Digital Image Processing

Lecture #5

Ming-Sui (Amy) Lee

Course Information

- **Following Schedule**

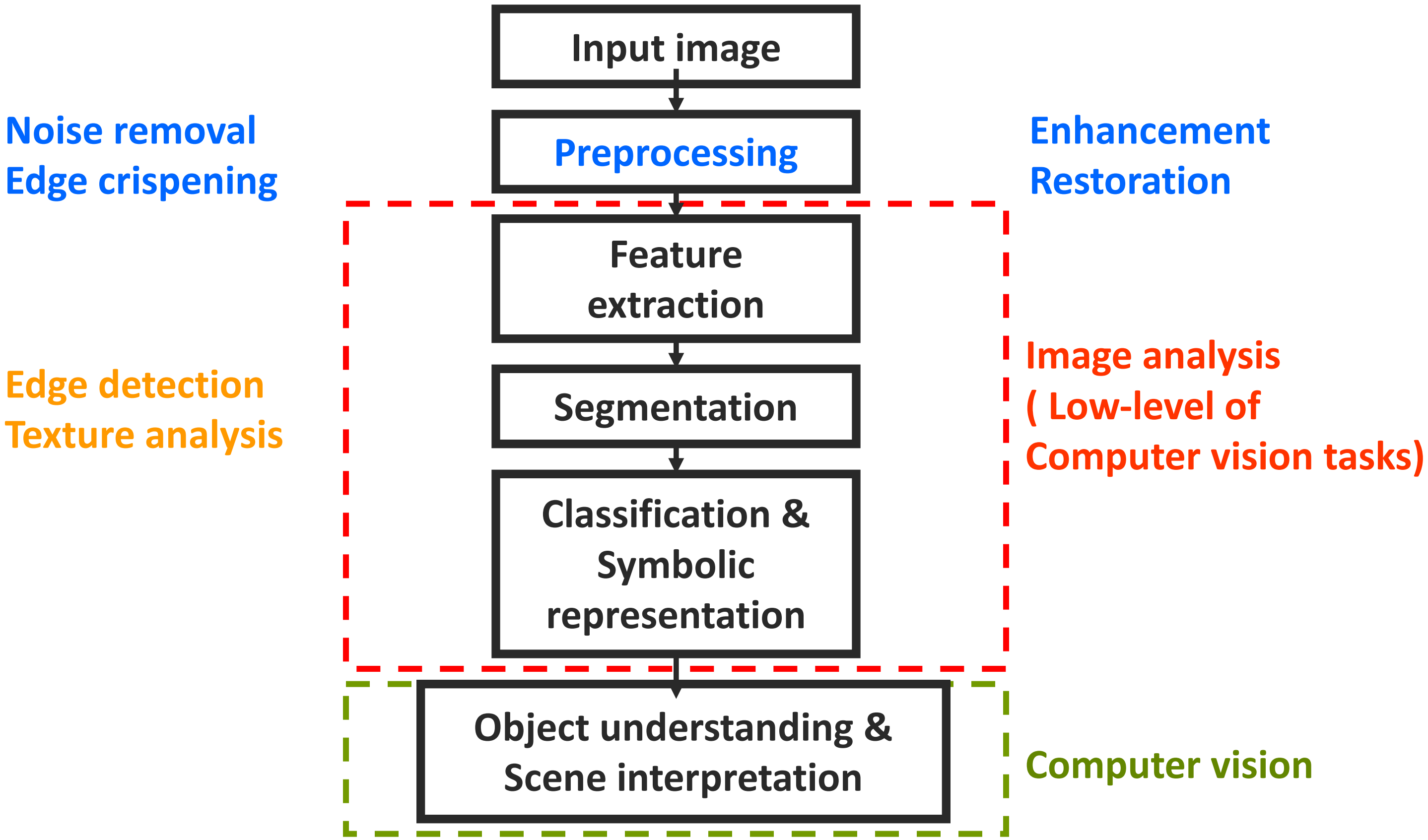
02/25	Lecture 1	04/29	Lecture 8
03/04	Lecture 2	05/06	Proposal
03/11	Lecture 3	05/13	Lecture 9
03/18	Lecture 4	05/20	Lecture 10
03/25	Lecture 5	05/27	Lecture 11
04/01	溫書假	06/03	Lecture 12
04/08	Lecture 6	06/10	Demo
04/15	Lecture 7	06/17	Demo
04/22	Midterm	06/24	Final Package Due

- **Homework #2**
  - **Due: 11:59pm, Apr. 7, 2021**

Texture Analysis

Texture Analysis

- **Image analysis and its applications**



Texture Analysis

- **What is texture?**

Texture Analysis

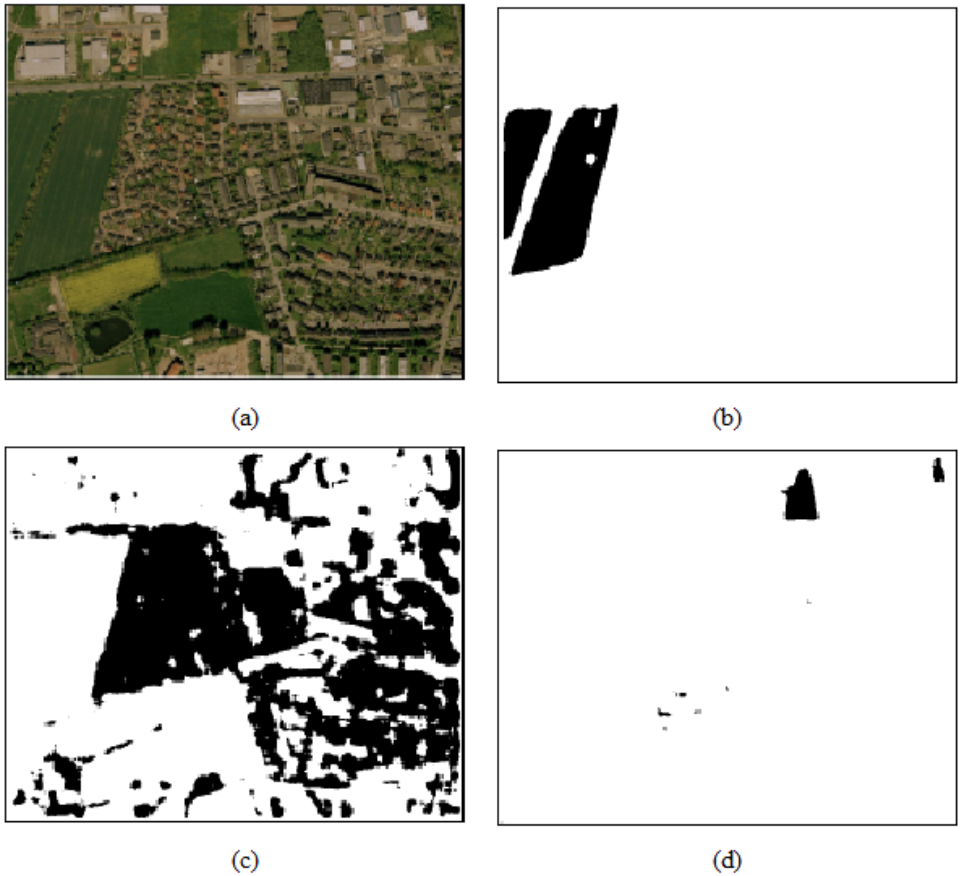
- What is texture?
  - No mathematical definition
  - Two dimensional arrays of variations
  - Semi-regular structured patterns of object
  - E.g. Surfaces such as sand, grass, wool, cloth, leaves, etc.

Texture Analysis

- Why texture analysis?
  - People started to be interested in late 50’s and early 60’s
    - Analyze aerial images / texture patches

Texture Analysis

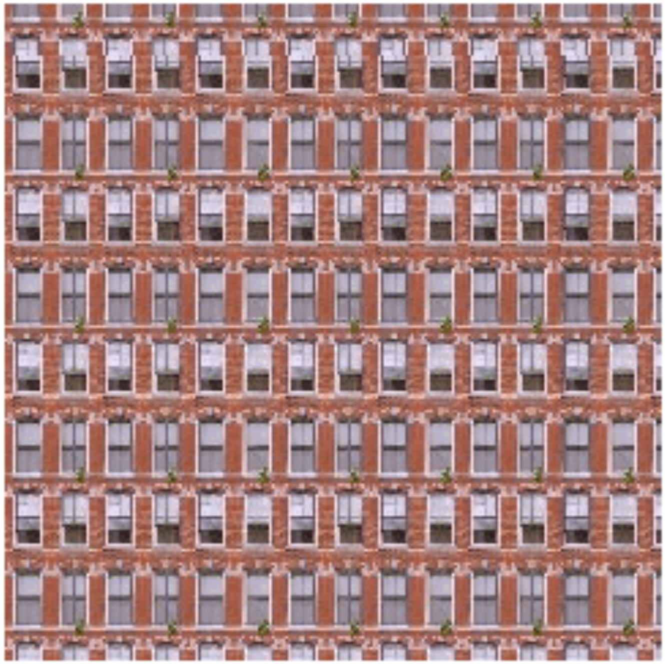
- **Example (an aerial image)**



(a) Aerial photo (b)Field (c) Residential area (d) Vegetation area

Texture Analysis

- **Example**
  - **Texture Synthesis**



?

Texture Analysis

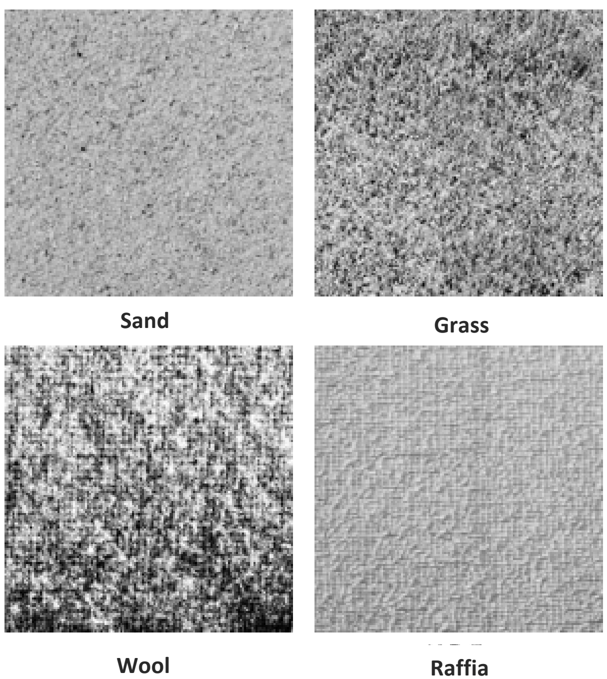
- History of texture analysis
  - Fourier Spectral Methods
  - Edge Detection Methods
  - Autocorrelation Methods
  - Decorrelation Methods
  - Dependency Matrix Method

Texture Analysis

- Fourier Spectral Methods
  - Right direction but incomplete development
  - No continuous work for a long while
- Edge Detection Methods
  - Edge detection
  - Use edge density and orientation as texture features

Texture Analysis

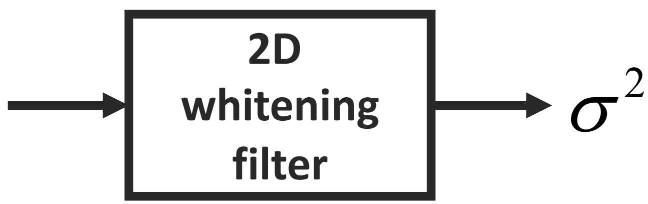
- Autocorrelation Methods
  - Treat the texture pattern as a 2D random process, denoted as F(x,y)
  - Statistical approach



$$E\{F(x,y)F(x-\Delta x, y-\Delta y)\}$$

Texture Analysis

- Decorrelation Methods
  - 2D whitening filter
    - Special type of decorrelation operator



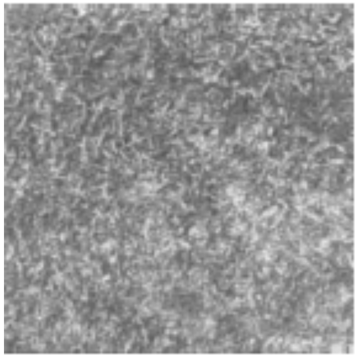
$$\hat{W}(j,k) = F(j,k) \otimes H_W(j,k)$$

  - Spatially decorrelated
    - Form histogram as its feature

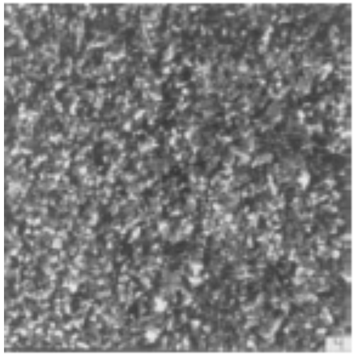
Texture Analysis

- **Dependency Matrix Method**
  - **Joint probability**
  - **Also called Co-occurrence method**

$$P(a,b \mid j,k,\Delta j,\Delta k)$$
$$= \mathrm{Prob}\{F(j,k)=a, F(j-\Delta j,k-\Delta k)=b, 0 \le a,b \le L-1\}$$



Grass



Ivy

Texture Analysis

- History of texture analysis
  - Fourier Spectra methods
  - Edge Detection Methods
  - Autocorrelation Methods
  - Decorrelation Methods
  - Dependency Matrix Method

→ Not successful!!

Texture Analysis

- Laws' Method
  - Micro-structure (Multi-channel) method
    - Emphasize the microstructure of the texture
    - Two steps
      - step 1: Convolution
      - step 2: Energy computation

Microstructure array

$$M_i(j,k) = F(j,k) \otimes H_i(j,k)$$

Local features

$$F(j,k) \rightarrow \boxed{H_i(j,k)} \rightarrow M_i(j,k) \rightarrow \boxed{\text{Energy computation}} \rightarrow T_i(j,k)$$

For 3x3 mask
i=1,2,3,...,9

convolution

Texture Analysis

- **Laws' Method**
  - //Step 1// Convolution $M_i(j,k) = F(j,k) \otimes H_i(j,k)$
    - Micro-structure impulse response arrays (a basis set)

$H_i(j,k)$

for 3x3 mask, i=1,2,3,…,9

for 5x5 mask, i=1,2,3,…,25

How to choose the mask size?

$$\frac{1}{36}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$
Laws 1

$$\frac{1}{12}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$
Laws 2

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$
Laws 3

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$
Laws 4

$$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$
Laws 5

$$\frac{1}{4}\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$
Laws 6

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$
Laws 7

$$\frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}$$
Laws 8

$$\frac{1}{4}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
Laws 9

Texture Analysis

- **Laws' Method**
  - **Micro-structure impulse response arrays**
    - **Generated by the tensor product of the 1D horizontal and vertical masks**

$$L_3 = \frac{1}{6}[1 \quad 2 \quad 1] \qquad E_3 = \frac{1}{2}[-1 \quad 0 \quad 1] \qquad S_3 = \frac{1}{2}[1 \quad -2 \quad 1]$$

Local averaging | **Edge detector (1st-order gradient)** | **Spot detector (2nd–order gradient)**

- **E.g.**

$$L_3^T \otimes E_3 = \frac{1}{6}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \frac{1}{2}[-1 \quad 0 \quad 1] = \frac{1}{12}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Laws 2

Texture Analysis

- **Laws' Method**
  - **Micro-structure impulse response arrays**
    - 1979 → 1984,1986 mathematical analysis of Laws' filters
    - Examine the frequency response of L_3, E_3, and S_3

$$L_3 = \frac{1}{6}[1 \quad 2 \quad 1]$$

Kroneckor Delta

$$h[n] = \frac{1}{6}\left(\delta[n-1] + 2\delta[n] + \delta[n+1]\right) \qquad \delta[n] = \begin{cases} 1 & n = 0 \\ 0 & otherwise \end{cases}$$

$$H(\omega) = \frac{1}{6}\left(e^{-j\omega} + 2 + e^{j\omega}\right) = \frac{2}{6}(1 + \cos\omega)$$

→ Low-pass filter

Texture Analysis

- **Laws' Method**
  - **Micro-structure impulse response arrays**
    - **Examine the frequency response of** $L_3,\ E_3,\ and\ S_3$

$$E_3 = \frac{1}{2}[-1 \quad 0 \quad 1] \qquad h[n] = \frac{1}{2}\left(-\delta[n-1] + \delta[n+1]\right)$$

$$H(\omega) = \left(-e^{-j\omega} + e^{j\omega}\right) = 2j\sin\omega$$

→ **Bandpass filter**

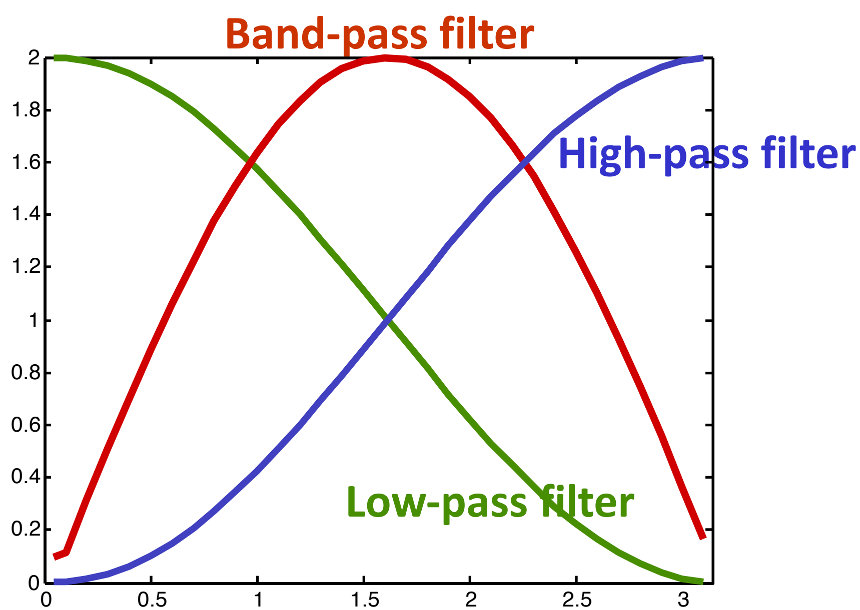
$$S_3 = \frac{1}{2}[1 \quad -2 \quad 1] \quad h[n] = \frac{1}{2}\left(\delta[n-1] - 2\delta[n] + \delta[n+1]\right)$$

$$H(\omega) = \frac{1}{2}\left(e^{-j\omega} - 2 + e^{j\omega}\right) = \cos\omega - 1$$

→ **High-pass filter**

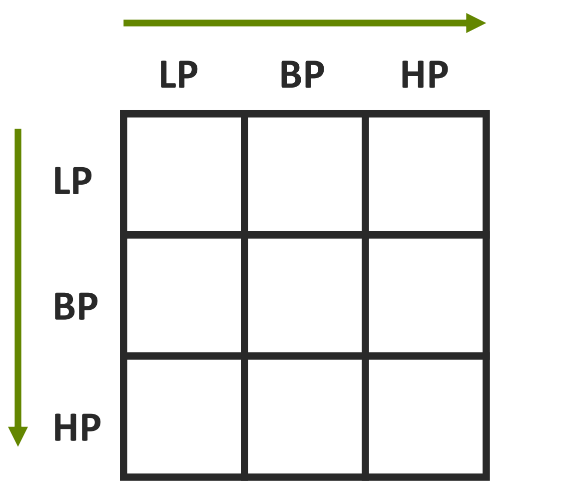
Texture Analysis

- **Laws' Method**
  - Micro-structure impulse response arrays
    - Examine the frequency response of $L_3,\ E_3,\ and\ S_3$



$$H_i(j,k)$$

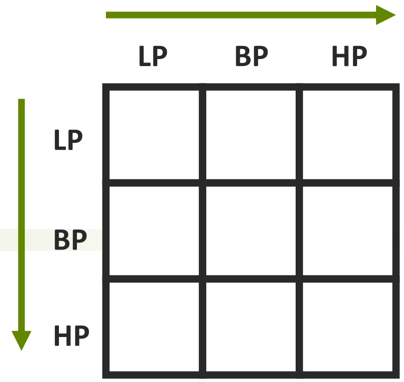
for 3x3 mask, i=1,2,3,…,9

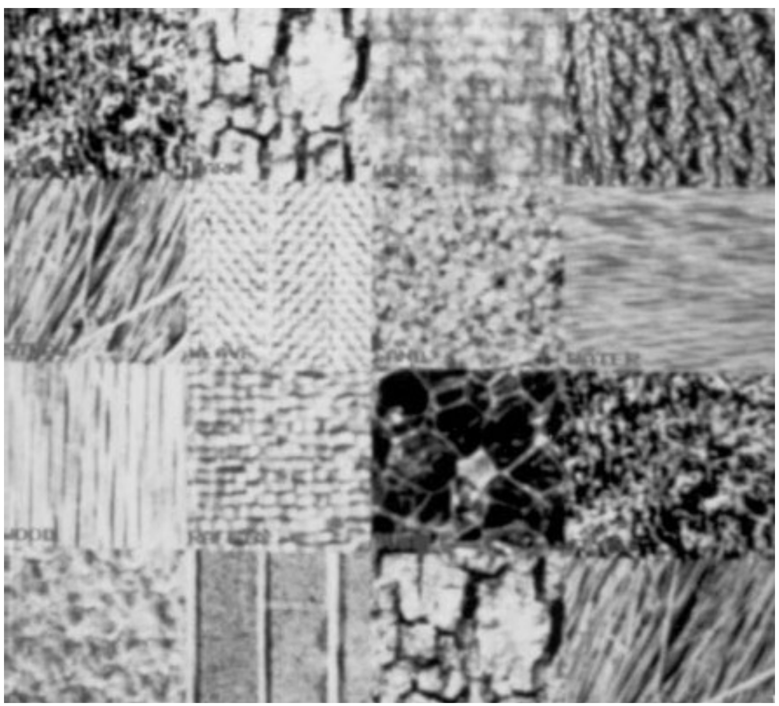


// Multi-channel method //

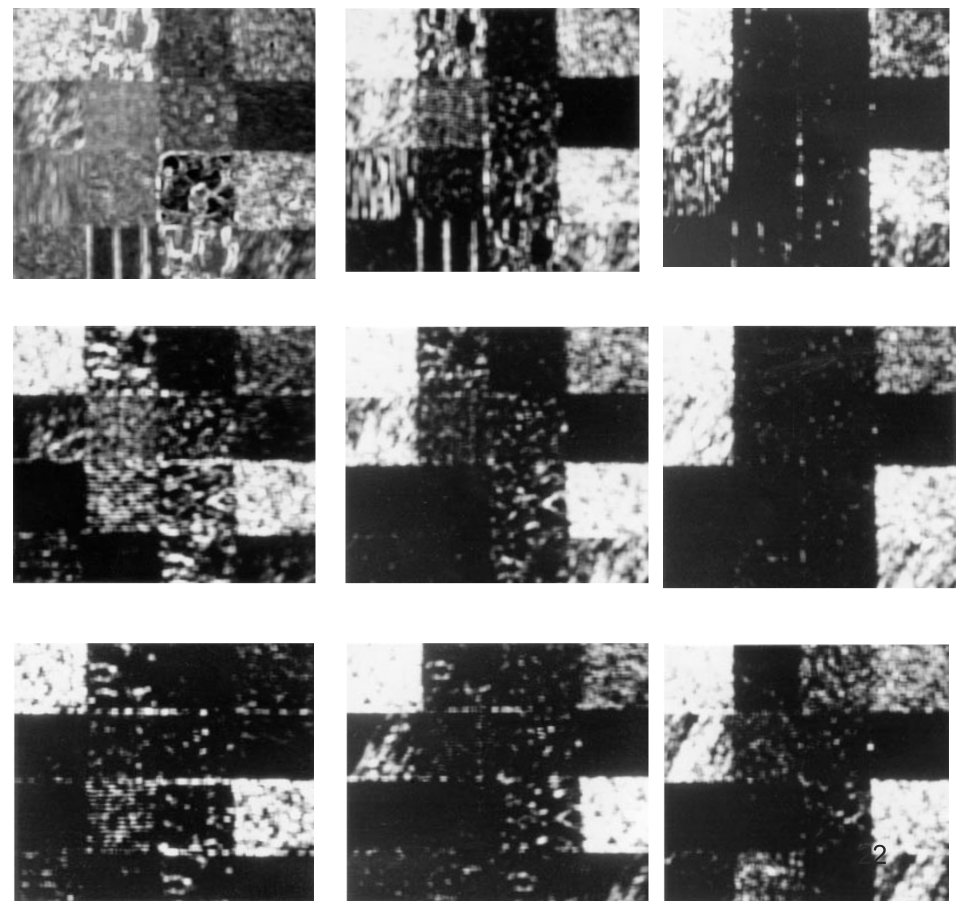
Texture Analysis

- Example





original image

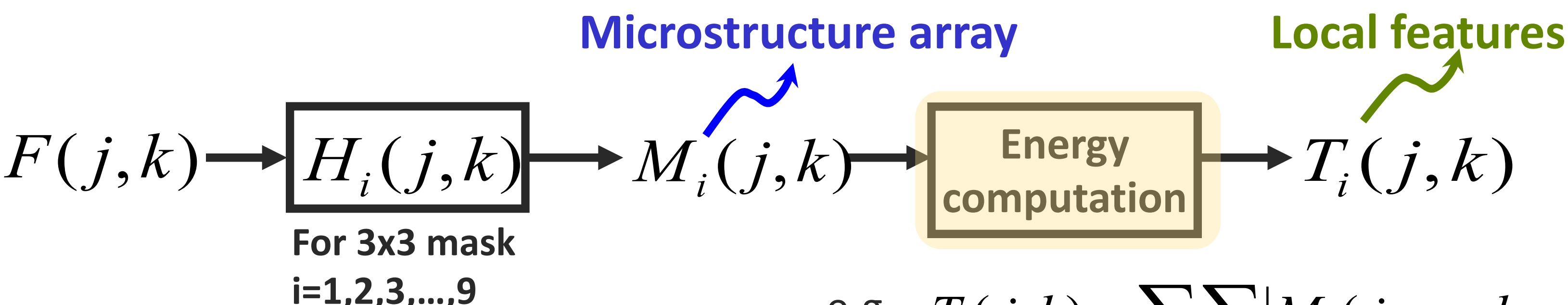


Texture Analysis

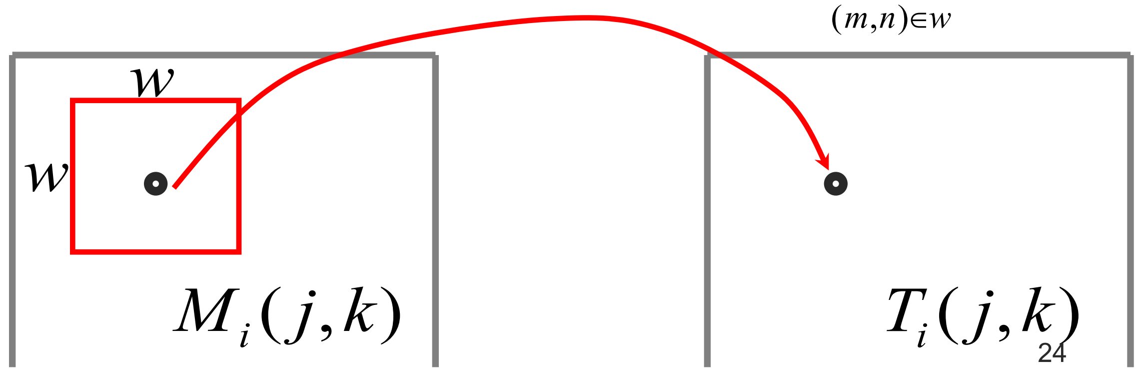
- Laws' Method
  - //Step 2// Energy Computation $T_i(j,k)$
    - Extract features over a window that has a few cycles of the repetitive texture
    - How to choose the window size?
    - Global/local energy computation
    - 9 energy features correspond to the energy in the 9 subbands. We use the energy distribution in these 9 subbands to differentiate different texture types
    - Features
      - Mean, standard deviation, energy, smoothness etc.

Texture Analysis

- **Laws' Method**
  - //Step 2// Energy Computation



e.g. $$T_i(j,k) = \sum_{(m,n)\in w}\sum \left|M_i(j+m,k+n)\right|^2$$



Texture Analysis

- Notes for Laws' method
  - How to choose the mask size? $H_i(j,k)$
  - Fixed subband structure vs
    Dynamic subband structure
  - How to choose the window size for energy computation?
    - For texture analysis, window size is usually set to be 13x13 or 15x15

Texture Analysis

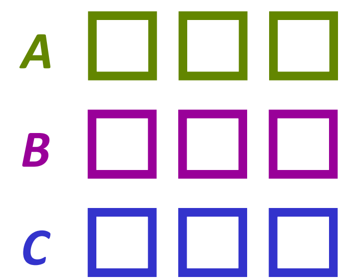
- Texture classification/segmentation
  - Given 9 feature sets, $T_1, \ T_2, \ T_3, \cdots \ T_9$

    How do we do texture classification?
  - Two cases
    - Each input is homogeneous
    - Single input consists of more than one texture
  - Two approaches
    - Supervised texture classification
    - Un-supervised texture classification

Texture Analysis

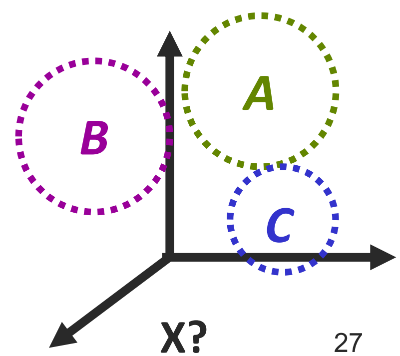
- Texture classification
  - Supervised texture classification
    - For each given texture type

$$\begin{array}{lcl} textureA & \rightarrow & T_{A1},\ T_{A2},\ T_{A3},\cdots\ T_{A9} \\ textureB & \rightarrow & T_{B1},\ T_{B2},\ T_{B3},\cdots\ T_{B9} \\ textureC & \rightarrow & T_{C1},\ T_{C2},\ T_{C3},\cdots\ T_{C9} \end{array}$$



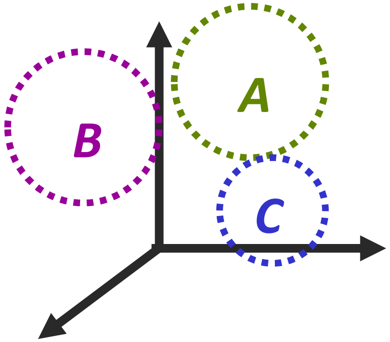
    - Texture space → 9 dimensional
    - Given texture X

      Use nearest neighbor classification rule



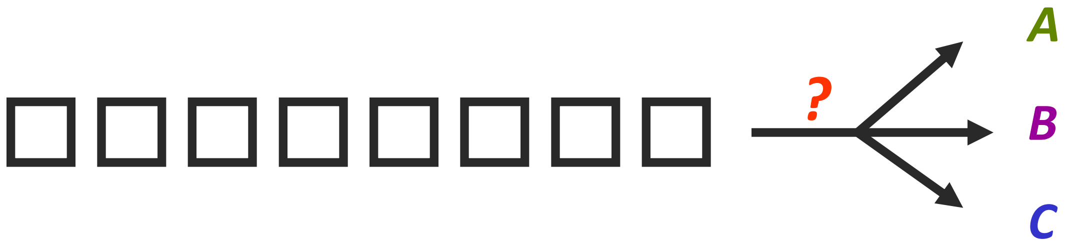
Texture Analysis

- Texture classification
  - Feature space dimension reduction
    - Not considering all 9 features equally
    - More important feature
      - More discriminating power
      - Weighted more
    - Less important feature
      - Weighted less
      - Taken out from the feature set



Texture Analysis

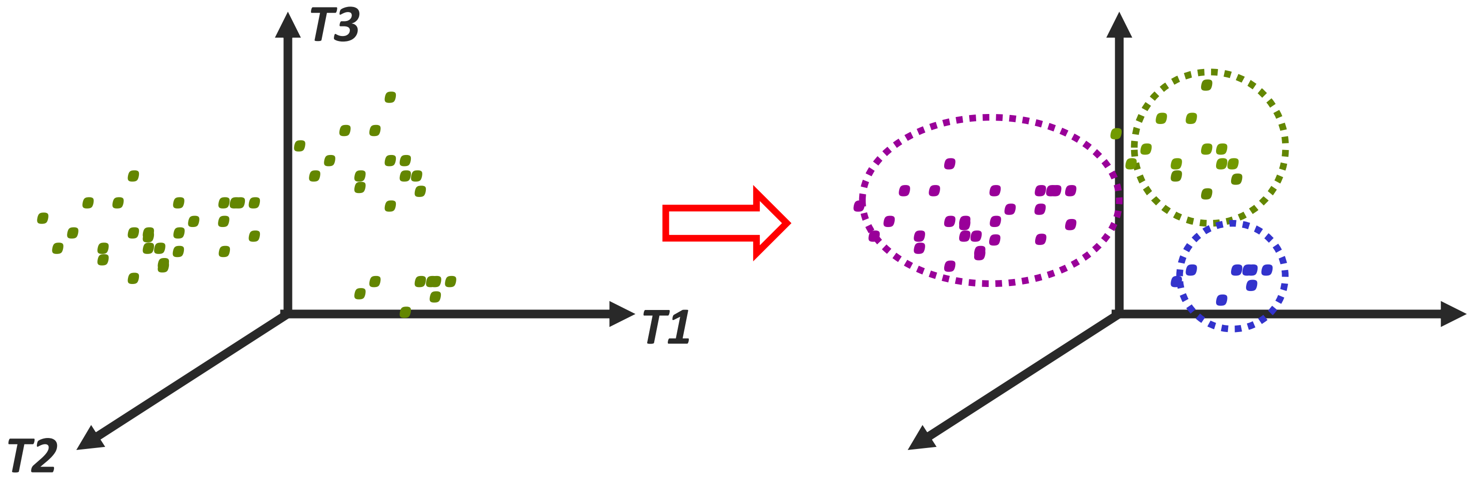
- Texture classification
  - Un-supervised texture classification
    - For several texture patches



  - K-means algorithm
    - The famous tool to handle unsupervised classification problem

Texture Analysis

- K-means algorithm
  - K=3



  - Good classification
    - Inter-clustering →
    - Intra-clustering →

Texture Analysis

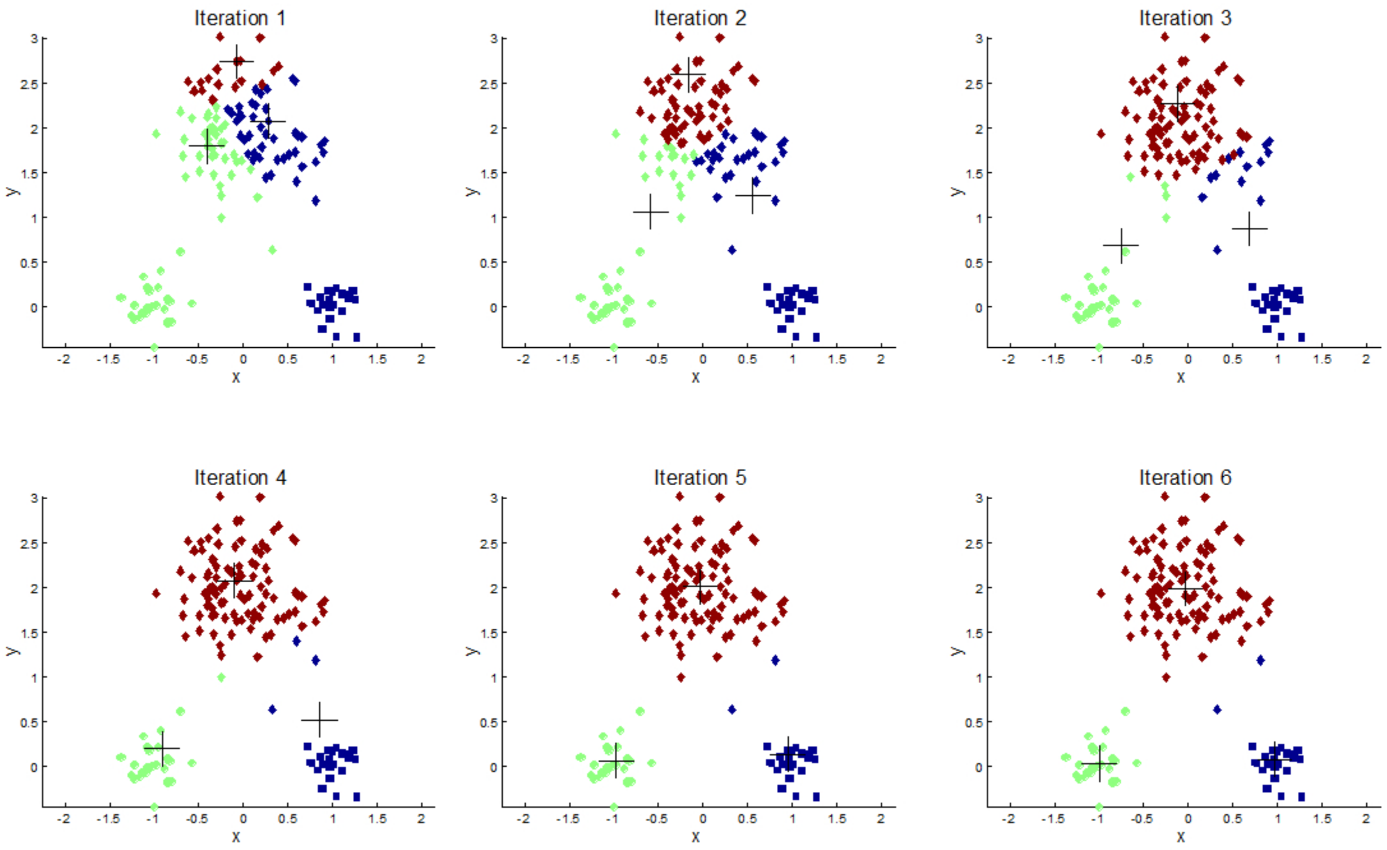
- K-means algorithm
  - Two issues
    - How to choose k?

      depends on the inter-cluster and intra-cluster statistical analysis
      OR by the problem set-up (domain knowledge)

    - Given k, how to do the clustering?
      - // Initialization //
        - Select k vectors as the initial centroids
        - Do the following iterations
      - // step1 // Form k clusters using the NN rule
      - // step2 // re-compute the centroid of each cluster

Texture Analysis

- **K-means algorithm demo**



Texture Analysis

- **Texture classification**
  - **Two criteria**
    - **If pixels belong to the same type of texture, their associated feature vectors are close to each other in the feature space**
    - **Pixels belong to the same texture type should be close to each other in the space domain**
  - **What is a good segmentation result?**
    - **Regions of a segment should be homogeneous w.r.t. some properties (i.e. feature vectors are close to each other in the feature space)**
    - **Region interior should be simple and without many holes**
    - **Boundaries of each segment should be simple, not ragged**