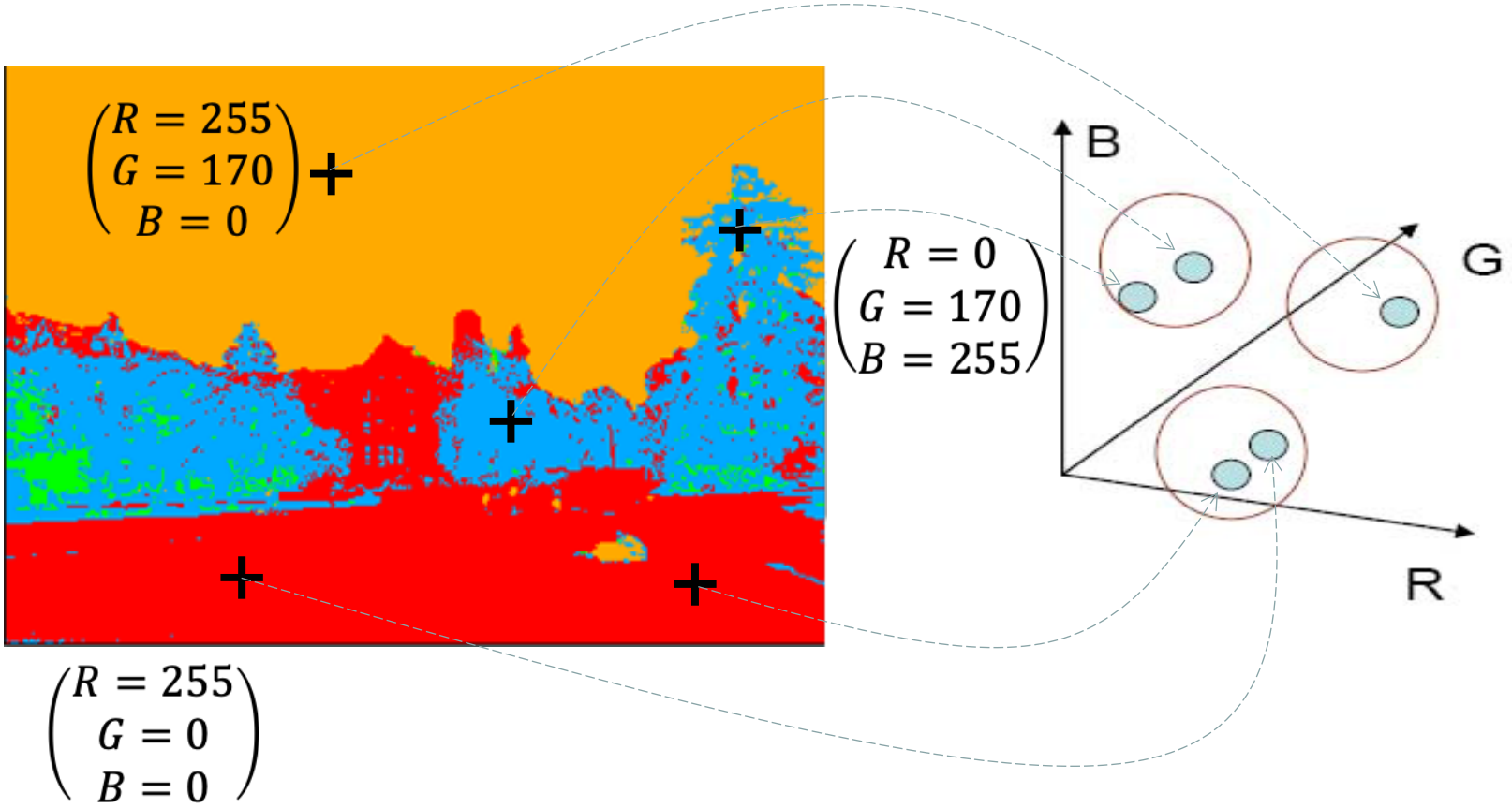
# K-means clustering

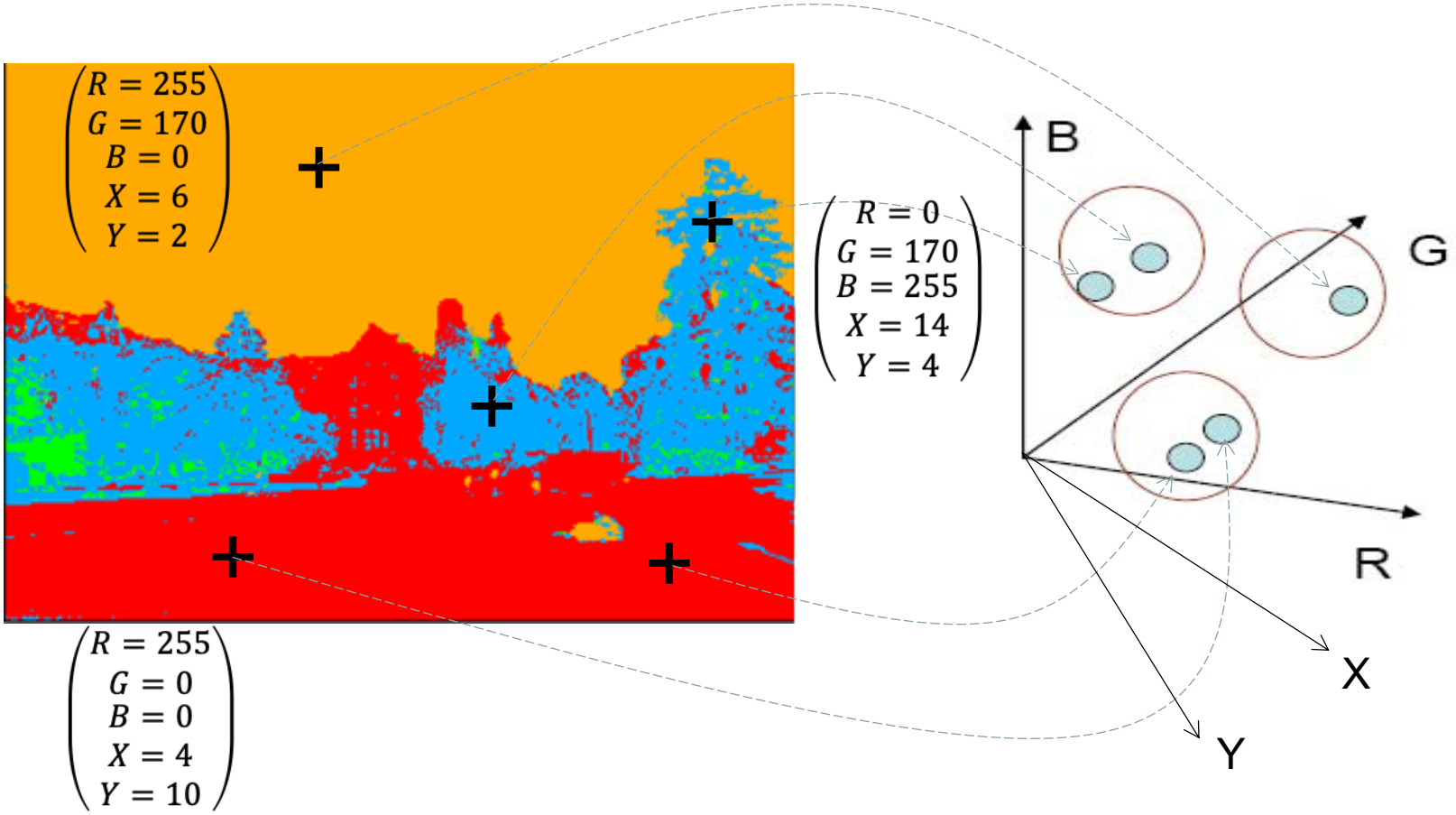
# Segmentation as clustering

- Cluster similar pixels using colour features



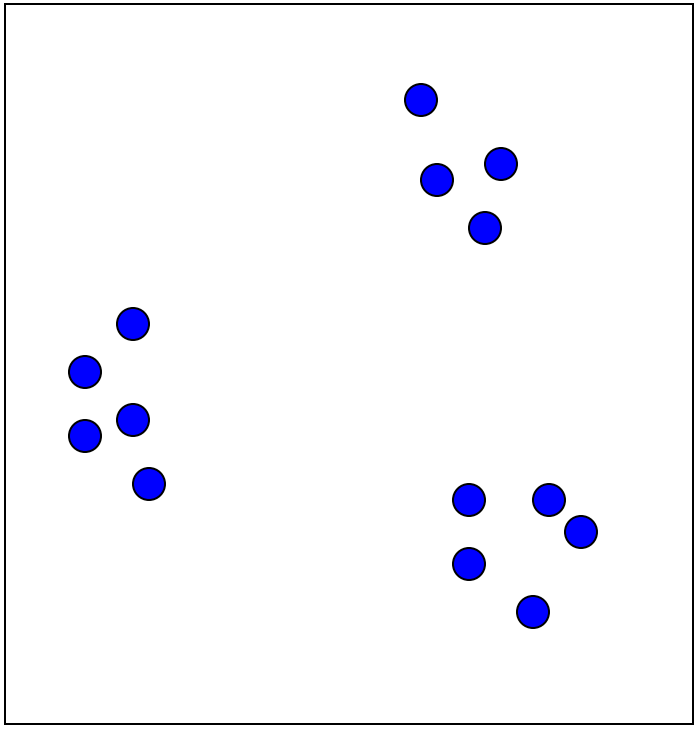
# Segmentation as clustering

- Cluster similar pixels together (RGB+ XY)



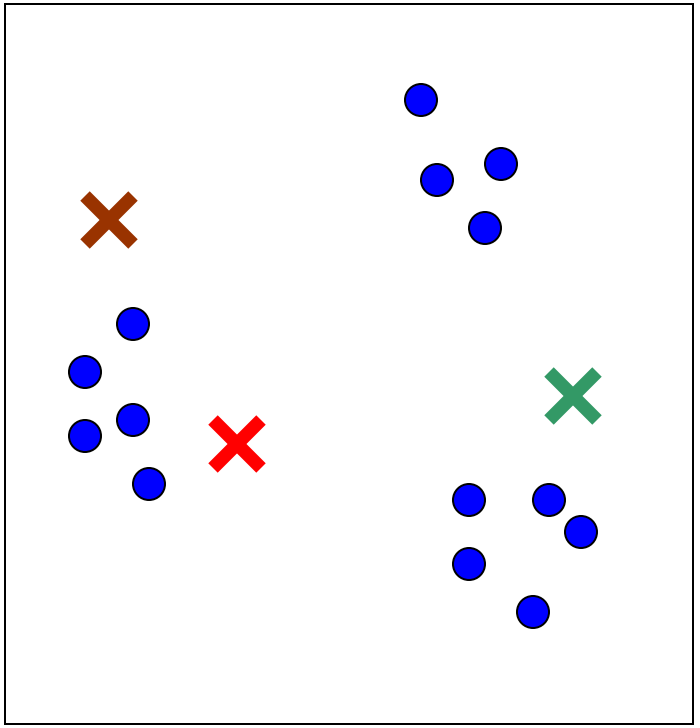
# K-means clustering

- "Guess" the number of clusters K before start (e.g. K=3)



# K(=3) means clustering

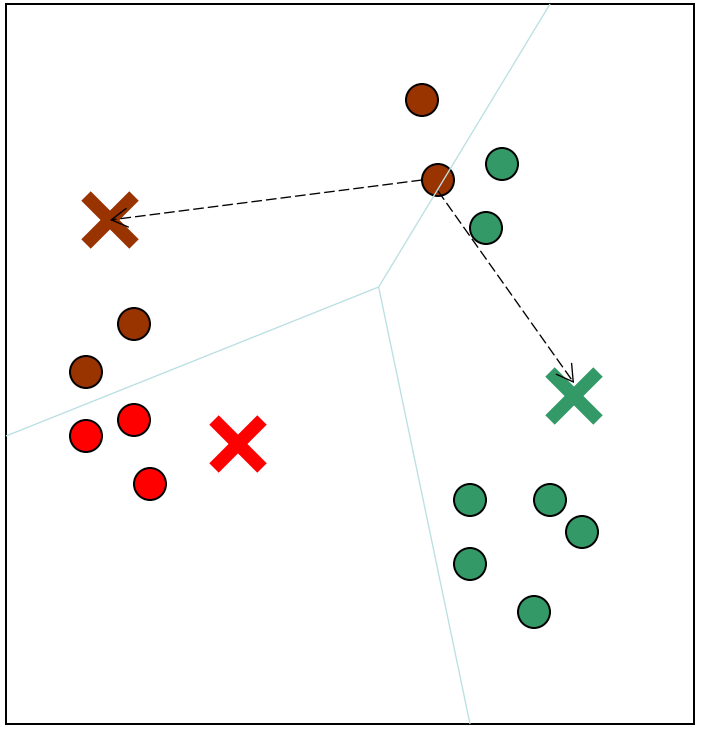
- Start with 3 *random* positions of the cluster *centers*.



Iteration = 0

# K-means clustering

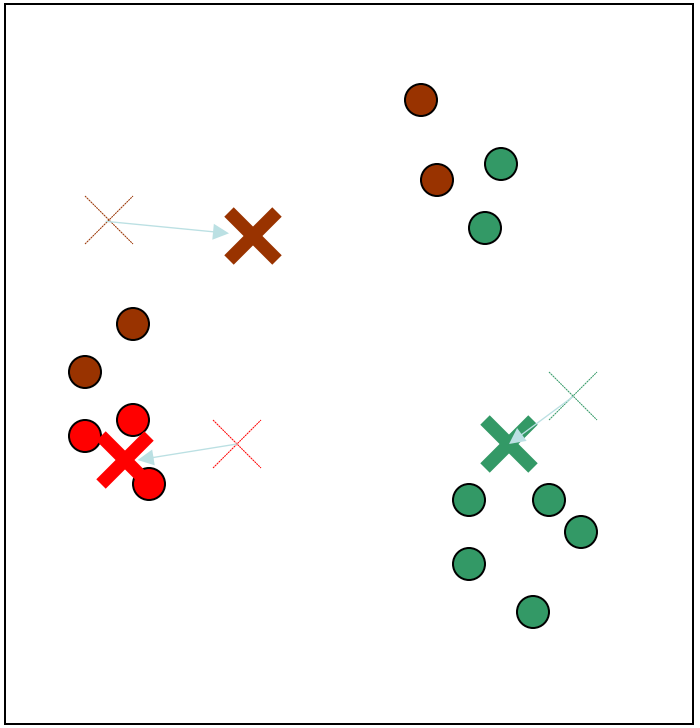
- Start with 3 random positions of the cluster centers.
- By computing the distance, *assign* each data point to the closest center.



Iteration = 1

# K-means clustering

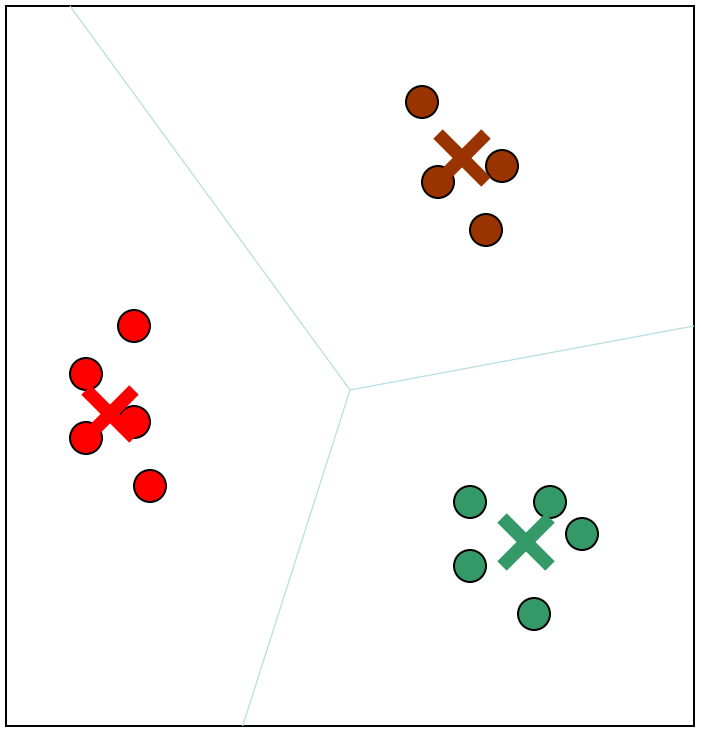
- Start with 3 random positions of the cluster centers.
- By computing the distance, *assign* each data point to the closest center.
- *Re-compute* the centers after the assignments.



Iteration = 1

# K-means clustering

- Start with 3 random positions of the cluster centers.
- By computing the distance, *assign* each data point to the closest center.
- Re-compute the centers after the assignments.
- Iterate until no points are reassigned.



Iteration = 3

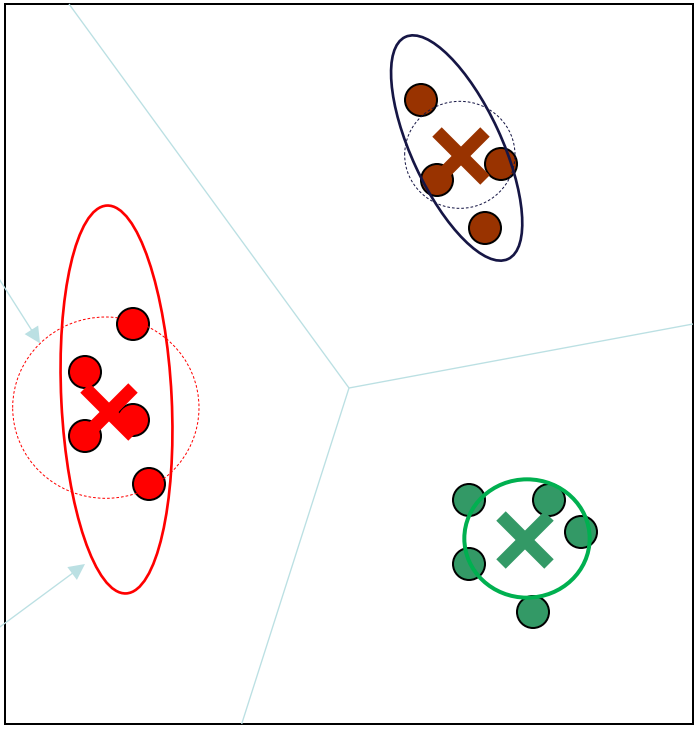
# K-means clustering

- Question) What have we minimised?

- We have selected the means and membership which minimise the sum of squared distances to the centre means (centroids).
- The sum of squared distances to the means is actually the variance of the cluster

# K-means clustering

- Spherical variance is used
- Q) What about the elongated variances?
  - Called Gaussian Mixture Model based clustering



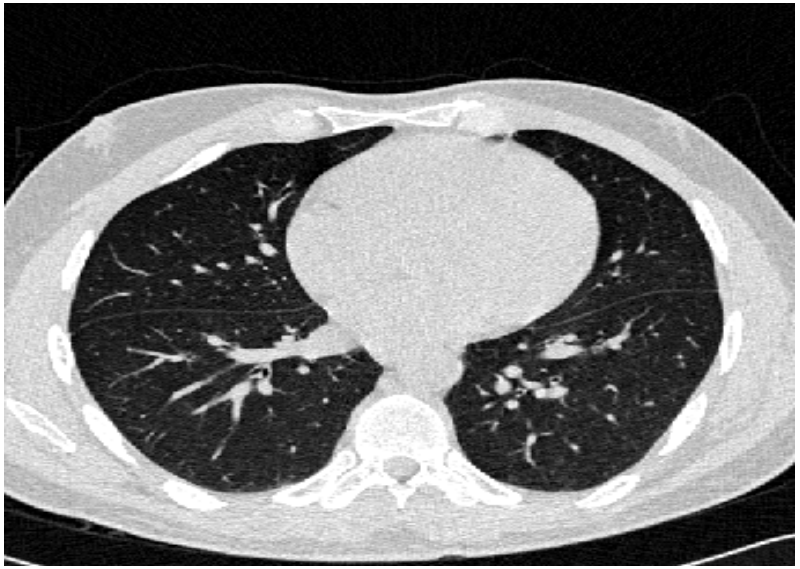
# K-means clustering

1. Select a value of K
2. Select a feature vector for every pixel (color, texture, position, or combination of these etc.)
3. Define a similarity measure between feature vectors (Usually Euclidean distance)
4. Apply the K-means algorithm to all the feature vectors

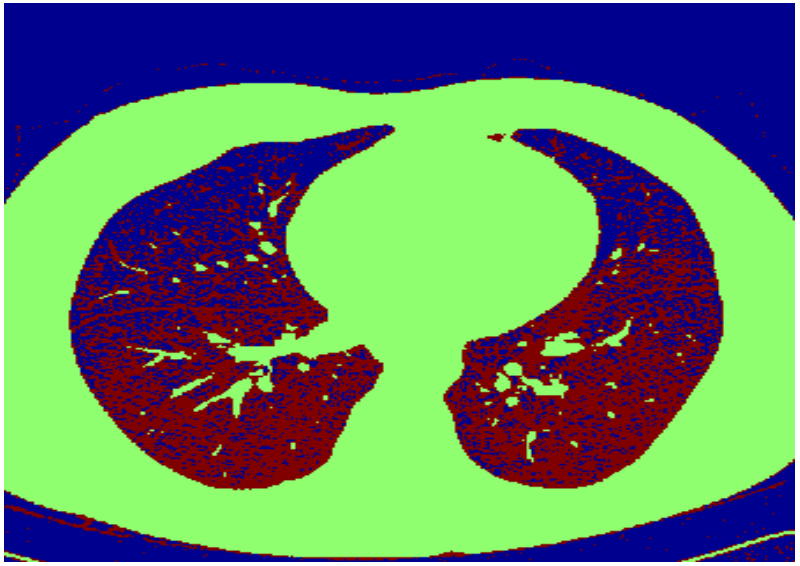


* From Marc Pollefeys COMP 256 2003

# K-means Image Segmentation



Input image (*I*)



Three-cluster image using the gray levels of *input image*

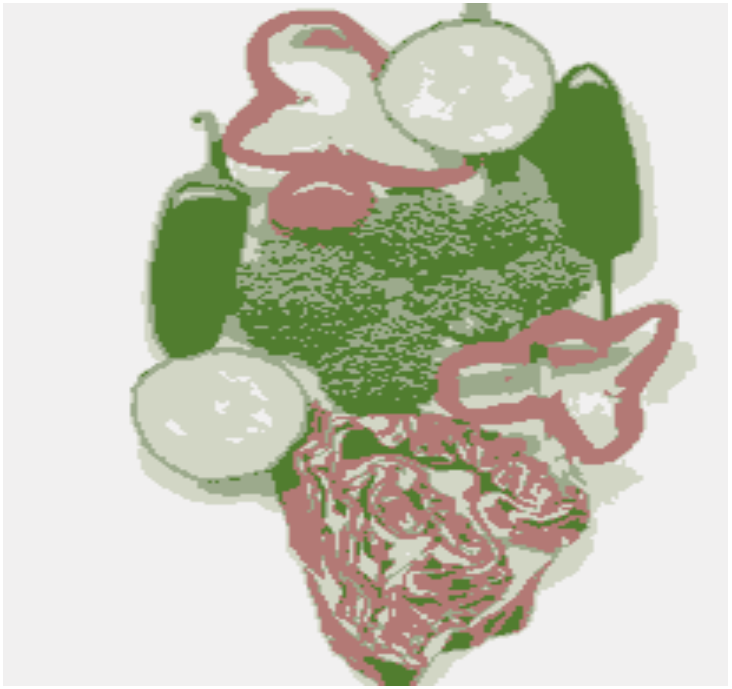
Matlab code:

I = double(imread('…'));

J = reshape(***kmeans***(I(:),3),size(I)); % need statistics toolbox

# K-means Image Segmentation

Color Image



Segmented on color

- K-means clustering using color information



* From Marc Pollefeys COMP 256 2003

# K-means Pros and Cons

Pros:

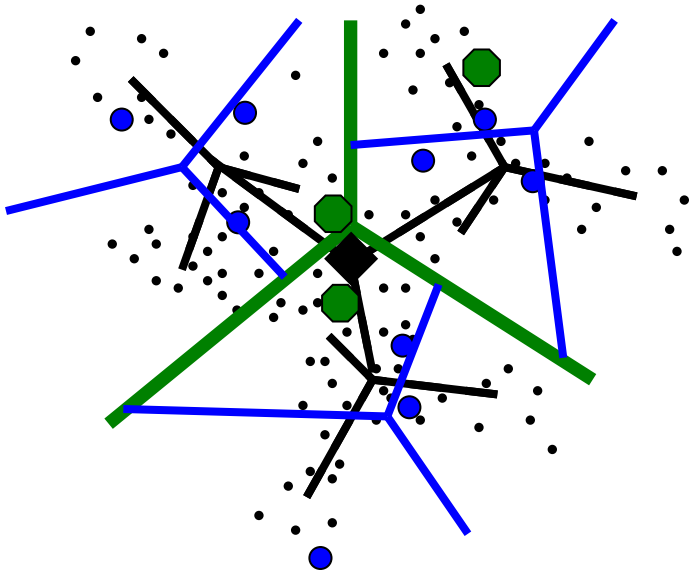
- Finds cluster centers that minimize variance (good representation of data)
- Simple to implement, widespread applications.

Cons:

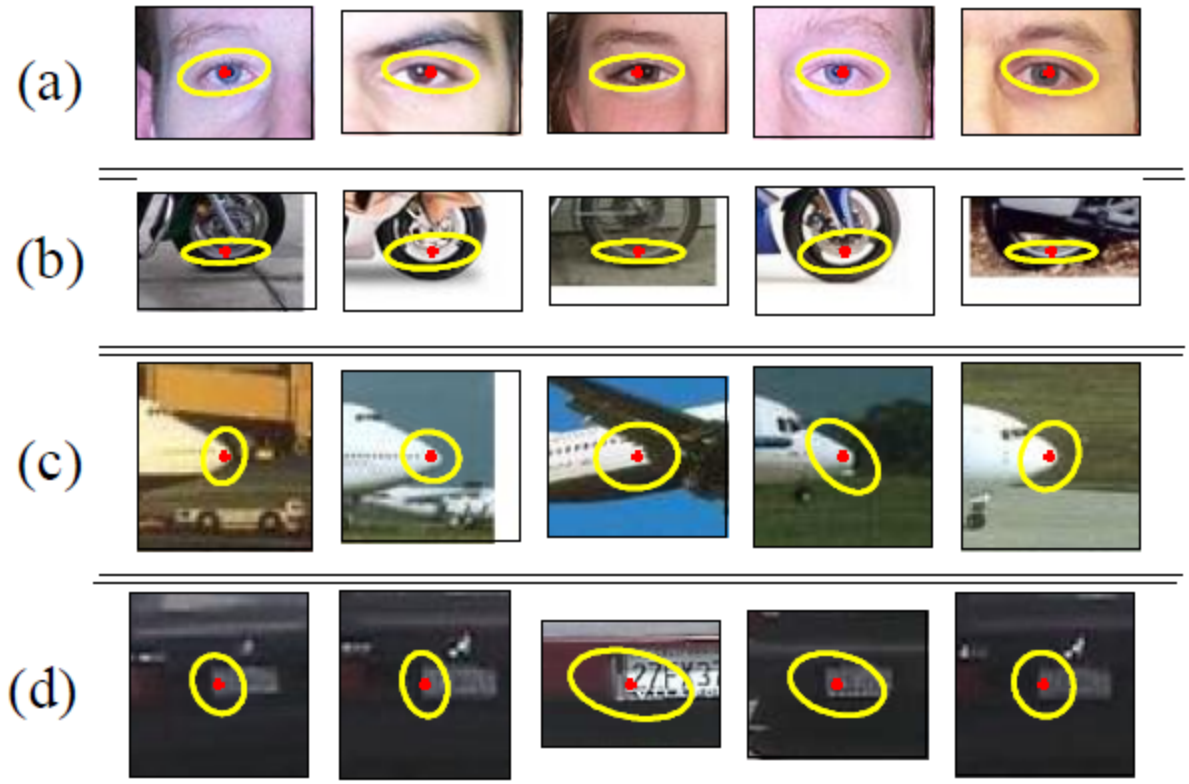
- All clusters have spherical distribution (same to all directions or isotropic)
- Hard membership/assignment (i.e. 1 or 0 membership)
- Prone to local minima
- Need to choose K
- Can be very slow:  each iteration is O(KN) for N-dimensional points

# Other application of K-means: Building Visual Dictionaries

1. Sample patches from a database
   – E.g., 128 dimensional SIFT vectors
2. Cluster the patches
   – Cluster centers are the dictionary
3. Assign a codeword (number) to each new patch, according to the nearest cluster



# Examples of learned visual words




Most likely codewords for 4 learned “topics”
EM with multinomial (problem 3) to get topics



http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic05b.pdf Sivic et al. ICCV 2005