UNIT IV

IMAGE ANALYSIS AND PATTERN RECOGNIZATION

- Spatial feature extraction
- Edge detection
- Boundary extraction and representation
- Region and moment Representation
- Structure – Texture
- Scene matching and detection
- Image segmentation
- Classification techniques
- Pattern recognition
- Perceptron for two pattern classes
- Training algorithm
- Non separable classes.

In computer vision systems such as the one shown in Fig.   the input image is first preprocessed, which may involve restoration, enhancement, or just proper representation of the data. Then certain features are extracted for *segmentation* of the image into its components—for example, separation of different objects by extracting their boundaries. The segmented image is fed into a classifier or an image understanding system. Image classification maps different regions or segments into one of several objects, each identified by a label. For example, in sorting nuts and bolts, all objects identified as square shapes with a hole may be classified as nuts and those with elongated shapes, as bolts. Image understanding systems determine the relationships between different objects in a scene in order to provide its description.

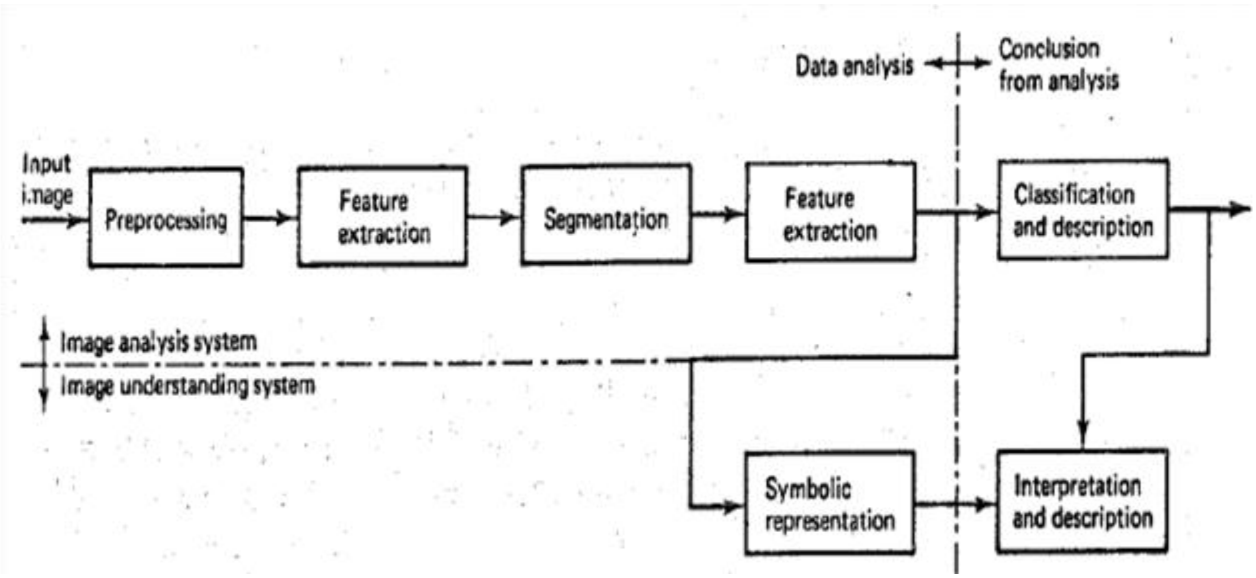
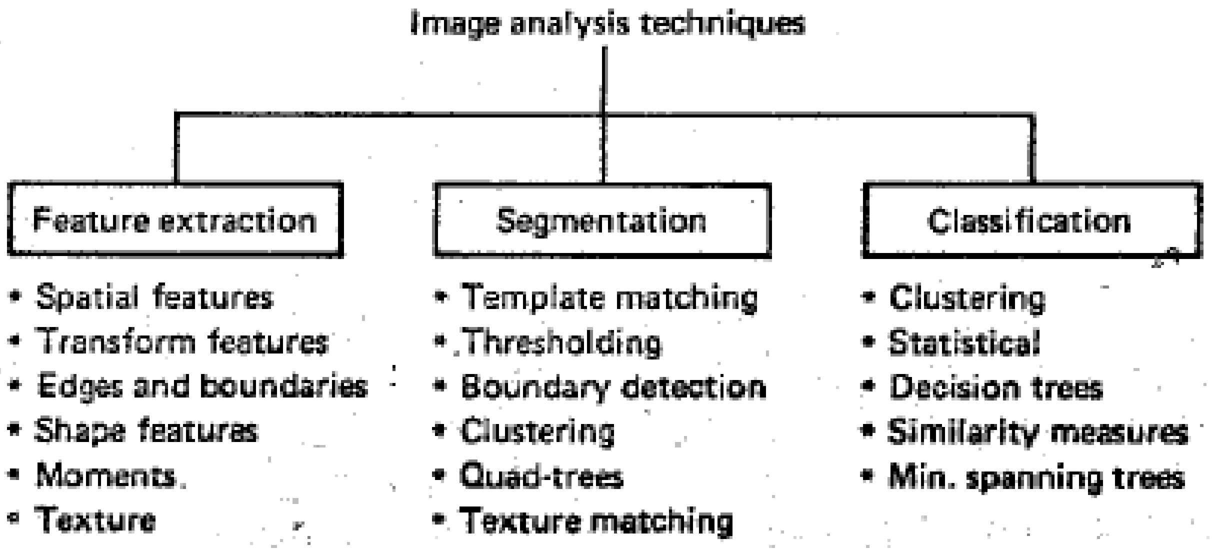


image analysis is quite different from other image processing operations, such as restoration, enhancement, and coding, where the output is another image. Image analysis basically involves the study of *feature extraction*, *segmentation*, and *classification* techniques



SPATIAL FEATURE EXTRACTION

Spatial features of an object may be characterized by its gray levels, their joint probability distributions, spatial distribution, and the like.

Amplitude Features

The simplest and perhaps the most useful features of an object are the amplitudes of its physical properties, such as reflectivity, transmissivity, tristimulus values (color), or multispectral response. For example, in medical X-ray images, the gray-level amplitude represents the absorption characteristics of the body masses and enables discrimination of bones from tissue or healthy tissue from diseased tissue. In infrared (IR) images amplitude represents temperature, which facilitates the segmentation of clouds from terrain In radar images, amplitude represents the *radar cross section,* which determines the size of the object being imaged. Amplitude features can be extracted easily by intensity window slicing or by the more general point transformations

Amplitude features: e.g. the brightness levels can identify regions of interest in the image:

Amplitude features may be discriminative enough if intensity is enough to distinguish wanted info from the rest of the scene

=> defining the best parameters of the transformation for feature extraction – most difficult

=> amplitude feature space representation is not necessarily binary; just that unwanted parts of the scenes should be represented uniquely (i.e. black) in the feature space

=> sometimes adaptive thresholding/adaptive grey scale slicing is needed.

Histogram Features

Histogram features are based on the histogram of a region of the image. Let u be a random variable representing a gray level in a given region of the image. Define

$$p_u(x) \triangleq \text{Prob}[u = x] = \frac{\text{number of pixels with gray level } x}{\text{total number of pixels in the region}}, \quad x = 0, \ldots, L-1$$

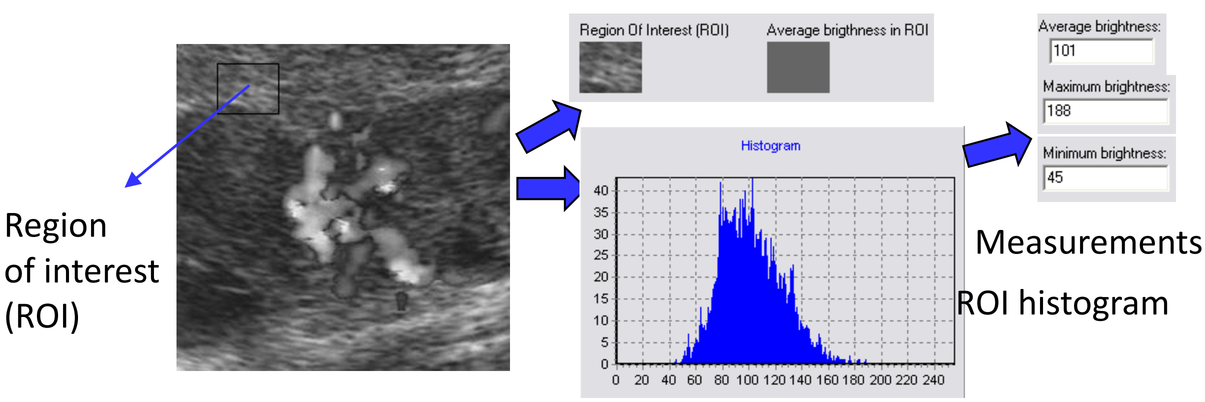
Some of the common histogram features are *dispersion* $= \hat{\mu}_1$, *mean* $= m_1$, *variance* $= \mu_2$, *mean square value or average energy* $= m_2$, *skewness* $= \mu_3$, *kurtosis* $= \mu_4 - 3$. Other useful features are the *median* and the *mode*. A narrow histogram indicates a low contrast region. Variance can be used to measure local activity in the amplitudes. Histogram features are also useful for shape analysis of objects from their projections

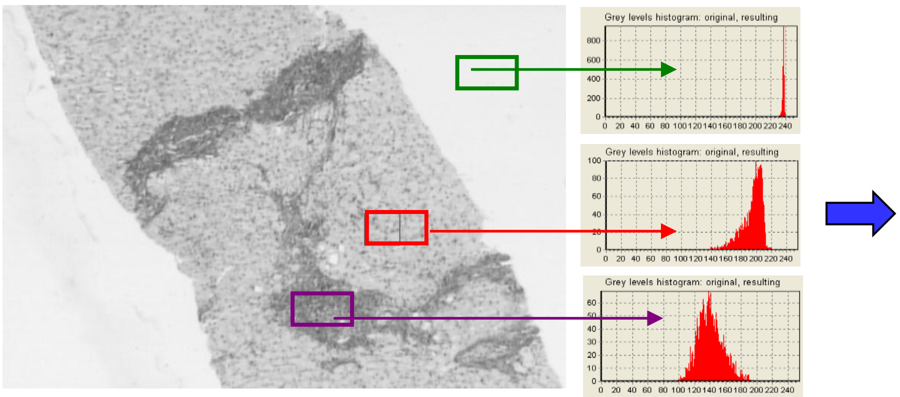
Histogram based features

Local histogram = a local statistical description of the image;

If u = an image pixel; x=a grey level => $p_u(x)$=the probability of appearance of the grey level x in the image region = a value in the **normalized histogram**

=> One can compute: the standard deviation; the entropy; the median; percentiles, of $p_u(x)$.





Tissue of interest is well discriminated from the microscopic slide by the standard deviation of the local histogram

TRANSFORM FEATURES

Transform-feature extraction techniques are also important when the source data originates in the transform coordinates. For example, in optical and optical-digital (hybrid) image analysis applications, the data can be acquired directly in the Fourier domain for real-time feature extraction in the focal plane.

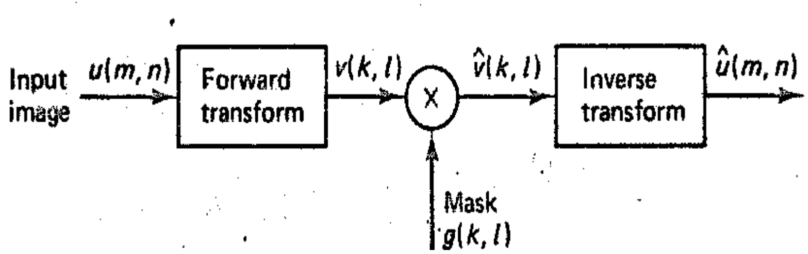


Image transforms provide the frequency domain information in the data. Transform features are extracted by zonal-filtering the image in the selected transform space (Fig. 9.4). The zonal filter, also called the feature mask, is simply a slit or an aperture.

Generally, the high-frequency features can be used for edge and boundary detection, and angular slits can be used for detection of orientation.

A combination of an angular slit with a bandlimited low-pass, band-pass or high-pass filter can be used for discriminating periodic or quasiperiodic textures. Other transforms, such as Haar and Hadamard, are also potentially useful for feature extraction.

EDGE DETECTION

- Edge detection significantly **reduces the amount of data and filters out useless information**, while **preserving the important structural properties in an image**.
- Edges are **boundaries between different textures**.
- Edge also can be defined as **discontinuities in image intensity from one pixel to another**.
- The edges for an image are always the important characteristics that offer an **indication for a higher frequency**.
- Detection of edges for an image may help for **image segmentation, data compression,** and also help for well matching, such as **image reconstruction** and so on.
- Edge detection is **difficult in noisy images**, since both the noise and the edges contain high-frequency content.

Edge Detectors

- Robert
- Sobel
- Prewitt
- Laplacian and
- Canny

Sobel Edge Detector

- The salient features of the sobel edge detectors are listed as follows
- It has two 3x3 convolution kernels or masks, Gx and Gy, as shown in fig 1. both Gx and Gy can be joined together to find the absolute magnitude and the orientation of the gradient.

-1	0	1
2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Fig. 1 3x3 convolution kernels or masks, Gx and Gy

$$|G| = \sqrt{(G_x^{\ 2} + G_y^{\ 2})}$$

- Used to detect edges along the horizontal and vertical axis
- Based on convolving the image with a small, integer valued filter (3×3 kernel) in both horizontal and vertical direction. So this detector requires less computation
- The sobel edge detection masks search for edges in horizontal and vertical directions and then combine this information into a single metric
- In this, image intensity is calculate at every pixel (pixel) and presenting the direction of the maximum possible change from white (light) to black (dark) and the rate of change in that direction.
- Simplicity
- Detection of edges and their orientations

- Sensitivity to noise
- Inaccurate

PREWITT EDGE DETECTOR

The salient features of the Prewitt Edge Detector are listed as follows

- This edge detector is very similar of sobel operator
- Simplicity
- Detection of horizontal and vertical edges and their orientations
- Sensitivity to noise
- Inaccurate
- The kernel used in the Prewitt detector is shown in fig 2.

-1	0	+1
-1	0	+1
-1	0	+1

+1	+1	+1
0	0	0
-1	-1	-1

Fig. 2 Kernel of prewitt detector

LAPLACIAN OF GAUSSIAN EDGE DETECTOR

The salient features of the Laplacian of Gaussian edge detector is listed as follows

- Proposed by Marr and Hildreth in 1980
- This is a **combination of the Gaussian filtering and Laplacian gradient operator**.
- **Laplacian gradient operator determines the regions where the rapid intensity changes**. So it is best suit for edge detection.
- After the laplacian process is over, the **image is given to Gaussian filter to remove the noise pixels.**
- The laplacian gradient of an image is given by

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Where I(x, y) = pixel intensity values in image

- In this, an image is divided where the intensity varies to detect the edges effectively.
- It is very difficult to find the orientation of edges due to laplacian filter.
- Used to determine exact location of edges
- Does not produce good result where the gray level function varies (corners, curves)
- Not useful for finding the orientation of edges

- The two 3X3 kernels used for laplacian edge detector is shown in fig 3.

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$Y = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

Fig. 3 kernel of laplacian edge detector

ROBERTS EDGE DETECTOR

The salient features of the Roberts Edge Detector is listed as follows

- This detector has two 2x2 convolution kernels in which one of the kernels is rotated by 90^0.
- Fast computation
- Performs a simple and fast two dimensional spatial gradient measurement on an image.
- Each point (pixels) in the output image represents the expected absolute magnitude of the spatial gradient of the input Image at that point.

- Convolution mask is shown in fig 4.

+1	0
0	-1

0	+1
-1	0

Fig. 4 kernel of Roberts's edge detector

$$\Delta f = grad(f) = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The Magnitude of the vector is defined as

$$M(x, y) = \sqrt{(G_x{}^2 + G_y{}^2)}$$

CANNY EDGE DETECTOR

- The canny edge detector can be used to identify a wide range of real edges in images.
- The detector eliminates the unwanted noise pixels by the process of smoothening edges in an image because noise pixels generate false edges. \
- In this edge detection, the signal to noise ratio is improved as compared to other methods.
- This is the reason why this detector is extensively used for edge detection in image processing.

The procedure to find edges in images is explained as follows.

- **Initially the image is smoothened using a suitable filter** such as mean filter, Gaussian filter etc., **to reduce the effect of noise.**
- Then **local gradient and edge direction is calculated for every point**. This point has a maximum strength in the direction of the gradient.
- These edge **points give rise to ridges** in the gradient magnitude image.

- The **edge detector tracks along the top of these ridges and make all the pixels to zero that are not actually on the top of the ridge. So a thin line is generated in the output**.

    - These ridge pixels are threshold using two threshold values: upper threshold (T2) and lower threshold (T1).
    - Ridge pixels are classified as strong edge pixels if ridge pixel values are greater than upper threshold (T2) and ridge pixels are classified as weak edge pixels if ridge pixel values are between the lower threshold (T1) and the upper threshold (T2).
    - Finally, the edges in the image are linked by integrating the weak pixels which are connected to the strong pixels.

EDGE DETECTORS EXAMPLES

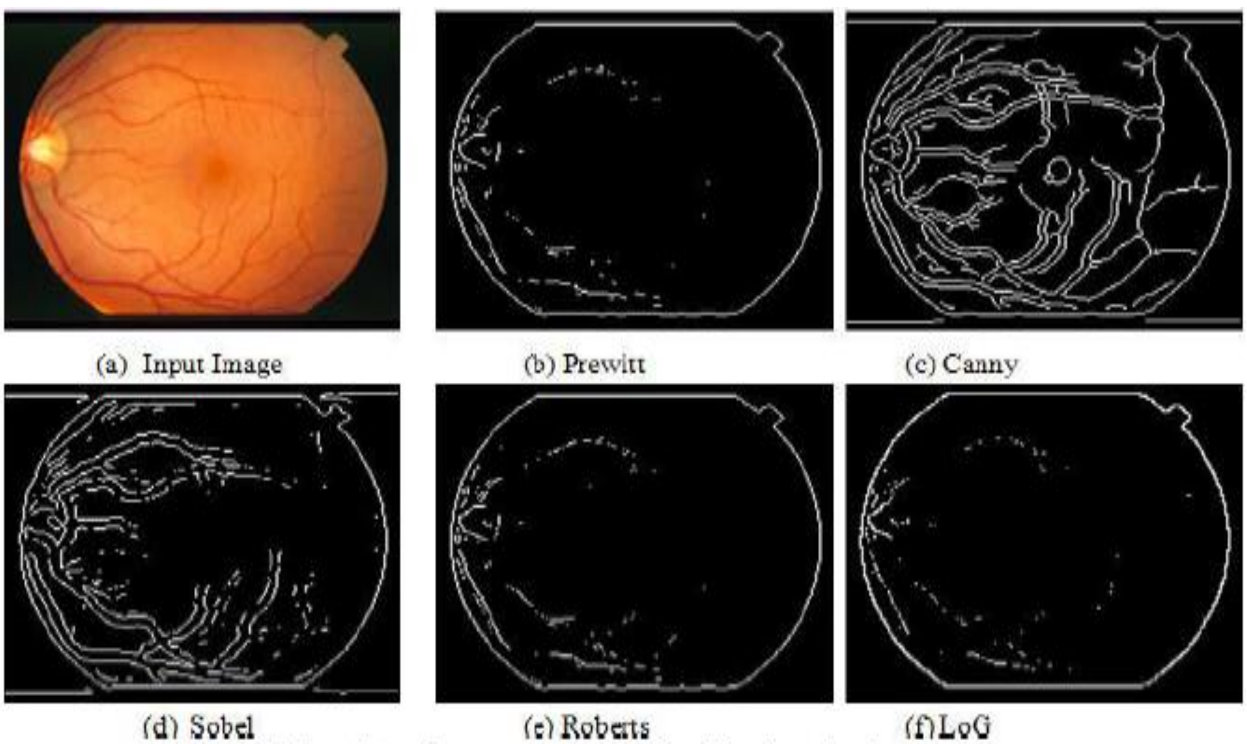


Fig. 6.Edge detection result for medical (retina fundus.jpg) image

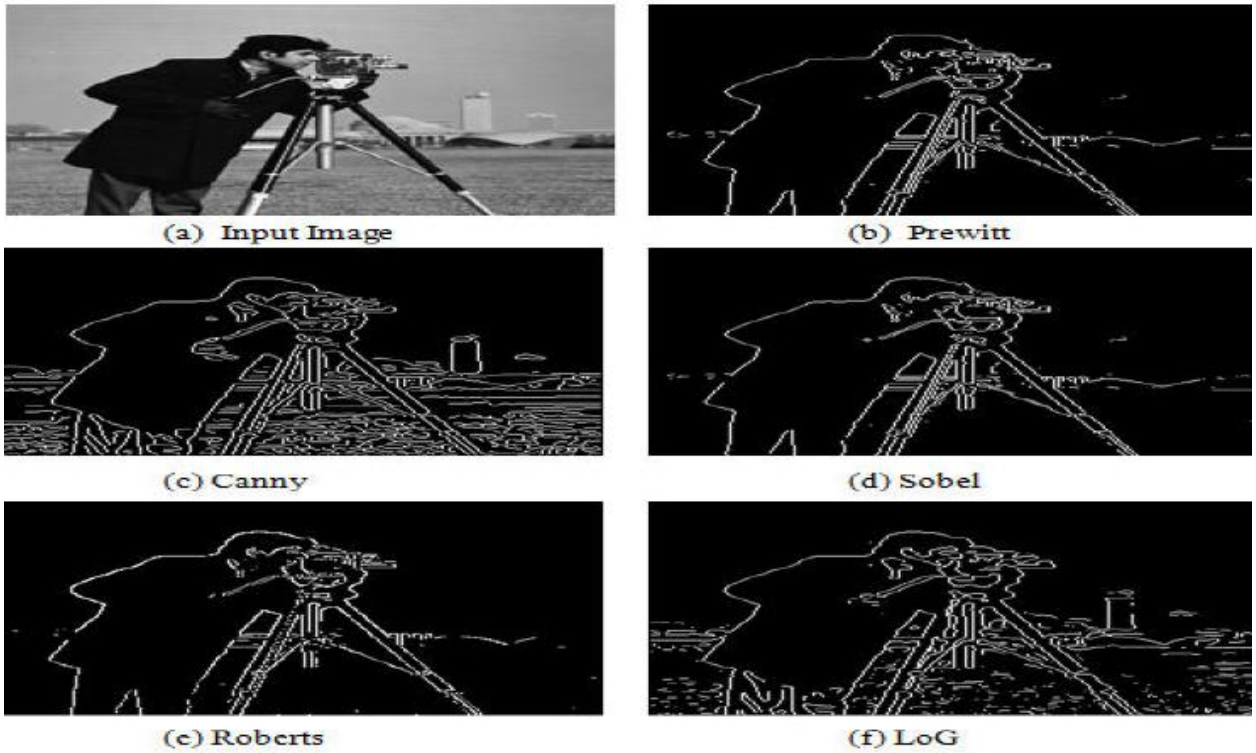


Fig. 7.Edge detection result for cameraman.tiff image

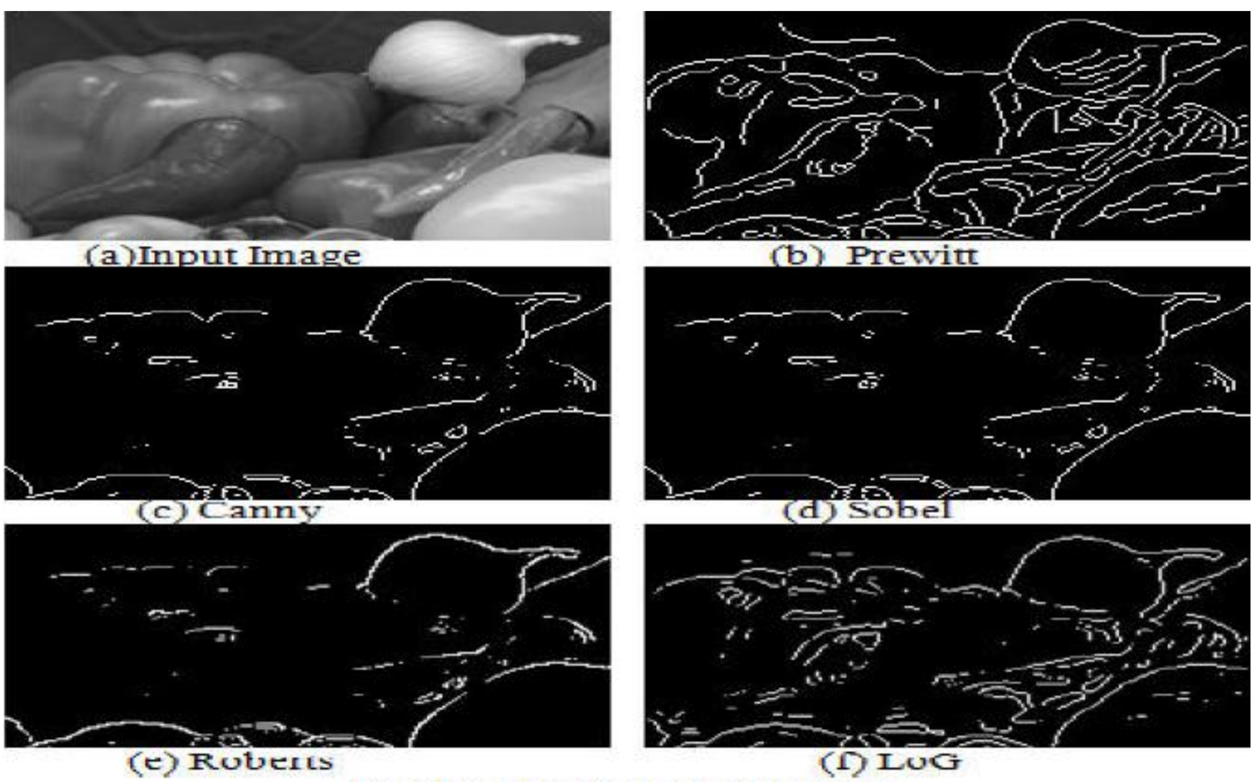


Fig. 8.Edge detection result for onion.png

COMPARISON OF EDGE DETECTORS

Edge Detector	Method	Advantages	Limitations
Roberts	Gradient Based	1. Easy and simple computation. 2. Edges are detected along with their orientation	1. These are more sensitive to noise. 2. Detection of edges is inaccurate 3. Less reliable
Sobel	Gradient Based	1. Easy and simple computation. 2. Edges are detected along with their orientation	1. These are more sensitive to noise. 2. Detection of edges is inaccurate 3. Less reliable
Prewitt	Gradient Based	1. Easy and simple computation. 2. Edges are detected along with their orientation	1. These are more sensitive to noise. 2. Detection of edges is inaccurate 3. Less reliable
Canny	Gaussian Based	1. Improved signal to noise ratio. 2. Suitable for noisy images i.e., more sensitive to noisy pixels 3. Accurate	1. Slow and Complex 2. False zero crossing.
LoG	Gradient Based	1. The detection of edges and their orientation is simple due to approximation of gradient magnitude is simple. 2. The characteristics are fixed in all directions. 3. Testing wide area around the pixel is possible.	1. Malfunctioning at the corners, curves and where the gray level intensity function varies 2.The magnitude of edges degrades as noise increases
DWT	Wavelet based	1.More accurate than other methods 2. Less computation	1.application oriented 2.complicated as compared to traditional methods

BOUNDARY EXTRACTION AND REPRESENTATION

Boundaries are linked edges that characterize the shape of an object. They are useful in computation of geometry features such as size or orientation.

Connectivity

Conceptually, boundaries can be found by tracing the connected edges. On a rectangular grid a pixel is said to be *four-* or *eight-connected* when it has the same properties as one of its nearest four or eight neighbors, respectively (Fig. 9.14).

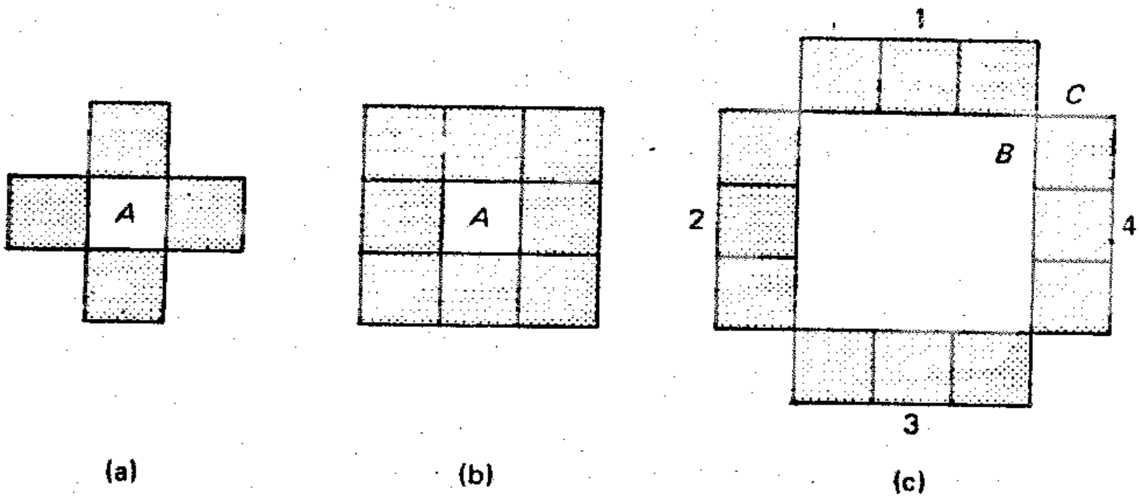
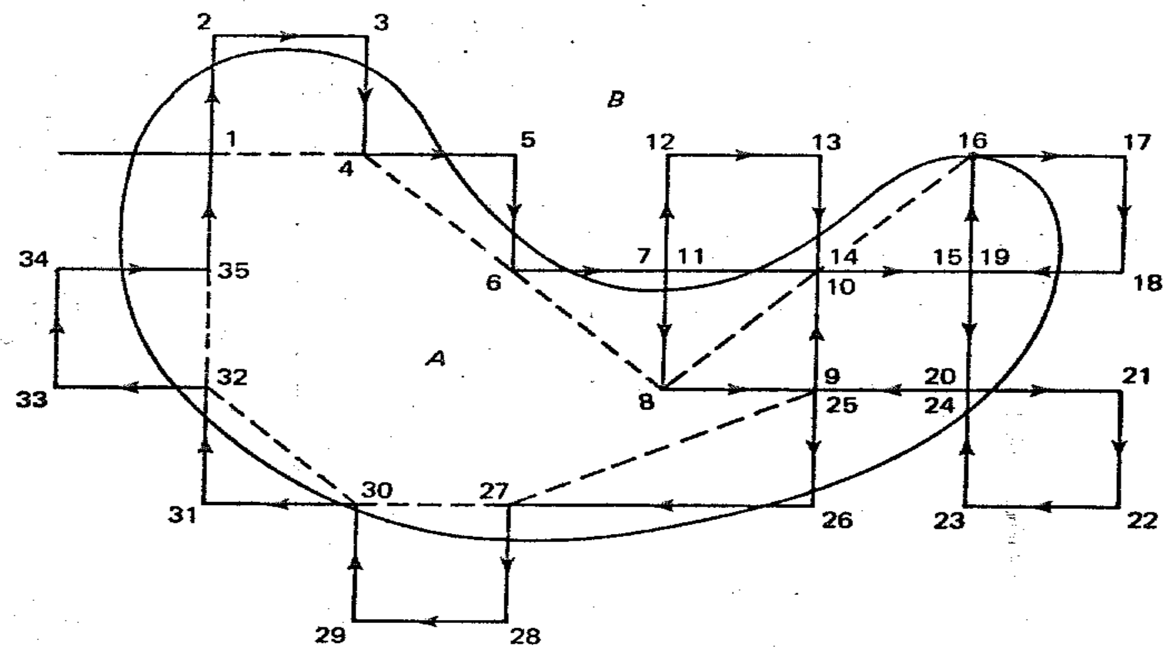


Figure 9.14 Connectivity on a rectangular grid. Pixel A and its (a) 4-connected and (b) 8-connected neighbors; (c) connectivity paradox: "Are B and C connected?"

A modified version of this contour-following method is called the *crack-following algorithm*. In that algorithm each pixel is viewed as having a square-shaped boundary, and the object boundary is traced by following the edge-pixel boundaries.

As the name suggests, contour-following algorithms trace boundaries by ordering successive edge points. A simple algorithm for tracing closed boundaries in binary images is shown in Fig. . This algorithm can yield a coarse contour, with some of the boundary pixels appearing twice.



Algorithm

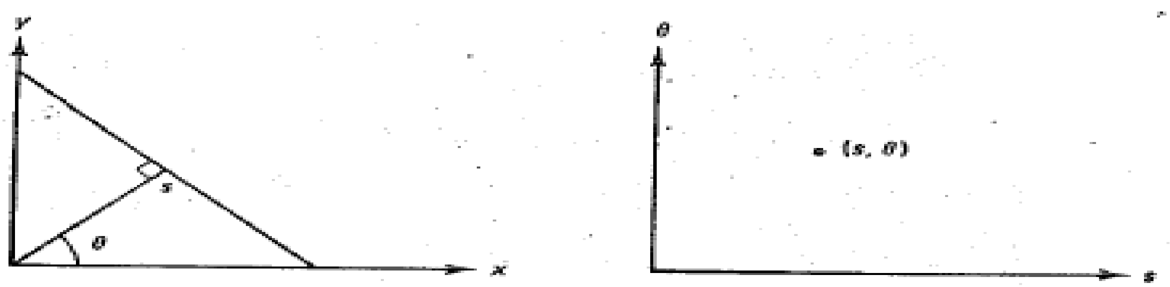
1. Start inside A (e.g., 1)
2. Turn left and step to next pixel if in region A, (e.g., 1 to 2), otherwise turn right and step (e.g., 2 to 3)
3. Continue until arrive at starting point 1

Figure 9.15 Contour following in a binary image.

Boundary Extraction and Representation using Hough Transform

A straight line at a distance s and orientation θ (Fig. 9.19a) can be represented as

$$s = x \cos\theta + y \sin\theta$$



(a) Straight line

(b) Hough transform

Figure The Hough transform.

The Hough transform of this line is just a point in the (s, θ) plane; that is, all the points on this line map into a single point This fact can be used to detect straight lines in a given set of boundary points. Suppose we are given boundary points (x_i, y_i), $i = 1, \ldots, N$. For some chosen quantized values of parameters s and θ, map each (x_i, y_i) into the (s, θ) space and count $C(s, \theta)$, the number of edge points that map into the location (s, θ), that is, set

$$C(s_k, \theta_l) = C(s_k, \theta_l) + 1, \quad \text{if} \quad x_i \cos\theta + y_i \sin\theta = s_k \quad \text{for} \quad \theta = \theta_l$$

Then the local maxima of $C(s, \theta)$ give the different straight line segments through the edge points.

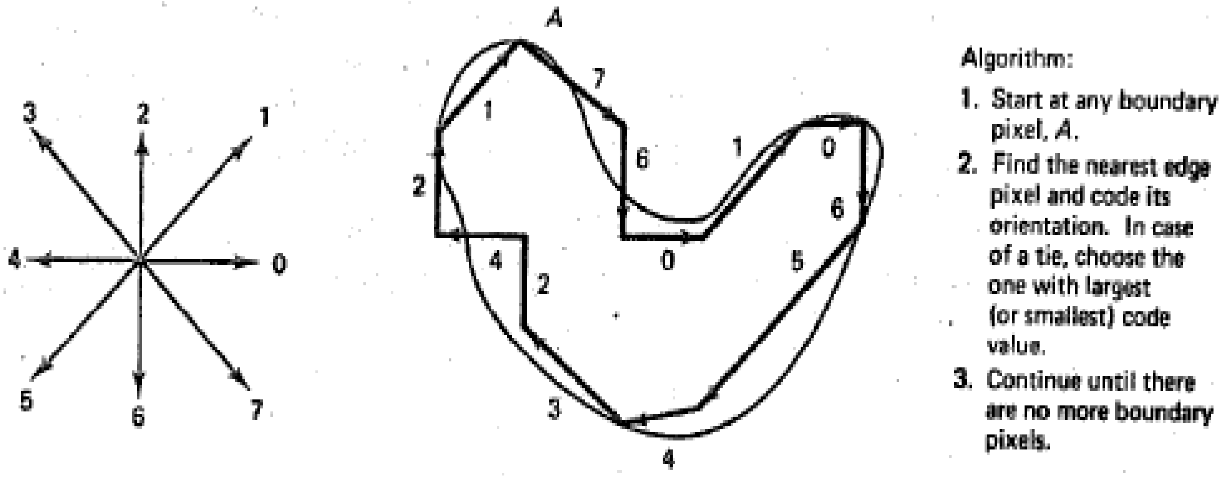
Boundary Representation

Proper representation of object boundaries is important for analysis and synthesis of shape. Shape analysis is often required for detection and recognition of objects in a scene. Shape synthesis is useful in computer-aided design (CAD) of parts and assemblies, image simulation applications such as video games, cartoon movies,

environmental modeling of aircraft-landing testing and training, and other computer graphics problems.

Chain Codes

In chain coding the direction vectors between successive boundary pixels are encoded. For example, a commonly used chain code employs eight directions, which can be coded by 3-bit code words. Typically, the chain code contains the start pixel address followed by a string of code words. Such codes can be generalized by increasing the number of allowed direction vectors between successive boundary pixels.



(b) Contour

Boundary pixel orientations: (A), 76010655432421

Chain code: A 111 110 000 001 000 110 101 101 110 011 010 100 010 001

Figure Chain code for boundary representation.

Fourier Descriptors

Once the boundary trace is known, we can consider it as a pair of waveforms $x(t), y(t)$. Hence any of the traditional one-dimensional signal representation techniques can be used. For any sampled boundary we can define

$$u(n) \triangleq x(n) + jy(n), \qquad n = 0, 1, \dots, N-1 \tag{9.49}$$

which, for a closed boundary, would be periodic with period N. Its DFT representation is

$$\begin{aligned} u(n) &\triangleq \frac{1}{N}\sum_{k=0}^{N-1} a(k)\exp\left(\frac{j2\pi kn}{N}\right), \qquad 0 \le n \le N-1 \\ a(k) &\triangleq \sum_{n=0}^{N-1} u(n)\exp\left(\frac{-j2\pi kn}{N}\right), \qquad 0 \le k \le N-1 \end{aligned} \tag{9.50}$$

The complex coefficients $a(k)$ are called the *Fourier descriptors* (FDs) of the boundary. For a continuous boundary function, $u(t)$, defined in a similar manner to (9.49), the FDs are its (infinite) Fourier series coefficients. Fourier descriptors have been found useful in character recognition problems [32].

Boundary matching. The Fourier descriptors can be used to match similar shapes even if they have different size and orientation. If $a(k)$ and $b(k)$ are the FDs of two boundaries $u(n)$ and $v(n)$, respectively, then their shapes are similar if the distance

$$d(u_0, \alpha, \theta_0, n_0) \triangleq \min_{u_0, \alpha, n_0, \theta_0} \left\{ \sum_{n=0}^{N-1} \left| u(n) - \alpha v(n + n_0) e^{j\theta_0} - u_0 \right|^2 \right\} \qquad (9.54)$$

is small. The parameters u_0, α, n_0, and θ_0 are chosen to minimize the effects of translation, scaling, starting points and rotation, respectively.

TEXTURE-BASED TECHNIQUES

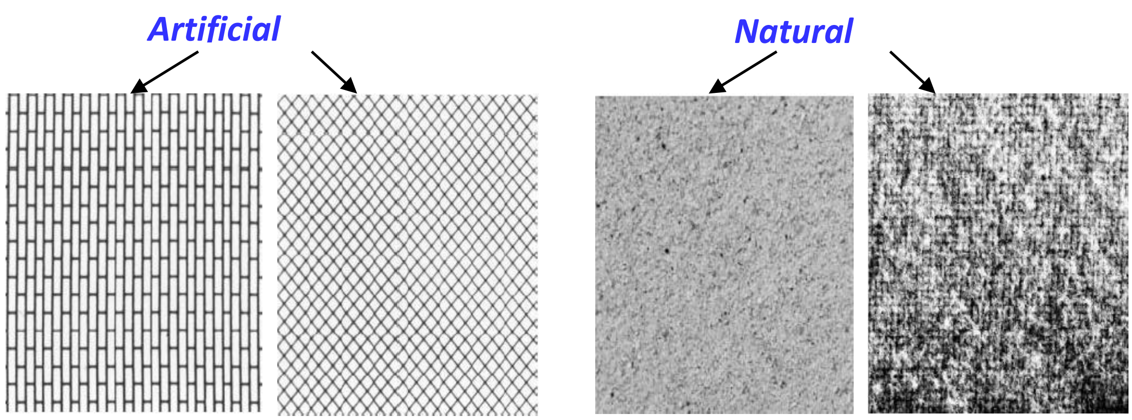
What is Texture?

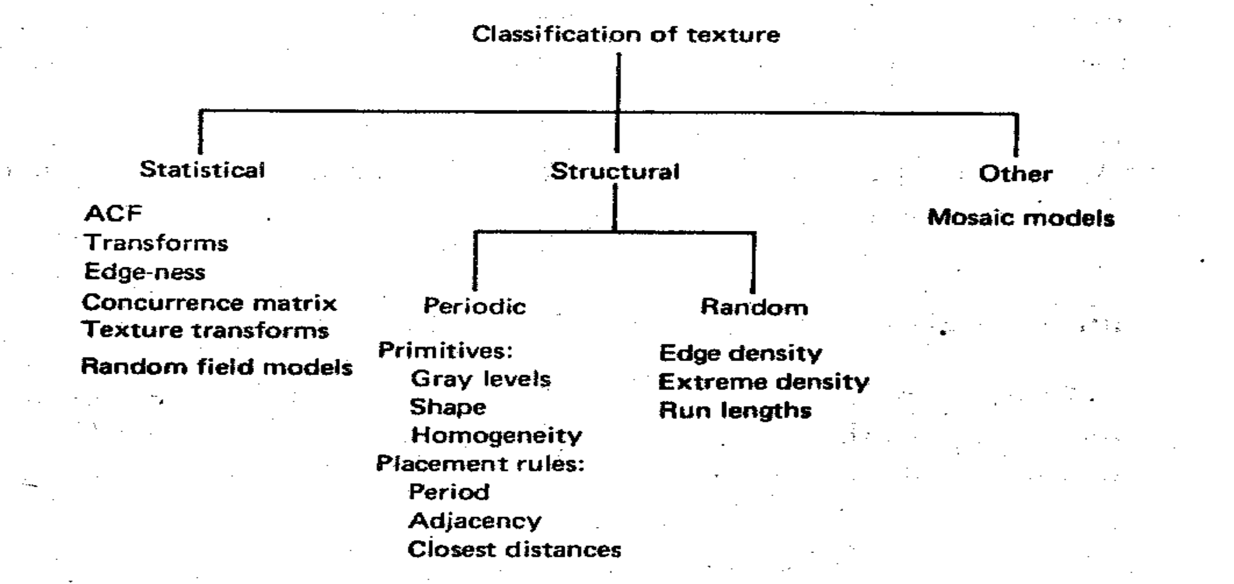
No one exactly knows.

In the visual arts, **texture** is the perceived surface quality of an artwork.

- "Texture" is an ambiguous word and in the context of texture synthesis may have one of the following meanings:
- 1. In common speech, **"texture" used as a synonym for "surface structure".**
- **Texture has been described by five different properties in the psychology of perception: *coarseness*, *contrast*, *directionality*, *line-likeness* and *roughness* [1].**
- 2. In 3D computer graphics, a texture is a digital image applied to the surface of a three-dimensional model by texture mapping to give the model a more realistic appearance. Often, the image is a photograph of a "real" texture, such as wood grain.
- 3. In image processing, **every digital image composed of repeated elements is called a "texture."**

- **Stochastic textures.** Texture images of stochastic textures look like noise: colour dots that are randomly scattered over the image, barely specified by the attributes minimum and maximum brightness and average colour. Many textures look like stochastic textures when viewed from a distance. An example of a stochastic texture is roughcast.
- · **Structured textures.** These textures look like somewhat regular patterns. An example of a structured texture is a stonewall or a floor tiled with paving stones.





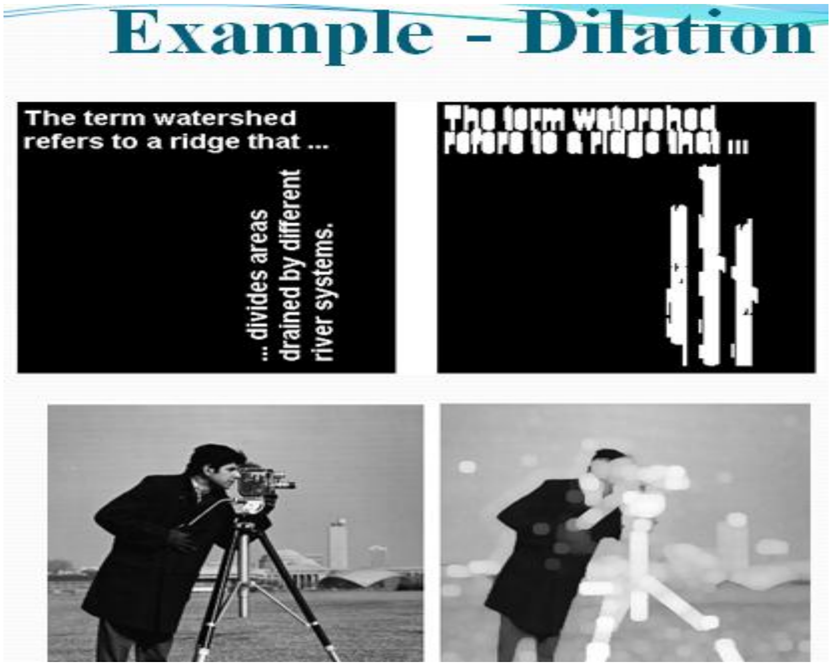
STRUCTURE

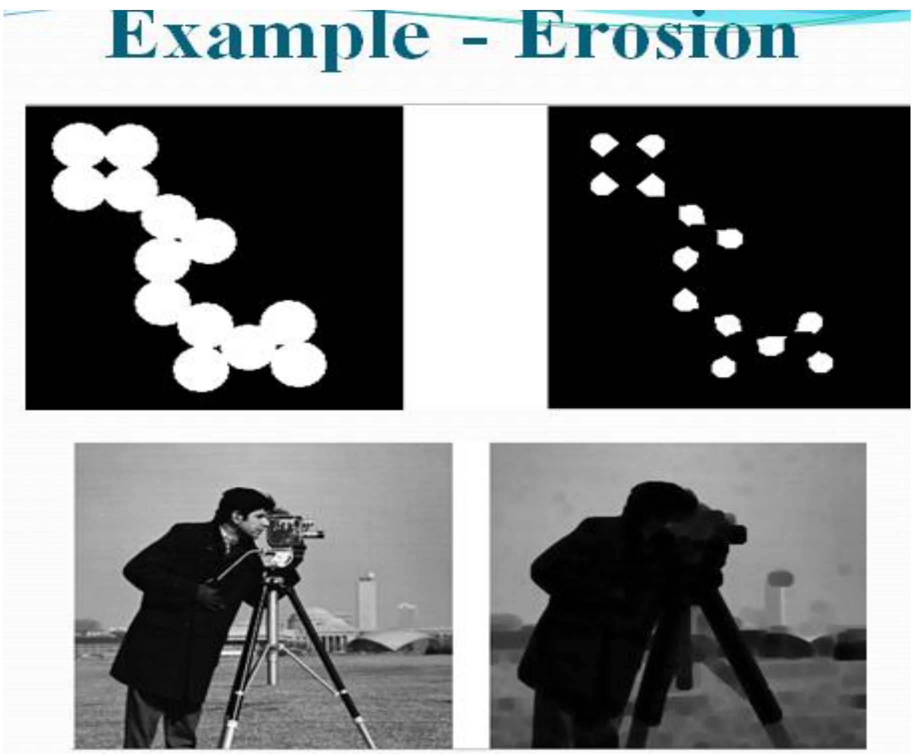
In many computer vision applications, the objects in a scene can be characterized satisfactorily by structures composed of line or arc patterns. Examples include handwritten or printed characters, fingerprint ridge patterns, chromosomes and biological cell structures, circuit diagrams and engineering drawings, and the like.

MORPHOLOGICAL OPERATIONS

- Morphology is a broad set of image processing operations that process images based on shapes.
- Morphological operations apply a structuring element to an input image, creating an output image of the same size.
- The most basic morphological operations are dilation and erosion.
- In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

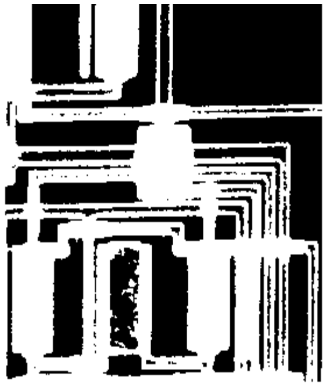
- **Dilation and erosion** are two fundamental morphological operations.
- **Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.**
- The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image.
- In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image.

- 

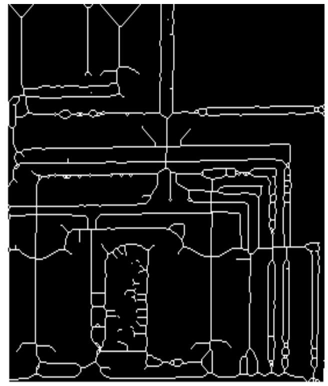


Skeletonization

- To reduce all objects in an image to lines, without changing the essential structure of the image, use the bwmorph function. This process is known as skeletonization.



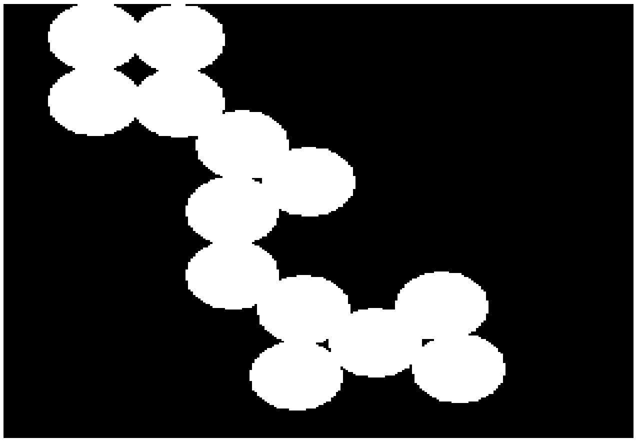
Original Image



Skeletonization of Image

Dilation- and Erosion-Based Functions

Function	Morphological Definition
bwhitmiss	Logical AND of an image, eroded with one structuring element, and the image's complement, eroded with a second structuring element.
imbothat	Subtracts the original image from a morphologically closed version of the image. Can be used to find intensity troughs in an image.
imclose	Dilates an image and then erodes the dilated image using the same structuring element for both operations.
imopen	Erodes an image and then dilates the eroded image using the same structuring element for both operations.
imtophat	Subtracts a morphologically opened image from the original image. Can be used to enhance contrast in an image.

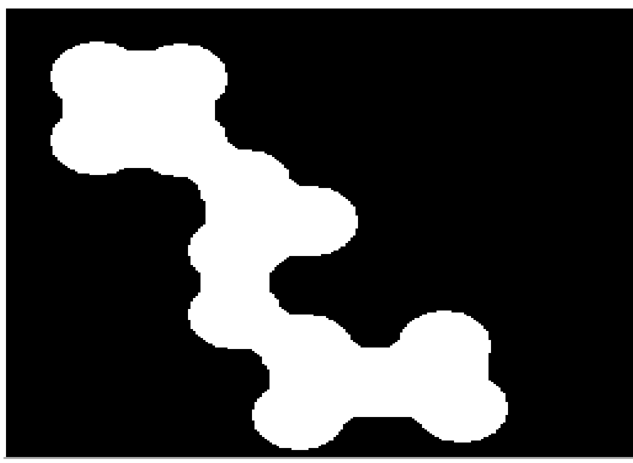


Create a disk-shaped structuring element. Use a disk radius of 10 pixels so that the largest gap gets filled.

```
se = strel('disk',10);
```

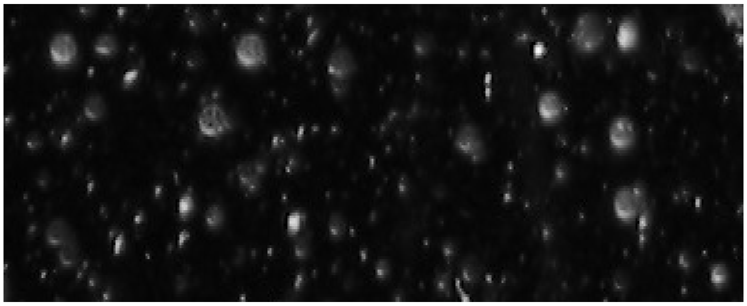
Perform a morphological close operation on the image

```
closeBW = imclose(originalBW,se);
figure, imshow(closeBW)
```



1. Read the image into the MATLAB workspace and display it.

```
I = imread('snowflakes.png');
imshow(I)
```



2. Create a disk-shaped structuring element with a radius of 5 pixels.

```
se = strel('disk',5);
```

3. Remove snowflakes having a radius less than 5 pixels by opening

```
I_opened = imopen(I,se);
figure, imshow(I_opened,[])
```

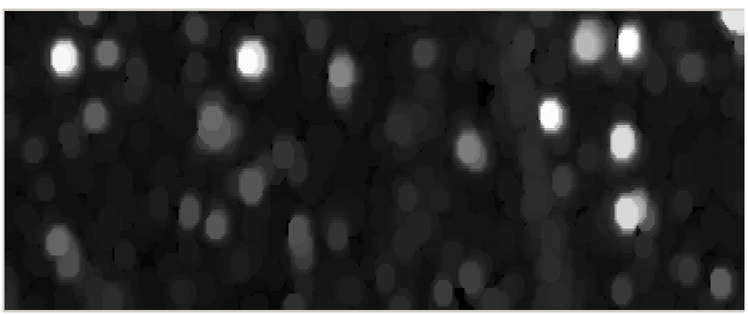


TABLE 9.10 Some Useful Morphological Transforms

Operation	Definition	Properties & Usage
Hit-Miss	$\mathbf{X} \circledast \mathbf{B} = (\mathbf{X} \ominus \mathbf{B}_{ob})/(\mathbf{X} \oplus \mathbf{B}_{bk})$	Searching for a match or a specific configuration. $\mathbf{B}_{ob}$: set formed from pixels in $\mathbf{B}$ that should belong to the object. $\mathbf{B}_{bk}$: . . . background.
Open	$\mathbf{X}_B = (\mathbf{X} \ominus \mathbf{B}) \oplus \mathbf{B}$	Smooths contours, suppress small islands and sharp caps of $\mathbf{X}$. Ideal for object size distribution study.
Close	$\mathbf{X}^B = (\mathbf{X} \oplus \mathbf{B}) \ominus \mathbf{B}$	Blocks up narrow channels and thin lakes. Ideal for the study of inter object distance.
Boundary	$\partial\mathbf{X} = \mathbf{X}/\mathbf{X} \ominus \mathbf{G}$	Gives the set of boundary points.
Convex Hull	$\mathbf{X}_1^1 = \mathbf{X}$ $\mathbf{X}_{i+1}^1 = (\mathbf{X}_i^1 \circledast \mathbf{B}^1)$ $\mathbf{X}_{CH} = \bigcup_{j=1}^{4} \mathbf{X}_{\infty}^j$	$\mathbf{B}^1, \mathbf{B}^2, \ldots$ are rotated versions of the structuring element $\mathbf{B}$. $\mathbf{C}$ is an appropriate structuring element choice for $\mathbf{B}$.
Skeleton	$S(\mathbf{X}) = \bigcup_{n=0}^{n_{\max}} s_n(x)$ $= \bigcup_{n=0}^{n_{\max}} [(\mathbf{X} \ominus n\mathbf{G})/(\mathbf{X} \ominus n\mathbf{G})_G]$ $\mathbf{X} = \bigcup_{n=0}^{n_{\max}} [s_n(x) \oplus n\mathbf{G}]$	$n_{\max}$: max size after which $\mathbf{X}$ erodes down to an empty set. The skeleton is a regenerative representation of the object.
Thin	$\mathbf{X} \bigcirc \mathbf{B} = \mathbf{X}/\mathbf{X} \circledast \mathbf{B}$ $\mathbf{X} \bigcirc \{\mathbf{B}\} = ((\ldots((\mathbf{X} \bigcirc \mathbf{B}^1) \bigcirc \mathbf{B}^2) \ldots) \bigcirc \mathbf{B}^n)$	To symmetrically thin $\mathbf{X}$ a sequence of structuring elements, $\{\mathbf{B}\} = \{\mathbf{B}^i, 1 \le i \le n\}$, is used in cascade, where $\mathbf{B}^i$ is a rotated version of $\mathbf{B}^{i-1}$. A widely used element is $\mathbf{L}$.
Thick	$\mathbf{X} \odot \mathbf{B} = \mathbf{X} \cup \mathbf{X} \circledast \mathbf{B}$	Dual of thinning.
Prune	$\mathbf{X}_1 = \mathbf{X} \bigcirc \{\mathbf{B}\}$ $\mathbf{X}_2 = \bigcup_{j=1}^{8} (\mathbf{X}_1 \circledast \mathbf{P}^j)$ $\mathbf{X}_{pn} = \mathbf{X}_1 \cup [(\mathbf{X}_2 \oplus \{\mathbf{G}\}) \cap \mathbf{X}]$	$\mathbf{E}$ is a suitable structuring element. $\mathbf{X}_2$: end points $\mathbf{X}_{pn}$: pruned object with Parasite branches suppressed.

The symbols "/" and " ∪ " represent the set difference and the set union operations

SCENE MATCHING AND DETECTION

A problem of much significance in image analysis is the detection of change or presence of an object in a given scene. Such problems occur in remote sensing for monitoring growth patterns of urban areas, weather prediction from satellite images, diagnosis of disease from medical images, target detection from radar images, and automation using robot vision, and the like. Change detection is also useful in alignment or spatial registration of two scenes imaged at different instants or using different sensors. For example, a large object photographed in small overlapping sections can be reconstructed by matching the overlapping parts.

Image Subtraction

- Subtract one image from another or subtract constant from image
- Syntax Z= imsubtract(X,Y)

- Description
- Z = imsubtract (X,Y) subtracts each element in array Y from the corresponding element in array X and returns the difference in the corresponding element of the output array Z.

Examples

Subtract two `uint8` arrays. Note that negative results are rounded to `0`.

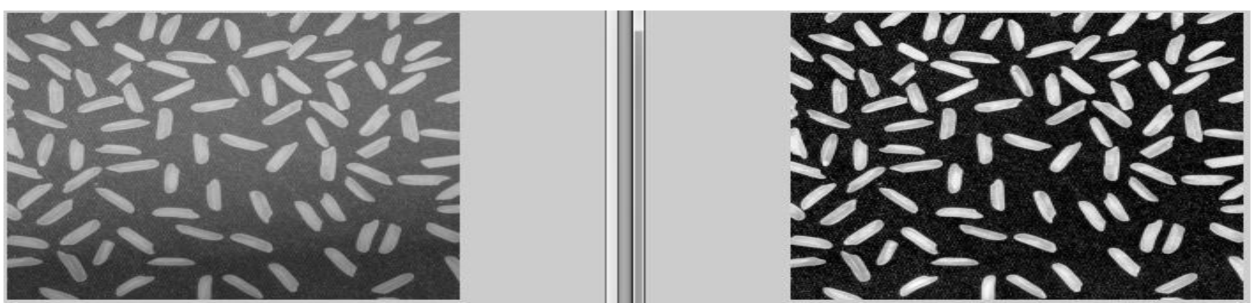
```
X = uint8([ 255 10 75; 44 225 100]);
Y = uint8([ 50 50 50; 50 50 50 ]);
Z = imsubtract(X,Y)
Z =

   205     0    25
     0   175    50
```

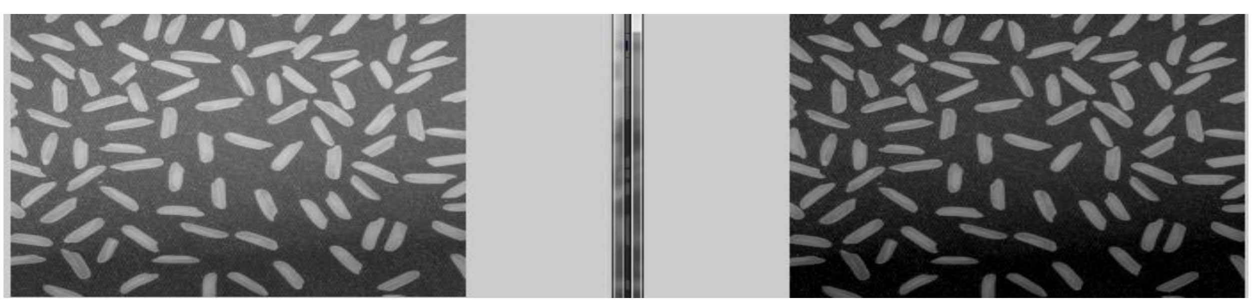
Estimate and subtract the background of an image:

```
I = imread('rice.png');
background = imopen(I,strel('disk',15));
Ip = imsubtract(I,background);
imshow(Ip,[])
```

Subtract a constant value from an image:

```
I = imread('rice.png');
Iq = imsubtract(I,50);
figure, imshow(I), figure, imshow(Iq)
```





SCENE MATCHING AND DETECTION

Template Matching and Area Correlation

The presence of a known object in a scene can be detected by searching for the location of match between the object template $u(m,n)$ and the scene $v(m,n)$. Template matching can be conducted by searching the displacement of $u(m,n)$, where the mismatch energy is minimum. For a displacement (p,q), we define the mismatch energy

$$\begin{aligned}\sigma_\eta^2(p,q) &\triangleq \sum_m \sum_n [v(m,n) - u(m-p,n-q)]^2 \\ &= \sum_m \sum_n |v(m,n)|^2 + \sum_m \sum_n |u(m,n)|^2 - 2\sum_m \sum_n v(m,n)u(m-p,n-q)\end{aligned} \tag{9.122}$$

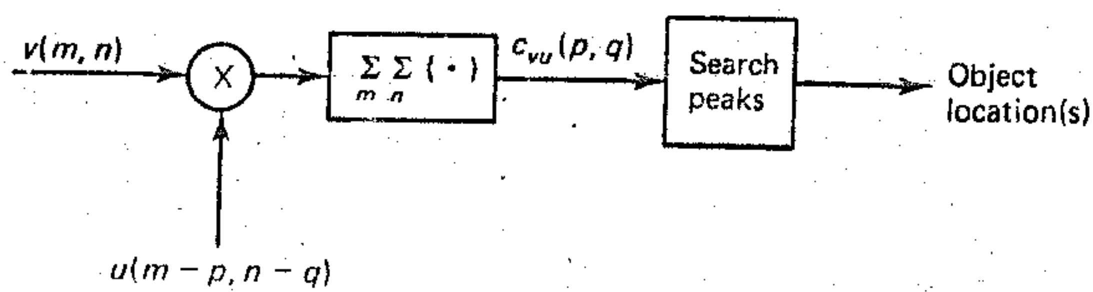


IMAGE SEGMENTATION

There are many definitions:

- Segmentation subdivides an image into its constituent regions or objects (Gonzales, pp567)
- Segmentation is a process of grouping together pixels that have similar attributes (Efford , pp250)

    - Image Segmentation is the process of partitioning an image into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous (Pal, pp1277)
- Segmentation is typically associated with pattern recognition problems. It is considered the first phase of a pattern recognition process and is sometimes also referred to as object isolation.

Why segmentation is difficult?

It can be difficult for many reasons:

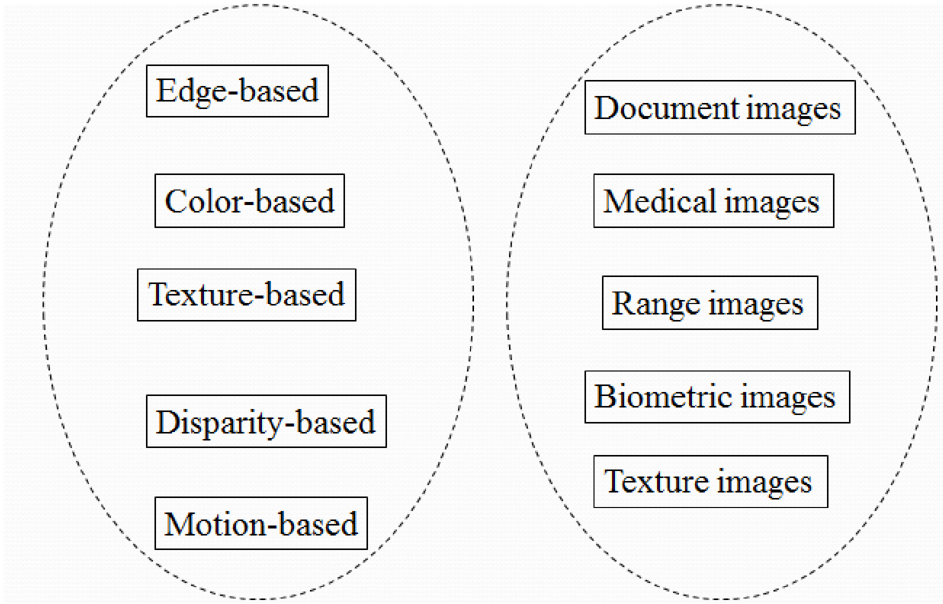
- Non- uniform illumination
- No control of the environment
- Inadequate model of the object of interest
- Noise

Why segmentation is useful?

Segmentation algorithms have been used for a variety of applications. Some examples are :

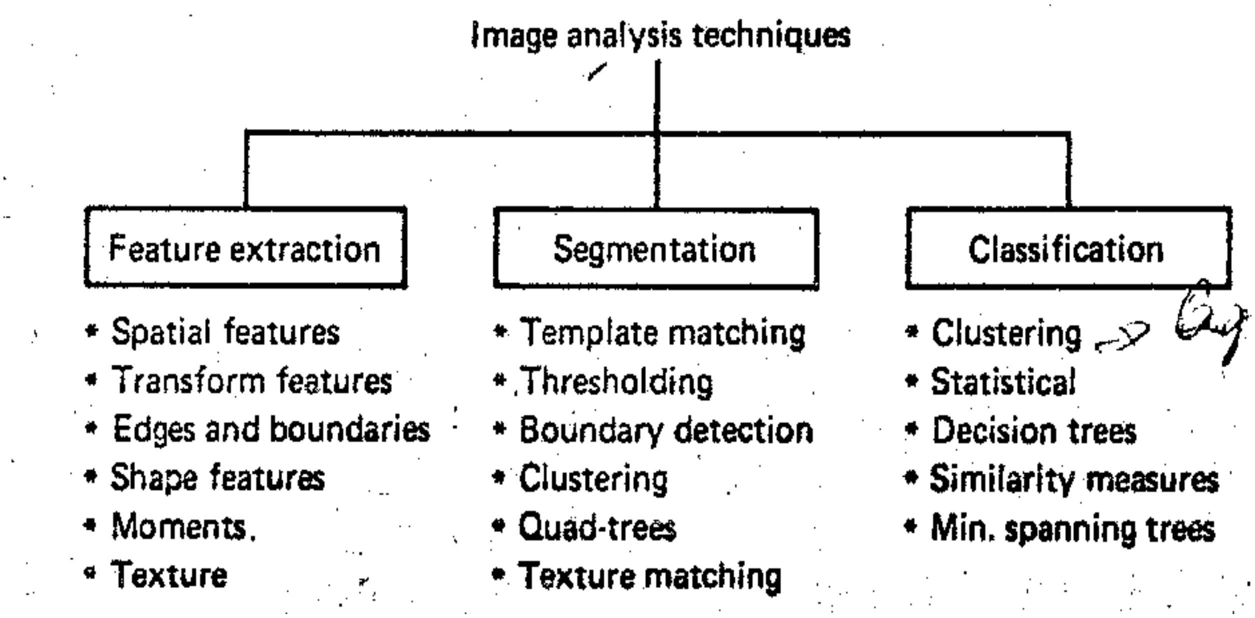
- Optical character recognition(OCR)
- Automatic Target Acquisition
- Colorization of Motion Pictures
- Detection and measurement of bone, tissue, etc, in medical images.

Overview of Segmentation Techniques



CLASSIFICATION TECHNIQUES

A major task after feature extraction is to classify the object into one of several categories. Figure 9.2 lists various classification techniques applicable in image



Supervised Learning

Supervised learning, also called supervised classification, can be *distribution free* or *statistical.* Distribution-free methods do not require knowledge of any a priori probability distribution functions and are based on reasoning and heuristics. Statistical techniques are based on probability distribution models, which may be parametric (such as Gaussian distributions) or nonparametric.

It should be mentioned that classification and segmentation processes have closely related objectives. Classification can lead to segmentation, and vice-versa. Classification of pixels in an image is another form of component labeling that can result in segmentation of various objects in the image. For example, in remote sensing, classification of multispectral data at each pixel location results in segmentation of various regions of wheat, barley, rice, and the like. Similarly, image segmentation by template matching, as in character recognition, leads to classification or identification of each object.

There are two basic approaches to classification, *supervised* and *nonsupervised,* depending on whether or not a set of prototypes is available.

Distribution-free classification. Suppose there are K different objects or pattern classes $S_1, S_2, \ldots, S_k, \ldots, S_K$. Each class is characterized by M_k prototypes, which have $N \times 1$ feature vectors $\mathbf{y}_m^{(k)}, m = 1, \ldots, M_k$. Let $\mathbf{x}$ denote an $N \times 1$ feature vector obtained from the observed image. A fundamental function in pattern recognition is called the *discriminant function.* It is defined such that the kth discriminant function $g_k(\mathbf{x})$ takes the maximum value if $\mathbf{x}$ belongs to class k, that is, the decision rule is

$$g_k(\mathbf{x}) > g_i(\mathbf{x}) \qquad k \neq i \Leftrightarrow \mathbf{x} \in S_k \tag{9.138}$$

Decision tree classification [60–61]. Another distribution-free classifier, called a *decision tree classifier,* splits the N-dimensional feature space into unique regions by a sequential method. The algorithm is such that every class need not be tested to arrive at a decision. This becomes advantageous when the number of classes is very large. Moreover, unlike many other training algorithms, this algorithm is guaranteed to converge whether or not the feature space is linearly separable.

Statistical classification. In statistical classification techniques it is assumed the different object classes and the feature vector have an underlying joint probability density. Let $P(S_k)$ be the a priori probability of occurrence of class S_k and $p(\mathbf{x})$ be the probability density function of the random feature vector observed as $\mathbf{x}$.

Bayes' minimum-risk classifier. The Bayes' minimum-risk classifier minimizes the average loss or risk in assigning $\mathbf{x}$ to a wrong class. Define

Similarity measure approach. The success of clustering techniques rests on the partitioning of the feature space into cluster subsets. A general clustering algorithm is based on split and merge ideas (Fig. 9.61). Using a *similarity measure,* the input vectors are partitioned into subsets. Each partition is tested to check whether or not the subsets are sufficiently distinct. Subsets that are not sufficiently distinct are merged. The procedure is repeated on each of the subsets until no further subdivisions result or some other convergence criterion is satisfied. Thus, a similarity measure, a distinctiveness test, and a stopping rule are required to define a clustering algorithm. For any two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, some of the commonly used similarity measures are:

$$\text{Dot product:} \quad \langle \mathbf{x}_i, \mathbf{x}_j \rangle \triangleq \mathbf{x}_i^T \mathbf{x}_j = \|\mathbf{x}_i\| \, \|\mathbf{x}_j\| \cos(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{Similarity rule:} \quad S(\mathbf{x}_i, \mathbf{x}_j) \triangleq \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - \langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$

$$\text{Weighted Euclidean distance:} \quad d(\mathbf{x}_i, \mathbf{x}_j) \triangleq \sum_k [x_i(k) - x_j(k)]^2 w_k$$

$$\text{Normalized correlation:} \quad \rho(\mathbf{x}_i, \mathbf{x}_j) \triangleq \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\sqrt{\langle \mathbf{x}_i, \mathbf{x}_i \rangle \langle \mathbf{x}_j, \mathbf{x}_j \rangle}}$$

Nonsupervised Learning or Clustering

In nonsupervised learning, we attempt to identify *clusters* or natural groupings in the feature space. A cluster is a set of points in the feature space for which their local density is large (relative maximum) compared to the density of feature points in the surrounding region. Clustering techniques are useful for image segmentation and for classification of raw data to establish classes and prototypes. Clustering is also a useful vector quantization technique for compression of images.

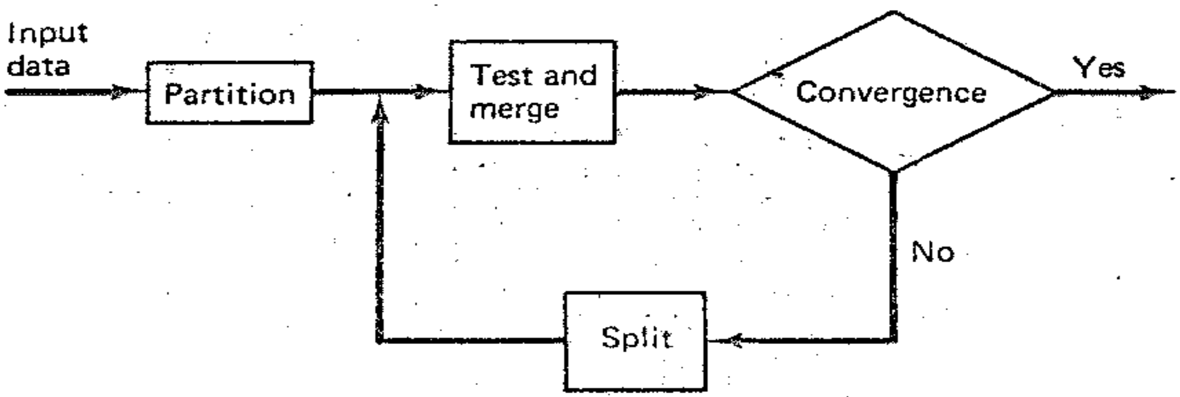


Figure 9.61 A clustering approach.

Chain method The first data sample is designated as the *representative* of the first cluster and similarity or distance of the next sample is measured from the first cluster representative. If this distance is less than a threshold, say η, then it is placed in the first cluster; otherwise it becomes the representative of the second cluster. The process is continued for each new data sample until all the data has been exhausted. Note that this is a one-pass method.

Other Methods

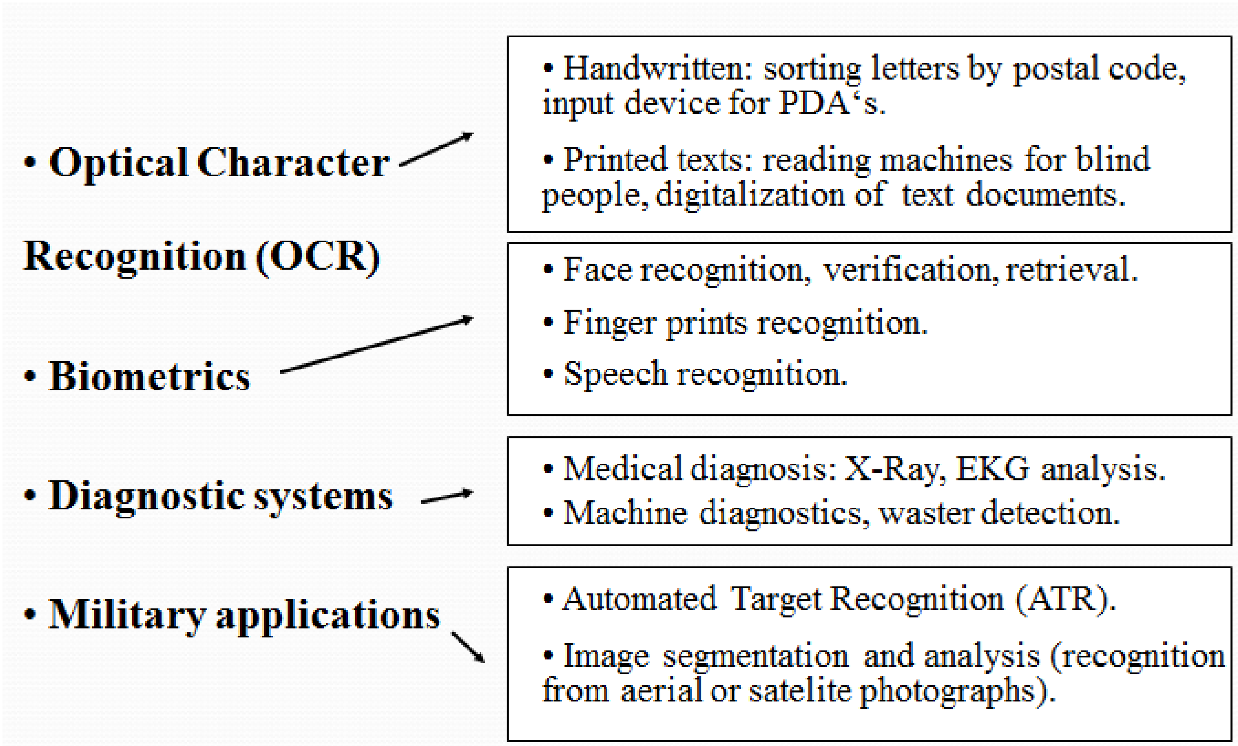
Clusters can also be viewed as being located at the nodes of the joint Nth-order histogram of the feature vector. Other clustering methods are based on statistical nonsupervised learning techniques, ranking, and intrinsic dimensionality determination, graph theory, and so on.

PATTERN RECOGNITION

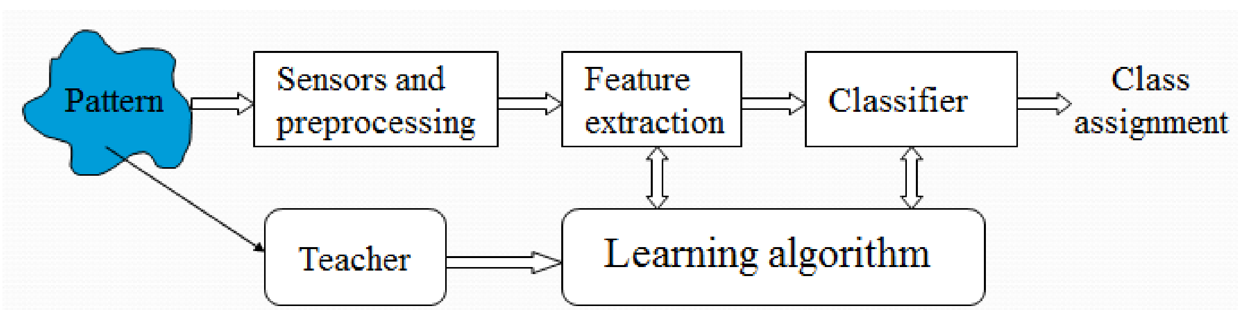
"The assignment of a physical object or event to one of several prespecified categeries" -- Duda & Hart

- A **pattern** is an object, process or event that can be given a name.
- A **pattern class** (or category) is a set of patterns sharing common attributes and usually originating from the same source.
- During **recognition** (or **classification**) given objects are assigned to prescribed classes.
- A **classifier** is a machine which performs classification.

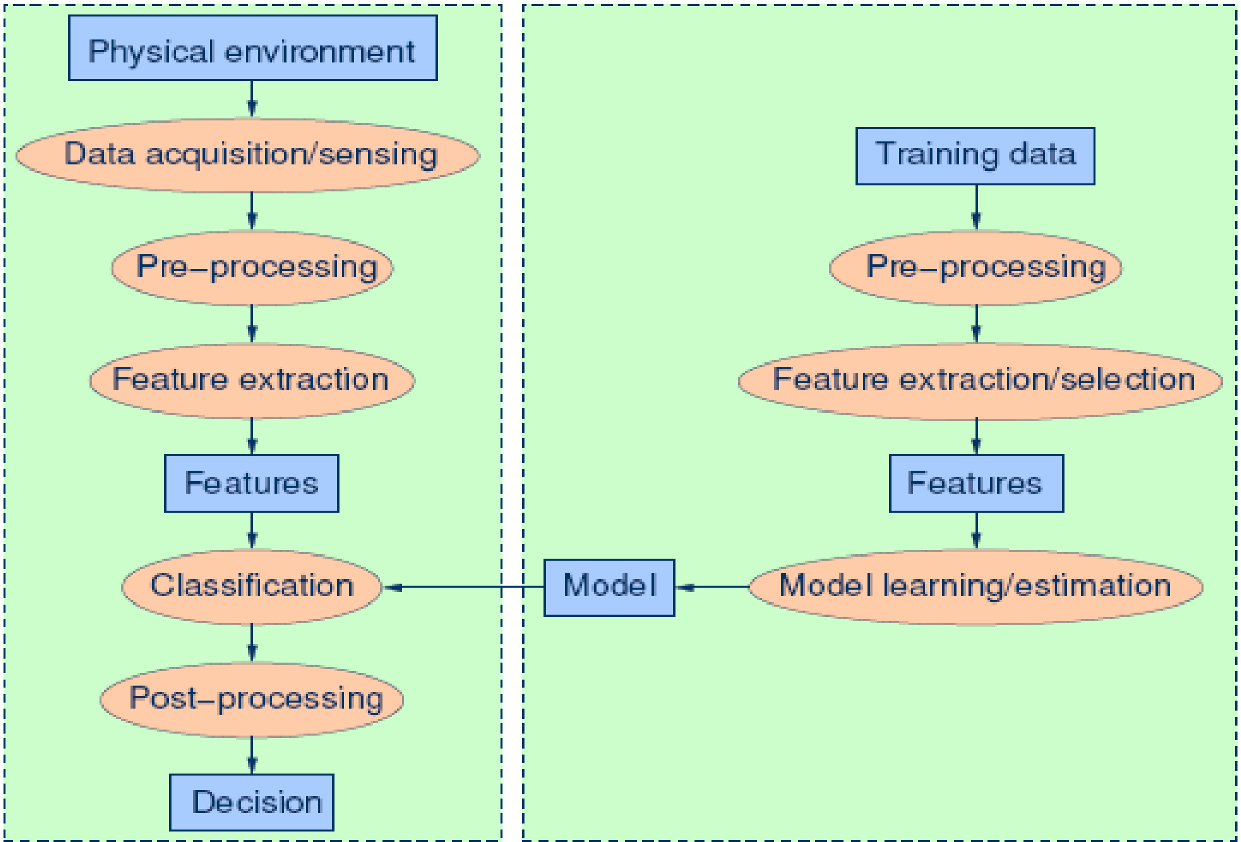
Applications



Components of PR system



- **Sensors and preprocessing**.
- **A feature extraction** aims to create discriminative features good for classification.
- **A classifier**.
- **A teacher** provides information about hidden state -- supervised learning.
- **A learning algorithm** sets PR from training examples.



Basic components of a pattern recognition system

- **Data acquisition and sensing**
- **Pre-processing**
  - Removal of noise in data.
  - Isolation of patterns of interest from the background.
- **Feature extraction**
  - Finding a new representation in terms of features.

(Better for further processing)

- **Model learning and estimation**
  - Learning a mapping between features and pattern groups.
- **Classification**
  - Using learned models to assign a pattern to a predefined category
- **Post-processing**
  - Evaluation of confidence in decisions.

- Exploitation of context to improve performances.

Examples of pattern recognition applications

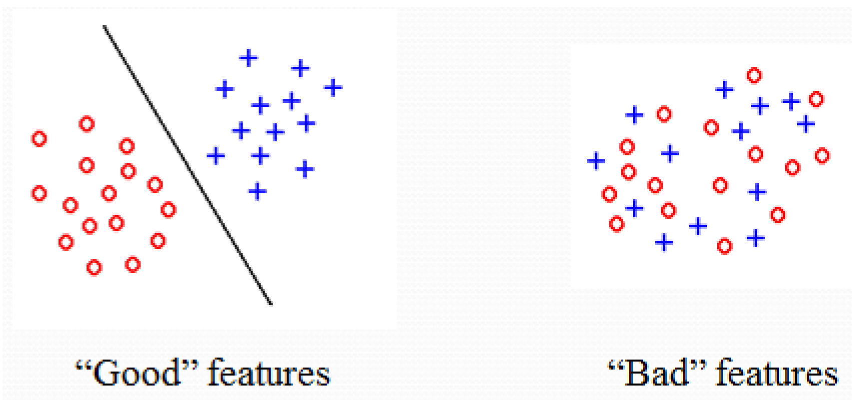
Problem Domain	Application	Input Pattern	Pattern Classes
Document image analysis	Optical character recognition	Document image	Characters, words
Document classification	Internet search	Text document	Semantic categories
Document classification	Junk mail filtering	Email	Junk/non-junk
Multimedia database retrieval	Internet search	Video clip	Video genres
Speech recognition	Telephone directory assistance	Speech waveform	Spoken words
Natural language processing	Information extraction	Sentences	Parts of speech
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Medical	Computer aided diagnosis	Microscopic image	Cancerous/healthy cell
Military	Automatic target recognition	Optical or infrared image	Target type
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective/non-defective product
Industrial automation	Fruit sorting	Images taken on a conveyor belt	Grade of quality
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories
Bioinformatics	Sequence analysis	DNA sequence	Known types of genes
Data mining	Searching for meaningful patterns	Points in multidimensional space	Compact and well-separated clusters

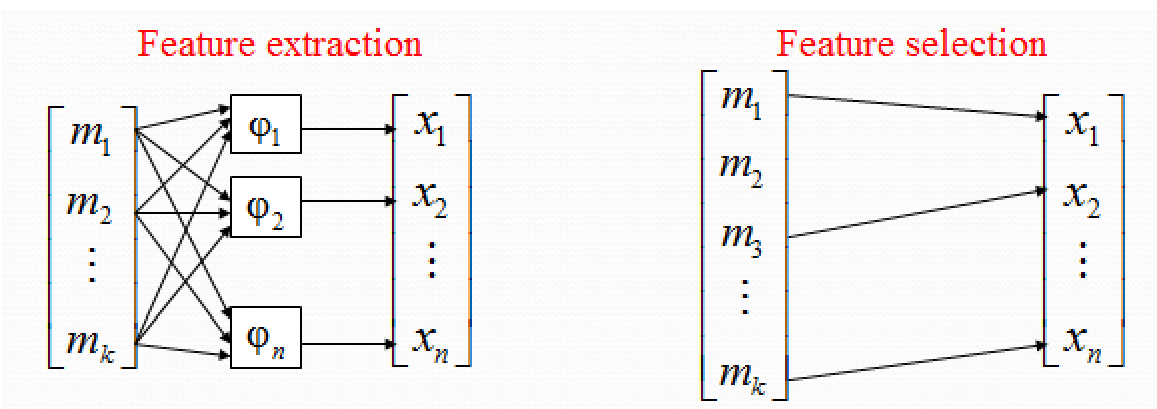
Feature extraction

Task: to extract features which are good for classification.

Good features:

- Objects from the same class have similar feature values.
- Objects from different classes have different values.





Problem can be expressed as optimization of parameters of featrure extractor .

Supervised methods: objective function is a criterion of separability (discriminability) of labeled examples, e.g., linear discriminat analysis (LDA).

Unsupervised methods: lower dimesional representation which preserves important characteristics of input data is sought for, e.g., principal component analysis (PCA).

Advantages

- economical to store features in memory
- experimental evidence consistent with features

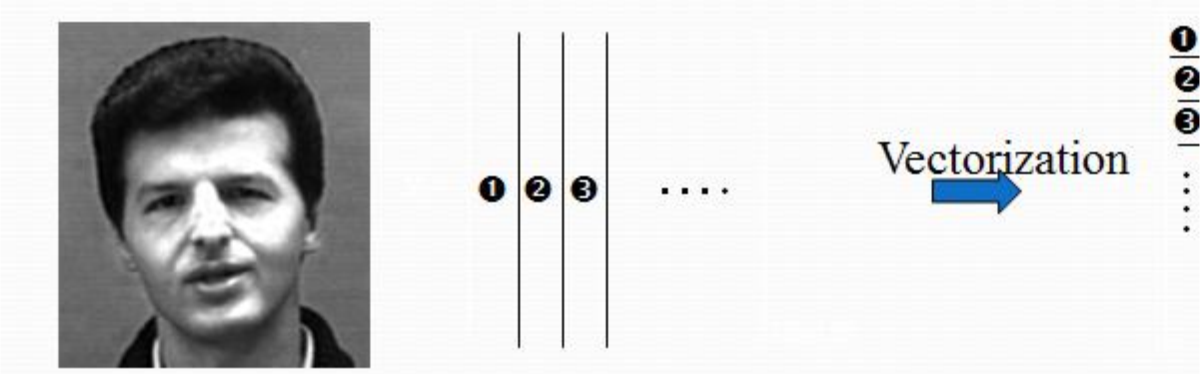
Disadvantages

- lack of applicability to a wide range of stimuli
- analysis of stimuli does not always begin with features
- treats all features as equivalent

Face Recognition

- User-friendly pattern recognition application
- Weakness of face recognition
  - Illumination problems
  - Pose problems(profile or frontal view)

- Eigenspace-based approach
  - A holistic approach
- Reducing the high dimensionality problem , and large computational complexity.



A face image of size 200×180

Approaches

- **Statistical PR:** based on underlying statistical model of patterns and pattern classes.
- **Structural (or syntactic) PR:** pattern classes represented by means of formal structures as grammars, automata, strings, etc.
- **Neural networks:** classifier is represented as a network of cells modeling neurons of the human brain (connectionist approach).
- Template matching is simple to implement but the template size must be small to decrease computational delay.
- Statistical methods highly depend on the assumption of distribution.
- Neural networks can adaptively refine the classifier and the decision surface in principle can be arbitrarily implemented.
- Syntactic methods concerned structural sense to encode but additional process to define primitives is required.

2 Mark Questions

1. What is meant by spatial feature extraction?
2. Write notes on Edge detection.
3. List out the various edge detectors.
4. Write notes on Boundary extraction and representation
5. What is meant by Structure and Texture?
6. Write notes on Scene matching and detection.
7. What is meant by Image segmentation?
8. What is meant by supervised and unsupervised classification?
9. What is meant by Pattern recognition?
10. List out the applications of Pattern recognition.

12 Mark Questions

1. Explain about various spatial feature extraction methods.
2. Explain about edge detection techniques.
3. Explain about Boundary extraction and representation.
4. Explain (a) Scene matching and detection (b) Image segmentation
5. Explain about various classification techniques
6. Explain Pattern recognition with a neat sketch.