Feature Extraction

By

Dr. Hetal Patel , Professor, ADIT

Motivation

- One of the major concern of image processing is image (object) recognition
  - Objects are represented as a collection of pixels in an image

- Our Task: To describe the region based on the chosen representation

Image acquisition => digital image

↓

Preprocessing => better image

↓

Segmentation => basic features

↓

Representation and description => advanced features

↓

Object recognition

REPRESENTATION

- Representation means that we make the object information more accessible for computer-interpretation
- Two types of representation
  - Using boundary (External characteristics)
  - Using pixels of region (Internal characteristics)

DESCRIPTION

- Description means that we quantify our representation of the object
- Boundary Descriptors
  - Geometrical descriptors: Diameter, perimeter, eccentricity, curvature
  - Shape Numbers
  - Fourier Descriptors
  - Statistical Moments
- Regional Descriptors
  - Geometrical descriptors: Area, compactness, Euler number
  - Texture
  - Moments of 2D Functions

Desirable properties of descriptors

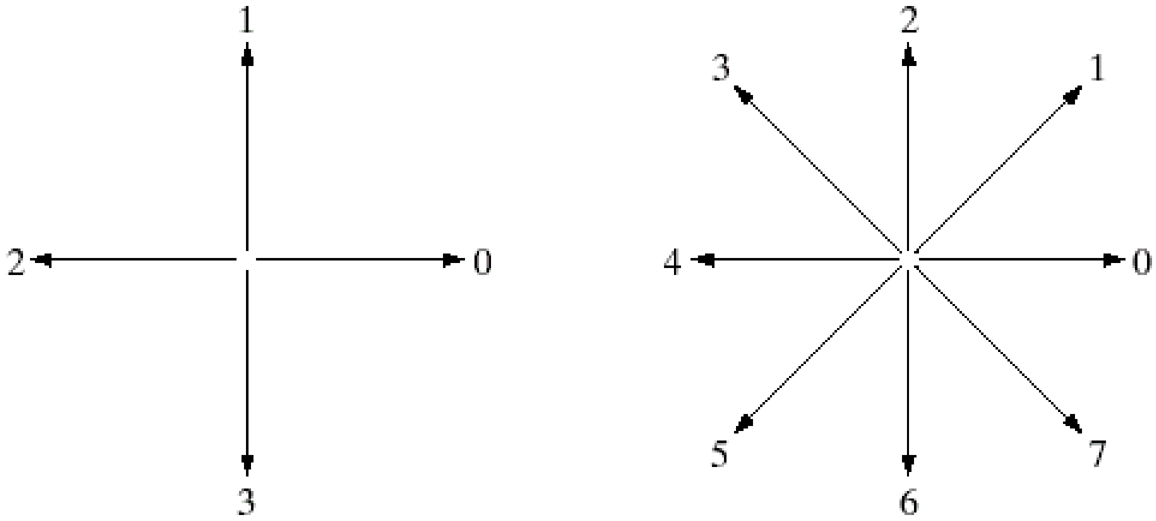
- They should define a complete set
  - Two objects must have the same descriptors if and only if they have the same shape
- They should be invariant to Rotation, Scaling and Translation (RST)
- They Should be a compact set
  - A descriptor should only contain information about what makes an object unique, or different from the other objects.
  - The quantity of information used to describe this characterization should be less than the information necessary to have a complete description of the object itself
- They should be robust
  - Work well against noise and distortion
- They should have low computational complexity

Chain Codes

- Chain code encodes a boundary of an object as connected sequence of straight line segments of a specific length and direction
- By allocating numbers based on directions, the boundary of an object is reduced to a sequence of numbers
- Two types of chain codes based on 4-neighbourhood or

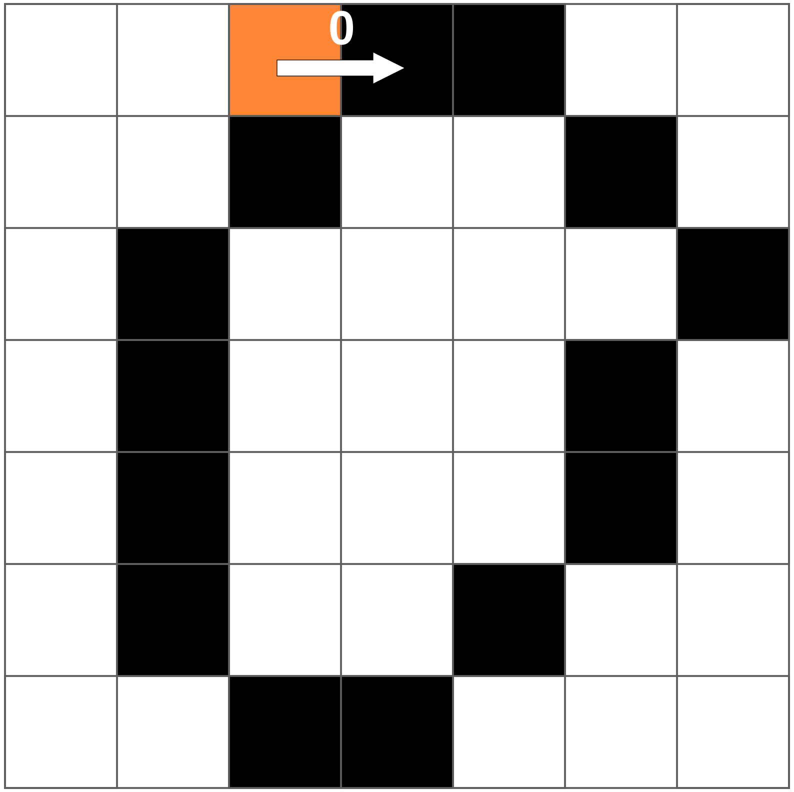
a b

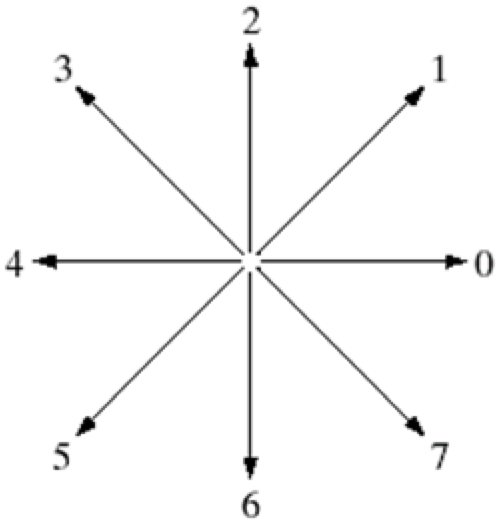
FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



CHAIN CODES

- Steps for construction chain codes
  - Select some starting point of the boundary and represent it by its absolute coordinates in the image
  - Represent every consecutive point by a chain code showing transition needed to go from current point to next point on the boundary
  - Stop if the next point is the initial point or the end of the boundary

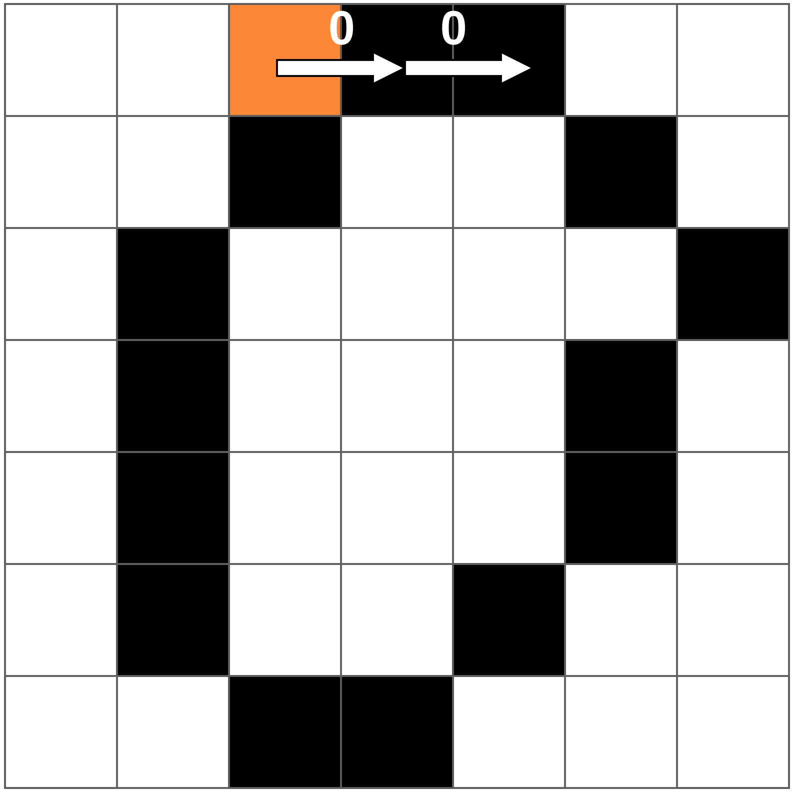




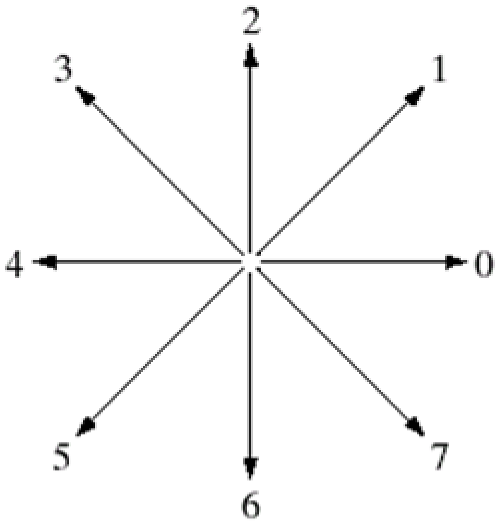
Chain Code:

0

BOUNDARY

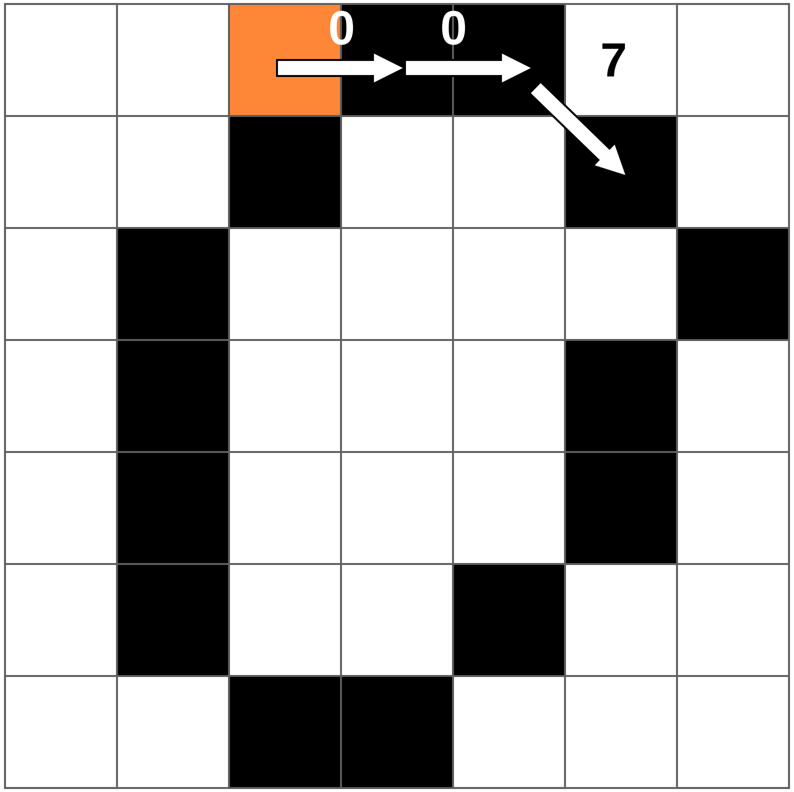


BOUNDARY

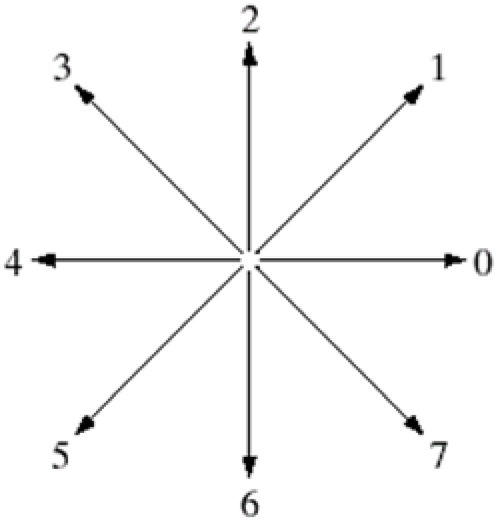


Chain Code:

0, 0

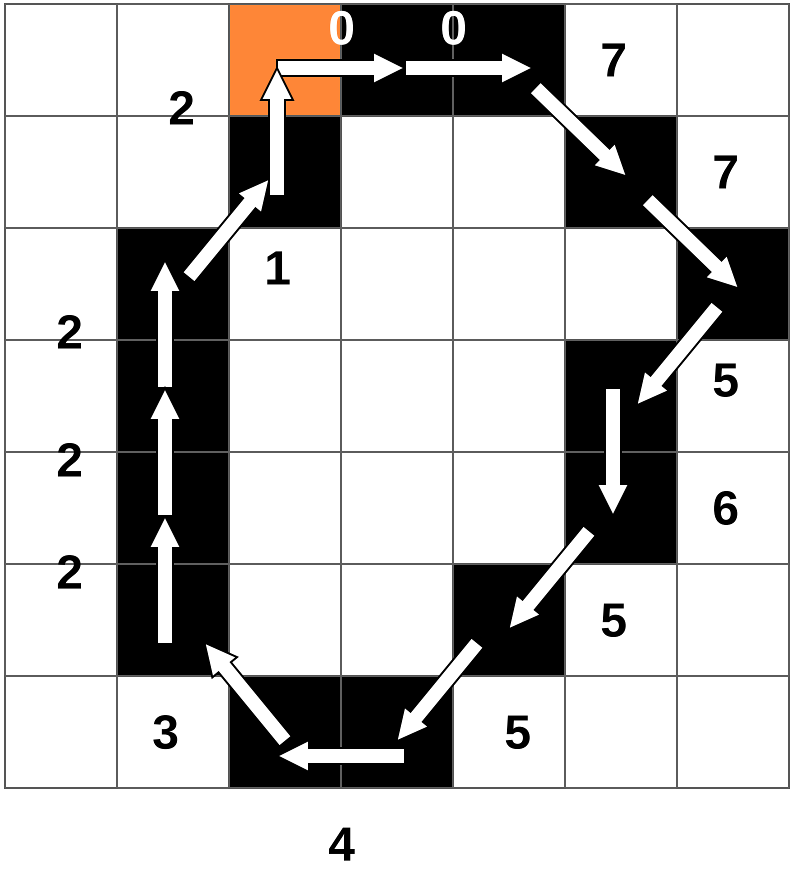


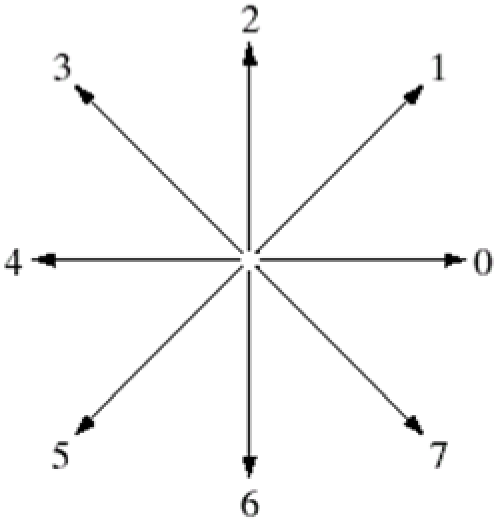
BOUNDARY



Chain Code:

0, 0, 7





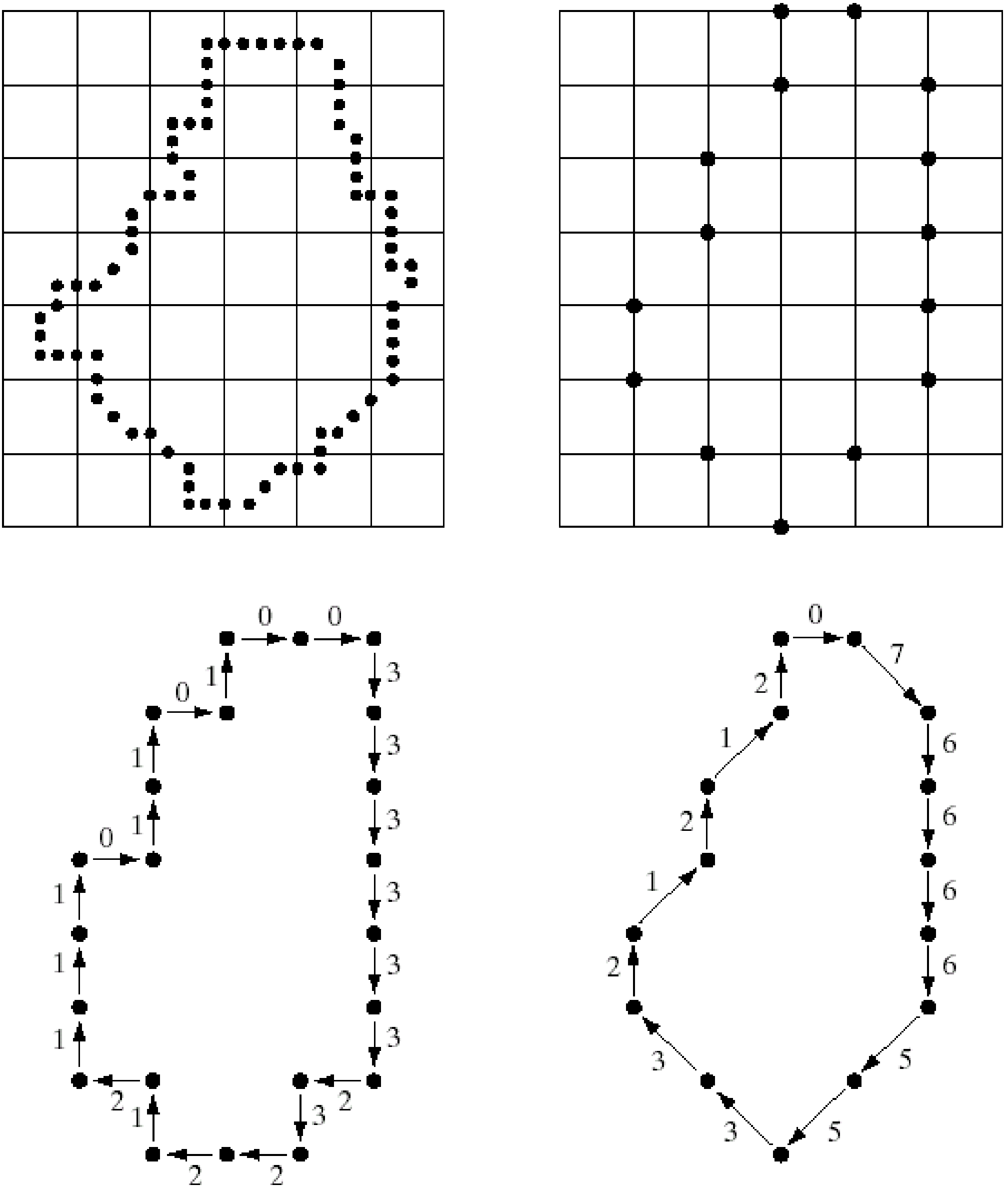
Chain Code:

0, 0, 7, 7, 5, 6, 5, 5
4, 3, 2, 2, 2, 1, 2

Chain Codes

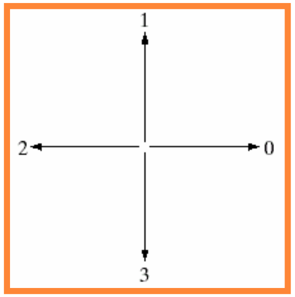
- Advantages
  - Preserves the information of interest
  - Provides good compression of boundary description
  - They are translation invariant
- Problems
  - Long chains of codes
  - No invariance to rotation and scale
  - Sensitive to noise
- Solution
  - Re-sample the image to a lower resolution before calculating the code

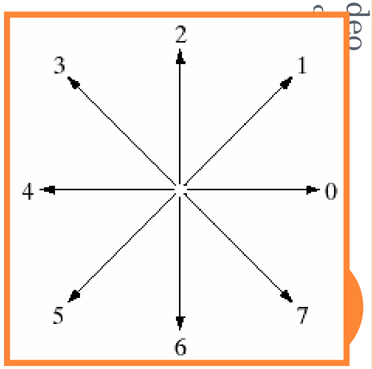
Chain Codes



a b
c d

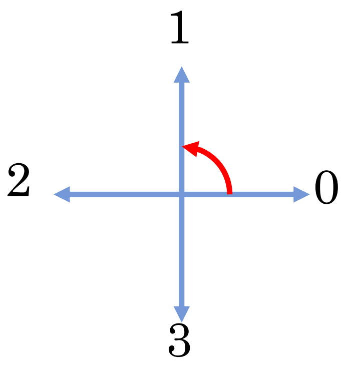
FIGURE 11.2 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 4-directional chain code. (d) 8-directional chain code.





CHAIN CODES

- Problem
- a chain code sequence depends on a starting point.
- Solution
  - treat a chain code as a circular sequence and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude after circular shift
    - 2 2 3 0 □ 0 2 2 3
  - The first difference of a chain code: counting the number of direction change (in counterclockwise) between 2 adjacent elements of the code



Chain code : The first difference

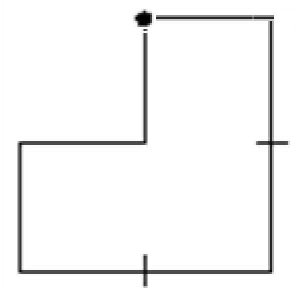
0 □ 1	1
0 □ 2	2
0 □ 3	3
1 □ 0	3
2 □ 1	3
3 □ 2	3

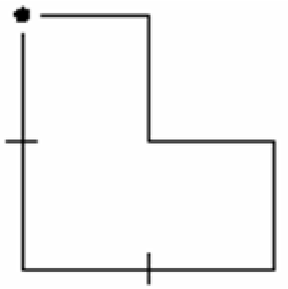
Example:

- a chain code: 10103322
- The first difference = 3133030
- Treating a chain code as a circular sequence, we get the first difference = 33133030

Shape Number as Boundary Descriptor

- The shape number of a boundary is defined as the first difference of smallest magnitude
- The order *n* of a shape number is defined as the number of digits in its representation





3 2 3 0 0 1 1 2	4-dir Chain Code	3 3 0 0 1 2 1 2
1 3 1 1 0 1 0 1	First Difference	1 0 1 0 1 1 3 1
0 1 0 1 1 3 1 1	Shape Number	0 1 0 1 1 3 1 1

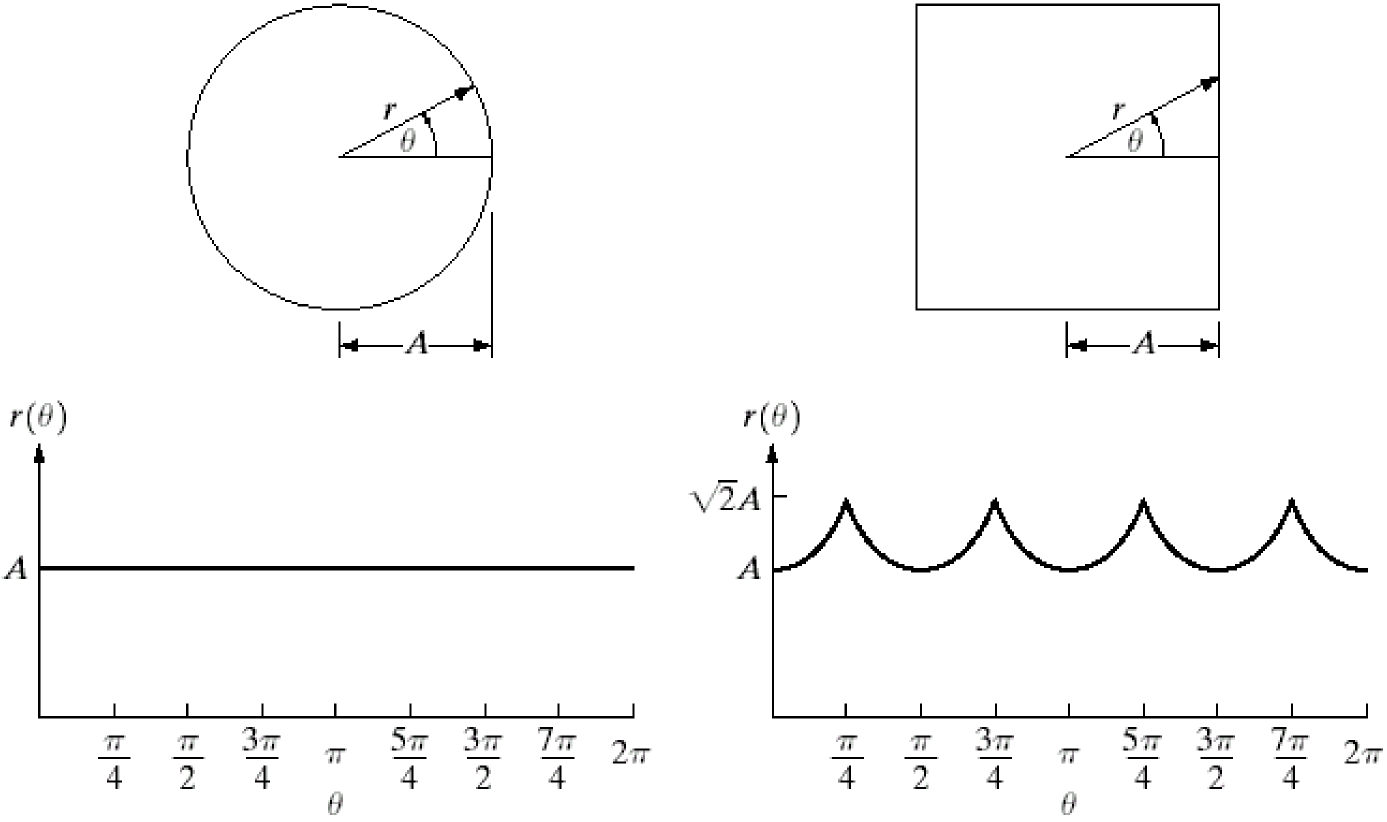
SIGNATURES

- The idea behind a signature is to convert a two dimensional boundary into a representative one dimensional function

6/22/2015

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 $\pi/4 < \theta \leq \pi/2$.



SIGNATURES

- Signatures are invariant to location, but will depend on rotation and scaling.
  - Rotation invariance can be improved by selecting a unique starting point (e.g. based on major axis)
  - Scale invariance can be achieved by normalizing amplitude of signature (divide by variance)

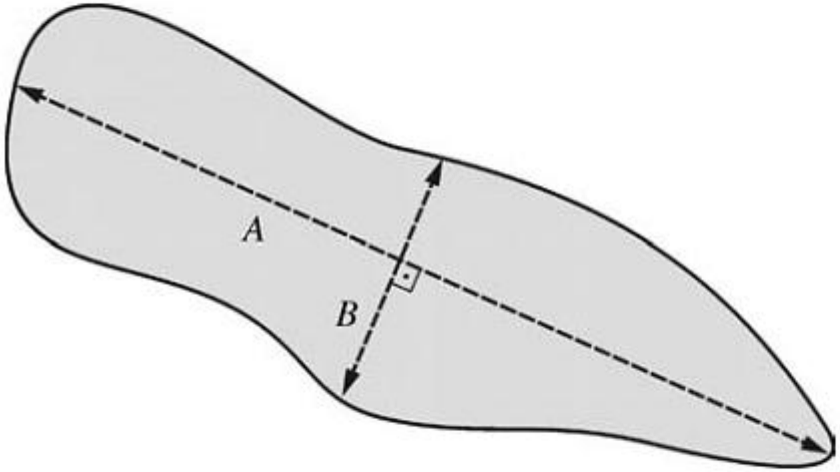
BOUNDARY DESCRIPTORS

- There are several simple geometric measures that can be useful for describing a boundary
- Length
  - the number of pixels along a boundary gives a rough approximation of its length
  - For a chain-coded curve with unit spacing
    - Length = the number of vertical and horizontal components + √2 * the number of diagonal components
- Diameter (Major Axis)

$$Diam(B) = \max_{i,j}[D(p_i, p_j)]$$

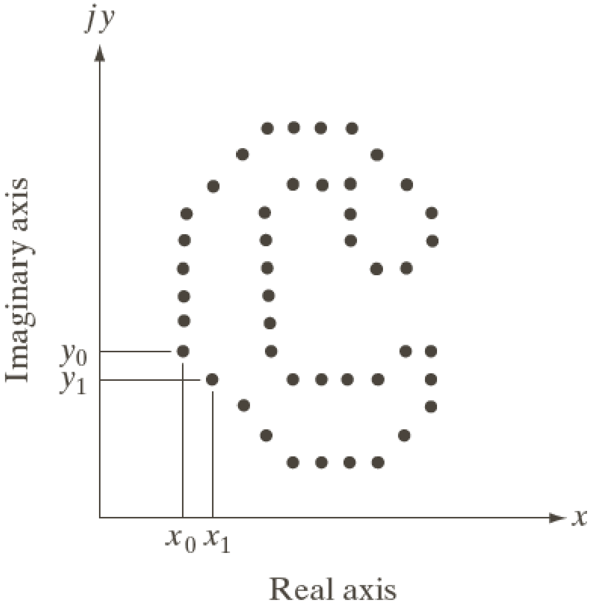
BOUNDARY DESCRIPTORS

- Minor Axis
  - the line perpendicular to the major axis
- Eccentricity
  - Ratio of major axis to minor axis



FOURIER DESCRIPTORS

- This is a way of using the Fourier transform to analyze the shape of a boundary.
  - The *x*-*y* coordinates of the boundary are treated as the real and imaginary parts of a complex number
  - Then the list of coordinates is Fourier transformed using the DFT
  - The Fourier coefficients are called the Fourier descriptors.
  - The basic shape of the region is determined by the first several coefficients, which represent lower frequencies
  - Higher frequency terms provide information on the fine detail of the boundary



Points on the boundary can be treated as a complex number $s(k)=x(k)+jy(k)$

$$a(u)=\frac{1}{N}\sum_{k=0}^{N-1} s(k)\exp[-j2\pi uk/N]$$

Fourier Descriptors

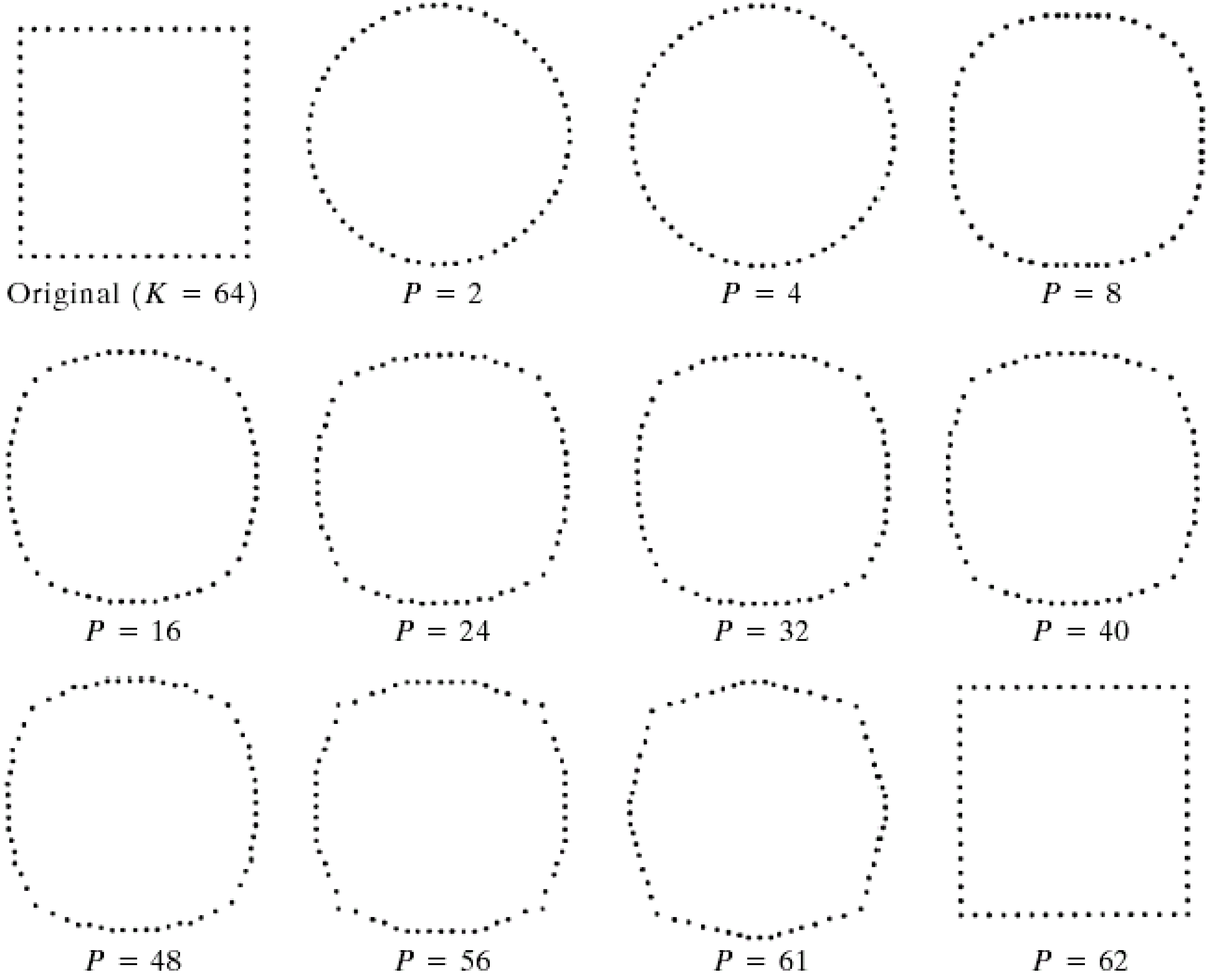
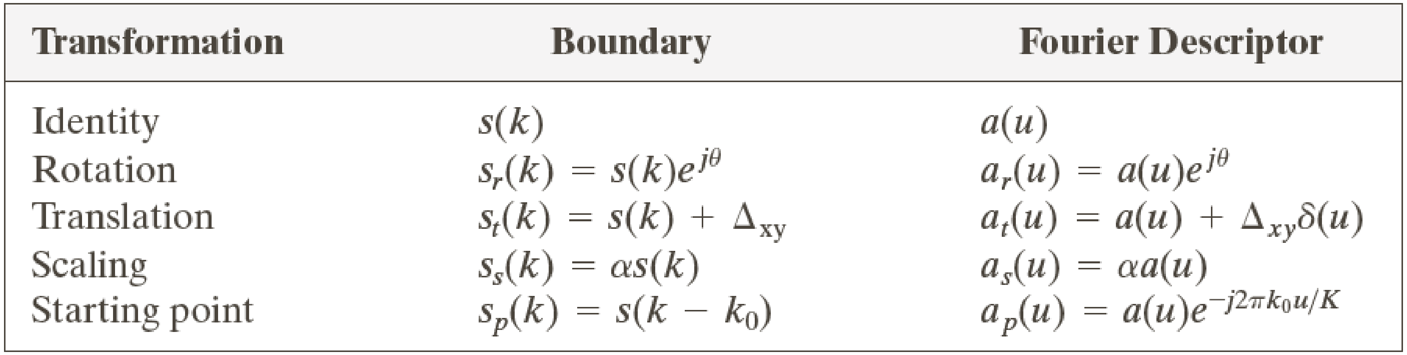


FIGURE 11.14 Examples of reconstruction from Fourier descriptors. P is the number of Fourier coefficients used in the reconstruction of the boundary.

Properties of Fourier Descriptors

- Translation
  - adding some constant to values of all coordinates
  - So, we only change the zero-frequency component. (Mean position only nothing about the shape)
  - So, except for the zero-frequency component, Fourier Descriptors are translation invariant.
- Rotation
  - rotation in the complex plane by angle θ is multiplication by exp(jθ)
  - So, rotation about the origin of the coordinate system only multiplies the Fourier descriptors by exp(jθ)
- Scaling
  - It means multiplying *x*(k) and *y*(*k*) by some constant.
  - Hence, Fourier descriptors are scaled by the same constant (Again, we ignore the value of the zero-frequency component)
- Starting Point
  - Changing starting point is equivalent to translation of the one-dimensional signal *s*(*k*) along the *k* dimension
  - Hence, translation in the spatial domain (in this case, *k*) is a phase-shift in the transform. So, the magnitude part of a(*u*)is invariant to the start point, and the phase part shifts accordingly



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	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

Regional Descriptors

- Area
  - Total number of pixels in the object
- Perimeter
  - length of boundary of a region
- Compactness (Shape Factor)
  - $(\text{Perimeter})^2/\text{Area}$
- Thinness Ratio
  - $(4\pi \times \text{Area})/(\text{Perimeter})^2$
  - Indicates roundness of the object
- The mean and median of the gray levels
- The minimum and maximum gray-level values
- The number of pixels with values above and below the mean

REGIONAL DESCRIPTORS

- Topological Properties
  - the number of holes (*H*)
  - the number of connected components (*C*)

- Euler number
  - $E = C - H$

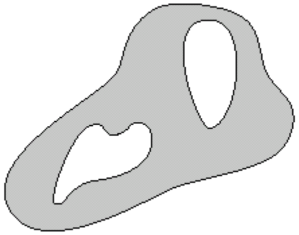


FIGURE 11.17 A region with two holes.

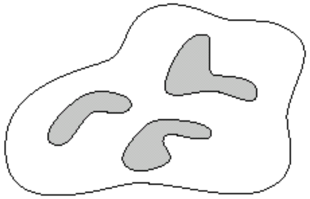
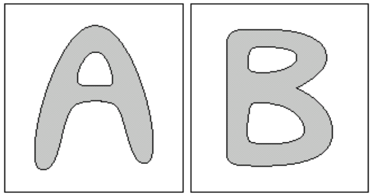


FIGURE 11.18 A region with three connected components.

E=0



E=-1

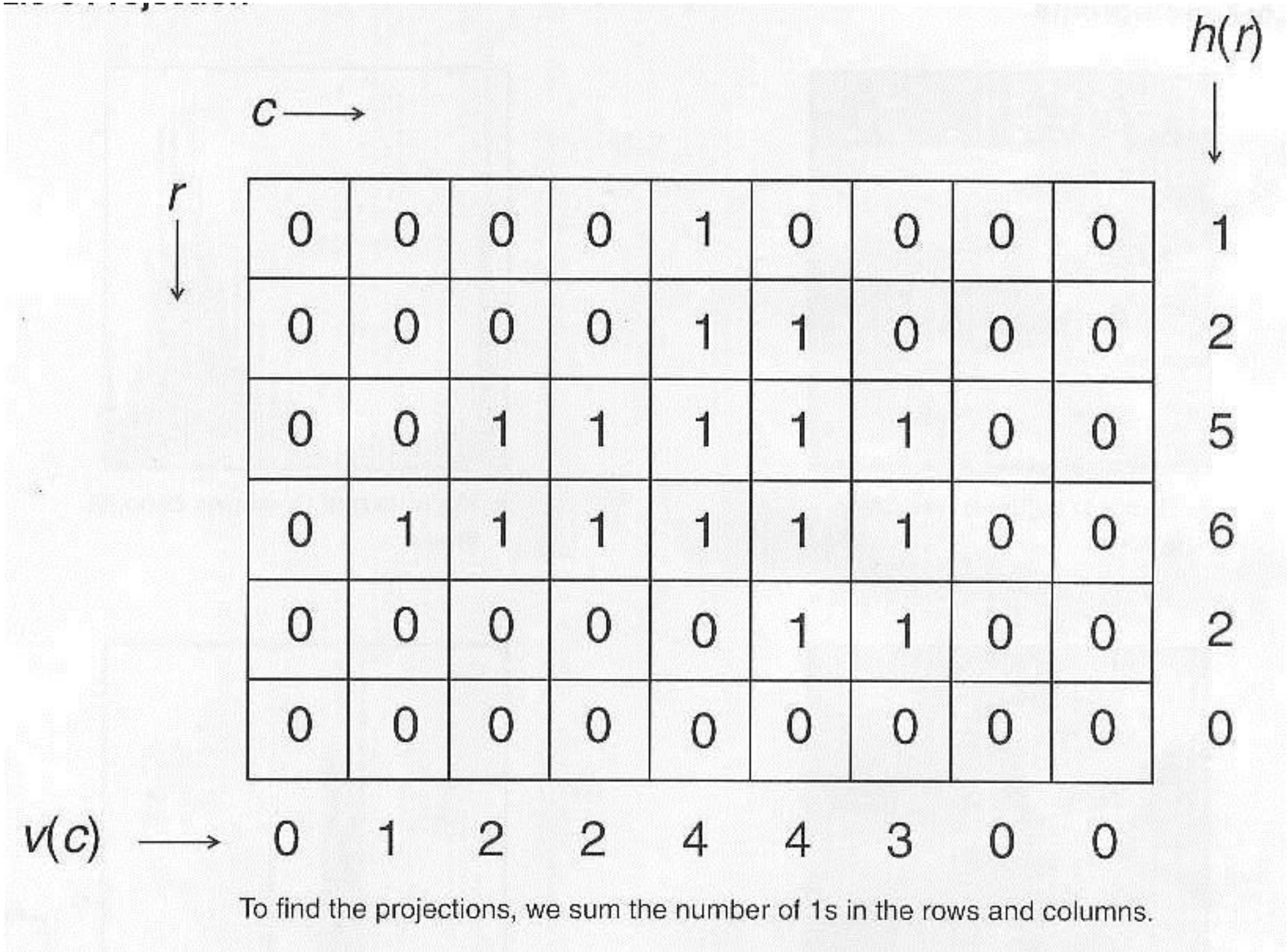
a b

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

Projection as a Descriptor

- The projection of a binary object, may provide useful information related to object’s shape.
- It can be found by summing all the pixels along the rows or columns.
  - Summing the rows give horizontal projection.
  - Summing the columns give the vertical projection.

Example



c →

r ↓

										$h(r)$ ↓
	0	0	0	0	1	0	0	0	0	1
	0	0	0	0	1	1	0	0	0	2
	0	0	1	1	1	1	1	0	0	5
	0	1	1	1	1	1	1	0	0	6
	0	0	0	0	0	1	1	0	0	2
	0	0	0	0	0	0	0	0	0	0
$v(c)$ →	0	1	2	2	4	4	3	0	0	

To find the projections, we sum the number of 1s in the rows and columns.

HISTOGRAM-BASED (STATISTICAL) FEATURES

- The simplest histogram-based descriptor is the *mean gray value of an image*, representing its average intensity *m and given by*

$$m = \sum_{j=0}^{L-1} r_j P(r_j)$$

- where r_j *is the jth gray level (out of a total of L possible values), whose probability* of occurrence is *p(rj).*

MEAN VALUE USED AS A DESCRIPTOR

- The mean gray value can also be computed directly from the pixel values from the original image *f (x, y) of size M × N as follows:*

$$m = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

- The mean is a very compact descriptor (one floating-point value per image or object) that provides a measure of the overall brightness of the corresponding image or object. It is also RST invariant.
- On the negative side, it has very limited expressiveness and discriminative power.

STANDARD DEVIATION AS A DESCRIPTOR

- The *standard deviation σ of an image is given by*

$$\sigma = \sqrt{\sum_{j=0}^{L-1} (r_j - m)^2 P(r_j)}$$

- The square of the standard deviation is the *variance, which is also known as the normalized second-order moment of the image*. The standard deviation provides a concise representation of the overall contrast.
- Similar to the mean, it is compact and RST invariant, but has limited expressiveness and discriminative power.

Skew as a descriptor

- The *skew of a histogram is a measure of its asymmetry about the mean level. It is* defined as

$$skew = \frac{1}{\sigma^3}\sum_{j=0}^{L-1}(r_j - m)^3 P(r_j)$$

- where *σ is the standard deviation given by equation* The sign of the skew indicates whether the histogram's tail spreads to the right (positive skew) or to the left (negative skew).

- The skew is also known as the *normalized third-order moment of the image.*

- If the image's mean value (*m*), *standard deviation* (*σ*), *and mode—defined as the* histogram's highest peak—are known, the skew can be calculated as follows:

$$skew = \frac{m - \ mode}{\sigma}$$

- The *energy descriptor provides another measure of how the pixel values are distributed* along the gray-level range: images with a single constant value have maximum energy (i.e., energy = 1); images with few gray levels will have higher energy than the ones with many gray levels.

Energy Descriptor

- The energy descriptor can be calculated as

$$energy = \sum_{j=0}^{L-1} \left(P(r_j)\right)^2$$

- Energy is used as a measure of uniformity.
- Histograms also provide information about the complexity of the image, in the form of *entropy descriptor. The higher the entropy, the more complex the image. Entropy* and energy tend to vary inversely with one another.

ENTROPY AS A DESCRIPTOR

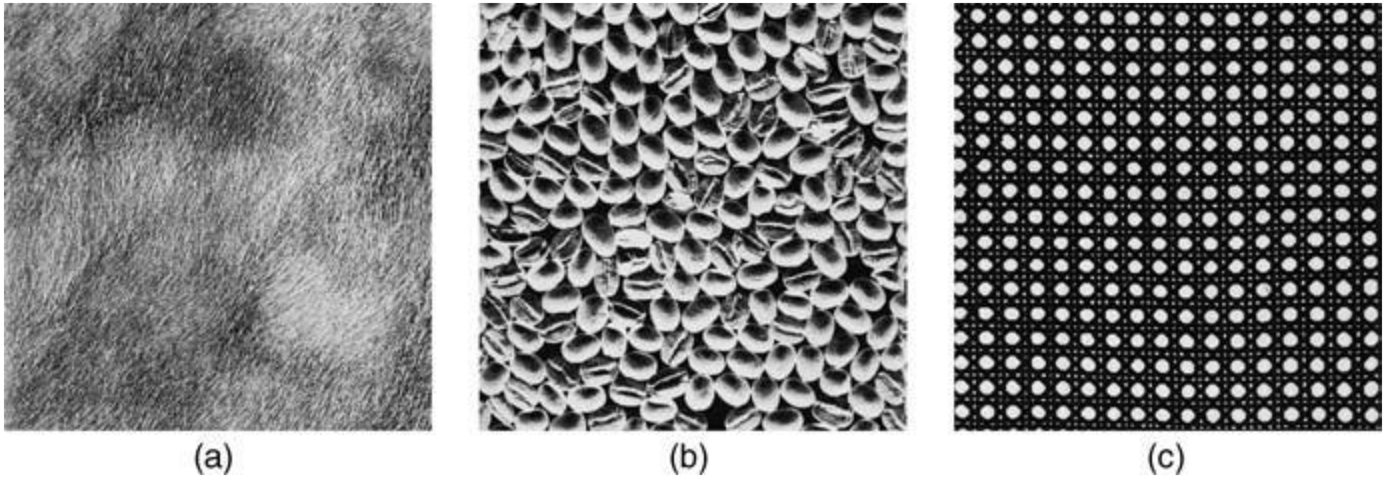
- The mathematical formulation for entropy is

$$entropy = -\sum_{j=0}^{L-1} P(r_j)\log P(r_j)$$

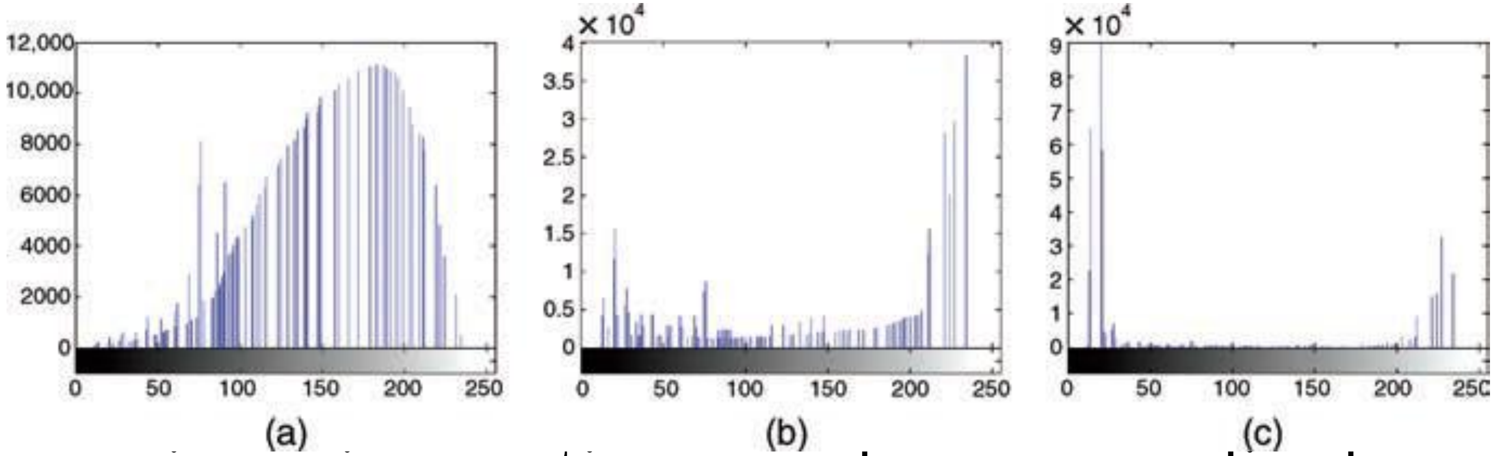
- Histogram-based features and their variants are usually employed as texture descriptors

TEXTURE FEATURES

- Texture can be a powerful descriptor of an image (or one of its regions). Although there is not a universally agreed upon definition of texture, image processing techniques usually associate the notion of texture with image (or region) properties such as *smoothness (or its opposite, roughness), coarseness, and regularity*



Texture Features



(a) (b) (c)

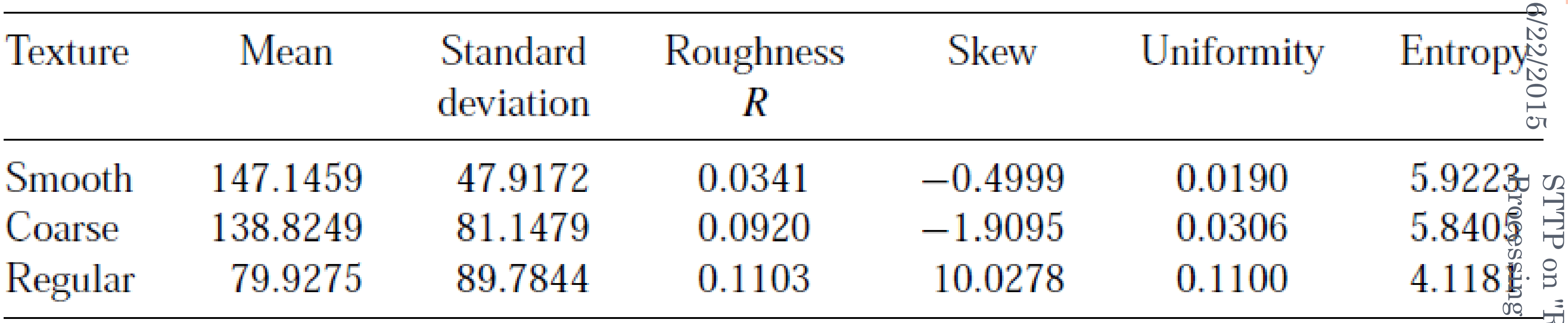
- The variance is sometimes used as a normalized descriptor of roughness (*R*), *defined as*

$$R = 1 - \frac{1}{1 + \sigma^2}$$

6/22/2015

STTP on "Recent Trends in Video Processing and Soft Computing

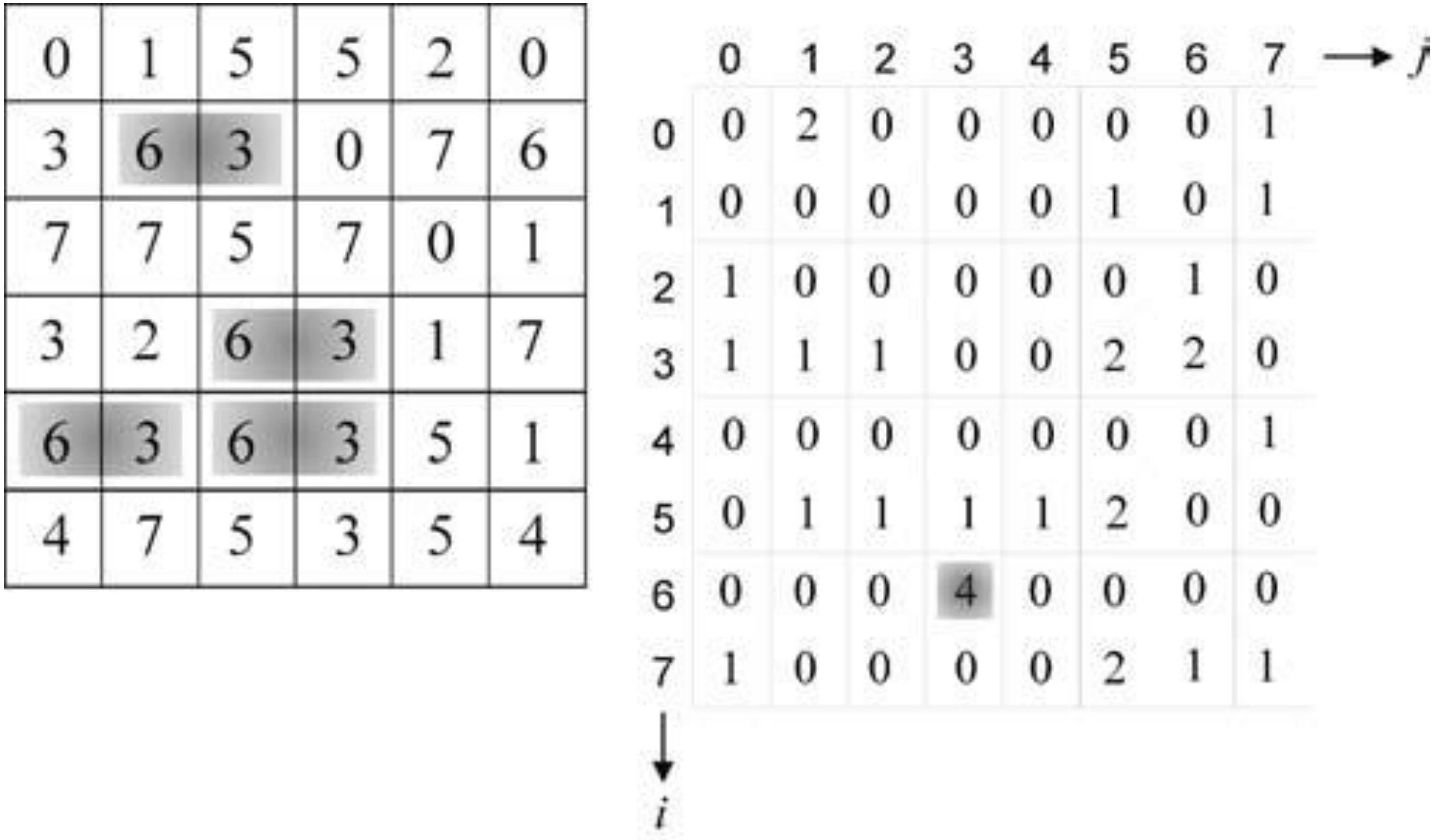
Texture Features



Texture	Mean	Standard deviation	Roughness R	Skew	Uniformity	Entropy
Smooth	147.1459	47.9172	0.0341	−0.4999	0.0190	5.9223
Coarse	138.8249	81.1479	0.0920	−1.9095	0.0306	5.8405
Regular	79.9275	89.7844	0.1103	10.0278	0.1100	4.1181

Highest uniformity has lowest entropy

- Histogram-based texture descriptors are limited by the fact that the histogram does not carry any information about the spatial relationships among pixels. One way to circumvent this limitation consists in using an alternative representation for the pixel values that encodes their relative position with respect to one another.
- One such representation is the *gray-level cooccurrence matrix* ***G, defined as a matrix whose*** element *g(i, j) represents the number of times that pixel pairs with intensities zi and zj occur in image f (x, y) in the position specified by an operator* ***d.***
- ***The vector d is*** known as *displacement vector:*



0	1	5	5	2	0
3	6	3	0	7	6
7	7	5	7	0	1
3	2	6	3	1	7
6	3	6	3	5	1
4	7	5	3	5	4

(a)

	0	1	2	3	4	5	6	7
0	0	2	0	0	0	0	0	1
1	0	0	0	0	0	1	0	1
2	1	0	0	0	0	0	1	0
3	1	1	1	0	0	2	2	0
4	0	0	0	0	0	0	0	1
5	0	1	1	1	1	2	0	0
6	0	0	0	4	0	0	0	0
7	1	0	0	0	0	2	1	1

(b)

GRAY LEVEL CO-OCCURANCE MATRIX

- The gray-level cooccurrence matrix can be normalized as follows:

$$N_g(i,j) = \frac{g(i,j)}{\sum_i \sum_j g(i,j)}$$

- where *Ng(i, j) is the normalized gray-level cooccurrence matrix. Since all values of Ng(i, j) lie between 0 and 1, they can be thought of as the probability that a pair of* points satisfying **d will have values (*zi, zj*).**

- Cooccurrence matrices can be used to represent the texture properties of an image. Instead of using the entire matrix, more compact descriptors are preferred. These are the most popular texture-based features that can be computed from a normalized gray-level cooccurrence matrix *Ng(i, j)*:

$$\text{Maximum probability} = \max_{i,j} N_g(i, j)$$

$$\text{Energy} = \sum_i \sum_j N_g^2(i, j)$$

$$\text{Entropy} = -\sum_i \sum_j N_g(i, j) \log_2 N_g(i, j)$$

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 N_g(i, j)$$

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

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

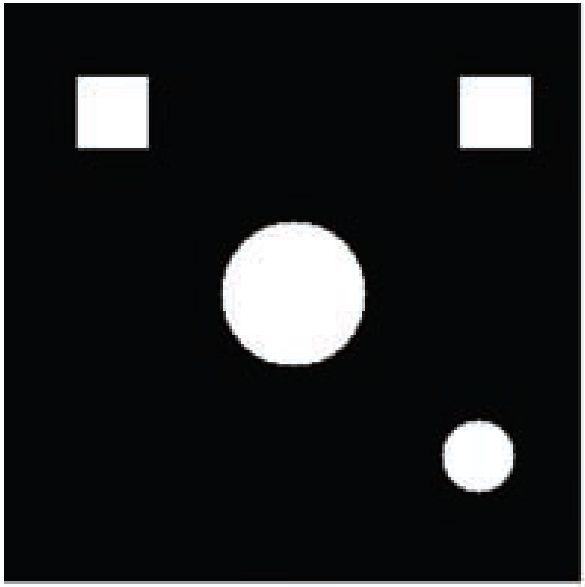
where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums $N_g(i)$ and $N_g(j)$, defined as

$$N_g(i) = \sum_i N_g(i, j) \qquad (18.35)$$

$$N_g(j) = \sum_i N_g(i, j) \qquad (18.36)$$

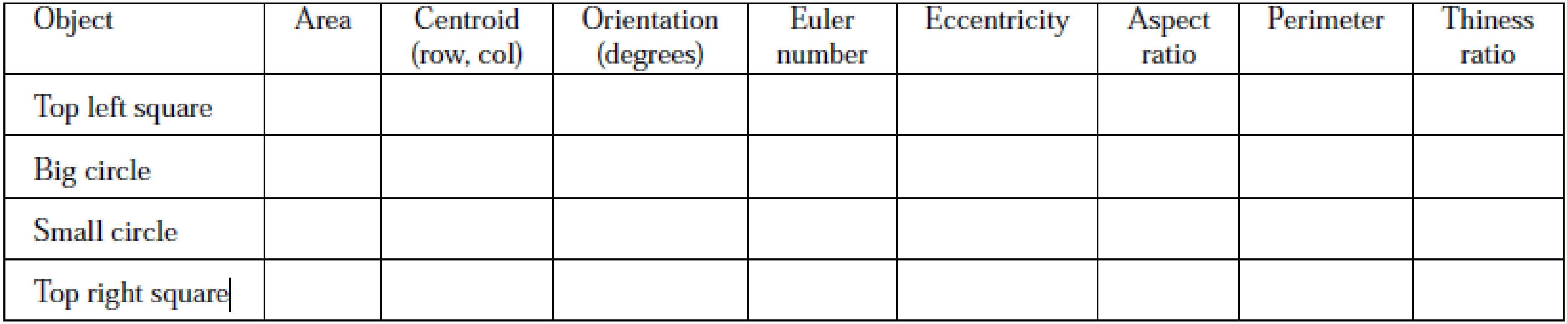
EXAMPLE

- For the given binary image compute the descriptor values and fill the it in the given table.



(a)

Table for feature extraction results



Object	Area	Centroid (row, col)	Orientation (degrees)	Euler number	Eccentricity	Aspect ratio	Perimeter	Thiness ratio
Top left square								
Big circle								
Small circle								
Top right square								

Question 2 *Do the results obtained for the extracted features correspond to your* expectations? Explain.

Question 3 *Which of the extracted features have the best discriminative power to* help tell squares from circles? Explain.

Question 4 *Which of the extracted features have the worst discriminative power* to help tell squares from circles? Explain.

Question 5 *Which of the extracted features are ST invariant, that is, robust to* changes in size and translation? Explain.

Question 6 *If you had to use only one feature to distinguish squares from circles,* in a ST-invariant way, which feature would you use? Why?