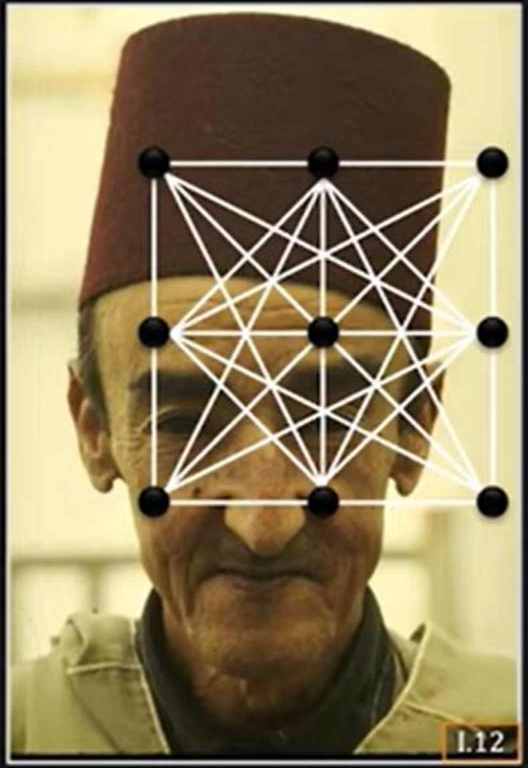
Graph Cut Segmentation

Graph Based Segmentation

Images as Graphs:

- A vertex for each pixel.
- An edge between each pair of pixels.
- Graph Notation: $G = (V, E)$ where V and E are the sets of vertices and edges, respectively.
- Each edge is weighted by the affinity or similarity between its two vertices.



Input Image

Measuring Affinity

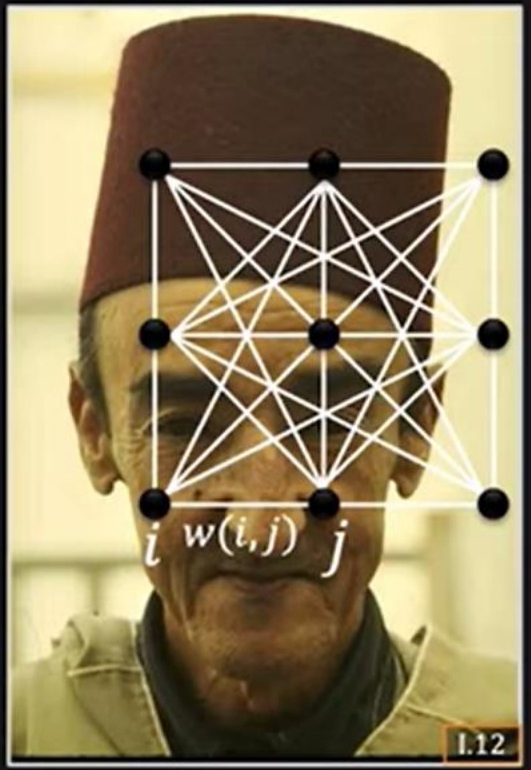
Let i and j be two pixels whose features are $\mathbf{f}_i$ and $\mathbf{f}_j$.

Pixel Dissimilarity:

$$S(\mathbf{f}_i, \mathbf{f}_j) = \sqrt{\left(\sum_k (f_{ik} - f_{jk})^2\right)}$$

Pixel Affinity:

$$w(i,j) = A(\mathbf{f}_i, \mathbf{f}_j) = e^{\left\{\frac{-1}{2\sigma^2} S(\mathbf{f}_i, \mathbf{f}_j)\right\}}$$



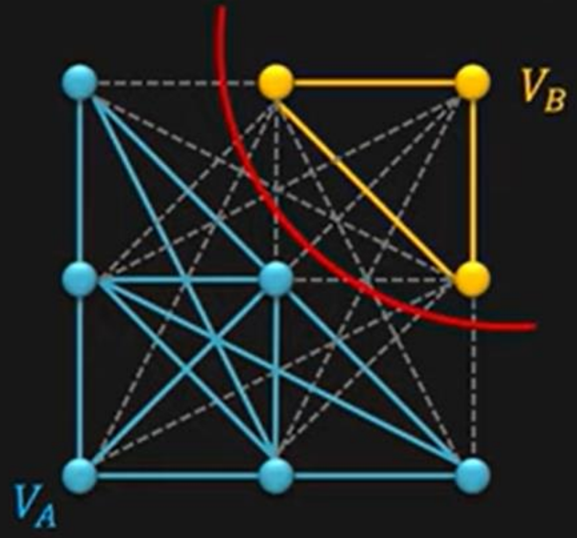
Input Image

Smaller the Dissimilarity, Larger the Affinity

with the edge, w i, j.

Graph Cut

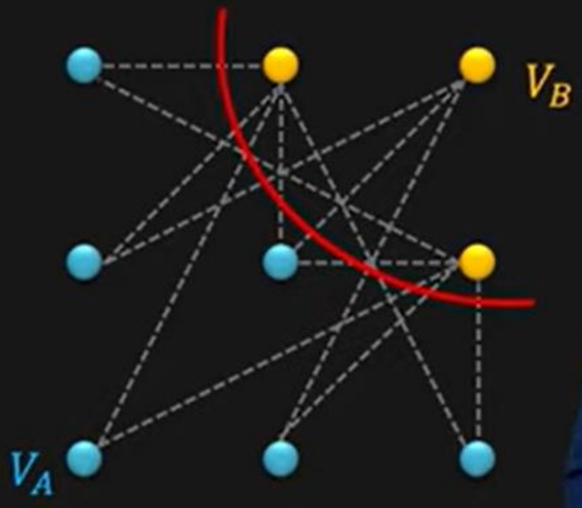
Cut $C = (V_A, V_B)$ is a partition of vertices V of a graph $G = (V, E)$ into two disjoint subsets V_A and V_B.



Cut-Set: Set of edges whose vertices are in different subsets of partition.

Cost of Cut: Sum of weights of cut-set edges.

$$\boxed{cut(V_A, V_B) = \sum_{u \in V_A, v \in V_B} w(u, v)}$$

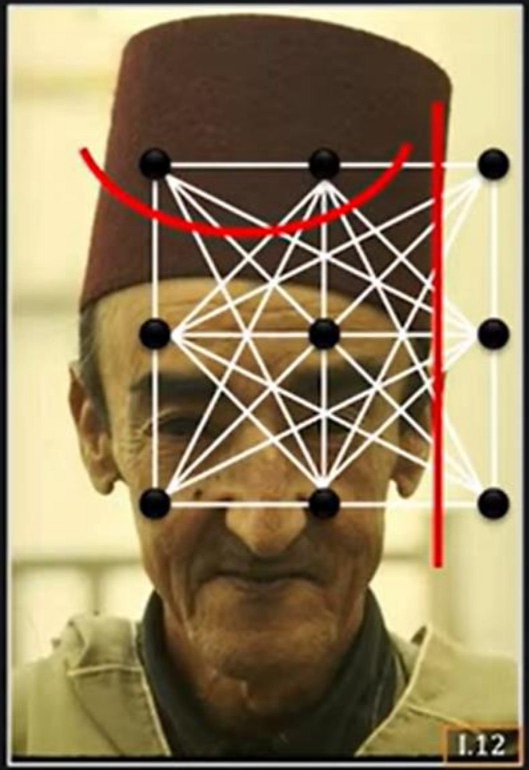


Graph Cut Segmentation

Criteria for Graph Cut:

- A pair of vertices (pixels) within a subgraph have high affinity.
- A pair of vertices from two different subgraphs have low affinity.

That is, minimize the cost of cut. Also called Min-Cut.

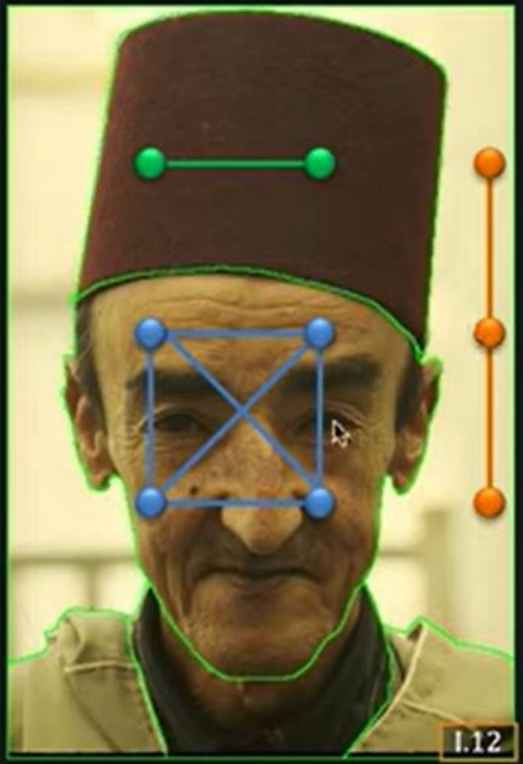


Input Image

Graph Cut Segmentation

Criteria for Graph Cut:

- A pair of vertices (pixels) within a subgraph have high affinity.
- A pair of vertices from two different subgraphs have low affinity.



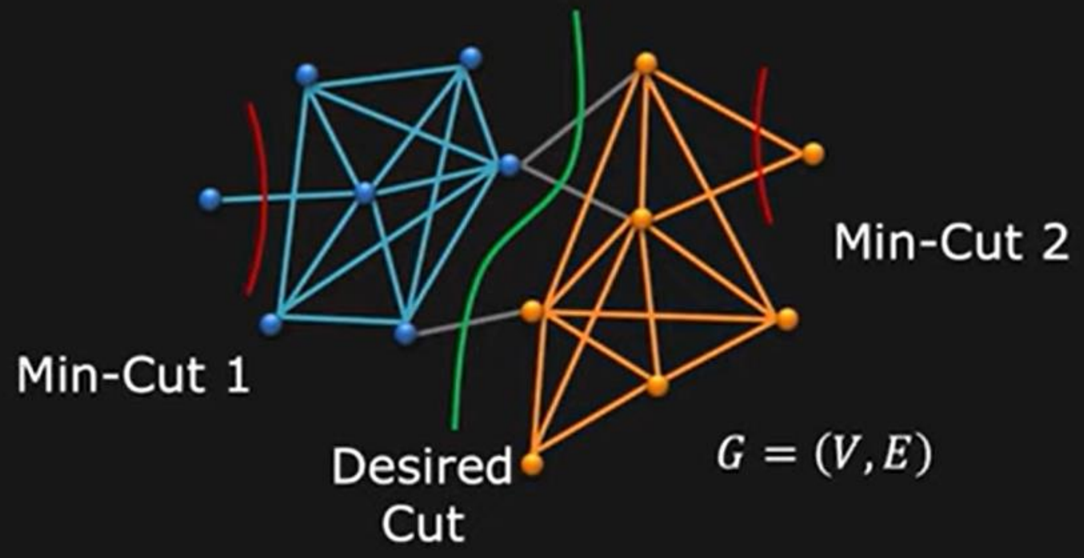
Input Image

That is, minimize the cost of cut.
Also called Min-Cut.

Each subgraph is an image segment.

Problem with Min-Cut

There is a bias to cut small, isolated segments.

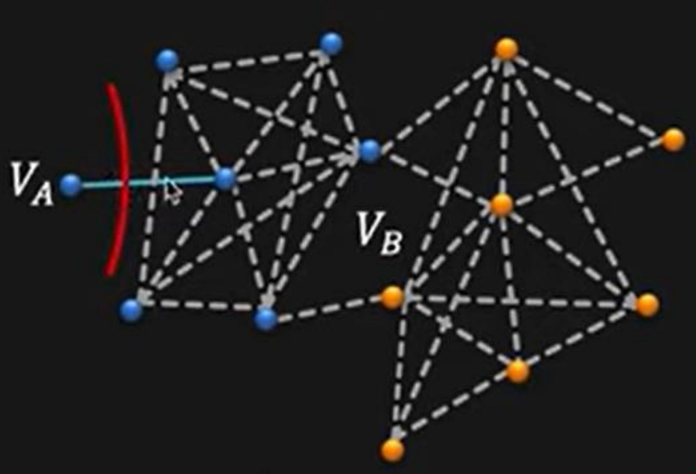


Solution: Normalize Cut to favor larger subgraphs.

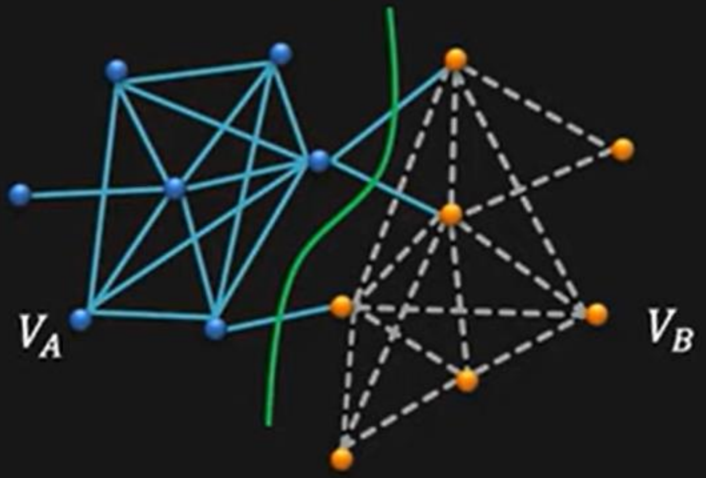
Measure of Subgraph Size

Compute how strongly vertices V_A are associated with vertices V.

$$\boxed{assoc(V_A, V) = \sum_{u \in V_A, v \in V} w(u, v)}$$



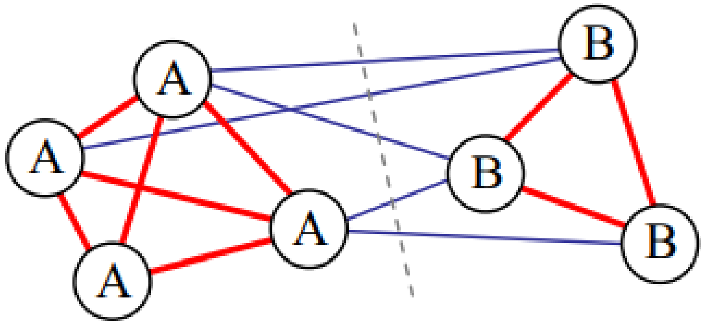
Weak $assoc(V_A, V)$



Strong $assoc(V_A, V)$

$assoc(\ )$ is the sum of the weights of the solid edges

[Shi 2000]



(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 $Ncut$.

where $assoc(A,A) = \sum_{i \in A, j \in A} w_{ij}$ is the *association* (sum of all the weights) within a cluster and $assoc(A,V) = assoc(A,A) + cut(A,B)$ is the sum of *all* the weights associated

Reference: Computer Vision: Algorithms and Applications (September 3, 2010 draft), Richard Szeliski

Normalized Cut (NCut)

Minimize Cost of Normalized Cut during Partition

$$\boxed{NCut(V_A, V_B) = \frac{cut(V_A, V_B)}{assoc(V_A, V)} + \frac{cut(V_A, V_B)}{assoc(V_B, V)}}$$

A better measure of segmentation is the normalized cut, which is defined as

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

Minimizing NCut has no known polynomial time solution.
It is NP-Complete.

Fast eigenvector-based approximations exist [Shi 2000].

NCut Segmentation Results

Segmented Images

Pixel Feature: $\{Brightness, Location\}$

Image Segmentation Summary

- Important for object detection and recognition.
- k-Means Segmentation
- Mean Shift Segmentation
- Normalized Graph Cut Segmentation
- Many variations of mean shift and graph based algorithms have been proposed. [Paris 2007] [Wang 2004] [Tolliver 2006] [Cour 2005] [Felzenszwalb 2004] [Boykov 2004]