Lecture 25

Image Segmentation by Clustering

ECEN5283 Computer Vision

Dr. Guoliang Fan

Oklahoma State University

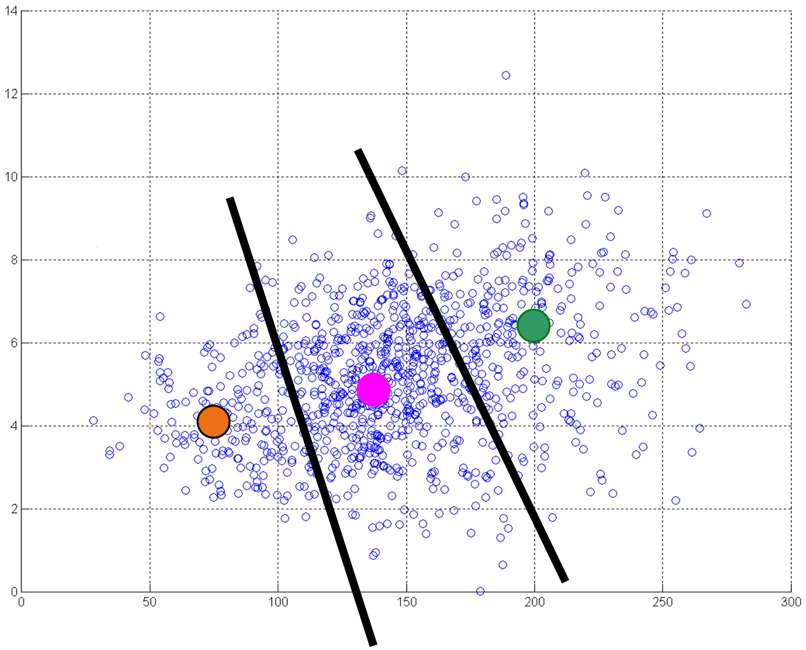
Goals

To review the Expectation Maximization (EM) algorithm and compare EM with K-mean

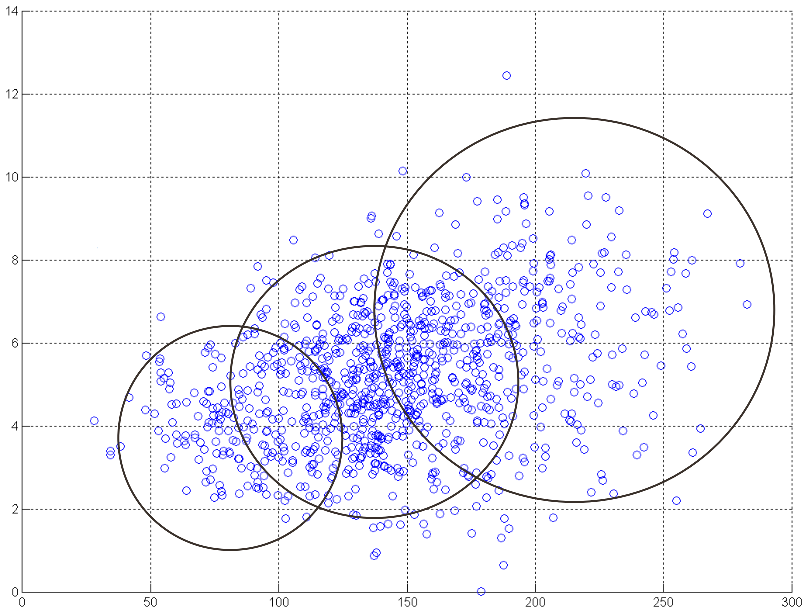
To introduce Project 4, image segmentation that involves Gabor based texture analysis and clustering

K-means vs. EM

$$\boldsymbol{y} = (y_1, y_2, \dots, y_N) \rightarrow \boldsymbol{x} = (x_1, x_2, \dots, x_N)$$



$$x_i = \arg_{j\in\{1,\dots,k\}} \min \left| y_i - \mathbf{c}_j \right|$$



$$x_i = \arg_{j\in\{1,\dots,k\}} \max p_{X|Y}(x_i = j | y_i, \Theta)$$

EM Algorithm: K-means Initialization

Random initialize k centers for the k-means algorithm.

$$C^0 = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$$

Run k-means multiple times and use the results with the minimum divergence.

$$\Phi(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{k} \left\{ \sum_{x_j = i} \left| y_j - \mathbf{c}_i \right|^2 \right\}$$

According to the class labels $\{x_1, x_2, \ldots, x_N\}$ of all data samples $\{y_1, y_2, \ldots, y_N\}$, initialize the multivariate Gaussian models for the EM.

$$\alpha_i^{(0)} = \frac{\#(x_j = i | j = 1, \ldots, N)}{N} \qquad \Sigma_i^{(0)} = \frac{\sum_{x_j=i} (y_j - \mu_i)(y_j - \mu_i)^T}{\#(x_j = i | j = 1, \ldots, N) - 1}$$

$$\mu_i^{(0)} = \frac{\sum_{x_j=i} y_j}{\#(x_j = i | j = 1, \ldots, N)}$$

\# counting the number

K-means vs. EM

	K-means	EM
Initialization	Initialize k means (cluster centers) $$C^0 = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$$	Initialize k Gaussian models that have equal weights. $$\Theta^0 = \{\alpha_i, \mu_i, \Sigma_i \mid i = 1, \ldots, k\}$$
Step 1.	Assume the cluster centers are known, and classify each data sample to the closest cluster center.	Given the model parameters, estimate the missing data in terms of the posterior probability of each data sample.
Step 2.	Assume the allocation (the class label of each sample) is known, and choose a new set of cluster centers. Each center is the mean of all points allocated to that cluster.	From the estimated missing data, to obtain the maximum likelihood estimate of the model parameters.

EM Algorithm: Additional Issues

Initialization:

- Both EM and K-means only converge to a local optimum.
- Using K-means result is a practical way to initialize the EM algorithm.
- Run K-means multiple times and pick the best result (?) for initialization.

Iteration:

- Iteration still the stop criteria is satisfied, e.g., no much change of the data log-likelihood or the iteration number can be fixed as a constant.

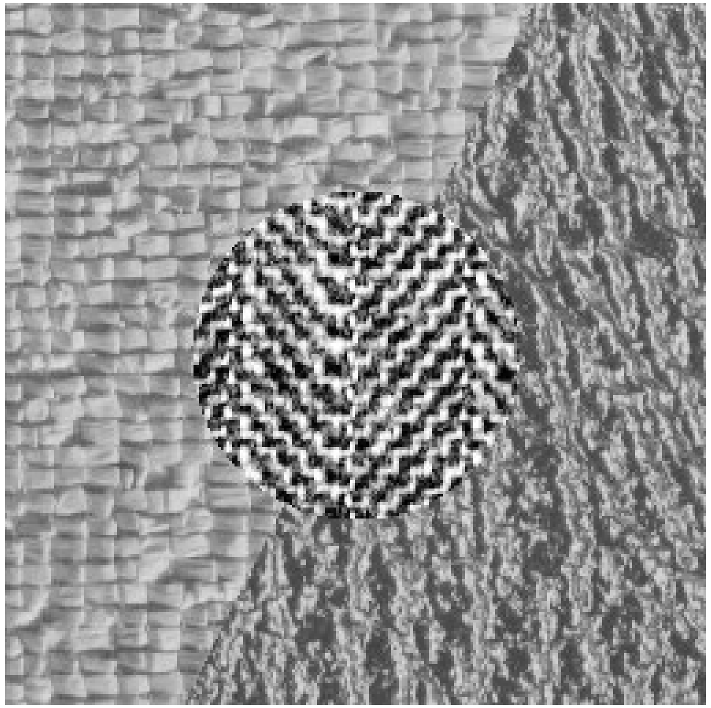
$$\log p_Y\left(\boldsymbol{y} \middle| \Theta^{(s+1)}\right) - \log p_Y\left(\boldsymbol{y} \middle| \Theta^{(s)}\right) < \Delta$$

Data classification:

- MAP classification $x_l = \arg_{m \in \{1,\ldots,g\}} \max \mathbf{I}(l, m)$

Image Segmentation

The goal of image segmentation is to segment an image into G homogeneous regions.



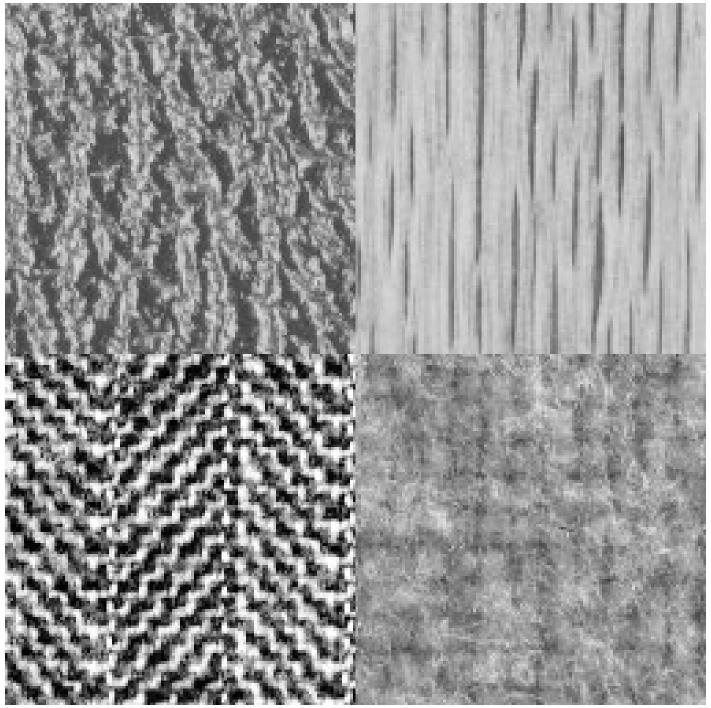


Image Segmentation by Clustering

Problem Formulation

- An image has totally *N* pixels each of pixel is associated with a feature vector $\boldsymbol{y} = (y_l | l = 1, \ldots, N)$, and the question is what is the class label for each pixel $\boldsymbol{x} = (x_l | l = 1, \ldots, N)$?

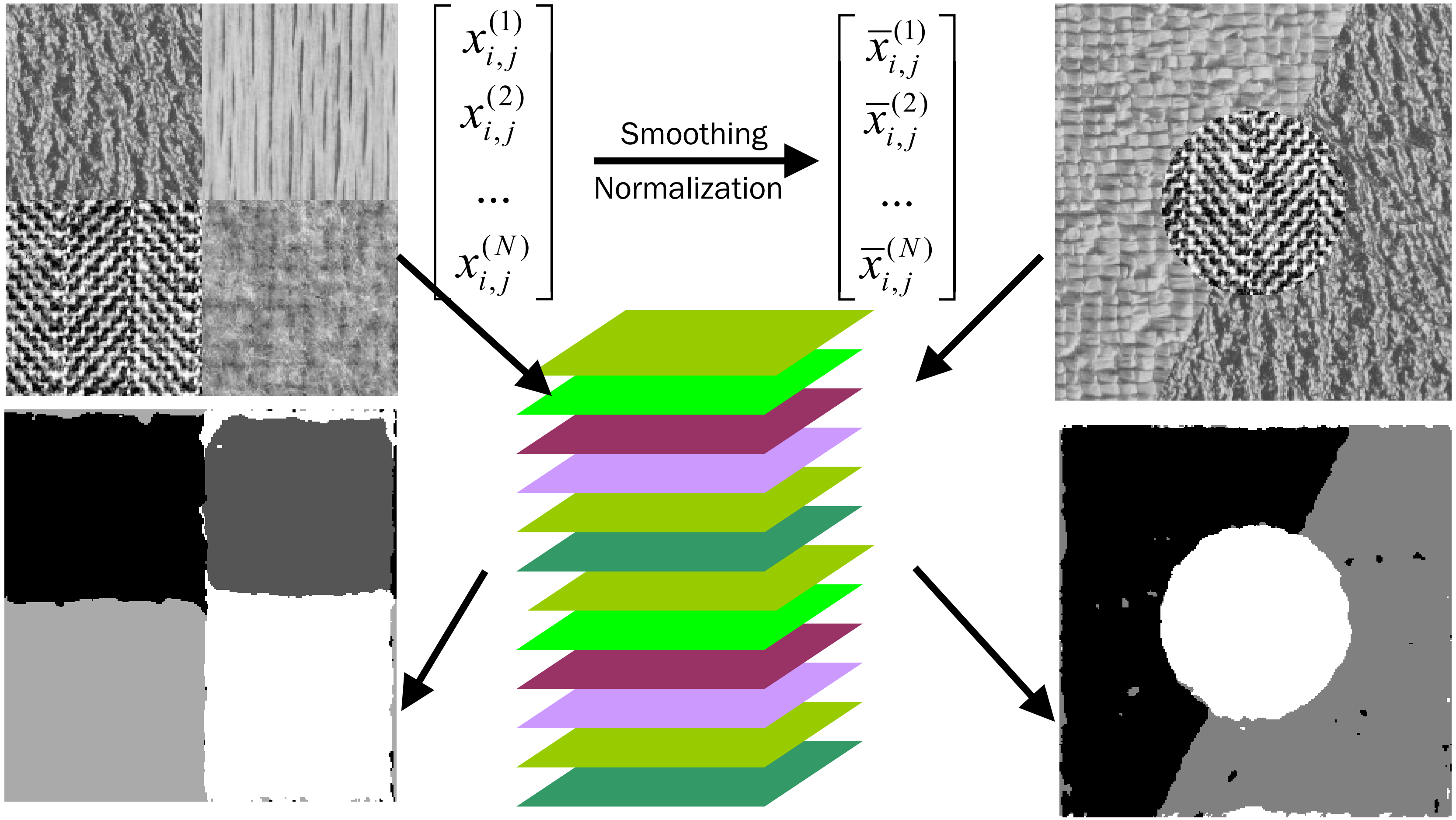
Assumption

- It is assumed that feature vectors of G classes are defined by G multivariate Gaussians $\Theta = \{\alpha_m, \theta_m = (\mu_m, \Sigma_m) | m = 1, \ldots, G\}$ with different priors, leading to a Gaussian Mixture Model (GMM) distribution.

Solution:

- Image segmentation becomes a clustering problem that can be solved by K-means or EM.

Feature Extraction: Gabor Filtering



Requirements of Project 4 (Due April 6).

Implement the EM algorithm for image segmentation and compare it with the K-means results.

- You can use the KMEANS function for K-means clustering.
- Try several runs of K-means to find the best clustering result (how?).
- Discuss the effect of initialization on EM based on different initial parameters.

Some plots to show to support your discussion and analysis

- Plot the data log-likelihood $\log p_Y(\boldsymbol{y}|\Theta)$ vs. the iteration number.
- Plot the segmentation accuracy vs. the iteration number

Additional requirements

- Create a video to show the progression of the segmentation map during EM iterations.
- Find a few color images and use RGB (3-channel colors) and/or XY (pixel coordinates) for clustering-based color segmentation.

Matlab Programming (1)
Built-in K-means Clustering Function

```
dx = kmeans(X,k)
[idx,C] = kmeans(X,k)
[idx,C,sumd] = kmeans(X,k)
[idx,C,sumd,D] = kmeans(X,k)
```

idx = kmeans(X,k) performs k-means clustering to partition the observations of the n-by-p data matrix X into k clusters, and returns an n-by-1 vector (idx) containing cluster indices of each observation. Rows of X correspond to points and columns correspond to variables. By default, kmeans uses the squared Euclidean distance metric and the k-means++ algorithm for cluster center initialization.

[idx,C] = kmeans(X,k) returns the k cluster centroid locations in the k-by-p matrix C.

[idx,C,sumd] = kmeans(X,k) returns the within-cluster sums of point-to-centroid distances in the k-by-1 vector sumd.

[idx,C,sumd,D] = kmeans(X,k) returns distances from each point to every centroid in the n-by-k matrix D.

Matlab Programming (2): Segmentation Accuracy Evaluation

```
function Per=accuracy(Truth,Result,Num);
% Truth is the groud-truth map;
% Result is the obtained map, Num is the number class
% Per is the segmentation accuracy
[X Y]=size(Truth);
Z=zeros(256,256);
for i=1:X
    for j=1:Y
        p=Truth(i,j)+1;
        q=Result(i,j);
        Z(p,q)=Z(p,q)+1;
    end
end
T=sum(max(Z));
Per=T/X/Y;
```

Ground Truth

1	1	1	2	2
1	1	1	2	2
3	3	4	4	4
3	3	4	4	4
3	3	4	4	4

Clustering Result

3	3	3	4	4
3	1	2	4	4
2	2	1	1	1
2	3	1	3	1
2	2	1	1	1

1	1	4	0
0	0	0	4
0	5	1	0
8	0	1	0
8	5	4	4

Sum=21

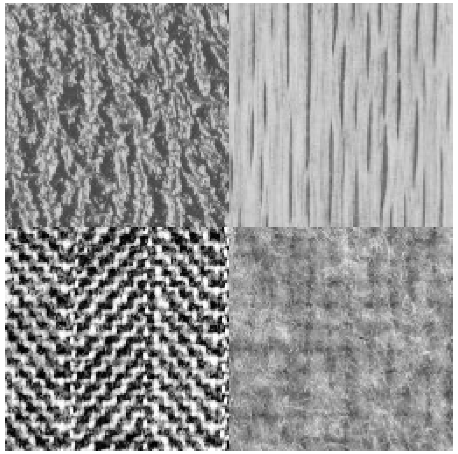
Matlab Programming (3)

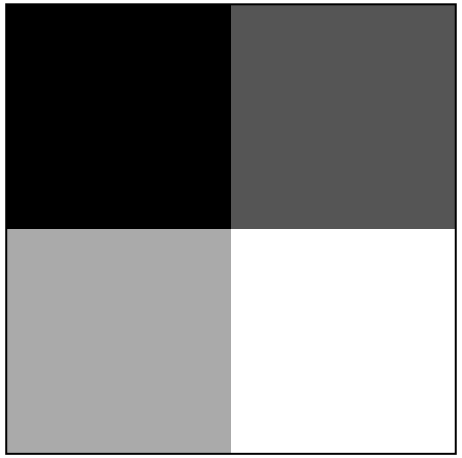
You can use the Matlab built-in Gaussian PDF (normpdf)

- y = normpdf(x) returns the probability density function (pdf) of the standard normal distribution, evaluated at the values in x.
- y = normpdf(x,mu) returns the pdf of the normal distribution with mean mu and the unit standard deviation, evaluated at the values in x.
- y = normpdf(x,mu,sigma) returns the pdf of the normal distribution with mean mu and standard deviation sigma, evaluated at the values in x.

Texture Segmentation using K-mean and EM Algorithms (Mosaic A)

Mosaic A





Ground truth segmentation

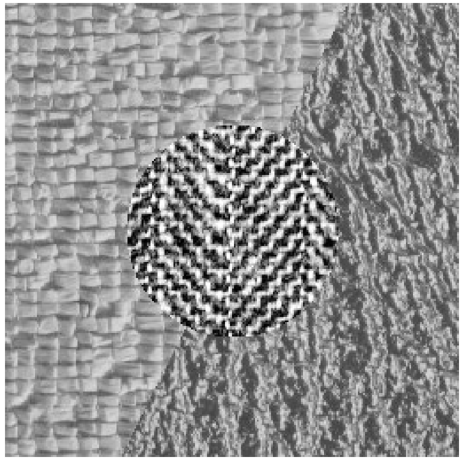
K-means clustering result

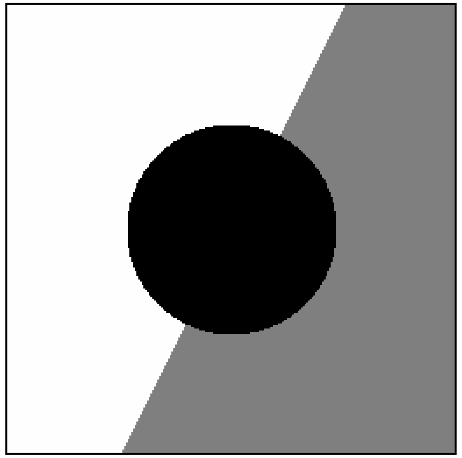
EM algorithm result

The 5-scales and 8-orientation Gabor filtering is used for texture feature extraction leading to a feature vector of 5x8=40.

Texture Segmentation using K-mean and EM Algorithms (Mosaic B)

Mosaic B

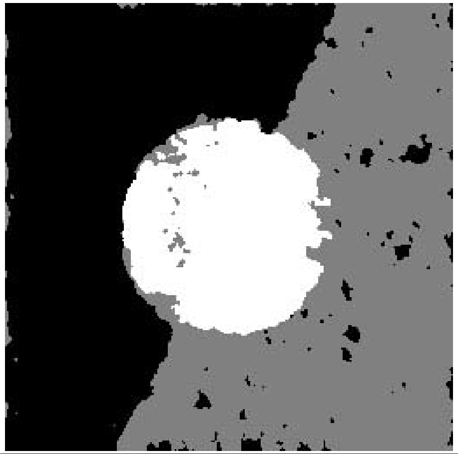


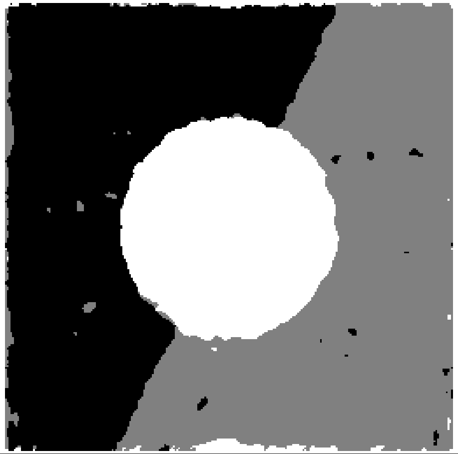


Ground truth segmentation

The 5-scales and 8-orientation Gabor filtering is used for texture feature extraction leading to a feature vector of 5x8=40.

K-means clustering result





EM algorithm result