# An Introduction to Image Segmentation

Caroline Pantofaru

# What is segmentation?

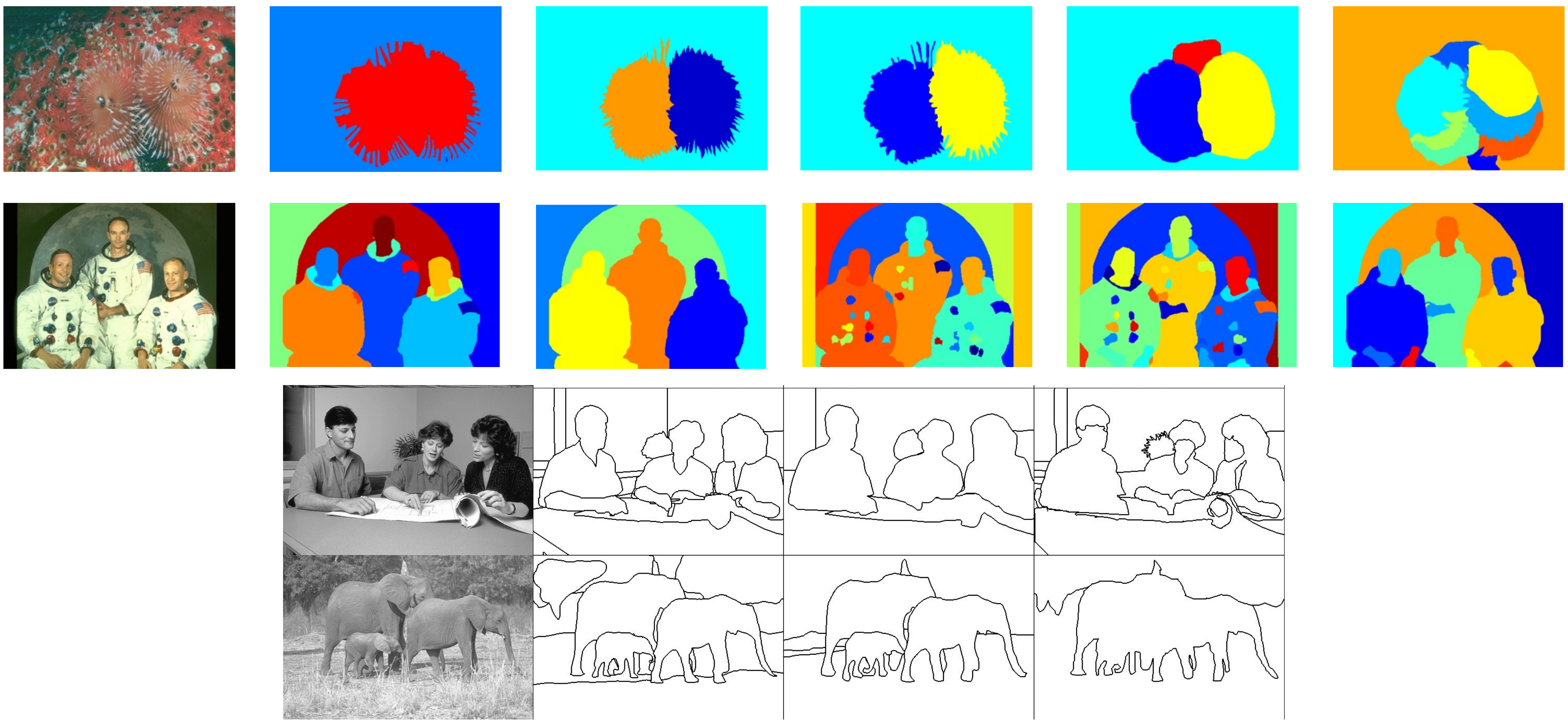
- What does it mean to be a good segmentation?

“You will be presented a photographic image. Divide the image into some number of segments, where the segments represent “things” or “parts of things” in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all segments have approximately equal importance.” -Martin

- Divide the pixels in an image into clusters in some natural way...

# Berkeley Segmentation Dataset

- 300 images
- 5-7 ground truth (human) segmentations per image
- [Martin *et al.* ICCV'01]



# Gestalt Laws of Perceptual Organization

Proximity

Closure

Similarity

Continuity

Images from : http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc.htm

# Today

- Unsupervised segmentation
- No a priori knowledge about objects
- Mean Shift and Normalized Cuts

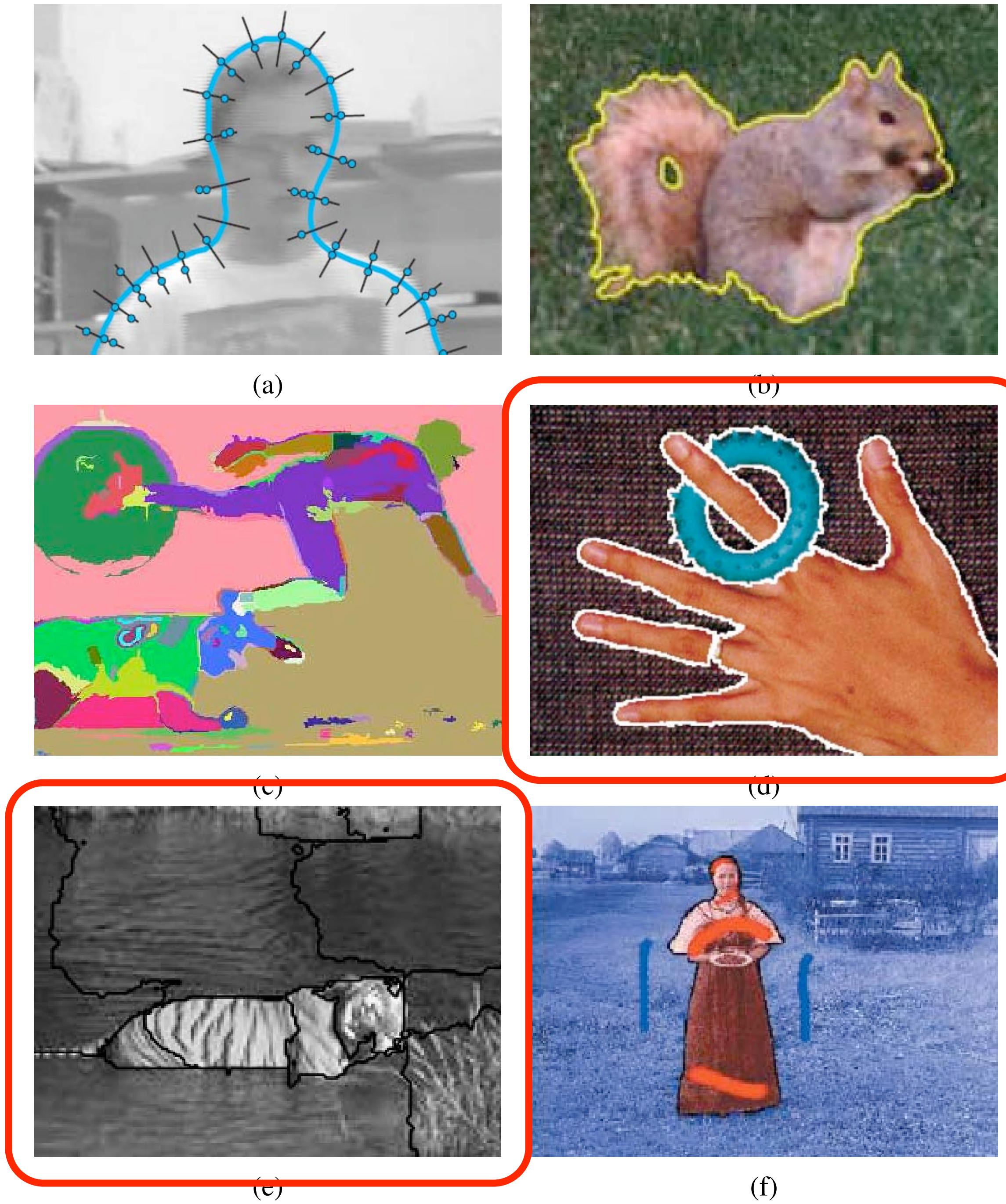
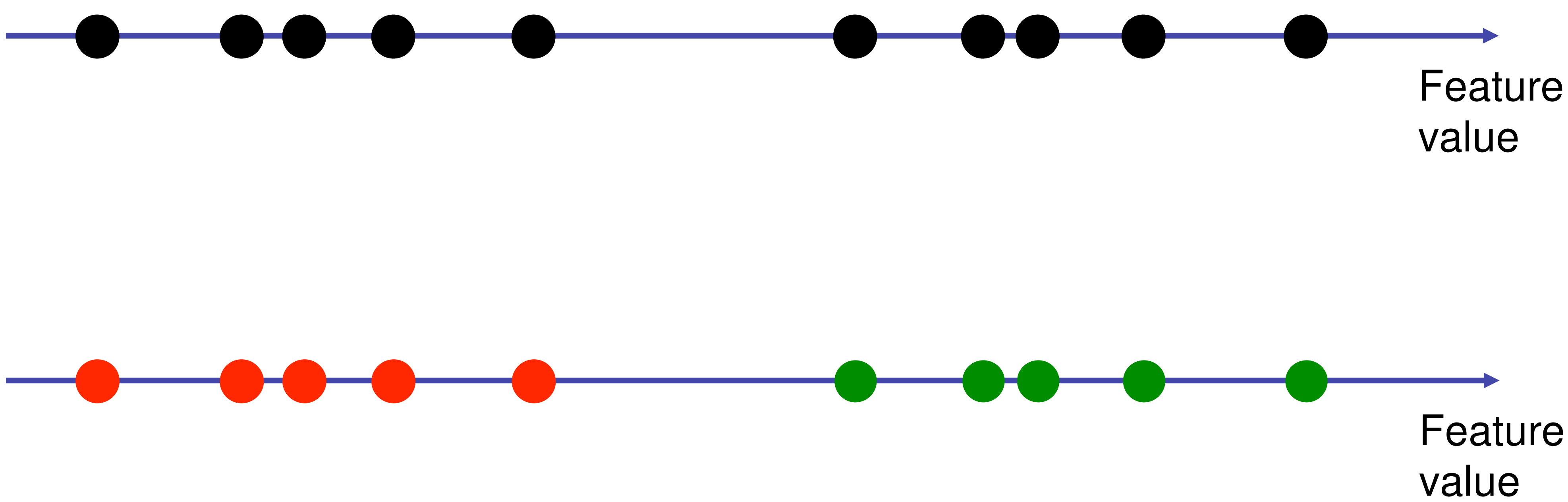


Figure 5.1: *Some popular image segmentation techniques: (a) active contours (Isard and Blake 1998); (b) level sets (Cremers* et al. *2007); (c) graph-based merging (Felzenszwalb and Huttenlocher 2004b); (d) mean shift (Comaniciu and Meer 2002); (e) texture and intervening contour-based normalized cuts (Malik* et al. *2001); (f) binary MRF solved using graph cuts (Boykov and Funka-Lea 2006).*

# Mean Shift

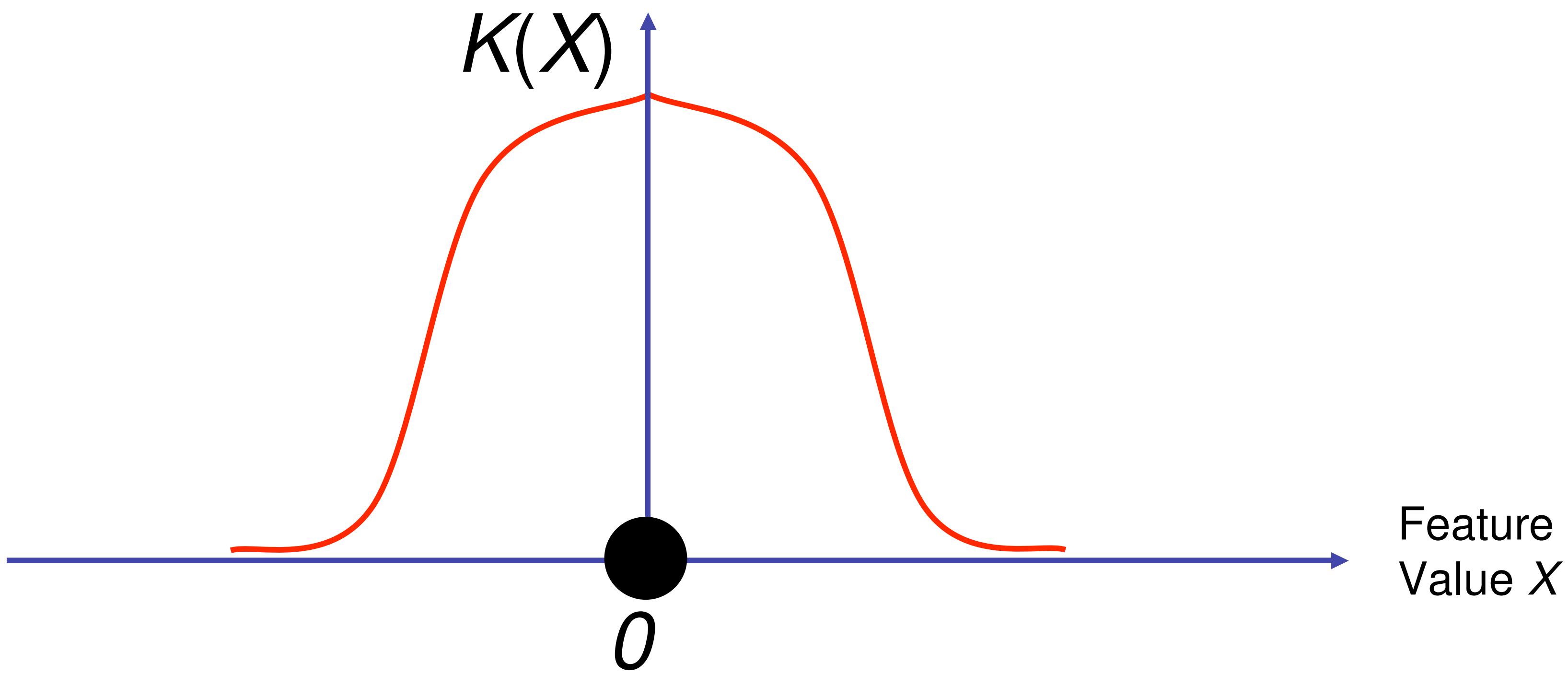
- Probabilistic technique.
- D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis”. IEEE Trans. PAMI, Vol. 24, No. 5, 2002.

# Clustering in 1-D




- Cluster around the high-density parts of the feature space.

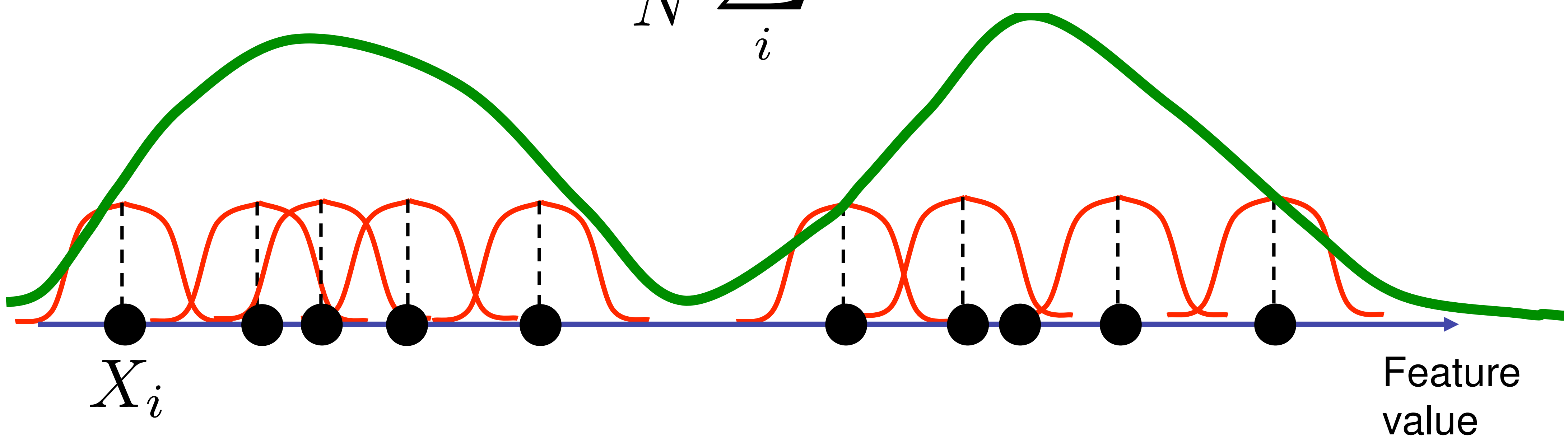
# Kernel




- Parzen window K:
  - is maximum at 0,
  - decays to 0 far from 0,
  - is symmetric,
  - integrates to 1.

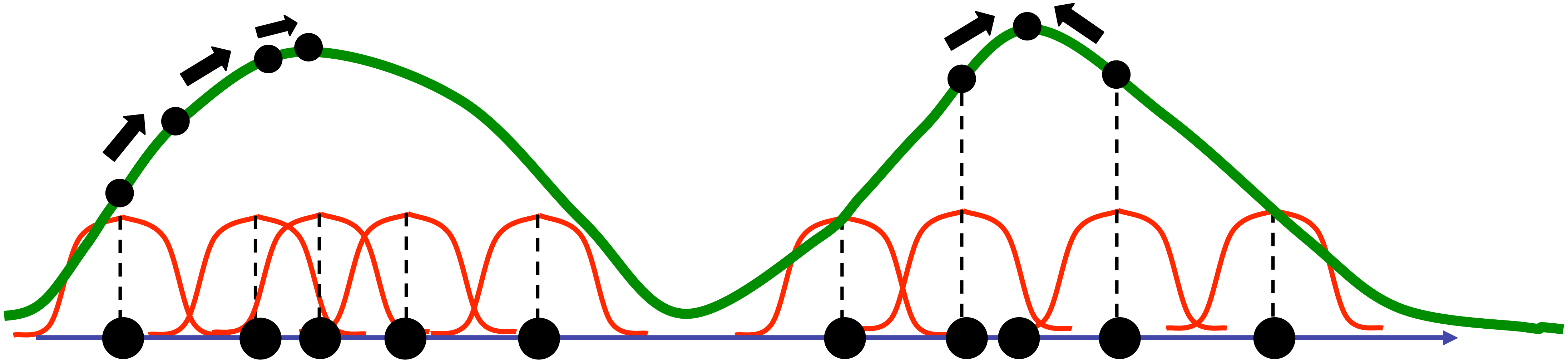
# Kernel Density Estimation

$$f(X) = \frac{1}{N} \sum_i K(X - X_i)$$




- f(x) approximates the probability of X given the data.
- Maxima of the pdf f = Modes of the density = Clusters in the data

# Finding the modes



For i=1 ... N

$$X \leftarrow X_i$$

Do

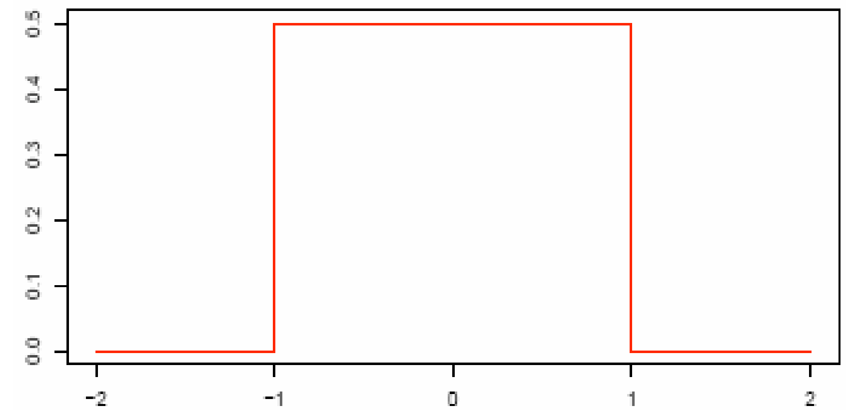
$$X \leftarrow X_i + \nabla f\left(X\right) = X_i + \frac{1}{N}\sum_i \nabla K(X - X_i)$$

Until X converges

# Possible kernels

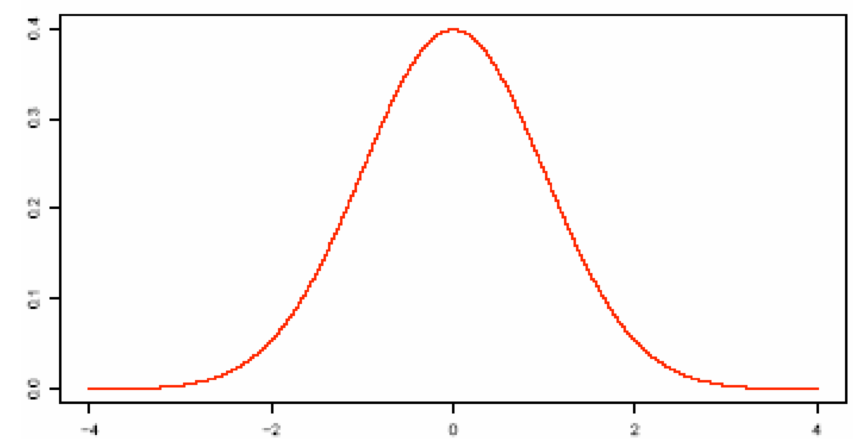
Uniform:

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



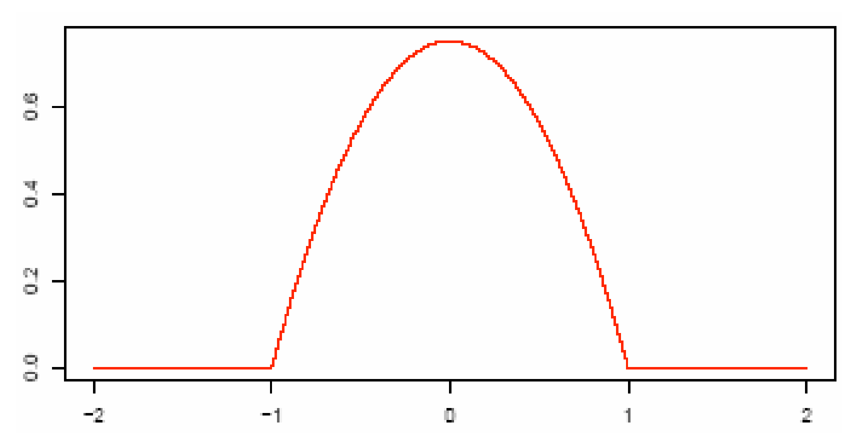

Gaussian:

$$K_N(\mathbf{x}) = c \cdot \exp\left( -\frac{1}{2} \|\mathbf{x}\|^2 \right)$$



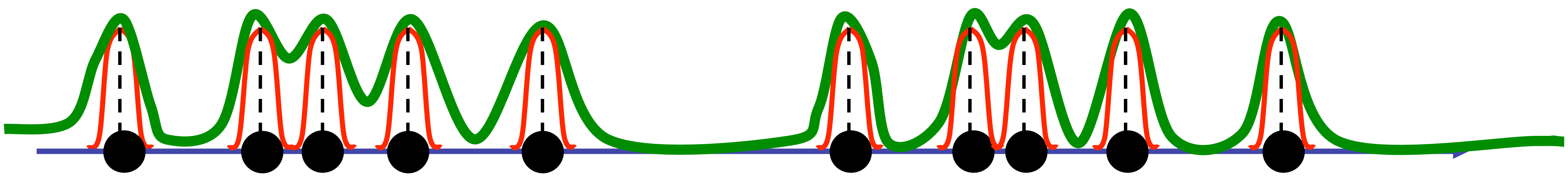

Epanechnikov:

$$K_E(\mathbf{x}) = \begin{cases} c\left(1 - \|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

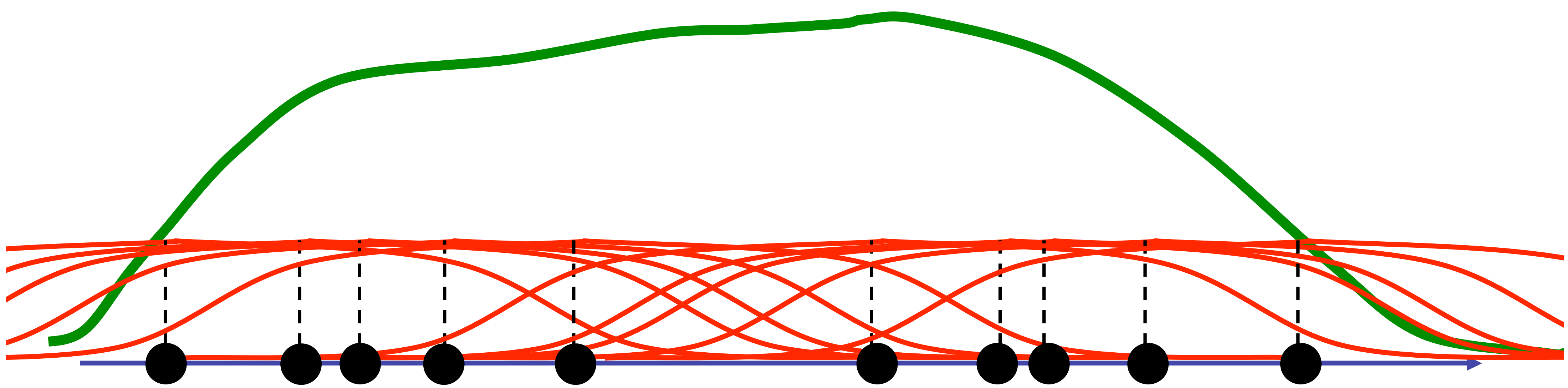


# Kernel size

Too small:



Too big:

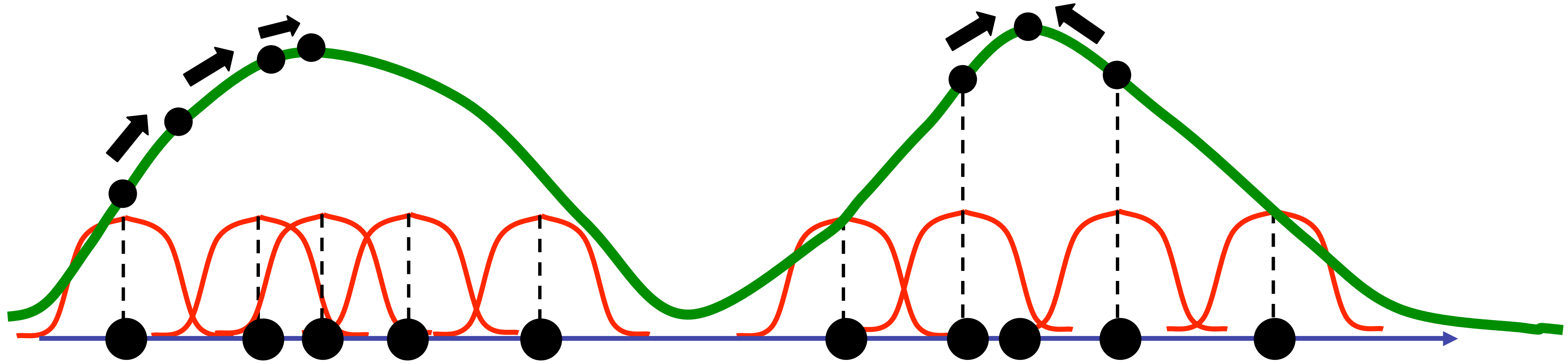


Control the size with the “bandwidth” *h*.

# Kernels

- By definition: $K(X) = ck\left(\left\|\frac{X}{h}\right\|^2\right)$
- Gaussian kernel: $k(t) = e^{-t/2}$
- Epanechnikov kernel: $k(t) = \begin{cases} (1-t) \text{ if } |t| < 1 \\ 0 \end{cases}$

# Back to the gradient



Climbing: $X \leftarrow X_i + \nabla f(X) = X_i + \frac{1}{N}\sum_i \nabla K(X - X_i)$

Define: $g(t) = -k'(t)$

$$\nabla K(X - X_i) = \nabla\left( ck\left( \frac{\|X - X_i\|^2}{h^2} \right)\right) = \frac{2c}{h^2}(X_i - X)g\left( \frac{\|X - X_i\|^2}{h^2} \right)$$

# Almost there...

The whole gradient:

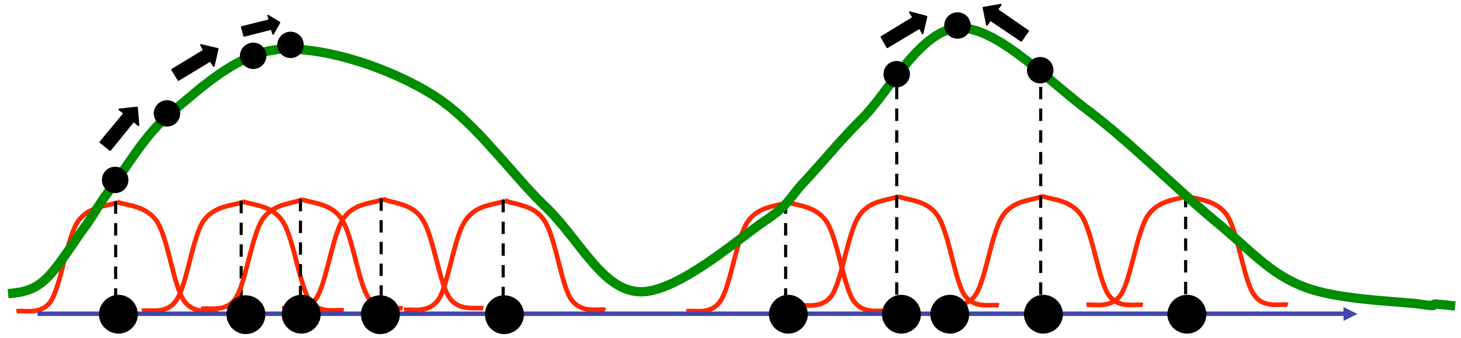
$$\nabla f(X) = \frac{2c}{Nh^2} \sum_i (X_i - X) g\left( \frac{\|X - X_i\|^2}{h^2} \right)$$

$$\nabla f(X) = \left( \frac{2c}{Nh^2} \sum_i g\left( \frac{\|X - X_i\|^2}{h^2} \right) \right) \left( \frac{\sum_i X_i g\left( \frac{\|X - X_i\|^2}{h^2} \right)}{\sum_i g\left( \frac{\|X - X_i\|^2}{h^2} \right)} - X \right)$$

The Mean Shift Vector!

Key: it points in the same direction as the gradient.

# The Full Algorithm



For i=1 ... N

$$X \leftarrow X_i$$

Do

$$X \leftarrow X + M(X) = \frac{\sum_i X_i g\left(\frac{\|X - X_i\|^2}{h^2}\right)}{\sum_i g\left(\frac{\|X - X_i\|^2}{h^2}\right)}$$

Until X converges

# This is easy!

Epanechnikov:

$$k(t) = (1 - t) \text{ if } |t| < 1, \text{ 0 otherwise}$$

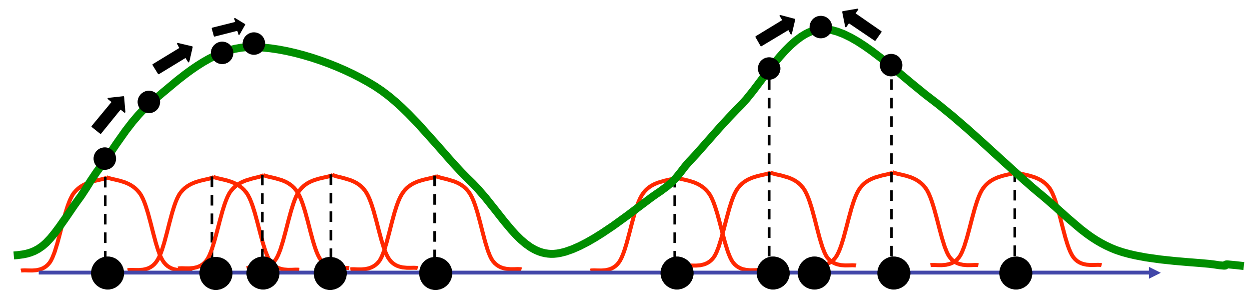
$$g(t) = k'(t) = 1 \text{ if } |t| < 1, \text{ 0 otherwise}$$

Mean shift vector:

$$\frac{\sum_i X_i g\left(\frac{\|X - X_i\|^2}{h^2}\right)}{\sum_i g\left(\frac{\|X - X_i\|^2}{h^2}\right)} = \frac{\sum_{\|X - X_i\| < h} X_i}{\sum_{\|X - X_i\| < h} 1} = \frac{\sum_{\|X - X_i\| < h} X_i}{N_h(X)}$$

Mean of the points within radius h!

# Mean Shift



For i=1 ... N

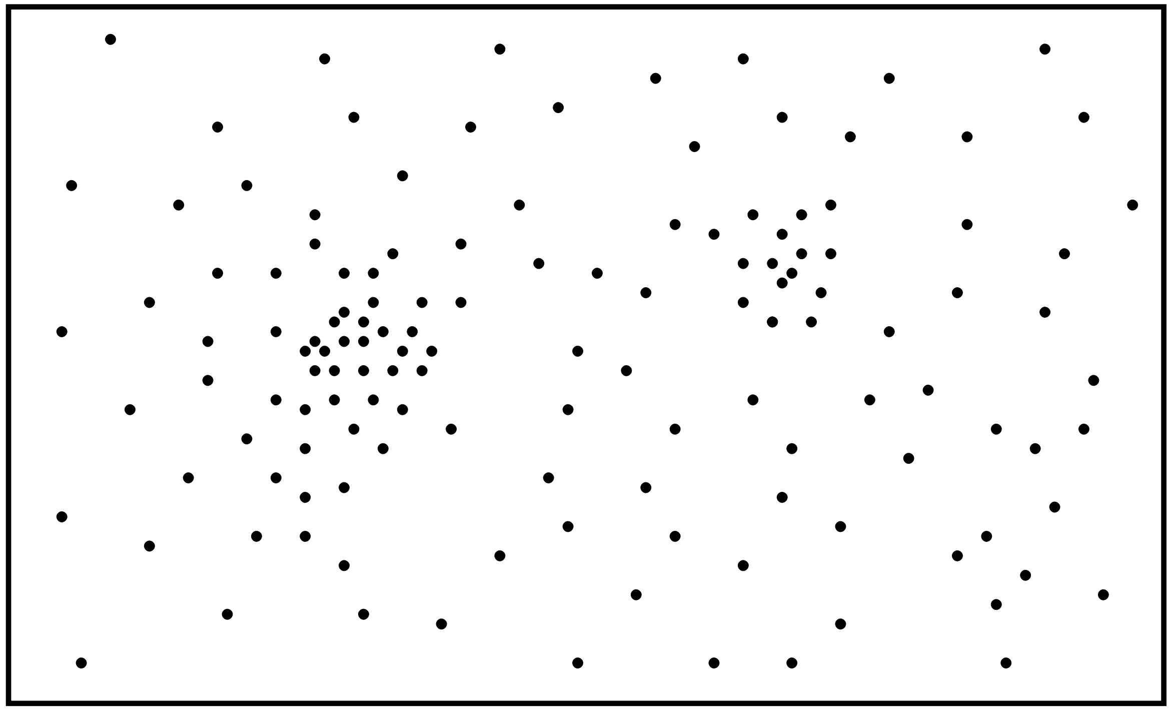
$$X \leftarrow X_i$$

Do

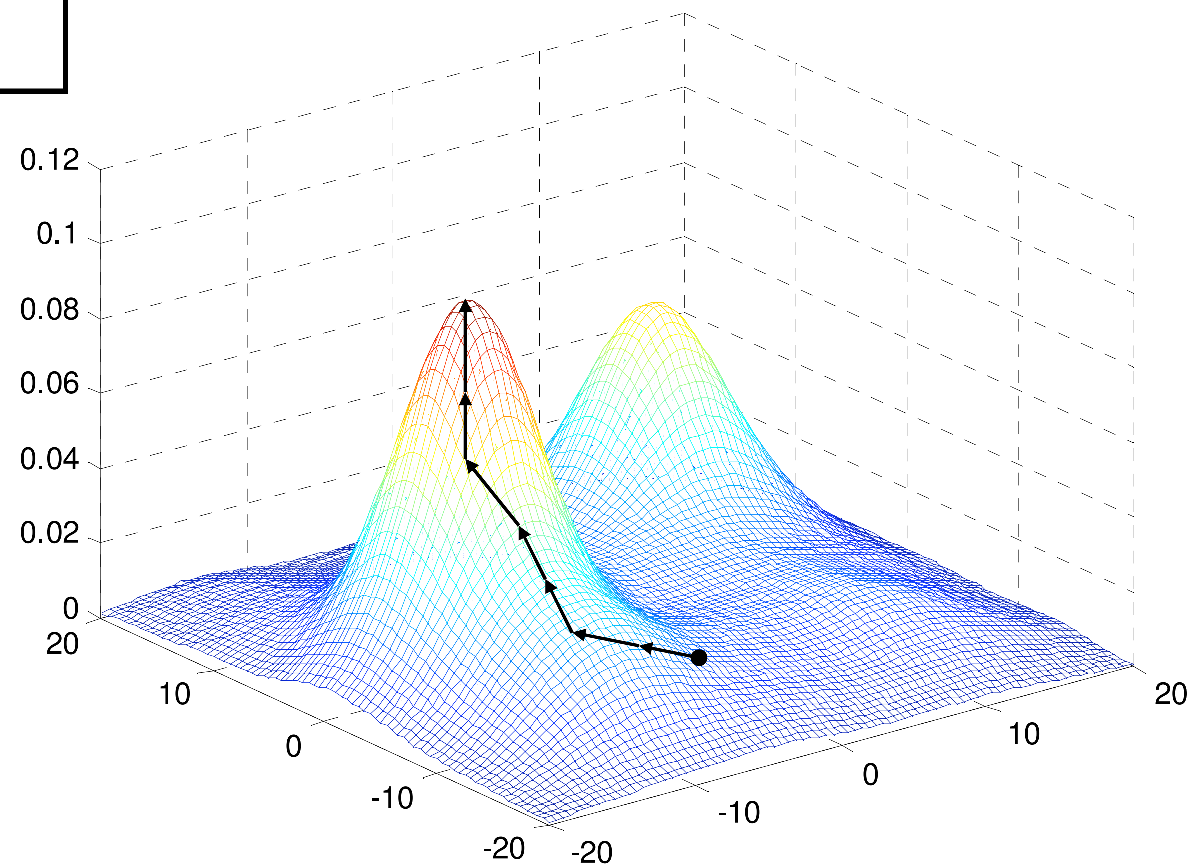
$$X \leftarrow X + M(X) = \frac{\sum_{||X - X_i|| < h} X_i}{N_h(X)}$$

Until X converges

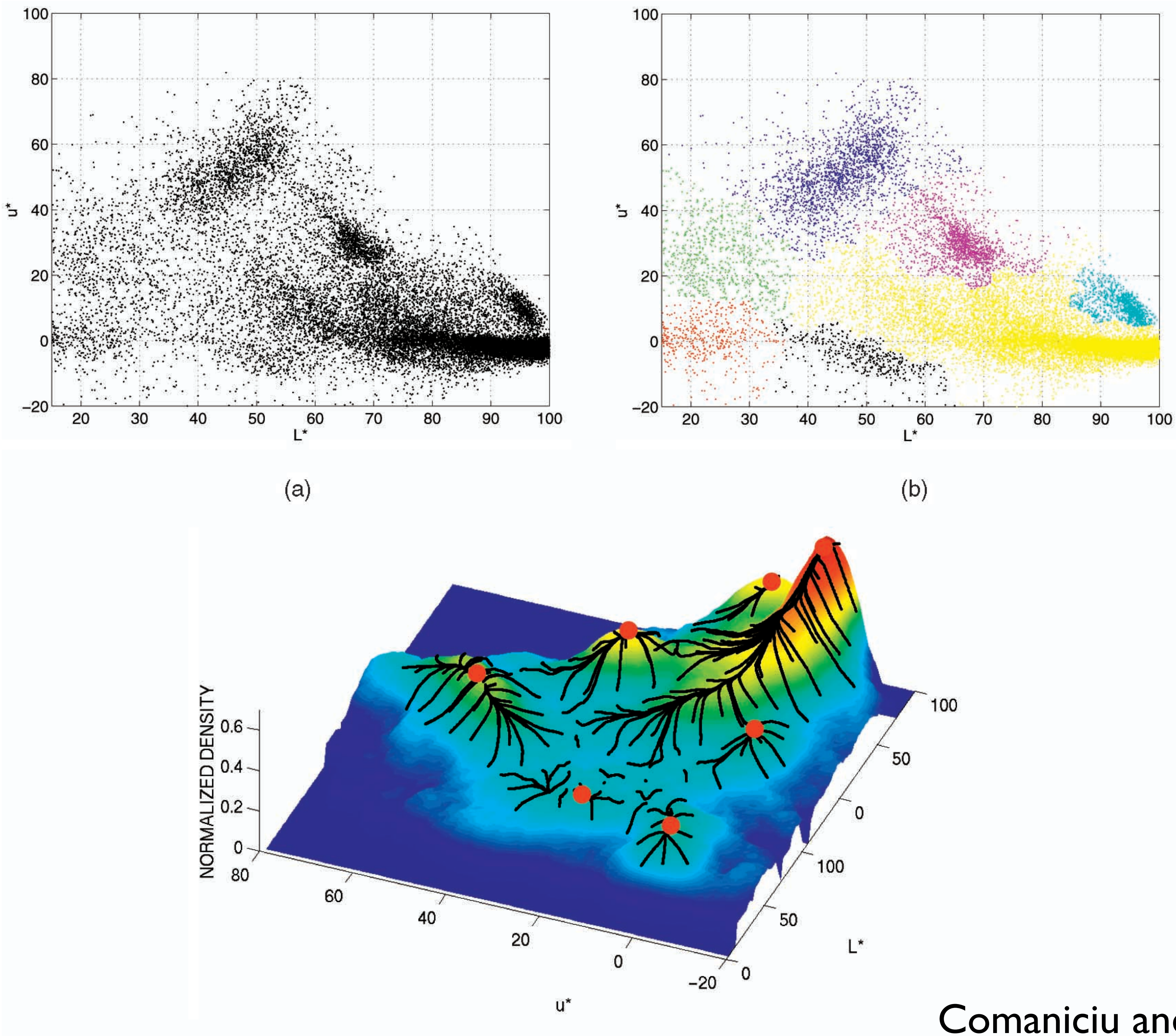
# 2D



$$f(X) = \frac{c}{N} \sum_{i=1}^{N} k\left( \left\| \frac{X - X_i}{h} \right\|^2 \right)$$



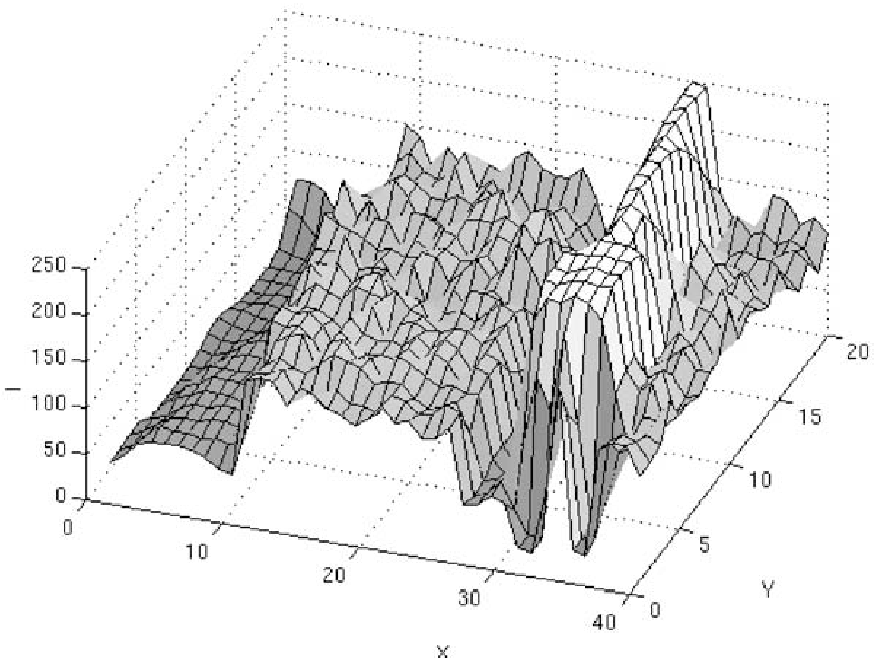
# 2D Example



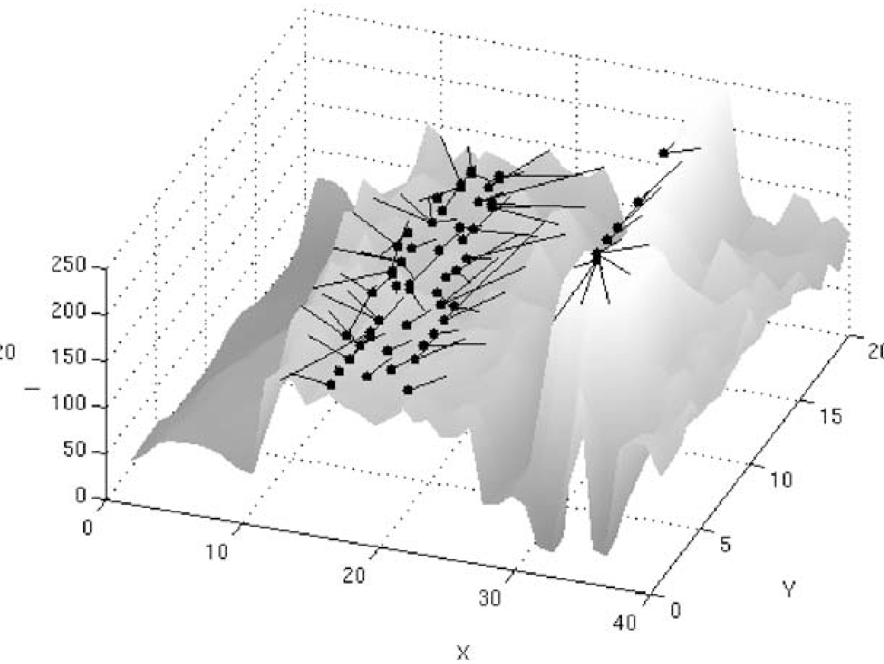

(c)

Comaniciu and Meer 2002

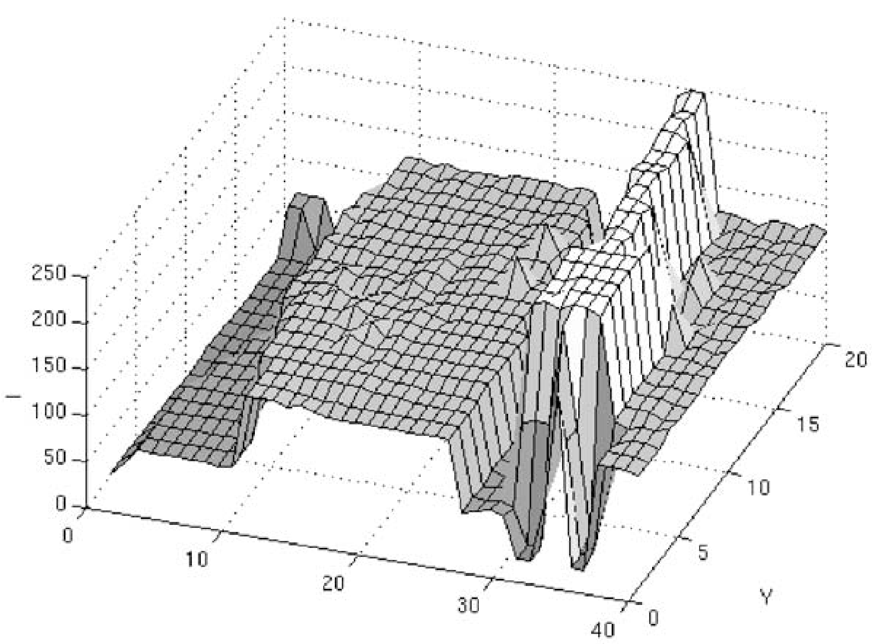
# Still need to cluster!



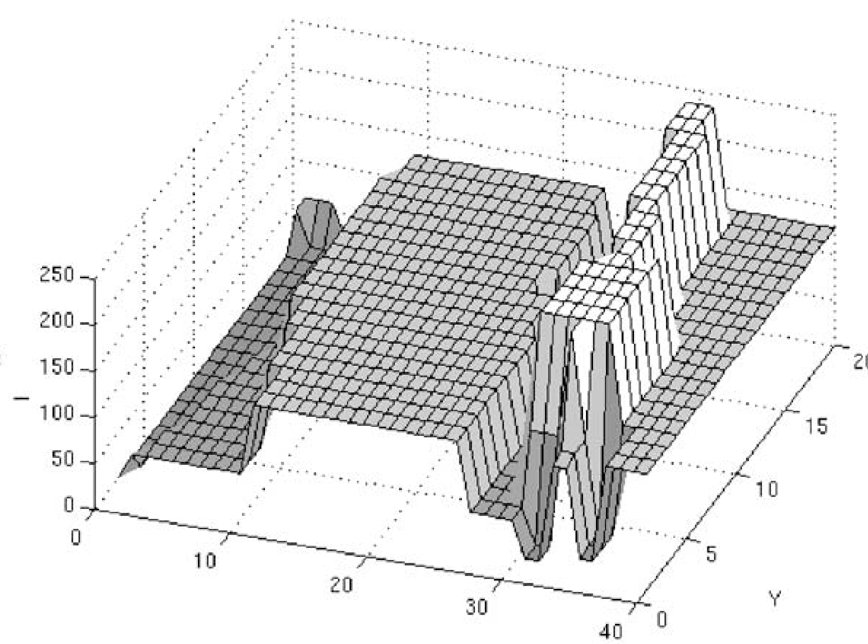

Original data




Mean shift iterations




Filtered data




Clustered data

Comaniciu and Meer 2002

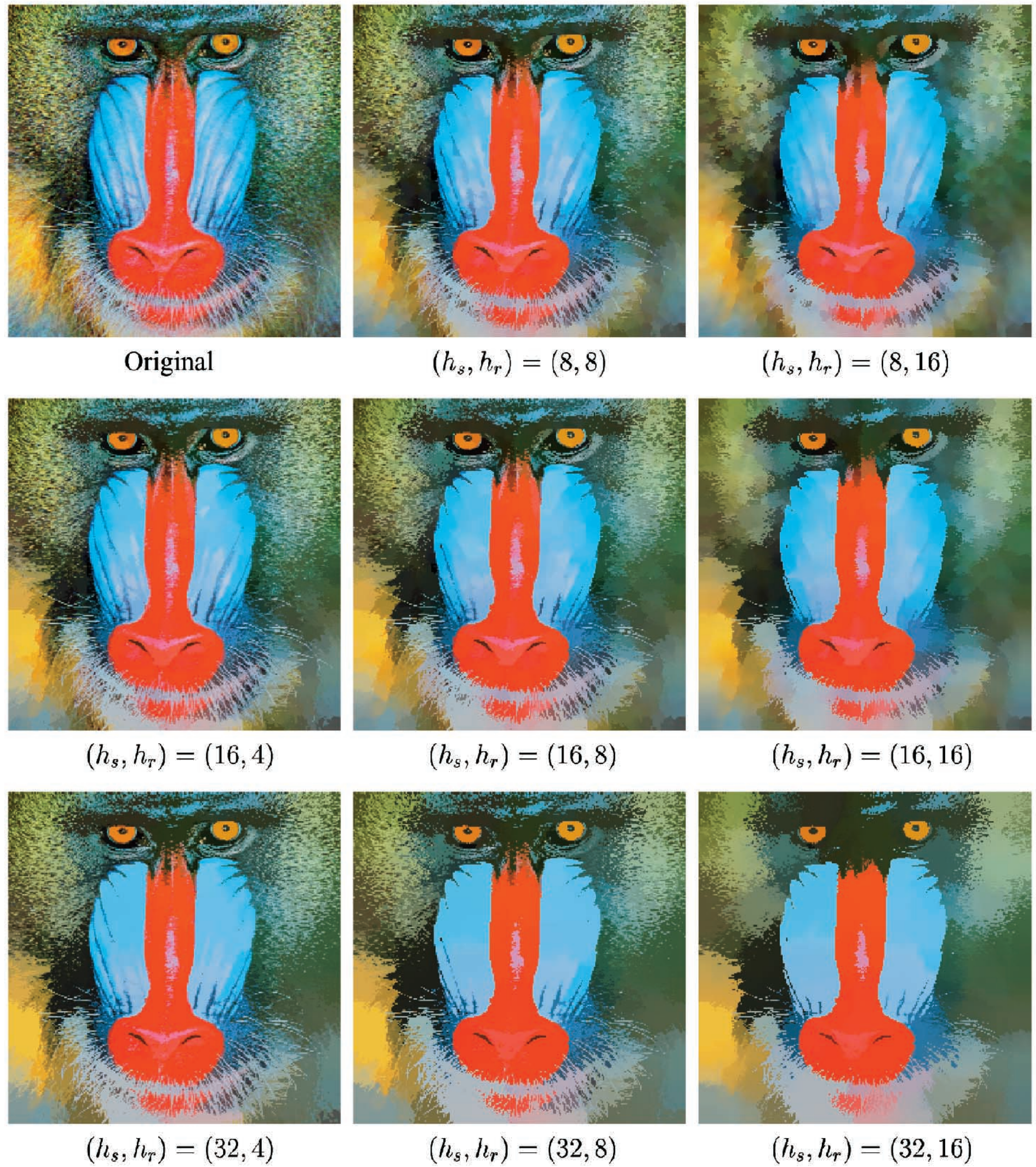
# Example: Color Segmentation

- Features: (x,y) position, (L,u,v) color
- Mean shift in 5D

$$K_{h_{pos}h_{color}} = ck\left(\frac{||X_{pos}||^2}{h_{pos}^2}\right)k\left(\frac{||X_{color}||^2}{h_{color}^2}\right)$$

- For each pixel, find the corresponding mode.
- Group all of the pixels corresponding to the same mode together.
- Don't need to specify the number of clusters, do need to specify the bandwidths.

# Filtering with different bandwidths



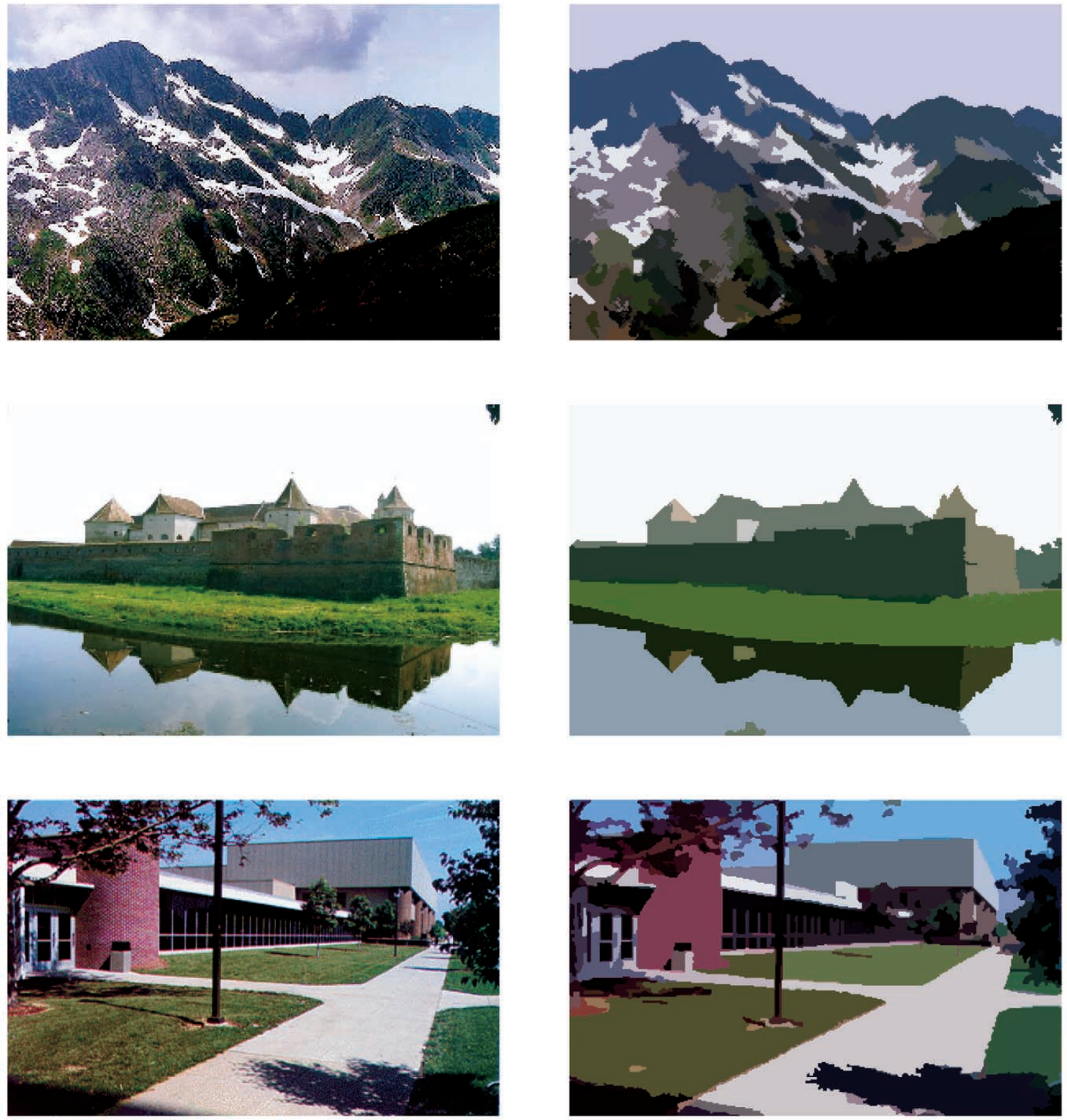

Comaniciu and Meer 2002

# Results



Comaniciu and Meer 2002

# Results



Comaniciu and Meer 2002

# References

- http://www.caip.rutgers.edu/riul/research/code.html
  - Papers and code (the EDISON system)
- D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis". IEEE Trans. PAMI, Vol. 24, No. 5, 2002.

# Normalized Cuts

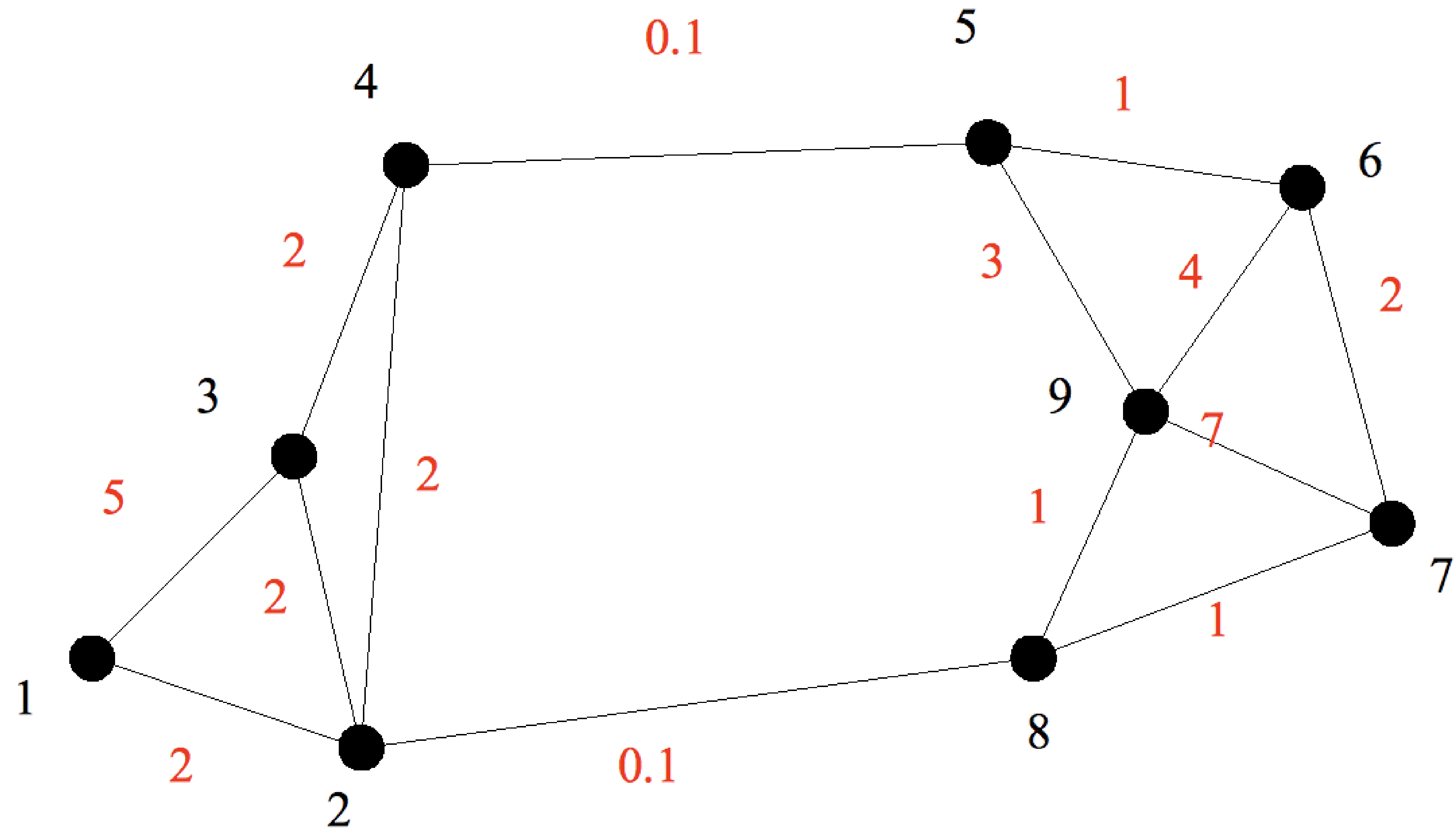
- Spectral graph technique.
- J. Shi and J. Malik, "Normalized cuts and image segmentation". IEEE Trans. on PAMI, 8(22), 2000.

# A Graph Problem

$G = (V, E)$

$V$ is a set of vertices with features $x_i$

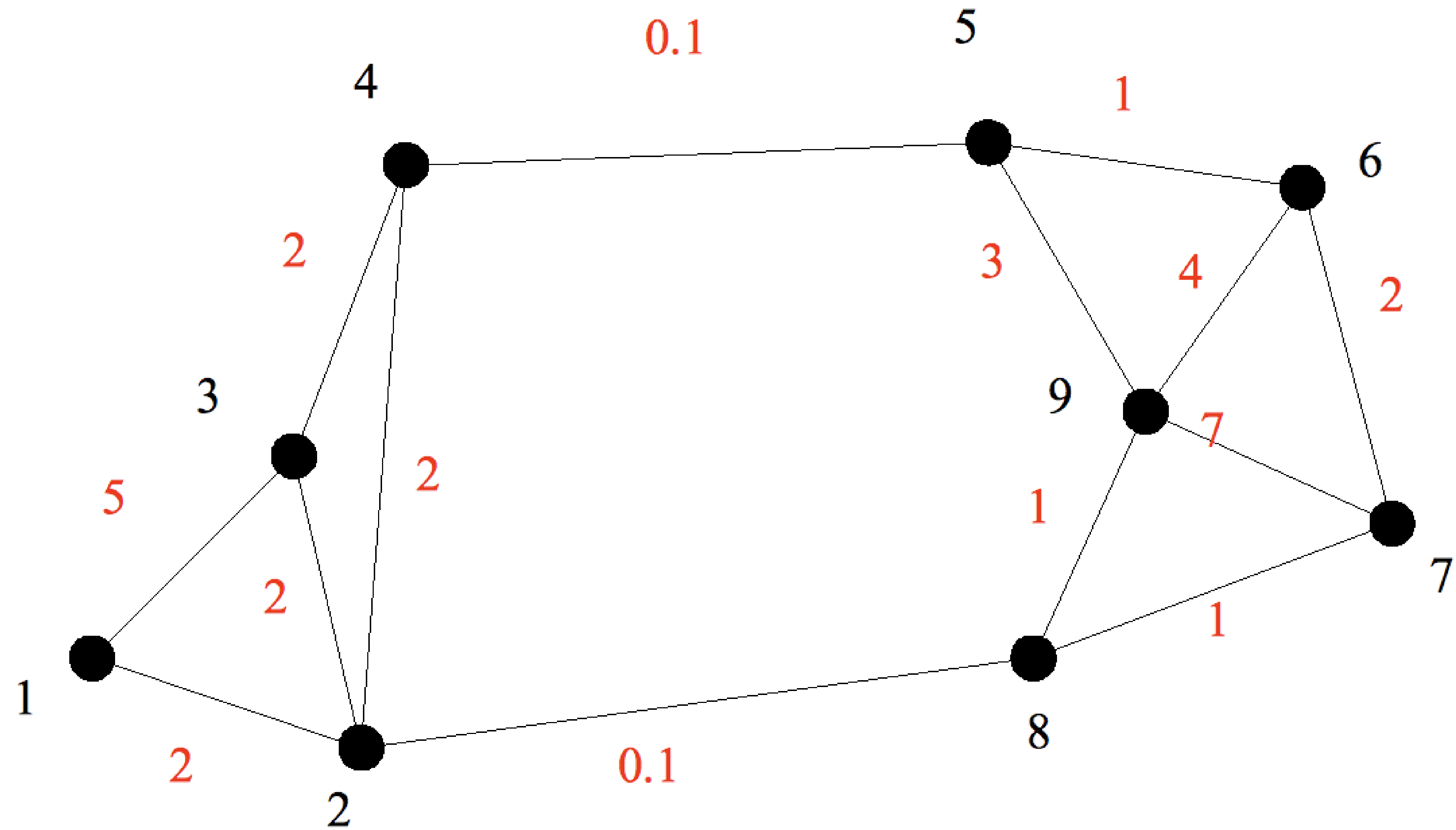
$E$ is a set of edges with weights (affinities) $m_{ij}$



# Segmentation as a Graph Problem

Find the connected components $C_i$

- separated by low-cost edges (*min cut*)
- with high in-cluster affinity (*high association*)



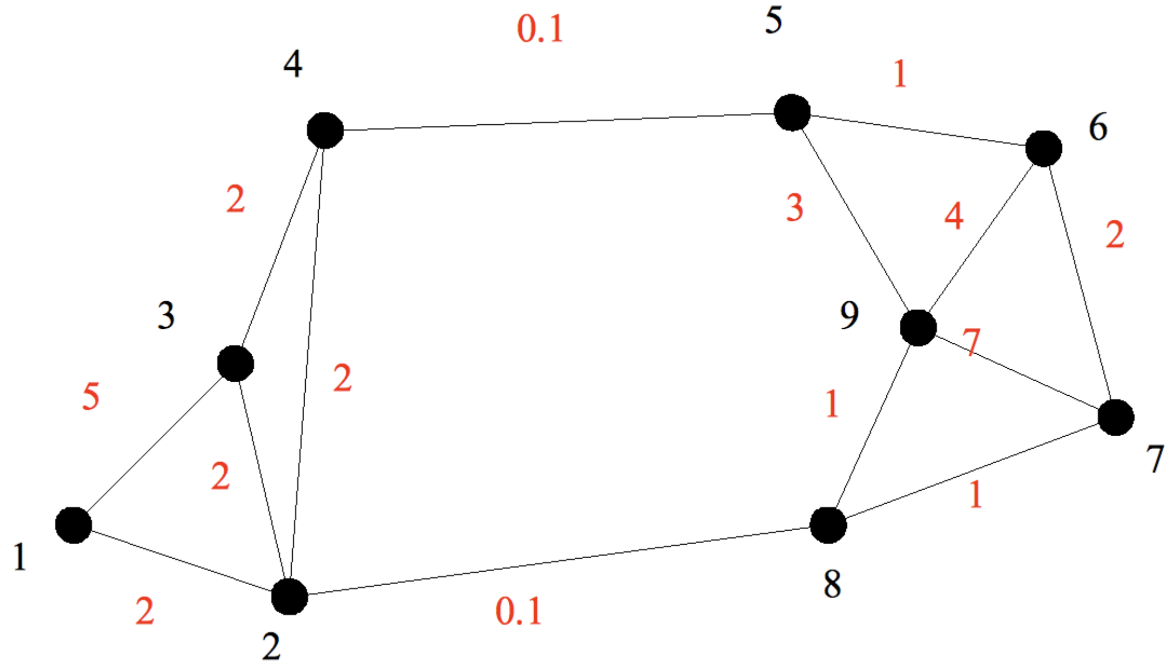
# What is the “affinity”?

(Usually) an exponentially decaying function of the “distance” between the features.

- intensity, color, position, edge position, edge length, edge orientation

$$m_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\sigma^2}\right)$$

# Affinity Matrix




$$M = \begin{pmatrix} 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0.1 & 0 \\ 5 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 7 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 3 & 4 & 7 & 1 & 0 \end{pmatrix}$$

# The Eigenvector Approach

- Weight (indicator) vector $w$ for connected component (cluster) $C$:

$$w_i = \begin{cases} 1 & \text{if} \quad i \in C \\ 0 & \text{if} \quad i \notin C \end{cases}$$

- Association of $C$:

$$w^t M w = \sum_{i,j \in C} m_{ij}$$

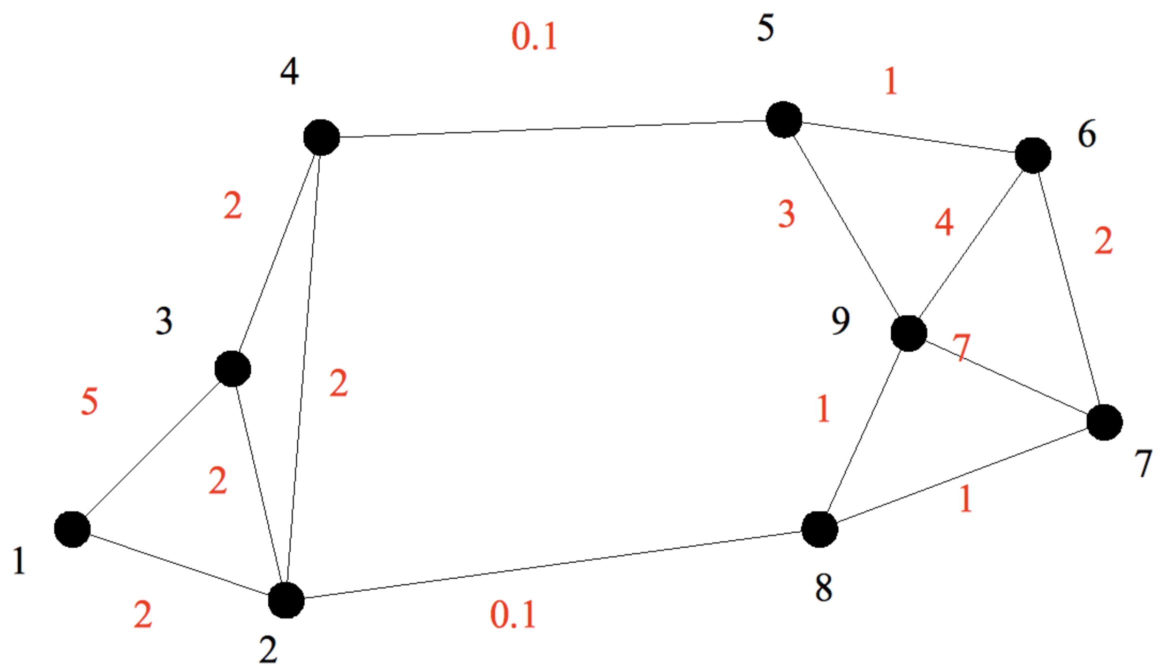
- Want to maximize the association.

# The Eigenvector Approach

- Problem: $w$ is binary
- Solution: let $w$ be continuous and threshold.
- Problem: the scale of $w$ isn't important.
- Solution: normalization $\max_w \frac{w^t M w}{w^t w}$
- Rayleigh's ratio theorem:

  For symmetric $M$, $\max_w \frac{w^t M w}{w^t w}$ occurs when $w_{max}$ is the eigenvector corresponding to the max eigenvalue $\lambda_{max}$ of $M$.

# Example

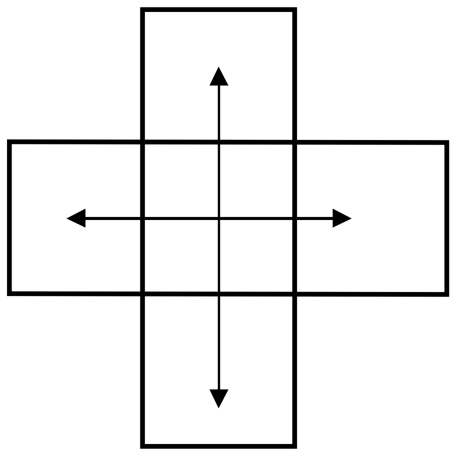



$$\lambda_1 = 9.8 \qquad \lambda_2 = 6.9$$

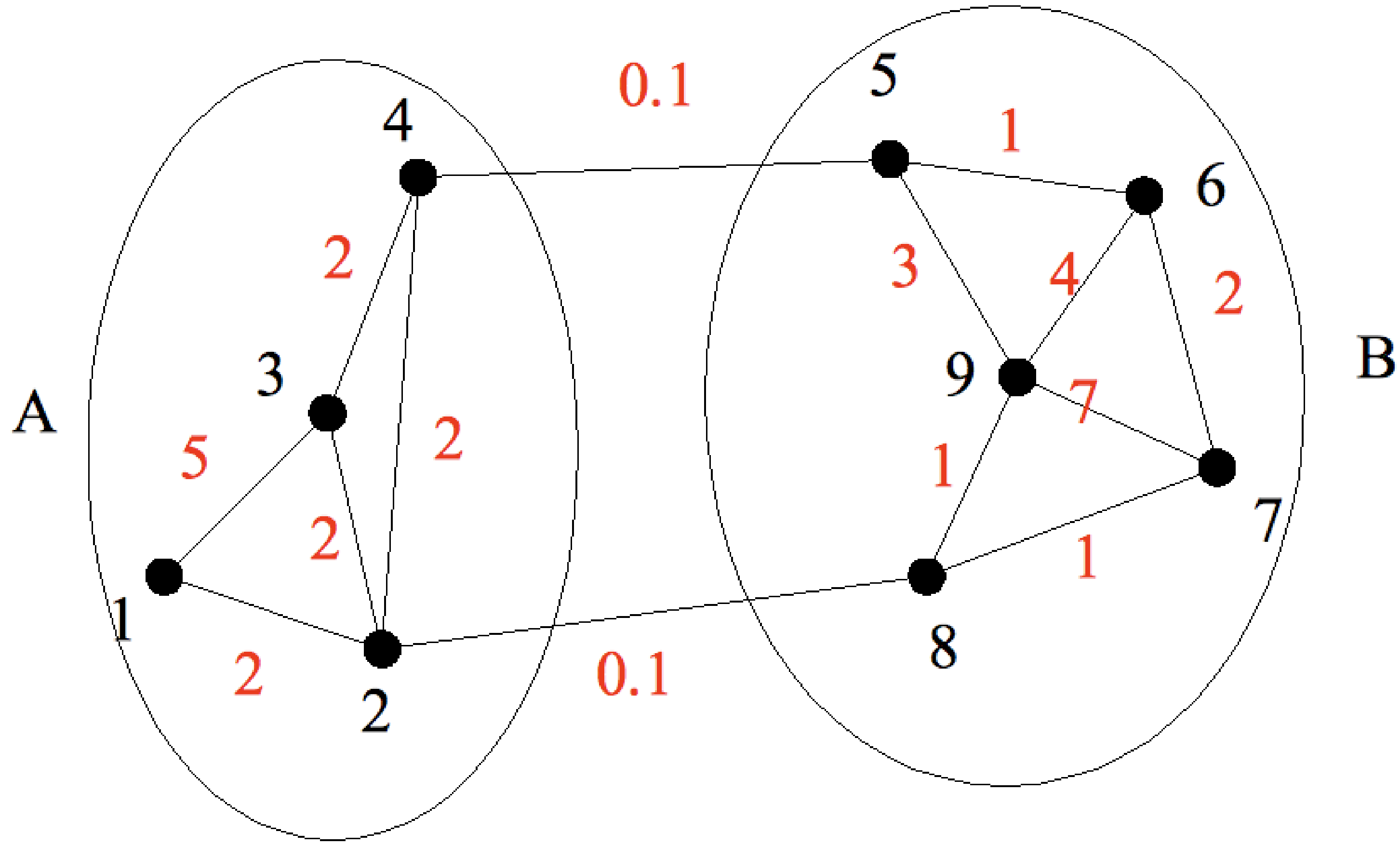
$$w_1 = \begin{vmatrix} 0.0050 \\ 0.0101 \\ 0.0058 \\ 0.0057 \\ 0.2437 \\ 0.4097 \\ 0.5673 \\ 0.1252 \\ 0.6596 \end{vmatrix} \quad w_2 = \begin{vmatrix} 0.5741 \\ 0.4326 \\ 0.6250 \\ 0.3043 \\ 0.0020 \\ -0.0030 \\ -0.0040 \\ 0.0052 \\ -0.0037 \end{vmatrix}$$

# The Eigenvector Approach

- Problem: *M* can be very large!
- Reality: *M* is usually very sparse.
  - N pixels → 4N entries
- Lanczos' algorithm



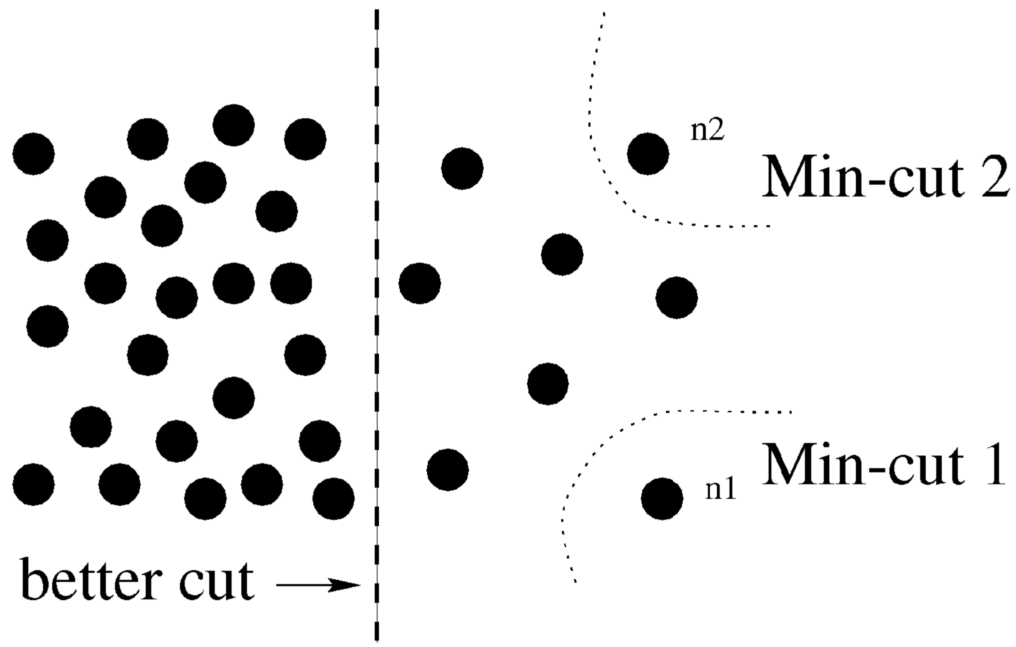
# The Minimum Cut




$$\text{cut}(A, B) = \sum_{i \in A, j \in B} m_{ij}$$

# The Problem with Min Cut

The min cut is not the "natural" solution.




Need to consider the association!

$$\mathrm{assoc}(A, V) = \sum_{i \in A, j \in V} m_{ij}$$

# Normalized Cut

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}$$

$$\text{assoc}(A, V) = \text{assoc}(A, A) + \text{cut}(A, B)$$

2-way segmentation: find *A*, *B* that minimize *Ncut*

NP-hard

# Exact (intractable) solution

$M$ is the affinity matrix

$D$ is the diagonal degree matrix

$y$ is the indicator vector

$$y_i = \{ \begin{array}{ll} 1 & (i \in A) \\ -b & (i \in B) \end{array} \qquad b = \frac{\text{assoc}(A,V)}{\text{assoc}(B,V)}$$

The min cut (A,B) minimizes:

$$\frac{y^t(D-M)y}{y^tDy}$$

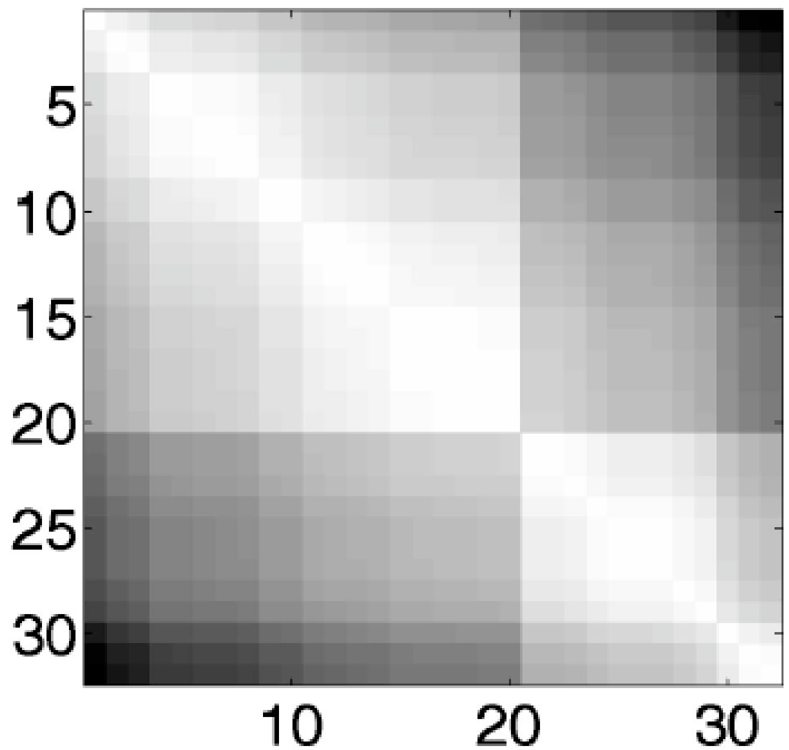
# Approximate (tractable) solution

- Solve for the indicator variable:
  - Generalized eigenvalues. Find the min $\lambda_{min}$ and corresponding $y_{min}$ such that:

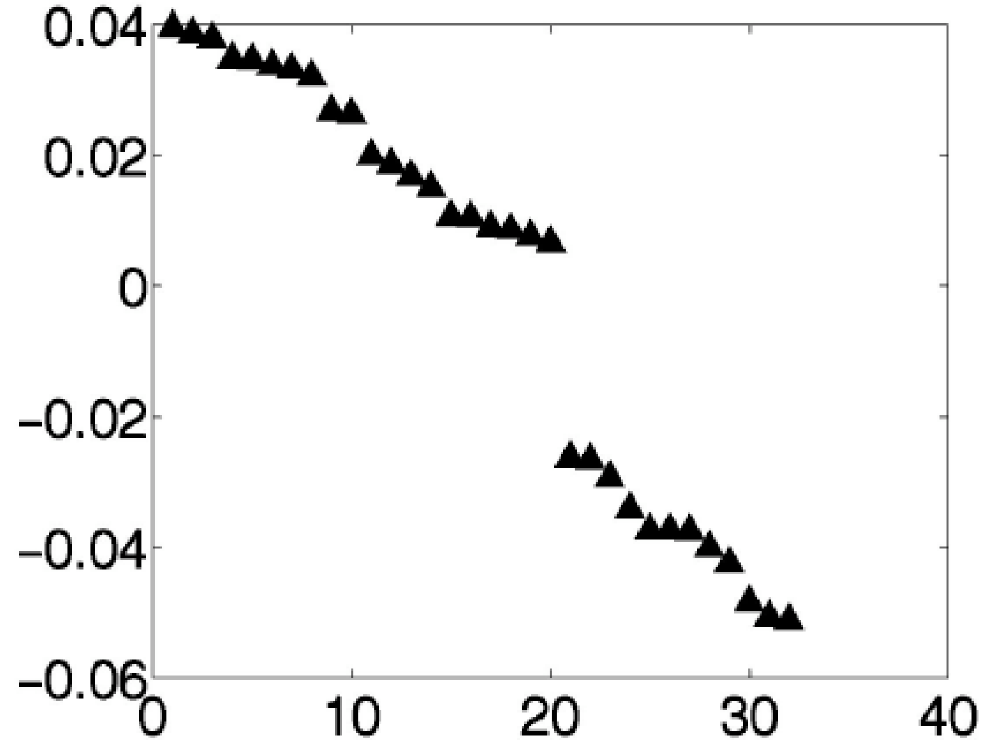
$$(D - M)y = \lambda D y$$

- Threshold
  - Vertices (nodes) $i$ such that $y_{min}^{i}$ is large are in the cluster.
- Note: the min eigenvalue corresponds to $(A, B) = (\emptyset, V)$, so use the second smallest

# Example




M



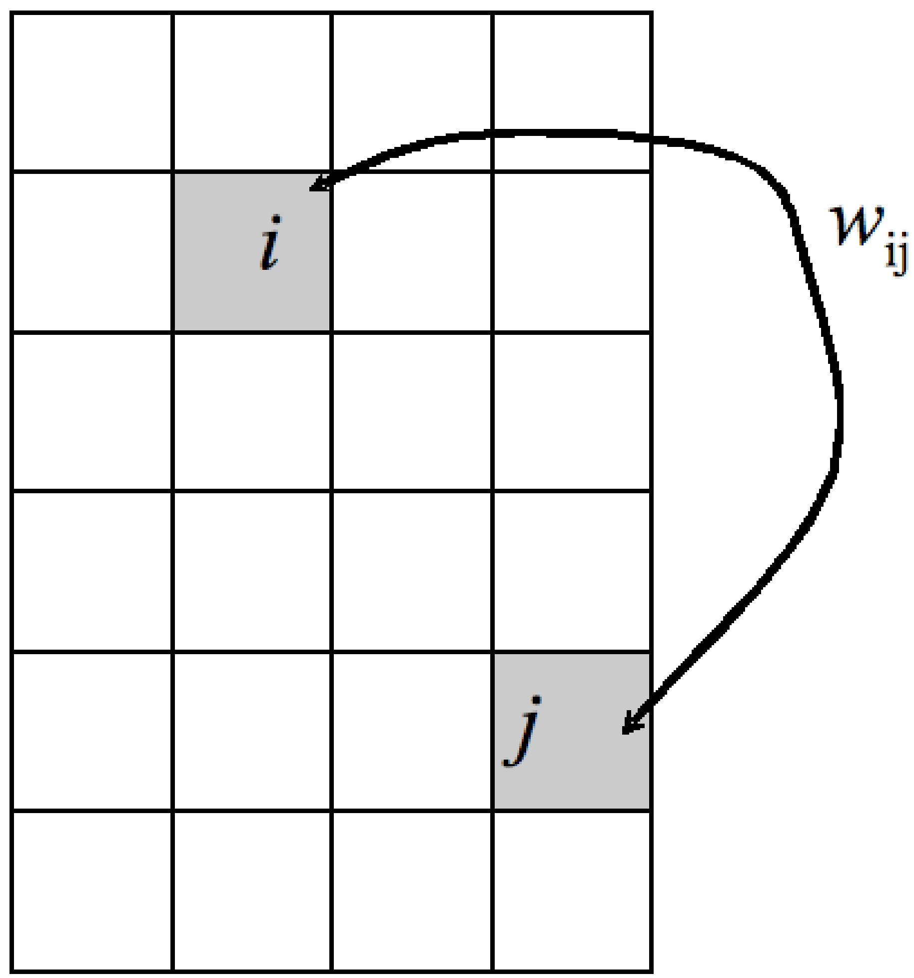

Second eigenvector

# Practicalities

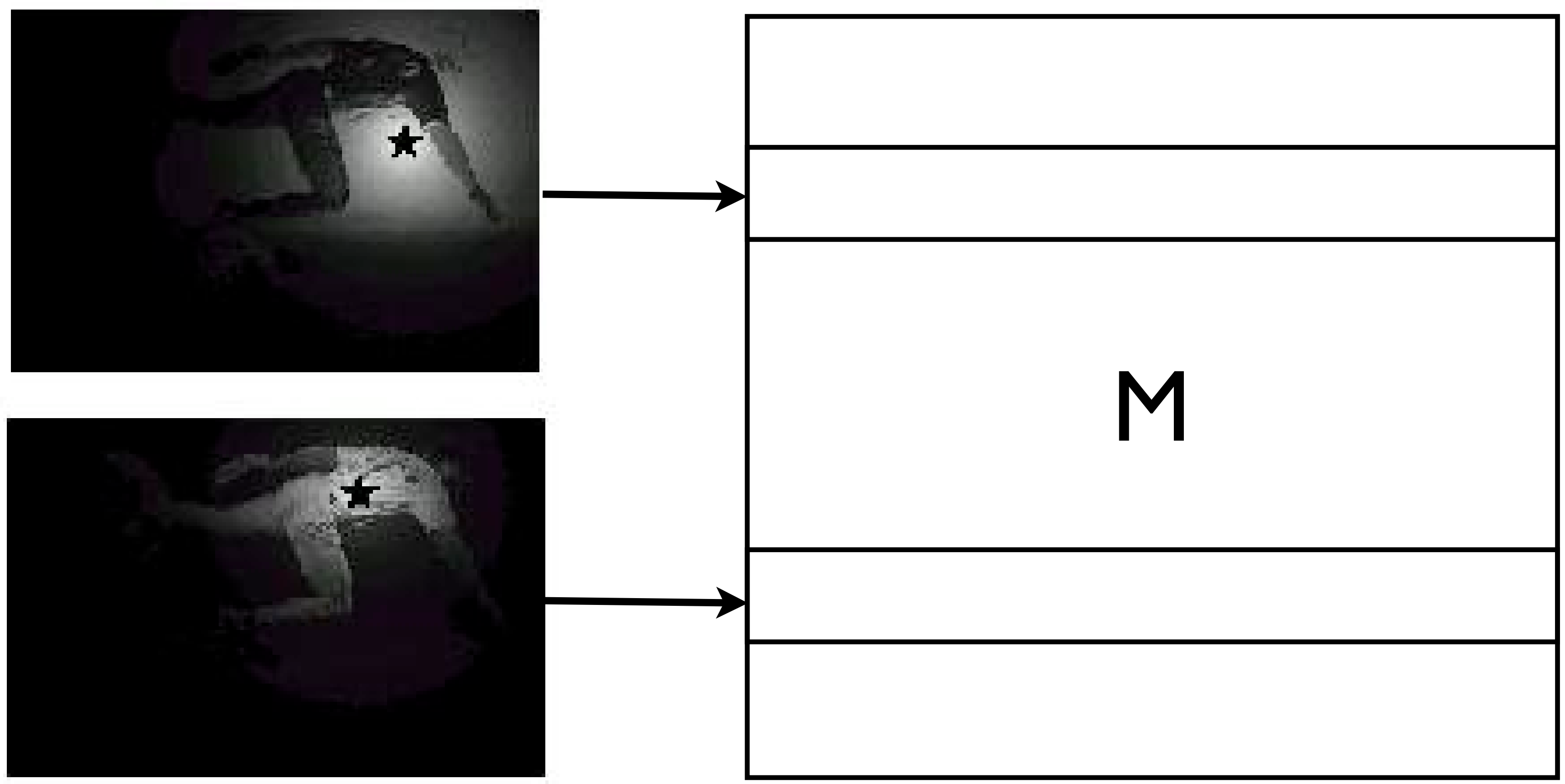
- Need to engineer fast-decaying affinities
- Could be expensive, O(n^3). Sparse connectivity, sparse matrix functions.
- Continuous →Discrete can be tricky.

# Affinities: an example

$$w_{ij} = \exp\left(-\frac{(I_i - I_j)^2}{\sigma_I}\right) \exp\left(-\frac{||X_i - X_j||^2}{\sigma_X}\right) 1_{||X_i - X_j|| < r}$$



# Affinities: an example

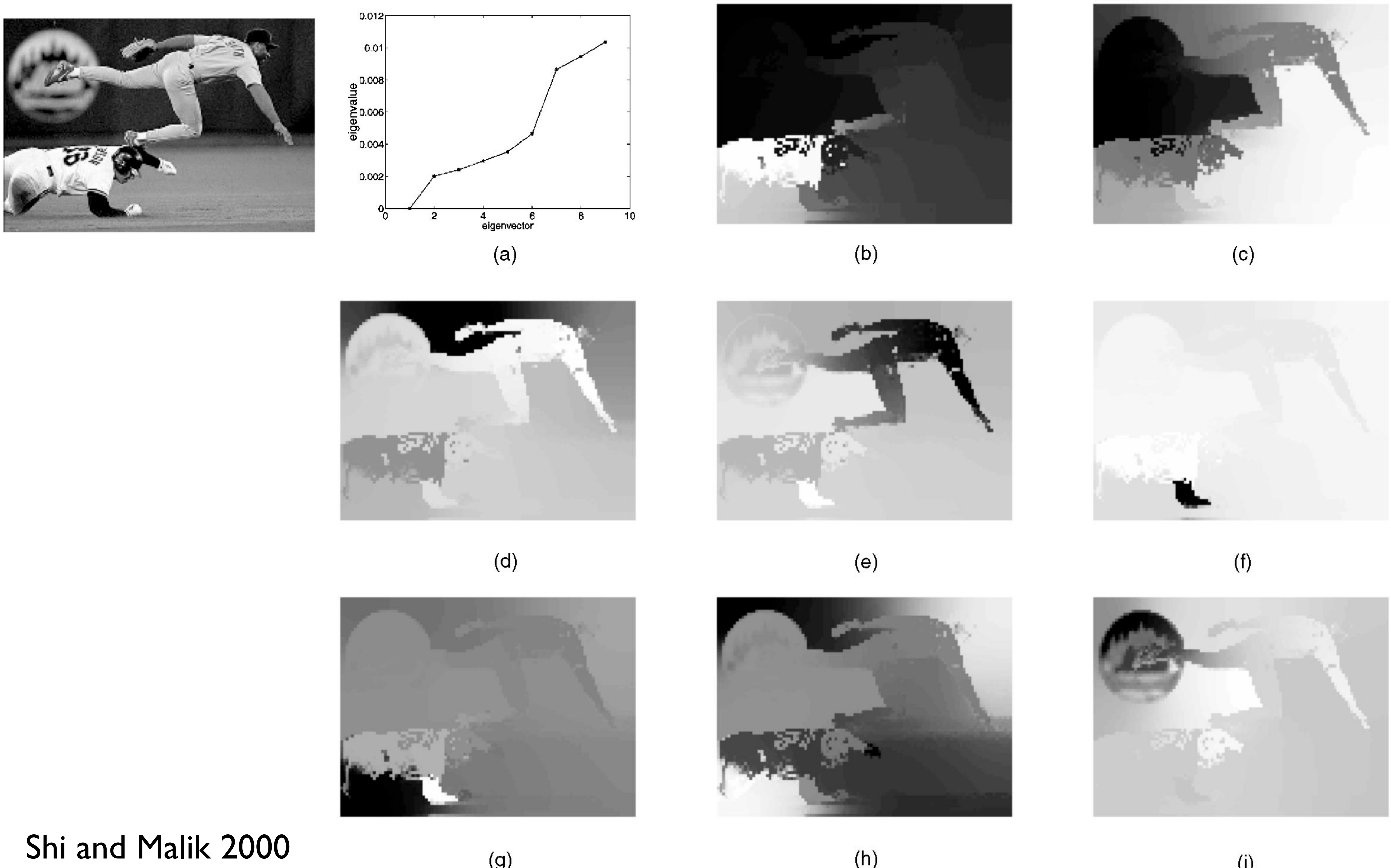


# The Spectrum

Group ⟷ Split

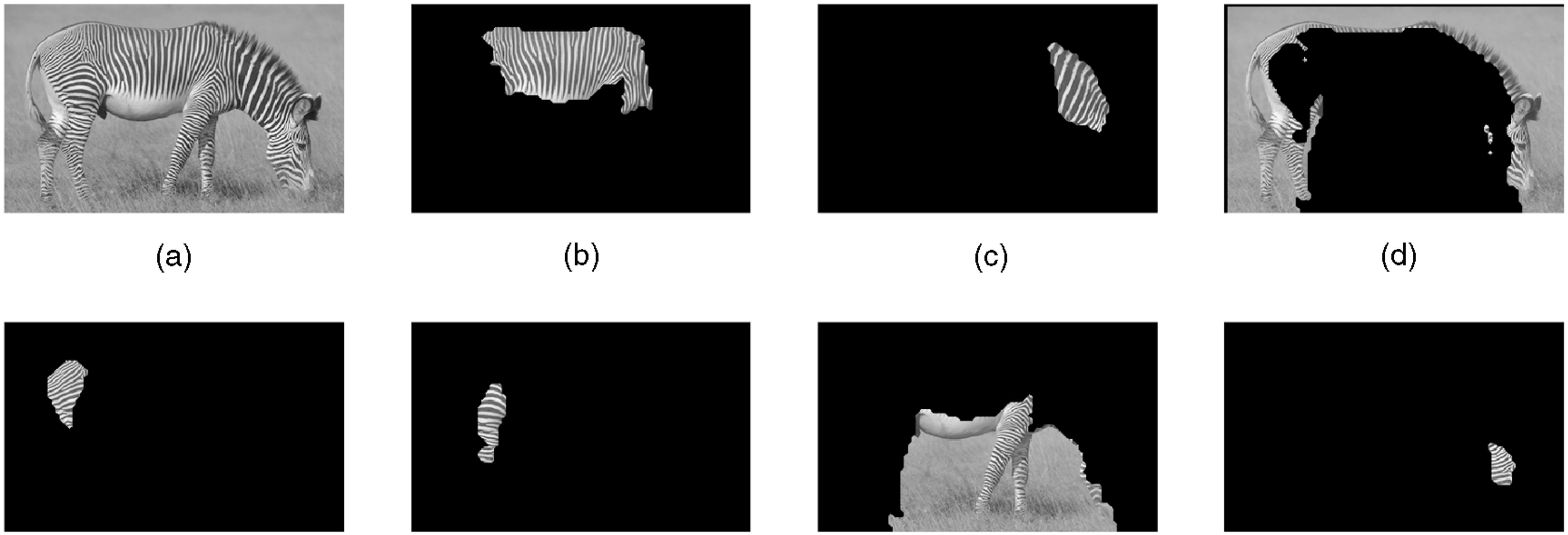
| Association | Normalized Cut | Cut |
|---|---|---|
| $\frac{assoc(A,A)}{\lvert A \rvert} + \frac{assoc(B,B)}{\lvert B \rvert}$ | $\frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$ <br> $2 - \left( \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \right)$ | $\frac{cut(A,B)}{\lvert A \rvert} + \frac{cut(A,B)}{\lvert B \rvert}$ |
| $\mathbf{Mx} = \lambda \ \mathbf{x}$ | $(\mathbf{D} - \mathbf{M})\mathbf{x} = \lambda \mathbf{D} \mathbf{x}$ | $(\mathbf{D} - \mathbf{M})\mathbf{x} = \lambda \mathbf{x}$ |

# Example



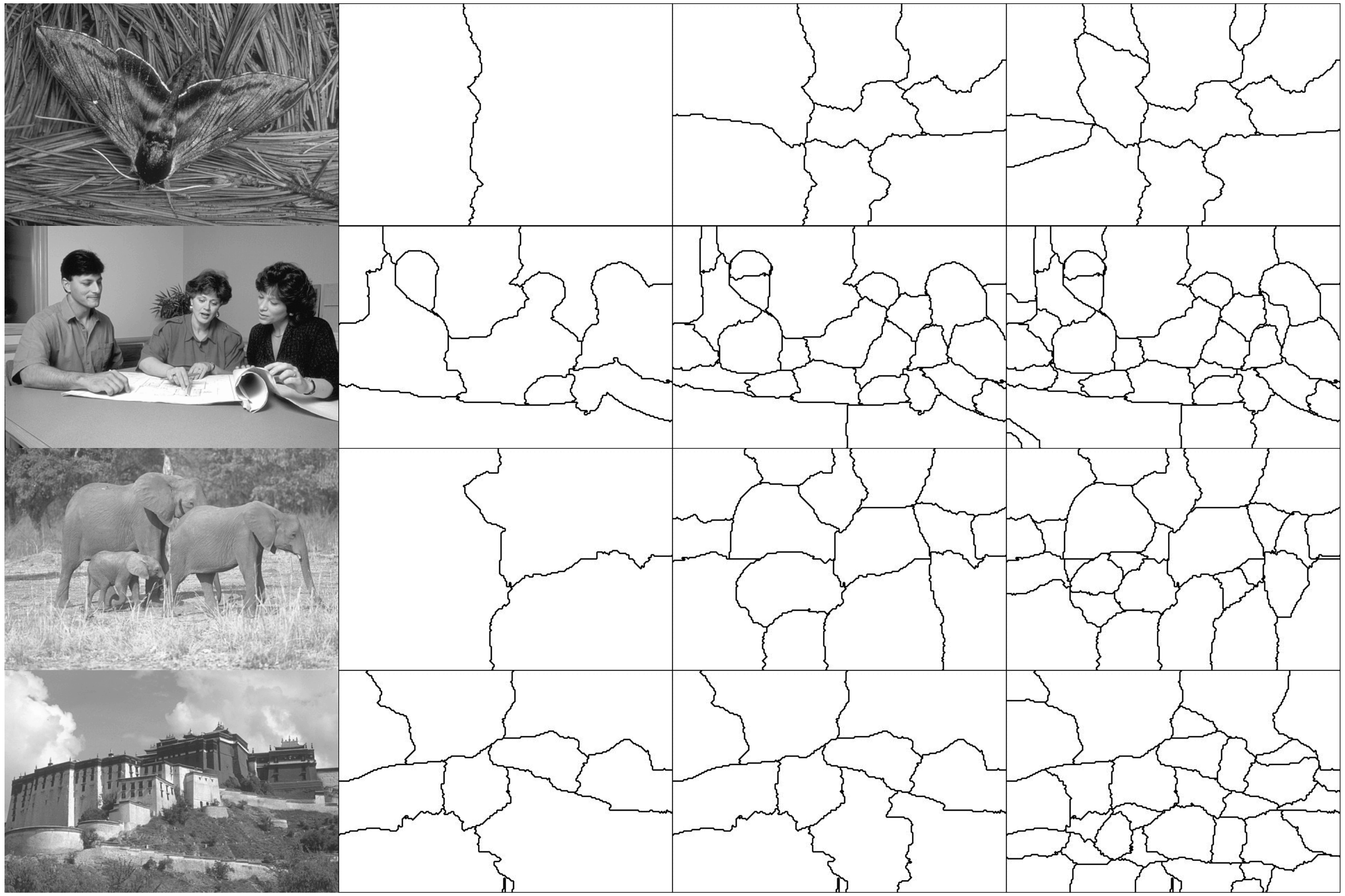

Shi and Malik 2000

# Example

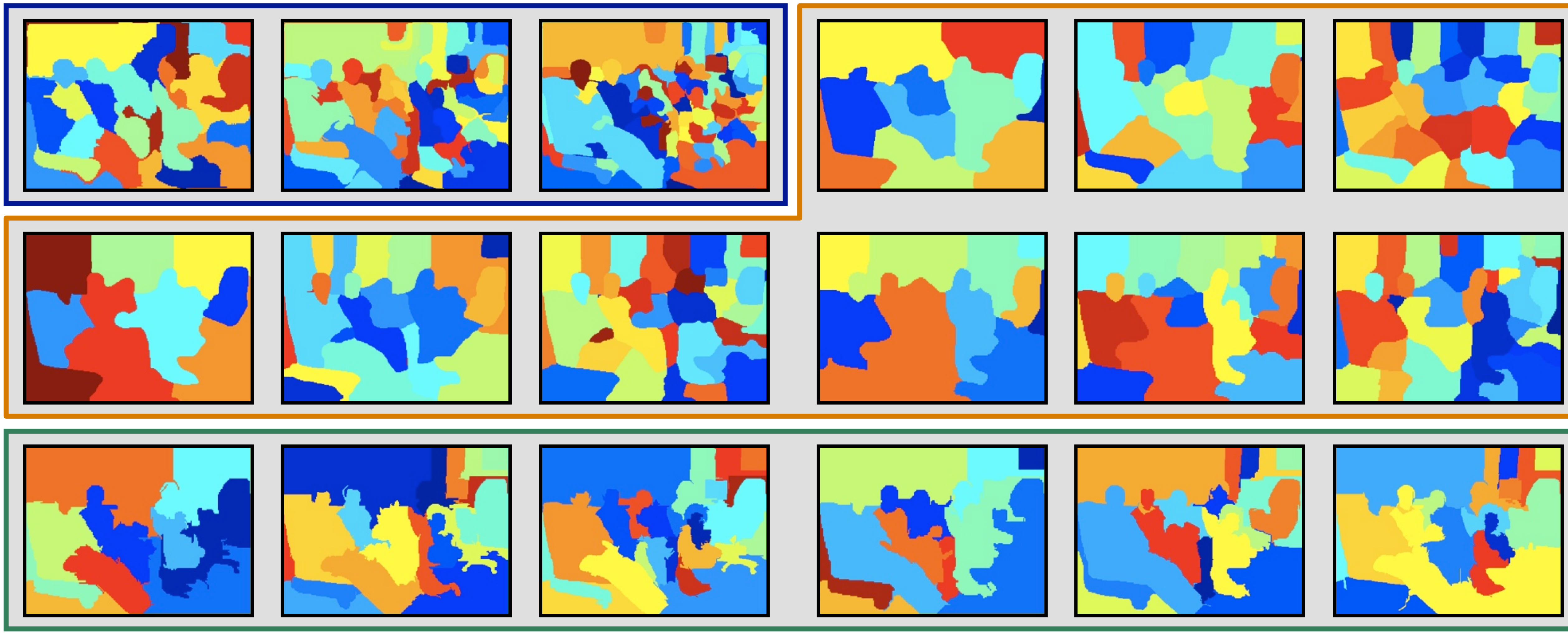



Shi and Malik 2000

# Examples



# Comparison: 3 Methods



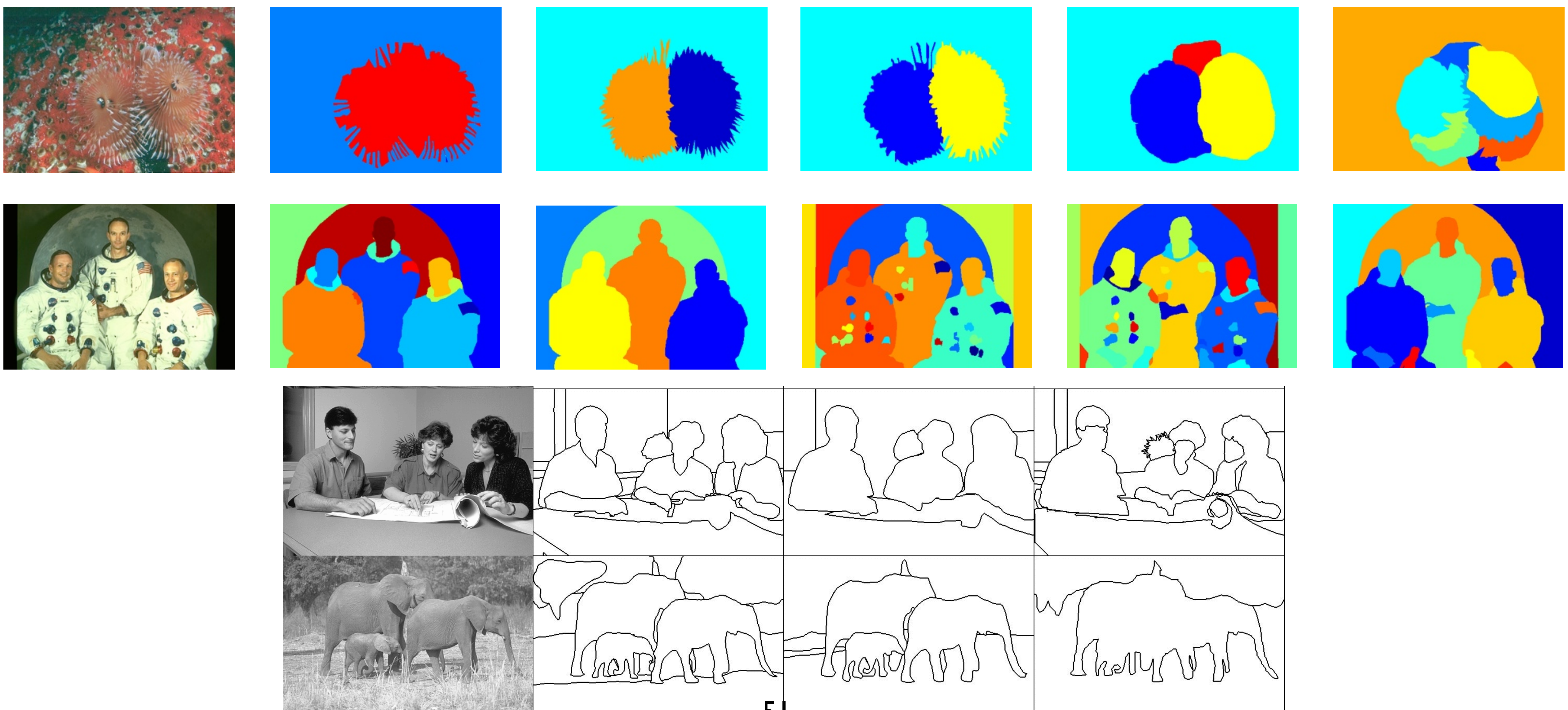
- Mean Shift [Comaniciu and Meer, PAMI'02]
- Normalized cuts with boundary estimates [Shi and Malik, PAMI'00; Fowlkes *et al.*, CVPR'03]
- Graph-based segmentation [Felzenszwalb and Huttenlocher, IJCV'04]

# Measures

- ✗ The results look good on 5 images.
- Region overlap-based
  - Martin, Fowlkes, Tal, Malik, ICCV'01
- Precision-recall on boundaries
  - Martin, Fowlkes, Malik, PAMI'04
- Variation of Information
  - Meila, ICML'05
- Normalized Probabilistic Rand (NPR) index
  - Unnikrishnan, Pantofaru, Hebert, PAMI'07

# Berkeley Segmentation Dataset

- 300 images
- 5-7 ground truth (human) segmentations per image
- [Martin *et al.* ICCV'01]



# Main References


- D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis". IEEE Trans. PAMI, Vol. 24, No. 5, 2002.
- J. Shi and J. Malik, "Normalized cuts and image segmentation". IEEE Trans. on PAMI, 8(22), 2000.

- The slides in this presentation are based on Martial Hebert's slides for Computer Vision 16-720 at CMU.