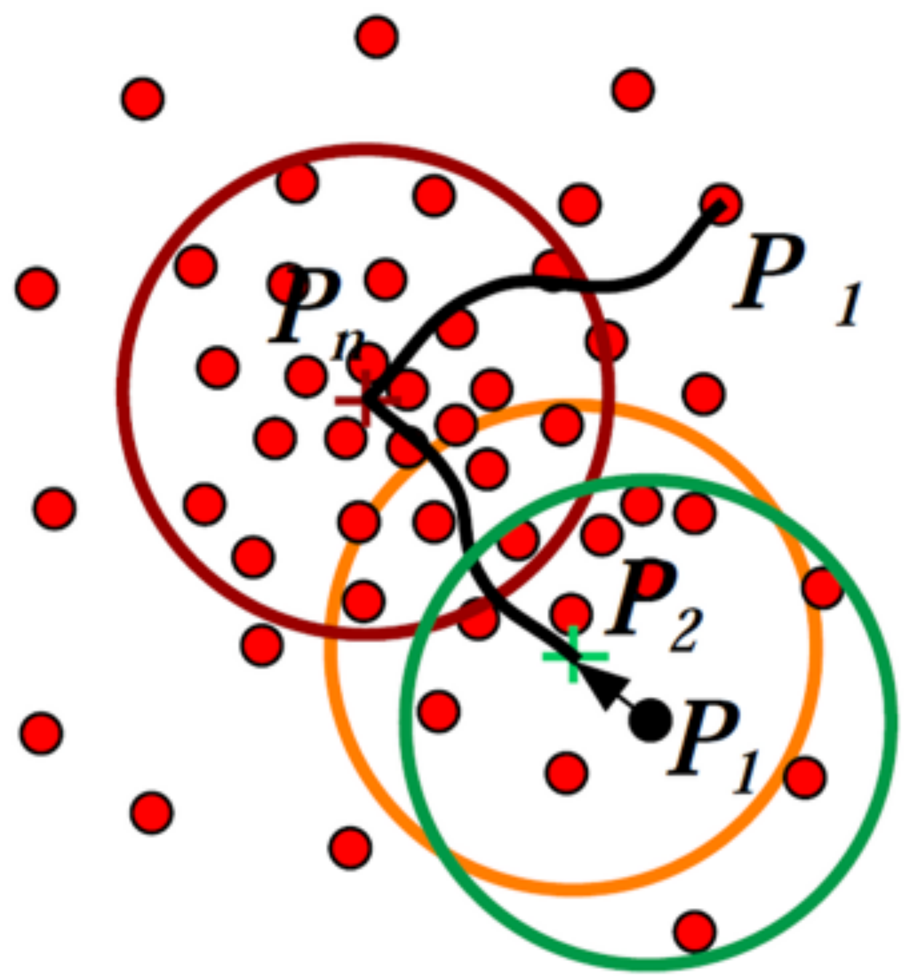
Search only in a neighborhood $2S$x$2S$ neighborhood around i

http://ivrg.epfl.ch/supplementary_material/RK_SLICSuperpixels/index.html



Meanshift clustering

Insufficiently well-known clustering algorithm: mean shift clustering

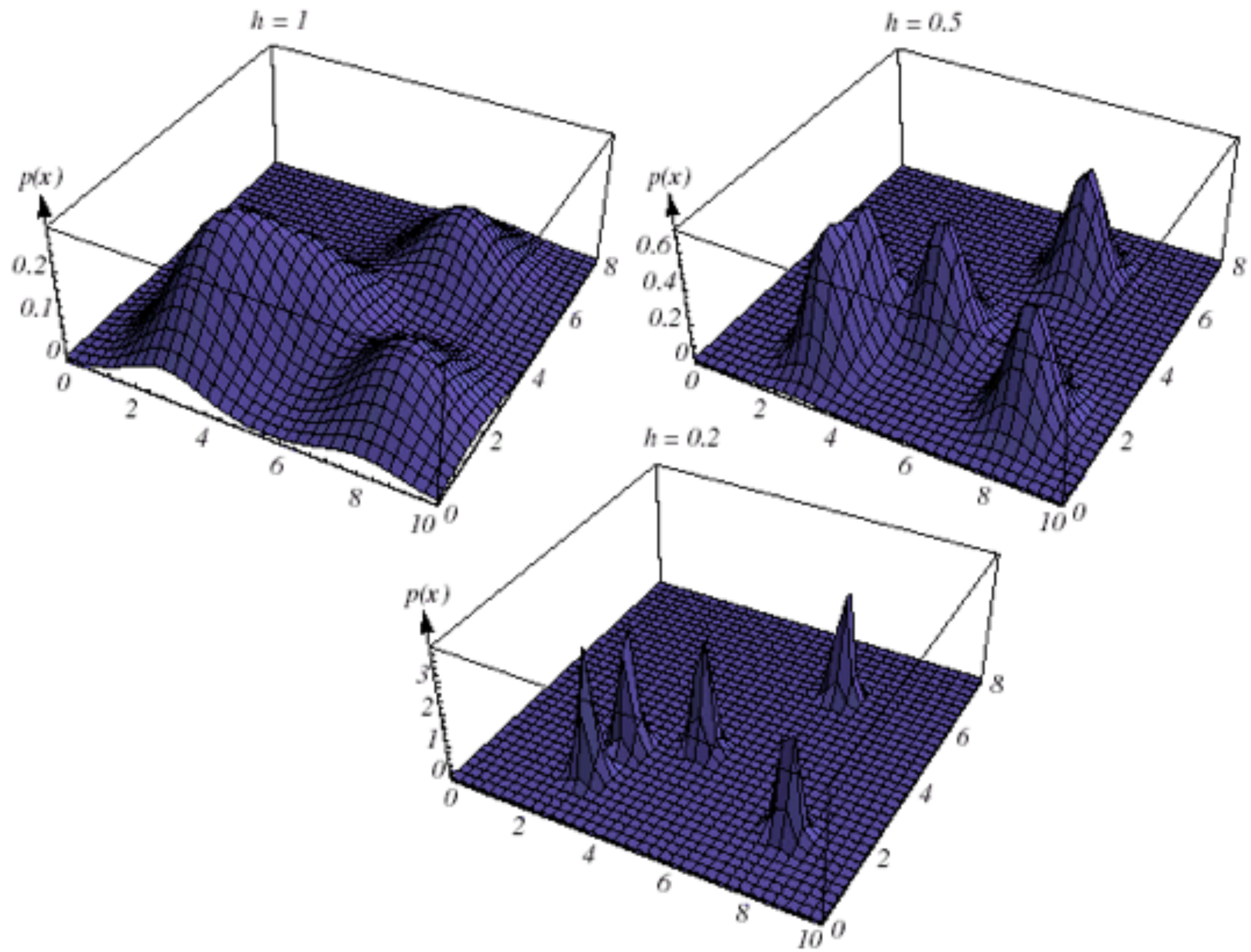


To find cluster center for P1:

Repeatedly find centroid of points in a sphere (init @ P1) and recenter at centroid until convergence

Parzen's window estimate

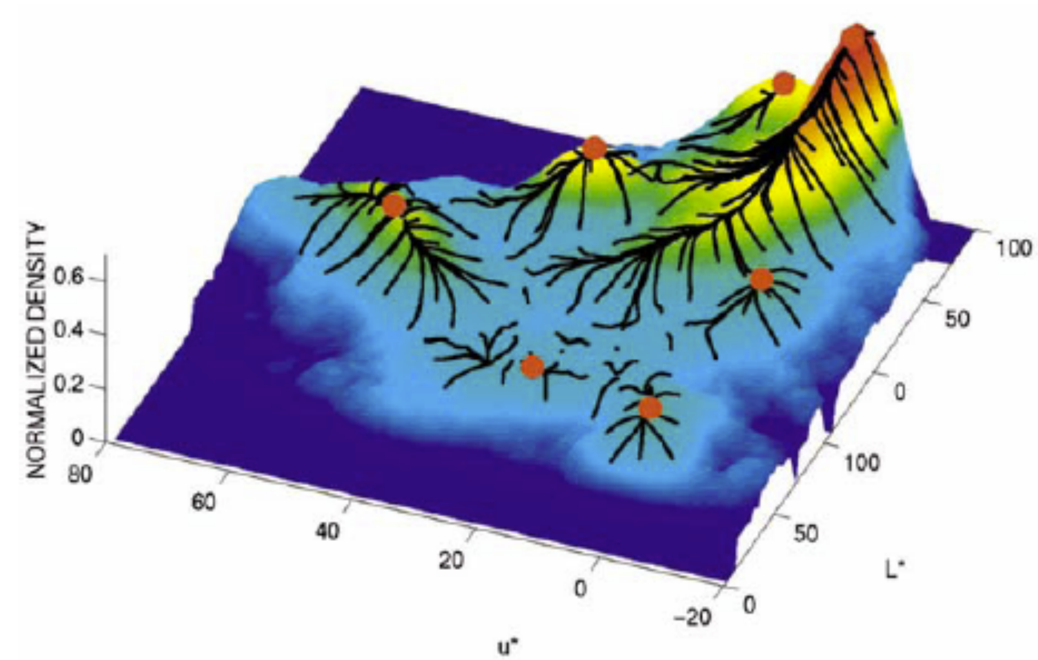
Construct probability density estimate out of data samples $\{x_i\}$

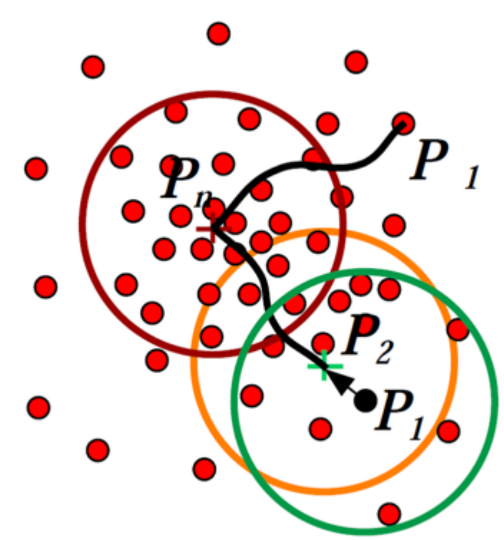


$$p(x) \propto \sum_i K(x - x_i)$$

Meanshift as mode-finding

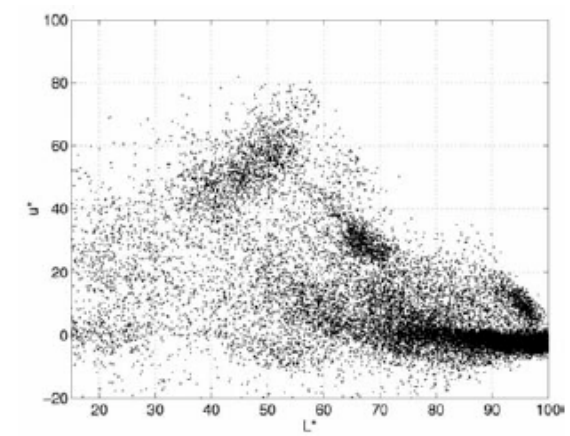
Meanshift steps perform gradient ascent on this probability model!

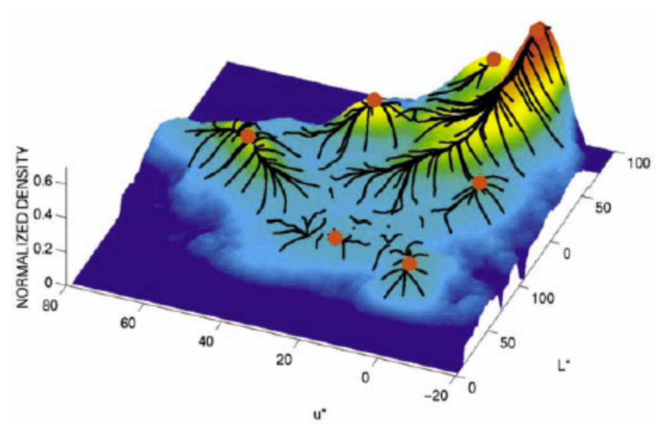


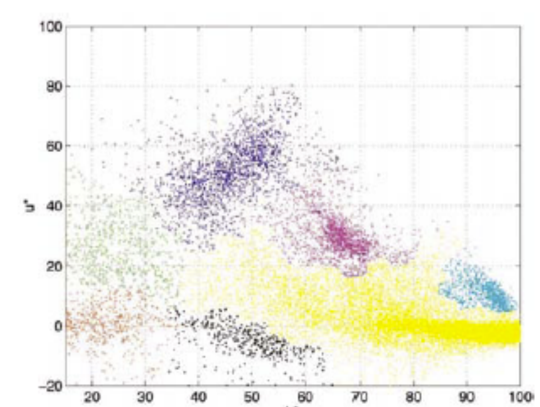


Meanshift clustering

Start mode-finding from each point
Label all points that converge to same point as one cluster



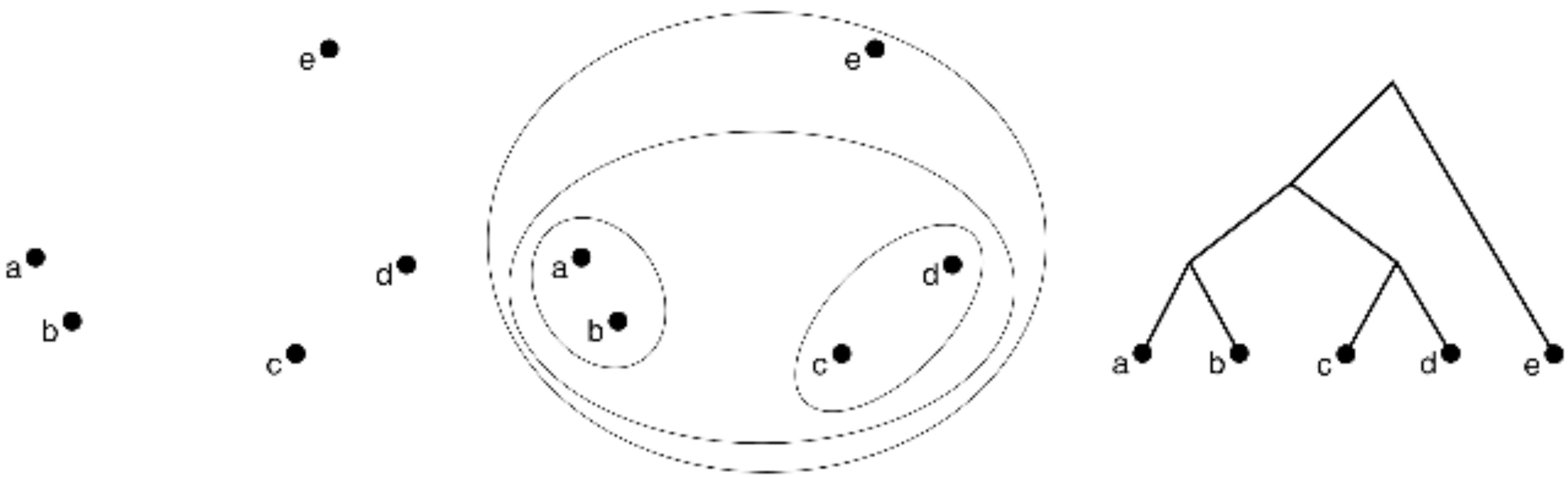




What is complexity of clustering?
How does one perform model selection (tune K)?

Agglomerative clustering

http://en.wikipedia.org/wiki/Hierarchical_clustering



(a) Data points (b) Hierarchical clusters (c) Dendrogram

Algorithm 15.3: Agglomerative clustering, or clustering by merging

Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the
      smallest inter-cluster distance
end

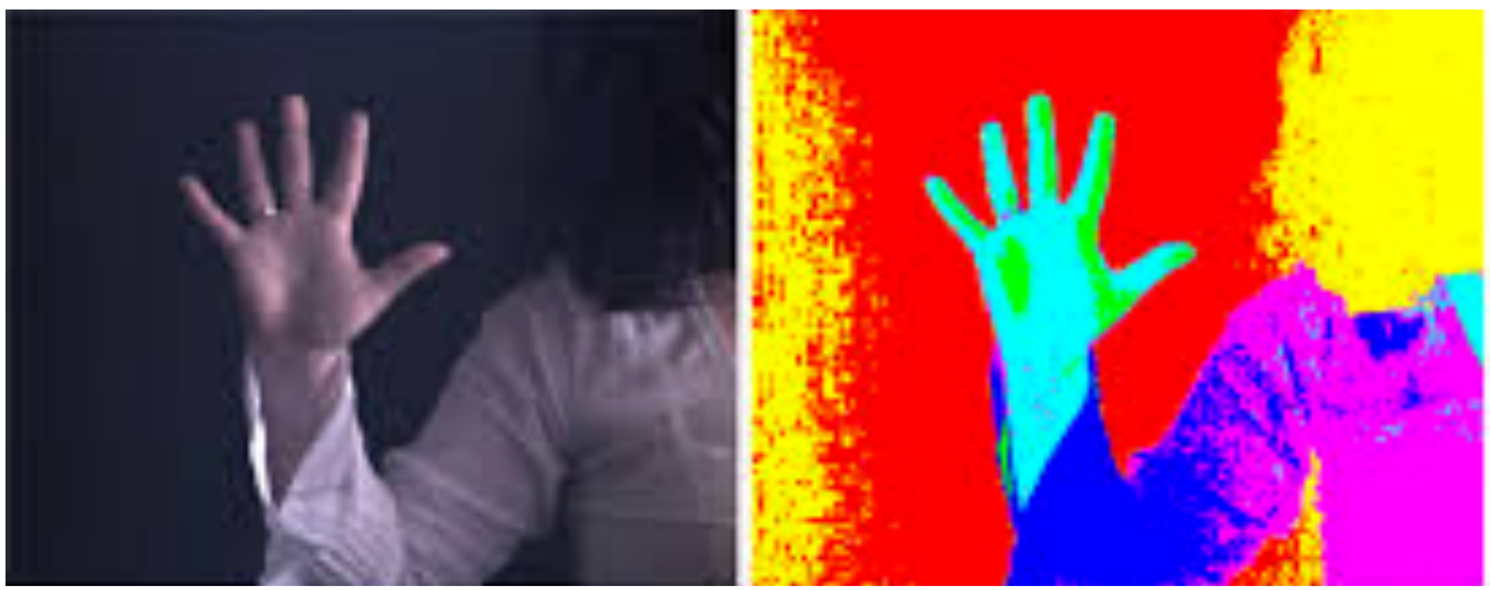
Agglomerative clustering

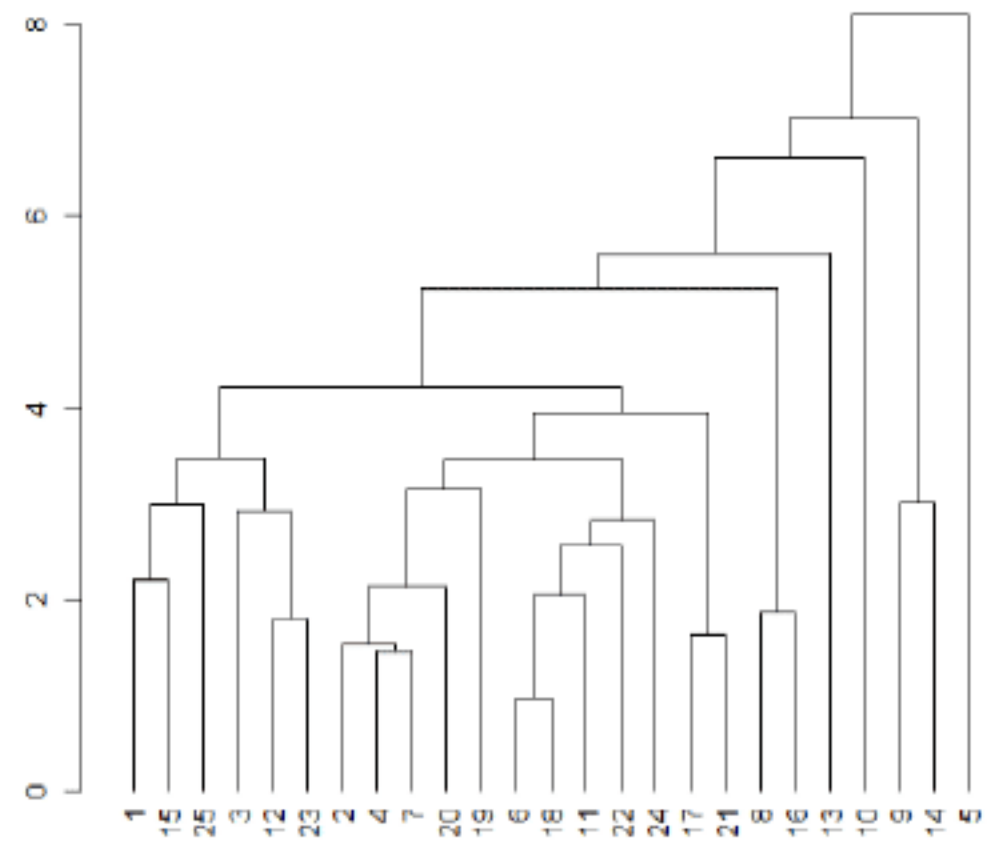
Strategies for measuring distance between two clusters

Names	Formula
Maximum or complete linkage clustering	$\max \{ d(a,b) : a \in A,\, b \in B \}.$
Minimum or single-linkage clustering	$\min \{ d(a,b) : a \in A,\, b \in B \}.$
Mean or average linkage clustering, or UPGMA	$\frac{1}{\|A\|\|B\|} \sum_{a \in A} \sum_{b \in B} d(a,b).$
Centroid linkage clustering, or UPGMC	$\|\|c_s - c_t\|\|$ where c_s and c_t are the centroids of clusters *s* and *t*, respectively.

Which requires a vector embedding?
Which requires a similarity?

Sample results



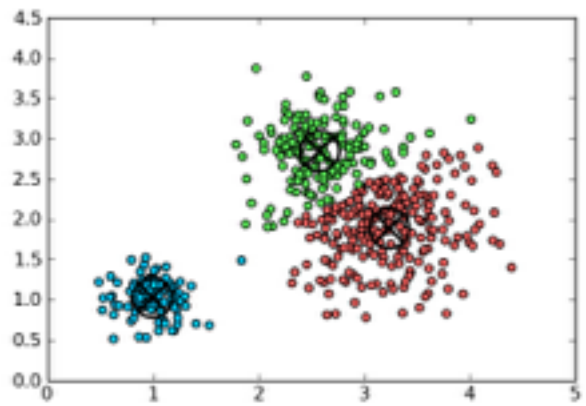


Seems nice; simple and we get a “multi-scale” clustering

Why doesn’t this solve the problem?

Outline

Model-based clustering

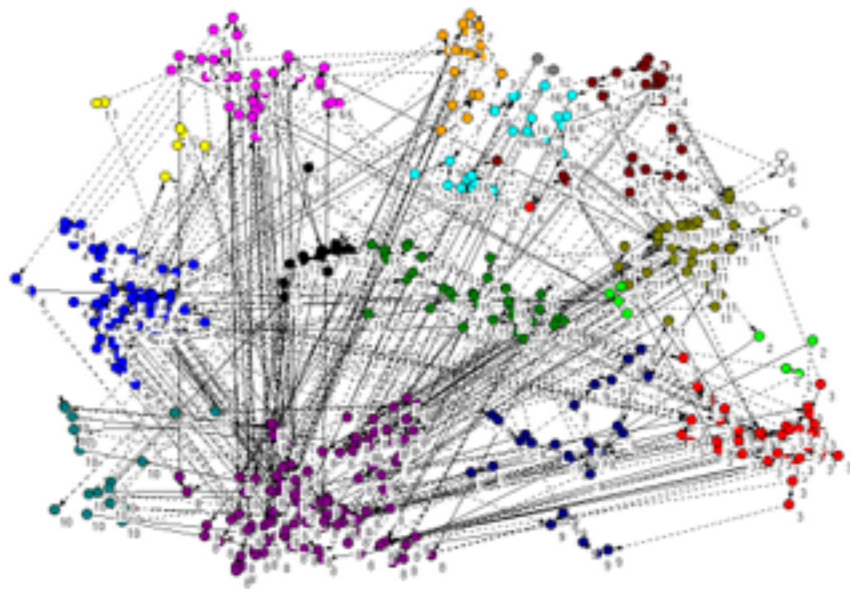


K-means
Sparse coding
Gaussian mixture models
....

Start by representing pixel as a vector

$$x_i \in R^N$$

Graph theoretic (pairwise)



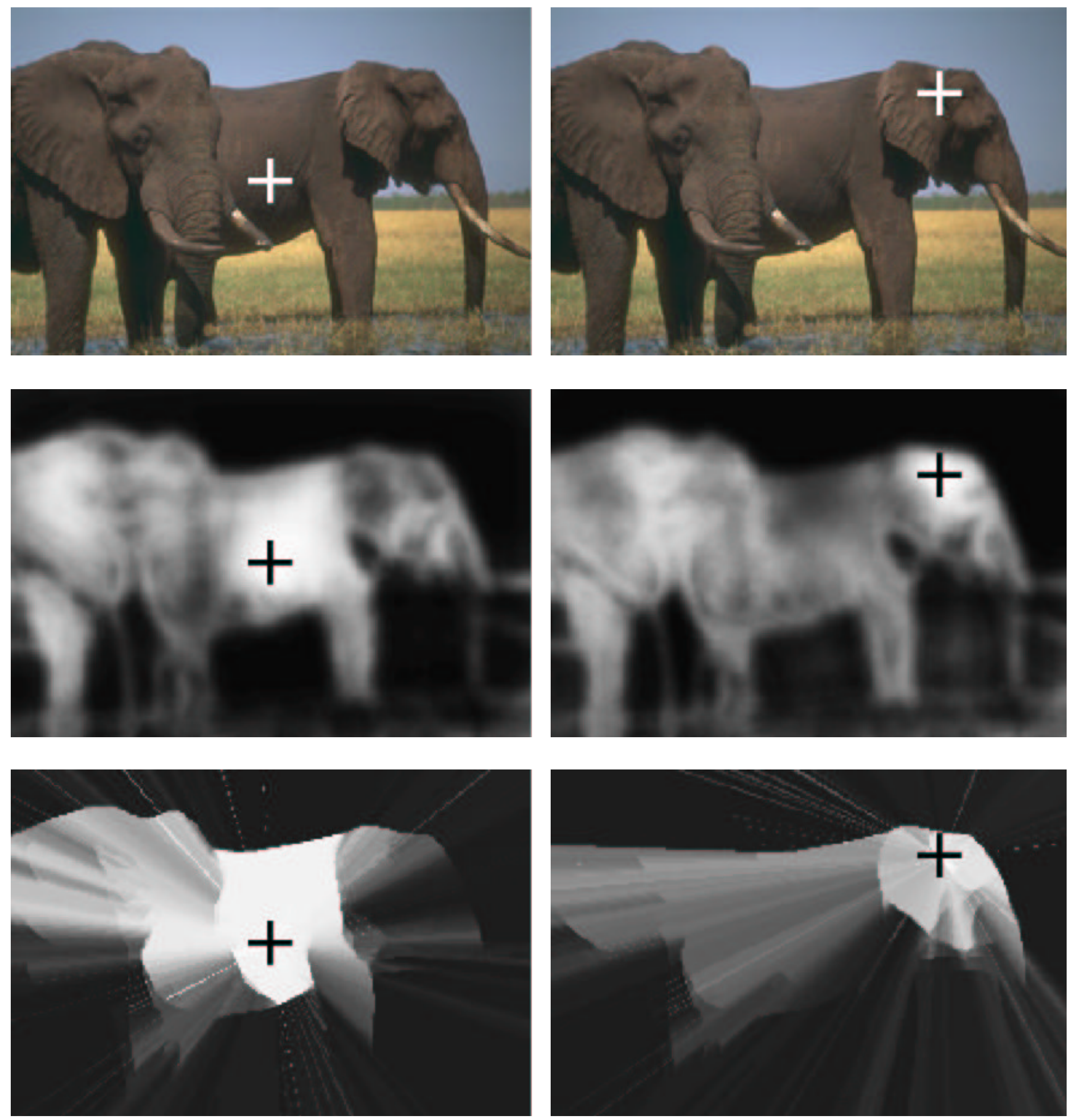
Graph cuts
Normalized cuts
Spectral clustering
....

Start by computing pairs of similarities/affinities between pixels

$$W_{ij} = e^{-||x_i - x_j||^2}$$

We want vector embedding or similarity to capture gestalt cues

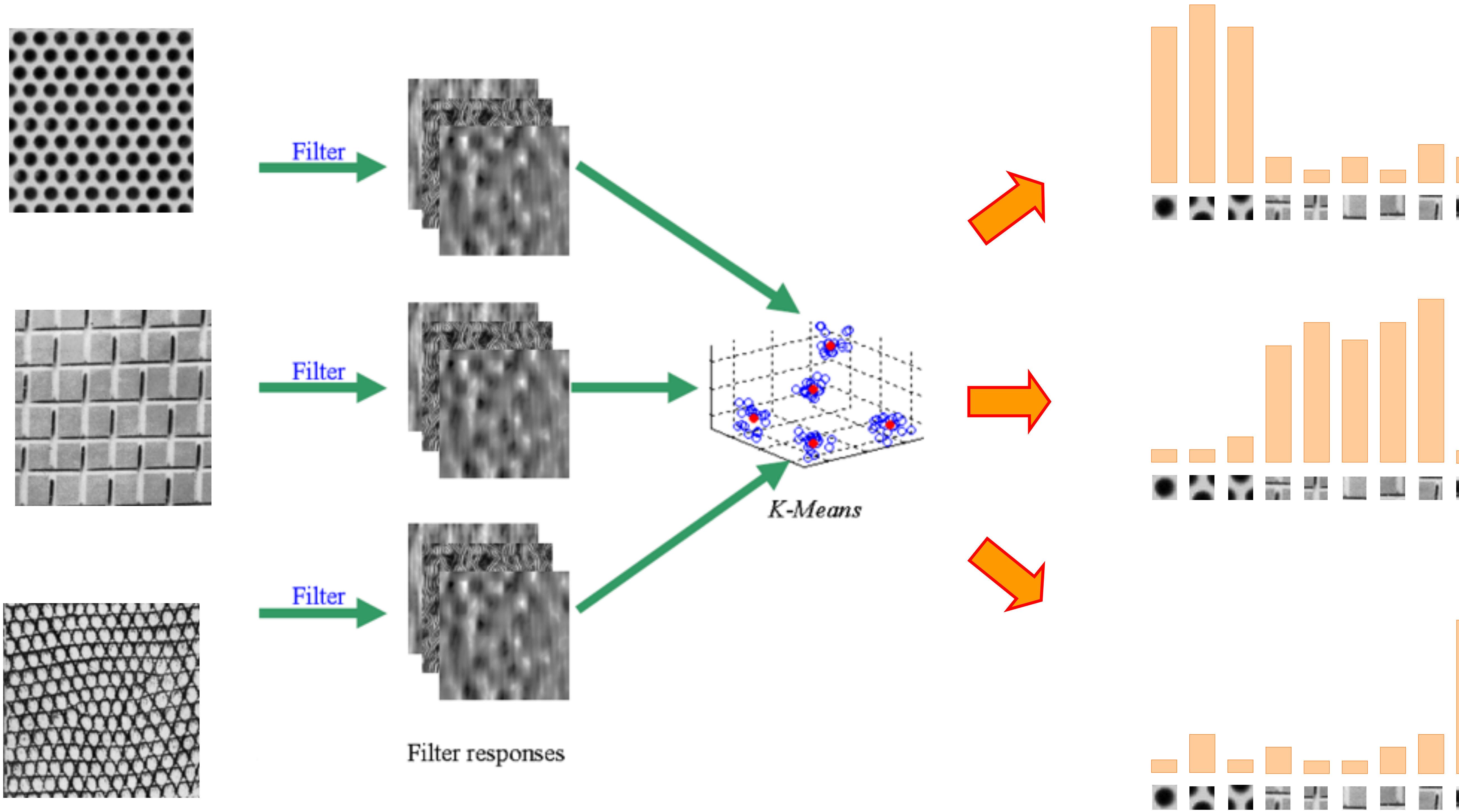
Flexibility of a similarity matrix



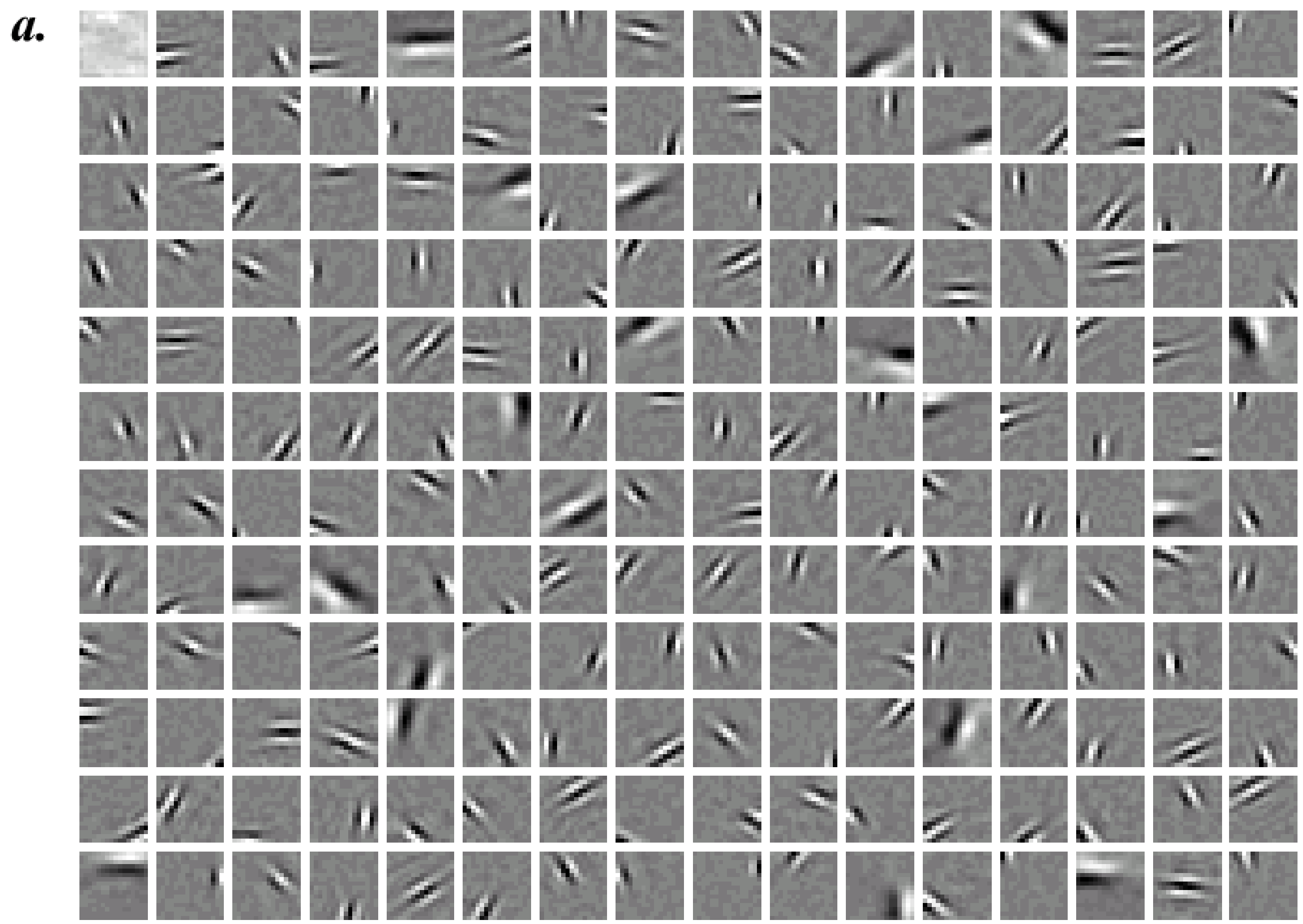
"Intervening contour cue":
two pixels are similar if there exists no strong edge between them

Other similarity functions

Historical perspective: texture cues



Alternate perspective: sparse coding



- **Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images.**
  Olshausen BA, Field DJ (1996). *Nature*, 381: 607-609

Sparse coding (0)

$$\min_{Z,D} C(Z,D,X) \quad \text{where} \quad C(Z,D,X) = \sum_i ||x_i - d_{z_i}||^2$$

Sparse coding (1)

$$\min_{D,Z} ||X - DZ||_F^2$$

$$X = [x_1, \ldots x_n]$$
$$D = [d_1, \ldots, d_K]$$
$$Z = [z_1, \ldots z_n]$$

K-means: $z_i = [\ldots, 0, 1, 0, \ldots]$

Sparse coding (2)

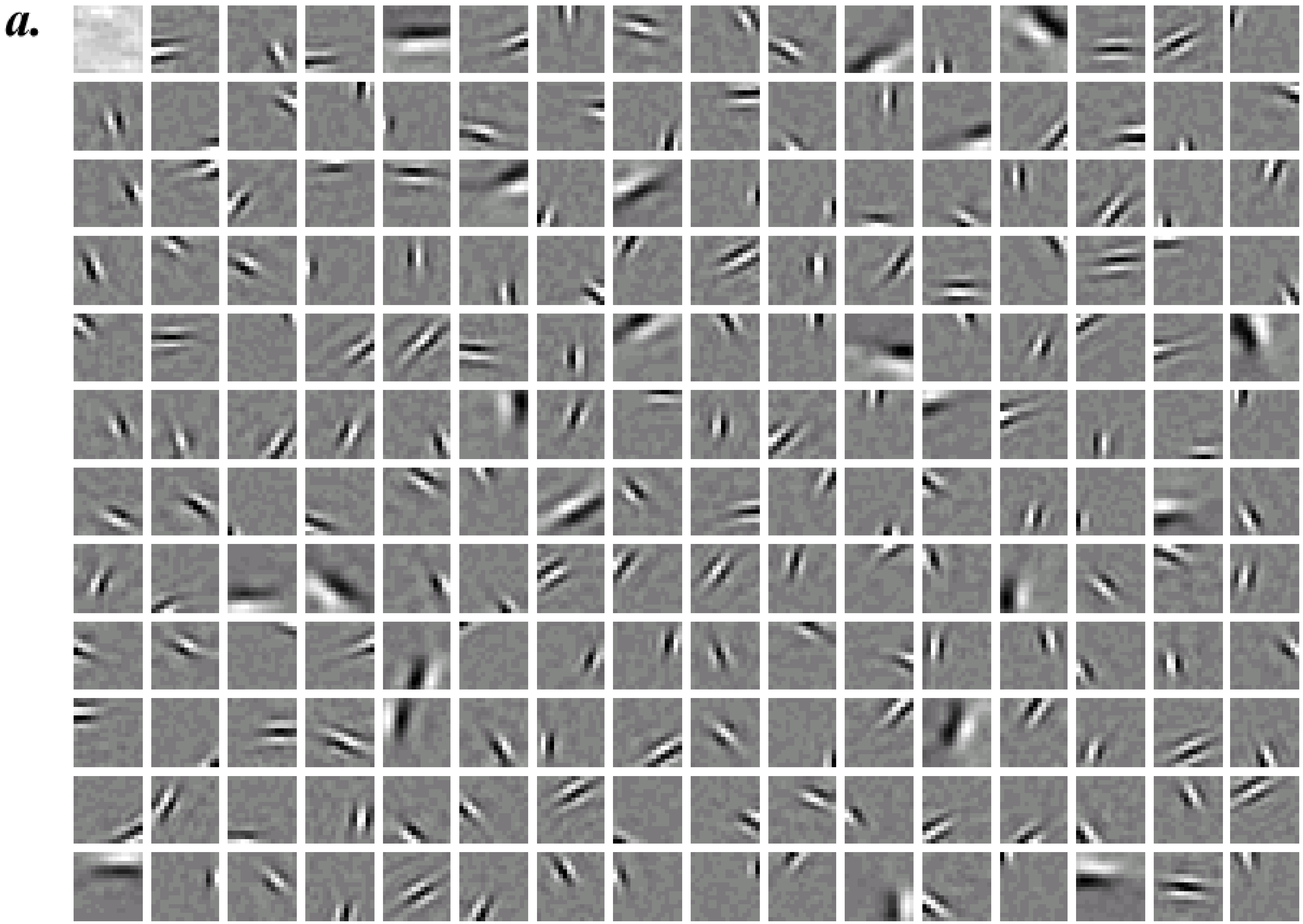
$$\min_{D,Z} ||X - DZ||_F^2$$ subject to sparse constraints on Z

$$X = [x_1, \dots x_n]$$
$$D = [d_1, \dots, d_K]$$
$$Z = [z_1, \dots z_n]$$

K-means: $z_i = [\dots, 0, 1, 0, \dots]$

L0 sparse-coding: $||z_i||_0 \leq M$ (greedy algorithms known as “matching pursuit”)

L1 sparse-coding: $||z_i||_1 \leq M$ (convex program)



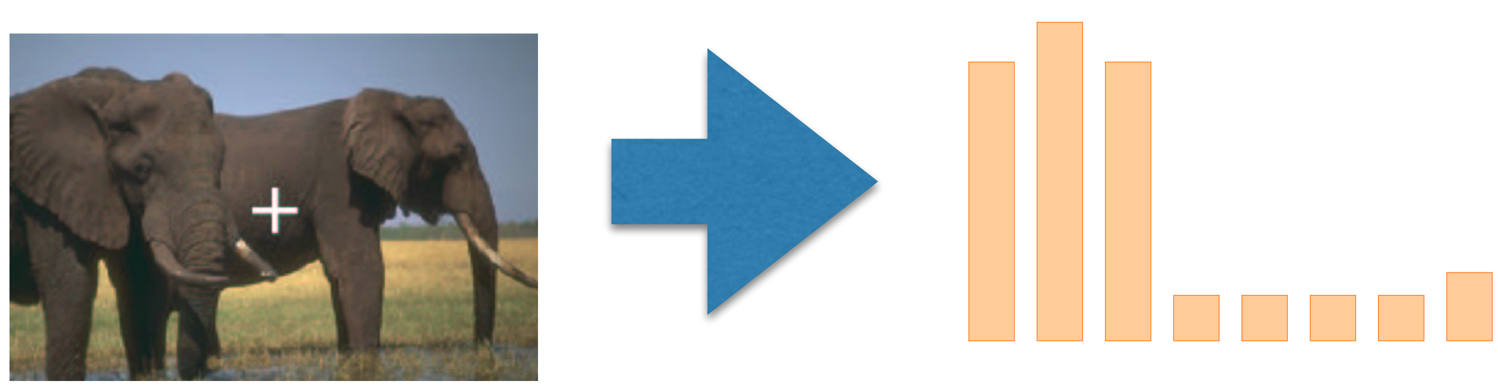
- **Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images.**
  Olshausen BA, Field DJ (1996). *Nature*, 381: 607-609

Extensions include *convolutional* sparse coding and *hierarchical* sparse coding (similar to unsupervised pre-training)

Recent similarity functions

(lower layer?) Activations of deep network

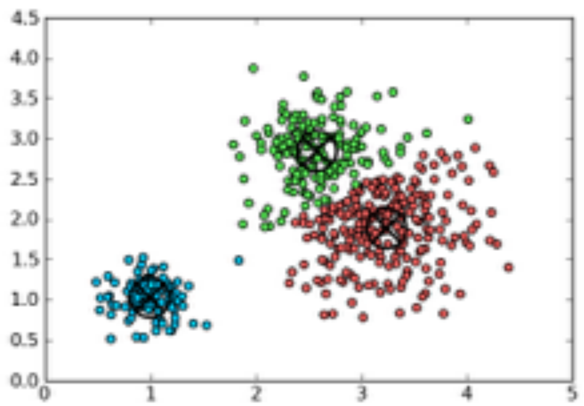
Recent work: train deep models to return back embeddings for pixels



LEARNING DENSE CONVOLUTIONAL EMBEDDINGS
FOR SEMANTIC SEGMENTATION

Bird's eye view of grouping

Model-based clustering

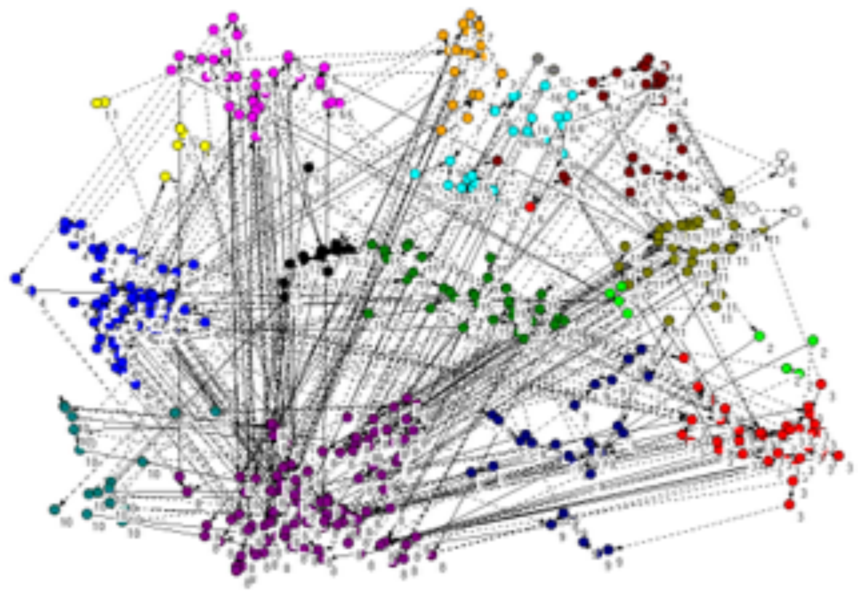


K-means
Sparse coding
Gaussian mixture models
....

Start by representing pixel as a vector

$$x_i \in R^N$$

Graph theoretic (pairwise)



Graph cuts
Normalized cuts
Spectral clustering
....

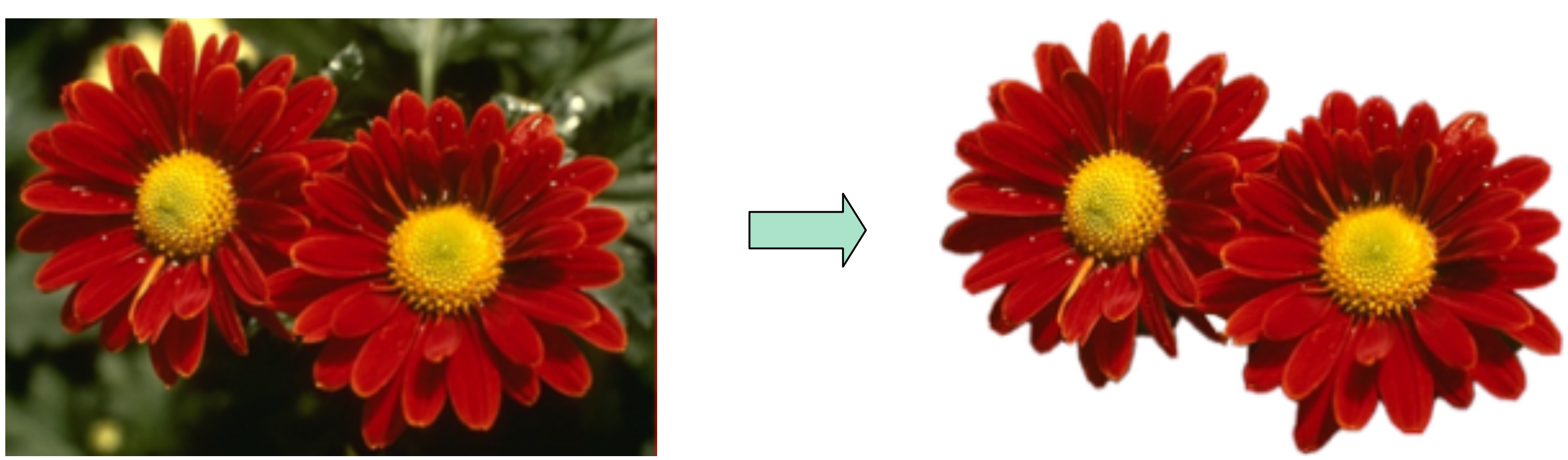
Start by computing pairs of similarities/affinities between pixels

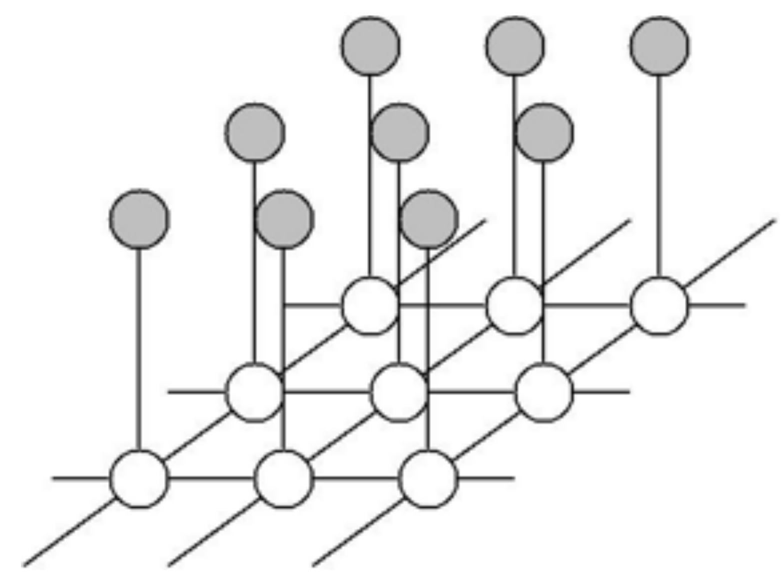
$$W_{ij} = e^{-||x_i - x_j||^2}$$

We want vector embedding or similarity to capture gestalt cues

Graph-theoretic grouping

Formalize grouping as a graph labeling problem
(cf, discrete MRFs for low-level vision)





Global energy function

Pixel labeling

$$\min_x E(X), \quad E(x) = \sum_i \phi_i(x_i) \quad x_i \in \{0, 1\}$$

$$y_i = \begin{bmatrix} r_i \\ b_i \\ g_i \end{bmatrix}$$

$$\phi(x_i = 1) = -\log p(y_i | x_i = 1)$$

$$\phi(x_i = 0) = -\log p(y_i | x_i = 0)$$

Markov Random Field (MRF) energy functions

$$E(x) = \sum_{ij \in \mathcal{E}} \psi_{ij}(x_i, x_j) + \sum_{i \in V} \phi_i(x_i)$$

Inference in MRFs

- In general, computing the min energy soln is NP complete
- Inference is tractable for some problems
    - Trees
    - Submodular functions (“graphcut-able”)

Iterated conditional modes

$$E(x) = \sum_{ij} \psi_{ij}(x_i, x_j) + \sum_i \phi_i(x_i)$$

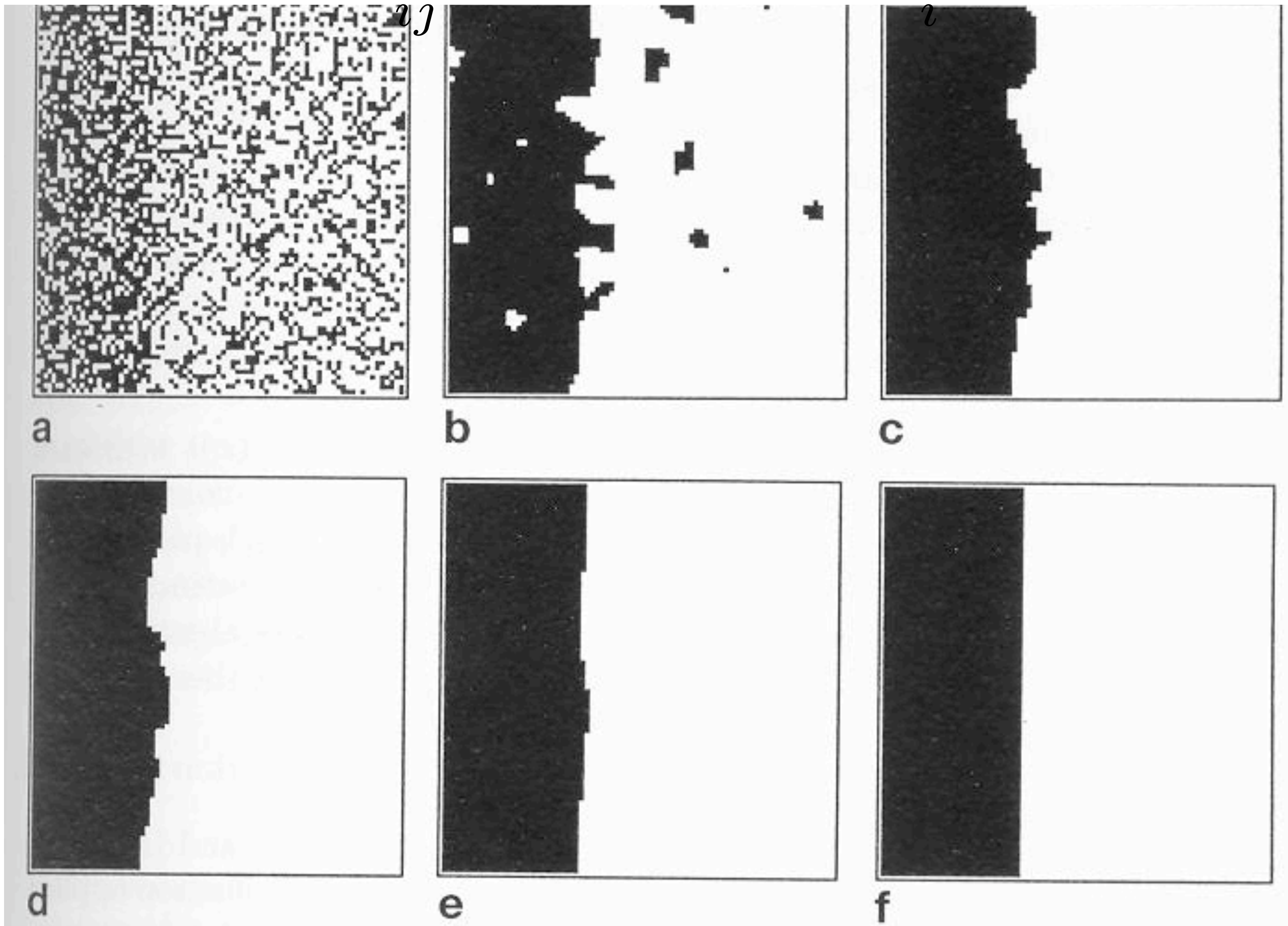
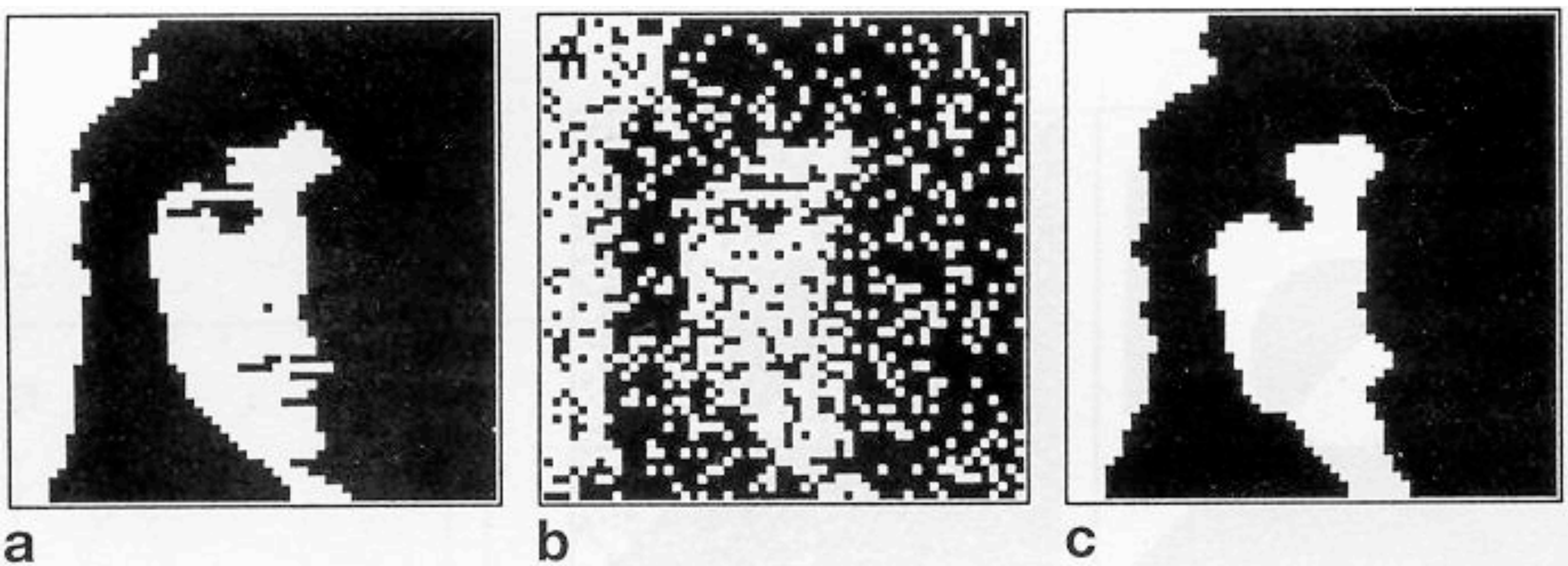


Fig. 6.2. Various steps of ICM

Sequentially update $x_i := \arg\min E(x)$



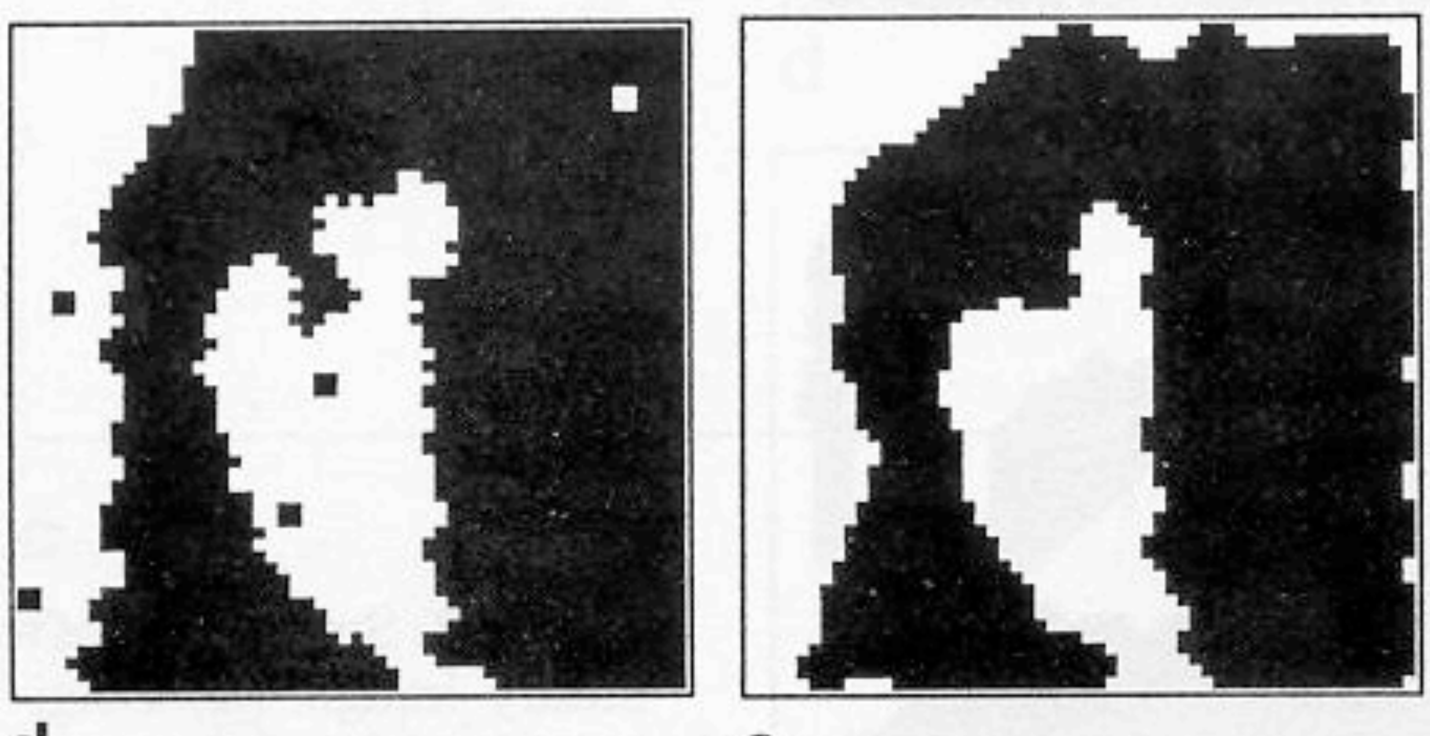


Fig. 2.3. Smoothing with the wrong prior. (**a**) Original, (**b**) degraded image, (**c**) MAP estimate $\beta = 1$, (**d**) MAP estimate $\beta = 0.3$, (**e**)

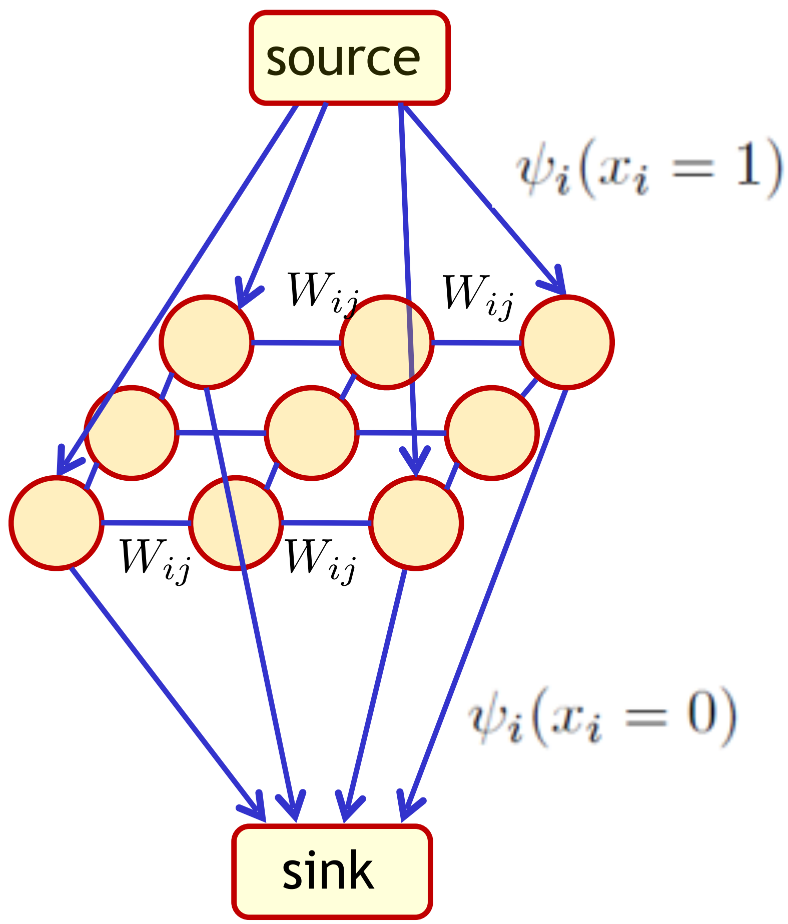
Markov Random Field (MRF) energy functions

$$E(x) = \sum_{ij} \psi_{ij}(x_i, x_j) + \sum_i \phi_i(x_i)$$

$$\psi_{ij}(0,0) + \psi_{ij}(1,1) \leq \psi_{ij}(0,1) + \psi_{ij}(1,0)$$

“Submodular” energy function (favors smooth labels)

Potts pairwise model

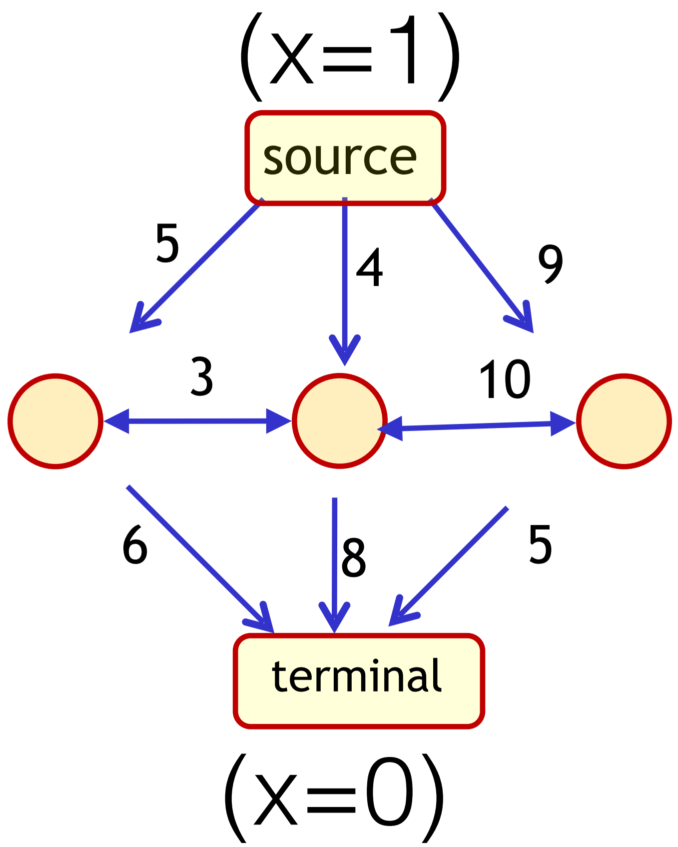


$$E(x) = \sum_{ij \in \mathcal{E}} W_{ij} I(x_i \neq x_j) + \sum_{i \in V} \phi_i(x_i)$$

Graph construction

$$E(x) = \sum_{ij \in \mathcal{E}} W_{ij} I(x_i \neq x_j) + \sum_{i \in V} \phi_i(x_i)$$

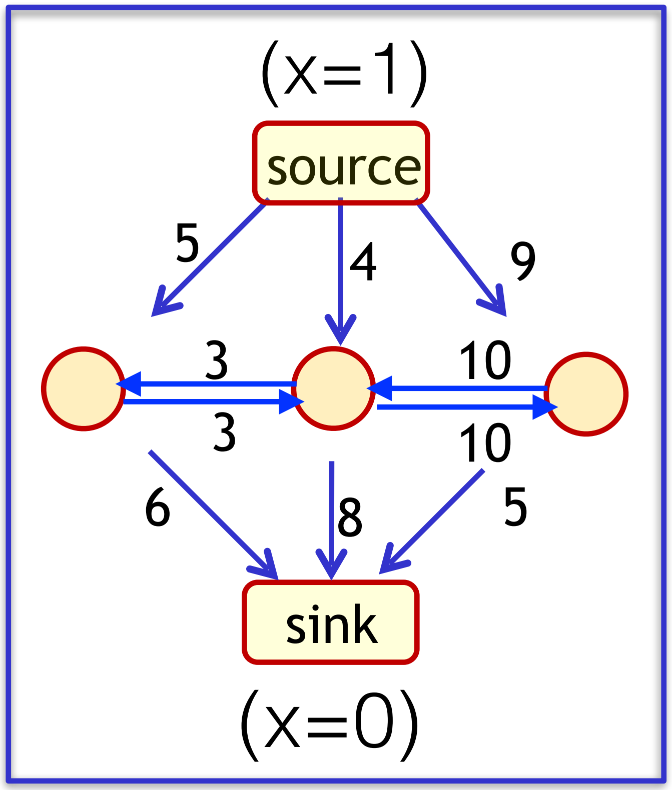
1. Define node for each pixel
2. Add edges between pixels with weight = cost of diagreeing label
3. Add a source + terminal node
4. Add edges from pixels to source+terminal with weight = cost of fg/bg label



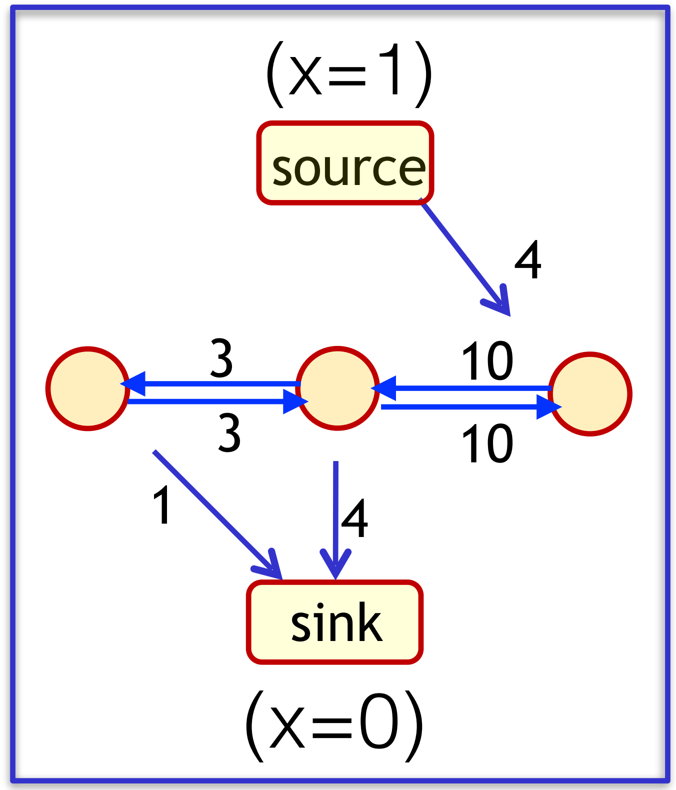
Claims: (1) minimum energy soln given by minimum cut that separates s-t nodes

(2) Equivalent to *min-cut max-flow* problem

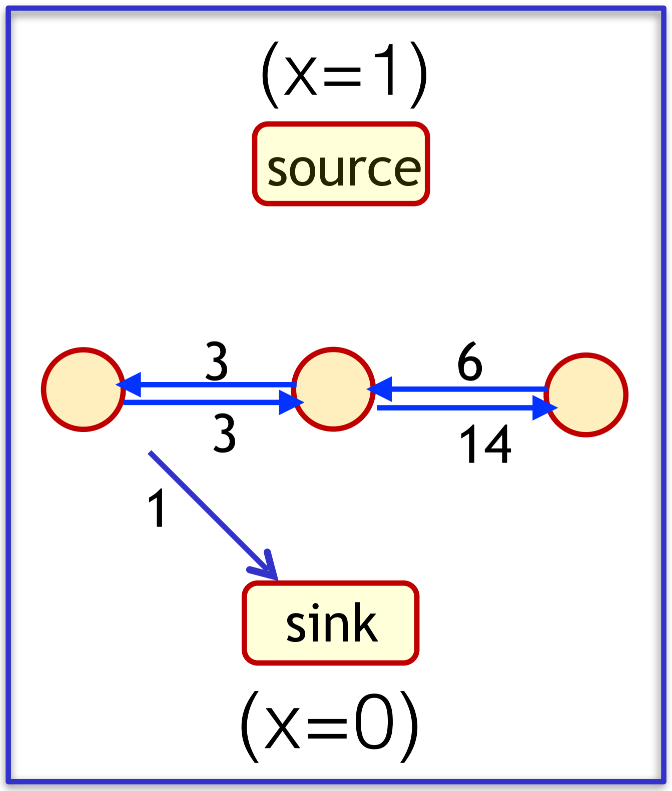
Max-flow (sketch)



Flow=0



Flow += 5+4+5 => 14



Flow += 4 => 18

(1) “Per-pixel” labeling: max-out flow along each s-pixel-t path

(2) Push out remaining flow (find flowable path with bread-first search)

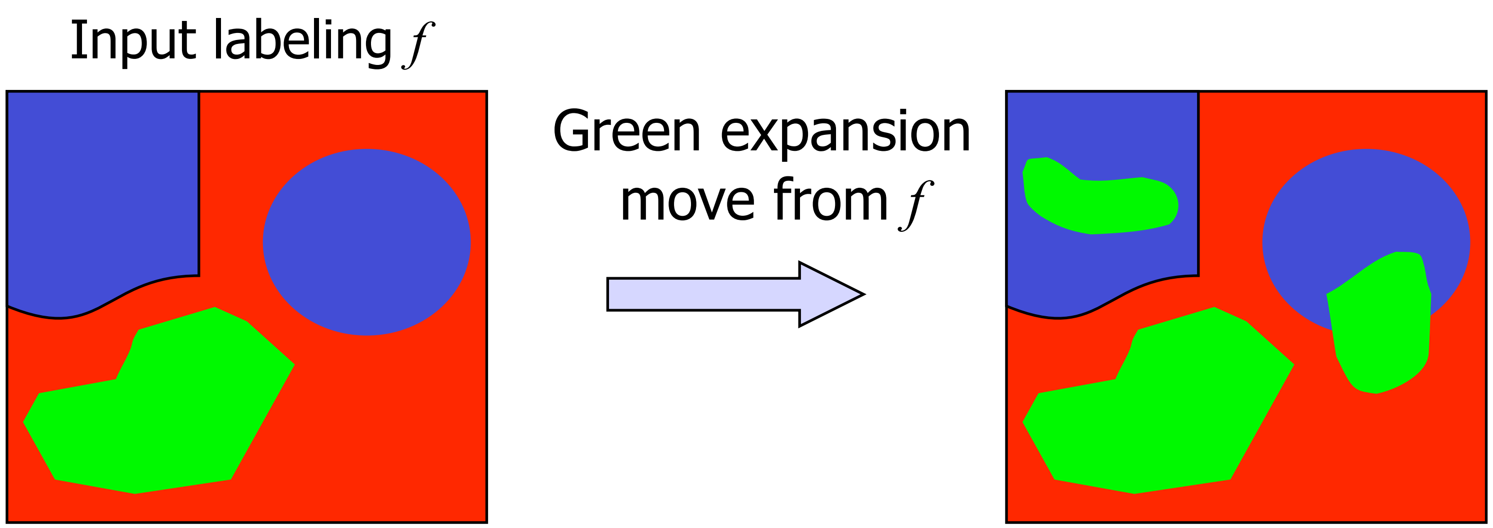
Repeat until you can’t push any more flow

An island of pixels will be cut when the costs of its perimeter is smaller than the (delta) cost of its area

Extensions - k-way labeling

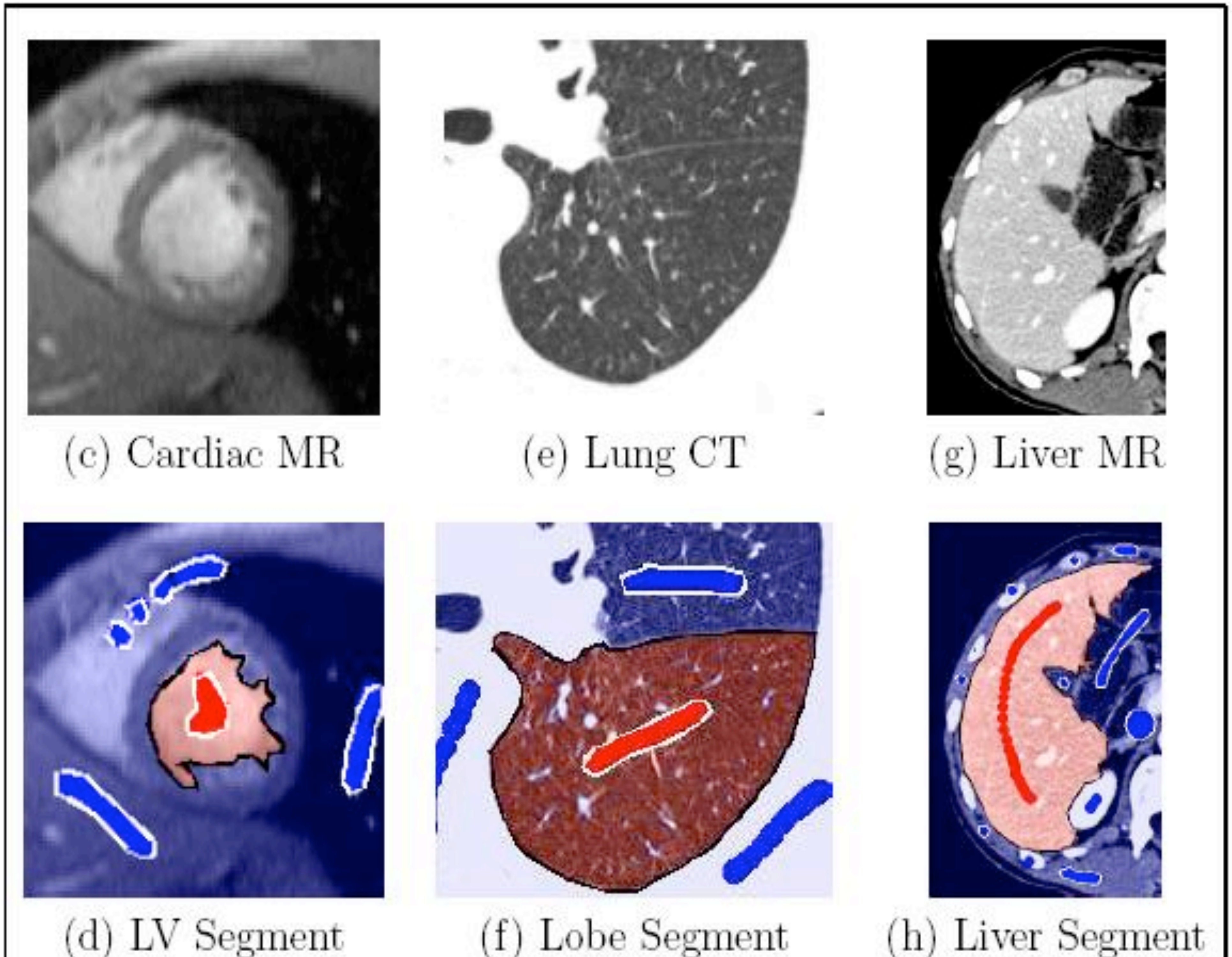
$$E(x) = \sum_{ij \in \mathcal{E}} W_{ij} I(x_i \neq x_j) + \sum_{i \in V} \phi_i(x_i)$$

$$x_i \in \{1 \ldots K\}$$



- Find green expansion move that most decreases E
  - Move there, then find the best blue expansion move, etc
  - Done when no alpha-expansion move decreases the energy, for any label alpha

Interactive segmentation



Use user-strokes to fix certain labels to fb/bg or learn initial color models

Combining k-means + graph cuts

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

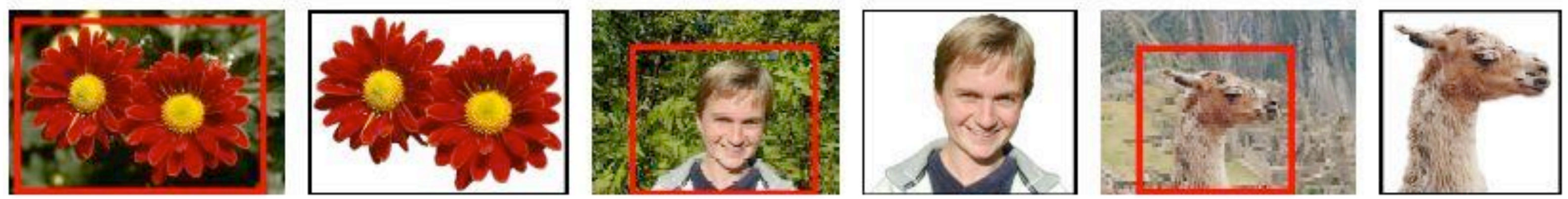


Figure 1: **Three examples of GrabCut.** The user drags a rectangle loosely around an object. The object is then extracted automatically.

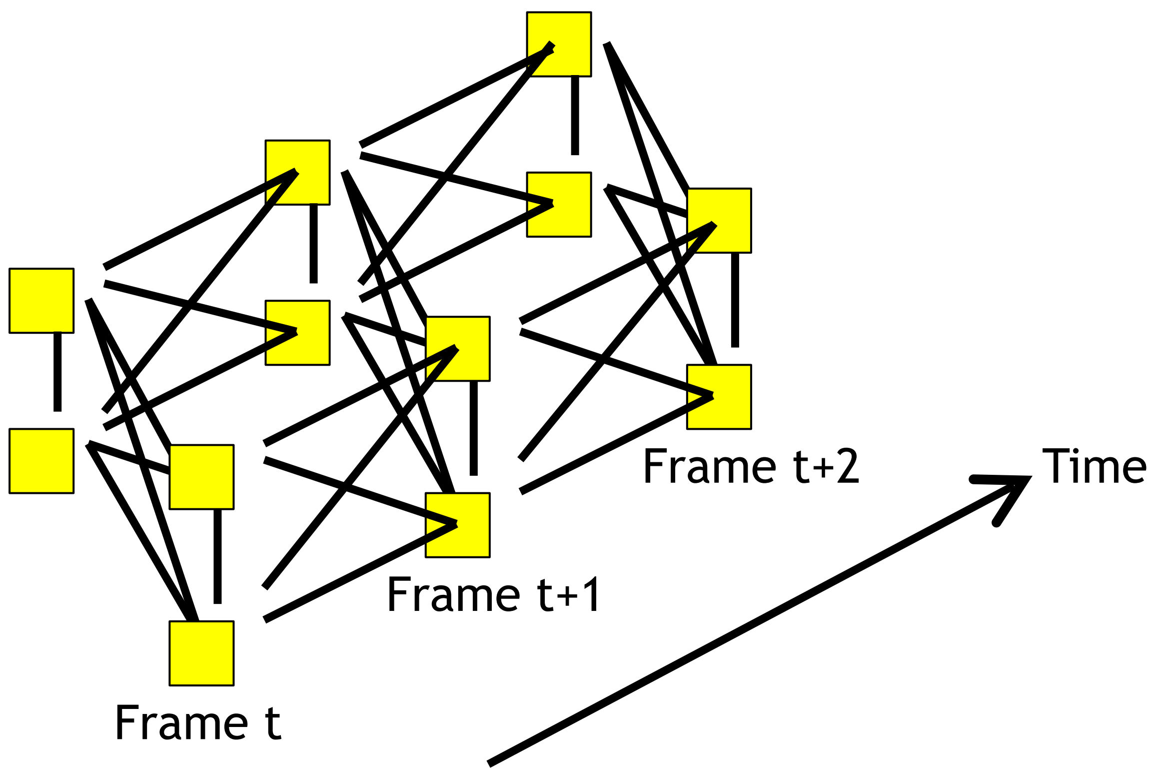
Spatially Coherent Clustering Using Graph Cuts

Ramin Zabih
Cornell University
Ithaca, NY 14853

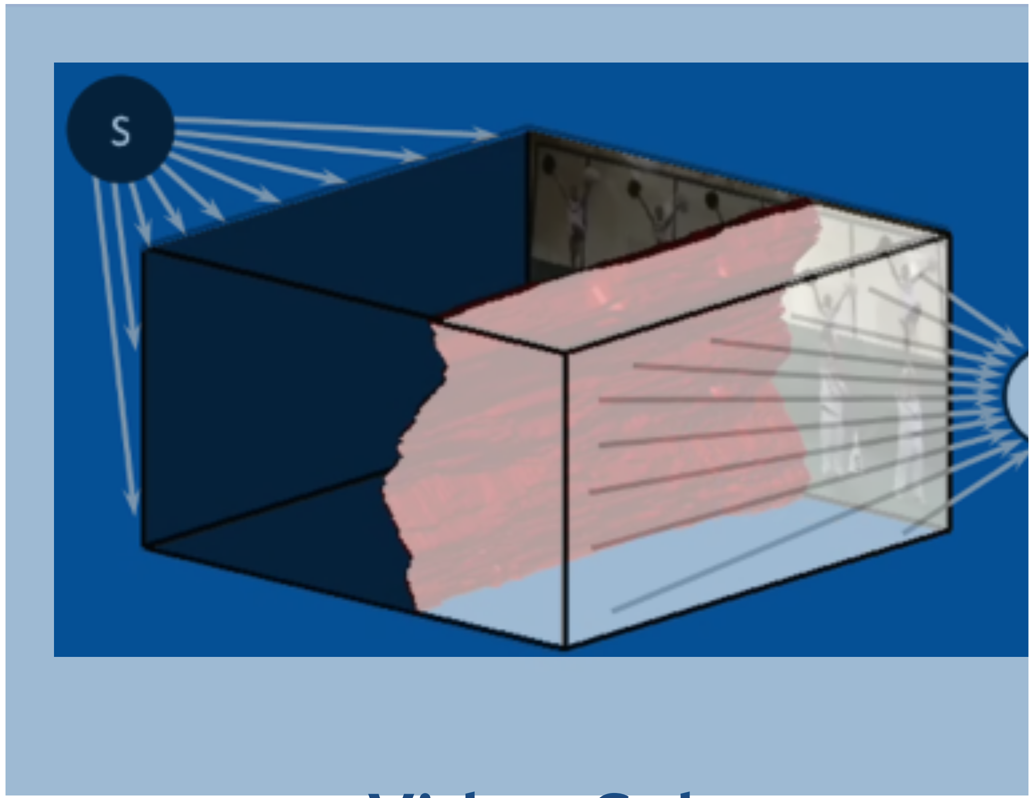
Vladimir Kolmogorov
Microsoft Research
Cambridge, UK

$$\min_{x,\mu} \sum_{ij \in \mathcal{E}} W_{ij} I(x_i \neq x_j) + \sum_{i \in V} ||y_i - \mu_{x_i}||^2$$

Multi-dimensional graphcuts

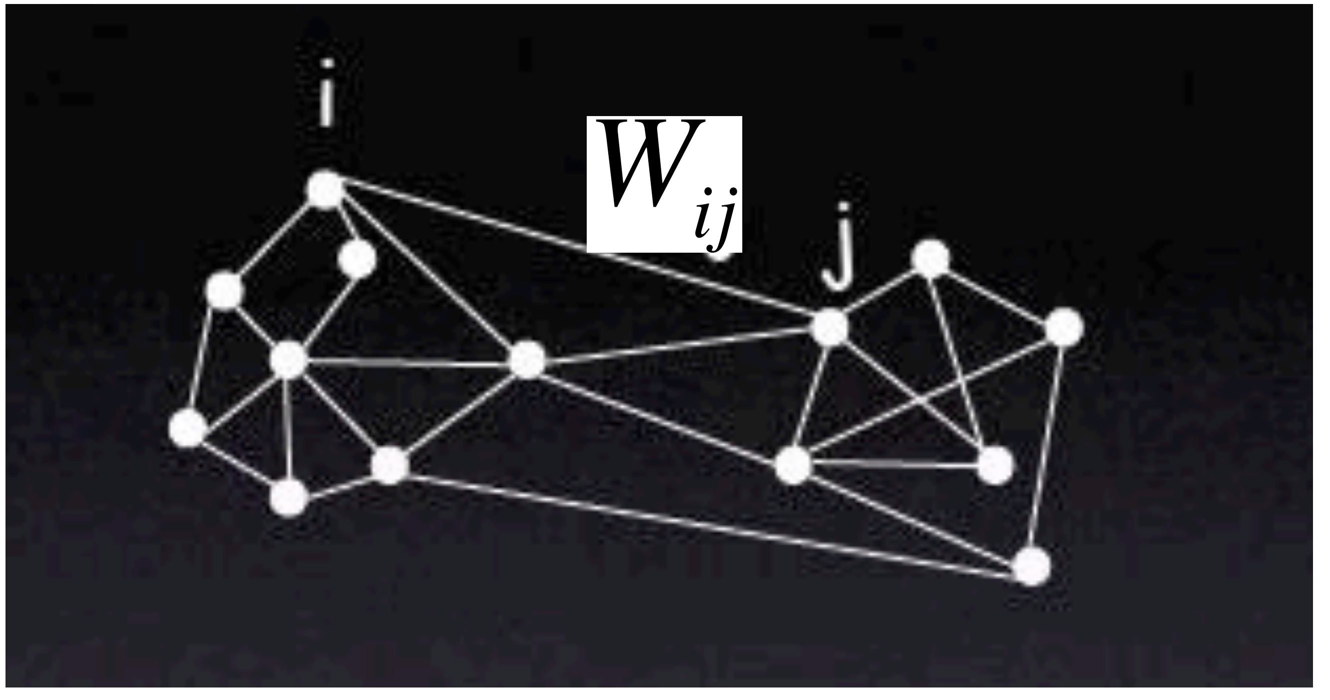


Video Cube



Video Cube

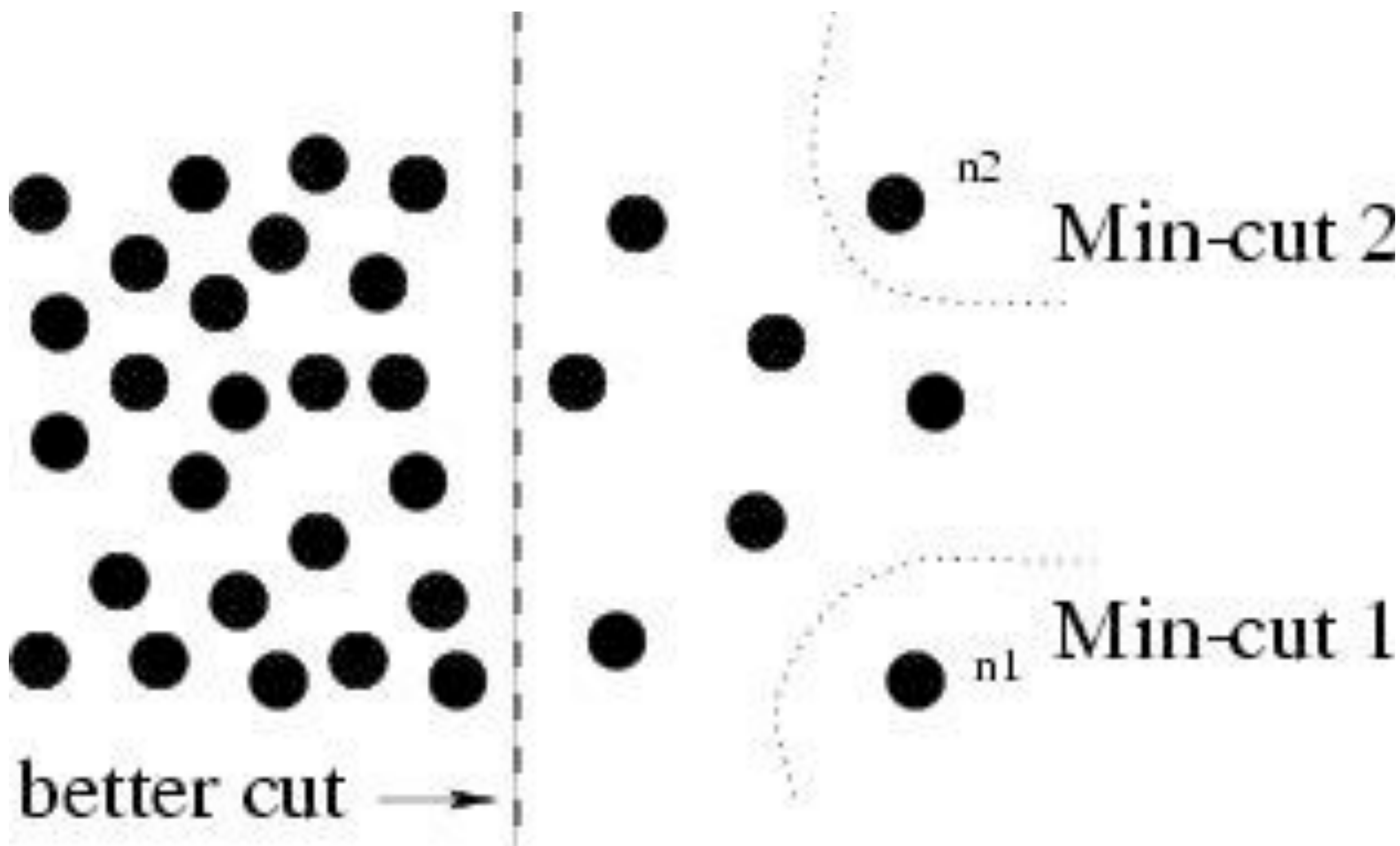
Can we define a generic cut without a local term?



Allows us to group pixels sole based off of pairwise properties

The problem with mincuts

(local terms reduce this, but tendency is still there in graphcuts !)



Up next…

Normalized Cuts and Image Segmentation

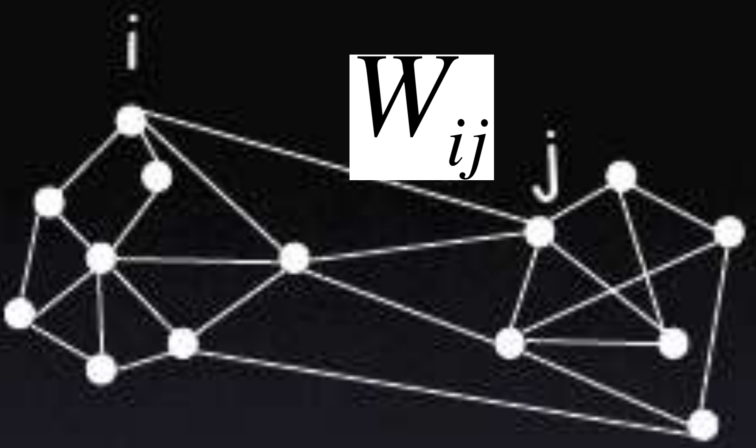
Jianbo Shi and Jitendra Malik, *Member*, *IEEE*

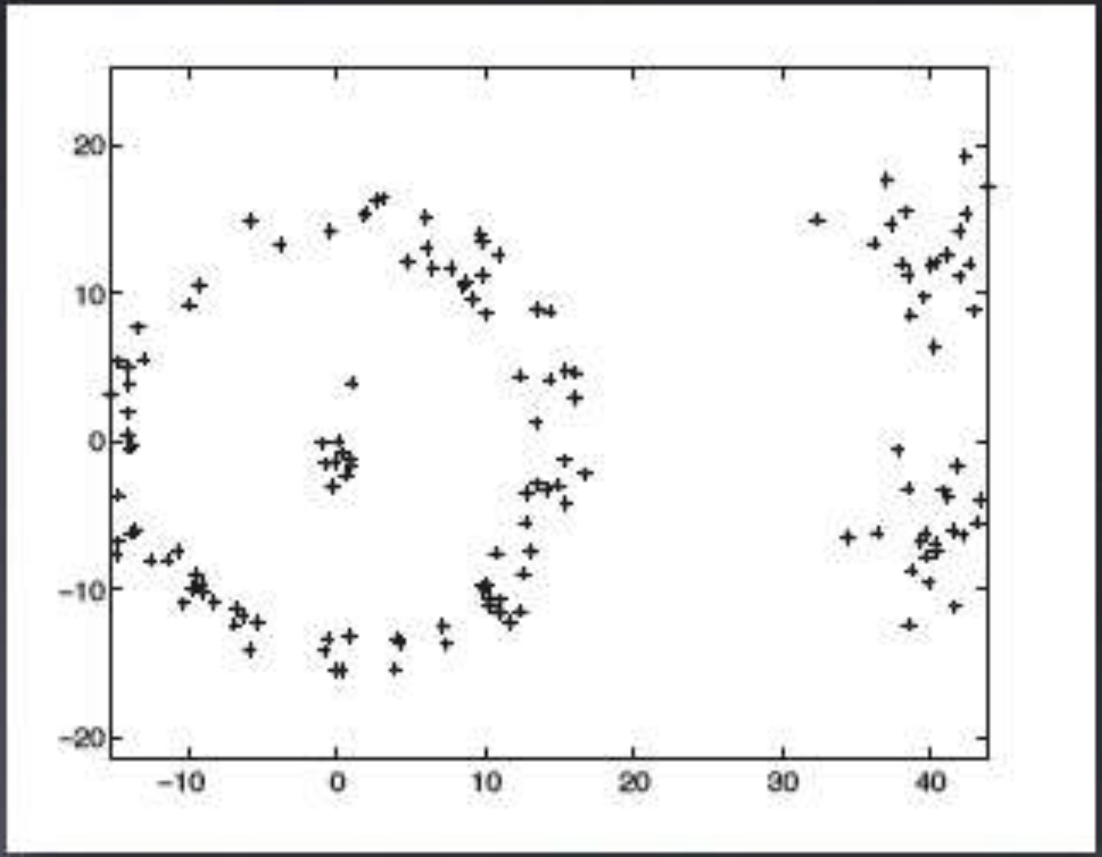
Abstract—We propose a novel approach for solving the perceptual grouping problem in vision. Rather than focusing on local features and their consistencies in the image data, our approach aims at extracting the global impression of an image. We treat image segmentation as a graph partitioning problem and propose a novel global criterion, the *normalized cut*, for segmenting the graph. The *normalized cut* criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. We show that an efficient computational technique based on a generalized eigenvalue problem can be used to optimize this criterion. We have applied this approach to segmenting static images, as well as motion sequences, and found the results to be very encouraging.

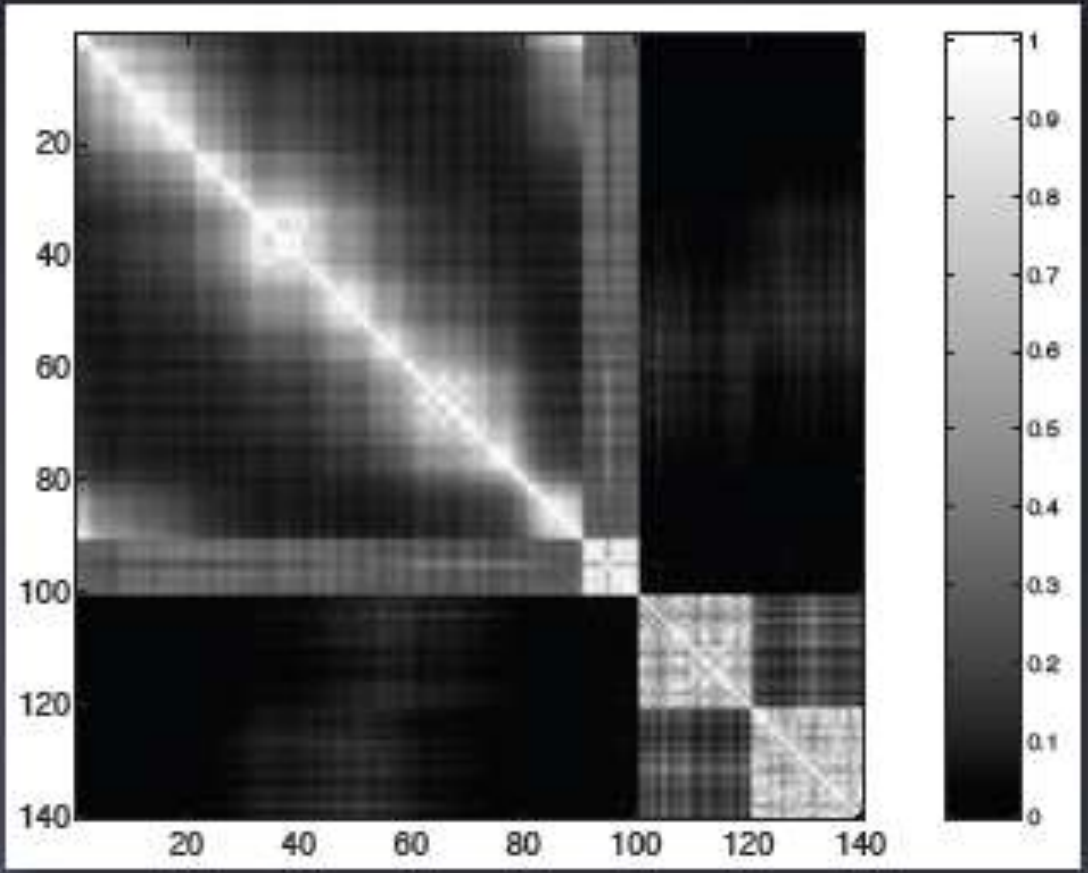
Graph Terminology

Weighted adjacency matrix:

$$W_{ij} = e^{-||y_i - y_j||^2}$$

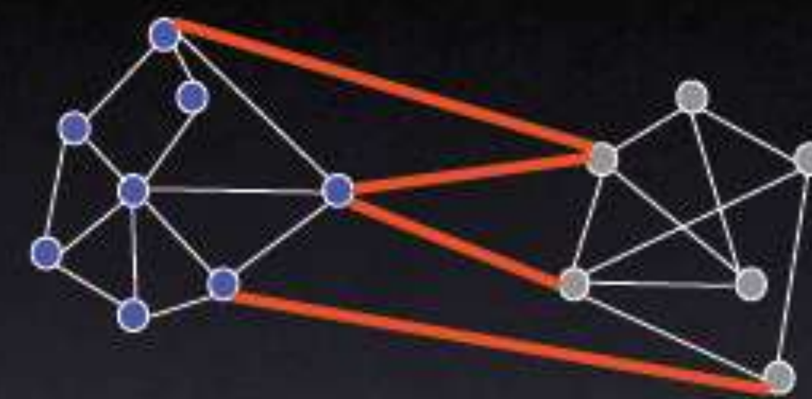


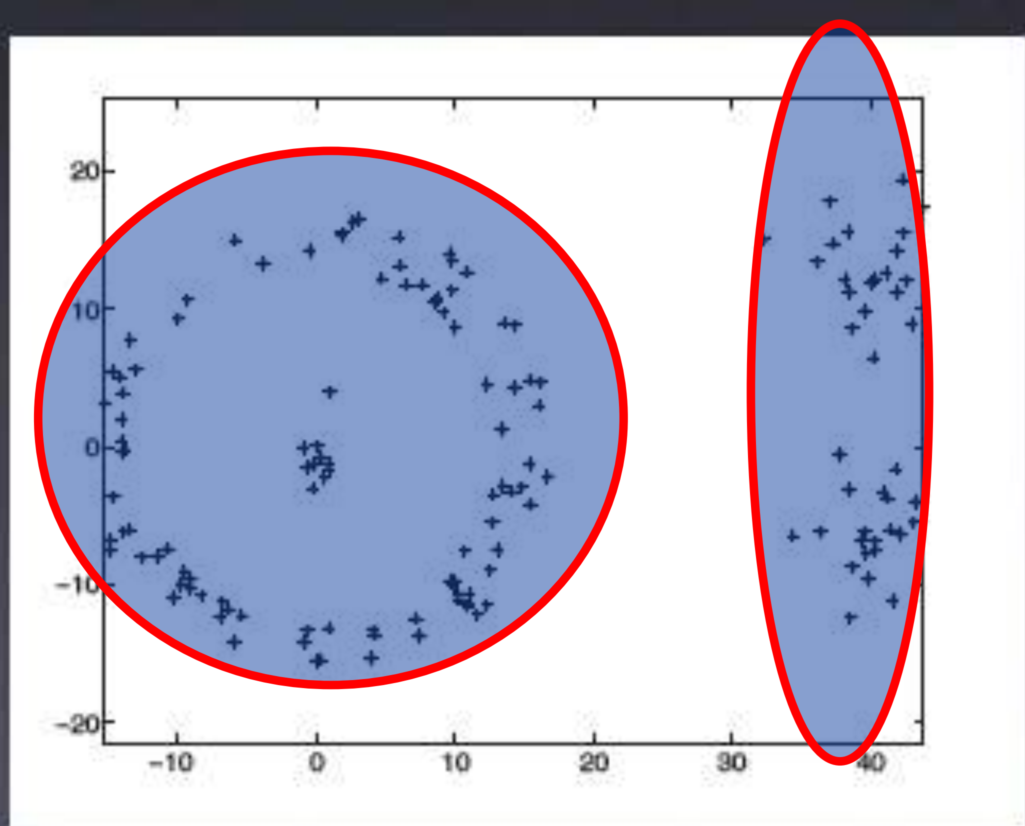


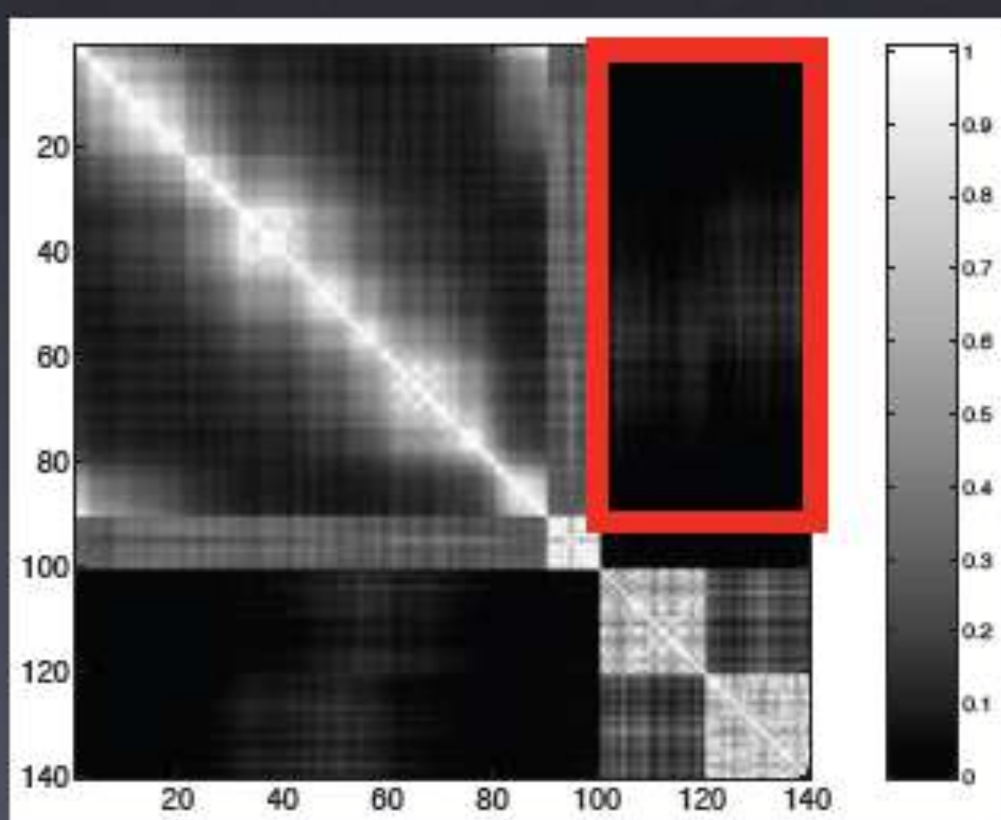


Cuts in a graph

$$cut(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} W_{ij}$$

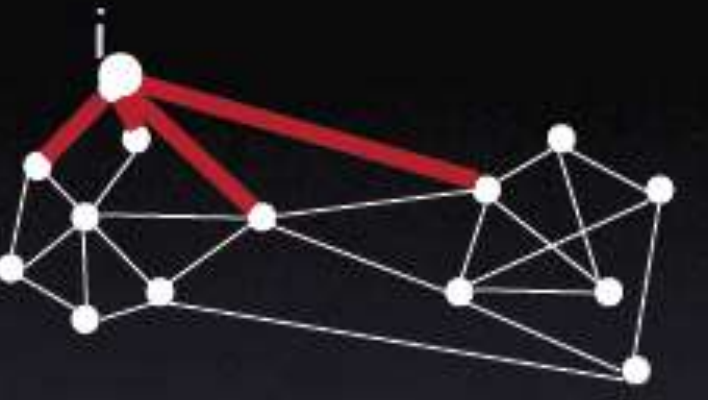




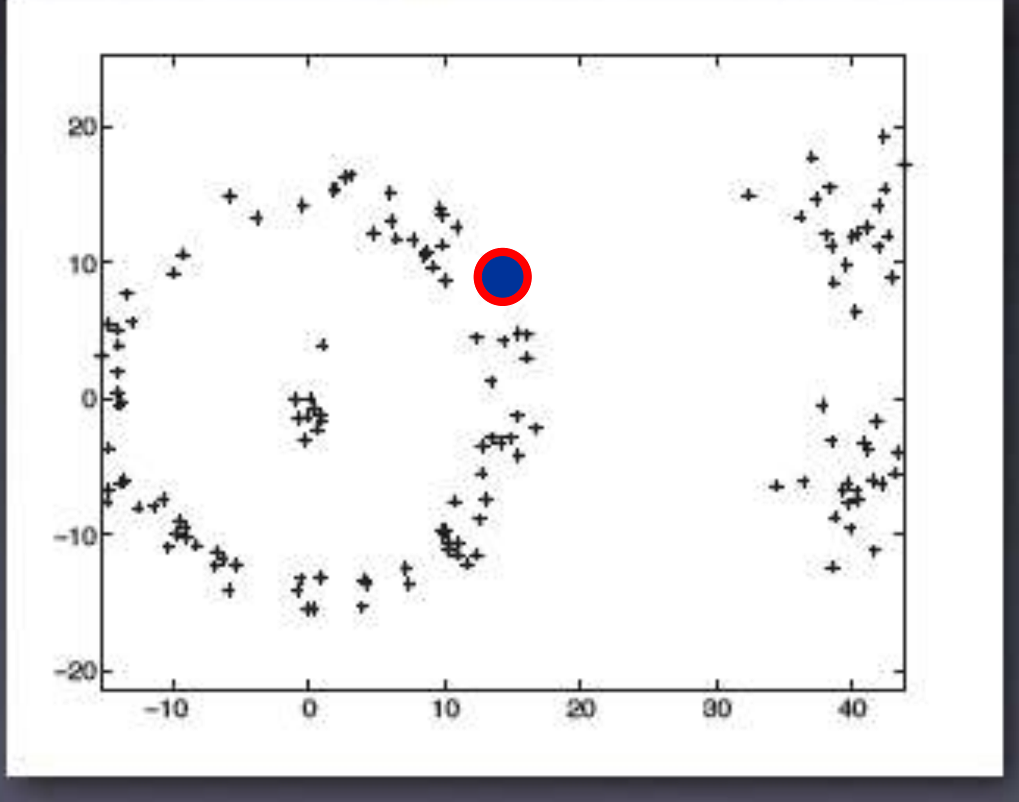


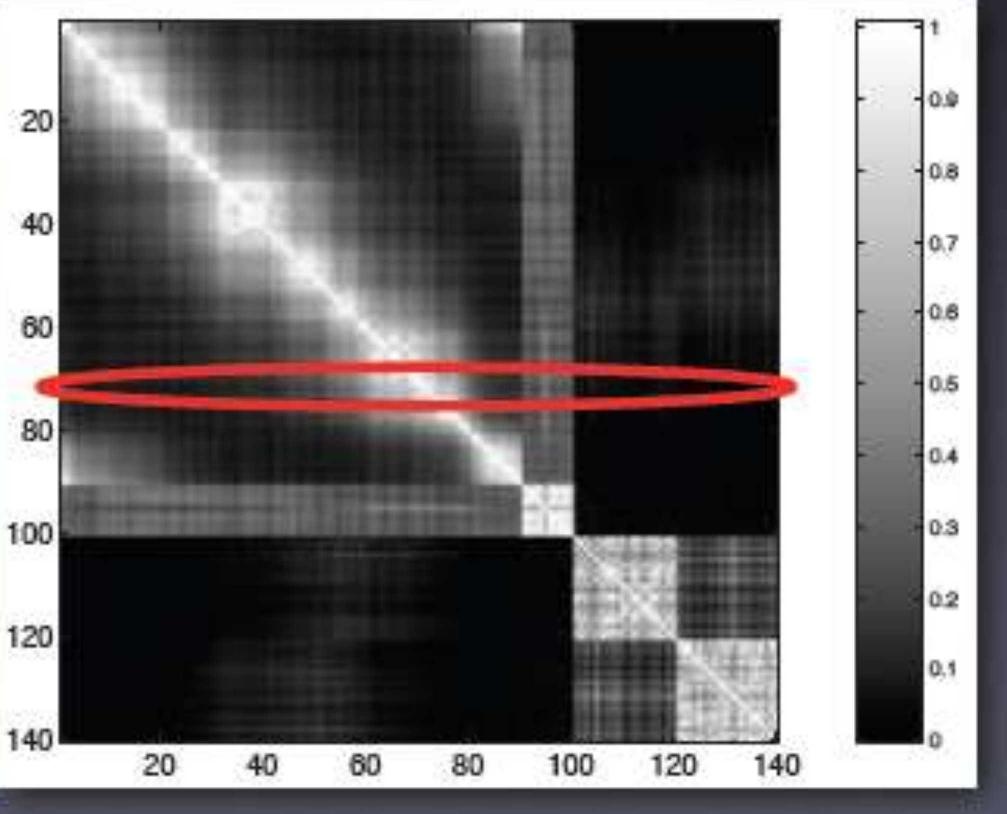
Graph Terminology

Degree of node: $$d_i = \sum_j W_{ij}$$



$$D = diag(\{d_i\})$$

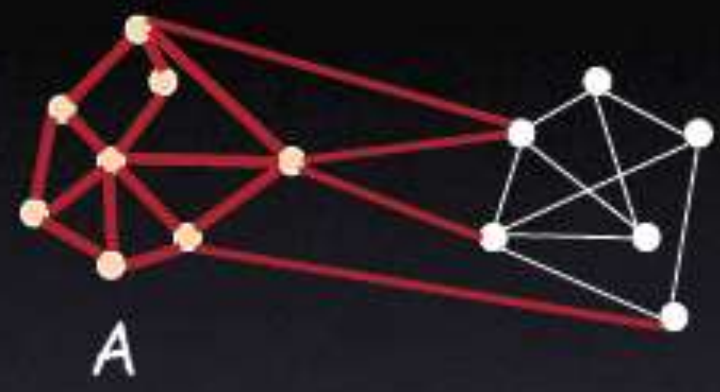


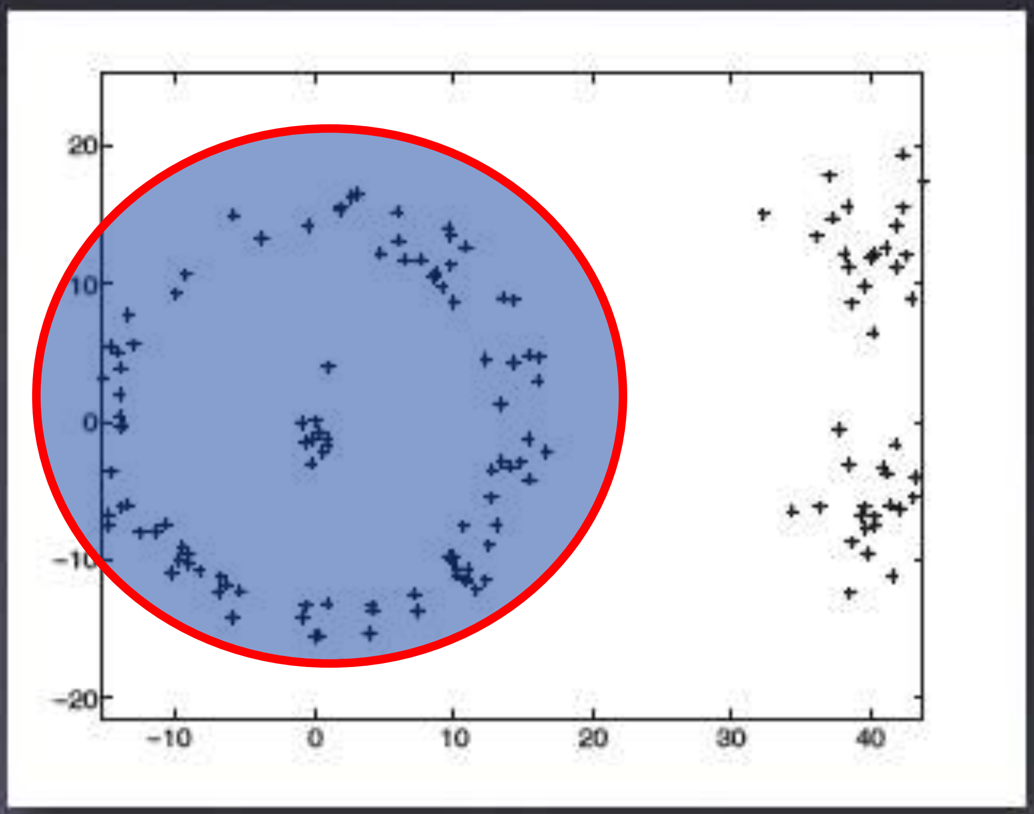


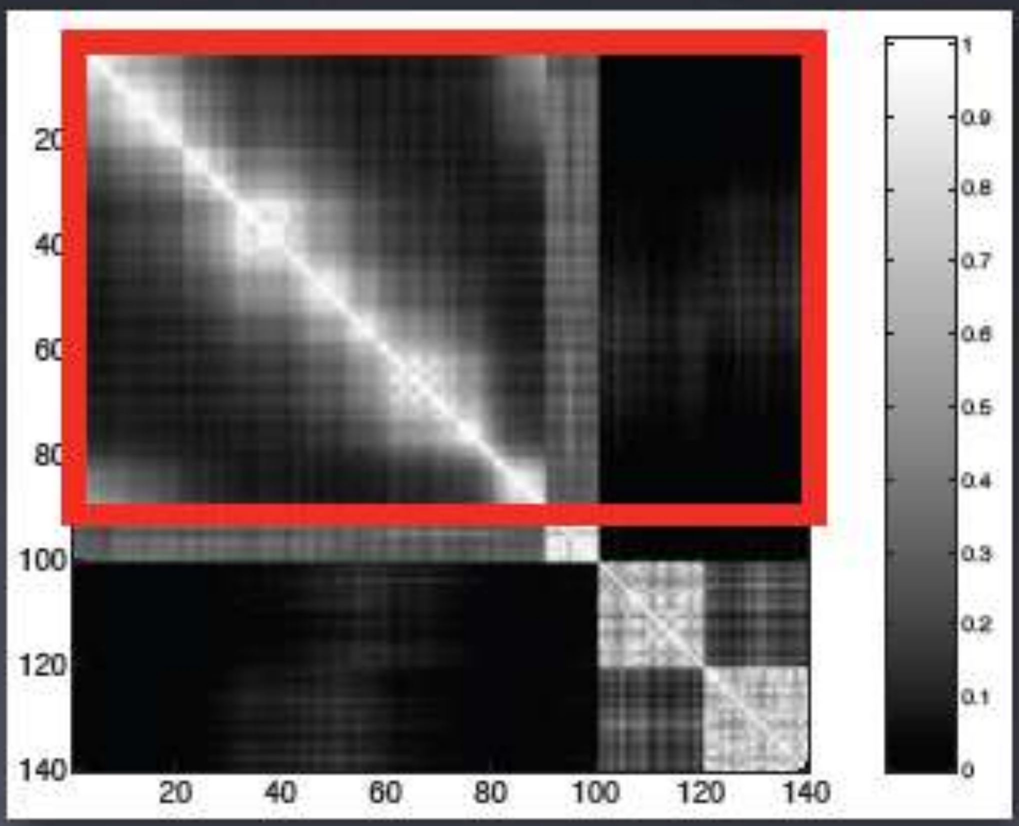
Graph Terminology

Volume of set:

$$vol(A) = \sum_{i \in A} d_i,\ A \subseteq V$$

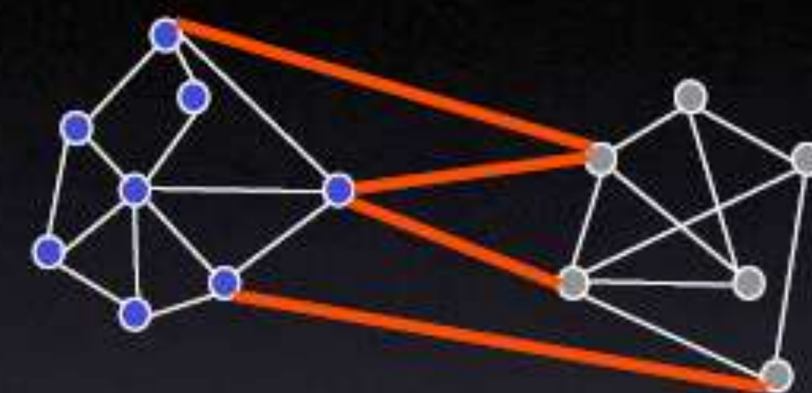






Normalize cuts in a graph

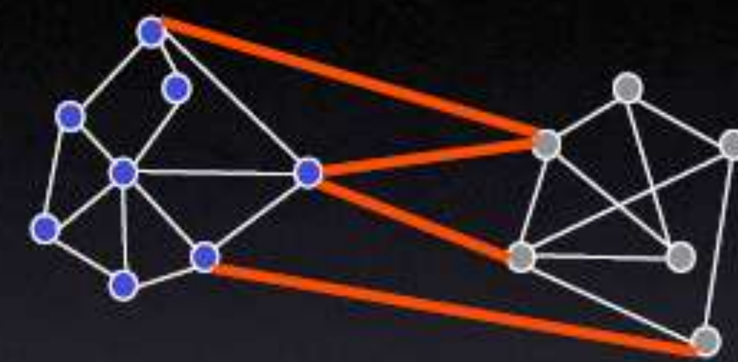
- (edge) Ncut = balanced cut



$$Ncut(A,B) = cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})$$

Normalize cuts in a graph

- (edge) Ncut = balanced cut



$$Ncut(V_1, V_2, \ldots V_k) = \frac{1}{K} \sum_{i=1}^{k} \frac{cut(V_i, V \setminus V_i)}{vol(V_i)}$$

Multiple cuts

Finding high quality cuts

- Problem: finding the minimum normalized cut is NP-hard, even for k=2.

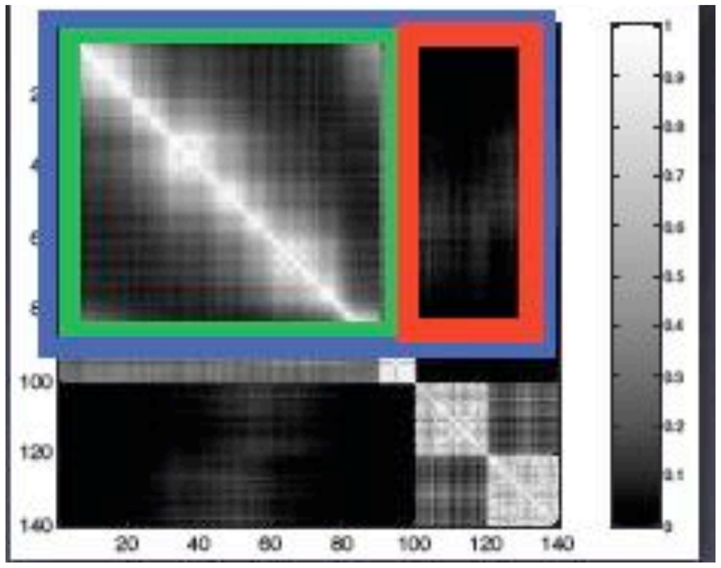
- Solution: Formulate as a quadratic program and relax integer constraints to get an approximate solution
  - uses nice ideas from linear algebra and spectral graph theory

Matrix formulation

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Define an indicator vector specifying partition it belongs to

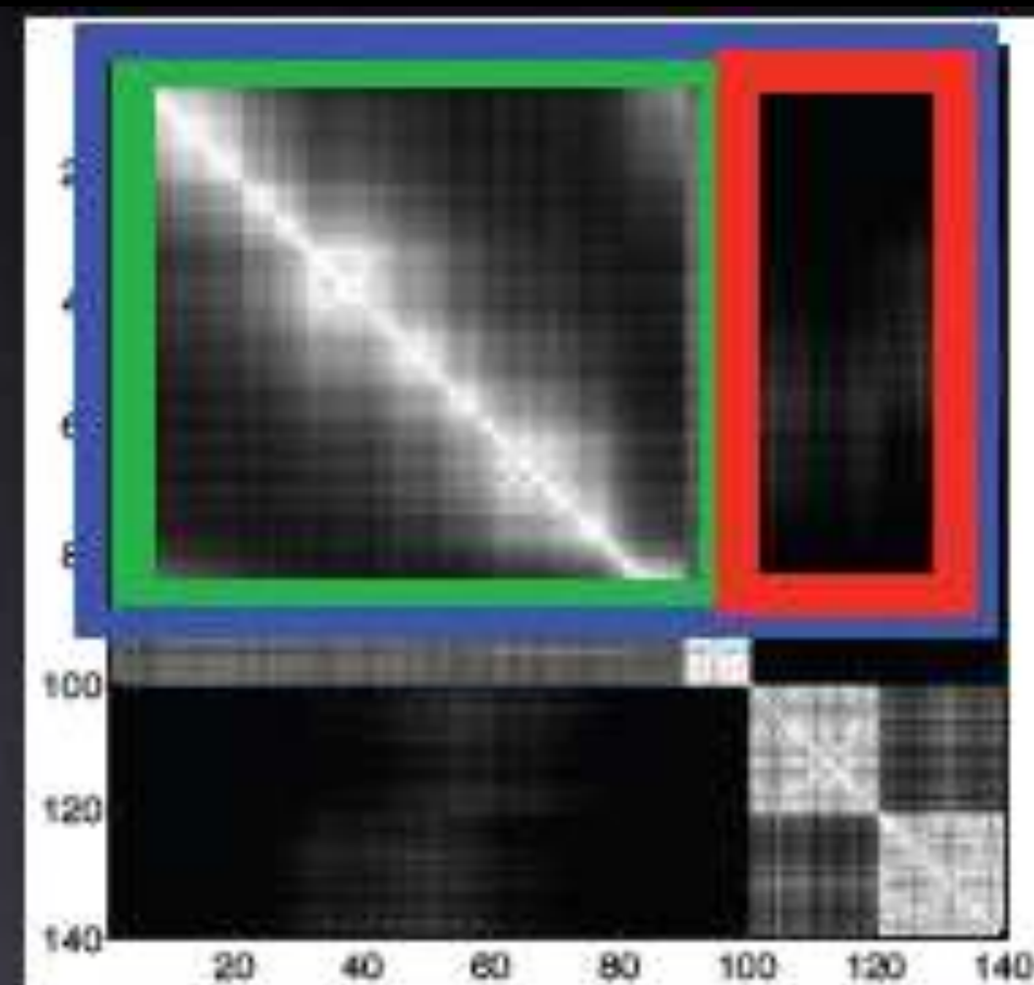
$$\text{Cut}(A,V-A) = x^T D x - x^T W x$$



$$Ncut(X) = \frac{1}{K}\sum_{l=1}^{K}\frac{cut(V_l, V - V_l)}{vol(V_l)}$$

$$= \frac{1}{K}\sum_{l=1}^{K}\frac{X_l^T(D - W)X_l}{X_l^T D X_l}$$

$X \in \{0,1\}^{N\times K}, X1_K = 1_N$



D-W often called the graph laplacian

becomes

$$Ncut(Z) = \frac{1}{K} tr(Z^T W Z) \qquad Z^T D Z = I_K$$

Rayleigh and Ritz Says:

Eigensolutions

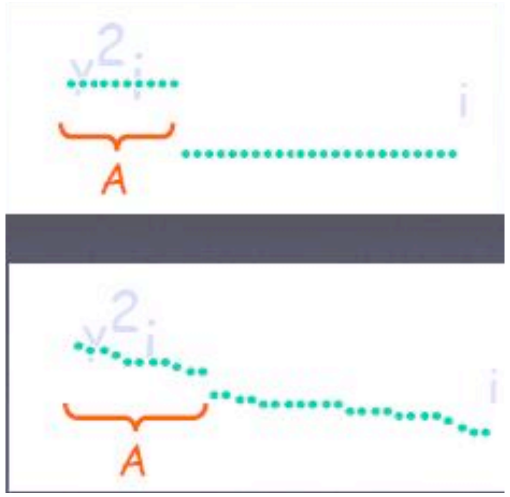
$$(D - W) z^* = \lambda D z^*$$

$$Z^* = [z_1^*, z_2^*, ..., z_k^*]$$

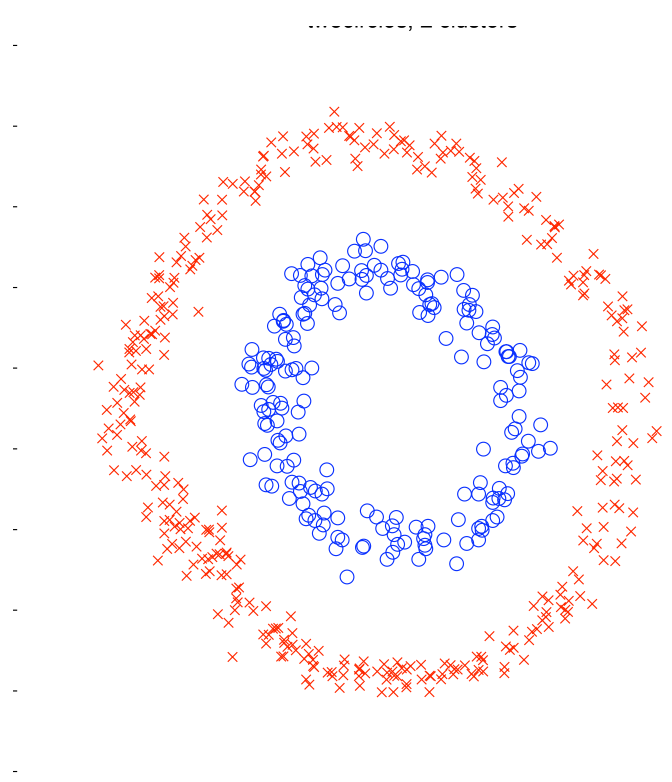
Image

Eigenvectors

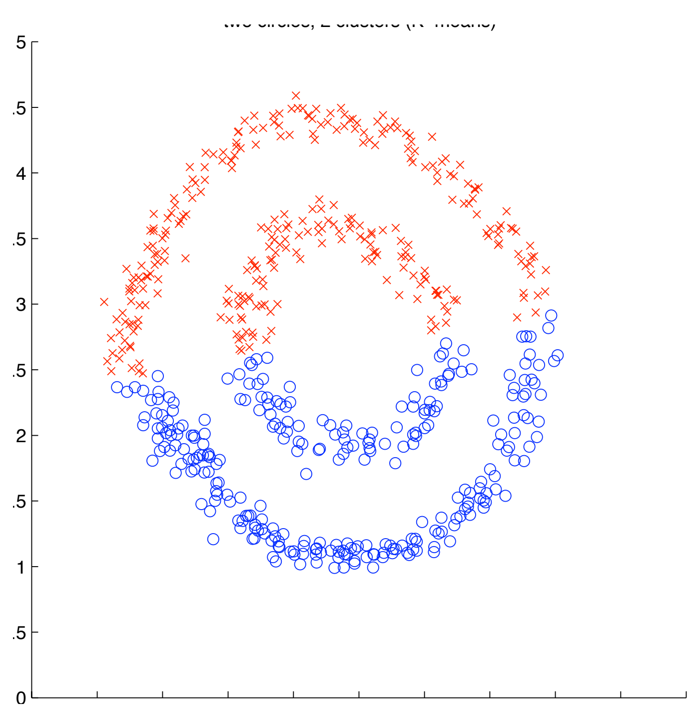
How to discretize eigenvectors?



Heuristic that works well: interpret k eigenvectors as coordinates for points in R^k and cluster again (using say, k-means)



K-means on eigenvectors
“spectral clustering”

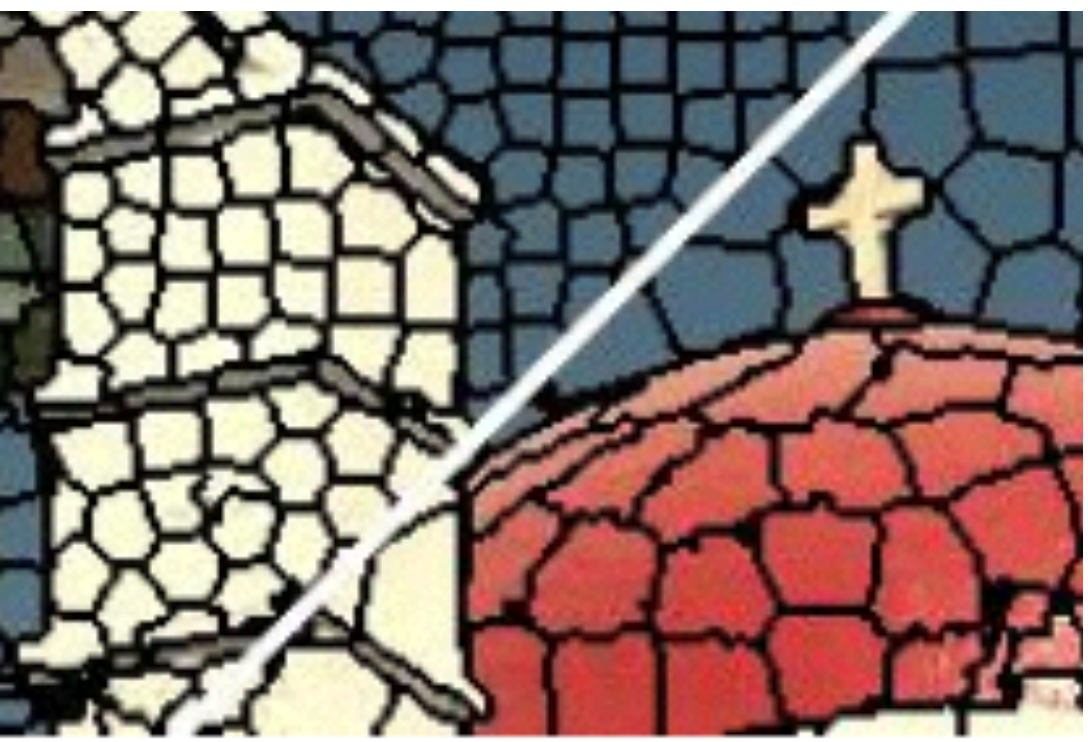


K-means

Intuition: eigenvectors capture transitive similarity properties

A look back

Superpixels
“oversegmentation”

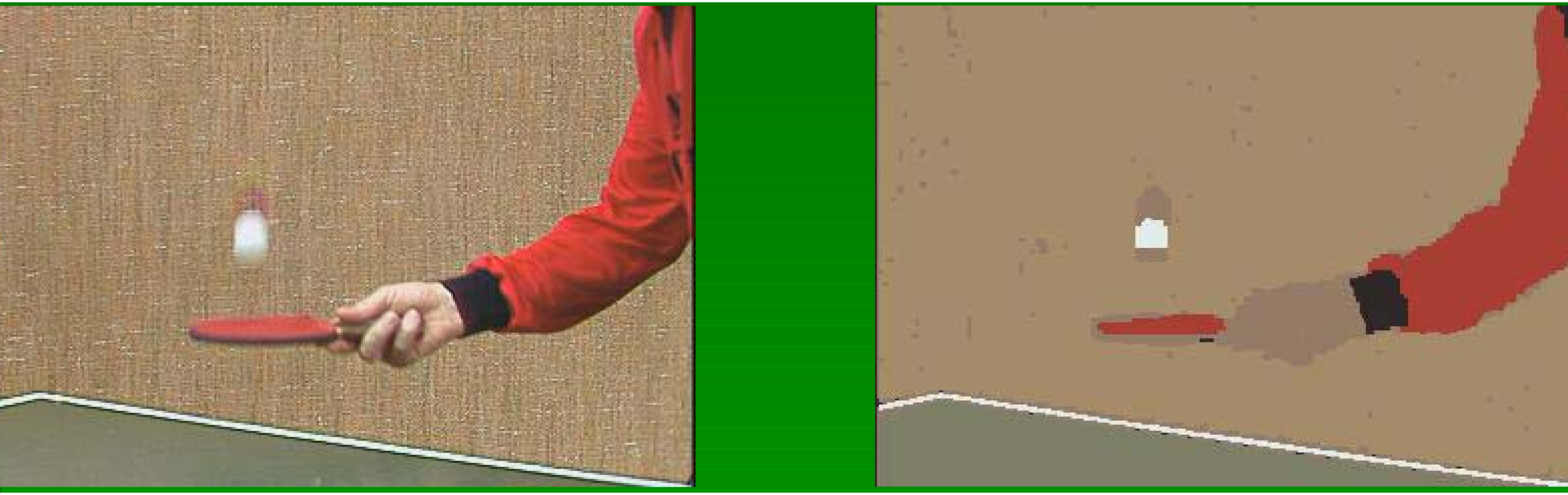


model-based clustering

Object proposals
“segmentation soup”

graphcuts

Mumford-Shah functional



$$E(f, C) = \int_{\Omega} (I(\mathbf{x}) - f(\mathbf{x}))^2 d\mathbf{x} + \int_{\Omega \setminus C} |\nabla f(\mathbf{x})| d\mathbf{x} + \int_C ds$$

f: piecewise smooth approximation of image I
C: set of curves around each segment

Optimize with gradient descent

Image snakes