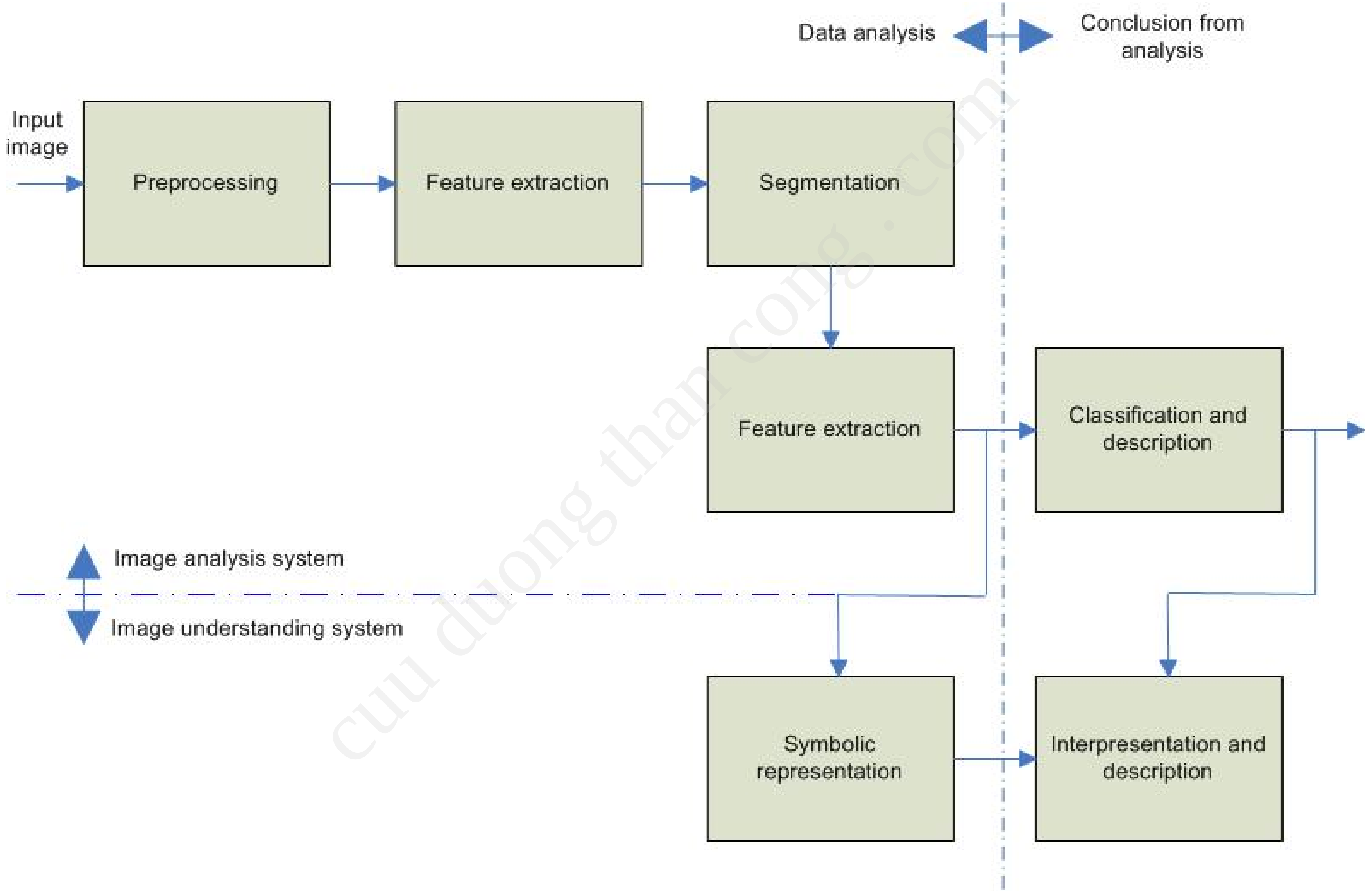
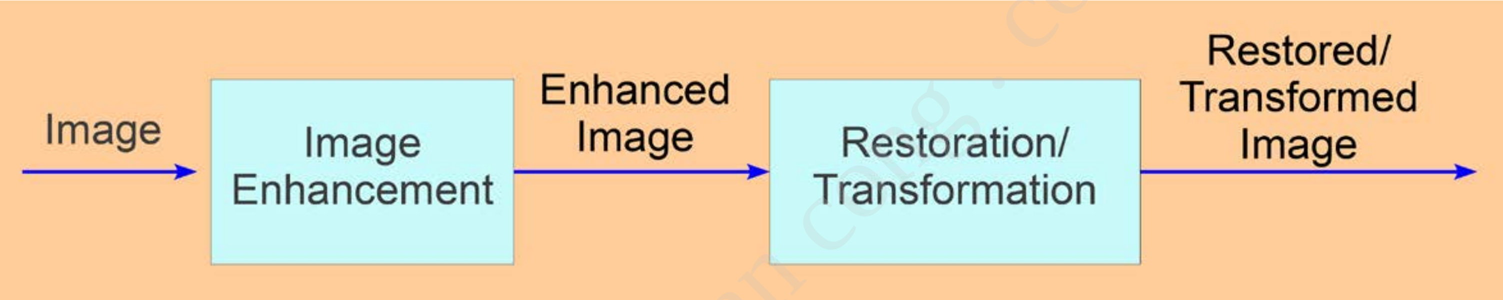
Chapter 6:
Image Representation and Description

6. Image Analysis

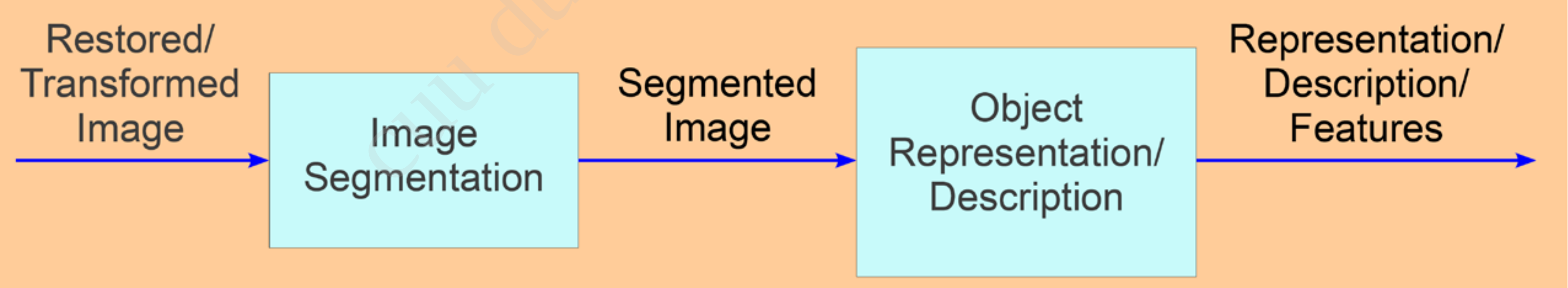


6. Image Representation & Description (IRD) (1)

- **Low-level image processing**: Image enhancement, restoration, transformation, etc.

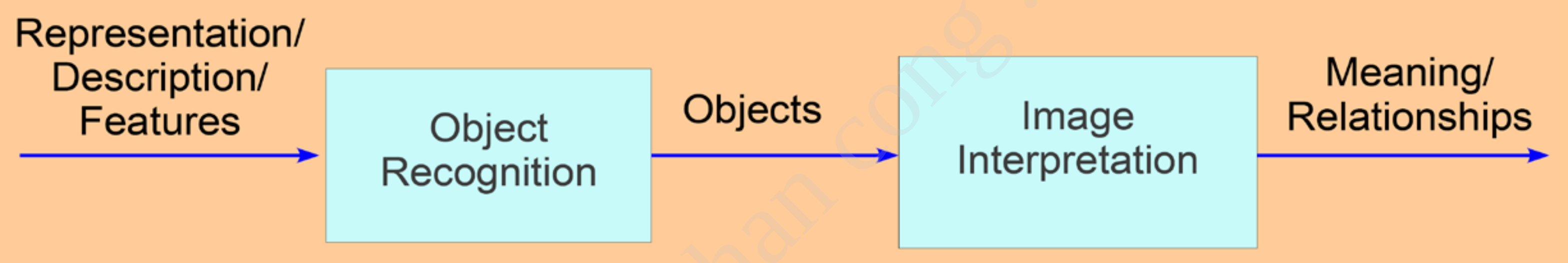


- **Mid-level image processing (image understanding)**: Object representation, description.



6. Image Representation & Description (2)

- **High-level image processing** (**recognition** and **interpretation**): Object recognition, interpretation of object relationships.



6. Image Representation & Description (3)

- **Representation** used to make the data useful to a computer (further process: description). Representing region in 2 ways:
  - in terms of its external characteristics (its boundary) ⇨ focus on shape characteristics.
  - in terms of its internal characteristics (its region) ⇨ focus on regional properties, e.g., color, texture,…

  Sometimes, we may need to use both ways.
- **Description** describes the region based on the chosen representation. E.g.:
  - representation ⇨ boundary
  - description ⇨ length of the boundary, orientation of the straight line joining its extreme points, and the number of concavities in the boundary.

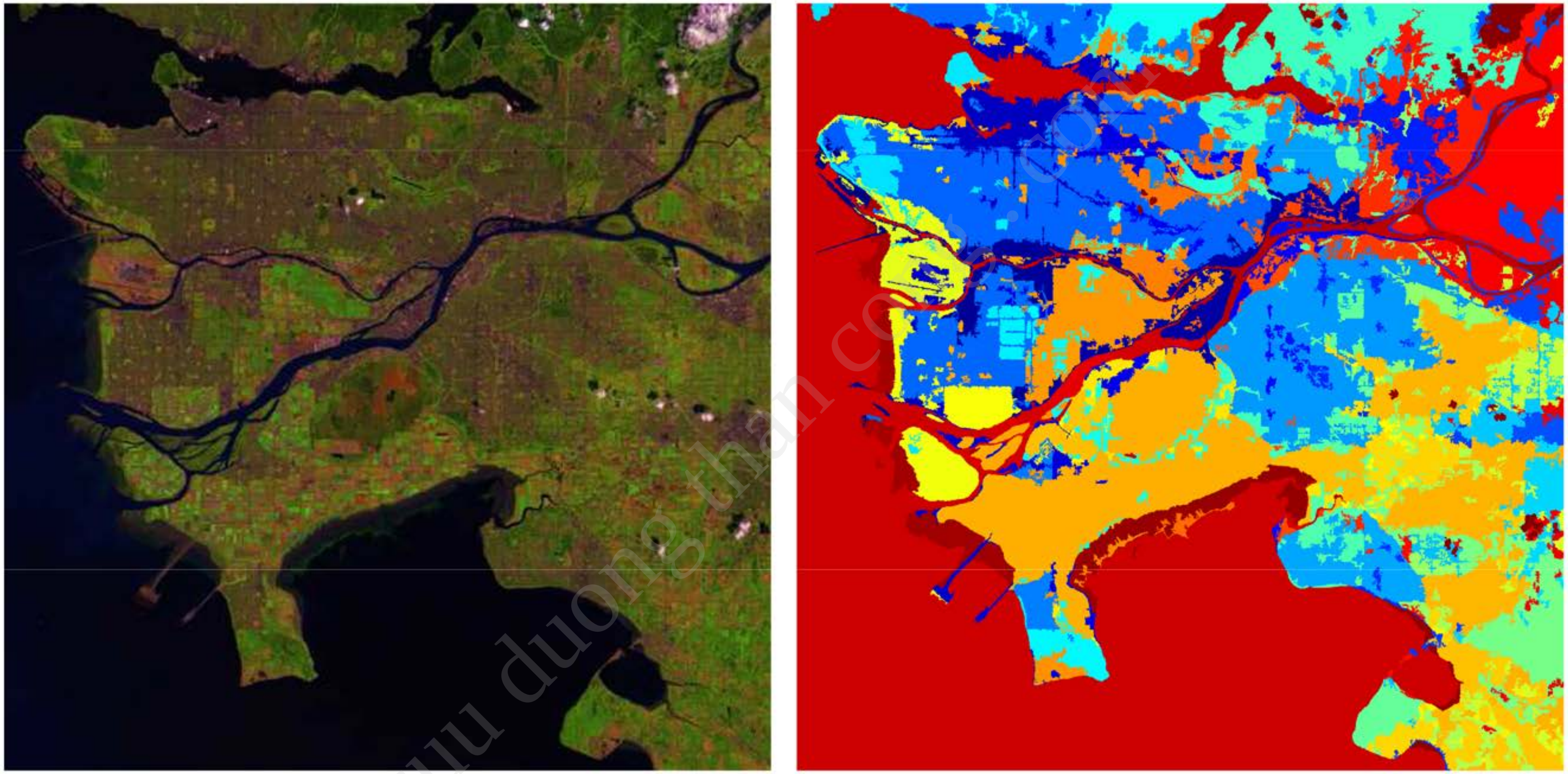
6. Image Representation & Description (4)

- Feature selected as descriptors should be **as insensitive as** possible to variations in
  - Size
  - Translation
  - Rotation
- Segmentation techniques yield raw data in the form of pixels along a boundary or pixels contained in a region. These data sometimes are used directly to obtain descriptors.
- Standard uses techniques to compute more useful data (descriptors) from the raw data in order to decrease the size of data.

6. IRD: Labeled Images and Overlays (1)

- **Labeled images** are good intermediate representations for regions.
  - The idea is to assign each detected region a unique identifier (an integer) and create an image where all pixels of a region will have its unique identifier as their pixel value.
  - A labeled image can be used as a kind of mask to identify pixels of a region.
  - Region boundaries can be computed from the labeled image and can be overlaid on top of the original image.

6. IRD: Labeled Images and Overlays (2)



A satellite image and the corresponding labeled image after segmentation

6. IRD: Labeled Images and Overlays (3)

A satellite image and the corresponding segmentation overlay

6. IRD: Chain Codes (1)

- Regions can be represented by their boundaries in a data structure instead of an image.

- The simplest form is just a linear list of the boundary points of each region.

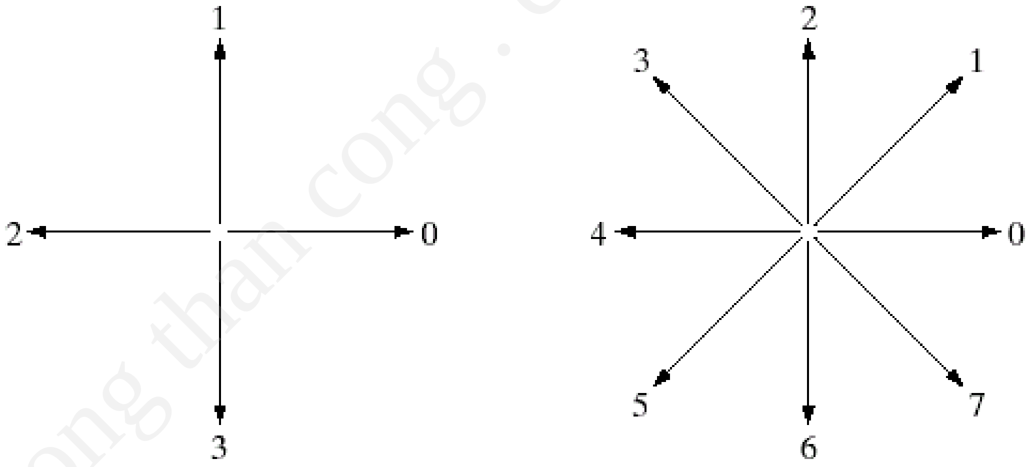
- This method generally is unacceptable because:
  - The resulting list tends to be quite long.
  - Any small disturbances along the boundary cause changes in the list that may not be related to the shape of the boundary.

- A variation of the list of points is the **chain code**, which encodes the information from the list of points at any desired quantization.

6. IRD: Chain Codes (2)

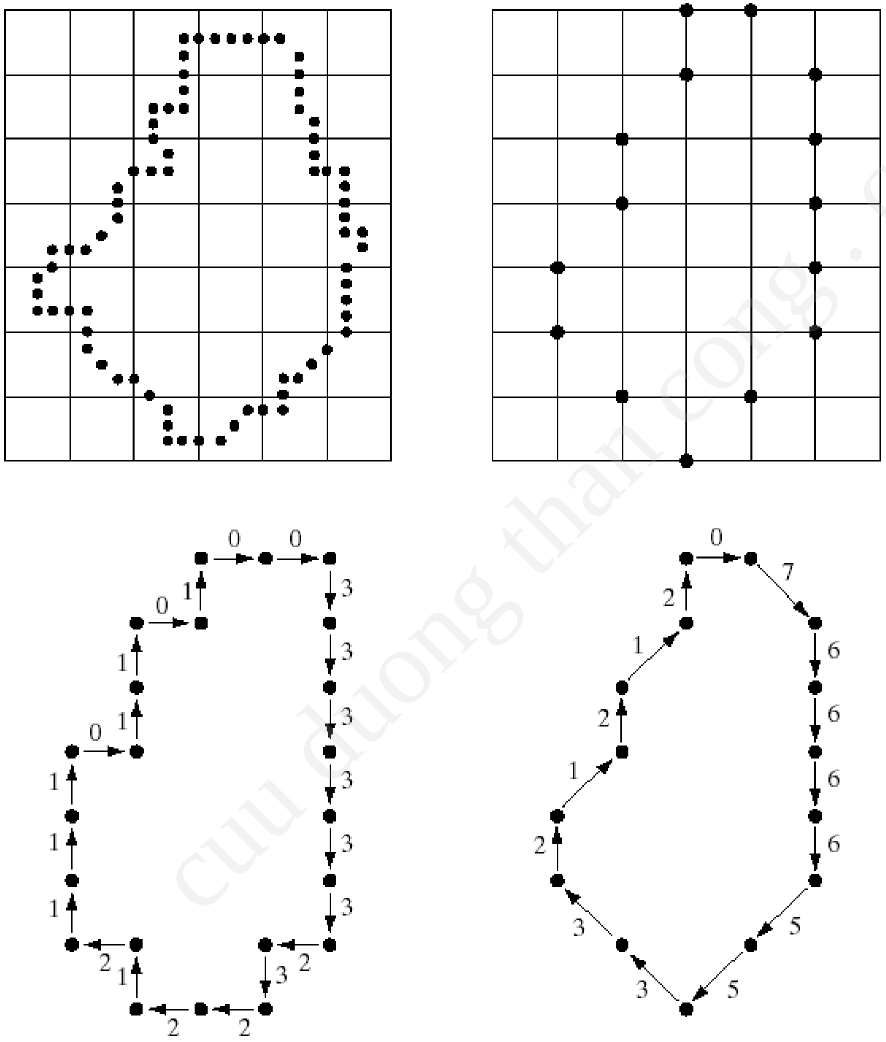
a b

FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



Based on 4 or 8 connectivity

6. IRD: Chain Codes (3)



a b
c d

FIGURE 11.2 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 4-directional chain code. (d) 8-directional chain code.

6. IRD: Chain Codes (4)

- Unacceptable because:
  - The resulting chain of codes tends to be quite long.
  - Any small disturbances along the boundary due to noise or imperfect segmentation cause changes in the code that may not be related to the shape of the boundary.
- Circumvent the problems by resampling the boundary by selecting a larger grid spacing. However, different grid can generate different chain codes.
- Starting point is arbitrary, need to normalize the generated code so that codes with different starting point will become the same.

6. IRD: Normalized Chain Codes

- Treat the chain code as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude ⇨ **shape numbers**.
- Or use rotation of the first different chain code instead.
- **Difference** is the number of direction changes in a counterclockwise direction.

Example:

- Code: 10103322
- Difference: 3133030
- circular chain code: 33133030
- rotation of circular chain code : 03033133 (shape number)

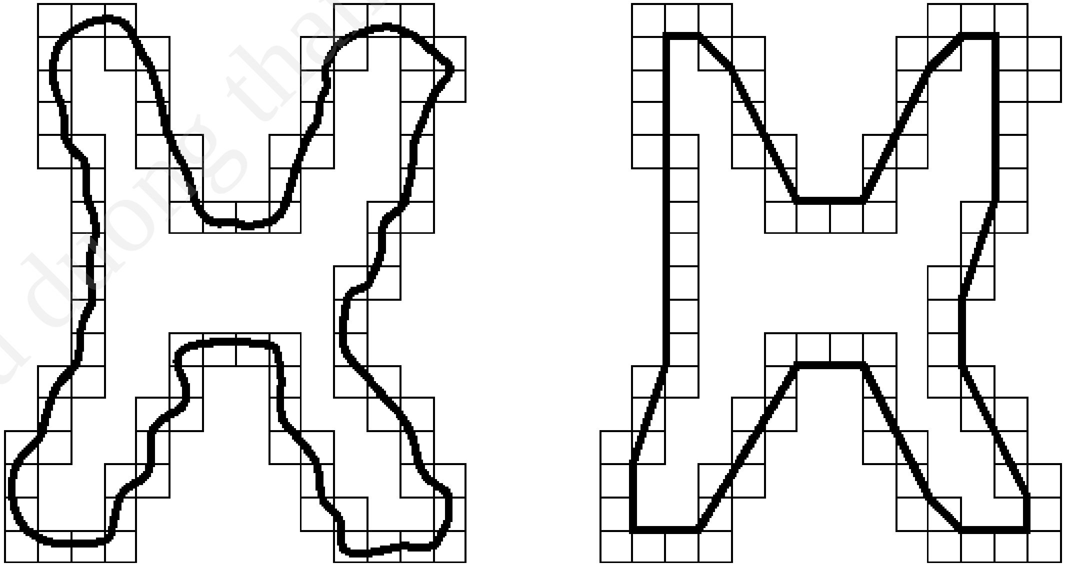
6. IRD: Polygonal Approximations (1)

- Boundary can be approximated with arbitrary accuracy by a polygon.
- Try to capture the “essence” of the boundary shape with the fewest possible polygonal segments.
- Not trivial and time consuming.

6. IRD: Polygonal Approximations (2)

❑ **Minimum Perimeter Polygons:**

- Enclose the boundary by a set of concatenated cells.
- Produce a polygon of minimum perimeter that fits the geometry established by the cell strip.



a b

FIGURE 11.3 (a) Object boundary enclosed by cells. (b) Minimum perimeter polygon.

6. IRD: Polygonal Approximations (3)

❑ **Merging Techniques**:

- Points along a boundary can be merged until the least square error line fit of the points merged so far exceeds a preset threshold.
- At the end of the procedure, the intersections of adjacent line segments form the vertices of the polygon.

6. IRD: Polygonal Approximations (4)

❑ **Splitting Techniques:**

1. Find the major axis.
2. Find minor axes which perpendicular to major axis and has distance greater than a threshold.
3. Repeat until we can't split anymore.

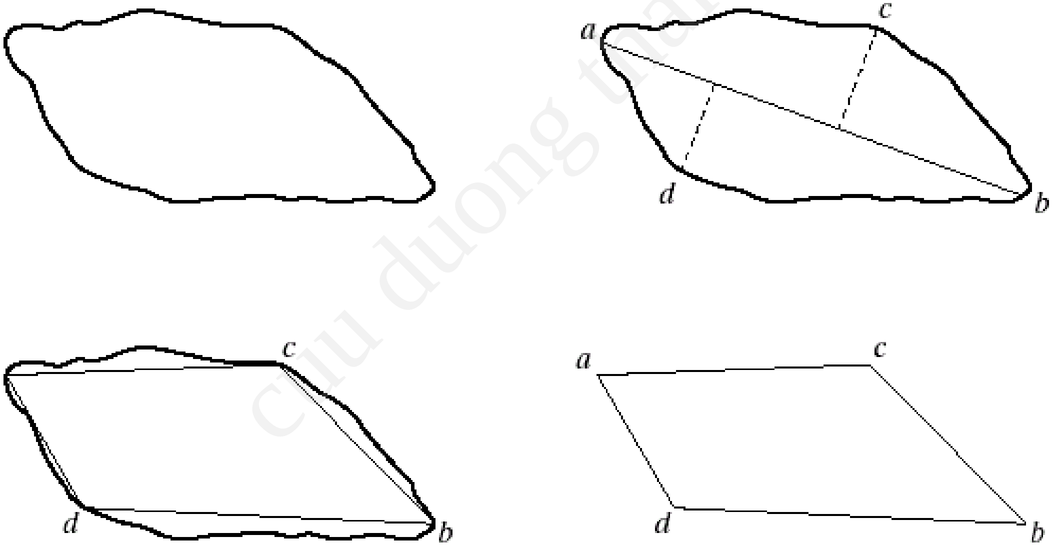
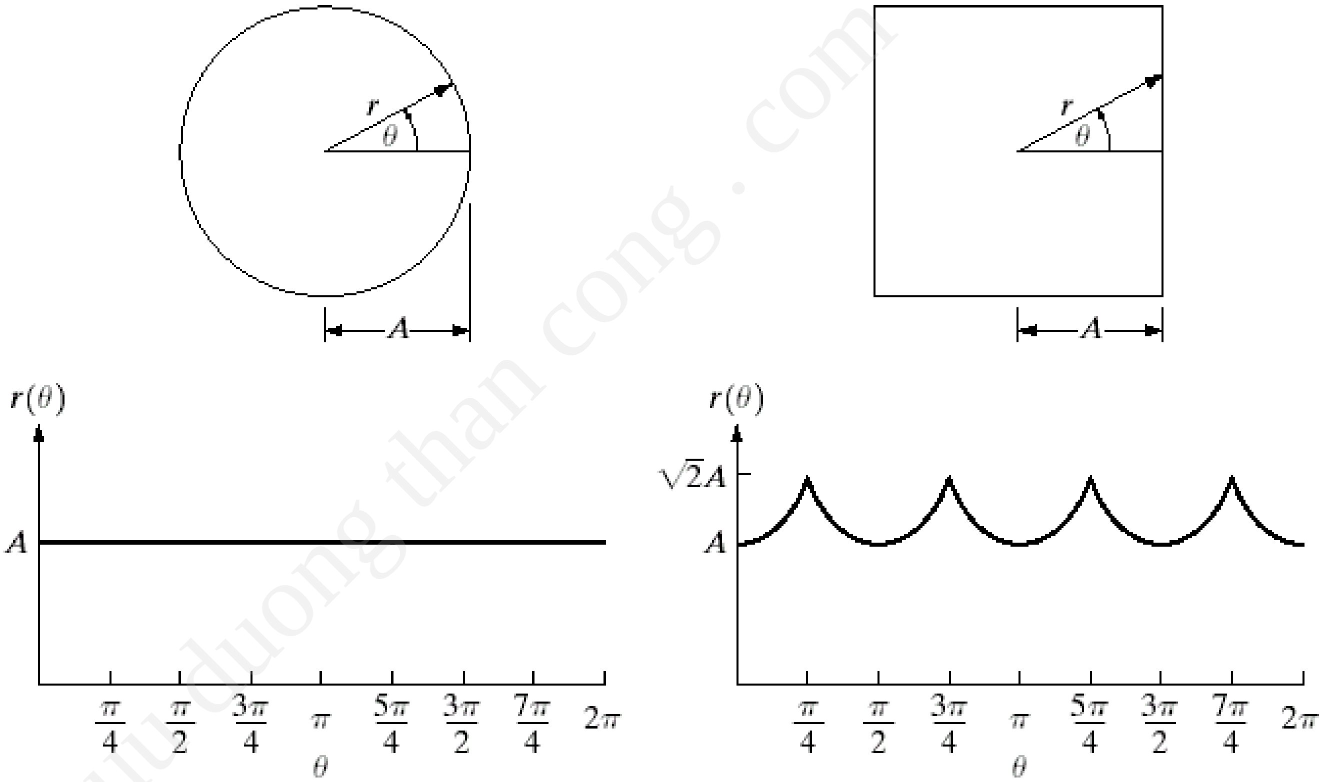


FIGURE 11.4 (a) Original boundary. (b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.

6. IRD: Signatures (Parametric Curves)

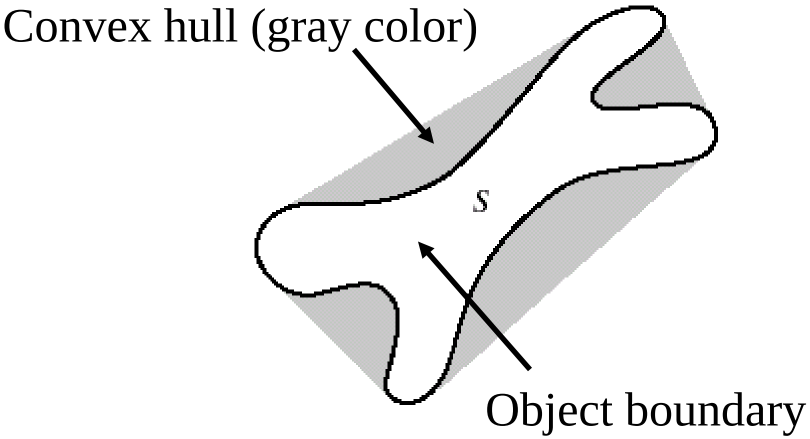
a b

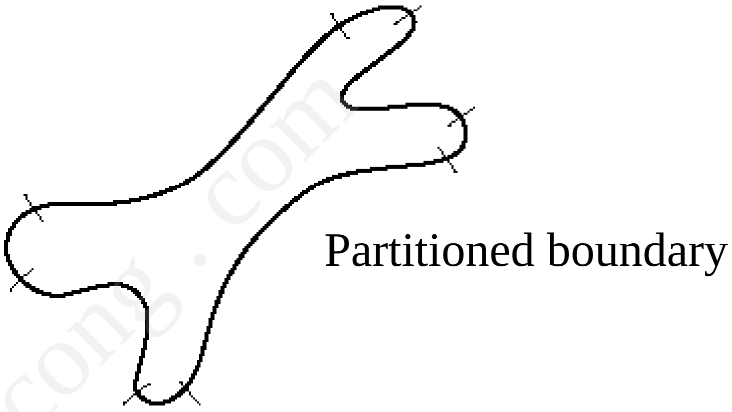
FIGURE 11.5 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A \sec\theta$ for $0 \leq \theta \leq \pi/4$ and $r(\theta) = A \csc\theta$ for $\pi/4 < \theta \leq \pi/2$.



Mapping 2D function to 1D function

6. IRD: Boundary Segments





- Decomposing a boundary into segments can reduce the boundary's complexity and simplify the description process.
- **Convex hull** *H* of an arbitrary set *S* is the smallest convex set containing *S*.
- The set different *H* - *S* is called **convex deficiency** *D* of the set *S*.
- The region boundary can be partitioned by following the contour of *S* and marking the points at which a transition is made into and out of a component of the convex deficiency.

6. IRD: Skeletons (1)

- An important approach to representing the structural shape of a plane region is to reduce it to a graph.
- This reduction may be accomplished by obtaining the **skeleton** of the region.
- The **medial axis transformation** (**MAT**) can be used to compute the skeleton.
- The MAT of a region *R* with border *B* is as follows:
  - For each point *p* in *R*, we find its closest neighbor in *B*.
  - If *p* has more than one such neighbor, it is said to belong to the medial axis of *R*.

6. IRD: Skeletons (2)

Medial Axis Transform (MAT): The MAT can be computed using thinning algorithms that iteratively delete edge points of a region subject to the constraints that deletion of these points:

- does not remove end points,
- does not break connectivity, and
- does not cause excessive erosion of the region.

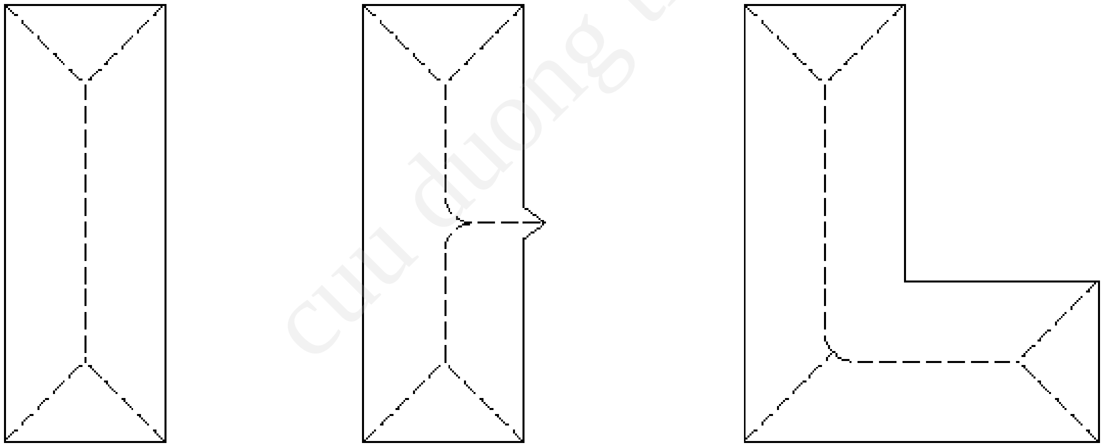


FIGURE 11.7 Medial axes (dashed) of three simple regions.

6. IRD: Thinning (1)

- Iterative deleting edge points of a region with constraints:
    - ✓ does not remove end points
    - ✓ does not break connectivity
    - ✓ does not cause excessive erosion of the region.
- Assume region points have value 1 and background points have value 0. Contour point is any pixel with value 1 and having at least one 8-neighbor valued 0.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

FIGURE 11.8 Neighborhood arrangement used by the thinning algorithm.

6. IRD: Thinning (2)

Step 1: Flag a contour point p_1 for deletion if the following conditions are satisfied:

$$(a) \quad 2 \le N(p_i) \le 6$$

$$(b) \quad T(p_1) = 1$$

$$(c) \quad p_2 \cdot p_4 \cdot p_6 = 0$$

$$(d) \quad p_4 \cdot p_6 \cdot p_8 = 0$$

$$N(p_1) = p_2 + p_3 + \ldots + p_8 + p_9$$

where $N(p_i)$ is the number of nonzero neighbors of p_i. After step 1 has marked every boundary points satisfy all 4 conditions, delete those pixels.

6. IRD: Thinning (3)

<u>Step 2</u>: Remain condition *(a)* and *(b)* but change conditions *(c)* and *(d)* to follows:

$$(c)' \quad p_2 \cdot p_4 \cdot p_8 = 0$$

$$(d)' \quad p_2 \cdot p_6 \cdot p_8 = 0$$

Flagged the remain border points for deletion, then delete the marked points.

Repeat step 1 and 2 until no more points to delete.

6. IRD: Thinning (4)

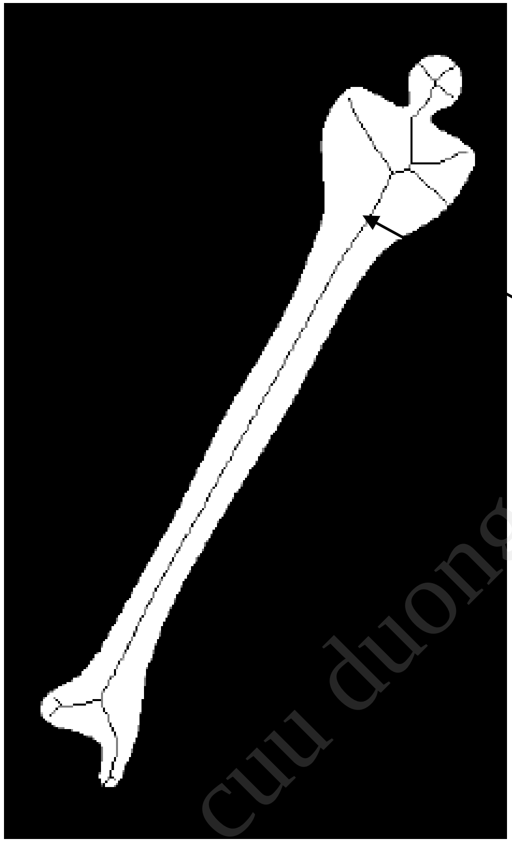


FIGURE 11.10 Human leg bone and skeleton of the region shown superimposed.

Skeleton obtained from thinning algorithm

6. IRD: Description

- **Boundary descriptors** use external representations to model the shape characteristics of regions.
- **Regional descriptors** use internal representations to model the internal content of regions.

Examples:

- Shape properties (both internal and external)
- Fourier descriptors (external)
- Statistics and histograms (internal)

6. IRD: Boundary Descriptors (1)

- ☐ **Length of a boundary:**
  - – The number of pixels along a boundary.
  - – Giving a rough approximation of its length.
- ☐ **Diameters:** $Diam(B) = \max_{i,j}[D(p_i, p_j)]$

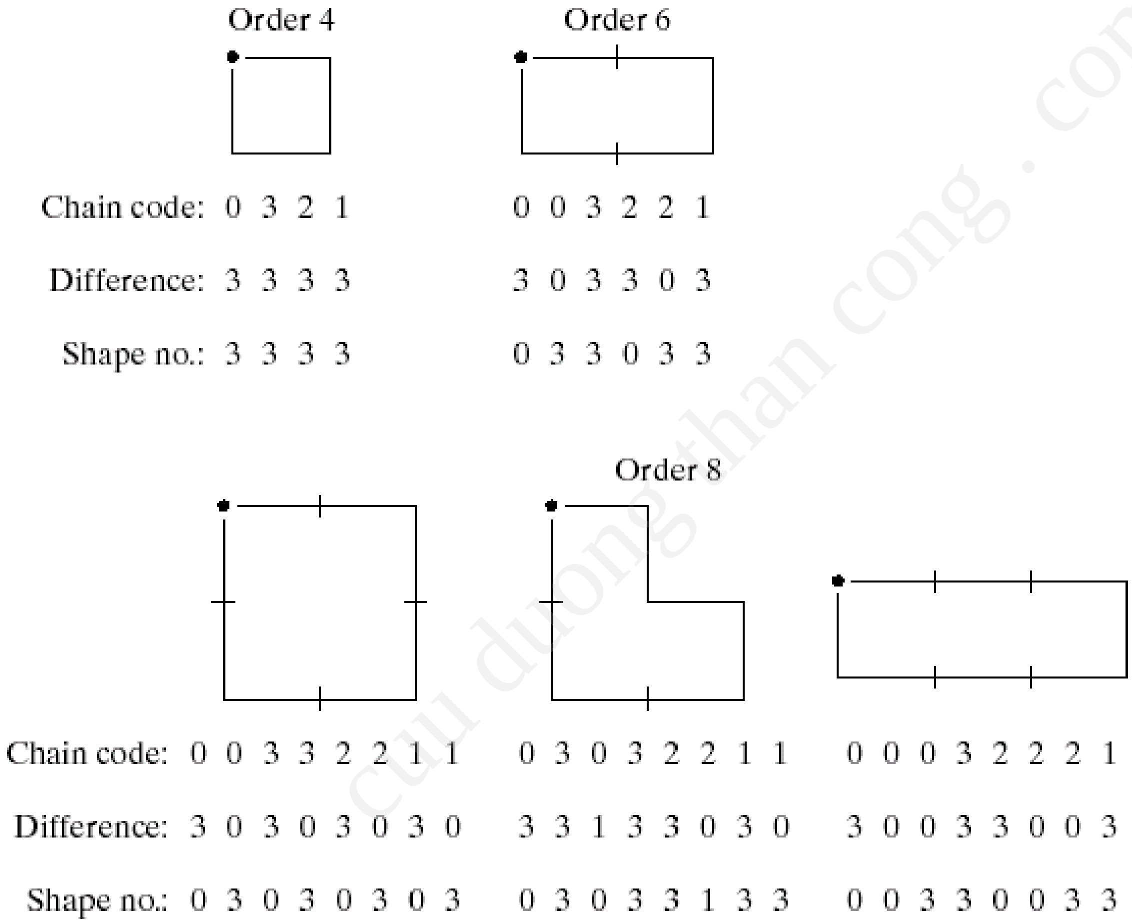
  where D is a distance measure; p_i and p_j are points on the boundary B.
- ☐ **Eccentricity:** Ratio of the major to the minor axis
  - – Major axis is the line connecting the two extreme points that comprise the diameter.
  - – Minor axis is the line perpendicular to the major axis.

6. IRD: Boundary Descriptors (2)

- **Curvature:** The rate of change of slope.
  - Difficult to do as digital boundaries tend to be locally "ragged".
  - Using the difference between the slopes of adjacent boundary segments (which represented as straight lines). Using Merging and Splitting techniques to create adjacent boundary segments.
  - Concave, convex and corner.

6. IRD: Boundary Descriptors (3)

❑ Shape numbers:



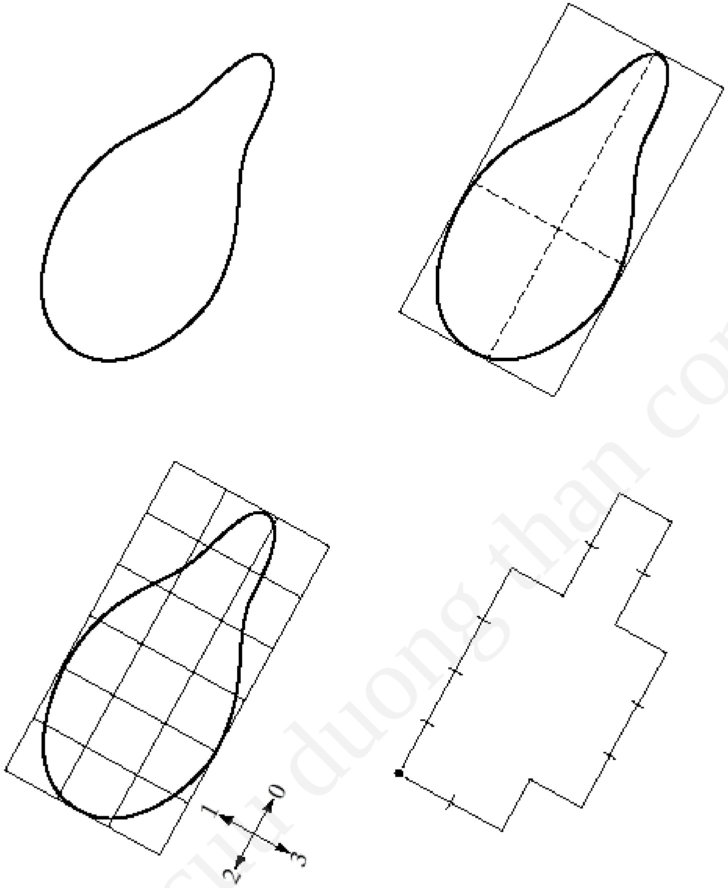
	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

	Order 8		
Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

FIGURE 11.11 All shapes of order 4, 6, and 8. The directions are from Fig. 11.1(a), and the dot indicates the starting point.

4-directional code

6. IRD: Boundary Descriptors (4)



a b
c d

FIGURE 11.12 Steps in the generation of a shape number.

Chain code: 0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape no.: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

6. IRD: Boundary Descriptors (5)

- ☐ **Fourier descriptors:**

Boundary = $(x_0,y_0), \ldots , (x_{K-1},y_{k-1})$

$$s(k) = x(k) + jy(k) \text{ for } k = 0,1,\ldots,K\text{-}1$$

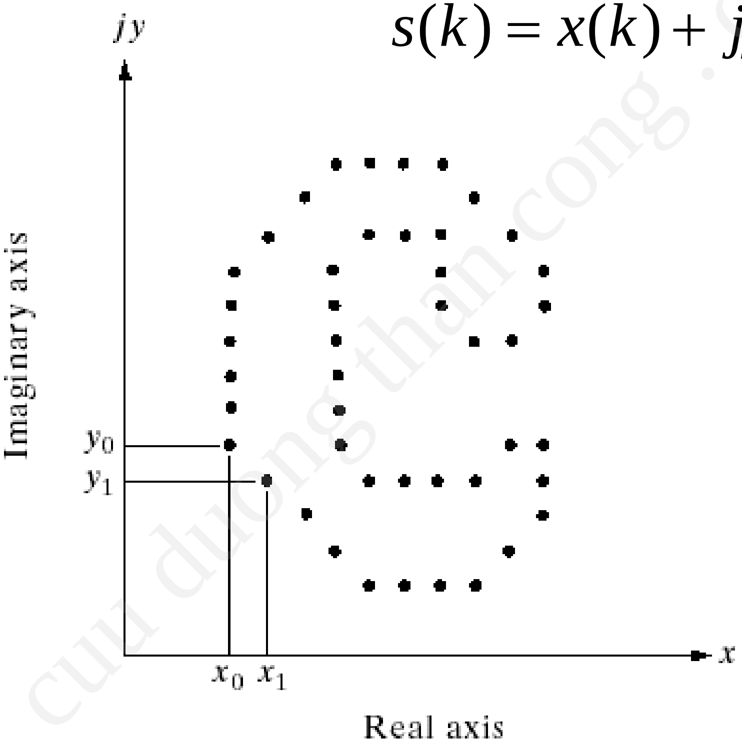


FIGURE 11.13 A digital boundary and its representation as a complex sequence. The points (x_0, y_0) and (x_1, y_1) shown are (arbitrarily) the first two points in the sequence.

6. IRD: Boundary Descriptors (6)

- Fourier transformation (DFT):

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K} \quad \text{for} \quad u = 0,1,\ldots,K\text{-}1$$

a(*u*) : Fourier coefficients (**Fourier descriptors**).

- Inverse Fourier transformation:

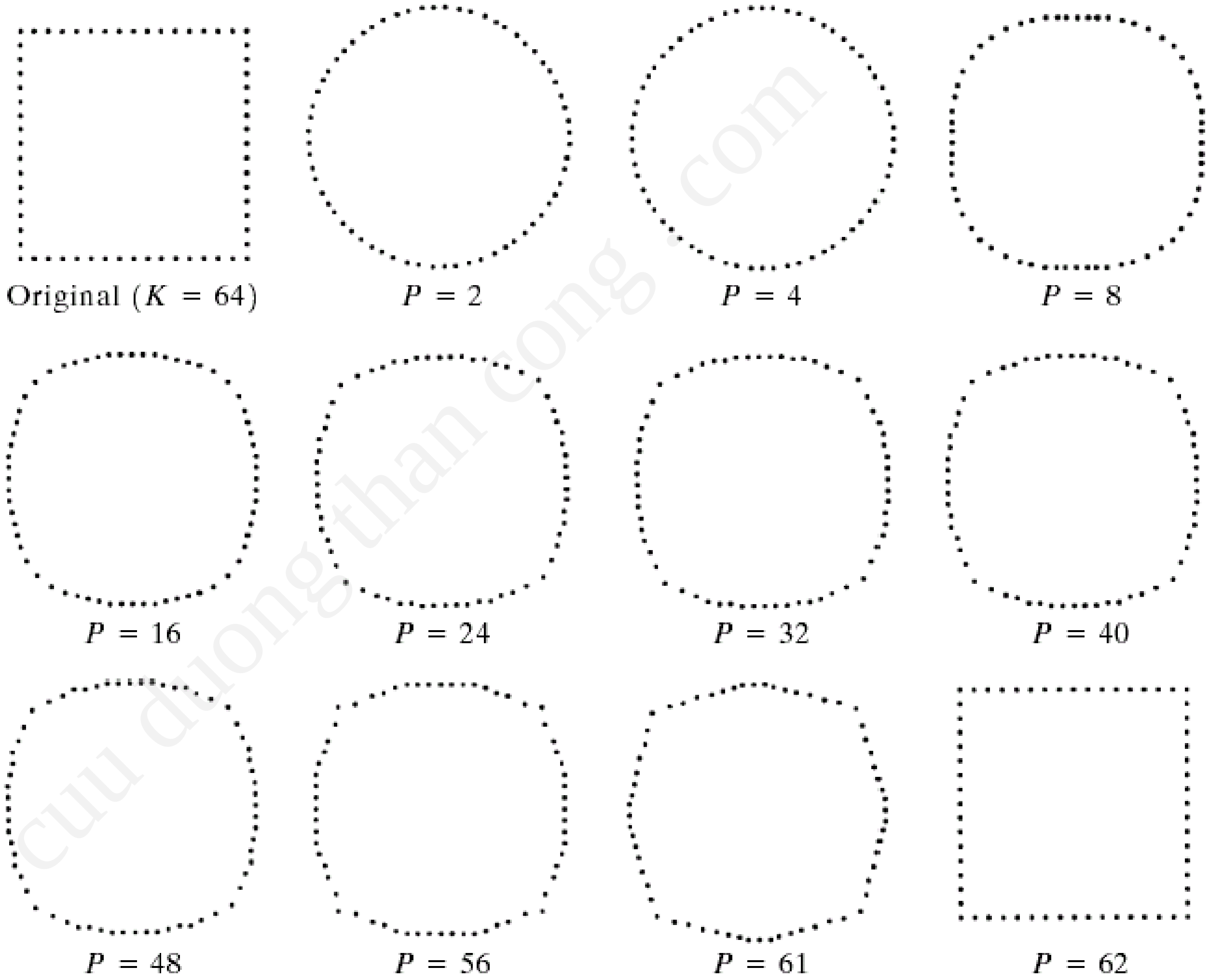
$$s(k) = \sum_{u=0}^{K-1} a(u)e^{j2\pi uk/K} \quad \text{for} \quad k = 0,1,\ldots,K\text{-}1$$

- *P* coefficients of Fourier Descriptors

$$\hat{s}(k) = \sum_{u=0}^{P-1} a(u)e^{j2\pi uk/K} \quad \text{for } k = 0,1,\ldots,K\text{-}1$$

Descriptors ⇨ *P* number of coefficients.

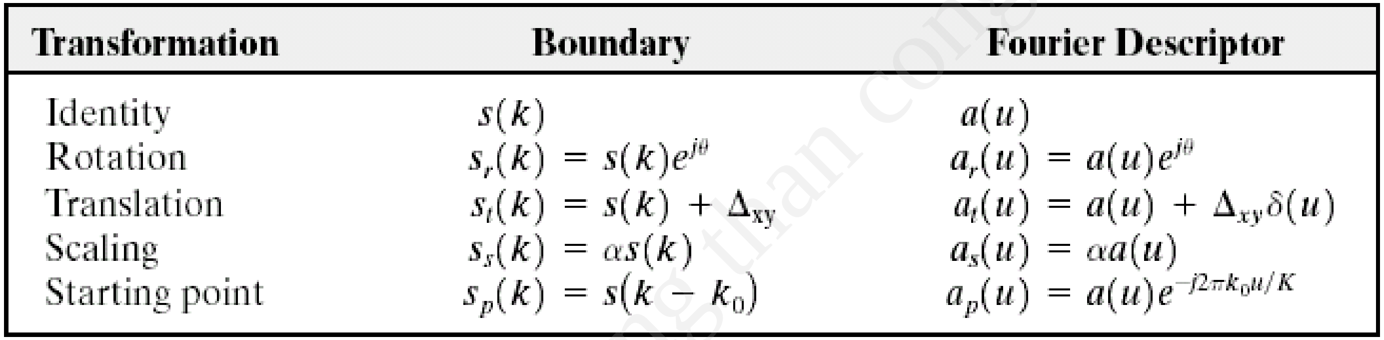
6. IRD: Boundary Descriptors (7)



Original ($K = 64$)
$P = 2$
$P = 4$
$P = 8$
$P = 16$
$P = 24$
$P = 32$
$P = 40$
$P = 48$
$P = 56$
$P = 61$
$P = 62$

FIGURE 11.14 Examples of reconstruction from Fourier descriptors. *P* is the number of Fourier coefficients used in the reconstruction of the boundary.

6. IRD: Boundary Descriptors (8)



Transformation	Boundary	Fourier Descriptor
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

TABLE 11.1
Some basic properties of Fourier descriptors.

6. IRD: Boundary Descriptors (9)

❑ **Statistical moments:**

Definition: the n^{th} moment:

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i)$$

where

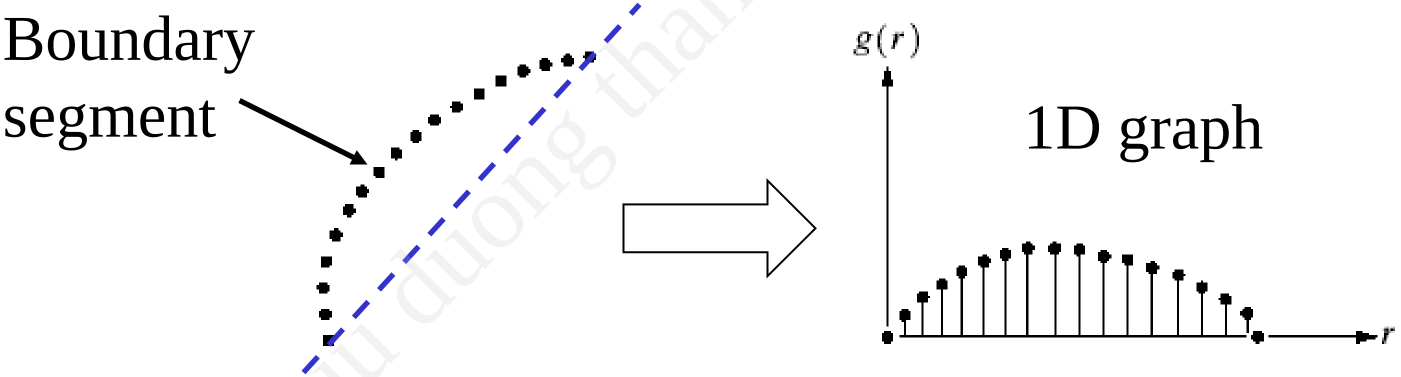
$$m = \sum_{i=0}^{K-1} r_i g(r_i)$$

Example of moment: The first moment = mean; the second moment = variance.

6. IRD: Boundary Descriptors (10)

Find statistical moments:

1. Convert a boundary segment into 1D graph.
2. View a 1D graph as a PDF function.
3. Compute the n^{th} order moment of the graph.



6. IRD: Regional Descriptors (1)

1. **Some simple regional descriptors:**
   - **Area:** The number of pixels in the region.
   - **Perimeter:** Length of its boundary.
   - **Compactness:** (perimeter)^2/area.
2. **Topological descriptors**
3. **Texture descriptors**
4. **Statistics and histograms**

6. IRD: Regional Descriptors (2)

- **Topological descriptors:**

$$E = C - H$$

where:
E is Euler number,
C is number of connected regions,
and H is number of holes.

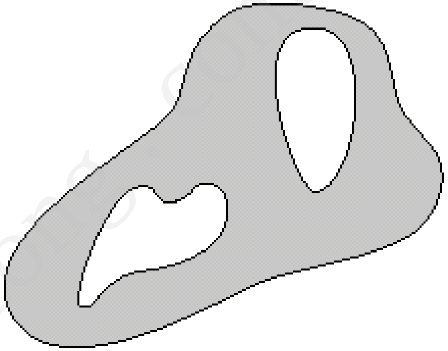


FIGURE 11.17 A region with two holes.

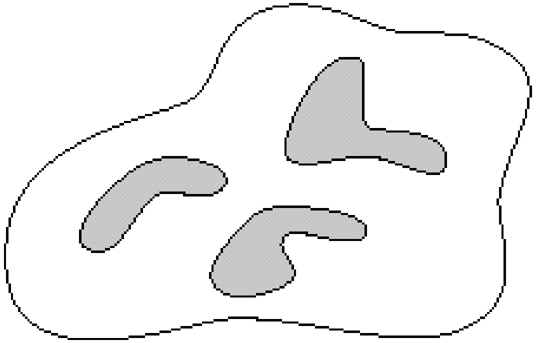
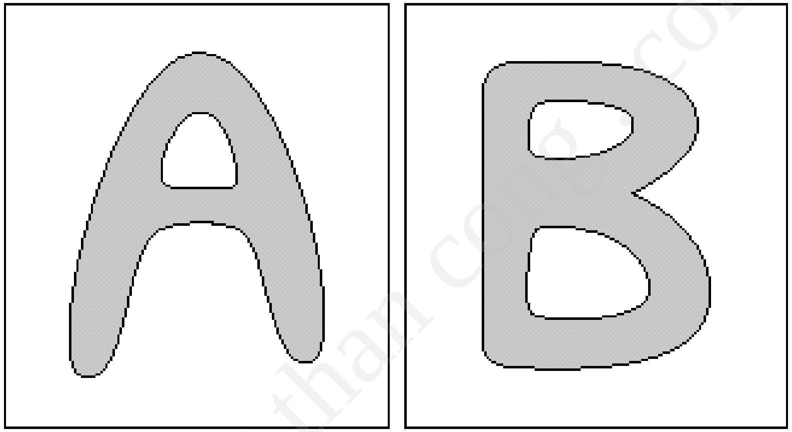


FIGURE 11.18 A region with three connected components.

6. IRD: Regional Descriptors (3)



a b

FIGURE 11.19 Regions with Euler number equal to 0 and −1, respectively.

6. IRD: Regional Descriptors (4)

Straight-line segments (polygon networks)

$V - Q + F = C - H = E$

V = number of vertices
Q = number of edges
F = number of faces

7-11+2 = 1-3 = -2

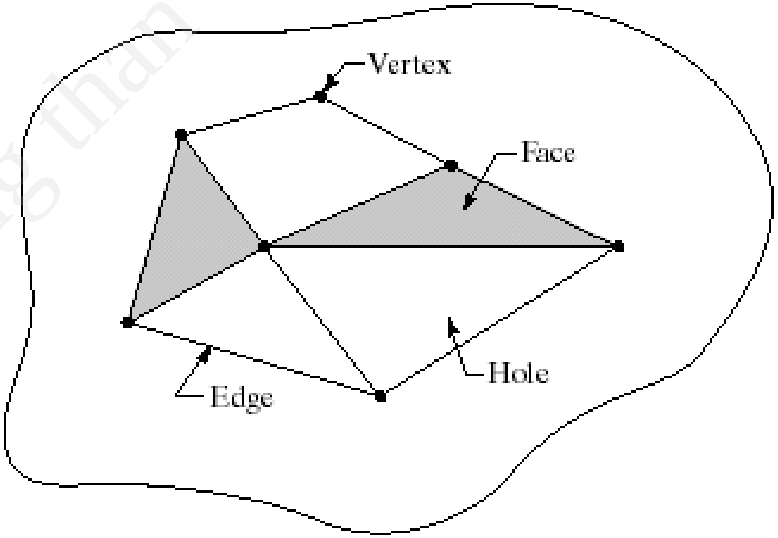


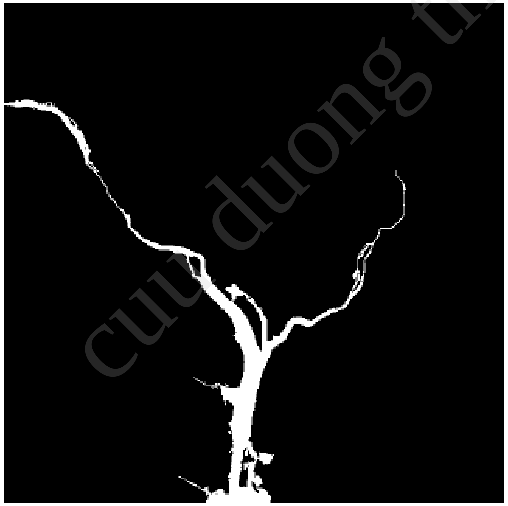
FIGURE 11.20 A region containing a polygonal network.

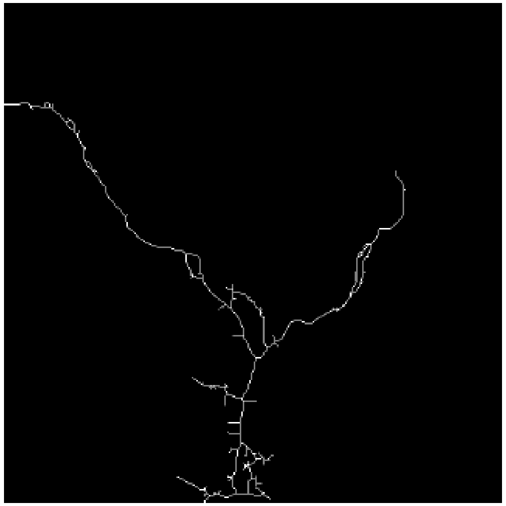
6. IRD: Regional Descriptors (5)

Original image: Infrared image of Washington D.C. area.

After intensity Thresholding (1591 connected components with 39 holes) Euler no. = 1552

The largest connected area (8479 Pixels) (Hudson river)





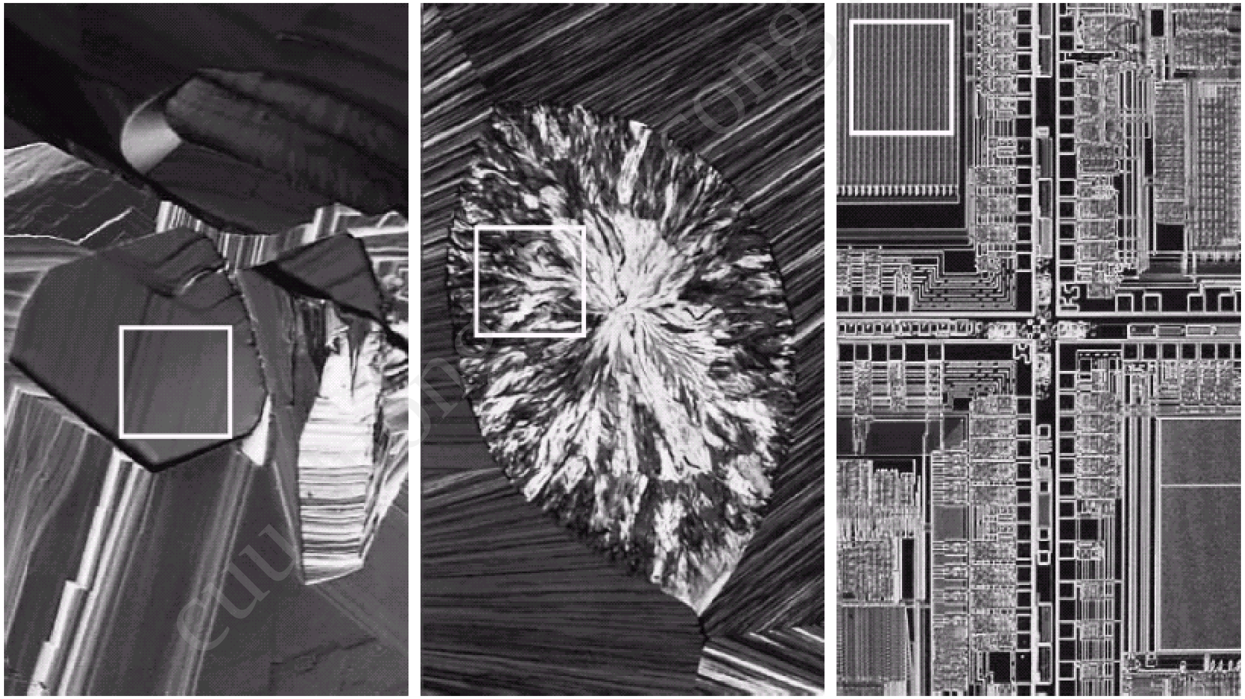
After thinning

6. IRD: Regional Descriptors (6)

❑ **Texture descriptors**

Purpose: to describe "texture" of the region.

Examples: Optical microscope images:



Superconductor (smooth texture)

Cholesterol (coarse texture)

Microprocessor (regular texture)

6. IRD: Regional Descriptors (7)

- **Statistical approaches for texture descriptors**:

$$\mu_n(z) = \sum_{i=0}^{K-1} (z_i - m)^n p(z_i)$$

where

$$m = \sum_{i=0}^{K-1} z_i p(z_i)$$

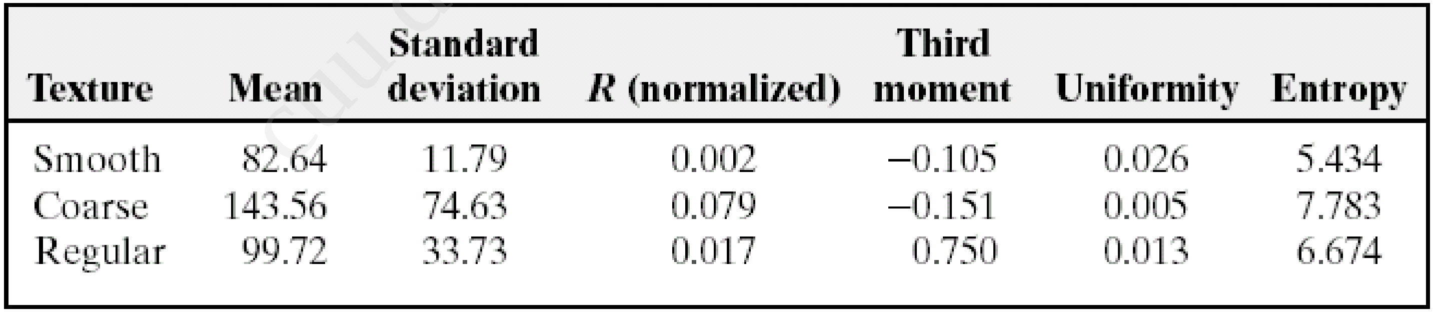
z = intensity

$p(z)$ = PDF or histogram of z.

The 2nd moment = variance → measure "**smoothness**"

The 3rd moment → measure "**skewness**"

The 4th moment → measure "uniformity" (**flatness**)

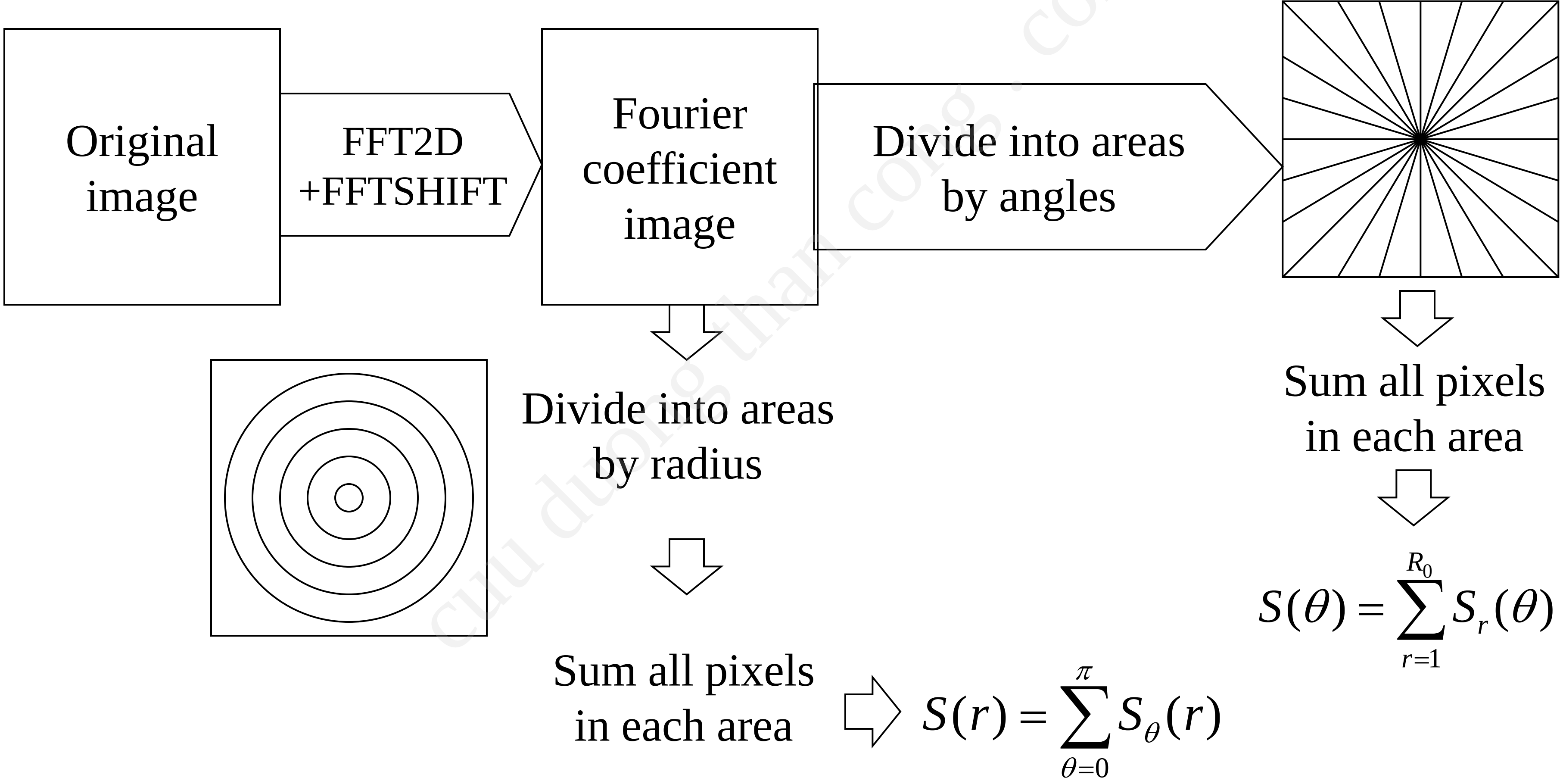


Texture	Mean	Standard deviation	R (normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

6. IRD: Regional Descriptors (8)

- **Fourier approach for texture descriptor:**

  Concept: convert 2D spectrum into 1D graphs.



Original image

FFT2D +FFTSHIFT

Fourier coefficient image

Divide into areas by angles

Sum all pixels in each area

$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)$$

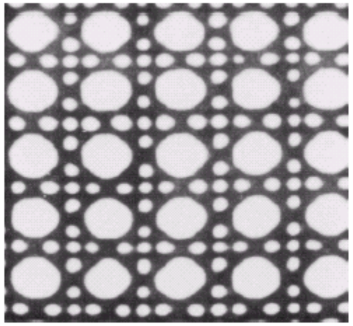
Divide into areas by radius

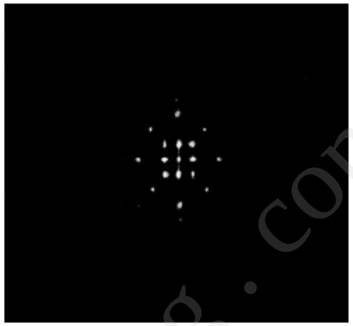
Sum all pixels in each area

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

6. IRD: Regional Descriptors (9)

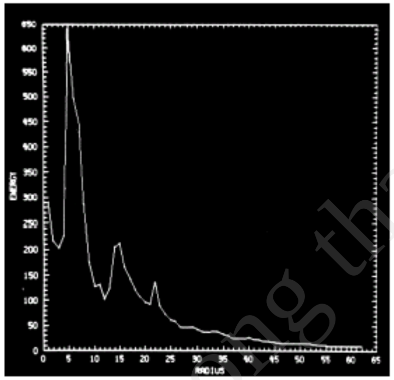
Original image

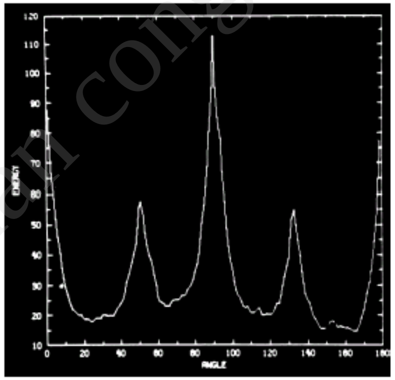




2D Spectrum (Fourier transform)

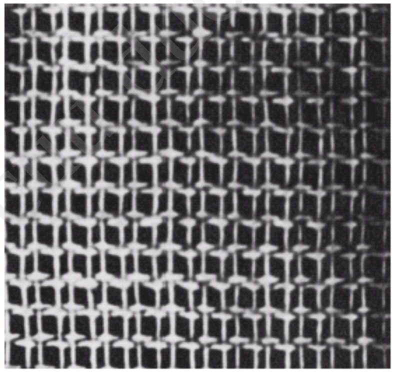
$S(r)$

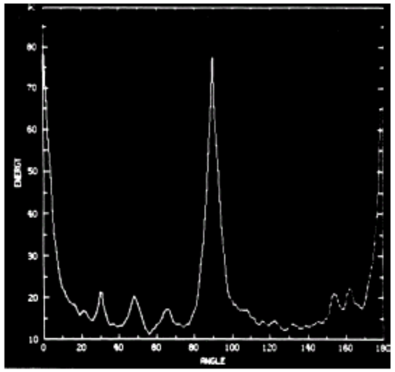




$S(\theta)$

Another image





Another $S(\theta)$

6. IRD: Regional Descriptors (10)

- **Statistics and histograms:**

Contents of regions can be summarized using statistics (e.g., mean, standard deviation) and histograms of pixel features.

Commonly used pixel features include

- Gray tone,
- Color (RGB, HSV, …),
- Texture,
- Motion.

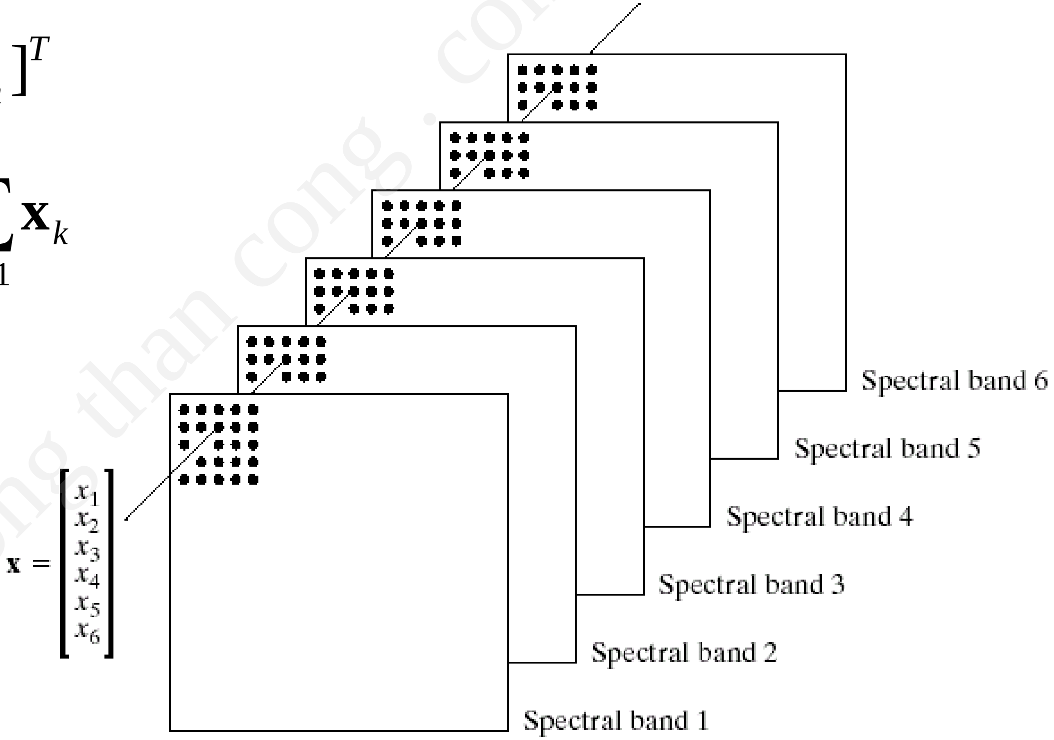
Then, the resulting region level features can be used for clustering, retrieval, classification, etc.

6. IRD: Principal Components for Description (1)

Purpose: to reduce dimensionality of a vector image while maintaining information as much as possible.

Let $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T$

Mean: $\mathbf{m}_x = E\{\mathbf{x}\} = \dfrac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k$



Covariance matrix: $\mathbf{C}_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} = \dfrac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T$

6. IRD: Principal Components for Description (2)

Hotelling transformation:

Let $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$

where **A** is created from eigenvectors of $\mathbf{C}_\mathrm{x}$ as follows: Row 1 contains the 1st eigenvector with the largest eigenvalue. Row 2 contains the 2nd eigenvector with the 2nd largest eigenvalue, etc.

Then we get

$$\mathbf{m}_y = E\{\mathbf{y}\} = 0$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T$$

and

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \lambda_1 \end{bmatrix}$$

Then elements of $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$ are uncorrelated. The component of **y** with the largest λ is called the **principal component**.

6. IRD: Principal Components for Description (3)

6 spectral images from an airborne scanner.

Channel	Wavelength band (microns)
1	0.40–0.44
2	0.62–0.66
3	0.66–0.72
4	0.80–1.00
5	1.00–1.40
6	2.00–2.60


Channel 1


Channel 2

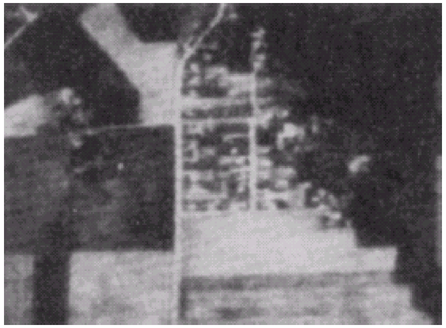

Channel 3

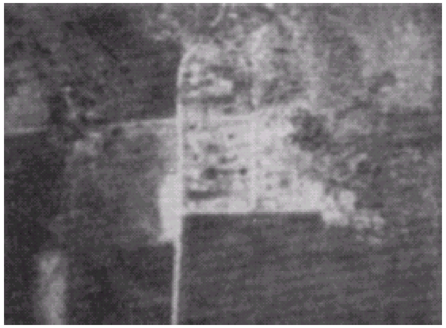

Channel 4

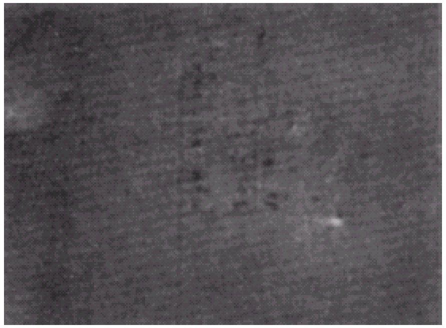

Channel 5


Channel 6

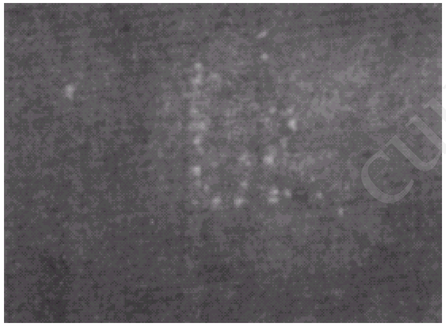
6. IRD: Principal Components for Description (4)


Component 1


Component 2


Component 3

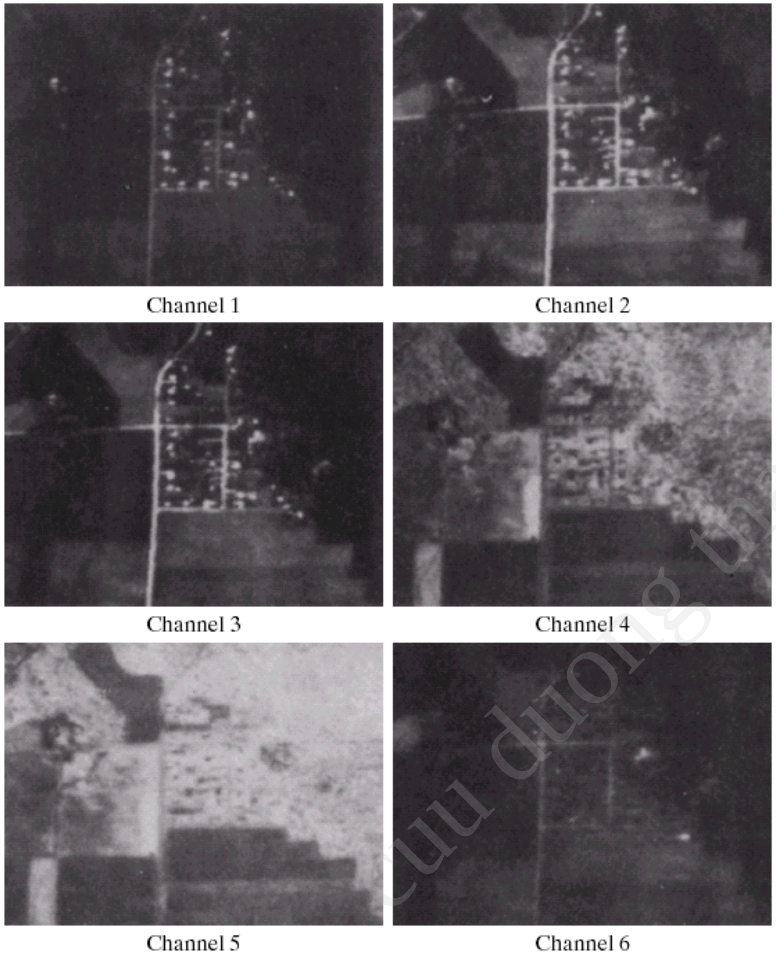

Component 4


Component 5


Component 6

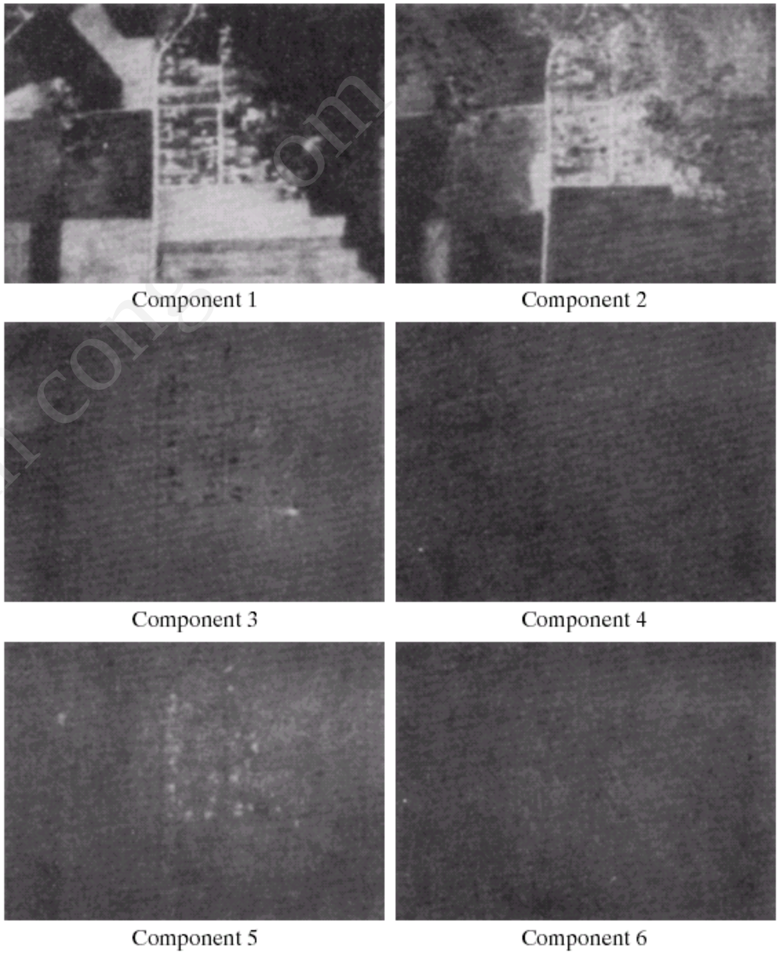
Component	λ
1	3210
2	931.4
3	118.5
4	83.88
5	64.00
6	13.40

6. IRD: Principal Components for Description (5)



Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6

Original image



Component 1
Component 2
Component 3
Component 4
Component 5
Component 6

After Hotelling transform

6. IRD: Principal Components for Description (6)

Principal components for description:

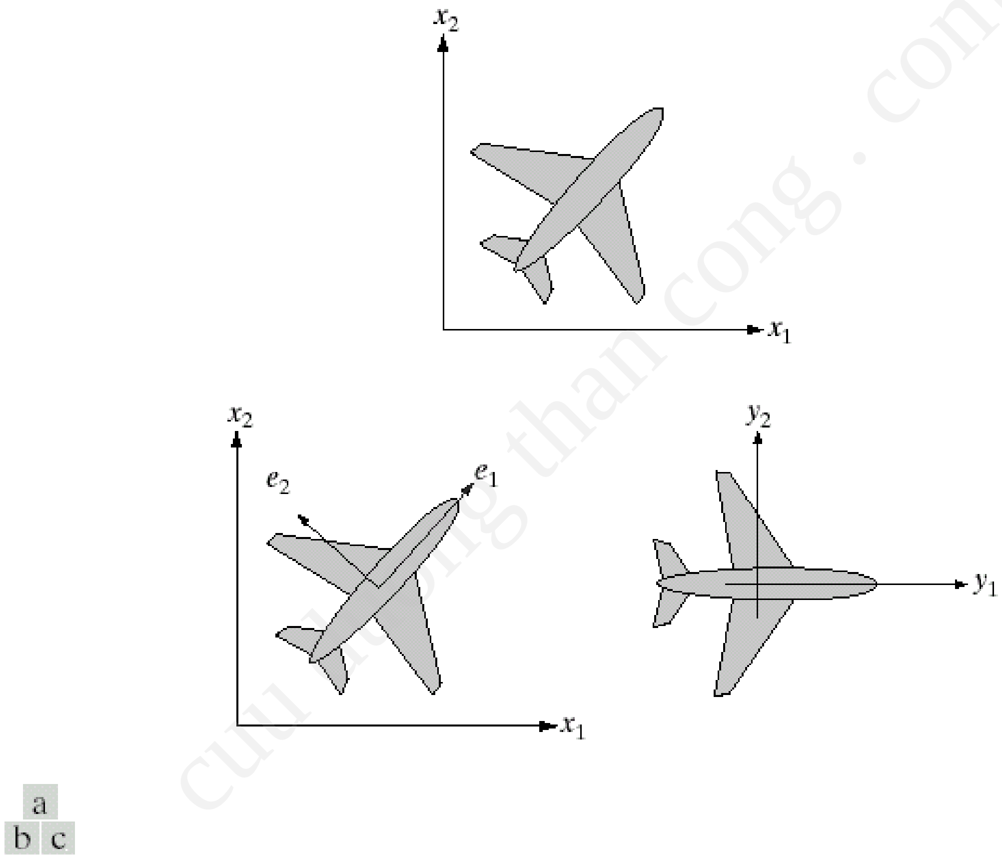


FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

6. IRD: Relational Descriptors (1)

a b

FIGURE 11.30 (a) A simple staircase structure. (b) Coded structure.

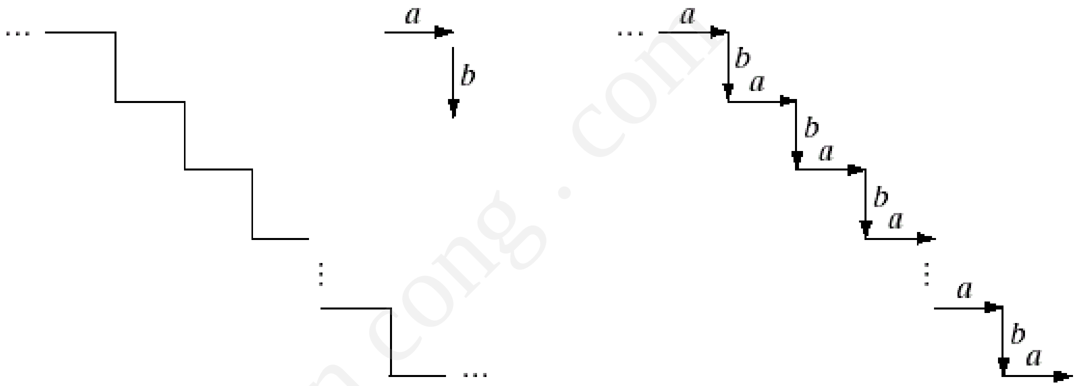
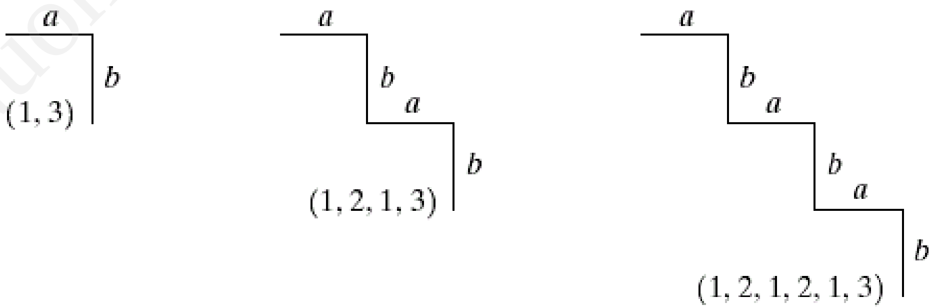


FIGURE 11.31 Sample derivations for the rules $S \rightarrow aA$, $A \rightarrow bS$, and $A \rightarrow b$.



6. IRD: Relational Descriptors (2)

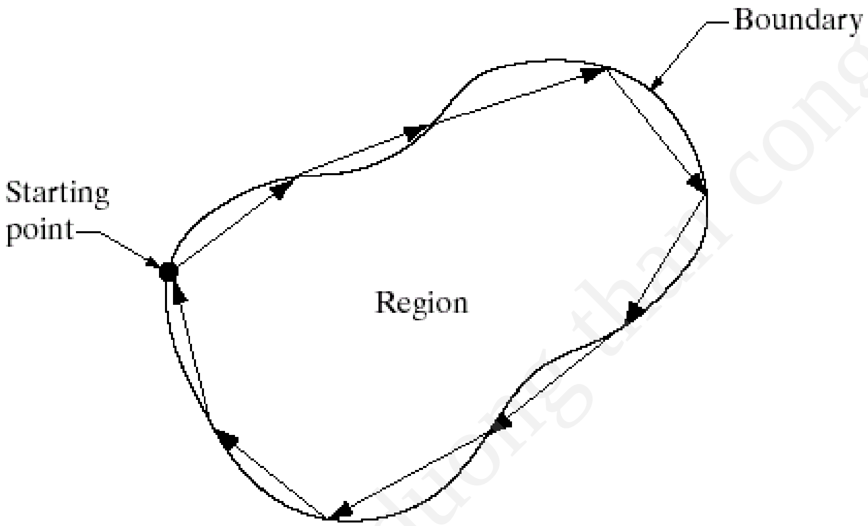
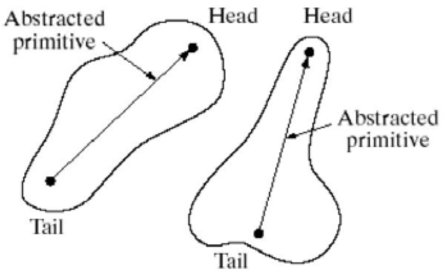
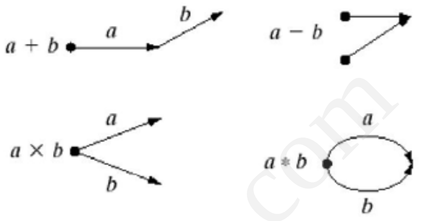


FIGURE 11.32 Coding a region boundary with directed line segments.

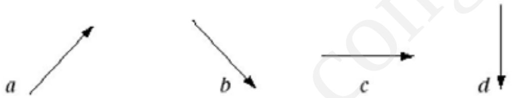
6. IRD: Relational Descriptors (3)

Abstracted primitives



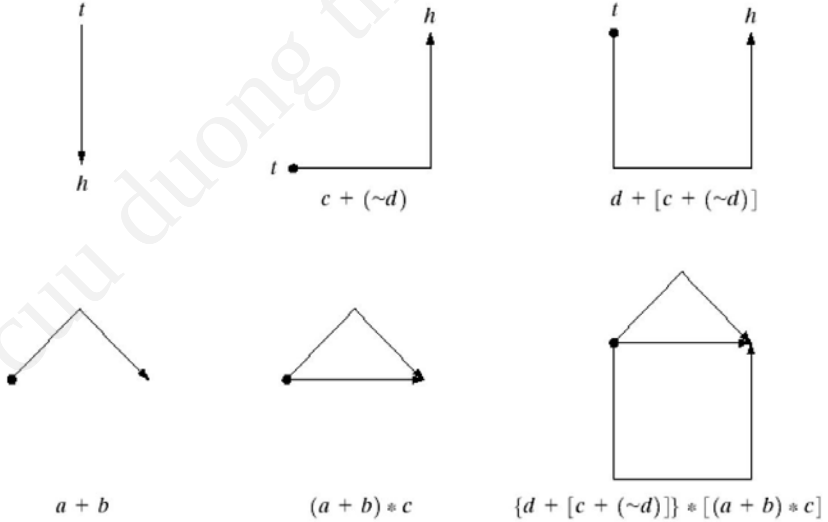


Operations among primitives



A set of specific primitives

Steps in building a structure



6. IRD: Relational Descriptors (4)

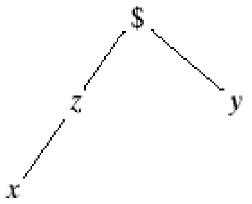
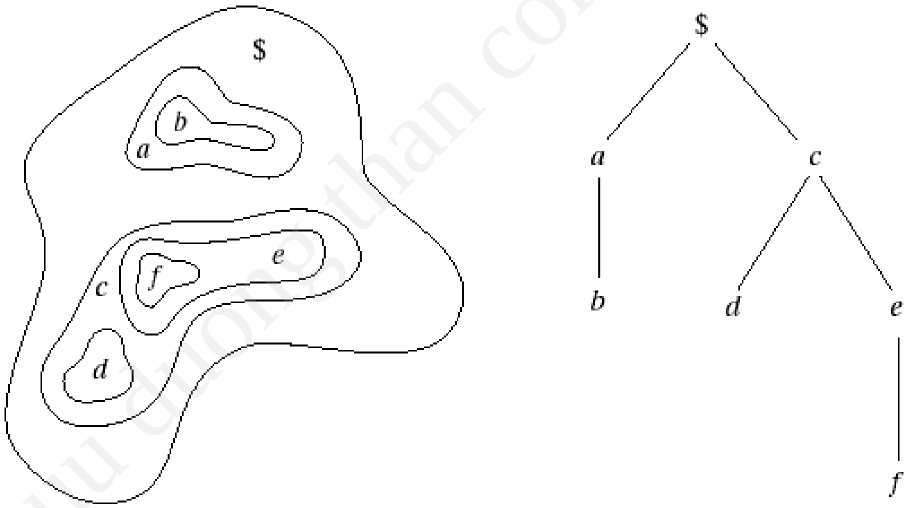


FIGURE 11.34 A simple tree with root $ and frontier *xy*.

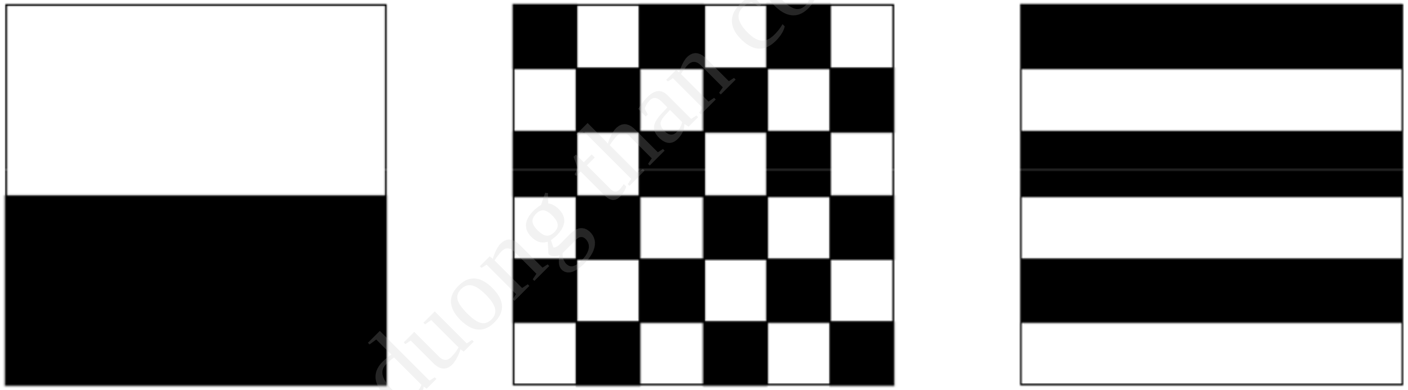


a b

FIGURE 11.35 (a) A simple composite region. (b) Tree representation obtained by using the relationship "inside of."

6. IRD: Texture Analysis (1)

- An important approach to image description is to quantify its texture content.
- Texture gives us information about the **spatial arrangement of the colors or intensities** in an image.



block pattern **checkerboard** **striped pattern**

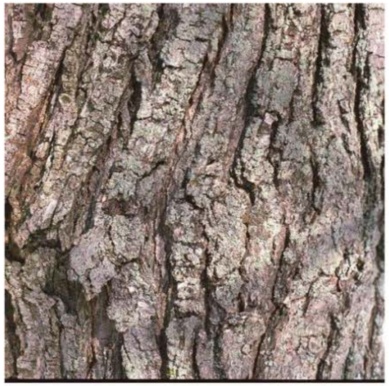
Three different textures with same distribution of black and white.

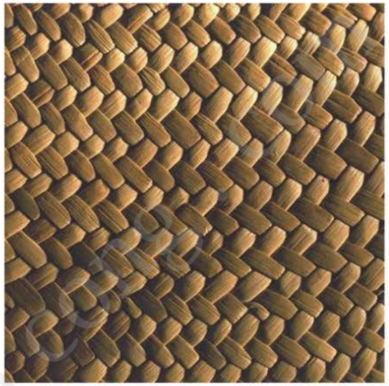
6. IRD: Texture Analysis (2)

- Although no formal definition of texture exists, intuitively it can be **defined as the uniformity, density, coarseness, roughness, regularity, intensity and directionality of discrete tonal features and their spatial relationships**.
- Texture is commonly found in natural scenes, particularly in outdoor scenes containing both natural and man-made objects.

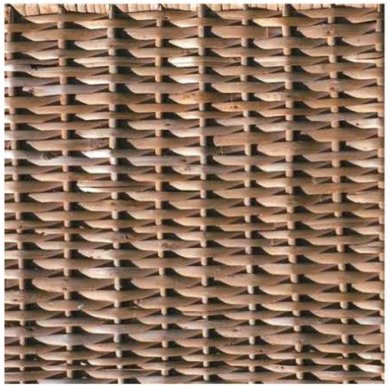
6. IRD: Texture Analysis (3)


Bark


Bark


Fabric


Fabric


Fabric


Flowers

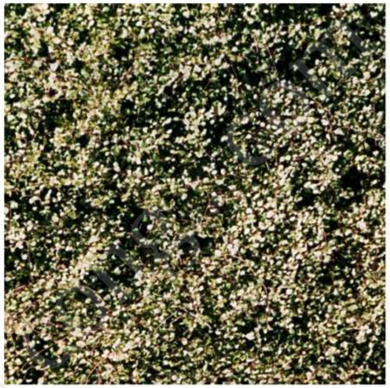

Flowers

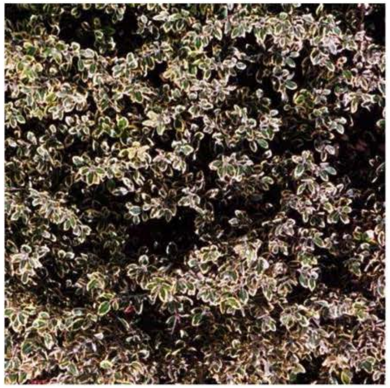

Flowers

6. IRD: Texture Analysis (4)


Food


Food


Leaves


Leaves


Leaves


Leaves


Water


Water

6. IRD: Texture Analysis (5)

- The approaches for characterizing and measuring texture can be grouped as:
  - **Structural approaches** that use the idea that textures are made up of primitives appearing in a near-regular repetitive arrangement,
  - **Statistical approaches** that yield a quantitative measure of the arrangement of intensities.
- While the first approach is appealing and can work well for man-made, regular patterns, the second approach is more general and easier to compute and is used more often in practice.

6. IRD: Texture Analysis (6)

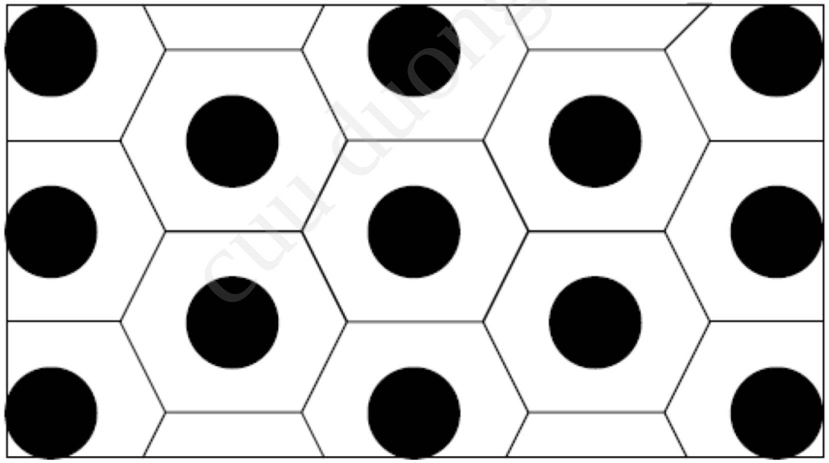
❑ **Structural approaches:**

- Structural approaches model texture as a set of primitive **texels** (**texture elements**) in a particular spatial relationship.
- A structural description of a texture includes a description of the texels and a specification of the spatial relationship.
- The texels must be identifiable and the relationship must be efficiently computable.

6. IRD: Texture Analysis (7)

Voronoi tessellation of texels by Tuceryan and Jain:

- Texels are extracted by thresholding.
- Spatial relationships are characterized by their Voronoi tessellation.
- Shape features of the Voronoi polygons are used to group the polygons into clusters that define uniformly texture regions.



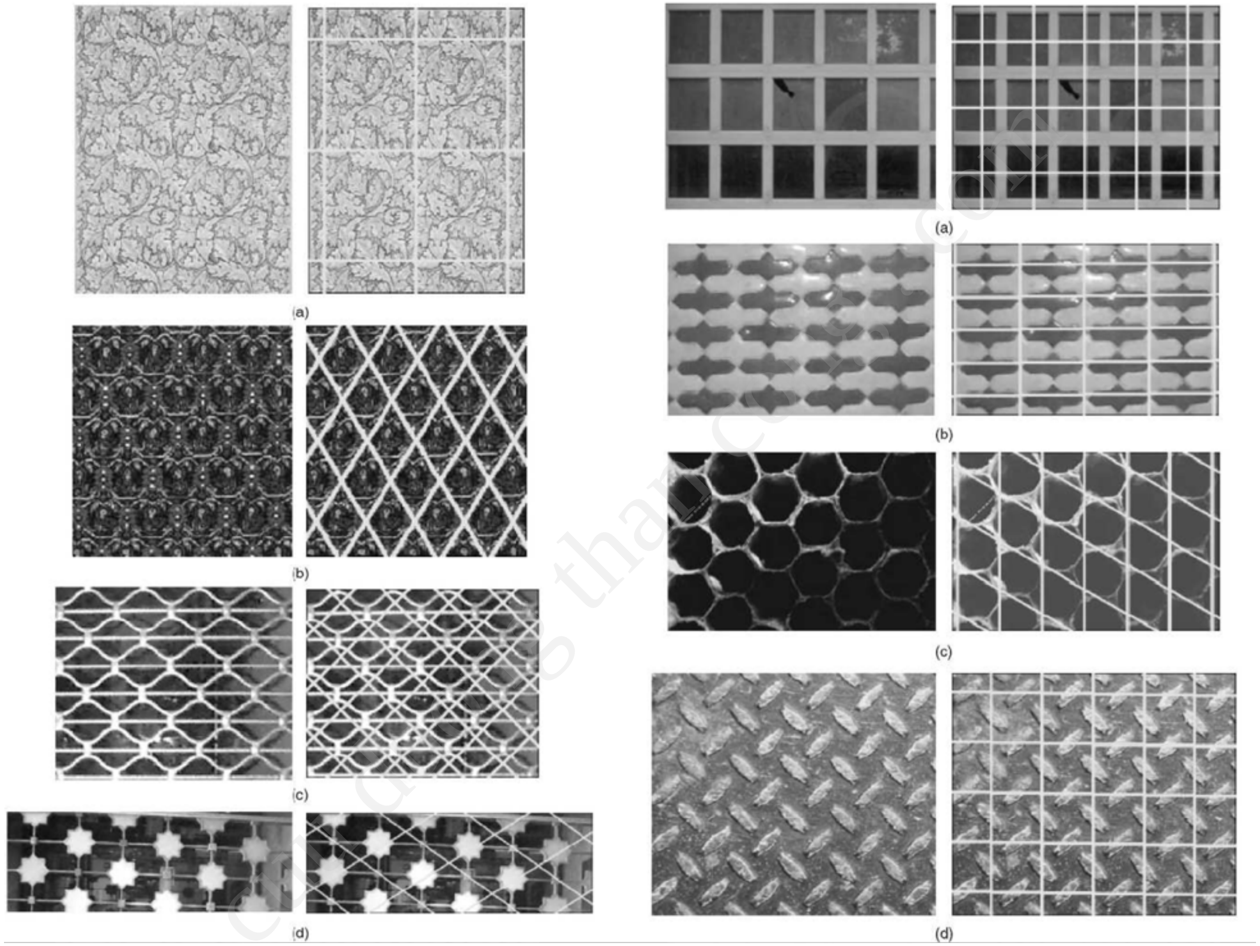
Voronoi tessellation of a set of circular texels.

6. IRD: Texture Analysis (8)

Computational model for periodic pattern perception based on frieze and wallpaper groups by Liu, Collins and Tsin:

- A frieze pattern is a 2D strip in the plane that is periodic along one dimension.
- A periodic pattern extended in two linearly independent directions to cover the 2D plane is known as a wallpaper pattern.

6. IRD: Texture Analysis (9)



Examples of periodic patterns that are extended in 2 linearly independent directions to cover the 2D plane. These patterns are also known as **wallpaper** patterns.

6. IRD: Texture Analysis (10)

❑ **Statistical approaches**

- Usually, segmenting out the texels is difficult or even impossible in real images.
- Instead, numeric quantities or statistics that describe a texture can be computed from the gray tones or colors themselves.
- This approach can be less intuitive, but is computationally efficient and often works well.

6. IRD: Texture Analysis (11)

- Some statistical approaches for texture:
  - Edge density and direction
  - Co- occurrence matrices
  - Local binary patterns
  - Statistical moments
  - Autocorrelation
  - Markov random fields
  - Autoregressive models
  - Mathematical morphology
  - Interest points
  - Fourier power spectrum
  - Gabor filters

6. IRD: Texture Analysis (12)

- **Edge density and direction:**
  - Use an edge detector as the first step in texture analysis.
  - The **number of edge pixels** in a fixed-size region tells us **how busy that region is**. The **directions of the edges** also help **characterize the texture**.

6. IRD: Texture Analysis (13)

- Edge-based texture measures:
  - **Edgeness per unit area** (measures the **busyness**)

    $F_{\text{edgeness}} = |\{ p \mid \text{gradient\_magnitude}(p) \geq \text{threshold} \}| / N$

    where N is the number of pixels of the unit area.
  - **Edge magnitude and direction histograms** (both **busyness** and **orientation**)

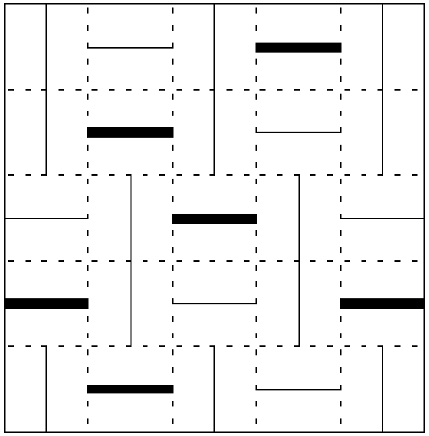
    $F_{\text{magdir}} = (H_{\text{magnitude}}, H_{\text{direction}})$

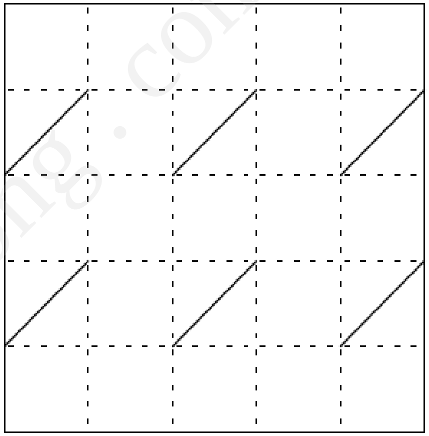
    where these are the normalized histograms of gradient magnitudes and gradient directions, respectively.
- Two n-bin histograms of two images can be compared by computing their L_1 or L_2 **distance** (for magnitude and direction histograms)

$$L_1(H_1, H_2) = \sum_{i=1}^{n} \left| H_1[i] - H_2[i] \right|$$

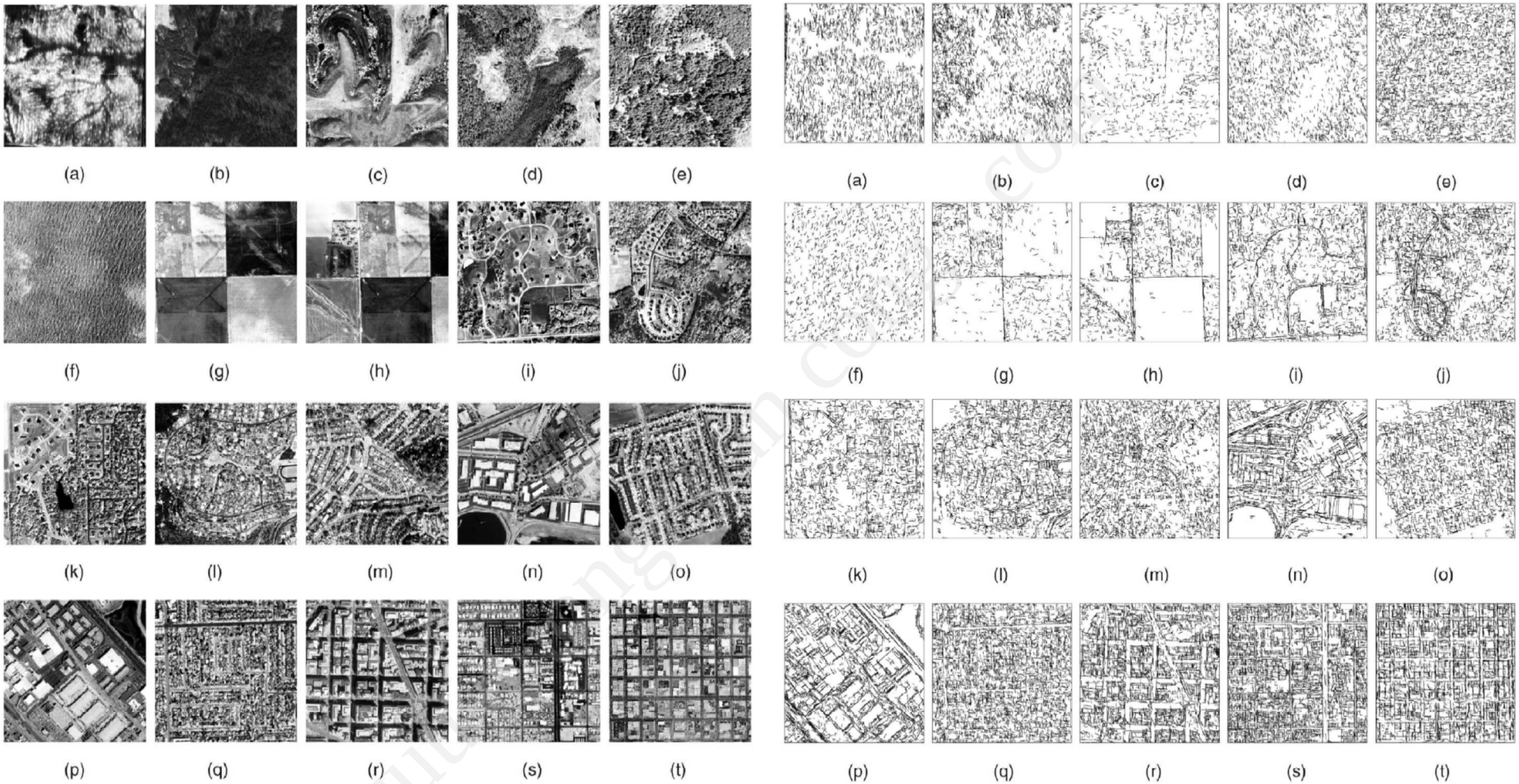
6. IRD: Texture Analysis (14)

Example:





6. IRD: Texture Analysis (15)



Satellite images sorted according to the amount of land development (left). Properties of the arrangements of line segments can be used to model the organization in an area (right).

6. IRD: Texture Analysis (16)

- **Co-occurrence matrices:**
  - Co-occurrence, in general form, can be specified in a matrix of relative frequencies $P(i, j; d, \theta)$ with which two texture elements separated by distance d at orientation θ occur in the image, one with property i and the other with property j.
  - In gray level co-occurrence, as a special case, texture elements are pixels and properties are gray levels.

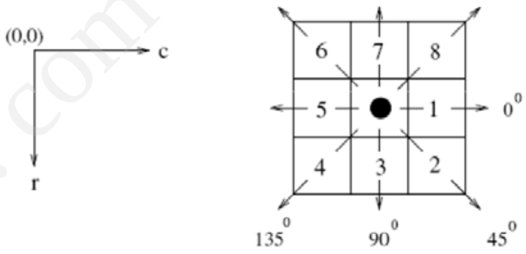
6. IRD: Texture Analysis (17)

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

(a) 4x4 image with gray levels 0-3.

		Gray Level			
		0	1	2	3
	0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
Gray	1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
Level	2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
	3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

(b) General form of co-occurrence matrices $P(i, j; d, \theta)$ for gray levels 0-3 where $\#(i, j)$ stands for number of times gray levels i and j have been neighbors.



$$P(i,j;1,0^\circ) = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

(c) $(d, \theta) = (1, 0^\circ)$

$$P(i,j;1,45^\circ) = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(d) $(d, \theta) = (1, 45^\circ)$

$$P(i,j;1,90^\circ) = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(e) $(d, \theta) = (1, 90^\circ)$

$$P(i,j;1,135^\circ) = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(f) $(d, \theta) = (1, 135^\circ)$

6. IRD: Texture Analysis (18)

– If a texture is coarse and the distance *d* used to compute the co-occurrence matrix is small compared to the sizes of the texture elements, pairs of pixels at separation *d* should usually have similar gray levels.

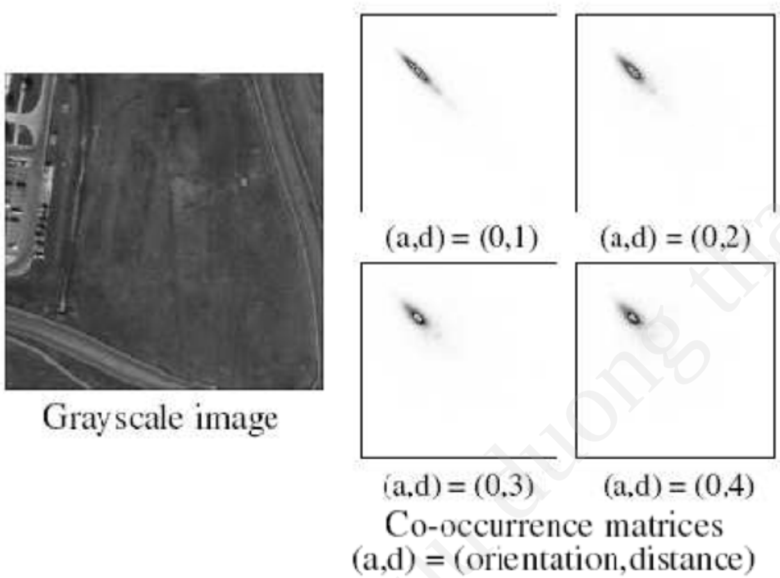
– This means that high values in the matrix $P(i, j; d, \theta)$ should be concentrated on or near its main diagonal.

– Conversely, for a fine texture, if *d* is comparable to the texture element size, then the gray levels of points separated by *d* should often be quite different, so that values in $P(i, j; d, \theta)$ should be spread out relatively uniformly.

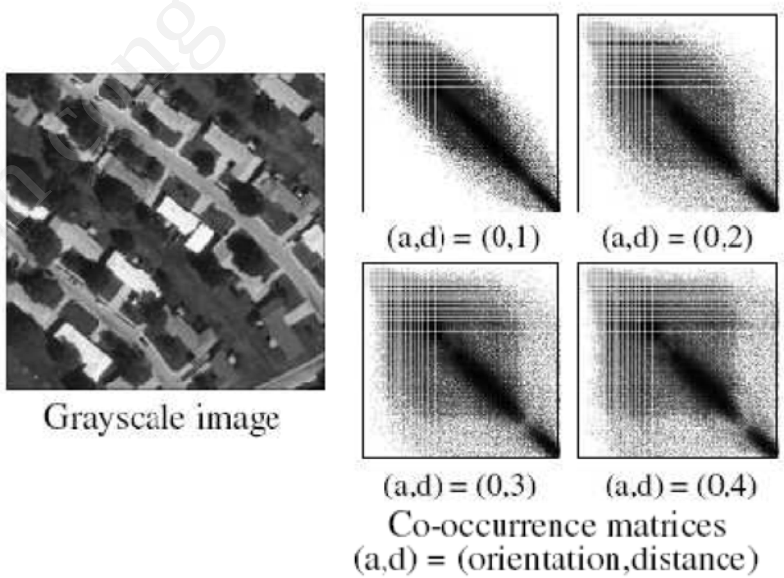
– Similarly, if a texture is directional, i.e., coarser in one direction than another, the degree of spread of the values about the main diagonal in $P(i, j; d, \theta)$ should vary with the orientation θ.

6. IRD: Texture Analysis (19)

- Thus texture directionality can be analyzed by comparing spread measures of $P(i, j; d, \theta)$ for various orientations.



(a) Co-occurrence matrices for an image with a small amount of local spatial variations.



(b) Co-occurrence matrices for an image with a large amount of local spatial variations.

6. IRD: Texture Analysis (20)

- The spatial relationship can also be specified as a displacement vector d with (dr, dc), where dr is a displacement in rows and dc is a displacement in columns. Let V be the set of gray levels. The **gray-level co-occurrence matrix** C_d for image I is defined by

$$C_d(i,j) = \left|\left\{(r,c) \mid I(r,c) = i \quad \text{and} \quad I(r+dr, c+dc) = j\right\}\right|$$

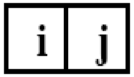
6. IRD: Texture Analysis (21)

Example:

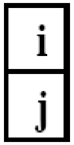
$$I = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 2 & 2 \\ \hline 0 & 0 & 2 & 2 \\ \hline \end{array}$$

Image I

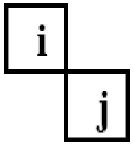
$$C_{(0,1)} = \begin{array}{cc|ccc|} & & & j & \\ & & 0 & 1 & 2 \\ \hline & 0 & 4 & 0 & 2 \\ i & 1 & 2 & 2 & 0 \\ & 2 & 0 & 0 & 2 \\ \hline \end{array}$$



$$C_{(1,0)} = \begin{array}{c|ccc|} & & j & \\ & 0 & 1 & 2 \\ \hline 0 & 4 & 0 & 2 \\ 1 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \\ \hline \end{array}$$



$$C_{(1,1)} = \begin{array}{c|ccc|} & & j & \\ & 0 & 1 & 2 \\ \hline 0 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ \hline \end{array}$$



6. IRD: Texture Analysis (22)

– There are two important variations of the standard gray-tone co-occurrence matrix. The first is the **normalized gray-tone co-occurrence matrix** N_d defined by

$$N_d(i,j) = \frac{C_d(i,j)}{\sum_i \sum_j C_d(i,j)}$$

which normalizes the co-occurrence values to lie between zero and one and allows them to be thought of as probabilities in a large matrix. The second is the **symmetric gray-level co-occurrence matrix** $S_d(i, j)$ defined by

$$S_d(i,j) = C_d(i,j) + C_{-d}(i,j)$$

which groups pairs of symmetric adjacencies.

6. IRD: Texture Analysis (23)

- In order to use the information contained in co-occurrence matrices, Haralick et al. defined 14 statistical features that capture textural characteristics such as **homogeneity**, **contrast**, **organized structure**, and **complexity**, etc. as

6. IRD: Texture Analysis (24)

$$Energy = \sum_i \sum_j N_d^2(i,j)$$

$$Entropy = -\sum_i \sum_j N_d(i,j) \log_2 N_d(i,j)$$

$$Contrast = \sum_i \sum_j (i-j)^2 N_d(i,j)$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i,j)}{1+|i-j|}$$

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d(i,j)}{\sigma_i \sigma_j}$$

6. IRD: Texture Analysis (25)

where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums $N_d(i)$ and $N_d(j)$ defined by

$$N_d(i) = \sum_j N_d(i, j)$$

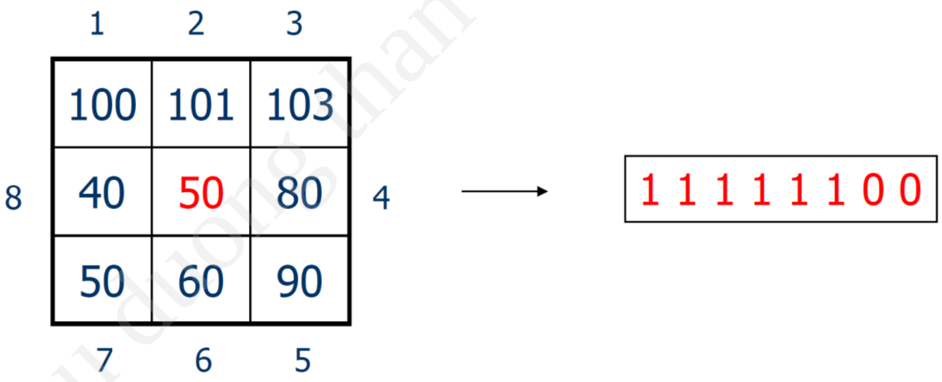
$$N_d(j) = \sum_i N_d(i, j)$$

- Zucker and Terzopoulos suggested using a Chi-square statistical test to select the values of d that have the most structure for a given class of images:

$$\chi^2(d) = \left(\sum_i \sum_j \frac{N_d^2(i, j)}{N_d(i) N_d(j)} - 1 \right)$$

6. IRD: Texture Analysis (26)

- **Local binary patterns (LBP):**
  - For each pixel *p*, create an 8-bit number $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8$, where $b_i = 0$ if neighbor *i* has value less than or equal to *p*'s value, and 1 otherwise.
  - Represent the texture in the image (or a region) by the histogram of these numbers.



100	101	103
40	50	80
50	60	90

1 1 1 1 1 1 0 0

  - Two images or regions are compared by computing the L_1 distance between their histograms as defined in the method of Edge density and direction.

6. IRD: Texture Analysis (27)

- **Autocorrelation:**

  The autocorrelation function of an image can be used to

  – detect repetitive patterns of texture elements, and

  – describe the fineness/coarseness of the texture.

  The autocorrelation function $\rho(dr, dc)$ for displacement $d = (dr, dc)$ is given by

$$\rho(dr,dc) = \frac{\sum_{r=0}^{N}\sum_{c=0}^{N} I(r,c)I(r+dr,c+dc)}{\sum_{r=0}^{N}\sum_{c=0}^{N} I^2(r,c)}$$

6. IRD: Texture Analysis (28)

Interpreting autocorrelation:

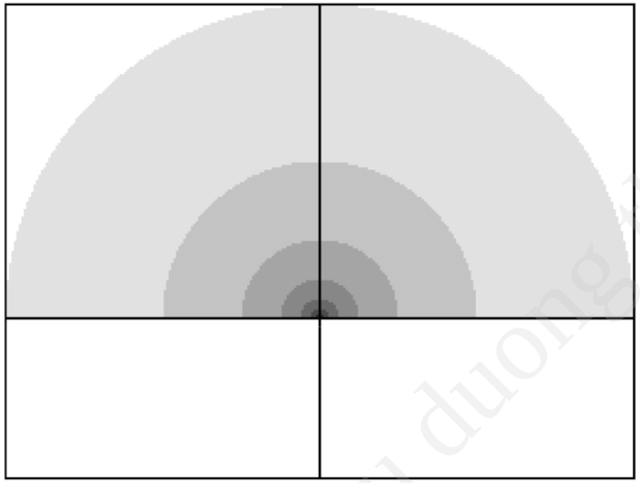
- Coarse texture function drops off slowly.
- Fine texture function drops off rapidly. It can drop differently for *r* and *c*.
- Regular textures function will have peaks and valleys; peaks can repeat far away from [0,0].
- Random textures only peak at [0,0]; breadth of peak gives the size of the texture.

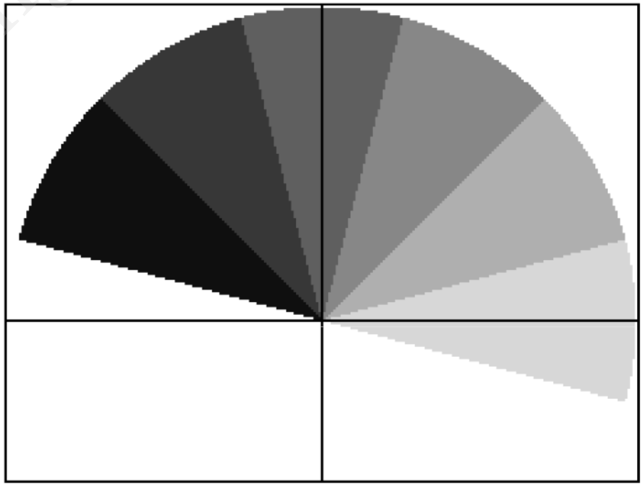
6. IRD: Texture Analysis (29)

- **Fourier power spectrum:**
  - The autocorrelation function is related to the power spectrum of the Fourier transform.
  - The power spectrum contains texture information because
    - ✓ prominent peaks in the spectrum give the principal direction of the texture patterns,
    - ✓location of the peaks gives the fundamental spatial period of the patterns.

6. IRD: Texture Analysis (30)

– The power spectrum, represented in polar coordinates, can be integrated over regions bounded by circular rings (for frequency content) and wedges (for orientation content).

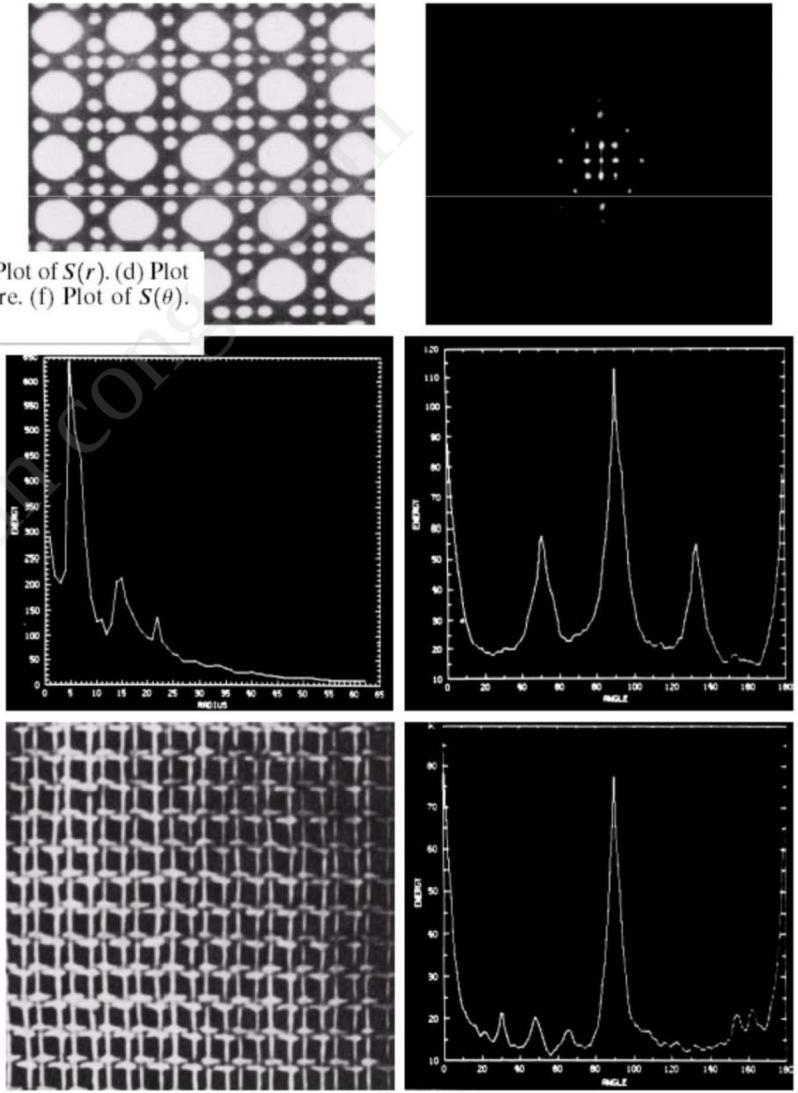




$$x_i = \sum_{r=r_i}^{r_{i+1}} \sum_{\theta=0}^{\pi} s(r,\theta)$$

$$y_i = \sum_{\theta=\theta_i}^{\theta_{i+1}} \sum_{r=1}^{r_{\max}} s(r,\theta)$$

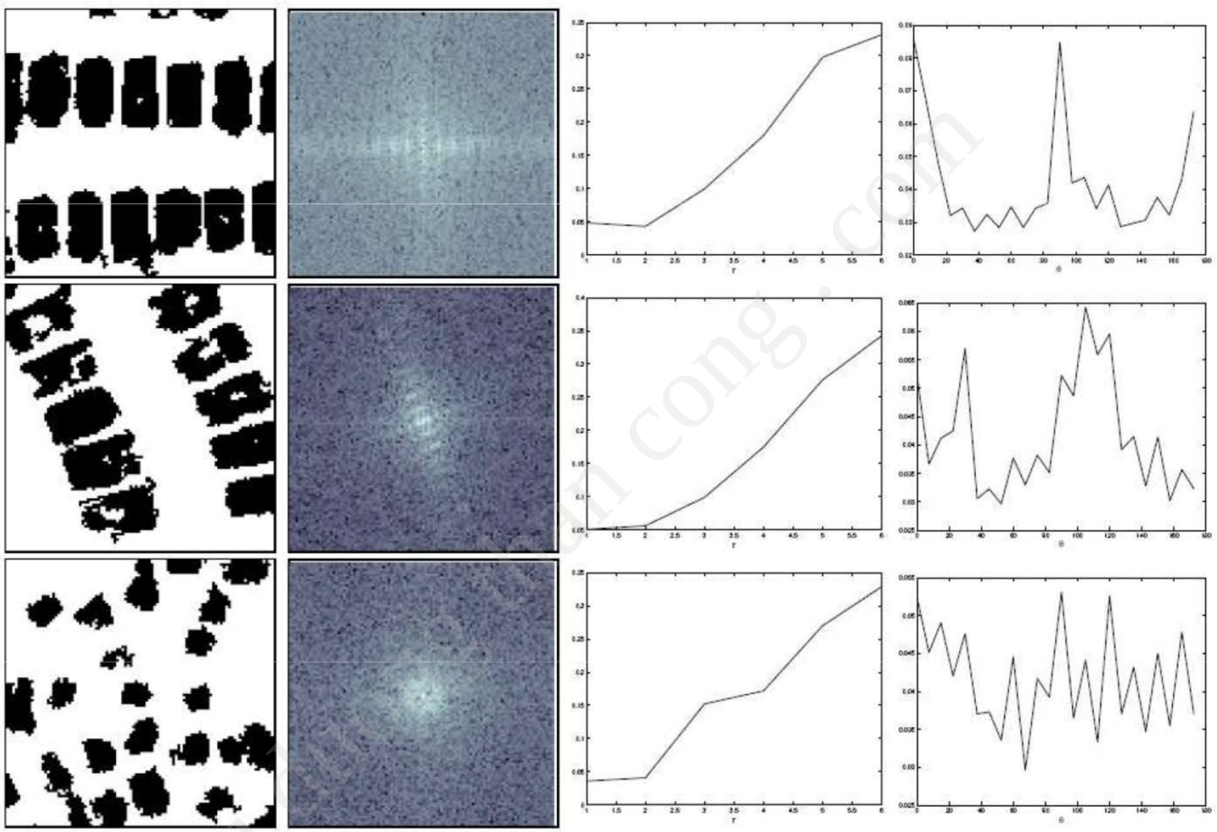
6. IRD: Texture Analysis (31)



a b
c d
e f

FIGURE 11.24 (a) Image showing periodic texture. (b) Spectrum. (c) Plot of $S(r)$. (d) Plot of $S(\theta)$. (e) Another image with a different type of periodic texture. (f) Plot of $S(\theta)$. (Courtesy of Dr. Dragana Brzakovic, University of Tennessee.)

6. IRD: Texture Analysis (32)



Example building groups (first column), Fourier spectrum of these images (second column), and the corresponding ring- and wedge-based features (third and fourth columns). X-axes represent the rings in the third column and the wedges in the fourth column plots. The peaks in the features correspond to the periodicity and directionality of the buildings, whereas no dominant peaks can be found when there is no regular building pattern.

6. IRD: Texture Analysis (33)

- **Gabor filters:**
  - The Gabor representation has been shown to be optimal in the sense of minimizing the joint two dimensional uncertainty in dimensional space and frequency.
  - These filters can be considered as orientation and scale tunable edge and line detectors.
  - Fourier transform achieves localization in either spatial or frequency domain but the Gabor transform achieves simultaneous localization in both spatial and frequency domains.

6. IRD: Texture Analysis (34)

– A 2D Gabor function and its Fourier transform are given as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right]$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$

6. IRD: Texture Analysis (35)

- Let U_l and U_h denote the lower and upper center frequencies of interest, K be the number of orientations, and S be the number of scales, the filter parameters can be selected as

$$a = (U_h / U_l)^{\frac{1}{S-1}}, \quad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{2\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}}$$

where $W = U_h$ and $m = 0, 1, \ldots, S\text{-}1$.

6. IRD: Texture Analysis (36)

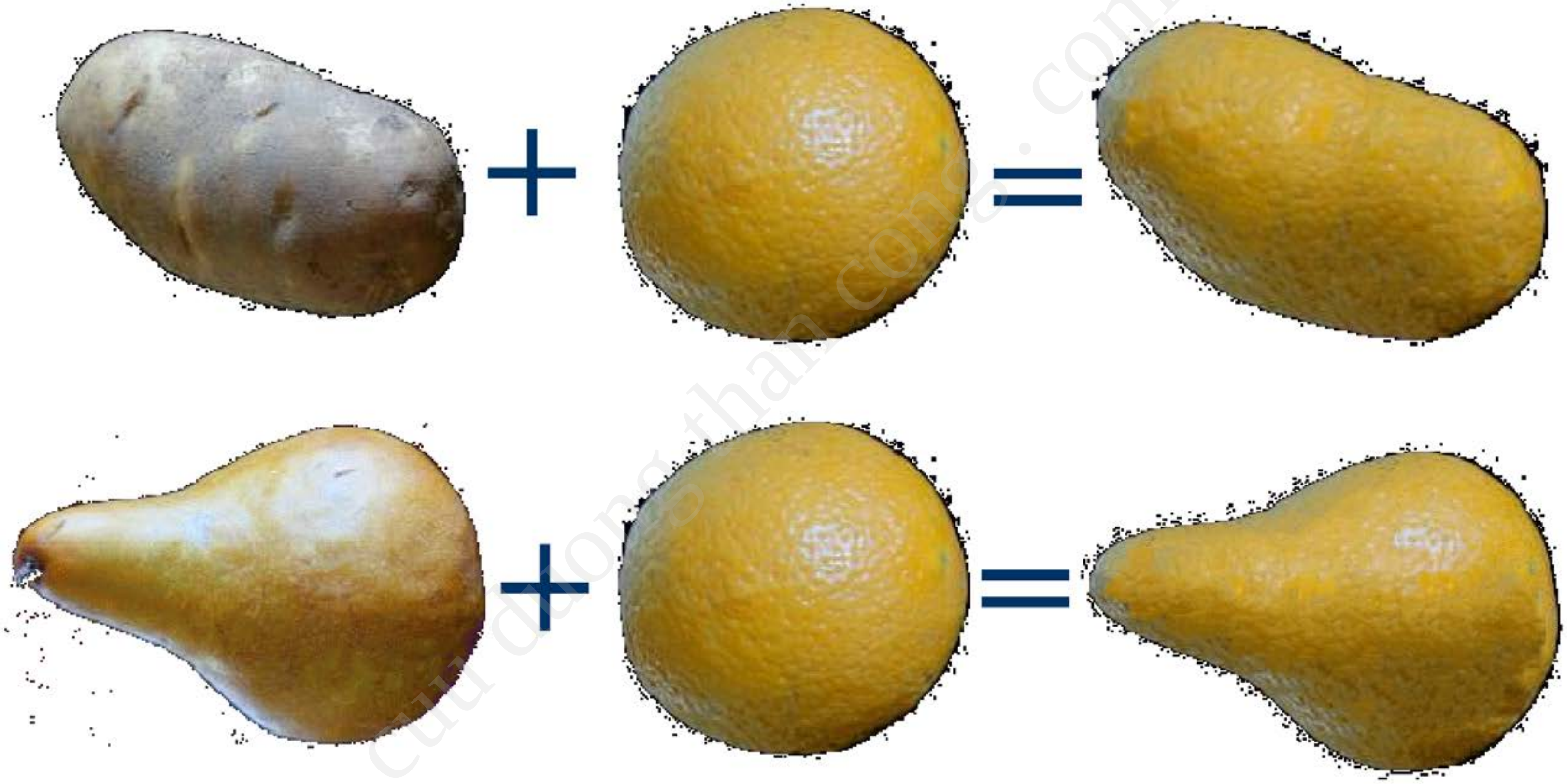
Gabor filter responses for a satellite image

6. IRD: Texture Synthesis (1)

- Goal of **texture analysis**: compare textures and decide if they are similar.
- Goal of **texture synthesis**: construct large regions of texture from small example images.
- It is an important problem for rendering in computer graphics.
- Strategy: to think of a texture as a sample from some probability distribution and then to try and obtain other samples from that same distribution.

6. IRD: Texture Synthesis (2)

6. IRD: Texture Synthesis (3)



6. IRD: Texture Synthesis (4)