Lecture 07: Clustering and Segmentation II

[AIX7021] Computer Vision

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

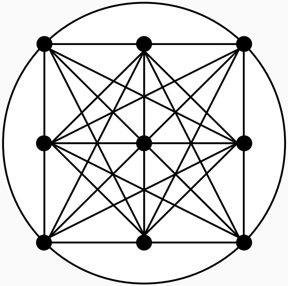
Tentative schedule

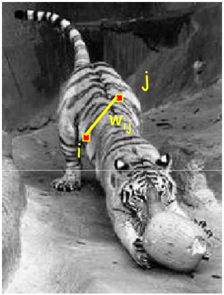
week	topic	date
1	Introduction and Basics	09.01
2	Image process I	09.08
3	Image process II	09.15
4	(휴강)	09.22
5	Feature detection and matching I	09.29
6	Feature detection and matching II	10.06
7	Clustering and segmentation I & II	10.13 & 10.16
8	**Mid-term exam**	**10.20**
9	Robust Fitting and Matching	10.27
10	Boosting and Face Detection	11.03
12	Dimensional Reduction and Face Recognition	11.10
13	Object recognition	11.17
14	Motion and Tracking	11.24
11	Image Classification	12.01
15	**Final exam**	**12.08**

Graph cut

Segmentation by graph partitioning

Image as a graph

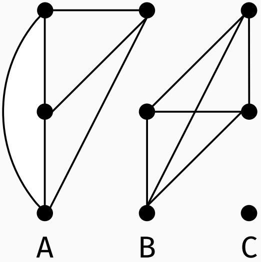




- Node for every pixel
- Edge between every pair of pixels or every pair of “sufficiently close” pixels
- Each edge is weighted by the *affinity* or similarity of the two nodes

Segmentation by graph partitioning

Break graph into segments / clusters

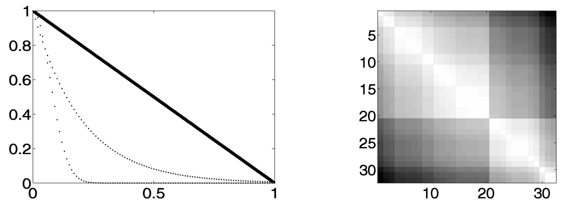


- Break links that have low affinity
- Similar pixels should be in the same segment
- Dissimilar pixels should be in different segments

Measuring affinity

Affinity matrix

$$W_{ij} = \exp\left(-\frac{\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right) \quad (1)$$



- $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)$: distance between two pixels based on
  - Feature
  - Pixel location
  - Feature + location
- Small σ: group only nearby points
- Large σ: group far-away points

Eigenvectors for clustering

We want to find a vector $\mathbf{x}$ giving the association between each element and a cluster

- The affinities of data within the same cluster should be high
- The affinities of data between the different clusters should be low

Summation of affinities

$$\max \underbrace{\sum_{i \in C_{+1}, j \in C_{+1}} w_{ij} + \sum_{i \in C_{-1}, j \in C_{-1}} w_{ij}}_{\text{within clusters}} - 2 \underbrace{\sum_{i \in C_{+1}, j \in C_{-1}} w_{ij}}_{\text{between clusters}} \tag{2}$$

$$\iff \max \sum_i \sum_j w_{ij} x_i x_j \quad \text{where } x_i \in \{+1, -1\} \tag{3}$$

$$\iff \max \mathbf{x}^T \mathbf{W} \mathbf{x} \tag{4}$$

$$\tag{5}$$

Eigenvectors for clustering

Problem formulation

$$\text{maximize} \quad \mathbf{x}^T \mathbf{W} \mathbf{x} \tag{6}$$

$$\text{subject to} \quad \mathbf{x}^T \mathbf{x} = 1 \tag{7}$$

- Always favor large x_i's since all affinities are positives
- Add a constraint: limit the size of $\mathbf{x}$ and normalize the objective function with the same size coefficient vector

Equivalently,

$$\text{maximize} \quad \mathbf{x}^T \mathbf{W} \mathbf{x} \quad \text{subject to} \quad \mathbf{x}^T \mathbf{x} = 1 \tag{8}$$

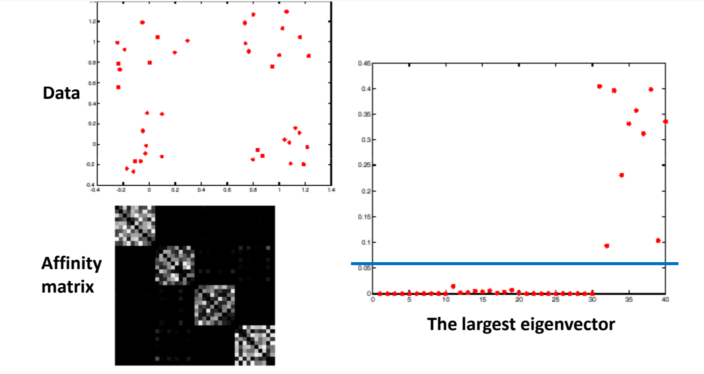
$$\Longleftrightarrow \text{maximize} \quad \frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \tag{9}$$

Hence,

$$\frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \lambda \quad \Longleftrightarrow \quad \mathbf{W} \mathbf{x} = \lambda \mathbf{x} \tag{10}$$

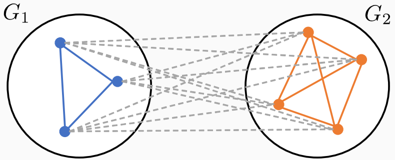
$$\text{i.e., find the maximum eigenvector!} \tag{11}$$

Example of eigenvector



Alternative understanding of objective function

Alternative representation



$$\max \frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \quad x_i \in \{0, 1\} \tag{12}$$

$$\iff \quad \mathbf{x}^T \mathbf{W} \mathbf{x} = \mathrm{assoc}(A, A) = \sum_{i \in A, j \in A} w_{ij}, \ \mathbf{x}^T \mathbf{x} = |A| \quad \text{where } \mathbf{W} = \left[\begin{array}{c|c} A & \cdot \\ \hline \cdot & B \end{array} \right] \tag{13}$$

Hence, equivalently graph cut by average assoc.

$$\max \frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \max \frac{\mathrm{assoc}(A, A)}{|A|} \quad \iff \quad \max \frac{\mathrm{assoc}(A, A)}{|A|} + \frac{\mathrm{assoc}(B, B)}{|B|} \tag{14}$$

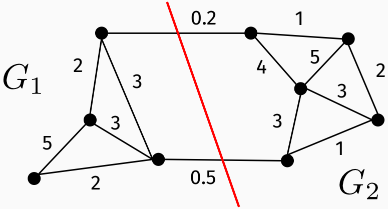
Minimum cut

Technique to cut a graph optimally: what is optimal?

- Minimize the cost of a cut
- Cost of a cut: sum of weights to cut edges

Minimum graph cut: segmentation by finding the *minimum cut* in a graph

$$\mathrm{cut}(G_1, G_2) = \sum_{i \in G_1, j \in G_2} w_{ij} \tag{15}$$

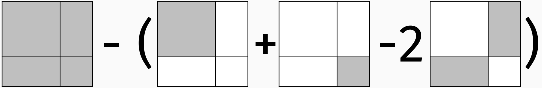


Minimum cut

Minimum cut algorithm is identical to the problem we solved

$$\max \sum_{i \in C_{+1}, j \in C_{+1}} w_{ij} + \sum_{i \in C_{-1}, j \in C_{-1}} w_{ij} - 2 \sum_{i \in C_{+1}, j \in C_{-1}} w_{ij} \tag{16}$$

$$\iff \min \sum_i \sum_j w_{ij} - \left(\sum_{i \in C_{+1}, j \in C_{+1}} w_{ij} + \sum_{i \in C_{-1}, j \in C_{-1}} w_{ij} - 2 \sum_{i \in C_{+1}, j \in C_{-1}} w_{ij} \right) \tag{17}$$



$$\iff \min \sum_{i \in C_{+1}, j \in C_{-1}} w_{ij} \quad \text{This is equivalent to min-cut!} \tag{18}$$

Minimum cut

Focus on the following objective function:

$$\min \underbrace{\sum_i \sum_j w_{ij}}_{\mathbf{x}^T\mathbf{D}\mathbf{x}} - \underbrace{\left(\sum_{i \in C_{+1}, j \in C_{+1}} w_{ij} + \sum_{i \in C_{-1}, j \in C_{-1}} w_{ij} - 2 \sum_{i \in C_{+1}, j \in C_{-1}} w_{ij} \right)}_{\mathbf{x}^T\mathbf{W}\mathbf{x}} \tag{19}$$

$$\text{where } \mathbf{D} = \mathrm{diag}([d_1, d_2, ..., d_n]) \text{ and} \tag{20}$$

$$d_i = \sum_j w_{ij} \quad \text{: row sum of affinity matrix} \tag{21}$$

$$\iff \quad \min \mathbf{x}^T\mathbf{D}\mathbf{x} - \mathbf{x}^T\mathbf{W}\mathbf{x} = \min \mathbf{x}^T(\mathbf{D} - \mathbf{W})\mathbf{x} \tag{22}$$

Solution of min-cut problem

- we have the same problem, but the solution is slightly different

$$\min \mathbf{x}^T(\mathbf{D} - \mathbf{W})\mathbf{x} \tag{23}$$

- convert the discrete problem to continuous domain

$$\text{minimize } \mathbf{x}^T(\mathbf{D} - \mathbf{W})\mathbf{x} \quad \text{subject to } \mathbf{x}^T\mathbf{x} = 1 \tag{24}$$

$$\iff \min \frac{\mathbf{x}^T(\mathbf{D} - \mathbf{W})\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{25}$$

- we need to find the minimum λ to satisfy the following equation

$$(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{x} \tag{26}$$

- find the second minimum eigenvector, why?

Minimum cut: alternative understanding of objective function

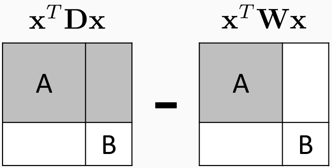
Alternative representation

$$\min \frac{\mathbf{x}^T(\mathbf{D}-\mathbf{W})\mathbf{x}}{\mathbf{x}^T\mathbf{x}}, \quad x_i \in \{0,1\} \tag{27}$$

$$\mathbf{x}^T\mathbf{D}\mathbf{x} = \mathrm{assoc}(A,V) = \sum_{i\in A, j\in V} w_{ij} \tag{28}$$

$$\mathbf{x}^T\mathbf{W}\mathbf{x} = \mathrm{assoc}(A,A) = \sum_{i\in A, j\in A} w_{ij} \tag{29}$$

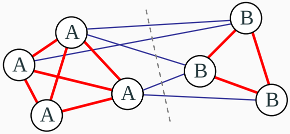
$$\mathrm{cut}(A,B) = \mathrm{assoc}(A,V) - \mathrm{assoc}(A,A) \tag{30}$$



Hence, equivalently graph cut by average cut.

$$\min \frac{\mathbf{x}^T(\mathbf{D}-\mathbf{W})\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \min \frac{\mathrm{cut}(A,B)}{|A|} \iff \min \frac{\mathrm{cut}(A,B)}{|A|} + \frac{\mathrm{cut}(A,B)}{|B|} \tag{31}$$

Minimum cut: alternative understanding of objective function



(a)

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

(b)

Figure 5.19 Sample weighted graph and its normalized cut: (a) a small sample graph and its smallest normalized cut; (b) tabular form of the associations and cuts for this graph. The $assoc$ and cut entries are computed as area sums of the associated weight matrix $\boldsymbol{W}$ (Figure 5.20). Normalizing the table entries by the row or column sums produces normalized associations and cuts $Nassoc$ and $Ncut$.

Average association vs. average cut

Average association

$$\max \frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \Leftrightarrow \max \frac{\mathrm{assoc}(A, A)}{|A|} + \frac{\mathrm{assoc}(B, B)}{|B|} \tag{32}$$

Average cut

$$\min \frac{\mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \Leftrightarrow \min \frac{\mathrm{cut}(A, B)}{|A|} + \frac{\mathrm{cut}(A, B)}{|B|} \tag{33}$$

Consider

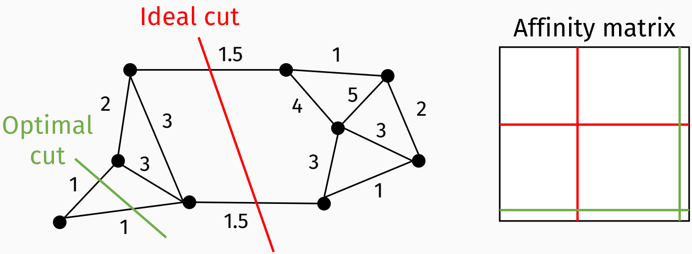
$$\max \mathbf{x}^T \mathbf{W} \mathbf{x} \Leftrightarrow \min \mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}, \quad x_i \in \{-1, +1\} \tag{34}$$

but, why?

$$\max \frac{\mathbf{x}^T \mathbf{W} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \not\Leftrightarrow \min \frac{\mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \quad x_i \in \{-1, +1\} \tag{35}$$

Favor for small and isolated clusters

$$\mathrm{cut}(G_1, G_2) = \sum_{i \in G_1, j \in G_2} w_{ij} \tag{36}$$



Normalized cut

Normalized cut

Drawback of minimum cut method: minimum cut tends to cut off very small, isolated components

Normalized cut: this can be fixed by normalizing the cut by the weight of all the edges incident to the segment

$$\mathrm{Ncut}(G_1, G_2) = \frac{\mathrm{cut}(G_1, G_2)}{\mathrm{assoc}(G_1, G)} + \frac{\mathrm{cut}(G_1, G_2)}{\mathrm{assoc}(G_2, G)} \tag{37}$$

$$\text{where } \mathrm{assoc}(G_1, G) = \sum_{i \in G_1, j \in G} w_{ij} \tag{38}$$

Solving normalized cut

Re-write the objective function

- $\mathbf{W}$: affinity matrix
- $\mathbf{D}$: diagonal matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$
- $\mathbf{y}$: binary vector representing the cluster association

$$\begin{array}{l} \mathbf{x}_i \in \{+1, -1\} \\ \mathbf{y}_i \in \{+1, -b\} \end{array} \quad \text{where } b = \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} \text{ and } \mathbf{y}^T \mathbf{D} \mathbf{1} = 0 \tag{39}$$

$$\mathrm{Ncut}(G_1, G_2) = \frac{\mathrm{cut}(G_1, G_2)}{\mathrm{assoc}(G_1, G)} + \frac{\mathrm{cut}(G_1, G_2)}{\mathrm{assoc}(G_2, G)} \tag{40}$$

$$= \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}} \tag{41}$$

This favors partitioning with equal size segments!

Solving normalized cut

How to find the optimal solution for the objective function

$$\min \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \tag{42}$$

- Finding the exact minimum of the normalized cut cost is NP-complete
- Relax $\mathbf{y}$ to take on arbitrary values, then minimize the relaxed cost

$$\frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} = \lambda \iff \mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{y}^T\mathbf{D}\mathbf{y} \tag{43}$$

We need to find $\mathbf{y}$ minimizing λ in

$$(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y} \tag{44}$$

which is identical to find $\mathbf{z}$ in

$$\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}\mathbf{z} = \lambda\mathbf{z} \text{ where } \mathbf{z} = \mathbf{D}^{1/2}\mathbf{y} \tag{45}$$

Solving normalized cut

How to find the optimal solution for the objective function

$$\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}\mathbf{z}=\lambda\mathbf{z} \quad \text{where } \mathbf{z}=\mathbf{D}^{1/2}\mathbf{y} \tag{46}$$

- The solution is given by the generalized eigenvector corresponding to the second smallest eigenvalue

$$\mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-1/2}: \text{symmetric \& positive semi-definite} \tag{47}$$

$$\mathbf{z}_0=\mathbf{D}^{1/2}\mathbf{1}: \text{corresponding to zero eigenvalue} \tag{48}$$

Converting the solution in continuous domain for discrete problem

- $\mathbf{y}_i$: “soft” indication of the component membership of the i-th pixel
- Binary classification by simple thresholding: use 0 or median value
- Note: the solution is not optimal!

Normalized cut algorithm

Normalized cut algorithm

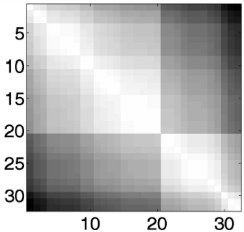
1. Represent the image as a weight graph $G = (V, E)$, compute the weight of each edge, and summarize the information in $\mathbf{D}$ and $\mathbf{W}$
2. Solve $(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda \mathbf{D}\mathbf{y}$ for the eigenvector with the second smallest eigenvalue
3. Use the entries of the eigenvector to bipartition the graph

Original normalized cut is a binary clustering method

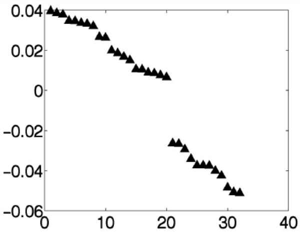
Extension to more than 2 classes

- Recursively bipartition the graph
- Run k-means clustering on values of several eigenvectors

Normalized cut



(a)



(b)

Figure 5.20 Sample weight table and its second smallest eigenvector (Shi and Malik 2000) © 2000 IEEE: (a) sample 32×32 weight matrix $\boldsymbol{W}$; (b) eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem $(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}$.

Normalized cut

Fig. 2. A gray level image of a baseball game.

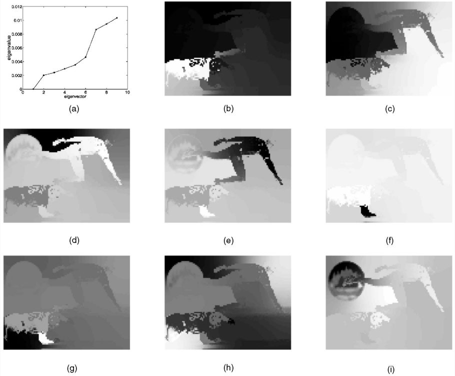


Fig. 3. Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

Normalized cut

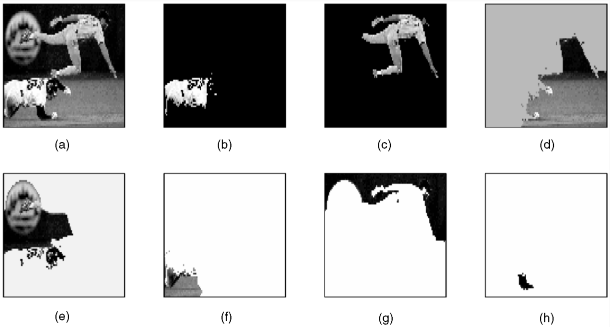


Fig. 4. (a) shows the original image of size 80×100. Image intensity is normalized to lie within 0 and 1. Subplots (b)-(h) show the components of the partition with $Ncut$ value less than 0.04. Parameter setting: $\sigma_I = 0.1$, $\sigma_X = 4.0$, $r = 5$.

Normalized cut

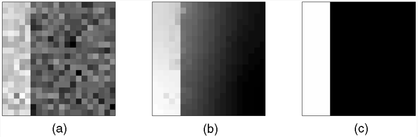


Fig. 6. (a) A synthetic image showing a noisy “step” image. Intensity varies from 0 to 1, and Gaussian noise with $\sigma = 0.2$ is added. Subplot (b) shows the eigenvector with the second smallest eigenvalue and subplot (c) shows the resulting partition.

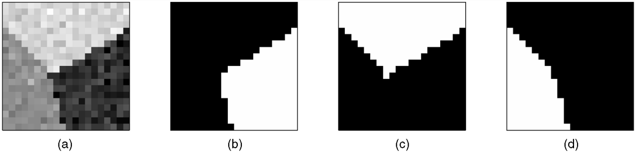


Fig. 7. (a) A synthetic image showing three image patches forming a junction. Image intensity varies from 0 to 1 and Gaussian noise with $\sigma = 0.1$ is added. (b)-(d) show the top three components of the partition.

Normalized cut

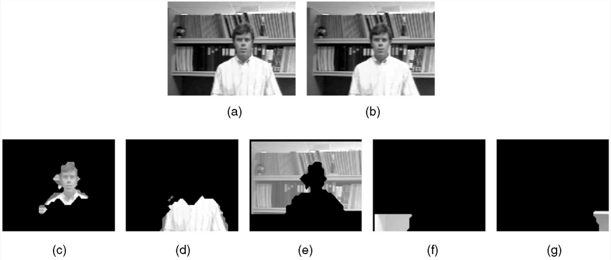


Fig. 11. Subimages (a) and (b) show two frames of an image sequence. Segmentation results on this two frame image sequence are shown in subimages (c) to (g). Segments in (c) and (d) correspond to the person in the foreground and segments in (e) to (g) correspond to the background. The reason that the head of the person is segmented away from the body is that, although they have similar motion, their motion profiles are different. The head region contains 2D textures and the motion profiles are more peaked, while, in the body region, the motion profiles are more spread out. Segment (e) is broken away from (f) and (g) for the same reason.

Characteristics of normalized cut

Pros

- Generic framework can be used with many different features and affinity formulations

Cons

- High storage requirement and time complexity
- Bias towards partitioning into equal segments
- Need the number of clusters as parameter

Appendix

Reference and further reading

- "Chap 5 | Segmentation" of R. Szeliski, Computer Vision: Algorithms and Applications
- "Chap 9 | Segmentation by Clustering" of Forsyth and Ponce, Computer Vision: A Modern Approach
- "Lecture12 | Clustering and Image Segmentation (Part II)" and "Lecture13 | Clustering and Image Segmentation (Part III)" of Bohyung Han, CSED441: Introduction to Computer Vision, POSTECH (2011)

Reference and further reading

- J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 888–905 (2000)

Normalized cuts and **image segmentation**

J Shi, J Malik - IEEE Transactions on pattern analysis and ..., 2000 - ieeexplore.ieee.org

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 ...