Computational Photography

Prof. Feng Liu

Spring 2017

http://www.cs.pdx.edu/~fliu/courses/cs510/

05/15/2017

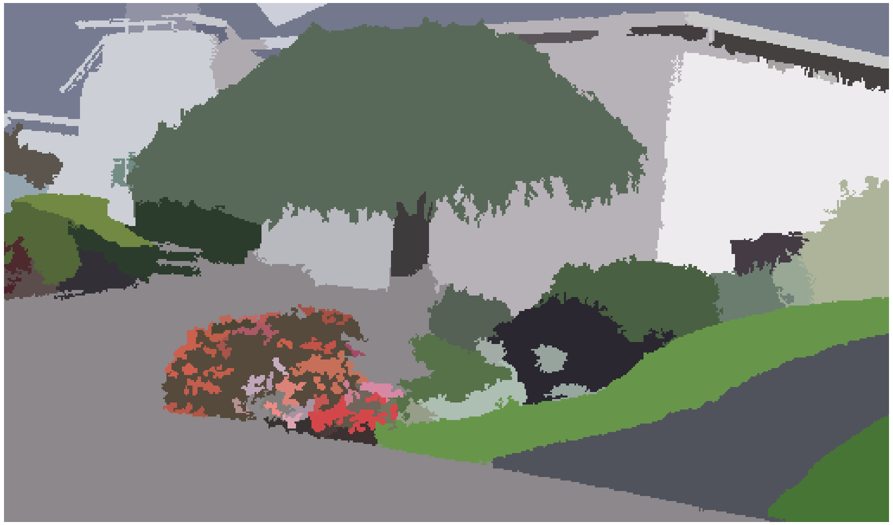
Last Time

- Panorama

Today

- Segmentation

Input



Output

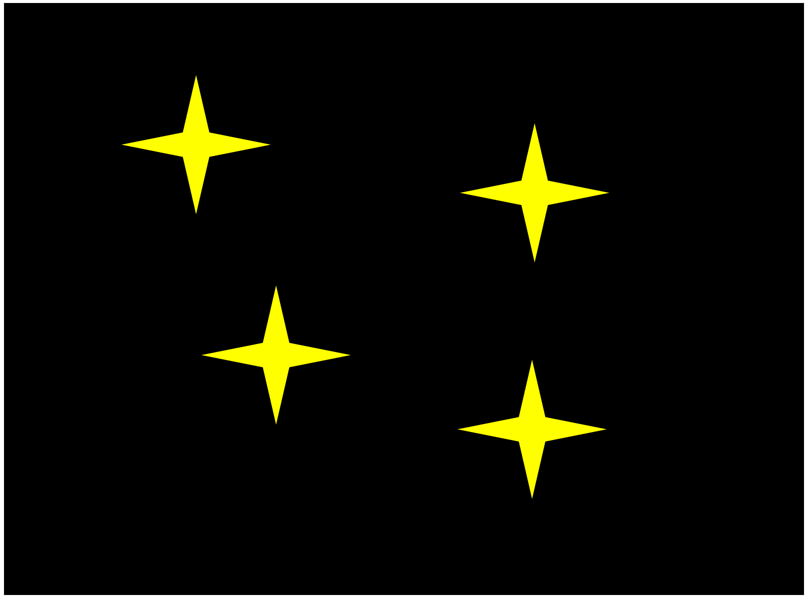
With slides by C. Dyer

Image Segmentation

- How do we know which groups of pixels in a digital image correspond to the objects to be analyzed?
  - Objects may be uniformly darker or brighter than the background against which they appear
    - Black characters imaged against the white background of a page
    - Bright, dense potatoes imaged against a background that is transparent to X-rays
  - Challenging for many other cases

Credit: C. Dyer

Image Segmentation



Right: USFWS Photo by Jim Rorabaugh, http://www.fws.gov/southwest/es/arizona/Reptiles.htm

Image Segmentation: Definitions

- "Segmentation is the process of partitioning an image into semantically interpretable regions."
  - H. Barrow and J. Tennenbaum, 1978
- "An image segmentation is the partition of an image into a set of non-overlapping regions whose union is the entire image. The purpose of segmentation is to decompose the image into parts that are meaningful with respect to a particular application."
  - R. Haralick and L. Shapiro, 1992

Credit: C. Dyer

Image Segmentation: Definitions

- “The neurophysiologists’ and psychologists’ belief that figure and ground constituted one of the fundamental problems in vision was reflected in the attempts of workers in computer vision to implement a process called *segmentation. The* purpose of this process is very much like the idea of separating figure from ground ...”
  - D. Marr, 1982

Credit: C. Dyer

Segmentation methods

- Automatic segmentation
  - Mean-shift
  - Watershed
  - Normalized Cut Method
  - …
- Interactive segmentation
  - Lazy snapping
  - Grab cut
  - Interactive geodesic segmentation
  - …

Normalized Cuts and Image Segmentation

- Normalized Cuts and Image Segmentation
  - J. Shi and J. Malik, *IEEE Trans. Pattern Analysis and Machine Intelligence 22(8), 1997*
    - Citation 5823 according to Google Scholar as 2012-05-21
    - Citation 6417 according to Google Scholar as 2013-05-06
    - Citation 8698 according to Google Scholar as 2014-05-05
    - Citation 10032 according to Google Scholar as 2015-05-04
    - Citation 13215 according to Google Scholar as 2017-05-15
  - Divisive (aka splitting, partitioning) method
  - Hierarchical partitioning
  - Graph-theoretic criterion for measuring goodness of an image partition

Segmentation by Partition

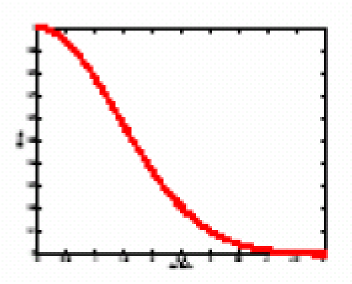
- Criterion for measuring a candidate partitioning:
  - Affinity measure between elements **within each region** is **high, and the affinity between elements across** regions is **low**
  - Affinity: element × element -> R^+
    - Defines the similarity of a pair of data elements.
    - Examples of components of an affinity function: spatial position, intensity, color, texture, motion.

Credit: C. Dyer

Affinity (Similarity) Measures

- Intensity

$$\mathrm{aff}(\mathbf{x},\mathbf{y}) = e^{-\left\| I(\mathbf{x}) - I(\mathbf{y}) \right\|^2 / 2\sigma_I^2}$$



- Distance

$$\mathrm{aff}(\mathbf{x},\mathbf{y}) = e^{-\left\| \mathbf{x} - \mathbf{y} \right\|^2 / 2\sigma_d^2}$$

- Color
- Texture
- Motion

Credit: C. Dyer

Problem Formulation

- Given an undirected graph G = (*V, E), where V is a set* of nodes, one for each data element (e.g., pixel), and *E* is a set of edges with weights representing the affinity between connected nodes
- Find the image partition that maximizes the "association" within each region *and minimizes the* "disassociation" between regions
- Finding the optimal partition is NP-complete

Credit: C. Dyer

Cut

- Let A, B partition G. Therefore $\mathrm{A} \bigcup \mathrm{B} = \mathrm{V}$, and $\mathrm{A} \bigcap \mathrm{B} = \varnothing$
- The affinity or similarity between A and B is defined as

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

= total weight of edges removed

- The optimal bi-partition of G is the one that minimizes *cut*
- *Cut is biased towards small regions*

Credit: C. Dyer

Normalized Cut

- So, instead define the **normalized similarity, called the** *normalized-cut(A,B), as*

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

where $assoc(A,V) = \sum_{i \in A, k \in V} w_{ik}$

= total connection weight from nodes in A to all nodes in G

- *N-cut measures the similarity between regions ("disassociation" measure)*
- *N-cut removes the bias based on region size (usually)*

Credit: C. Dyer

Normalized Cut

- Similarly, define the "**normalized association:**"

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

- *Nassoc* measures how similar, on average, nodes within *the* groups are to each other
- New goal: Find the bi-partition that minimizes *ncut(A,B)* ***and*** maximizes *nassoc(A,B)*
- But, it can be proved that *ncut(A,B) = 2 - nassoc(A,B), so we* can just **minimize *ncut:***
  y = arg min ncut

- Let y be a P = |V| dimensional vector where

$$y_i = \begin{cases} 1, & \text{if node } i \in A \\ -1, & \text{otherwise} \end{cases}$$

- Let $d(i) = \sum_j w_{ij}$
  define the affinity of node *i with all other nodes*

- Let D = P x P diagonal matrix:

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \ldots & 0 \\ 0 & d_2 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & d_\mathrm{P} \end{bmatrix}$$

"degree matrix"

Credit: C. Dyer

Let **A** = P x P symmetric matrix:

"affinity matrix"

$$\mathbf{A} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1P} \\ w_{21} & w_{22} & \cdots & w_{2P} \\ & & \cdots & \\ w_{P1} & w_{P2} & \cdots & w_{PP} \end{bmatrix}$$

It can be shown that

y = arg min$_x$ *ncut*(**x**)

$$= \arg\min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{A})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \text{ subject to } \mathbf{y}^T\mathbf{D}\mathbf{1} = 0$$

Relaxing the constraint on **y** so as to allow it to have real values means that we can **approximate the solution** by solving an equation of the form: $(\mathbf{D} - \mathbf{A})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$

Credit: C. Dyer

The solution, **y**, is an eigenvector of (**D** – **A**)

An eigenvector is a characteristic vector of a matrix and specifies a segmentation based on the values of its components; similar points will hopefully have similar eigenvector components.

Theorem: If **M** is any real, symmetric matrix and **x** is orthogonal to the *j*-1 smallest eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}$, then $\mathbf{x}^T\mathbf{M}\mathbf{x} / \mathbf{x}^T\mathbf{x}$ is minimized by the next smallest eigenvector $\mathbf{x}_j$ and its minimum value is the eigenvalue λ_j

Credit: C. Dyer

Smallest eigenvector is always 0

because $A=V$, $B=\{\}$ means $ncut(A,B)=0$

Second smallest eigenvector is the real-valued **y** that minimizes *ncut*

Third smallest eigenvector is the real-valued **y** that optimally sub-partitions the first two regions

Note: Converting from the real-valued **y** to a binary-valued **y** introduces errors that will propagate to each sub-partition

NCUT Segmentation Algorithm

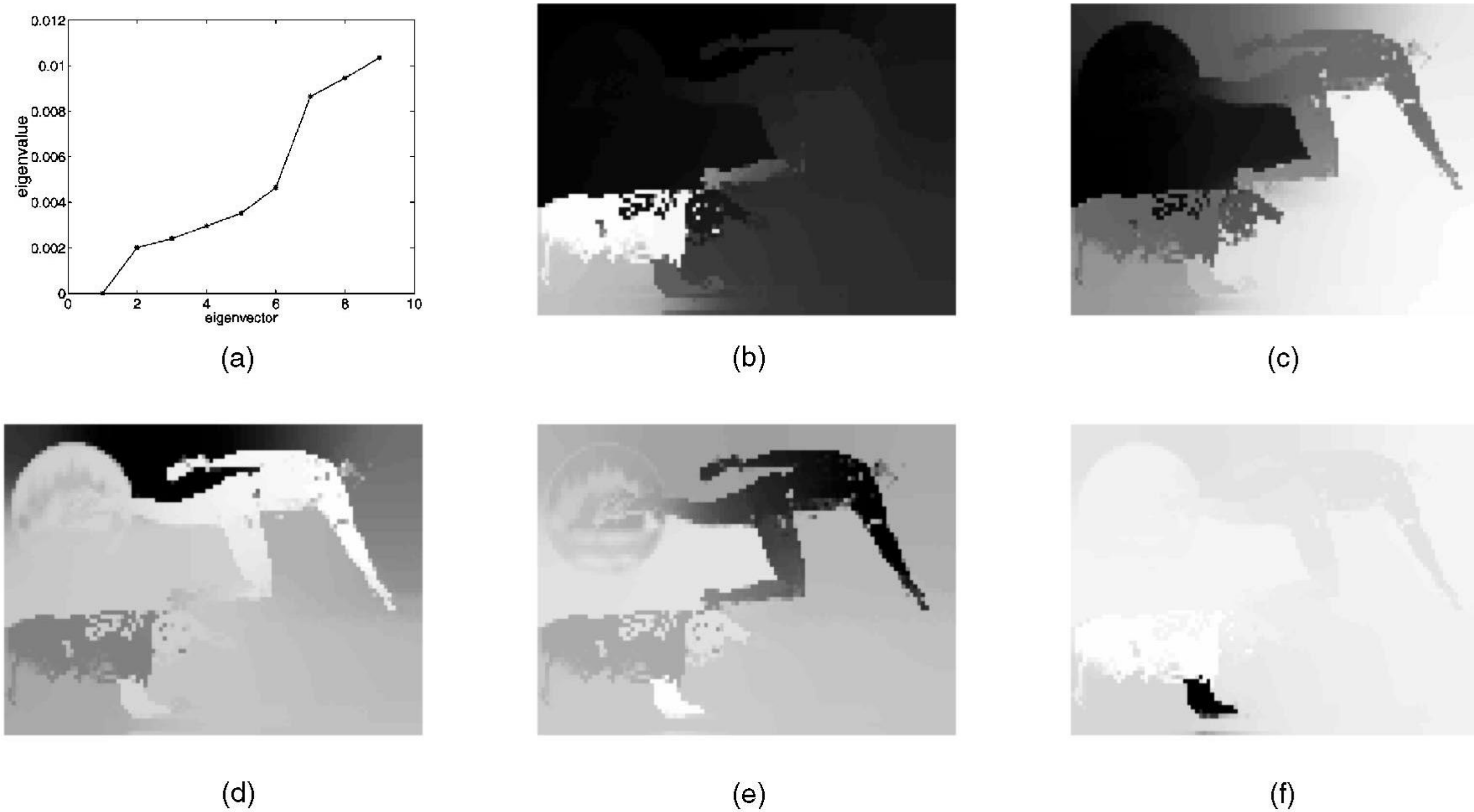
1. Set up problem as G = (V,E) and define affinity matrix **A** and degree matrix **D**
2. Solve $(\mathbf{D} - \mathbf{A})\mathbf{x} = \lambda \mathbf{D}\mathbf{x}$ for the eigenvectors with the smallest eigenvalues
3. Let $\mathbf{x}_2$ = eigenvector with the 2nd smallest eigenvalue λ_2
4. Threshold $\mathbf{x}_2$ to obtain the binary-valued vector $\mathbf{x}'_2$ such that $ncut(\mathbf{x}'_2) \geq ncut(\mathbf{x}^t_2)$ for all possible thresholds t
5. For each of the two new regions, if *ncut* < threshold T, then recurse on the region

Credit: C. Dyer

Example

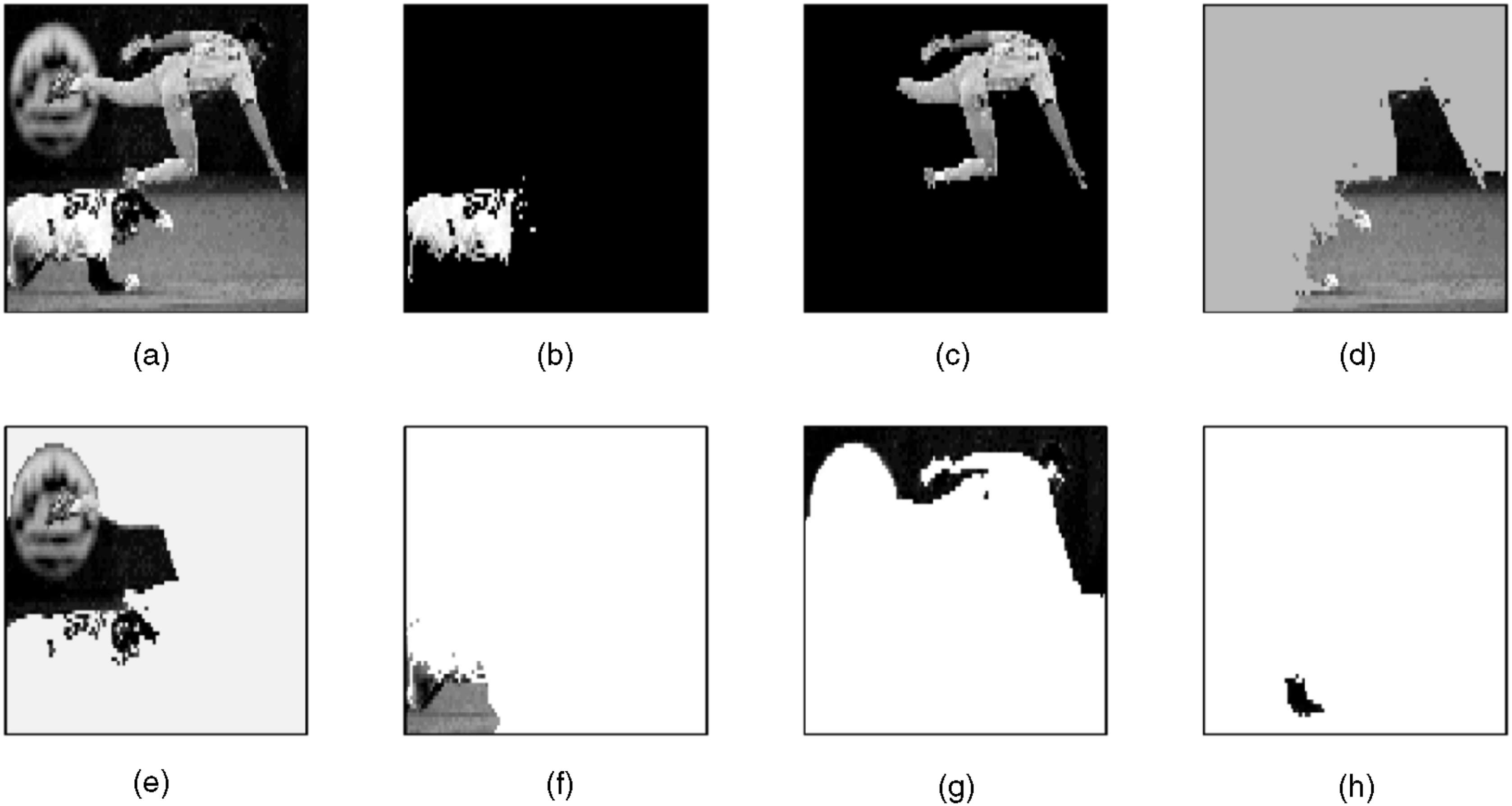
Input image

Eigen values and vectors



Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(f) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

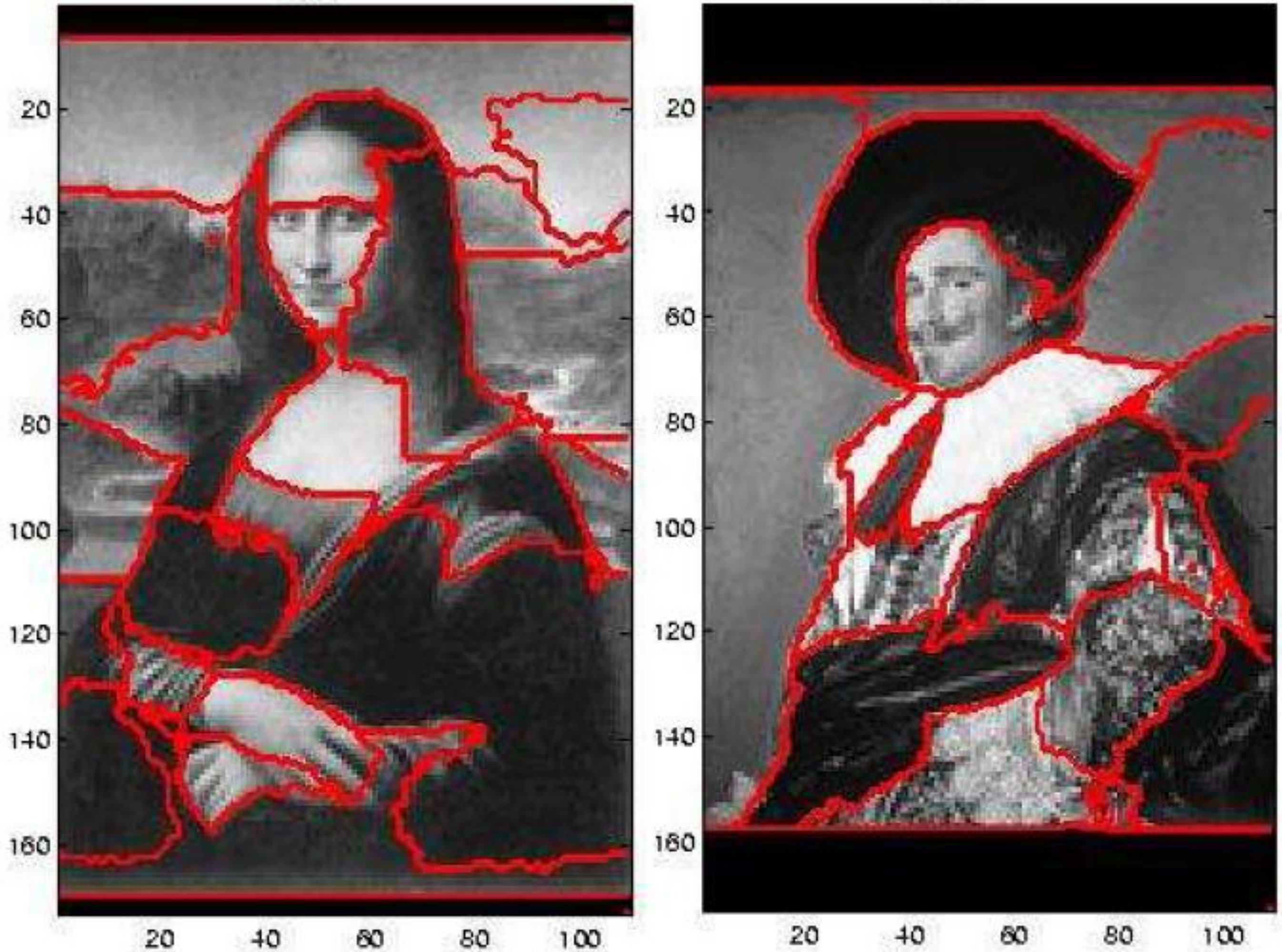
Reprint from [Shi and Malik 97]



(a) shows the original image of size 80x100. Image intensity is normalized to lie within 0 and 1. Subplots (b)-(h) show the components of the partition with Ncut value less than 0.04.

Reprint from [Shi and Malik 97]

More examples



Credit: C. Dyer

Comments on NCUT

- Recursively bi-partitions the graph instead of using the 3rd, 4th, etc. eigenvectors for robustness reasons (due to errors caused by the binarization of the real-valued eigenvectors)
- Solving standard eigen-value problems takes $O(P^3)$ time
- Can speed up algorithm by exploiting the "locality" of affinity measures, which implies that **A is sparse (nonzero** values only near the diagonal) and (**D – A) is** sparse. This leads to a $O(P^{1.5})$ *time algorithm*

Credit: C. Dyer

Next Time

- Interactive segmentation
- Matting