LECTURE 11: SEGMENTATION

Venu Madhav Govindu
Department of Electrical Engineering
Indian Institute of Science, Bengaluru

2022

In this lecture we shall look at image **segmentation**

Segmentation

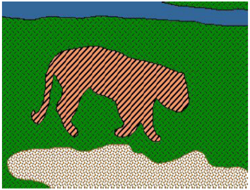


Image Segmentation

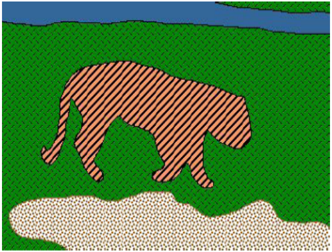
- Find groups of pixels (meaningful)
- Why segment ?
  - Too many pixels
  - Focus on relevant information
  - Helps downstream processing
- Related terms:
  - segmentation
  - grouping
  - perceptual organization

Segmentation

Image Segmentation

- What the basis of this grouping?
- Context dependent
- Ambiguous with multiple interpretations
- General principles based on image primitives alone
- Will **not** consider motion segmentation
- Consider a few well-known examples
- There are **many more** methods in the literature
- Conventional approaches to segmentation (this lecture)
- Learning based methods (next lecture)

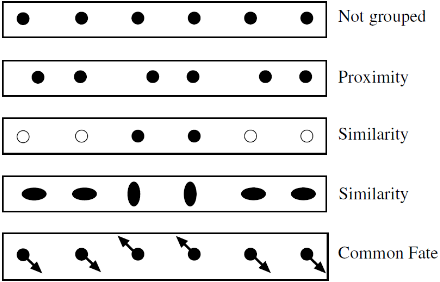
Segmentation

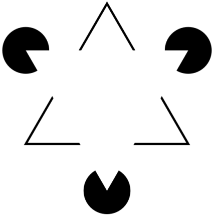


Principles ?

- Notion of similarity
- Spatial relationships in images matter
- Local+Global both matter (scale)
- Global relationship hard to define
- See later how learning methods handle it

Segmentation



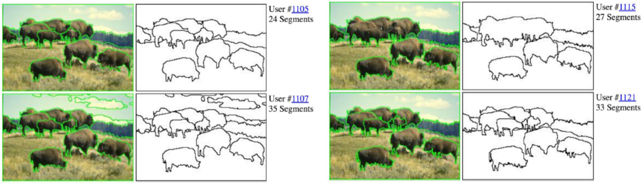


Grouping Principles

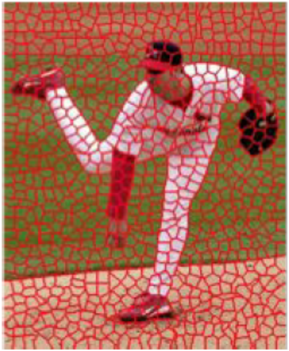
- Gestalt principles
- Translate to algorithms ?
- Why segment is an important question

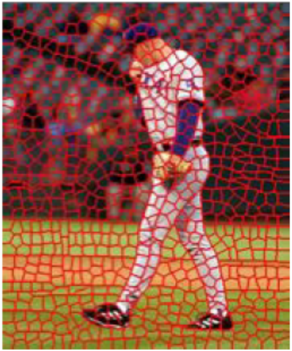
Forstyth+Ponce 2nd edition; Wikipedia

Segmentation



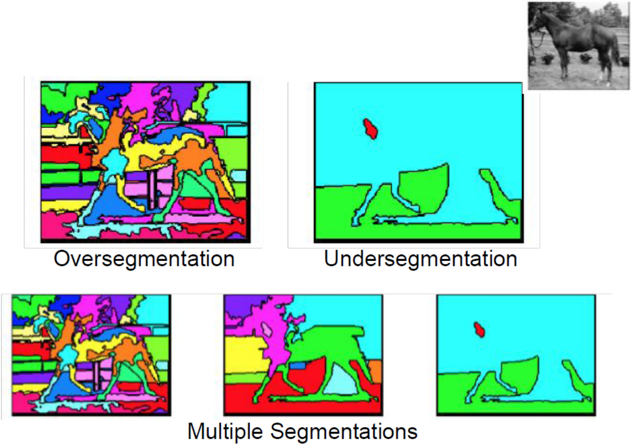
Segmentation





Segmentation

Segmentation



Segmentation

Segmentation Approaches

- Clustering or grouping features
- Graph based approaches
- Encode desired outcomes into methods

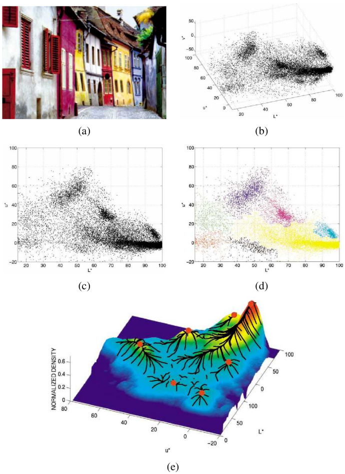
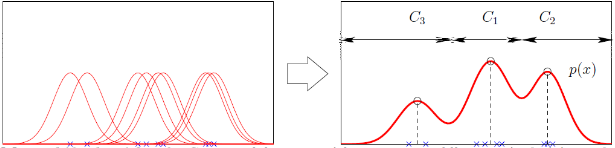


Figure 7.58 *Mean-shift image segmentation (Comaniciu and Meer 2002) © 2002 IEEE: (a) input color image; (b) pixels plotted in L*u*v* space; (c) L*u* space distribution; (d) clustered results after 159 mean-shift procedures; (e) corresponding trajectories with peaks marked as red dots.*

Segmentation



Mean-shift algorithm for Gaussian kde:maxima(also minima, saddle points) of $p(\mathbf{x})$ satisfy

Mean Shift

- Estimate density function given samples

$$f(x) = \sum_i k(x - x_i) = \sum_i G\left(\frac{\|x - x_i\|^2}{h^2}\right)$$

Parzen window method (assume a Gaussian kernel)

https://faculty.ucmerced.edu/mcarreira-perpinan/teaching/ee589/lecture-notes.pdf

Mean Shift

- Given f(x) can find local maxima
- Too expensive to evaluate f(x) in h-d space
- Solution : find mode of pdf directly
- Start at a random point
- Assume a Gaussian kernel

$$
\begin{aligned}
f(x) &= \frac{1}{N}\sum_{i=1}^{N} K(x - x_i) \\
\Rightarrow \nabla f(x) &= \frac{1}{N}\sum_{i=1}^{N} \nabla K(x - x_i)
\end{aligned}
$$

Assume Gaussian kernel

$$
\begin{aligned}
K(x - x_i) &= G\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \\
\Rightarrow \nabla K(x - x_i) &= -G^{'}\left(\left\|\frac{x - x_i}{h}\right\|^2\right)(x_i - x)
\end{aligned}
$$

Now denote $G^{'}\left(\left\|\frac{x - x_i}{h}\right\|^2\right) = g_i$

$$
\begin{aligned}
\nabla f(x) &= \frac{1}{N} \sum_{i=1}^{N} g_i . (x_i - x) \\
&= \frac{1}{N} \left(\sum g_i x_i - \sum g_i x \right) \\
&= \frac{1}{N} \left(\sum g_i \right) \underbrace{\left[\frac{\sum g_i x_i}{\sum g_i} - x \right]}_{m(x)}
\end{aligned}
$$

$m(x)$ is the mean shift vector

Mean Shift

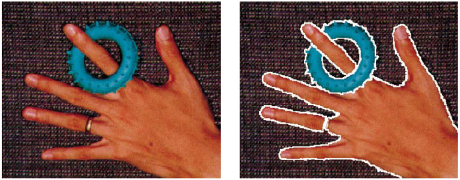
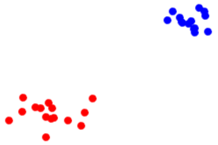


Figure 5.18 Mean-shift color image segmentation with parameters $(h_s, h_r, M) = (16, 19, 40)$ (Comaniciu and Meer 2002) © 2002 IEEE.

Figure: Colour image segmentation using mean shift

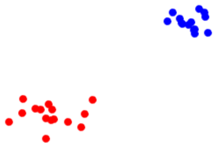
- $\boldsymbol{K}(\boldsymbol{x}_j) = k\left(\frac{||x_r||^2}{h_r^2}\right) k\left(\frac{||x_s||^2}{h_s^2}\right)$
- Spatial location : $x_s = (x, y)$
- Colour values : x_r
- $\boldsymbol{x}_j$ contains both spatial location and colour value
- M specifies size below which clusters are discarded

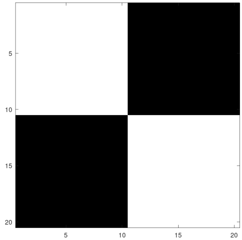
SPECTRAL CLUSTERING



- Powerful linear algebraic approach
- Similarity matrix of relationships
- Eigen-decomposition of similarity matrix
- Well studied and used in vision problems - NCut etc.
- Recent extensions use multiple eigen-vectors that encode segmentation

SPECTRAL CLUSTERING





- Powerful linear algebraic approach
- Similarity matrix of relationships
- Eigen-decomposition of similarity matrix
- Well studied and used in vision problems - NCut etc.
- Recent extensions use multiple eigen-vectors that encode segmentation

SPECTRAL CLUSTERING

Graph Laplacians

- Main tool for spectral clustering
- Graph Laplacian Matrix : $L = D - W$
- W is undirected weighted graph $(w_{ij} = w_{ji})$
- D is the degree matrix $d_i = \sum_j w_{ij}$

Unnormalized Graph Laplacians

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

- L is symmetric and positive semi-definite
- Smallest eigen-vector/value are $\mathbb{1}$ and 0 resp.
- L has n non-negative, real-valued eigenvalues, $0 = \lambda_1 \leq \lambda_2 \leq \cdots \lambda_n$

Normalized Graph Laplacians

- Two versions in literature
- $D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$
- $D^{-1}L = I - D^{-1}W$
- Usually solve generalised eigen-value problem $Lu = \lambda Du$

SPECTRAL CLUSTERING

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $v_1, \ldots, v_k$ of L.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \ y_j \in C_i\}$.

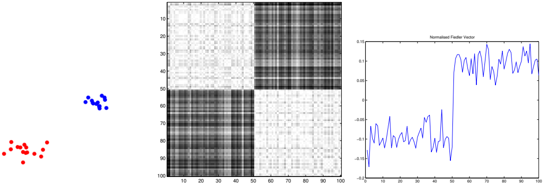
Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $v_1, \ldots, v_k$ of the generalized eigenproblem $Lv = \lambda Dv$.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

SPECTRAL CLUSTERING



Similarity Measure

Similarity based on Euclidean Distance

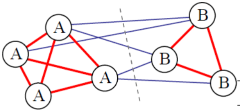
$$\boldsymbol{W}(\boldsymbol{i},\boldsymbol{j}) = e^{-\frac{1}{2\sigma^2}(\boldsymbol{p}_j-\boldsymbol{p}_i)^T(\boldsymbol{p}_j-\boldsymbol{p}_i)}$$

σ : expected data noise level (control parameter)

Eigen-Decomposition

- $\boldsymbol{W}(i,j) = e^{-\frac{1}{2\sigma^2}(\boldsymbol{p}_j - \boldsymbol{p}_i)^T(\boldsymbol{p}_j - \boldsymbol{p}_i)}$
- Define $\boldsymbol{D}$: $\boldsymbol{D}(i,i) = \sum_j \boldsymbol{W}(i,j)$
- *Laplacian matrix* : $\boldsymbol{D} - \boldsymbol{W}$
- *Normalised Laplacian* : $\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{D}^{-\frac{1}{2}}$
- *Fiedler* vector : second smallest eigen-vector

Normalized Cuts



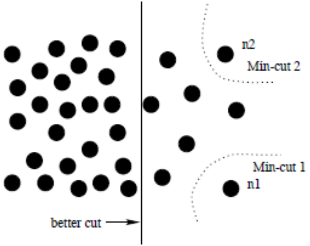
	A	B	sum
A	$assoc(A, A)$	$cut(A, B)$	$assoc(A, V)$
B	$cut(B, A)$	$assoc(B, B)$	$assoc(B, V)$
sum	$assoc(A, V)$	$assoc(B, v)$	

Segmentation using Affinities

- Consider graph of similarity relations $G = (V, E)$
- In our case V can be image pixels
- E : edges between neighbouring pixels (say)
- A cut
  - Deletes some edges
  - Separates vertices into two sets
  - Results in two segments

Normalized Cuts

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



Properties of a Cut

- Cut between two sets of vertices A and B
- $A \cup B = V$ and $A \cap B = \varnothing$
- Cost of cut is the sum of weights of deleted edges
- Can result in degenerate solutions (minimisation)
- Cut isolated vertex

Normalized Cuts

Modify to better reflect desired properties
Replace

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

with

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

Normalized Cuts

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

where

$$assoc(A,A) = \sum_{i \in A, j \in A} w_{ij}$$

$$assoc(A,V) = assoc(A,A) + cut(A,B)$$

- $assoc(A,A)$: association within cluster A
- $assoc(A,V)$: sum of all weights with vertices in A
- Normalization for relative sizes of A and B

Normalized Cuts

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

Hardness of Cut

- Solving for optimal cut is NP-complete
- Can be solved by relaxing labels
- Define indicator $\mathbf{x}$
- $x_i = +1 \iff i \in A$
- $x_i = -1 \iff i \in B$

Normalized Cuts

- Define $\mathbf{d} = \mathbf{W1}$
- Define $\mathbf{D} = diag(\mathbf{d})$
- Let $\mathbf{y} = ((\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x}))/2$ s.t. $\mathbf{y}.\mathbf{d} = 0$
- Normalized cuts problem can be redefined

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}$$

Normalized Cuts

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}$$

- Cost function solving by minimising Rayleigh quotient
- Equivalent to solving $(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$
- Equivalent to $(\mathbf{I}-\mathbf{N})\mathbf{z} = \lambda\mathbf{z}$
- $\mathbf{N} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ (Normalized affinity matrix)
- $\mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y}$
- Laplacian matrix $\mathbf{L} = \mathbf{I}-\mathbf{N}$
- Spectral Clustering

Normalized Cuts

Issues and Details

- Multiple segmentation carried out recursively
- Can also use multiple eigenvectors to classify
- Weights can capture spatial and image similarity relationships

$$w_{ij} = exp\left(-\frac{||\mathbf{F}_i - \mathbf{F}_j||^2}{2\sigma_F^2} - \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_s^2}\right)$$

- Original method is slow
- Many variations and modifications to speed up
- Modern methods : Discrete optimization on MRF's
- Neighbourhood need not be local
- $\mathbf{L}$ is sparse

Normalized Cuts

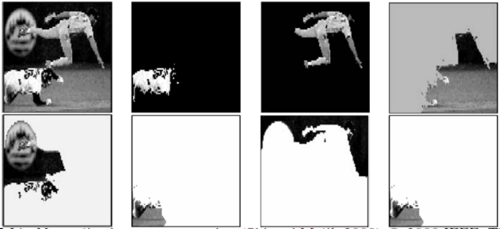


Figure 5.21 Normalized cuts segmentation (Shi and Malik 2000) © 2000 IEEE: The input image and the components returned by the normalized cuts algorithm.

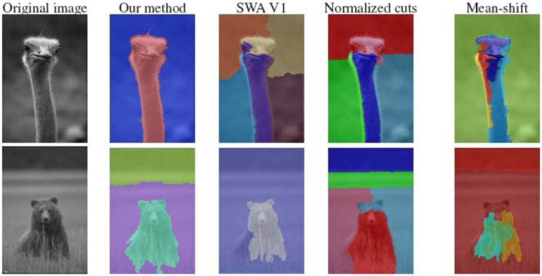


Figure 5.22 Comparative segmentation results (Alpert, Galun, Basri *et al.* 2007) © 2007 IEEE. "Our method" refers to the probabilistic bottom-up merging algorithm developed by Alpert *et al.*