Lecture : Segmentation

Dr. V Masilamani
masila@iiitdm.ac.in

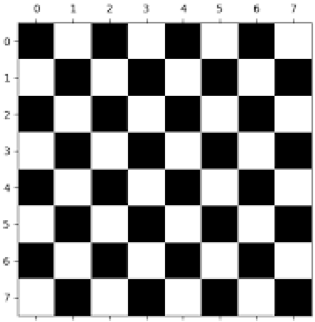
Department of Computer Science and Engineering
IIITDM Kancheepuram
Chennai-127

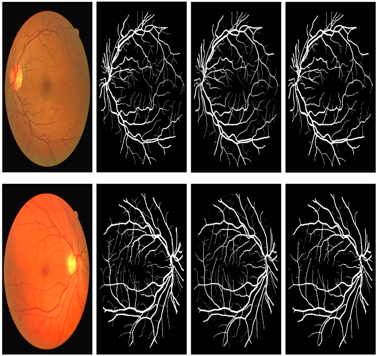
Overview

1. What is segmentation
2. A simple algorithm to detect characters in number plate
3. Network Flow
   - Ford-Fulkerson algorithm to find Max Flow
   - Max-Flow Min-Cut Theorem
   - Image Reconstruction using Max-Flow
   - Image Segmentation using Min-Cut
4. Mean shift segmentation algorithm
5. Segmentation using K-Means Clustering

Definition of segmentation

- Let I be the image, R be the set of pixel locations in I
- The segmentation of image I is the partition of R into $(R_1, R_2, ..., R_k)$ with the following conditions
  - $R = \cup_{i=1}^{k} R_i$
  - $R_i \cap R_j = \phi$ for all $i \neq j$
  - For each i, R_i is connected
  - For each i, $Q(R_i)$ is true, where Q is some predicate on similarity
  - For each i, j, If $Q(R_i) =$ True, and $Q(R_j)$ is true, and also R_i and R_j are adjacent, then $Q(R_i U Q_j)$ is false

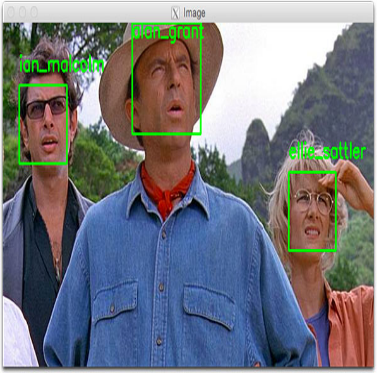


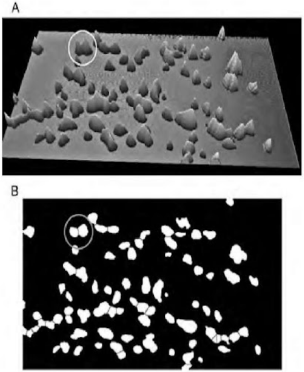


(a) (b) (c) (d)

- In Chessboard image, each square(black/white) is a segment
- For the retina, white pixels constitute a region, black pixels constitute another region

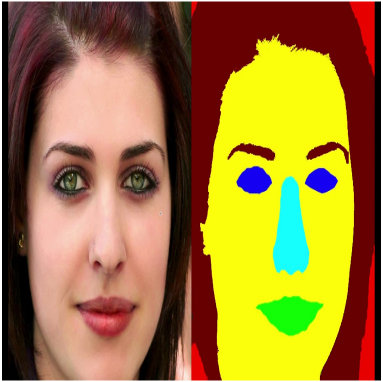
Examples of segmentation (cont.)

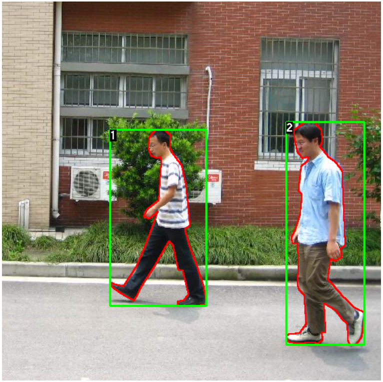




- Face detection is a segmentation problem
- Face recognition is a classification problem

Examples of segmentation (cont.)





The semantics of segments are problem dependent

- In the same input image, whole face can be a single segment or each part may be a single segment
- In the same input image, whole person may a single segment or face alone may be a single segment

A naive segmentation algorithm

Algorithm to segment the characters in the number plate

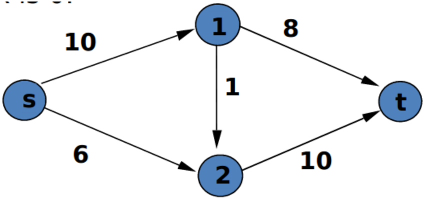
- Convert RGB to gray scale image
- Binarize the gray scale image using thresholding
- Find connected component(CC), and declare each CC as a segment

Network Flow

Definition of flow network:

Flow network is a directed weighted graph G=(V,E,c) such that

- Weight(capacity) $c(u, v) \geq 0$.
- Two distinguished vertices exist in G namely :
  - Source (denoted by s) : In-degree of this vertex is 0.
  - Sink (denoted by t) : Out-degree of this vertex is 0.



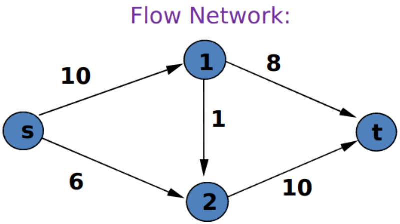
Definition of flow

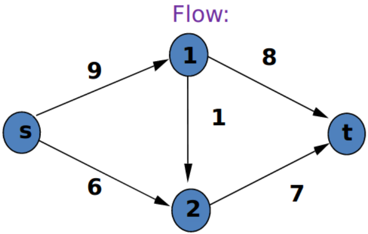
Flow in a network is an integer-valued function f defined on the edges of G satisfying:

- $0 \leq f(u, v)$, for every edge (u,v) in E.
- Capacity constraint: $\forall u, v \in V, f(u, v) \leq c(u, v)$
- Skew symmetry: $\forall u, v \in V, f(u, v) = -f(u, v)$
- Flow conservation: For each vertex v other than s ant t, inflow(v)=outflow(v)

Skew symmetry condition implies that f(u,u)=0

Network Flow (cont.)





Definition of Max Flow

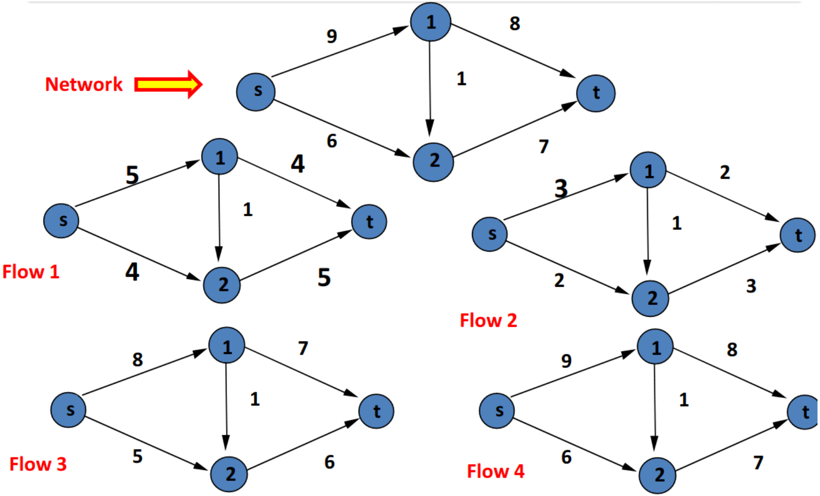
- The value of a flow is given by :

$$|f| = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$$

- Definition of max flow:
  - Given a graph G=(V, E) with capacities on edges, find flow f, such that $|f|$ is maximum.

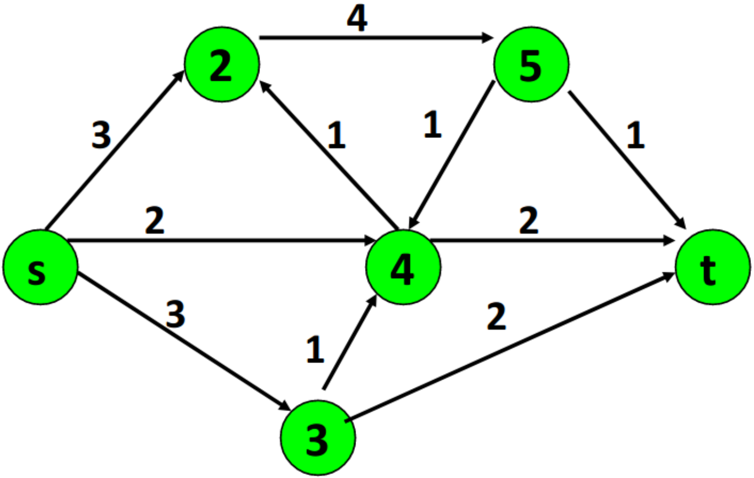
Network Flow (cont.)

Example of a Max Flow

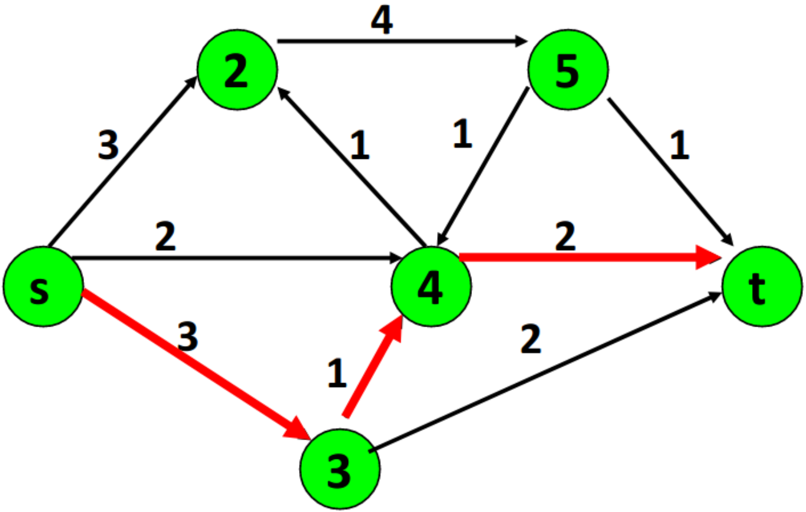


Ford-Fulkerson algorithm to find Max Flow

Flow network

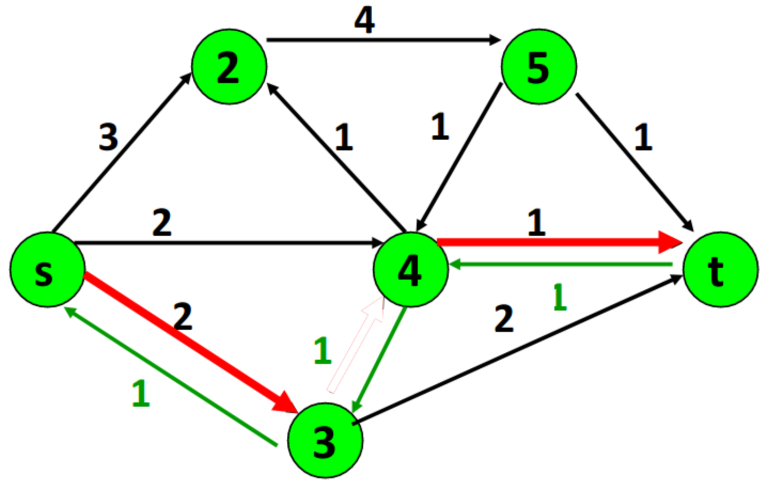


Find any s-t path in G(x)

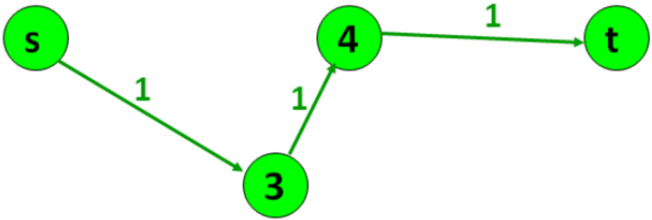


Ford-Fulkerson algorithm to find Max Flow (cont.)

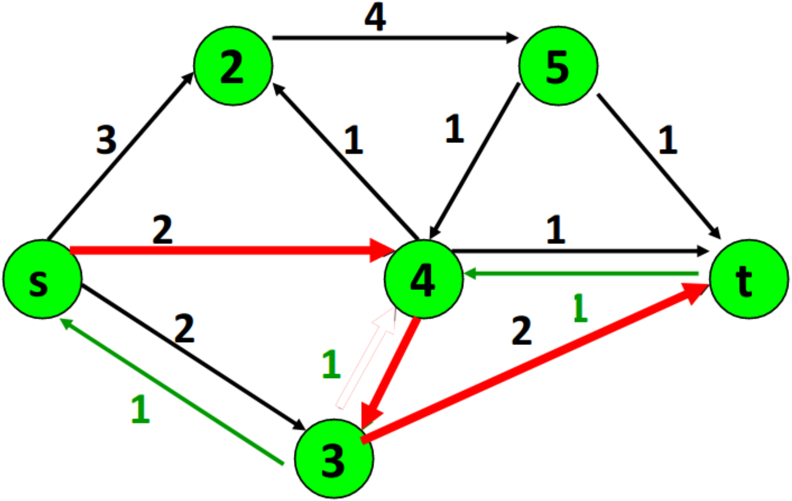
Residual graph:

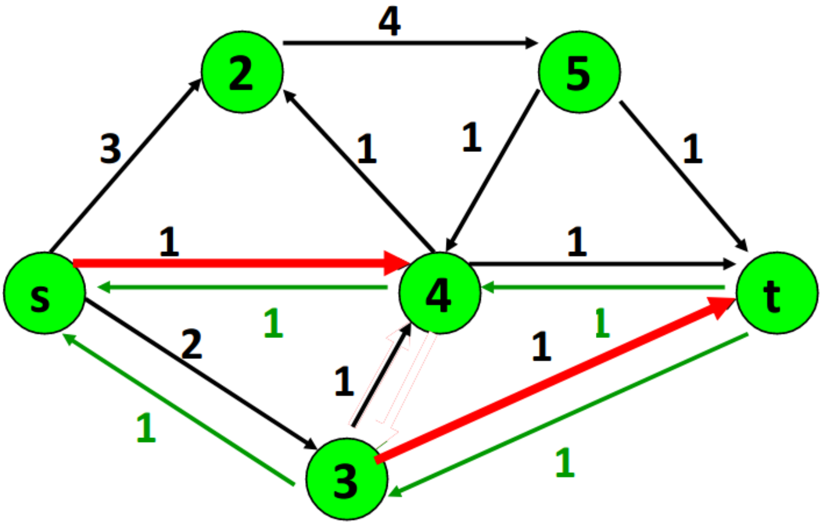


Flow:



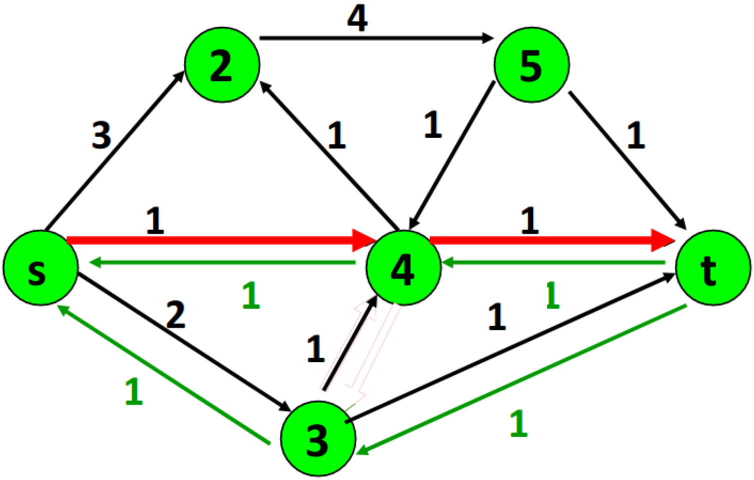
Ford-Fulkerson algorithm to find Max Flow (cont.)



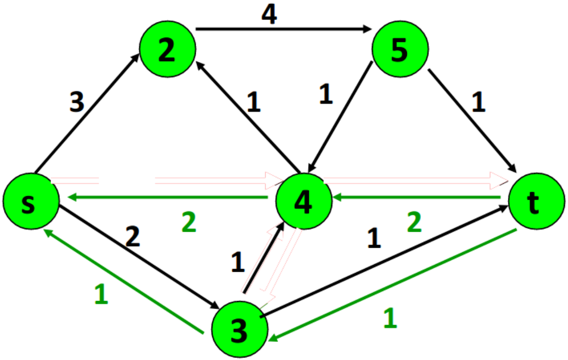

4
2
5
3
1
1
1
s
1
4
1
t
1
2
1
1
1
1
3
1

Ford-Fulkerson algorithm to find Max Flow (cont.)

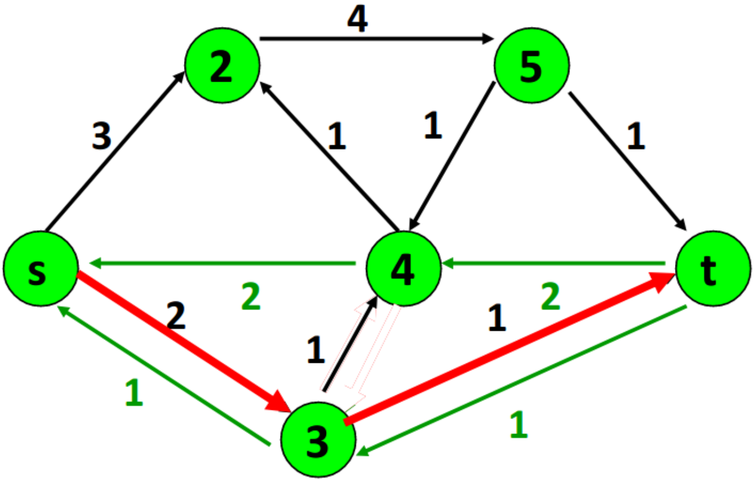
Find any s-t path



- Determine the capacity Δ of the path
- Send Δ units of flow in the path
- Update residual capacities

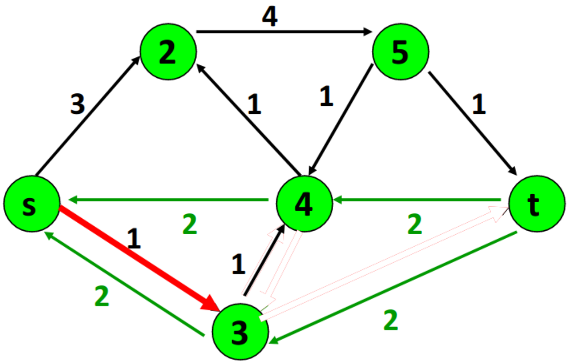


Find any s-t path

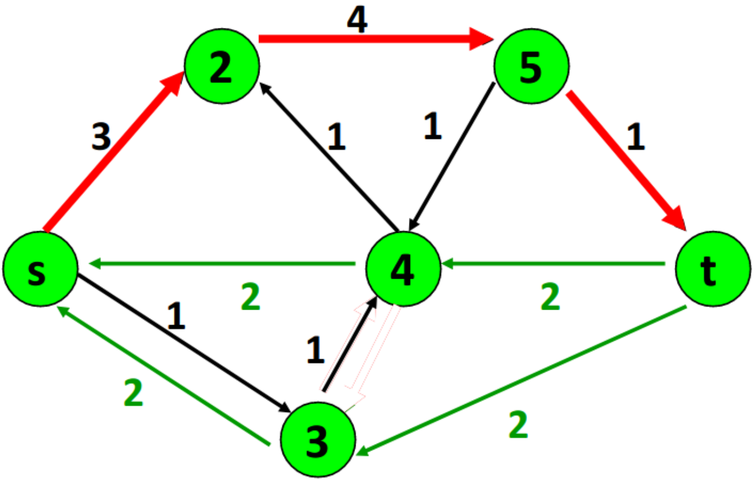


Ford-Fulkerson algorithm to find Max Flow (cont.)

- Determine the capacity Δ of the path
- Send Δ units of flow in the path
- Update residual capacities

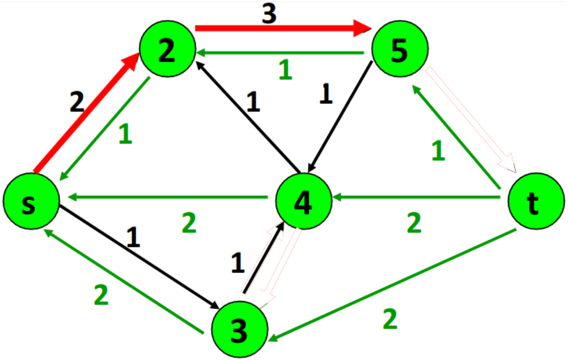


Find any s-t path

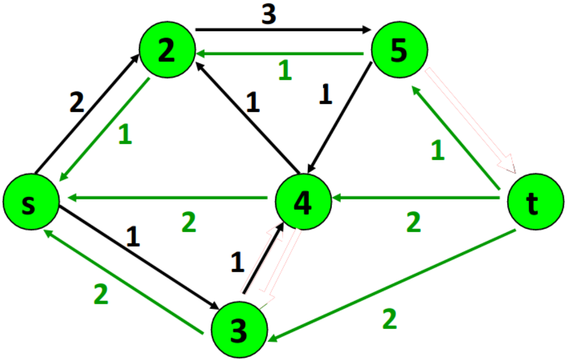


Ford-Fulkerson algorithm to find Max Flow (cont.)

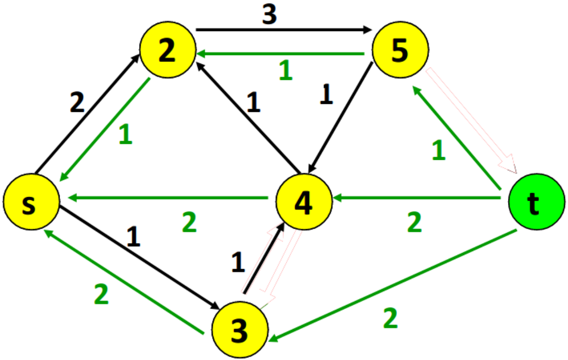
- Determine the capacity Δ of the path
- Send Δ units of flow in the path
- Update residual capacities



- There is no s-t path in the residual network.

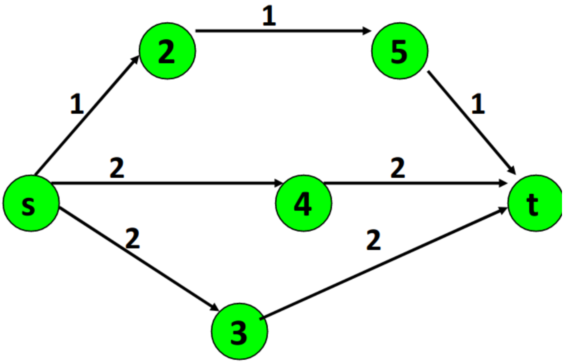


- These are the nodes that are reachable from node s.



Ford-Fulkerson algorithm to find Max Flow (cont.)

- Here is the optimal flow



Steps of Ford Fulkerson Algorithm

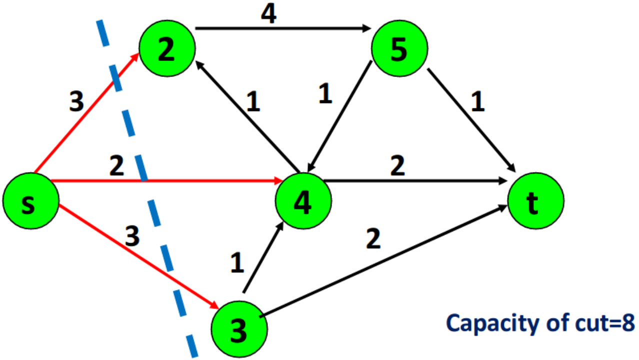
Input G=(V,E,C),s,t

Assume $G = (V, V \times V, C)$, where C(u,v)=0 if (u, v) is not in E

- Find s-t path in G
- Mind min weight in the s-t path, say a
- Save the flow with the s-t path with weight a
- Find residual graph $G' = (V, V, C')$
  where
  - $C'(u, v) = C(u, v) - a$ when (u,v) is in the path
  - $C'(v, u) = C(v, u) + a$ when (u,v) is in the path
  - $C'(u, v) = C(u, v)$ when (u,v) is not in the path
- $G = G'$
- Repeat the above five steps until there is no s-t path
- output the accumulated flow

Definition of s-t Cut

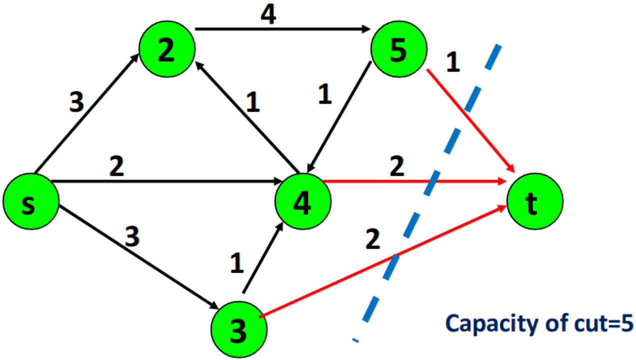
- Set of edges is said to be a cut if they are removed from the graph, graph should be disconnected into two components, one with source and the other one with sink.



Max-Flow and Min-cut Theorem

Max-Flow Min-Cut Theorem (cont.)

- The maximum flow value is the same as the minimum cut value


4
2
5
1
3
1
1
2
2
s
4
t
3
2
1
3
Capacity of cut=5

Max-Flow Min-Cut Theorem (cont.)

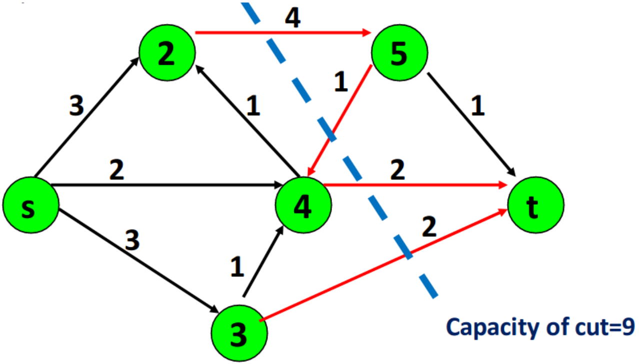
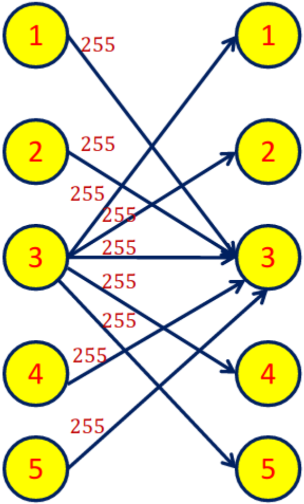


Image Reconstruction using Max-Flow

Image Reconstruction using Max-Flow (cont.)

	1	2	3	4	5
1	0	0	255	0	0
2	0	0	255	0	0
3	255	255	255	255	255
4	0	0	255	0	0
5	0	0	255	0	0

Image



Graph representation

Image Segmentation using Min-Cut

Image Segmentation

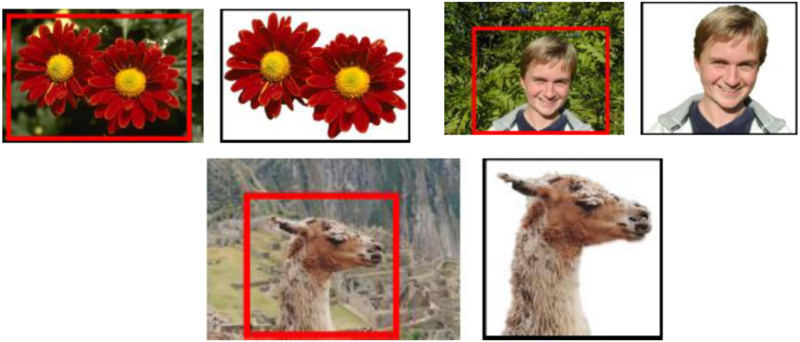
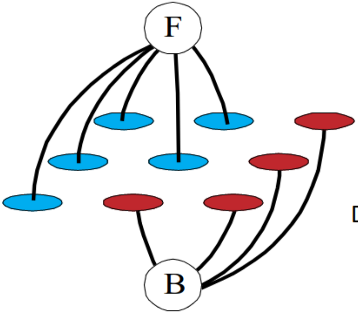


Image Segmentation

Image Segmentation using Min-Cut (cont.)

- Each pixel = node
- Add two nodes F & B
- Labeling: link each pixel to either F or B

F	F	B
F	F	B
F	B	B



Desired result

Image Segmentation using Min-Cut (cont.)

- Construct graph with data term
  - Put one edge between each pixel and F
  - Put one edge between each pixel and B
  - Weight of edge between i and F: $w_{iF} = -\lambda log(P_B(i))$
  - Weight of edge between i and B: $w_{iB} = -\lambda log(P_F(i))$

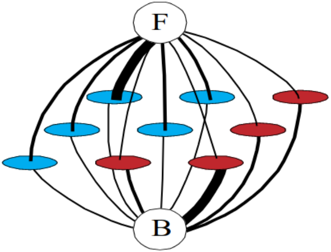


Image Segmentation using Min-Cut (cont.)

- Add Smoothness term to the Graph
  - Add an edge between each neighbor pair (i,j)
  - Weight of edge between i and j: $w_{ij} = exp(-(I_i - I_j)^2/2\sigma^2)$

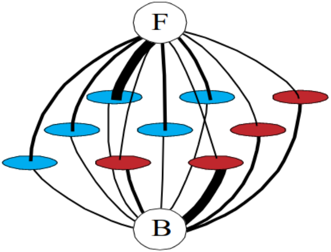


Image Segmentation using Min-Cut (cont.)

- Min-Cut
  - Cut: Remove edges to disconnect F from B
  - Min Cut: Cut with sum of its edge weights is minimum

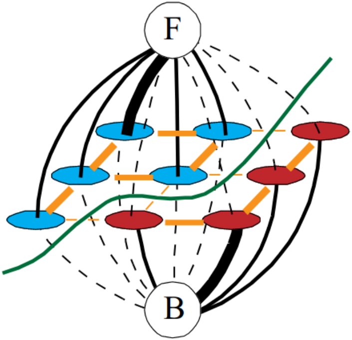


Image Segmentation using Min-Cut (cont.)

- In order to be a cut:
  - For each pixel, either the F or G edge has to be cut
- In order to be minimal
  - Only one edge to F or B per pixel can be cut

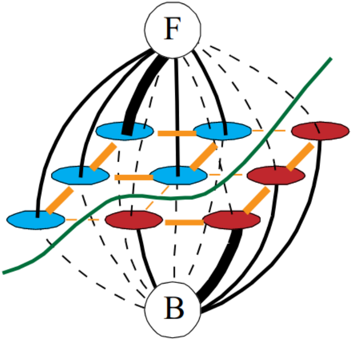


Image Segmentation using Min-Cut (cont.)

- Which edges are to be removed?
  - Edges with least cost

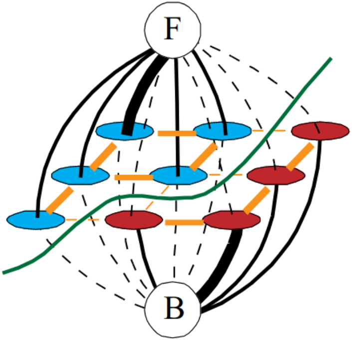


Image Segmentation using Min-Cut (cont.)

- Since $w_{iF} = -\lambda log(P_B(i))$, larger value of $P_B(i)$ will lead to smaller value of w_{iF}
  - Hence edge (i,F) will be removed, and edge (i,B) will stay
  - ie. Edge (i,B) will stay after cut if prob of i to be background is high

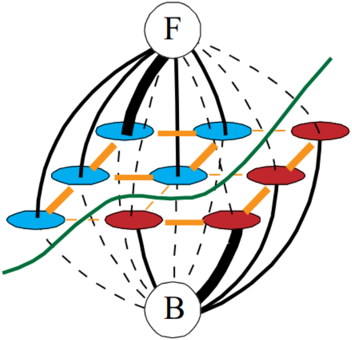


Image Segmentation using Min-Cut (cont.)

- Since $w_{iB} = -\lambda log(P_F(i))$, larger value of $P_F(i)$ will lead to smaller value of w_{iB}
  - Hence edge (i,B) will be removed, and edge (i,F) will stay
  - ie. Edge (i,F) will stay after cut if prob of i to be background is high

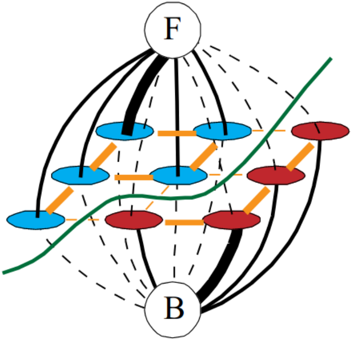
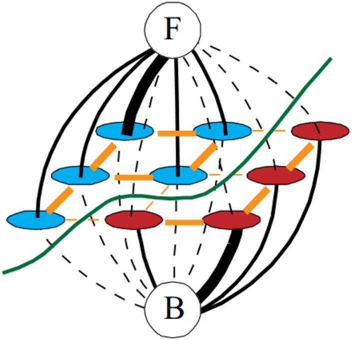


Image Segmentation using Min-Cut (cont.)

- Since for neighboring pixels (i,j), $w_{ij} = exp(-(I_i - I_j)^2/2\sigma^2)$, similar i and j, wij is very high
- Hence edge between similar pixels will stay after graph cut.
- This provides smoothness



- How to find min-cut
  - Apply max-flow algorithm
  - The output of max-flow algorithm will result in min-cut (max-flow min-cut theorem)

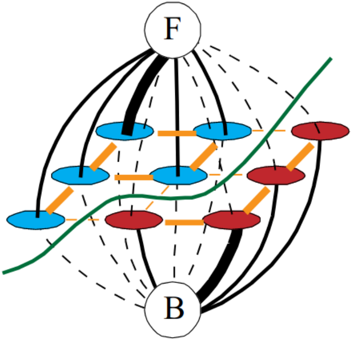


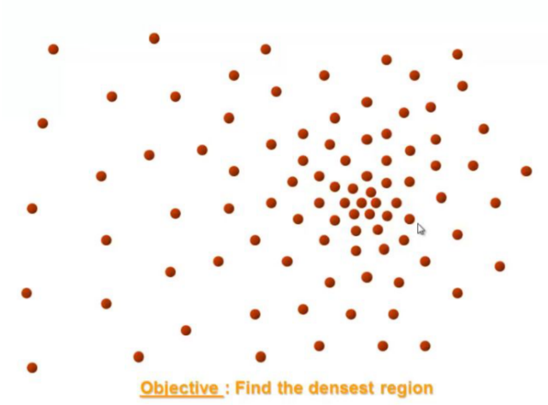
Image Segmentation using Min-Cut (cont.)

- How to find $P_B(i)$ and $_F(i)$ for pixel I?
  - User will give some seed of the background and foreground
  - From seed points of background compute $P_B(i)$
  - From seed points of foreground compute $P_F(i)$

Summary

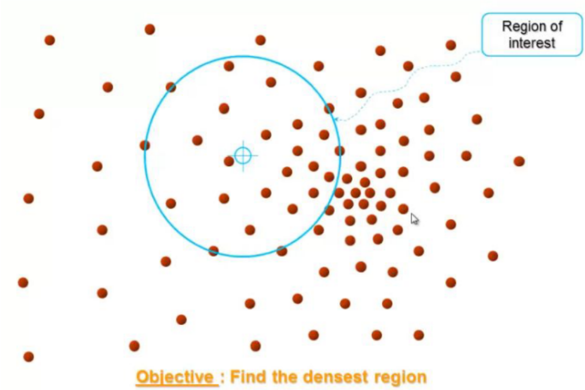
- Ford-Fulkerson algorithm to find max-flow is discussed
- Relationship between image and graph is discussed
- Graph-cut was computed using Ford-Fulkerson
- Couple of applications for graph-cut were illustrated
- As computation of graph-cut algorithm is polynomial, the applications discussed require only polynomial time.

Mean Shift: Given a distribution of points, and also a point x, mean shift is a procedure for finding the centroid of local densest region for x.



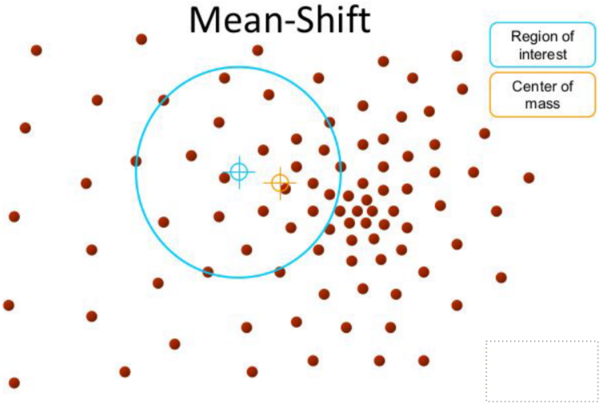
How to find centroid of local densest region for given x

Step 1: Find the centroid of points centered at x with radius r, say such centroid as c

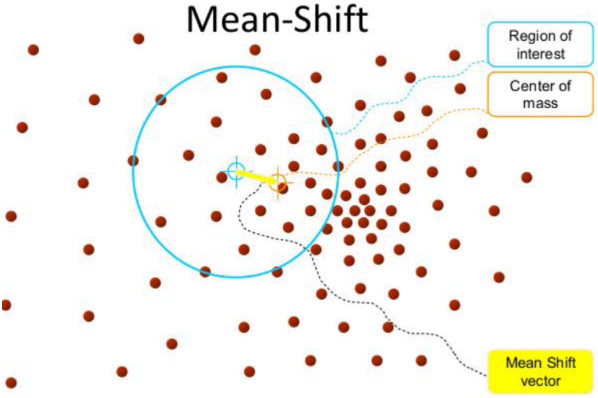


Mean shift segmentation algorithm (cont.)

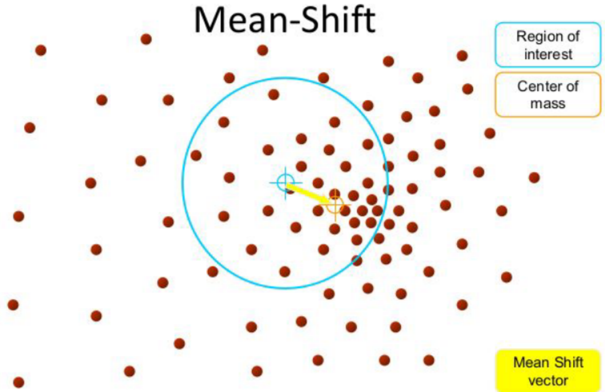
Step 2: Find the centroid of points centered at c with radius r, say such centroid as $c\hat{p}rime$



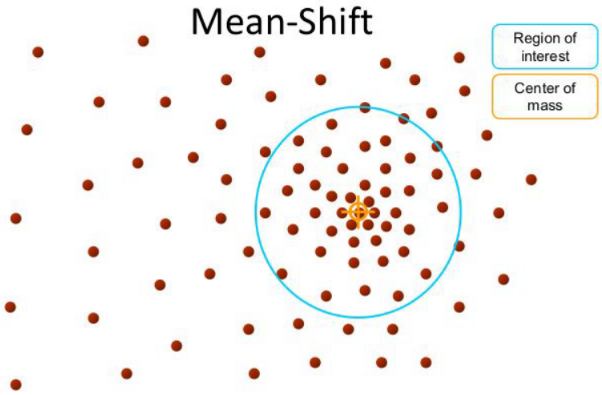
Step 3: $c = c'$, **and goto step 2 when** $|c - c'| > T$



Mean shift segmentation algorithm (cont.)



Mean shift segmentation algorithm (cont.)



Mean shift segmentation algorithm (cont.)

Mean shift procedure for labelling point x

Input: Set of points, and x

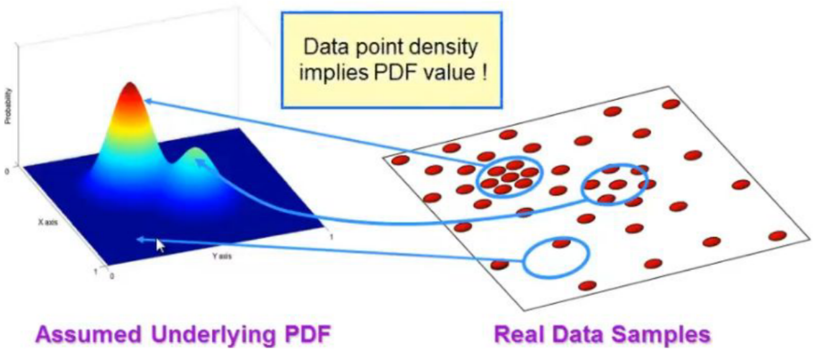
1. Find the centroid of points centered at x with radius r, say such centroid as c
2. Find the centroid of points centered at c with radius r, say such centroid as $c\hat{p}rime$
3. $c = c'$, and goto step 2 when $|c - c'| > T$
4. Assign a label l to the centroid c'
5. Assign the label l to x also

How to label pixel

- All the pixels with feature vector x will be labelled with the label of x

Mean Shift Image Segmentation

- For each point x, find the label using mean shift procedure
- All the pixels with feature vector x will be labelled with the label of x


Data point density implies PDF value !
Probability
X axis
Y axis
Assumed Underlying PDF
Real Data Samples

Mean shift segmentation algorithm (cont.)

Mean Shift Vector Computation

- Gaussian Kernel

$$K(x;h) = \frac{2\pi^{\frac{-d}{2}}}{h^d} exp(-\frac{1}{2}\frac{||x||^2}{h})$$

- let

$$f(x) = C\sum_{i=1}^{N} K(||\frac{x-x_i}{h}||^2)$$

- We want to find a maximum in f i.e gradient = 0

$$\nabla f(y) = 0$$

$$C\sum_i \nabla K(||\frac{x_i - y}{h}||^2) = 0$$

$$C\frac{2}{h}\sum_i [x_i - y][K'(||\frac{x_i - y}{h}||^2)] = 0$$

$$C\frac{2}{h}[\frac{\sum_i x_i K'(||\frac{x_i-y}{h}||^2)}{\sum_i k'(||\frac{x_i-y}{h}||^2)} - y] * [\sum_i k'(||\frac{x_i - y}{h}||^2)] = 0$$

Since $K'(||\frac{x_i - y}{h}||^2)$ is non zero,

$$[\frac{\sum_i x_i K'(||\frac{x_i - y}{h}||^2)}{\sum_i K'(||\frac{x_i - y}{h}||^2)} - y] = 0$$

$$y = \frac{\sum_i x_i K'(||\frac{x_i - y}{h}||^2)}{\sum_i K'(||\frac{x_i - y}{h}||^2)}$$

$$y^{(j+1)} = \frac{\sum_i x_i K'(||\frac{x_i - y^j}{h}||^2)}{\sum_i K'(||\frac{x_i - y^j}{h}||^2)}$$

here $[\frac{\sum_i x_i K'(||\frac{x_i - y}{h}||^2)}{\sum_i K'(||\frac{x_i - y}{h}||^2)} - y]$ is the mean shift vector

How to find mean shift with optimal weights?

Mean shift segmentation algorithm (cont.)

Examples

In[1]:= MeanShiftFilter[ , 3, .1, MaxIterations → 100]

Out[1]= 

Examples

Original Satellite Imagery

Mean Shift Clustered Imagery

Mean shift segmentation algorithm (cont.)

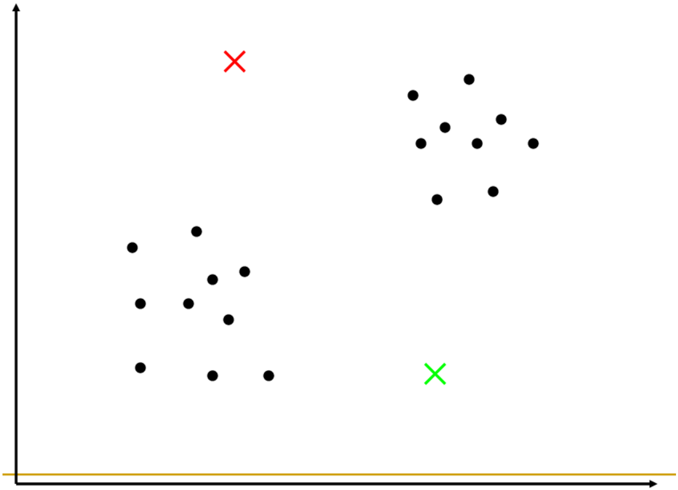
Examples

Segmentation using K-Means Clustering

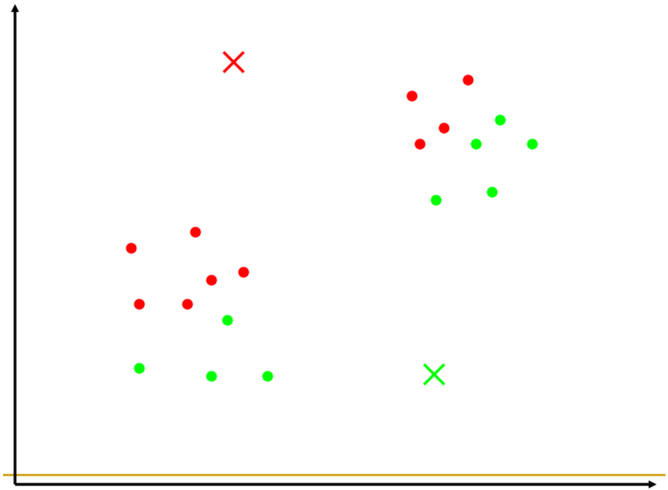
1. Partition the data points into K clusters randomly. Find the centroids of each cluster.
2. For each data point:
   2.1 Calculate the distance from the data point to each cluster.
   2.2 Assign the data point to the closest cluster.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

Segmentation using K-Means Clustering (cont.)

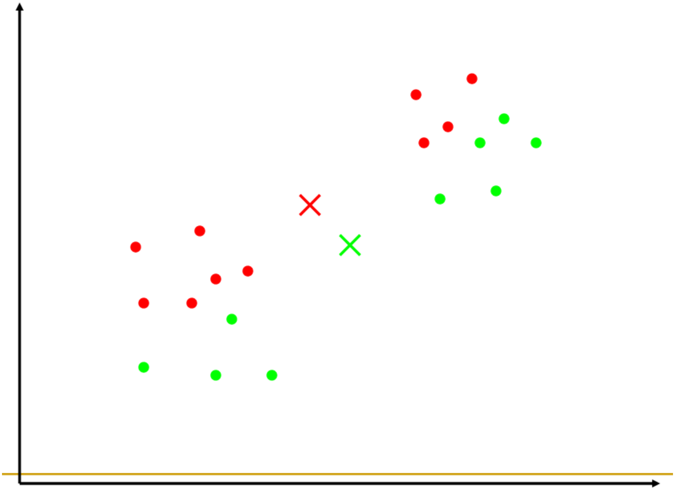
Segmentation using K-Means Clustering (cont.)



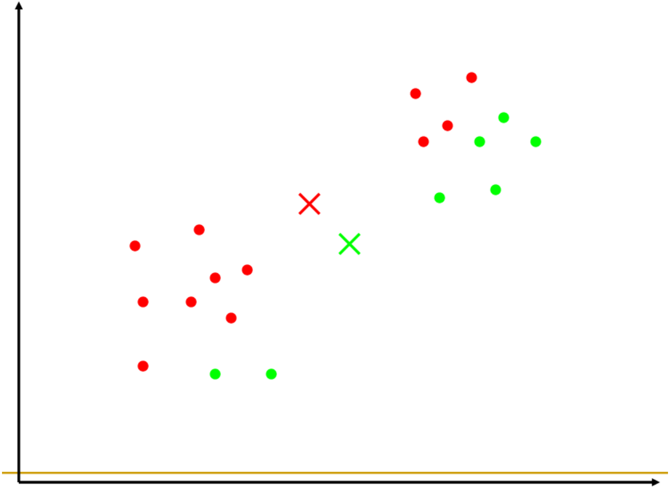
Segmentation using K-Means Clustering (cont.)



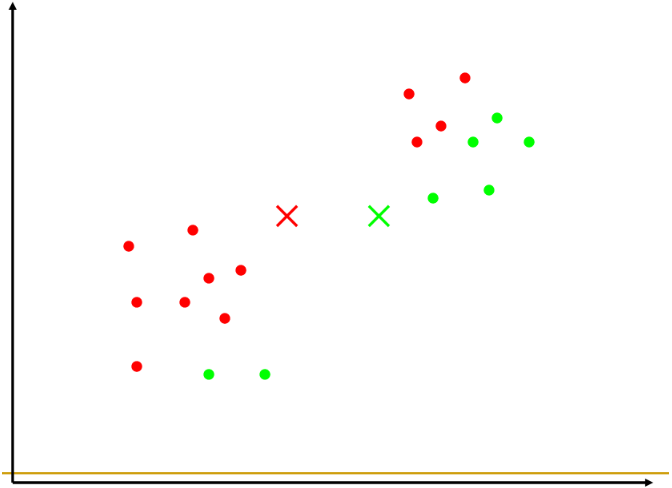
Segmentation using K-Means Clustering (cont.)



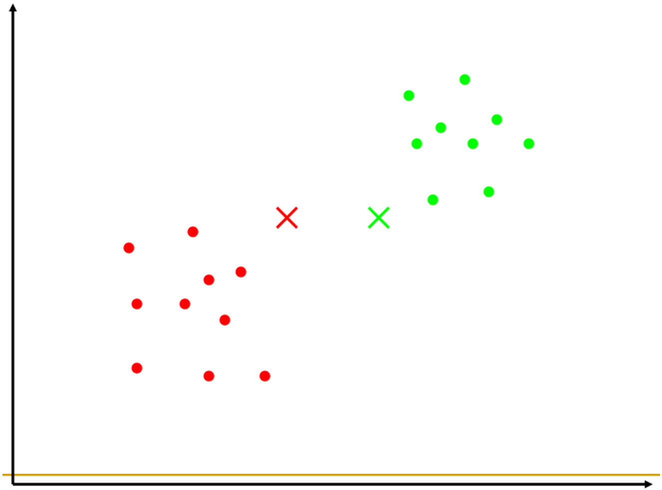
Segmentation using K-Means Clustering (cont.)



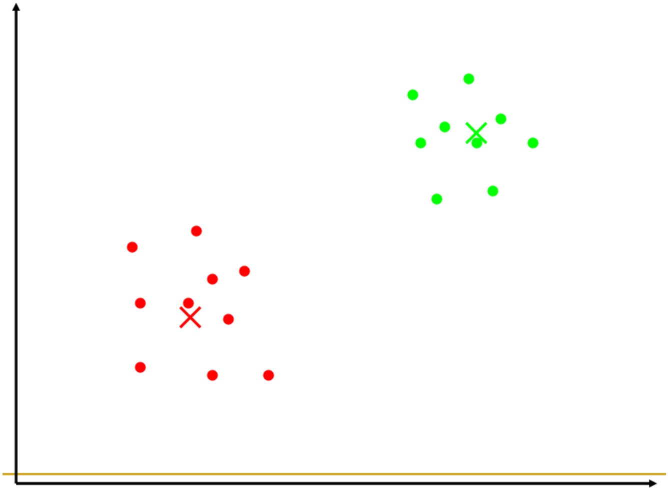
Segmentation using K-Means Clustering (cont.)



Segmentation using K-Means Clustering (cont.)

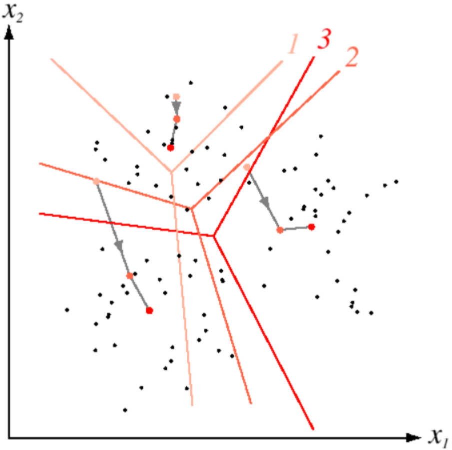


Segmentation using K-Means Clustering (cont.)



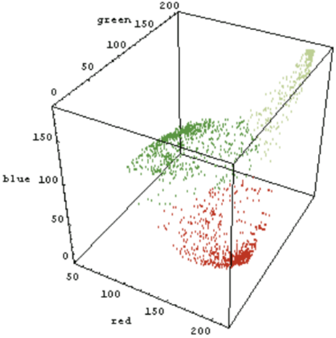
Segmentation using K-Means Clustering (cont.)

- Example



Duda et al.

- RGB vector

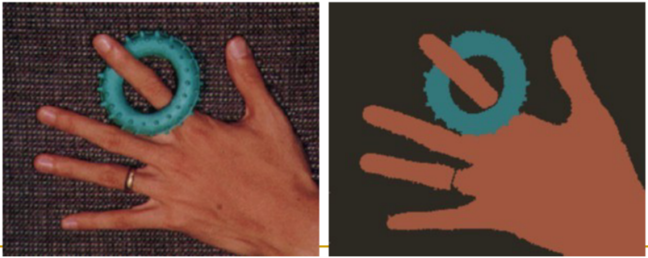


- K-means clustering minimizes
$\sum_{i \in clusters} \left\{ \sum_{j \in elementsofithcluster} ||x_j - \mu_i||^2 \right\}$

Examples

Segmentation using K-Means Clustering (cont.)

D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.



Segmentation using K-Means Clustering (cont.)

Original

K=5

K=11

K-Means Clustering: Axis Scaling

- Features of different types may have different scales.
  - For example, pixel coordinates on a 100x100 image vs. RGB color values in the range [0,1].
- Problem: Features with larger scales dominate clustering.
- Solution: Scale the features.