CSCI 435/MCS9435
Computer Vision

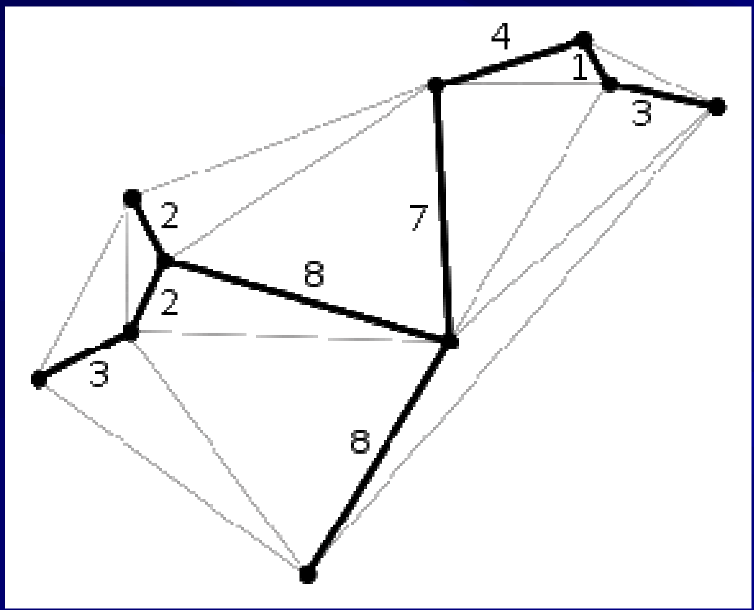
Image Segmentation (II)

Lecturer: Wanqing Li, PhD
Room: 3.101
email: wanqing@uow.edu.au
http://www.uow.edu.au/~wanqing

Outline

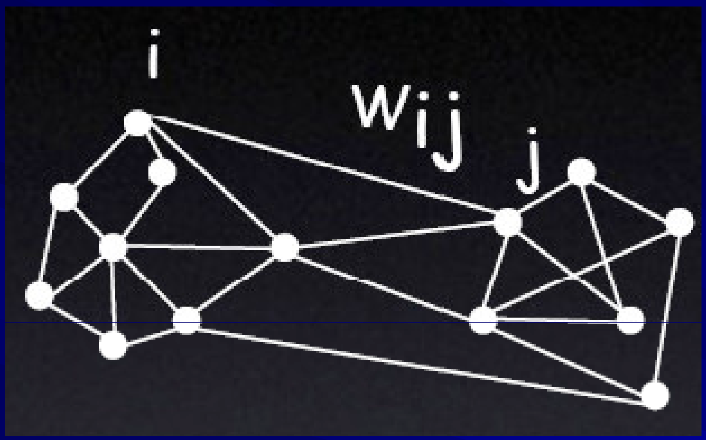
- Graph-based Image Segmentation
  - Shortest (minimum) spanning tree (SST)
  - Recursive RSST
  - Ncut
- Object Segmentation
  - Perceptual grouping
- Evaluation of segmentation

Basic Concepts of Graphs



- Weighed Graph G
  - Vertices, edges & weights
- A spanning tree (ST) G
  - a subgraph which is a tree and connects all the vertices together.
- A minimum ST (MST) or shortest ST (SST) of G
  - ST with the minimum sum of the weights
  - Prim's algorithm and Kruskal's algorithm

Graph-based Image Segmentation



G = {V,E}

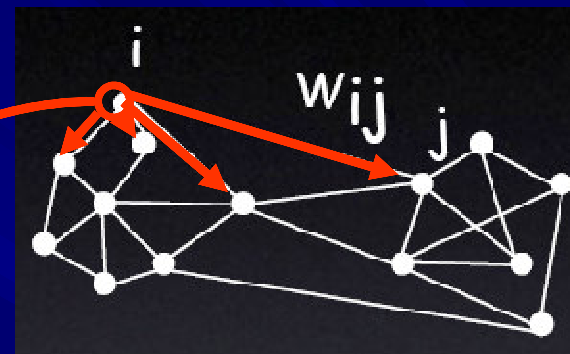
V: graph nodes
E: edges connection nodes

Pixels
Pixel similarity

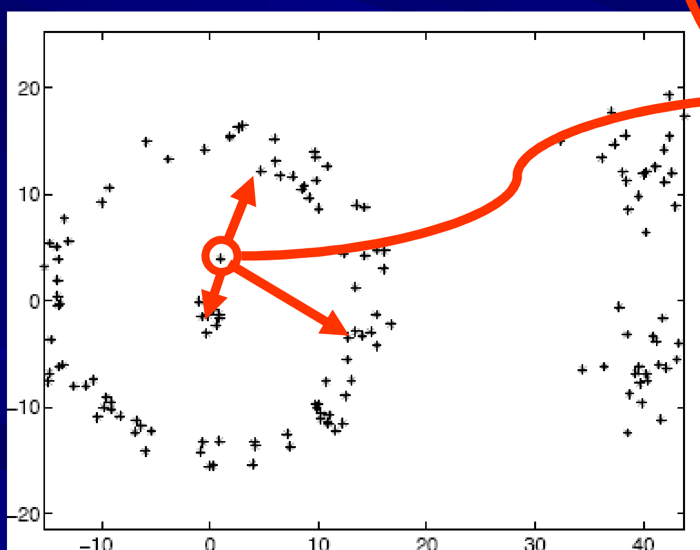
Slides from Jianbo Shi

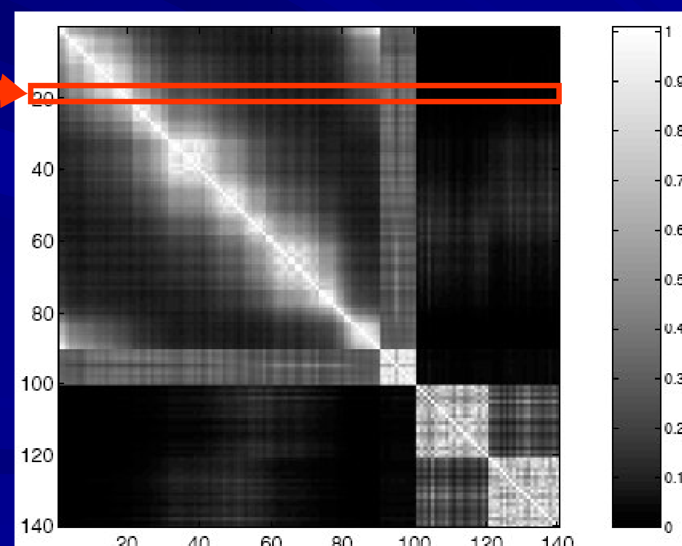
Graph terminology

- Similarity matrix: $W = \left[w_{i,j} \right]$



$$w_{i,j} = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$





Slides from Jianbo Shi

Pixel similarity functions

Intensity

$$W(i,j) = e^{\frac{-\left\| I_{(i)} - I_{(j)} \right\|_2^2}{\sigma_I^2}}$$

Distance

$$W(i,j) = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$

Texture

$$W(i,j) = e^{\frac{-\left\| c_{(i)} - c_{(j)} \right\|_2^2}{\sigma_c^2}}$$

Pixel similarity functions

Intensity

$$\frac{-\left\|I_{(i)}-I_{(j)}\right\|_2^2}{\sigma_I^2}$$

here c(x) is a vector of filter outputs.
A natural thing to do is to square the outputs
of a range of different filters
at different scales and orientations,
smooth the result, and rack
these into a vector.

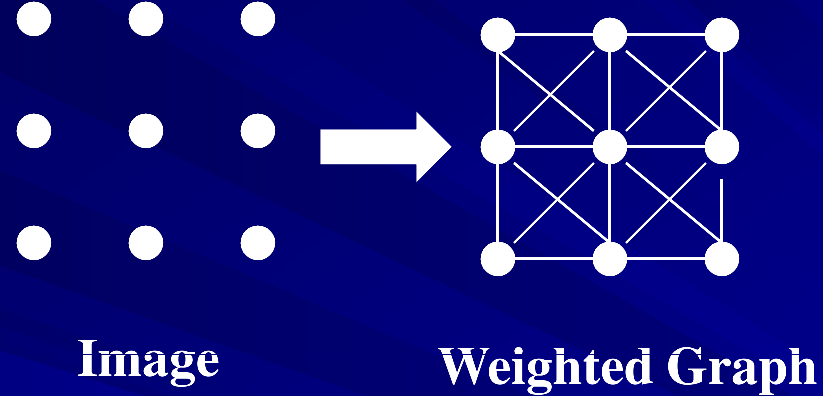
$$e^{\frac{-\left\|X_{(i)}-X_{(j)}\right\|_2^2}{\sigma_X^2}}$$

Texture

$$W(i,j)=e^{\frac{-\left\|c_{(i)}-c_{(j)}\right\|_2^2}{\sigma_c^2}}$$

Shortest Spanning Tree (SST) and Recursive SST based method

Segmentation using SST



Pixel x $f_x = (x_1, x_2, \cdots, x_p)$

Pixel y $f_y = (y_1, y_2, \cdots, y_p)$

similarity between x and y $w_{xy} = d(f_x, f_y)$

Segmentation using SST

- Definition of Shortest Spanning Tree (SST) or Minimum Spanning Tree (MST)
  - An SST of a weighted graph is a collection of edges connecting all the vertices such that the sum of the weights of the edges is at least as small as the sum of any other collection of edges connecting all the vertices

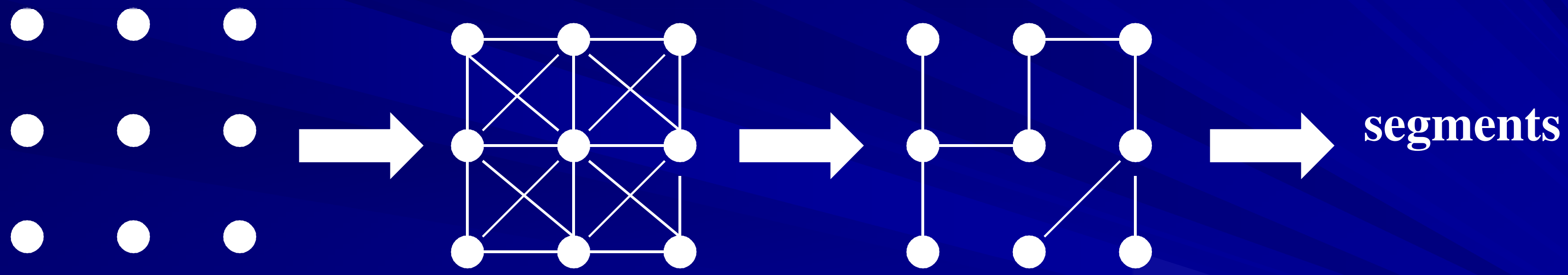


Image **Weighted Graph** **SST**

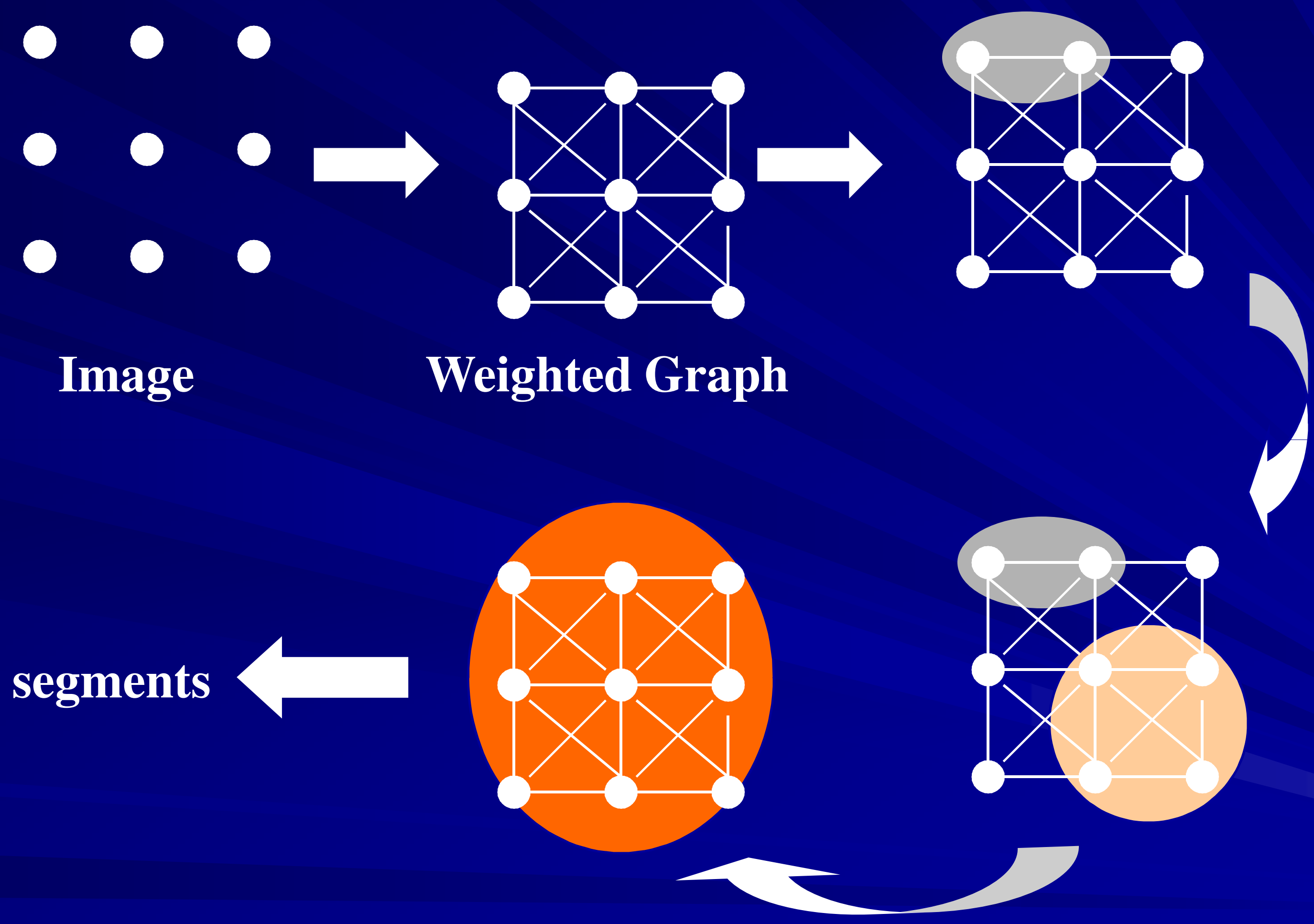
Finding SST **: Kruskal's algorithm**

Segments from SST**: Cut the most weighted R-1 edges.**

SST Segmentation (continue)

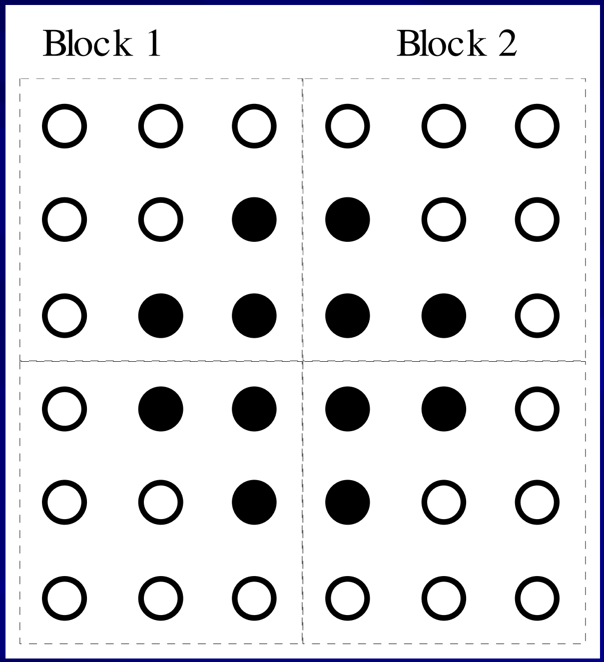
- Property of SST based segmentation
  - Spatial information is used
  - Region boundaries are defined very accurately
  - Regions produced are closed
  - SST has all information needed for segmenting the image into any number of regions (1 to number of pixels)
  - When a new region is added, only that specific region which is split in two is changed. This means that established boundaries do not move as move details is included in the segmentation when regions are added.
- Main problems to overcome
  - High computational cost
  - Noise sensitivity

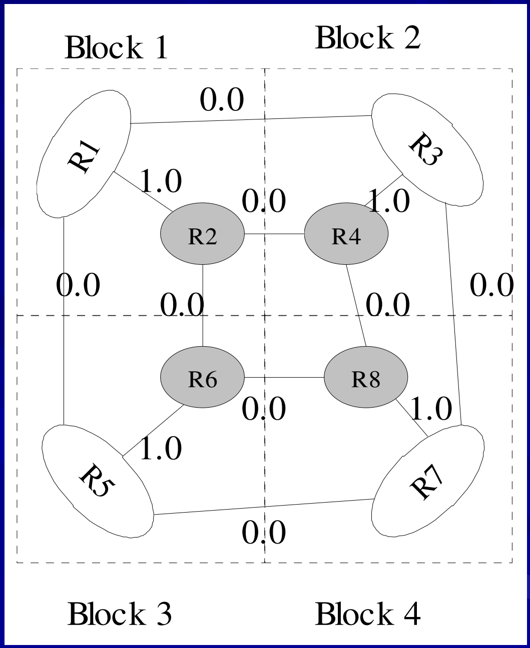
Recursive SST



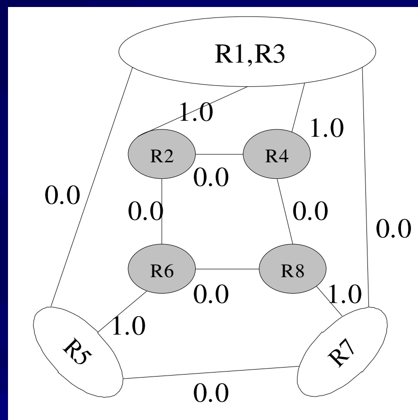
Blocked-based RSST (BRSST)

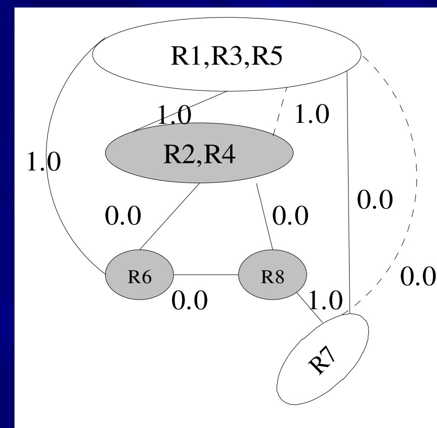
- BRSST = RSST + Divide-and-conquer

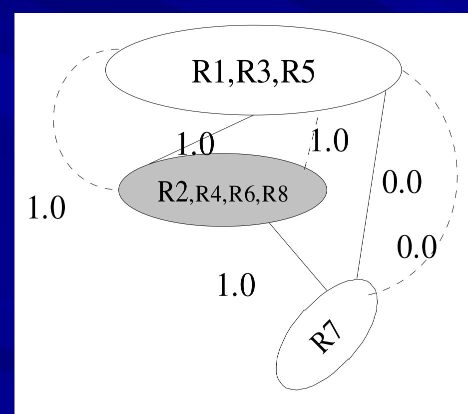


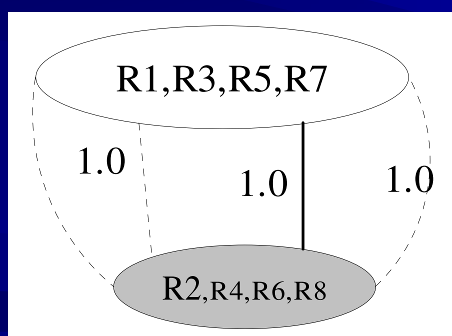


BRSST (continue)









BRSST (continue)

- Algorithm in a nutshell

  A divide and conquer implementation of RSST
  - Divide the image into blocks, e.g.8x8
  - Find the segments of each block using RSST
  - Map all segments into a combined graph
  - Apply the RSST to the combined graph and find the segments of the whole image

- Properties
  - Has all properties of SST segmentation
  - Hierarchical representation (one region to individual pixel as a region)

BRSST (continue)

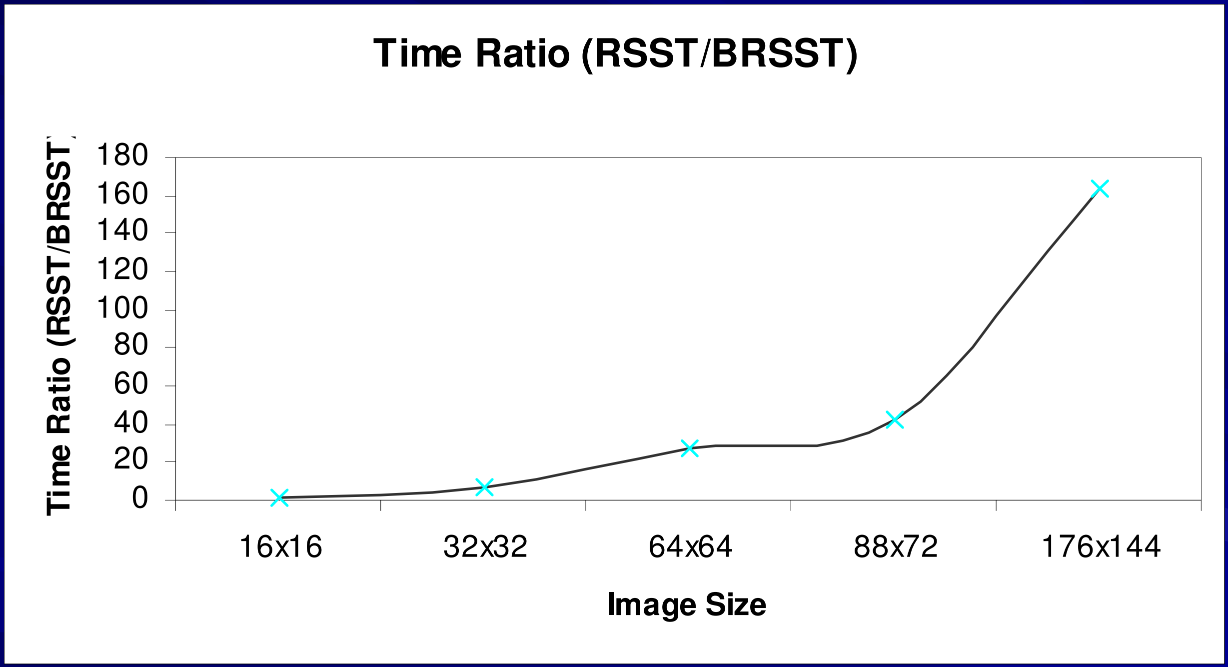
- Performance

RSST (9 regions)

BRSST (9 regions)

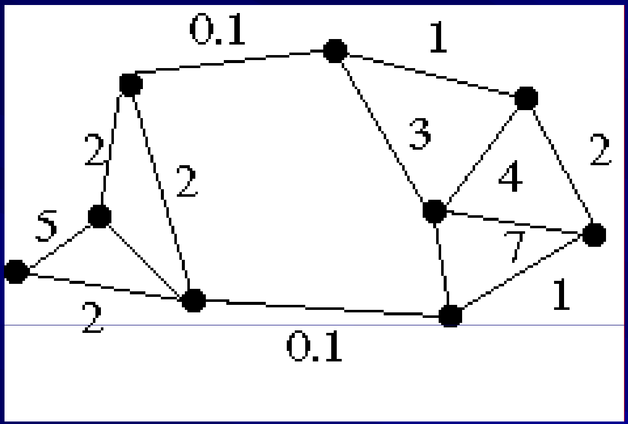
BRSST (continue)

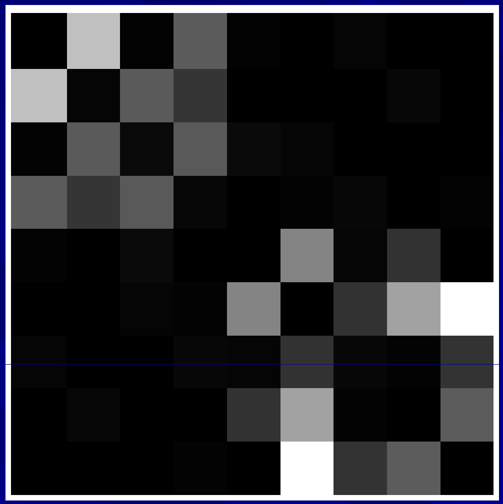
- Performance



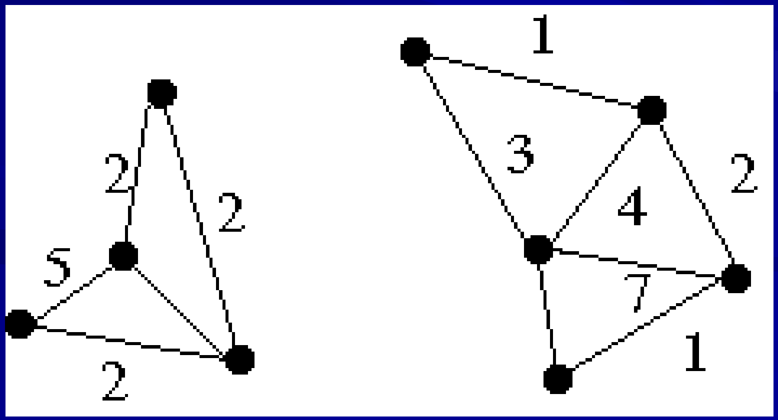
Minimum Cuts & Normalized Cuts based methods

Basic Idea

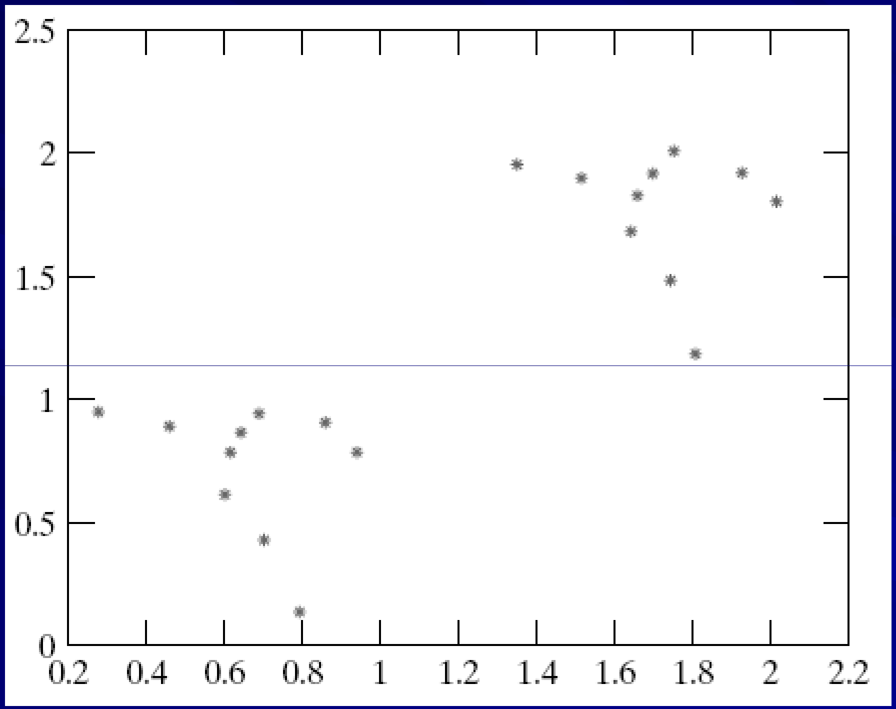


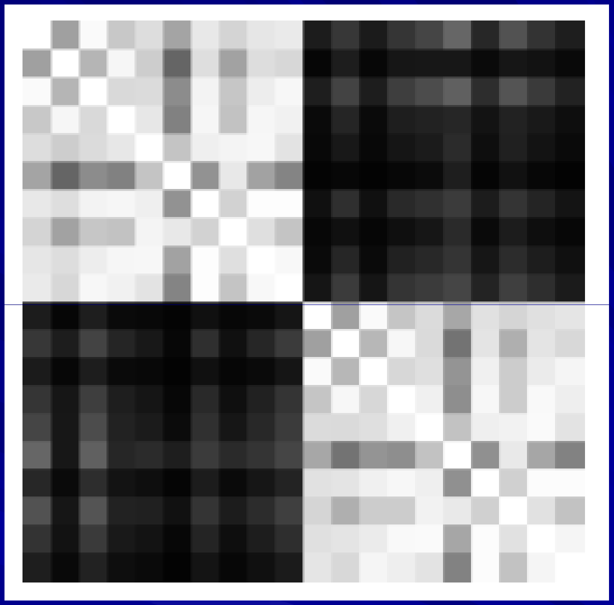


Affinity matrix

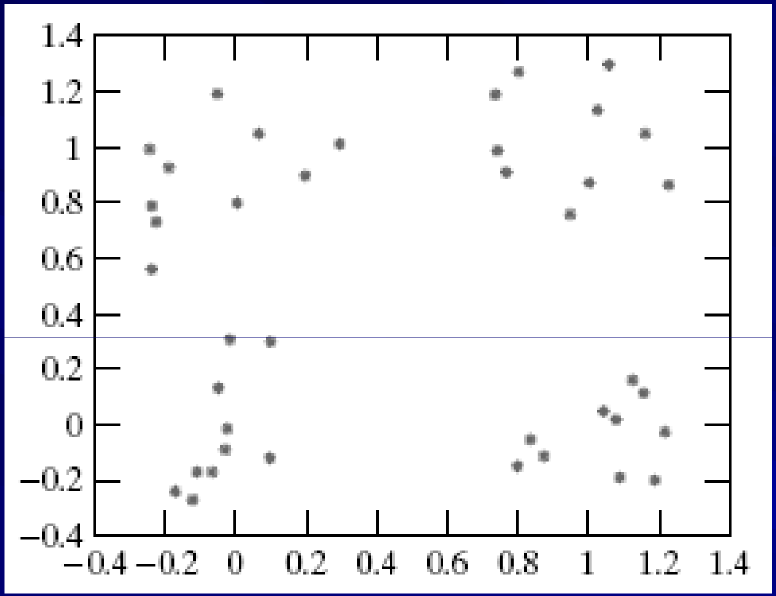


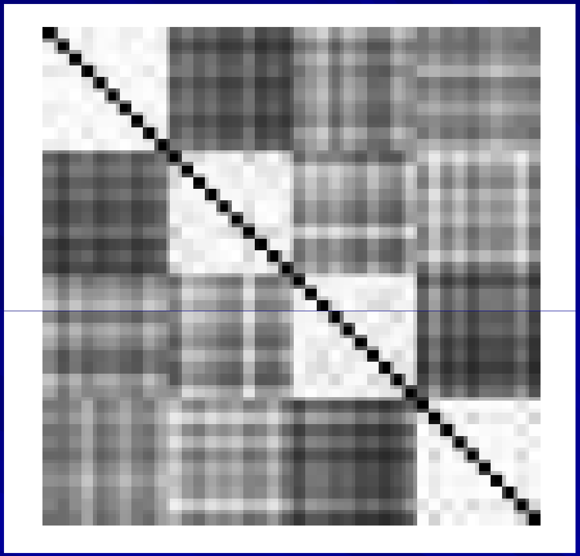
Basic Idea…



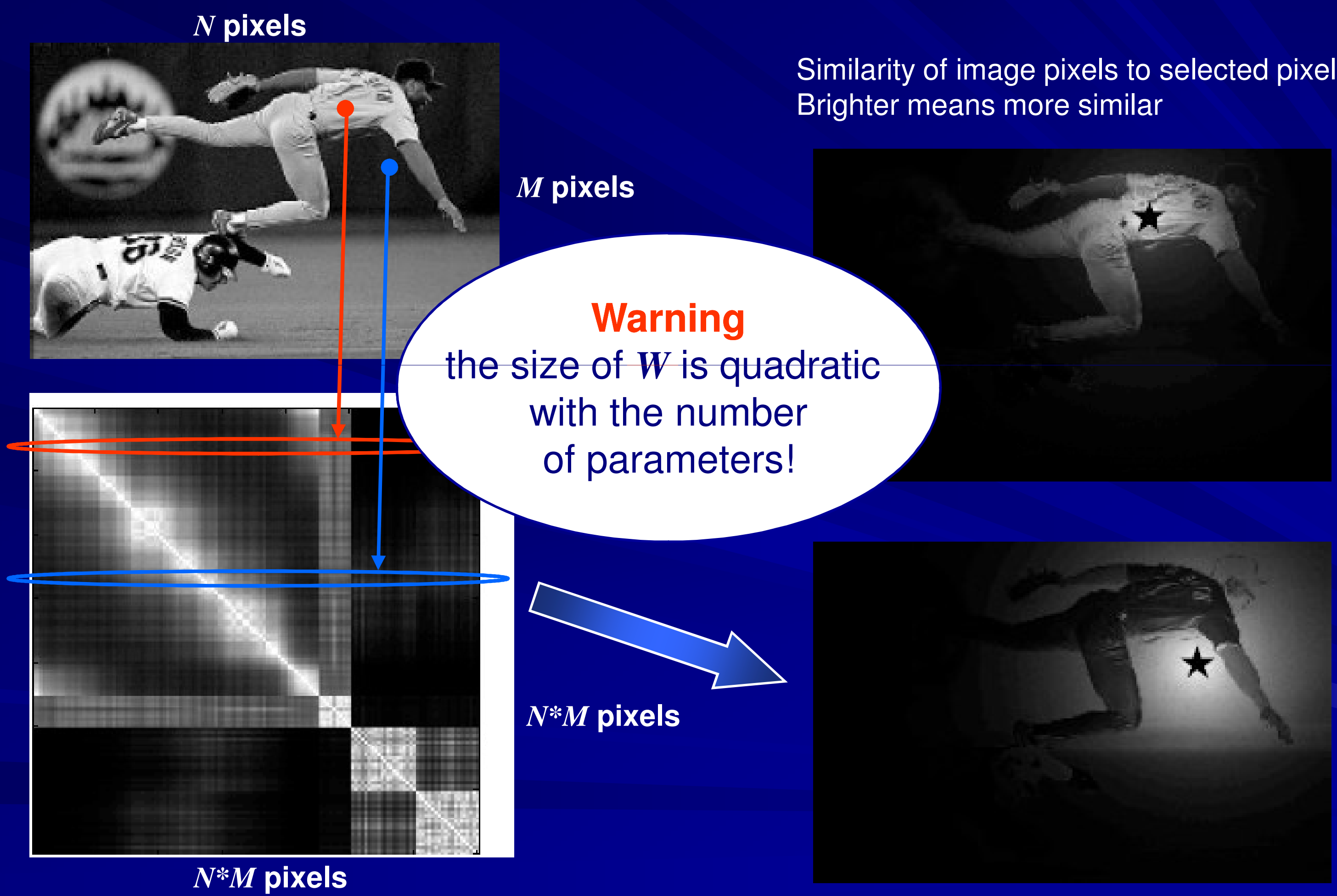


Basic idea…





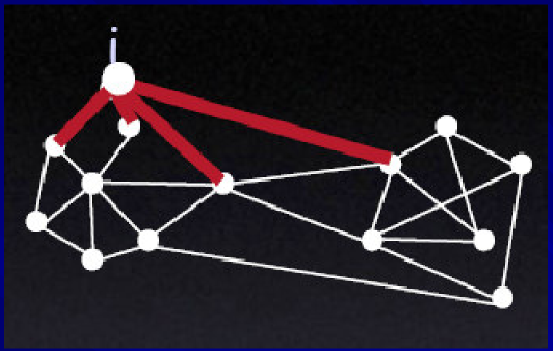
Affinity matrix

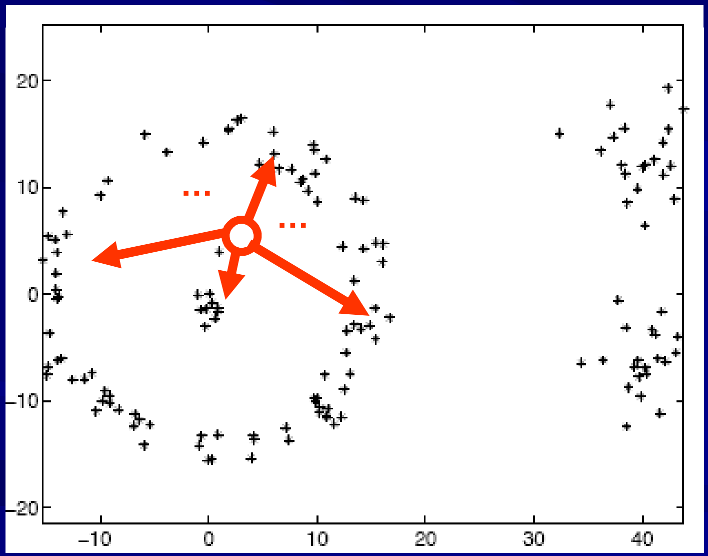


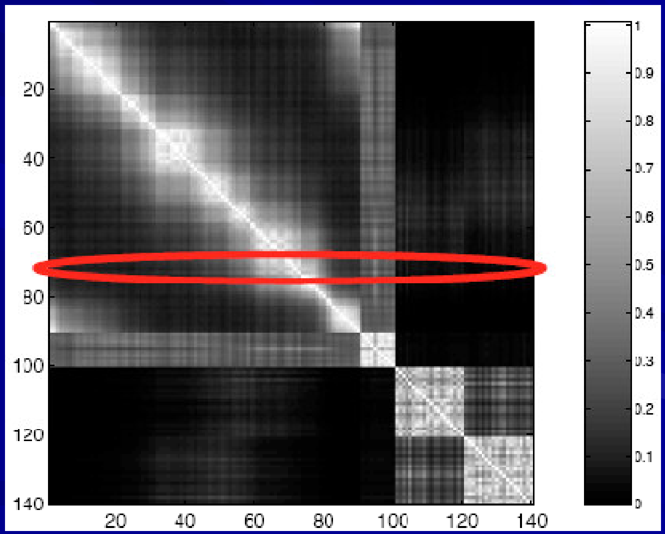
Graph terminology

- Degree of node:

$$d_i = \sum_j w_{i,j}$$





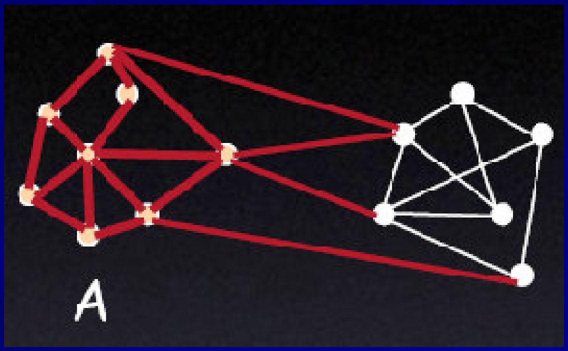


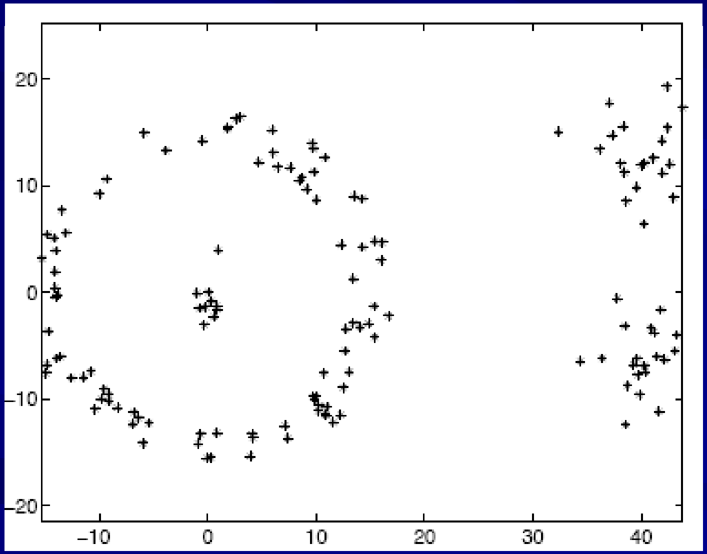
Slides from Jianbo Shi

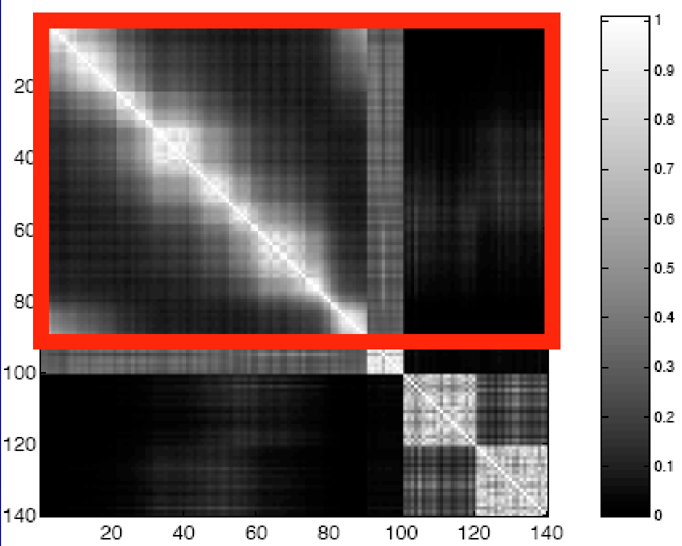
Graph terminology

- Volume of set:

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





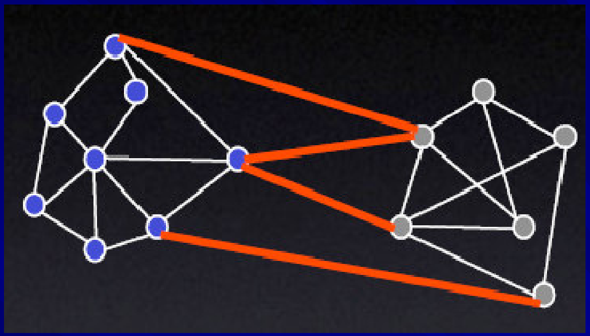


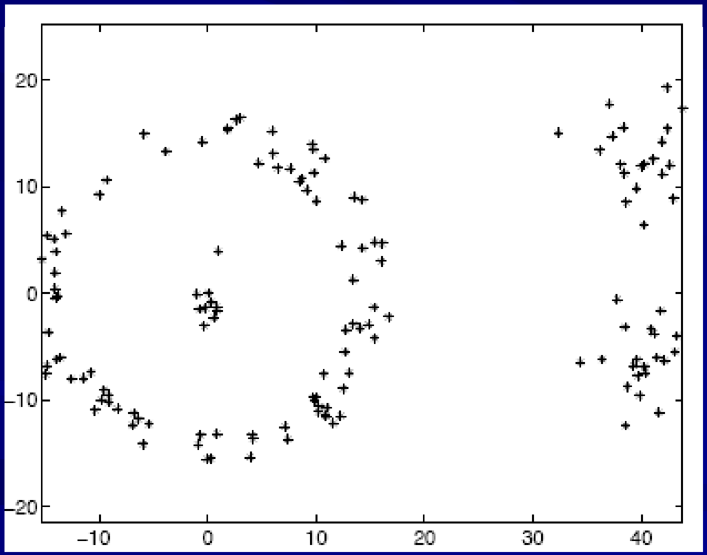
Slides from Jianbo Shi

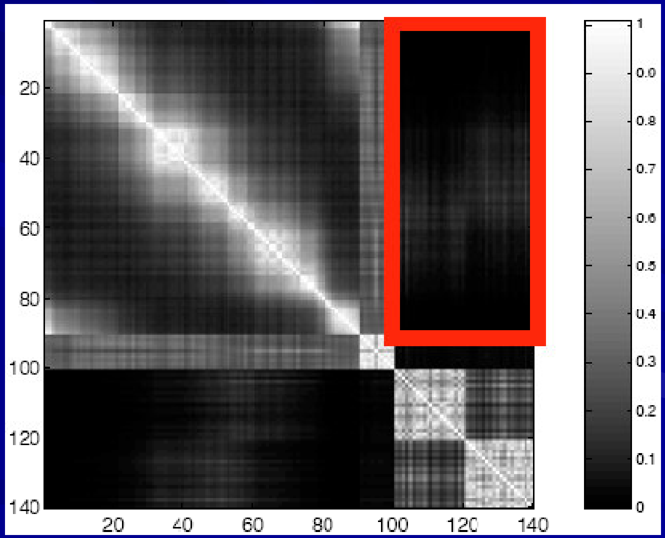
Graph terminology

- Cuts in a graph:

$$cut(A, \overline{A}) = \sum_{i \in A, j \in \overline{A}} w_{i,j}$$







Slides from Jianbo Shi

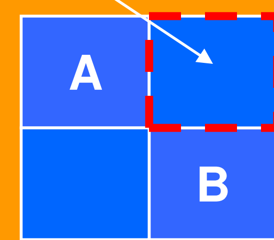
How do we extract a good cluster?

- Simplest idea: we want a vector $\boldsymbol{x}$ giving the association between each element and a cluster
- We want elements within this cluster to, on the whole, have strong affinity with one another
- We could maximize $x^T W x$
- But need the constraint $x^T x = 1$
- This is an eigenvalue problem - choose the eigenvector of $\boldsymbol{W}$ with largest eigenvalue.

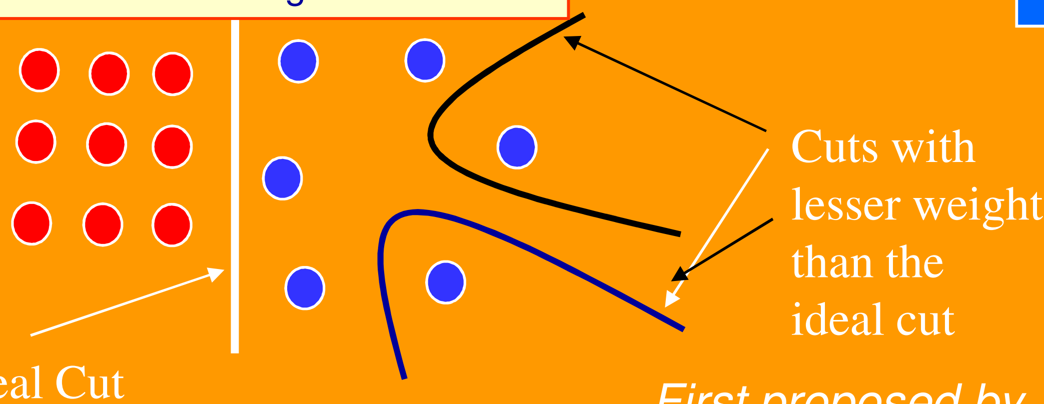
Minimum cut

- Criterion for partition:

$$\min cut(A,B) = \min_{A,B} \sum_{u \in A, v \in B} w(u,v)$$



Problem!
Weight of cut is directly proportional to the number of edges in the cut.

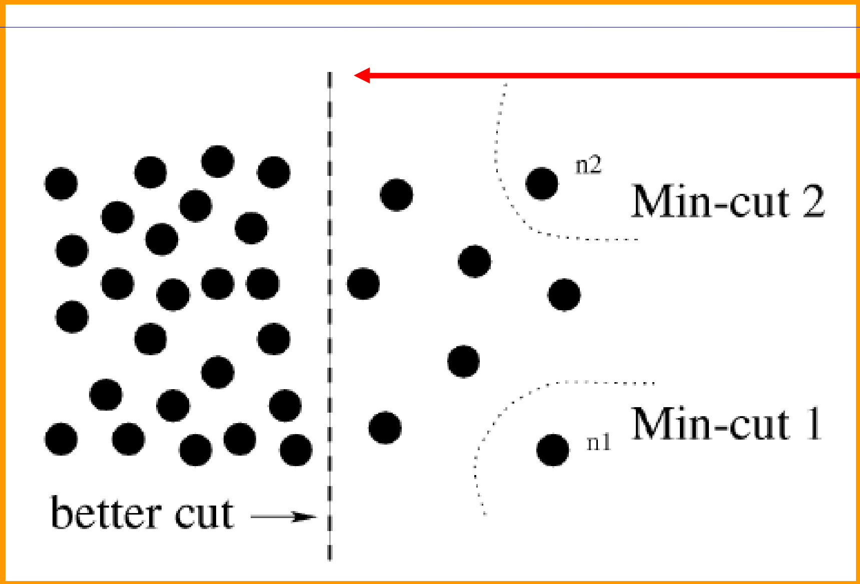


First proposed by Wu and Leahy

Normalized Cut

Normalized cut or balanced cut:

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

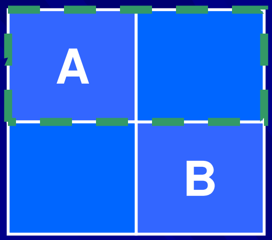


Finds better cut

Normalized Cut

- Volume of set (or association):

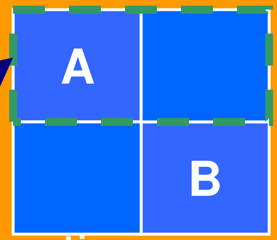
$$vol(A) = assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$



Normalized Cut

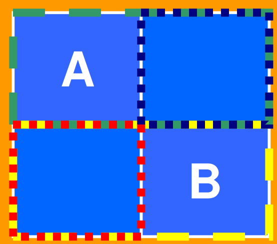
- Volume of set (or association):

$$vol(A) = assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$



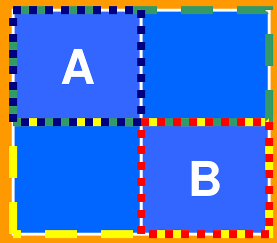
- Define normalized cut: “a fraction of the total edge connections to all the nodes in the graph”:

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



- Define normalized association: “how tightly on average nodes within the cluster are connected to each other”

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$



Observations(I)

- Maximizing *Nassoc* is the same as minimizing *Ncut*, since they are related:

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

- How to minimize *Ncut*?
  - Transform *Ncut* equation to a matricial form.
  - After simplifying:

$$D(i,i) = \sum_j W(i,j)$$

$$\min_x Ncut(x) = \min_y \frac{y^T (D - W) y}{y^T D y}$$

Rayleigh quotient

NP-Hard!

y's values are quantized

Algorithm

1. Define a similarity function between 2 nodes. i.e.:

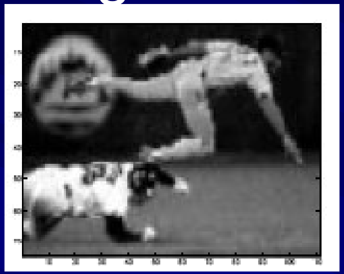
$$w_{i,j} = e^{\frac{-\left\| F_{(i)} - F_{(j)} \right\|_2^2}{\sigma_I^2} + \frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$

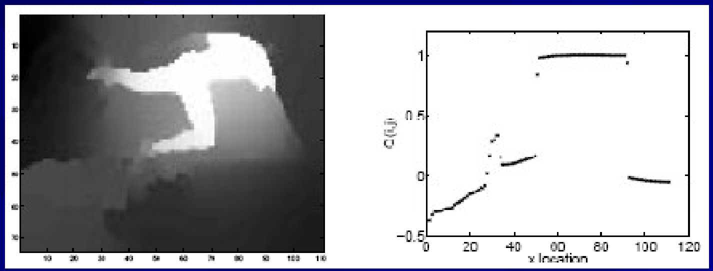
2. Compute affinity matrix (W) and degree matrix (D).
3. Solve $(D - W)y = \lambda D y$
4. Use the eigenvector with the second smallest eigenvalue to bipartition the graph.
5. Decide if re-partition current partitions.

Note: since precision requirements are low, W is very sparse and only few eigenvectors are required, the eigenvectors can be extracted very fast using Lanczos algorithm.

Discretization

- Sometimes there is not a clear threshold to binarize since eigenvectors take on continuous values.





- How to choose the splitting point?

a) Pick a constant value (0, or 0.5).

b) Pick the median value as splitting point.

c) Look for the splitting point that has the minimum *Ncut* value:

1. Choose *n* possible splitting points.
2. Compute *Ncut* value.
3. Pick minimum.

Experiments

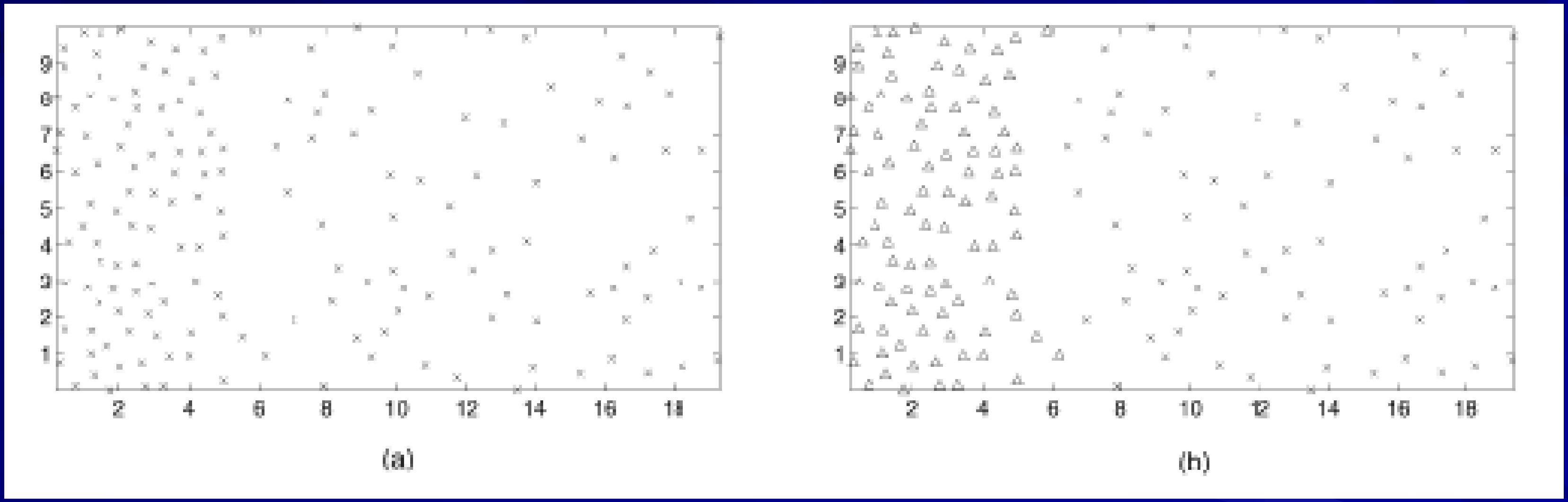
- Define similarity:

$$w_{i,j} = e^{\frac{-\left\|F_{(i)}-F_{(j)}\right\|_2^2}{\sigma_I^2}+\frac{-\left\|X_{(i)}-X_{(j)}\right\|_2^2}{\sigma_X^2}}$$

  - $F(i)=1$ for point sets.
  - $F(i)=I(i)$ for brightness images.
  - $F(i)=[v, v.s.\sin(h), v.s.\cos(h)]$ for HSV images.
  - $F(i)=[|I * f_1|, \cdots, |I * f_n|]$ in case of texture.

Experiments (I)

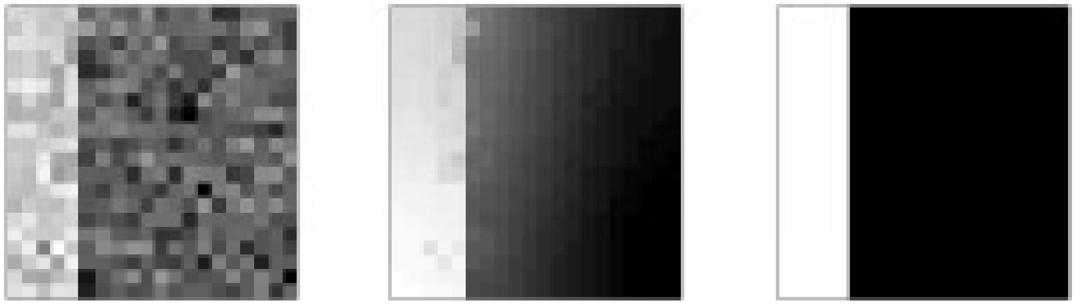
- Point set segmentation:



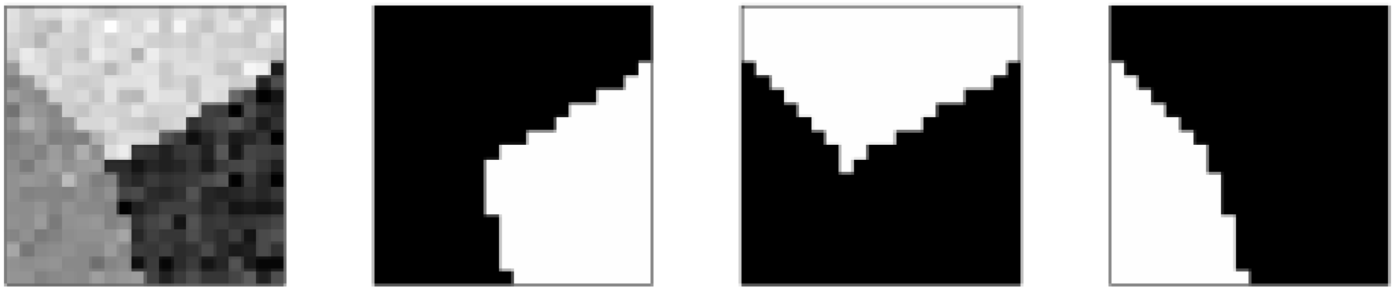
(a) Pointset generated by Poisson process. (b) Segmentation results.

Experiments (II)

- Synthetic images:



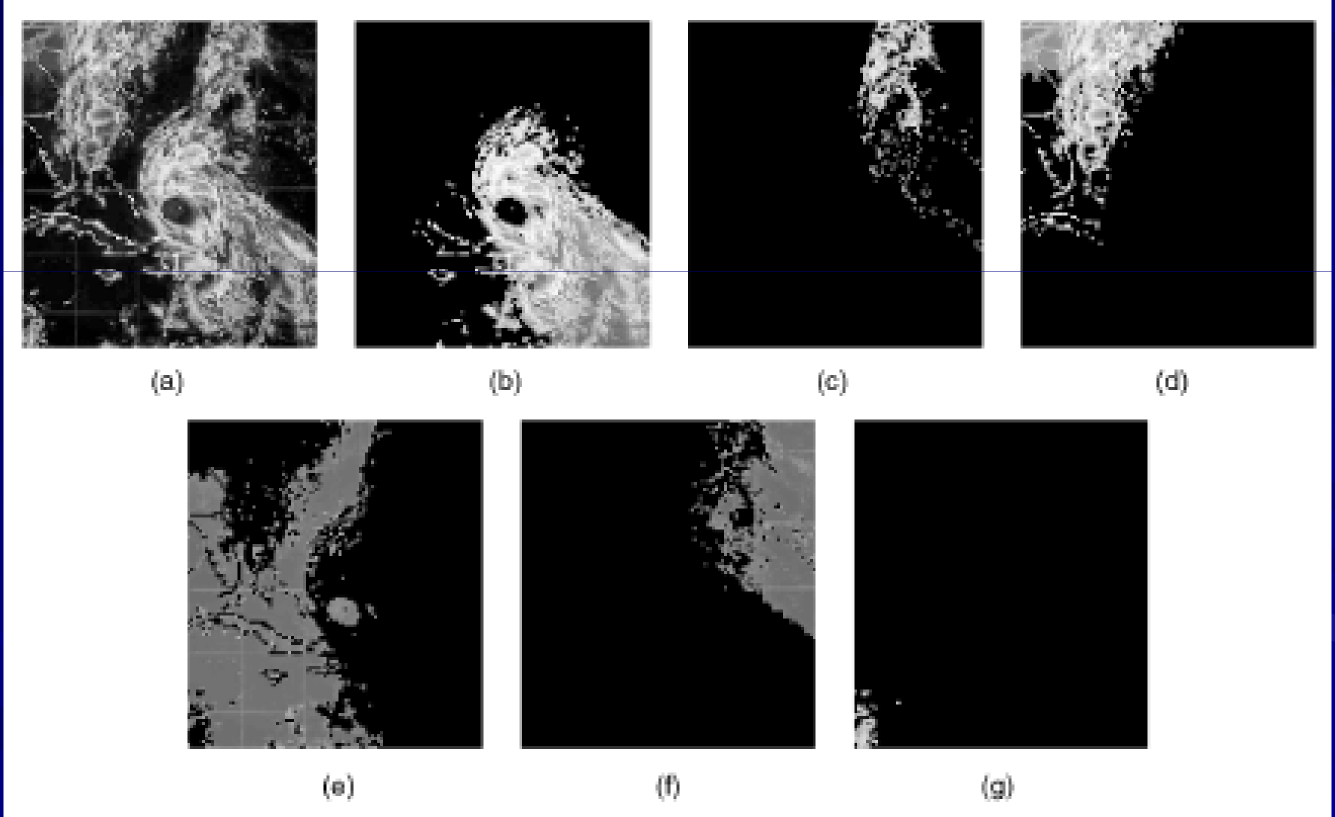
(a) (b) (c)



(a) (b) (c) (d)

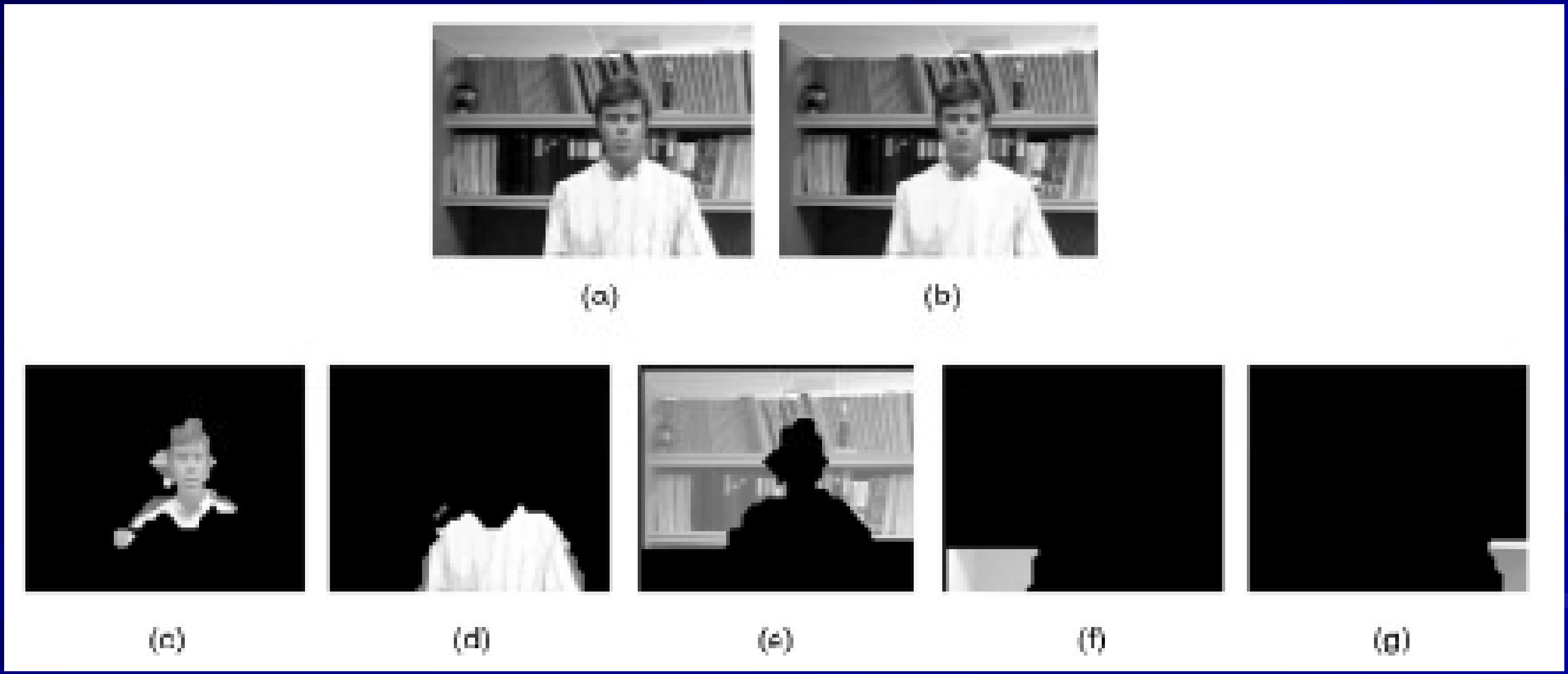
Experiments (III)

- Weather radar:



Experiments (IV)

- Motion segmentation



NCut Matlab tools

- http://www.cis.upenn.edu/~jshi/software/

Image Segmentation

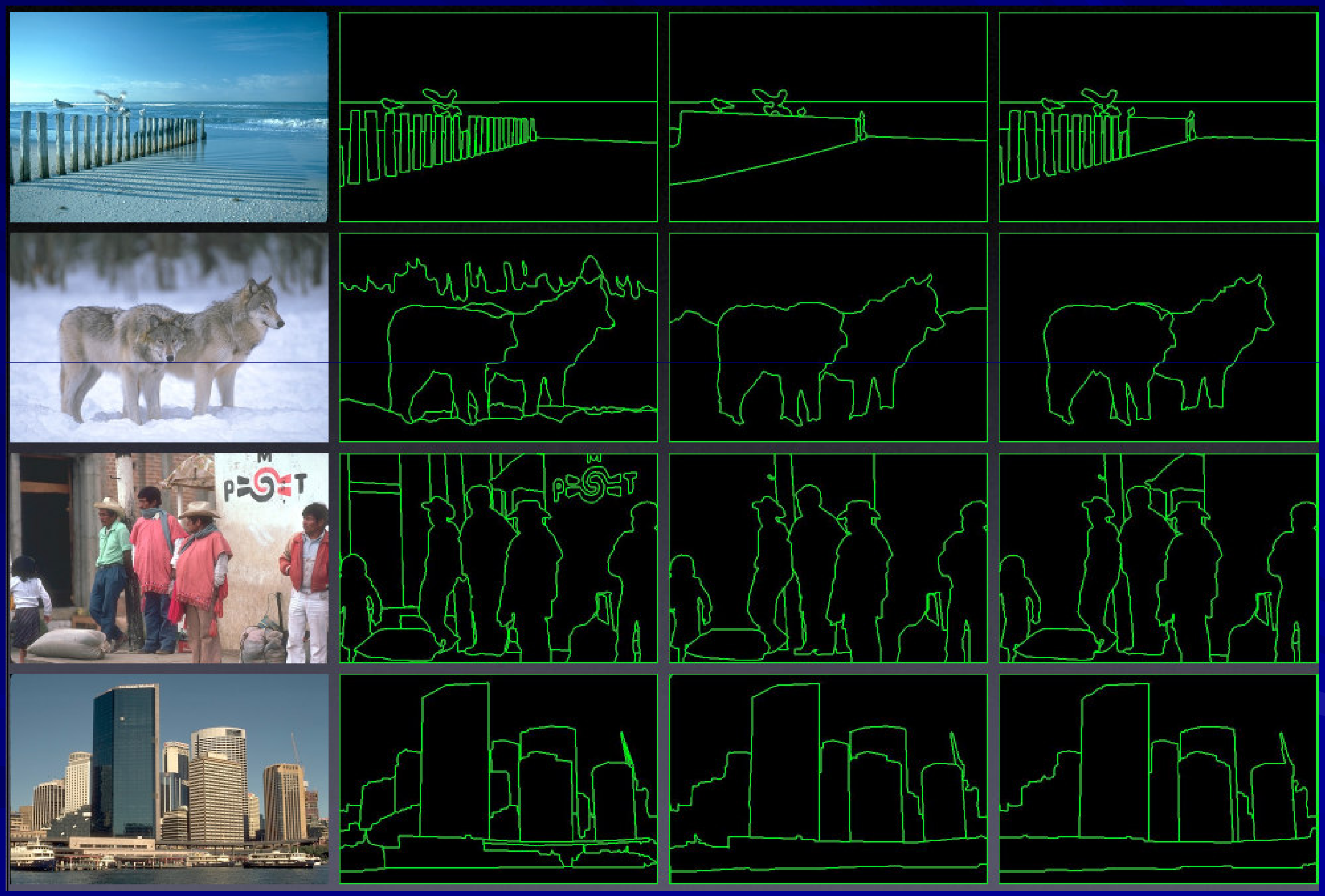
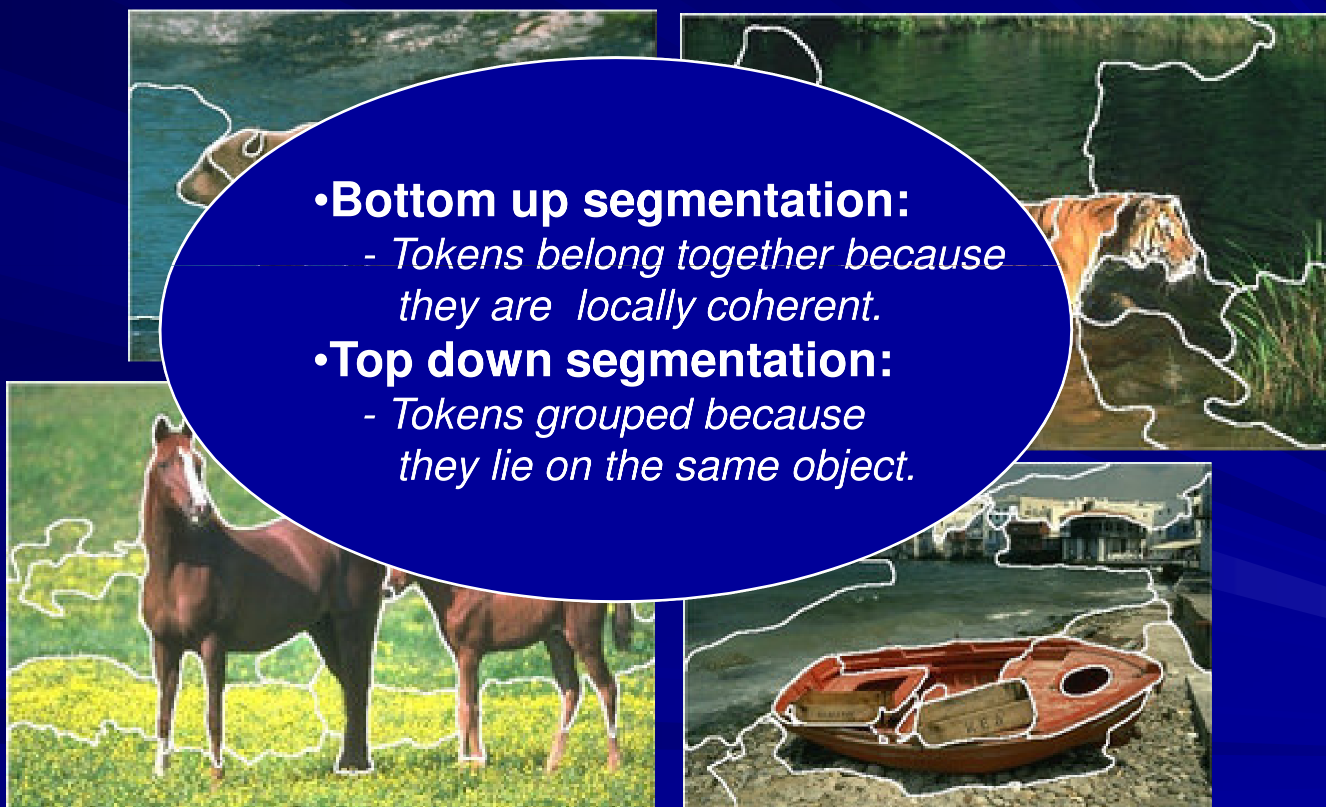


Image segmentation

- How do you pick the right segmentation?

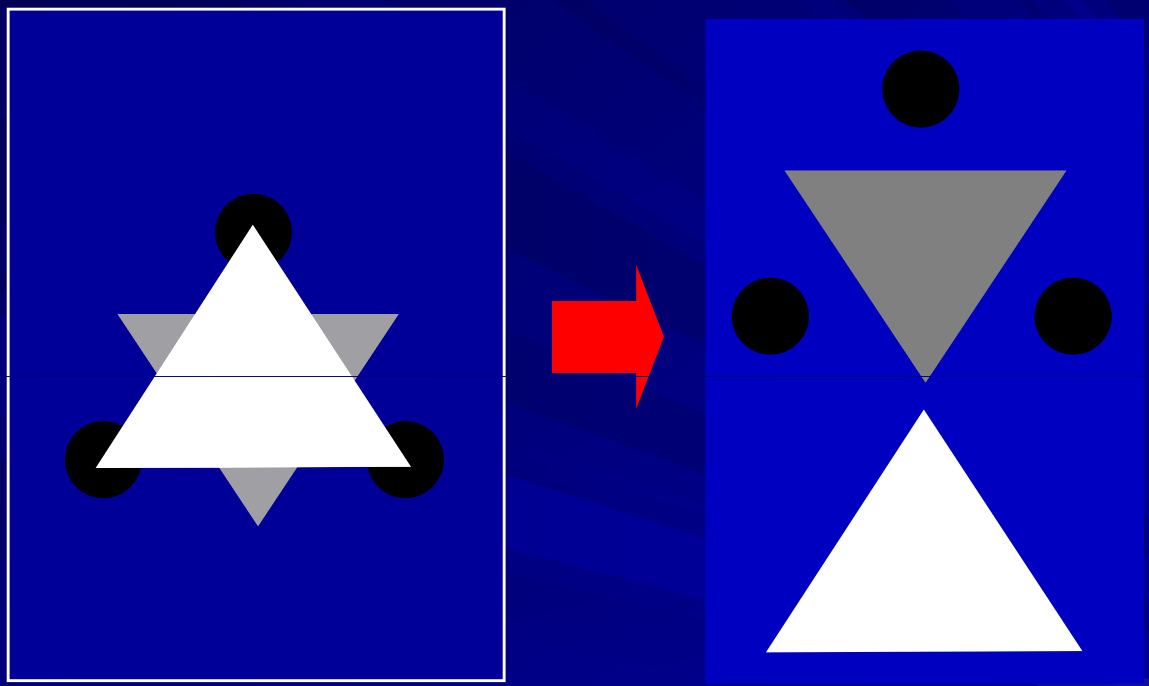


•**Bottom up segmentation:**
  - *Tokens belong together because they are locally coherent.*

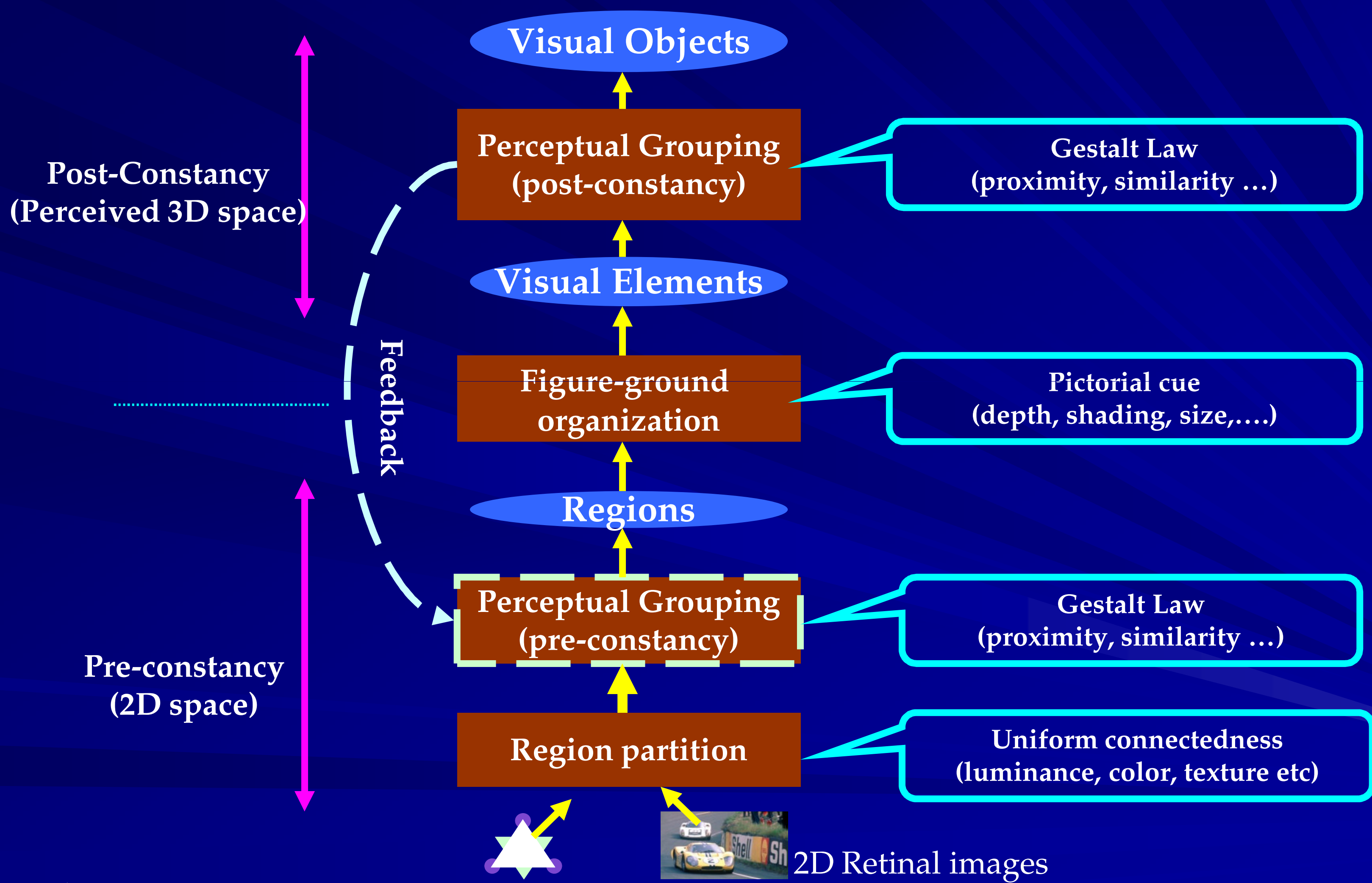
•**Top down segmentation:**
  - *Tokens grouped because they lie on the same object.*

Perceptual Grouping

Typical example (Kanizsa Triangle)

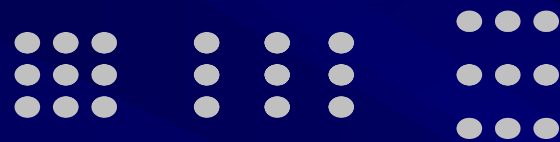


Visual Perception Process

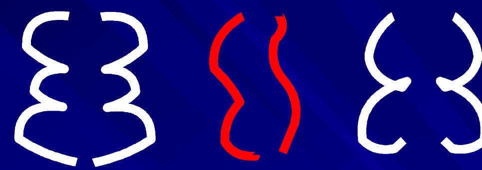


Palmer'2002

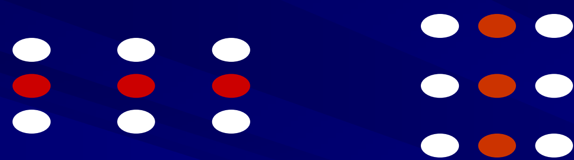
Gestalt Law (Principles of grouping)



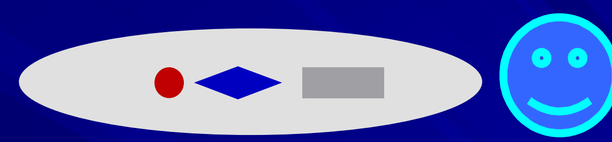
Proximity



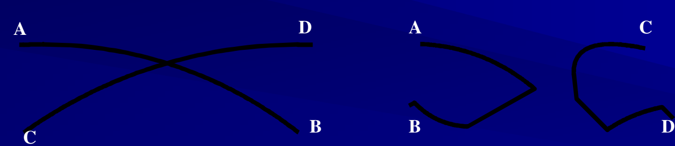
Orientation & Symmetry



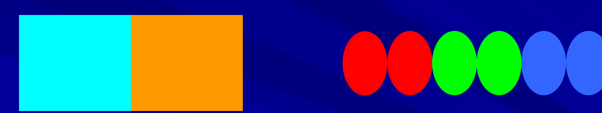
Similarity



Common region

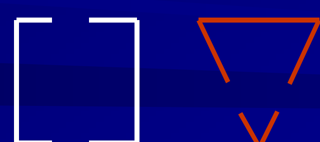


Good continuation



Element connectedness

- Common fate
- Synchrony



Closure

Gestalt Law & Image Segmentation

- In general, image/video processing and analysis are very much "*heuristic*". Gestalt law have been widely applied implicitly/explicitly.

- A key challenge is how to build a computation framework of Gestalt grouping, which could lead to a segmentation that is consistent with human segmentation

Modeling of Visual Perception for Segmentation

- Key components in the computational framework
  - Representation
    - Visual elements & their relationship
    - Hierarchy is essential
  - Quantification of the Gestalt Law
  - Grouping Inference

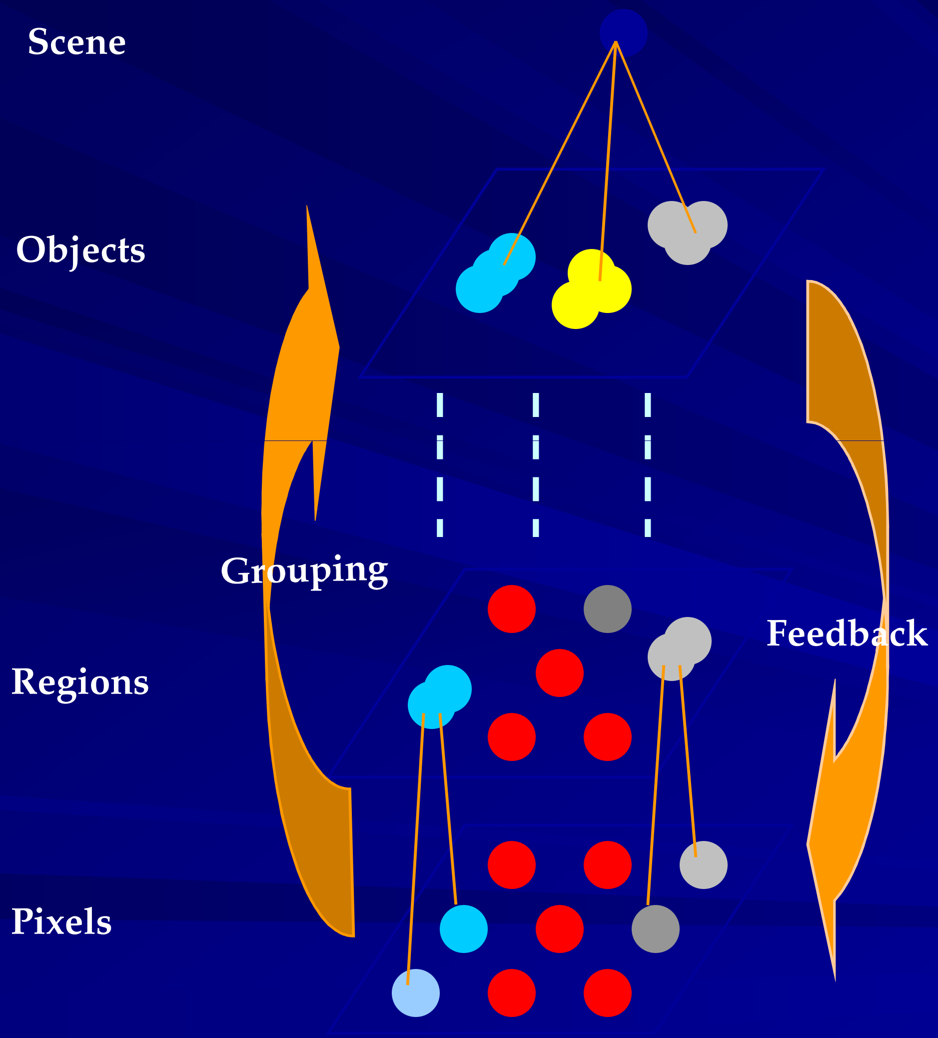
Modeling of Visual Perception for Segmentation

- Three major approaches
  - Statistical modeling (Zhu’96, 98, 03; Ommer’03)
    - Markov Random Field
  - Graph based (Zahn’71, Morris’86, Shi’00, **Li’04**)
    - Shortest Spanning Tree
  - Neural network (Kruger’02; Hasen’02, **Randhal’02 03 08**;)
    - Bayesian network, HCM (Hierarchical Cluster Model)

A Graph-based framework

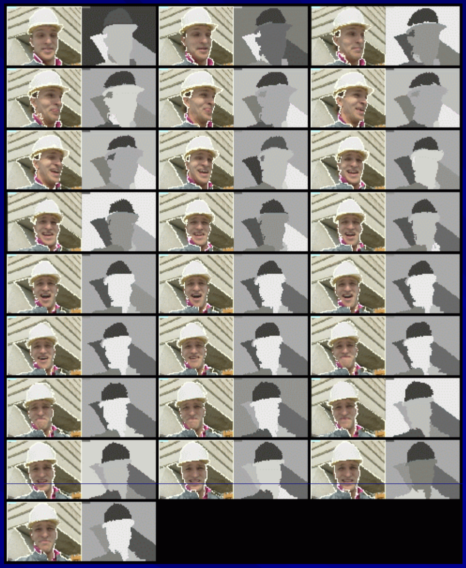
- Representation
  - Visual elements – vertices
  - Relations between elements – edges
  - Hierarchical
- Gestalt Law
  - Reflected by the weights of edges
  - Dynamic calculation at difference level in the hierarchy
- Grouping inference
  - Recursive shortest spanning tree

A Graph-based framework



- Closure
- Symmetry
- Similarity
- Proximity
- Common region
- Good Continuation
- Region size
- Texture
- Color
- illumination

A Graph-based framework



Used Grouping Principles:

- ✓ Uniform connectedness in luminance
- ✓ Element of connectedness
- ✓ Common region & relative size
- ✓ *Similarity*
- ✓ *proximity*

Extension to video:

- ✓ Common fate
- ✓ Synchrony

Neural Network Based Perceptual Grouping

- Hierarchical Cluster Model (Sutton'88)
- Input evidence

$$R_{ij} = \theta * G_{ij} + \tau * S_{ij} + \delta * C_{ij} + \eta * F_{ij}$$

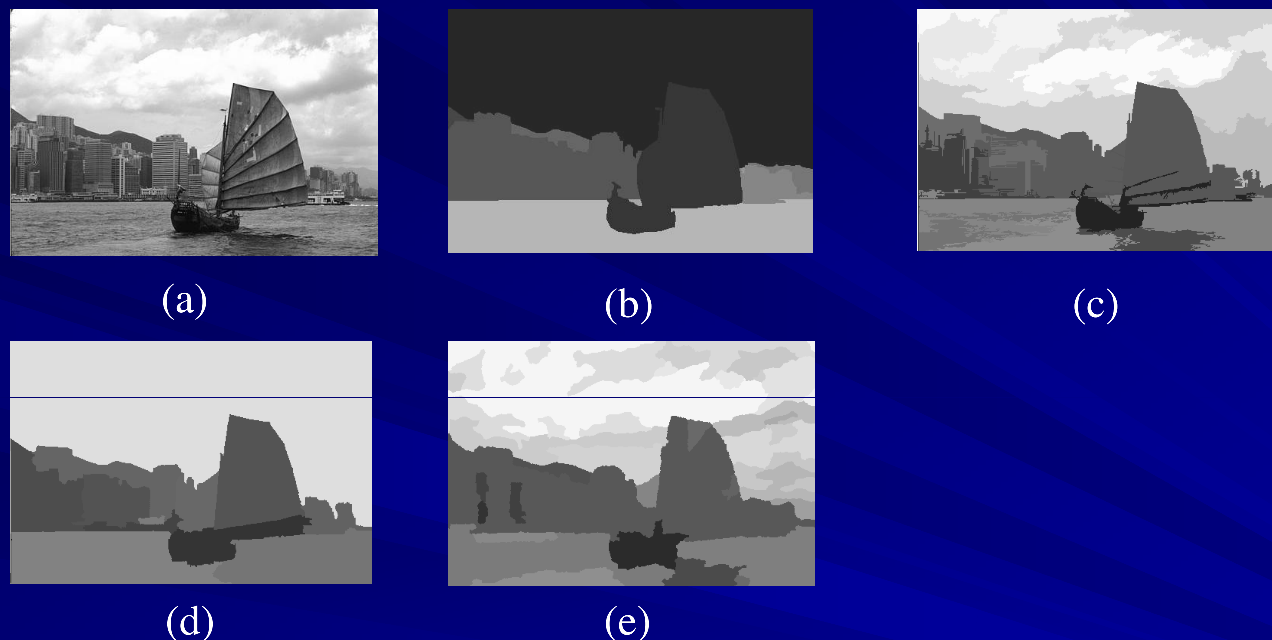
$$I_{ij} = sigmoid(R_{ij})$$

  – G_{ij} - the evidence for merging based on grey-level, e.g.
  – S_{ij} - a structural parameter which gives the evidence for merging based on closure,
  – C_{ij} - the evidence for merging based on continuity and
  – F_{ij} - the evidence for merging based on co-circularity.
  – The weights reflect the relative importance of each feature in the merging and $\theta + \tau + \delta + \eta = 1$

Experimental Results

(a) The original image; (b) Human segmented image obtained by the authors, (c) segmentation by proposed method with grey-level and closure property (LCE=0.025), (d) segmentation by proposed method including Gestalt properties (LCE=0.031) and (e) segmentation by JSEG [2] method (LCE=0.04).

Experimental Results



(a) The original image; (b) Human segmented image [12], (c) segmentation by proposed method with grey-level and closure property (LCE=0.11), (d) segmentation by proposed method including all Gestalt properties (LCE=0.038) and (e) segmentation by JSEG [2] method (LCE=0.07).

Experimental Results

Which one is the "best" ?

(a) The original image; (b) Human segmented image [12], (c) segmentation by proposed method with grey-level and closure property (LCE=0.163), (d) segmentation by proposed method including Gestalt properties (LCE=0.176) and (e) segmentation by JSEG [2] method LCE=0.22).

Evaluation of Segmentation

Motivation

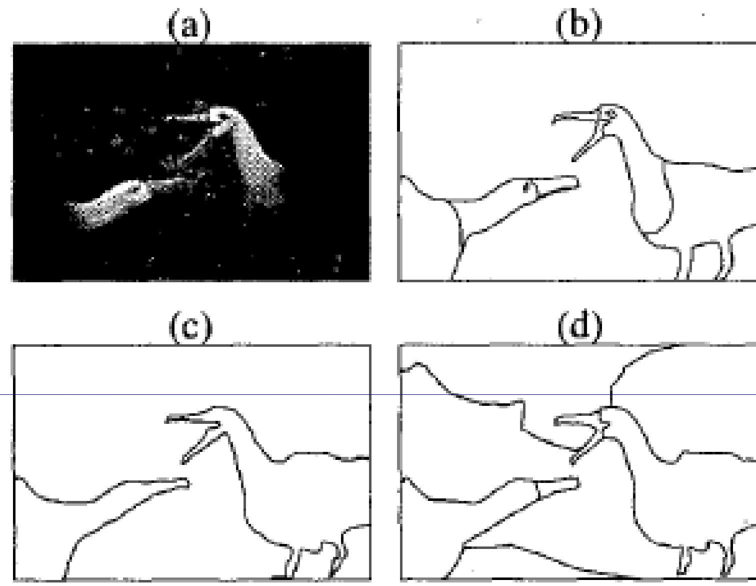


Figure 3: Motivation for making segmentation error measures tolerant to refinement. (a) shows the original image. (b)-(d) show three segmentations in our database by different subjects. (b) and (d) are both *simple refinements* of (c), while (b) and (d) illustrate *mutual refinement*.

D. Martion et al., "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics", ICCV 2001

Segmentation error measure

- **A** segmentation error measure takes two segmentations *S1* and S2 as input, and produces a real valued output in the range [0..1] where zero signifies no error.
- Any error measure should be
  - Tolerant to refinement
  - Independent of the coarseness of pixelation
  - Robust to noise along region boundaries,
  - Tolerant of different segment counts between the two segmentations.

Local Segmentation Error

- Notations
  - $|x|$ denote the cardinality of set x.
  - $R(S1, pi)$ is the set of pixels corresponding to the region in segmentation $S1$ that contains pixel pi
  - Local error between S1 and S2 that contains pixel pi is defined as

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|}$$

where \ is the difference

GCE & LCE

- Global Consistency Error & Local Consistency Error

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\}$$

$$LCE(S_1, S_2) = \frac{1}{n} \sum_i \min \left\{ E(S_1, S_2, p_i), E(S_2, S_1, p_i) \right\}$$

- *LCE <= GCE* for any two segmentations, it is clear that GCE is a tougher measure than LCE

Note on GCE & LCE

- Since both measures are tolerant of refinement, they are meaningful only when comparing two segmentations with an approximately **equal number of segments**.
  - This is because there are two trivial segmentations that achieve zero error:
    - One pixel per segment and one segment for the entire image. The former is

An examples

LCE between segmentation results and human segmentation

Method	LCE	
	Mean	S.D.
JSEG method [2]	0.141389	0.072703
Proposed method ($\theta = 2/3$, $\tau = 1/3$, $\delta = 0$, $\eta = 0$)	0.159722	0.098186
Proposed method ($\theta = 0.3$, $\tau = 0.3$, $\delta = 0.2$, $\eta = 0.2$)	0.109833	0.064409
Human segmentation[a]	~0.07	N/A
NCuts method [25][a]	~0.22	N/A

J. Randall, L. Guan, W. Li and X. Zhang, "The HCM for perceptual image segmentation", Neurocmputing, 71 (2008), pp.1966-1979

Key References

- D. A. Forsyth and J. Ponce, “Computer Vision: A Modern Approach”, Prentice Hall, 2003 – Chapter 14
- R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, 2nd Ed. Prentice Hall, 2002 – Chapter 10
- N. Otsu, “A Threshold Selection Method from Gray-Level Histograms”, IEEE Trans. SMC, SMC-9(1), 1979, pp.62-66
- J. Shi and J. Malik, “Normalized Cuts and Image Segmentation”, IEEE Trans. PAMI, 22(8), 2000, pp.888-905
- D. Martion et al., “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”, ICCV 2001