Image segmentation

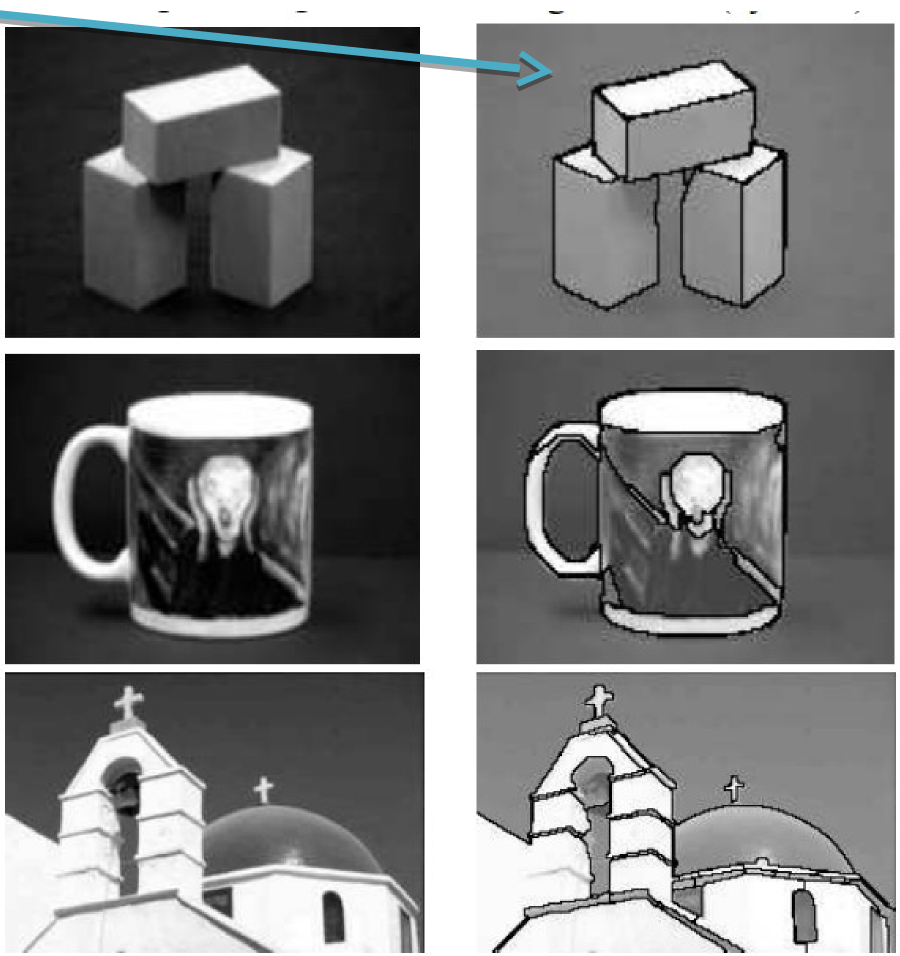
Francisco Gómez
MMS
U. Central and UJTL

Segmentation

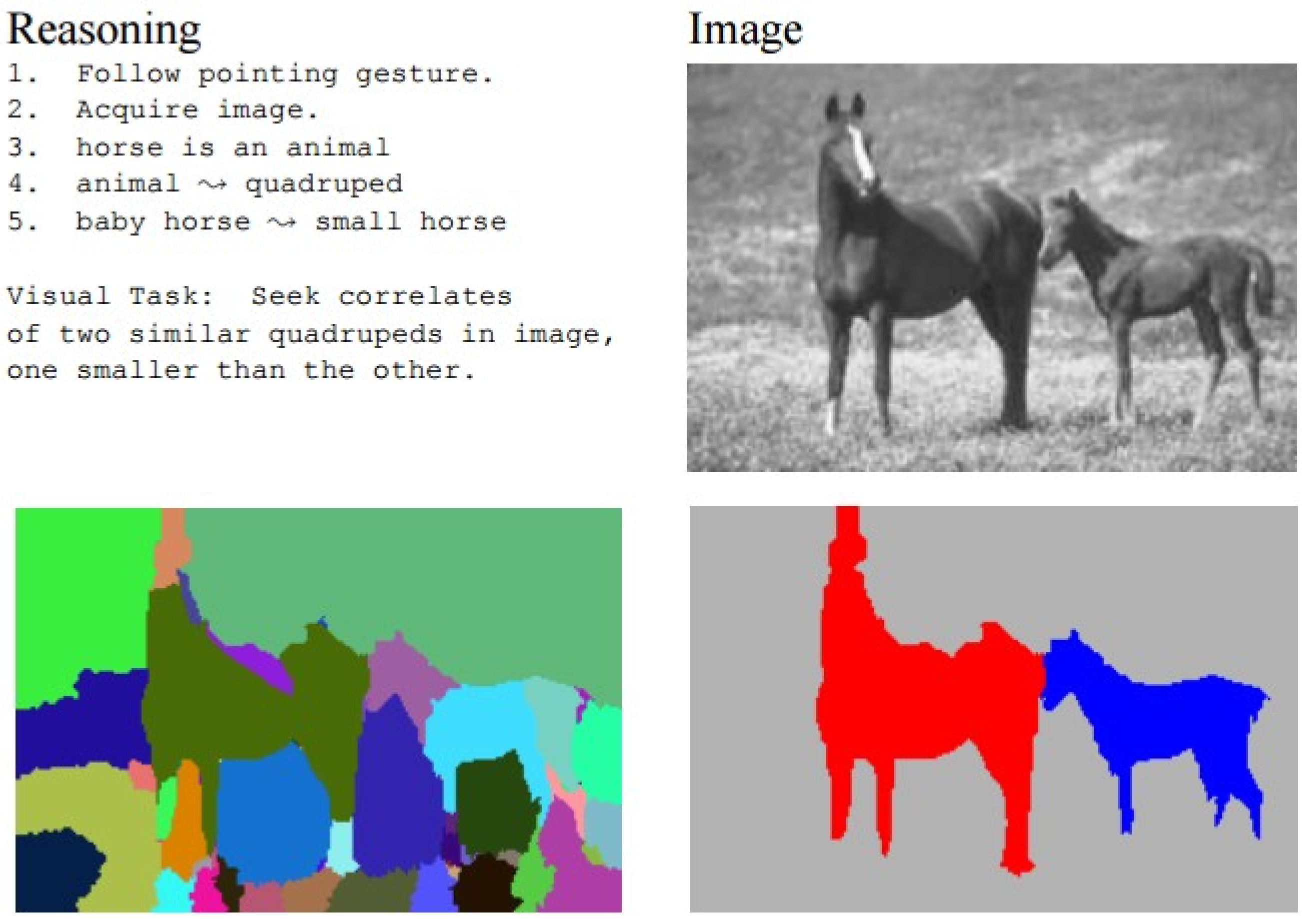
- Aim: cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects.

Example segmentations

Disjoint image regionss



- “Look, there is a baby horse with its mommy!”



Questions

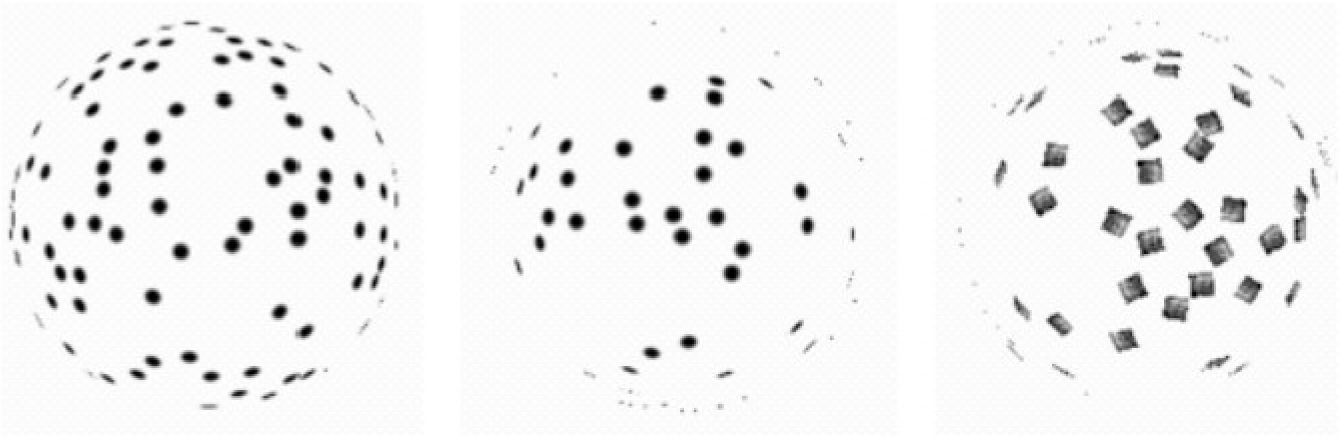
- How well can we expect to segment images without recognizing objects (i.e. bottom-up segmentation)?
- What determines a segment? How can we pose the problem mathematically?
- How do we solve the specified problem(s)?
- How can we evaluate the results?

Ideas

- Token whatever we need to group (pixels, points, surface elements)
- Bottom up segmentation
  - Tokens are together because they are locally coherent
- Top down segmentation
  - Tokens belong togheter because they lie on the same object

- Partitioning
  - Divide into regions with coherent internal properties
- Grouping
  - Identify sets of coherent tokens in the image

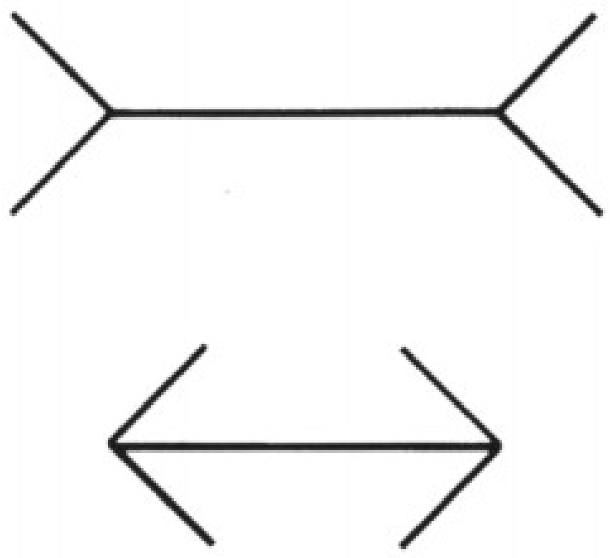
How you can group these tokens?



Grouping and Gestalt

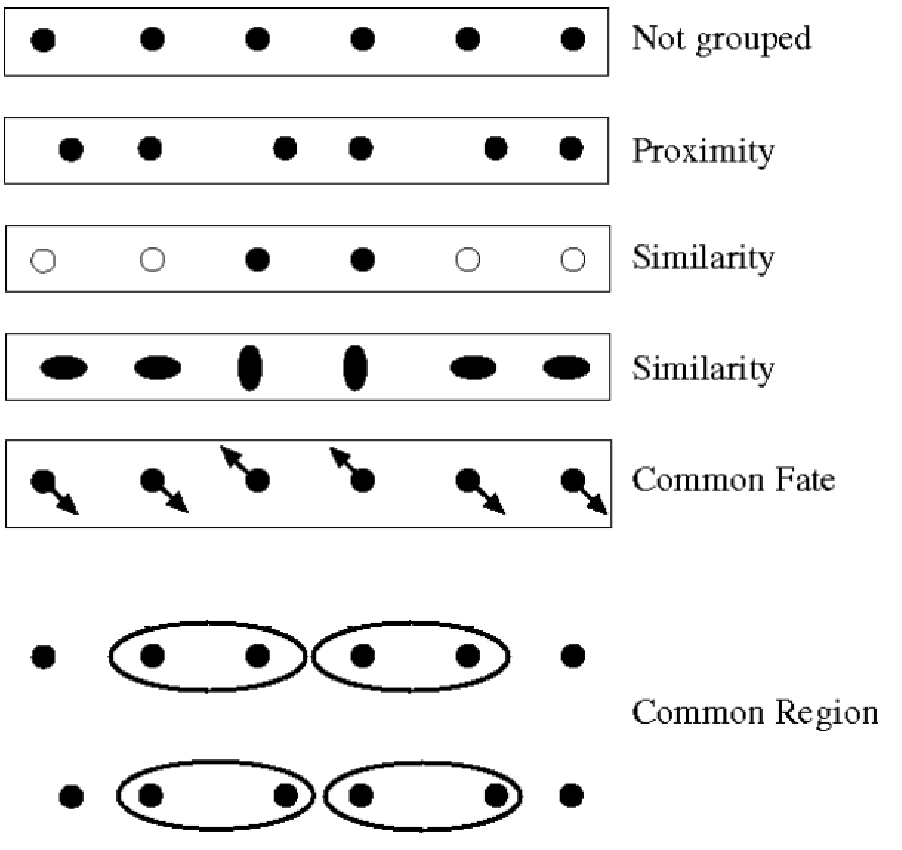
- Gestalt: German for form, whole, group
- Laws of Organization in Perceptual Forms (Gestalt school of psychology) Max Wertheimer 1912-1923

  "*there are contexts in which what is happening in the whole cannot be deduced from the characteristics of the separate pieces, but conversely; what happens to a part of the whole is, in clearcut cases, determined by the laws of the inner structure of its whole*"



Muller-Layer effect:
This effect arises from some property of the relationships that form the whole rather than from the properties of each separate segment.

Gestalt Laws



Emergence

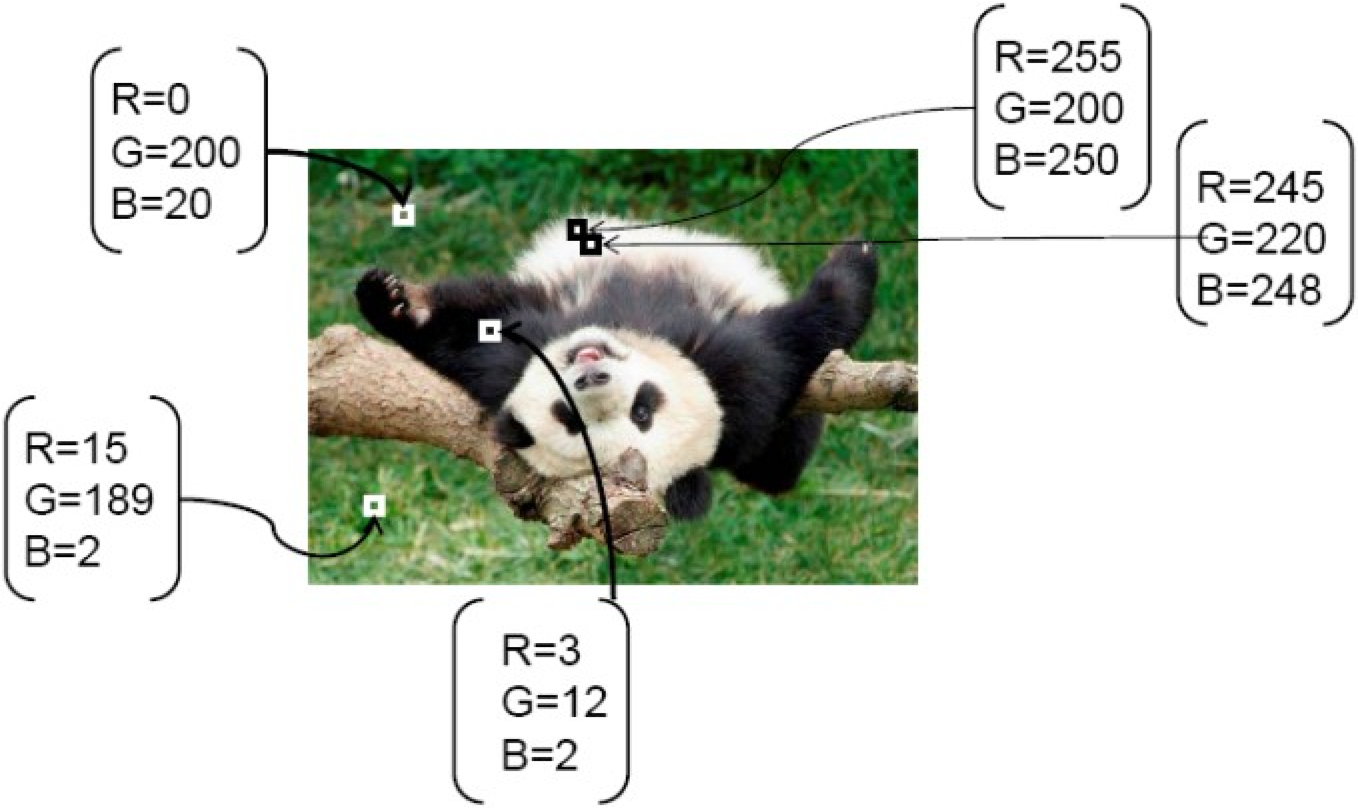
Segmentation as clustering

- Cluster together tokens that share similar visual characteristics
  - k-means
  - Mean shift
  - Graph cut

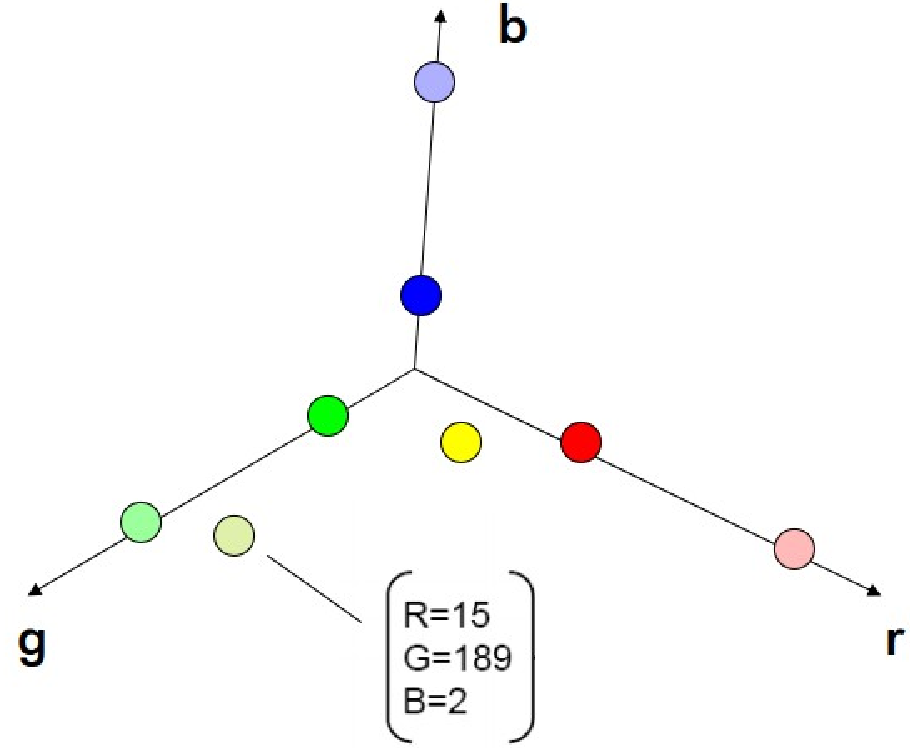
Feature space

- Tokens are identified by visual characteristics:
  - Position
  - Color
  - Texture
  - Motion vector
  - Size
  - Orientation

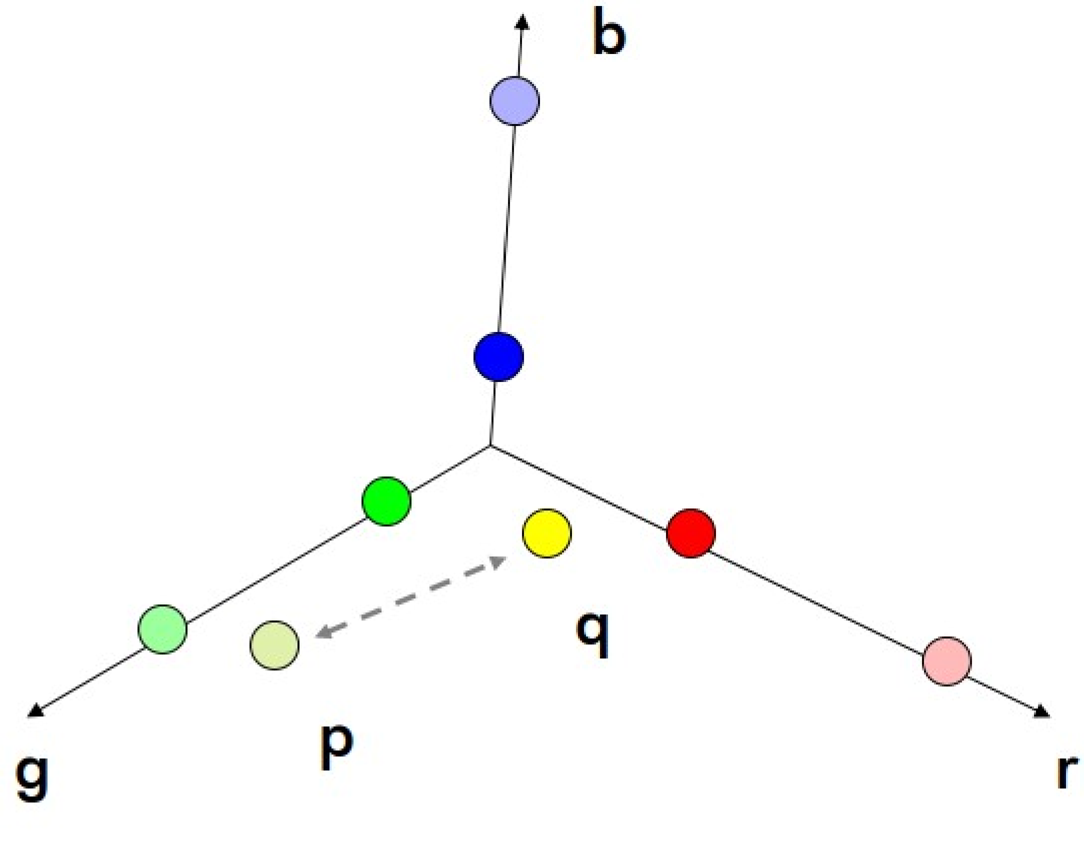
Feature space



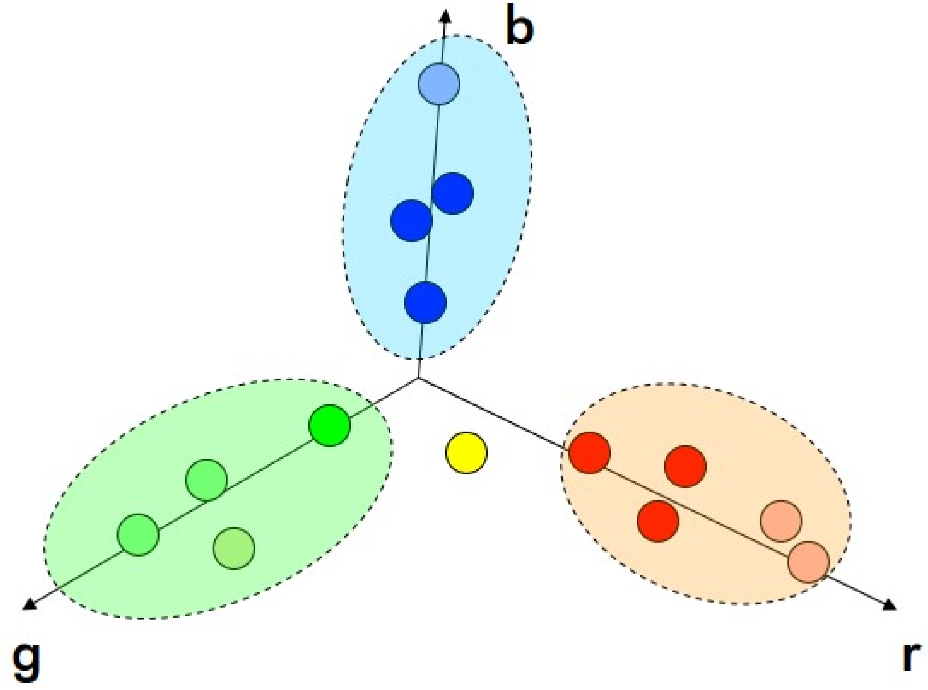
Tokens are points in the feature space



Distance between tokens



High similarity



K-means clustering

- Initialization: Given K categories, N points in feature space. Pick K points randomly; these are initial cluster centers (means) m1, ..., mK. Repeat the:
  1. Assign each of the N points, xj , to clusters by nearest mi
  2. Re-compute mean mi of each cluster from its member points
  3. If no mean has changed more than some ε, stop

$$e(\mathbf{m}_i) = \sum_{i=1}^{n_c} \sum_{j;c_j=i} |\mathbf{x}_j - \mathbf{m}_i|^2$$

$$\frac{\partial e}{\partial \mathbf{m}_k} = \sum_{j;c_j=k} -2(\mathbf{x}_j - \mathbf{m}_k) = 0$$

$$\mathbf{m}_k = \frac{\sum_{j;c_j=k} \mathbf{x}_j}{\sum_{j;c_j=k} 1} = \frac{1}{n_k} \sum_{j;c_j=k} \mathbf{x}_j$$

K-means intensity

example

```
he = imread('hestain.png');
imshow(he),
cform = makecform('srgb2lab');
lab_he = applycform(he,cform)
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
nColors = 3; % repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean', 'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
imshow(pixel_labels,[]),
title('image labeled by cluster index');
```

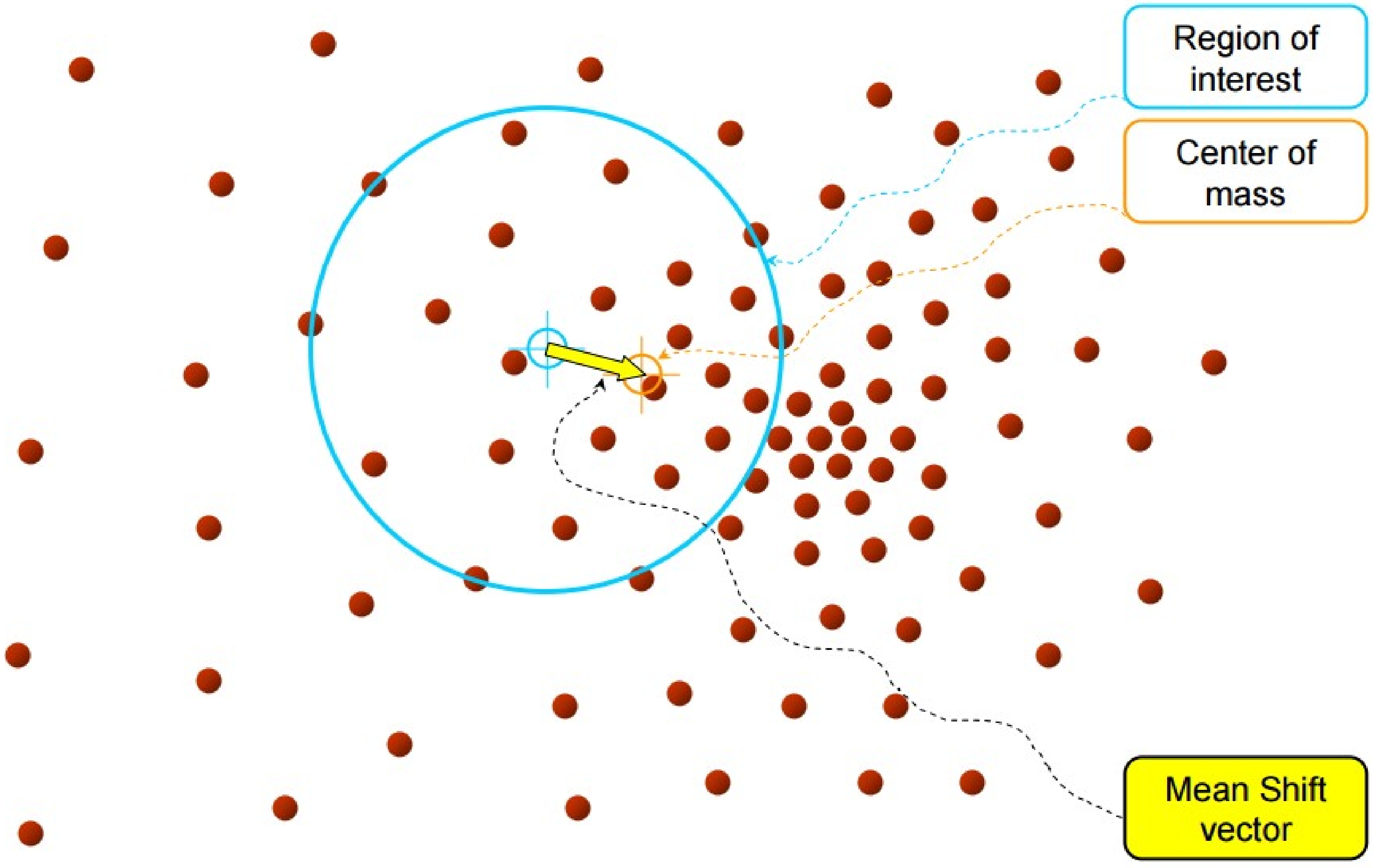
Pros and cons

- Pros
  - Simple and fast
  - Converge to a local minimum

- Cons
  - Need to pick K
  - Sensitive to initializations
  - Sensitive to outliers

Mean shift

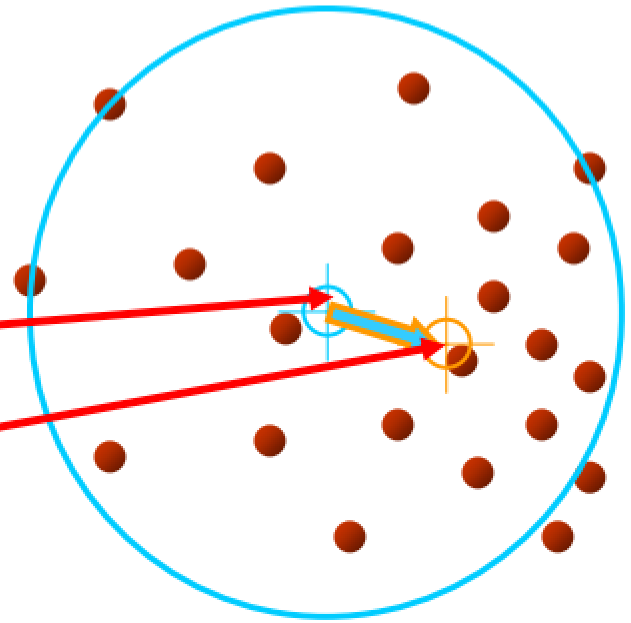
- The mean shift algorithm seeks a mode or local maximum of density of a given distribution
  - Choose a search window (width and location)
  - Compute the mean of the data in the search window
  - Center the search window at the new mean location
  - Repeat until convergence



Slide by Y. Ukrainitz & B. Sare

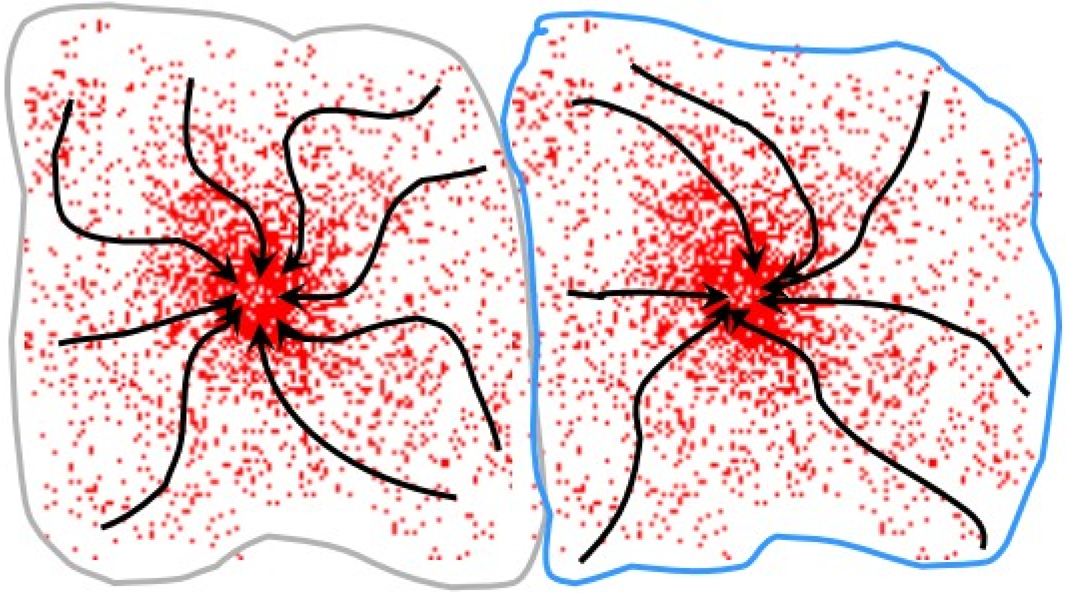
Computing the mean shift

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{h} \right)}{\sum_{i=1}^{n} g\left(\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{h} \right)} - \mathbf{x} \right]$$

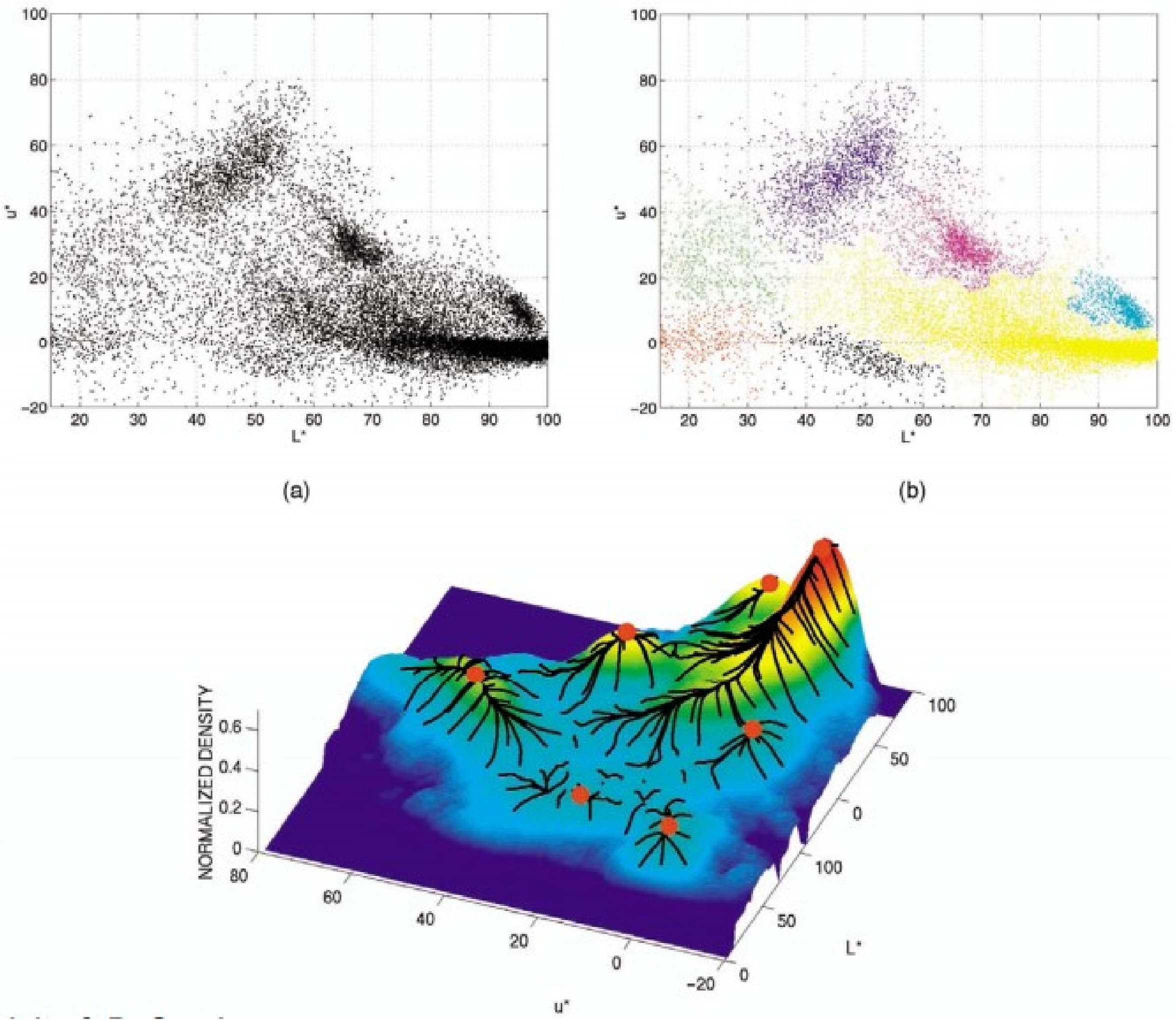


Attraction basin

- Cluster: all data points in the attraction basin of a mode



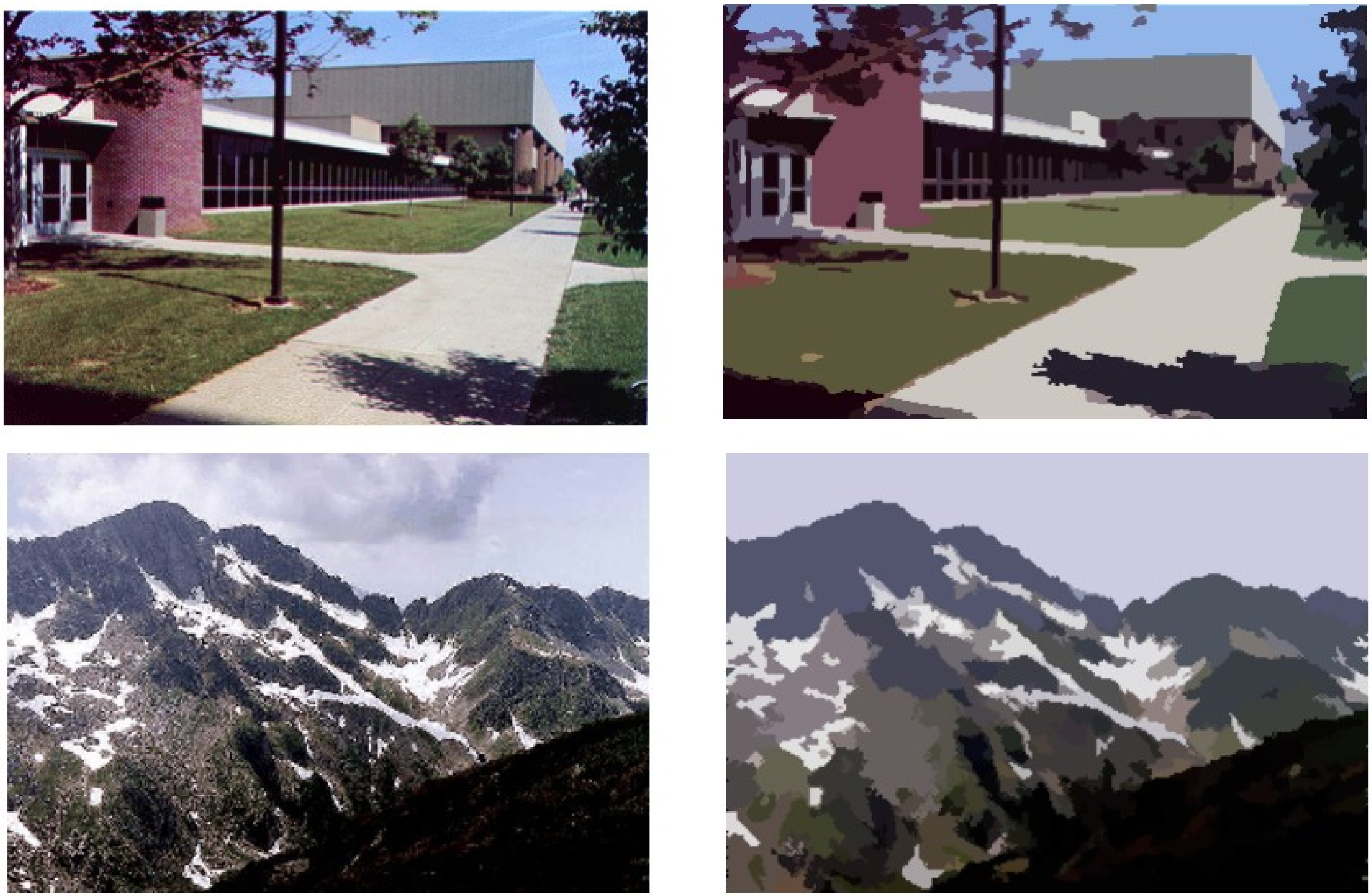
Attraction basin



Example

http://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering

Mean shift segmentation



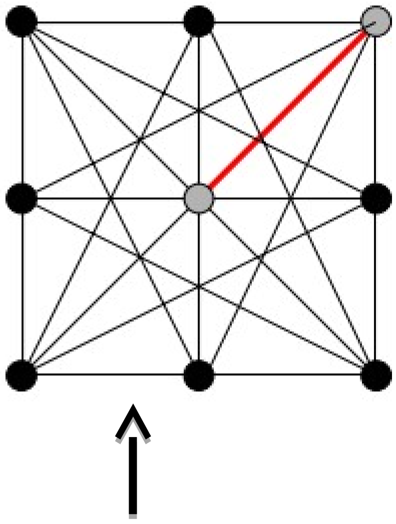
- Pros
  - Does not assume spherical clusters
  - Just a single parameter
  - Finds variable number of modes
- Cons
  - Output dependent of the window size
  - Computationally expensive
  - Problems with higher dimensions

Graph based segmentation

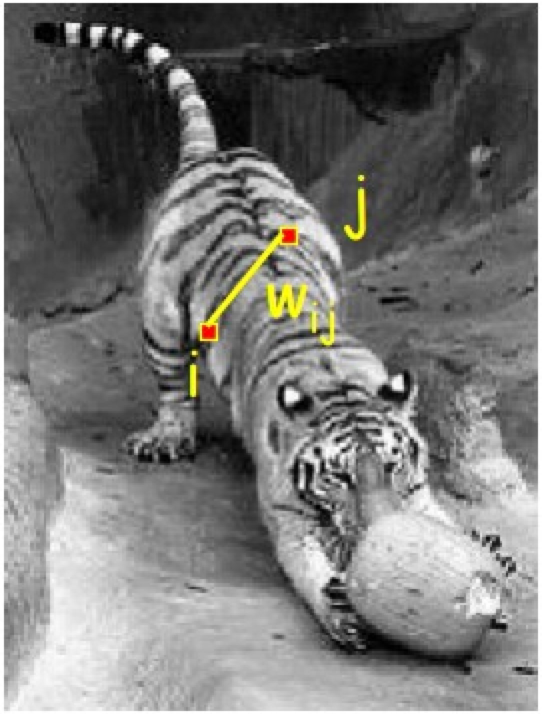
- Represent features and relationships using a graph

- Cut the graph and get the subgraphs with strong interior links and weak exterior links

Graph construction



Affinity



Affinity

Distance

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_d^2}\right)\left(\|x - y\|^2\right)\right\}$$

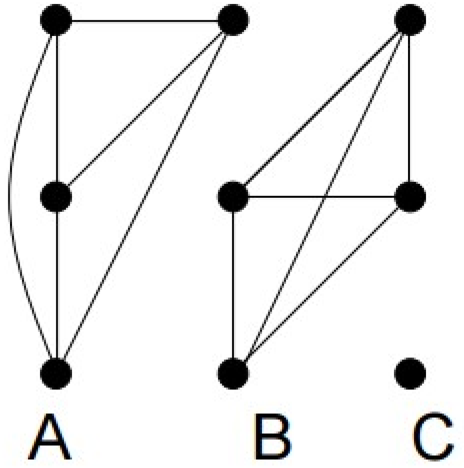
Intensity

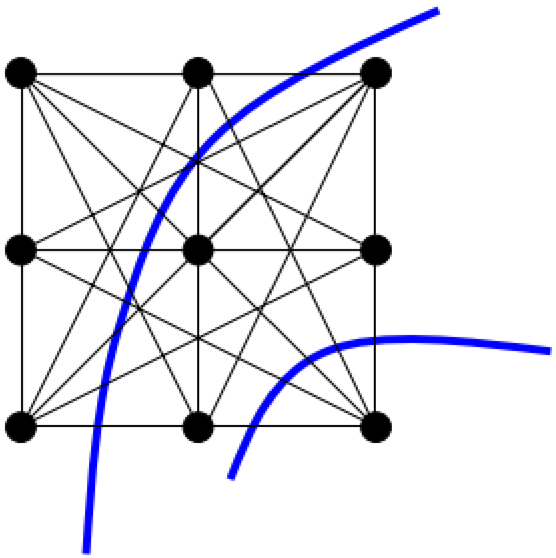
$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_i^2}\right)\left(\|I(x) - I(y)\|^2\right)\right\}$$

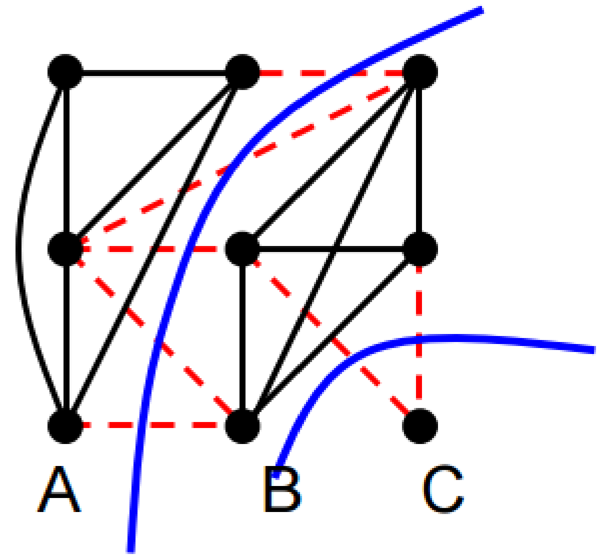
Color

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_t^2}\right)\left(\|c(x) - c(y)\|^2\right)\right\}$$

Segmentation by graph partitioning

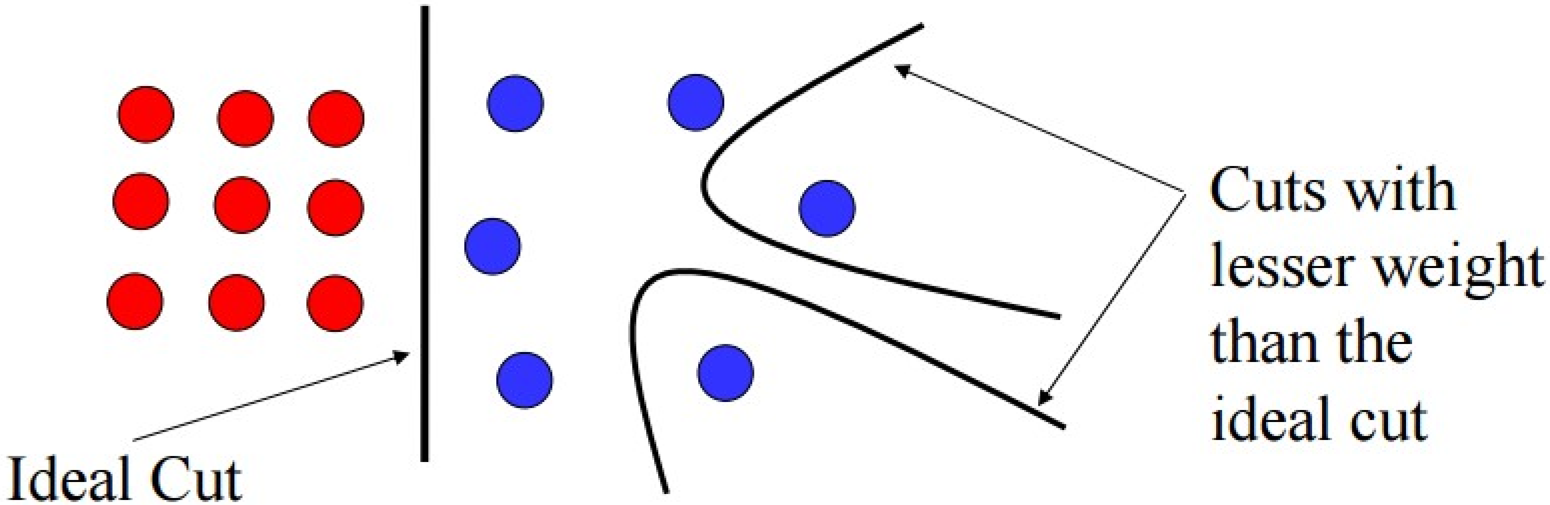






- CUT: Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges

A graph CUT is a segmentation



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

Normalized cut

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

assoc(*A*, *V*) = sum of weights of all edges in V that touch A

- The exact solution is NP-hard but an approximation can be computed by solving a generalized eigenvalue problem

- 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

Example