Digital Image Processing (CSE/ECE 478)

Lecture # 13: Representation

Avinash Sharma

Center for Visual Information Technology (CVIT),

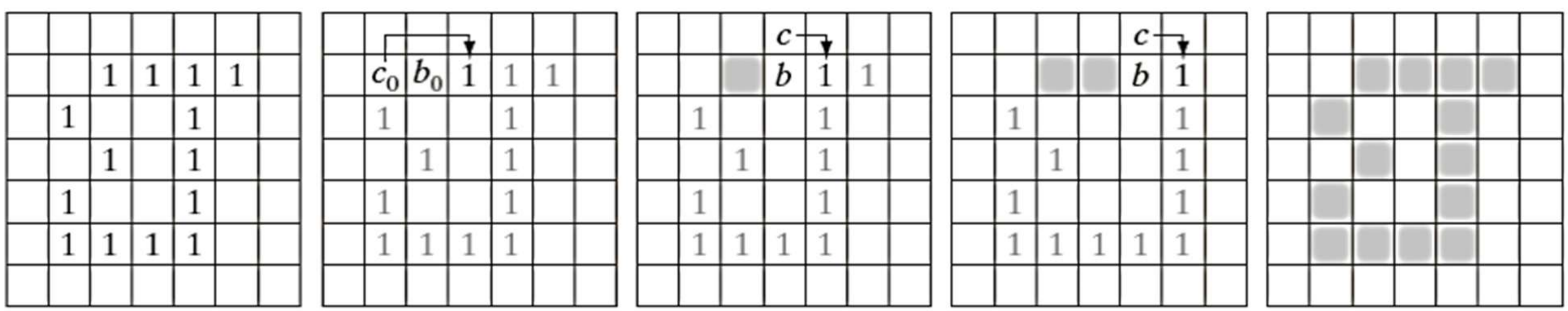
IIIT Hyderabad

Today's Lecture

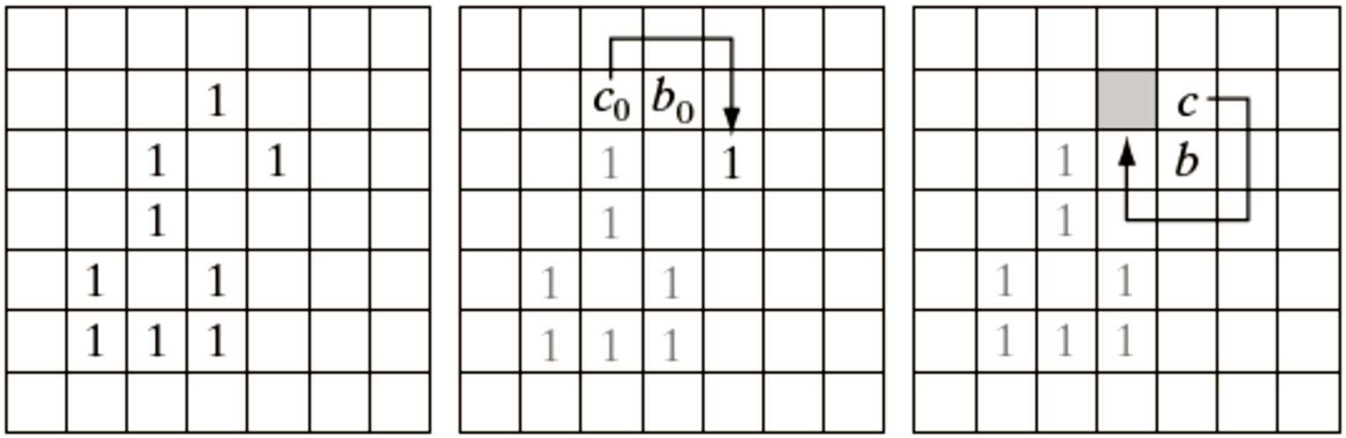
- **Shape (Boundary) descriptors**
  - Chain codes and Shape Number
  - Signature
  - Fourier descriptor
  - Moments
- Region descriptors
  - Simple descriptors
  - Statistical
  - Spectral

Boundary Following

- Start at uppermost, leftmost point
- Mark next boundary pixel and background pixel (store b_0, b_1)
- Keep marking next boundary pixel (b) and background pixel (c) iteratively
- Stop, when $b = b_0$ **and next boundary pixel is $\boldsymbol{b_1}$**



Boundary Following

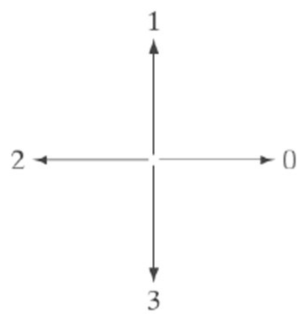


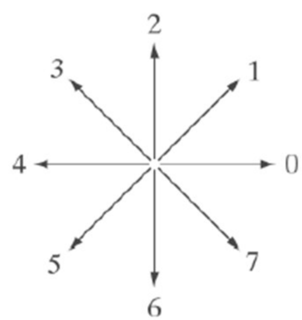
a b c

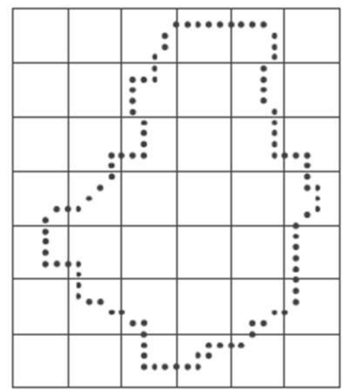
FIGURE 11.2 Illustration of an erroneous result when the stopping rule is such that boundary-following stops when the starting point, b_0, is encountered again.

Chain Codes

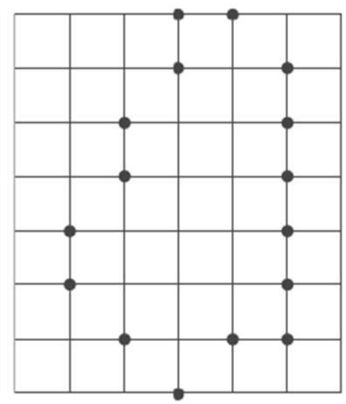
- Boundary representation as directional numbers



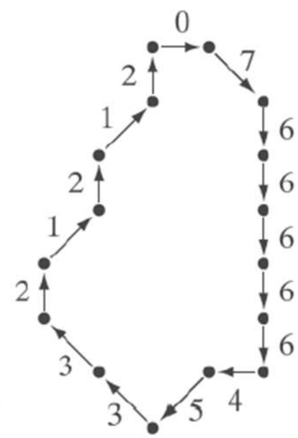




Original boundary



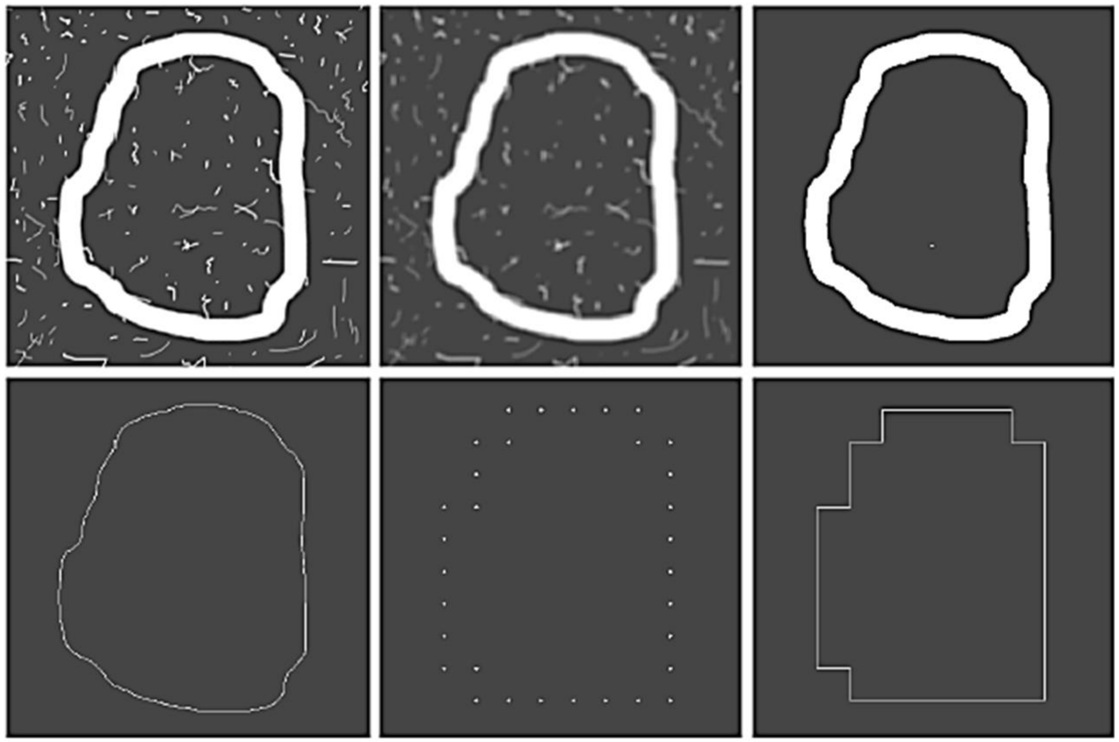
Sub-sampled boundary



Chain code of boundary

Boundary representation: 07666664533212121

Chain Codes



8 directional chain code: 00006066666666444444242222202202

Chain Codes in Practice

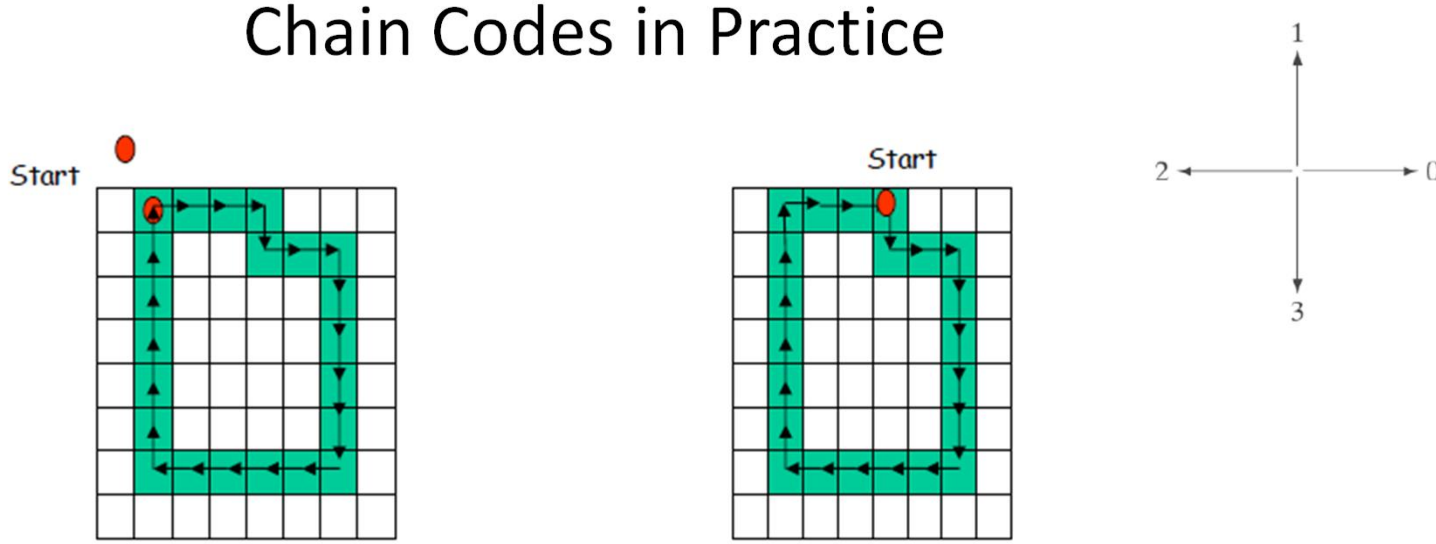
- Depends on starting point

- Normalize the chain code to address this problem
  - assume the chain is a circular sequence
  - Redefine the starting point such that we generate an integer of smallest magnitude

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

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

Chain Codes in Practice



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

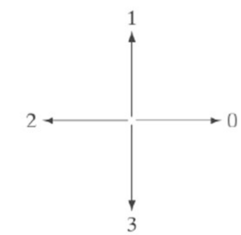
Chain Code 1

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

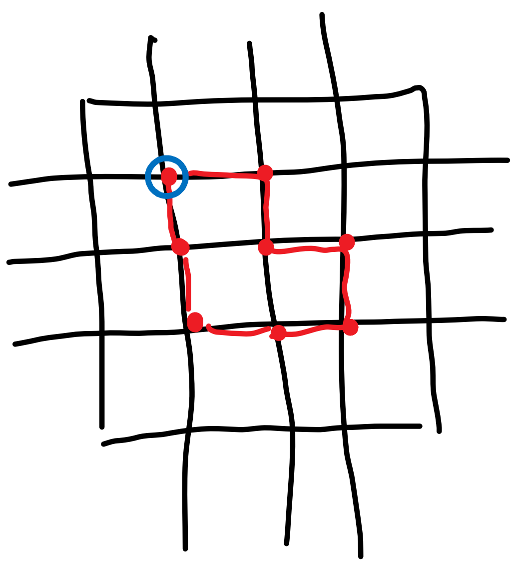
Chain Code 2

Normalized Code 0, 0, 0, 3, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1

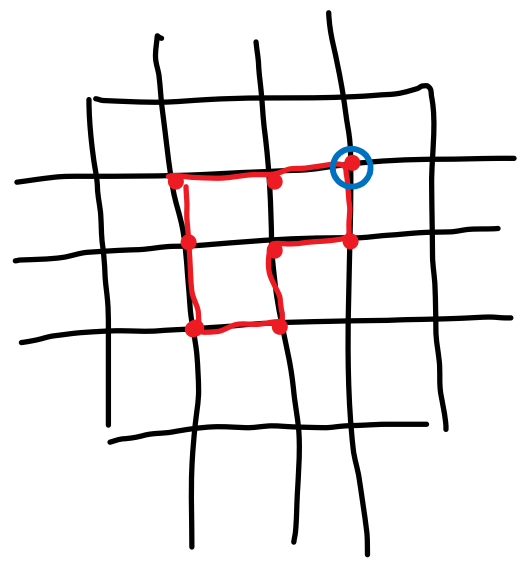
Chain Codes in Practice



- Changes code depends on orientation



chain code: 0,3,0,3,2,2,1,1



chain code: 3,2,3,2,1,1,0,0

Chain Codes in Practice

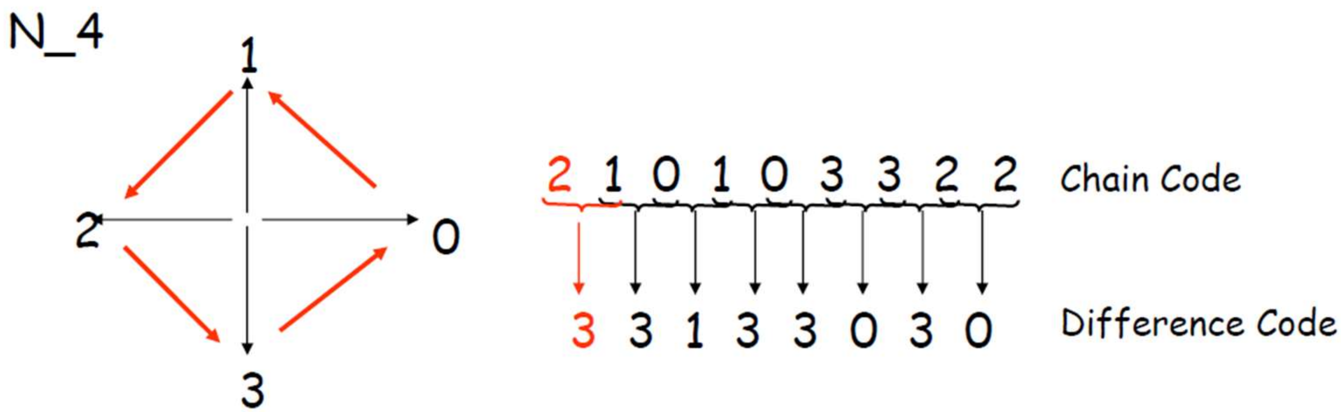
- How can we normalize for rotation?

- One solution
  - Use the “first difference” of the chain code, instead of the code itself

- The difference is obtained by simply counting (counter-clockwise) the number of directions that separate two adjacent elements

Chain Codes in Practice

- How can we normalize for rotation?



Difference: Count the number of separating directions in an anti-clockwise fashion

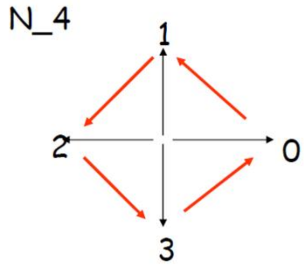
Chain Codes in Practice

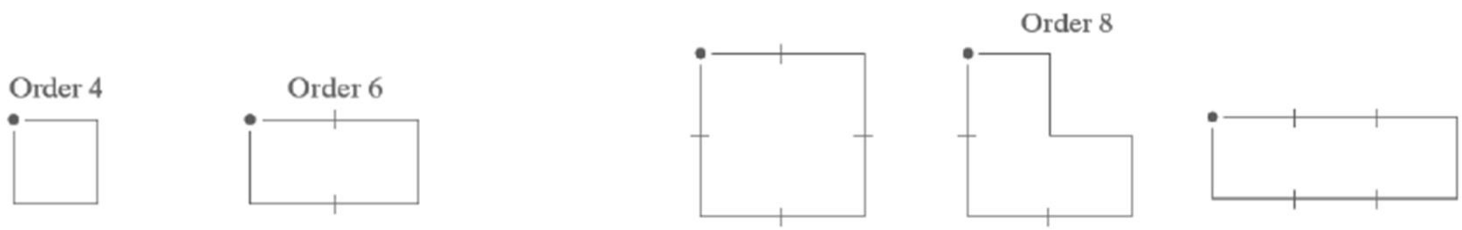
- Not Scale invariant
  - Several chain codes of the same object at different resolution

- While difference coding helps, it does not make a chain code completely invariant to rotation
  - Image digitization and noise can cause problems
  - One solution is to orient the resampling grid along the object principal axes (eigen axes).

- Nonetheless, a commonly used encoding scheme

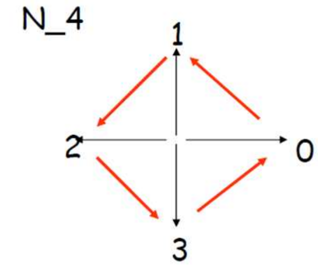
Boundary descriptor: Shape number

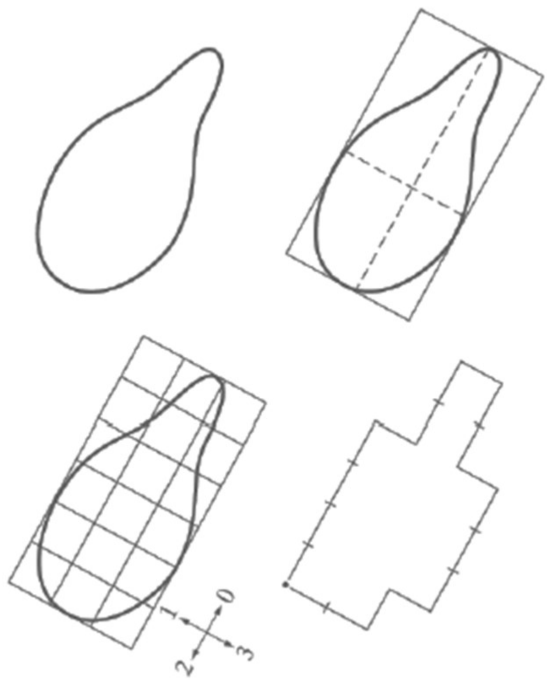




	Order 4	Order 6	Order 8		
Chain code:	0 3 2 1	0 0 3 2 2 1	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

Boundary descriptor: Shape number





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

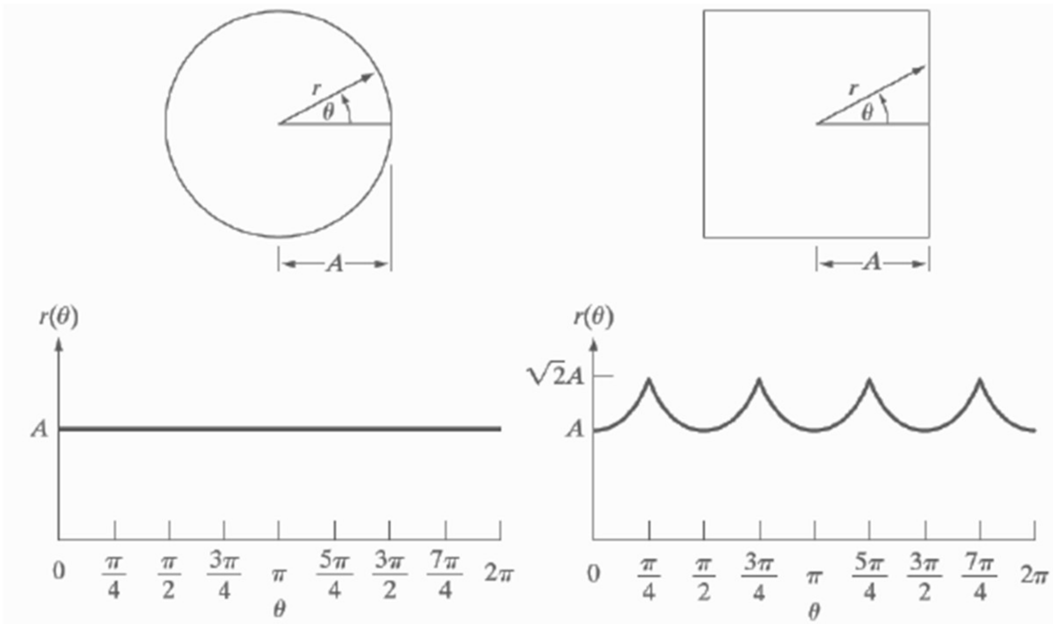
Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

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

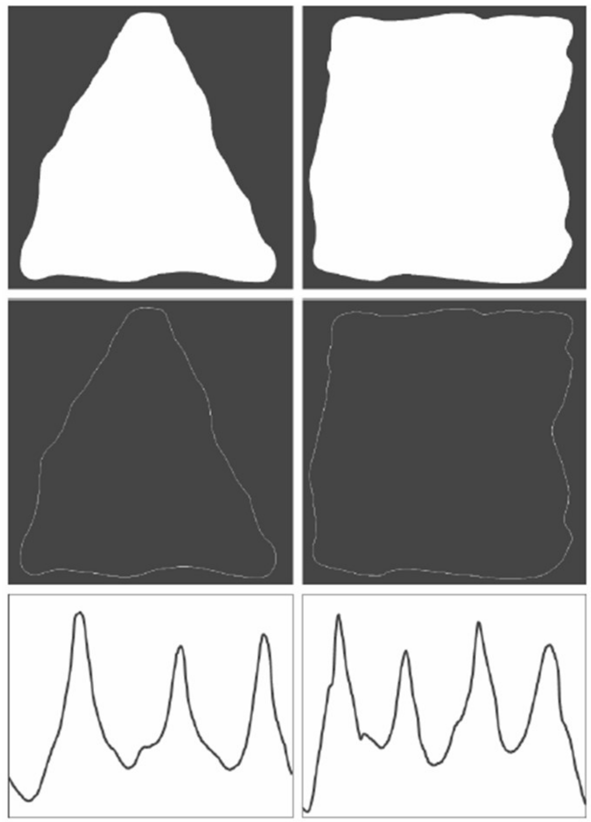
Boundary descriptor: Signature

- 2D shape as 1D signature

Distance vs Angle Curves



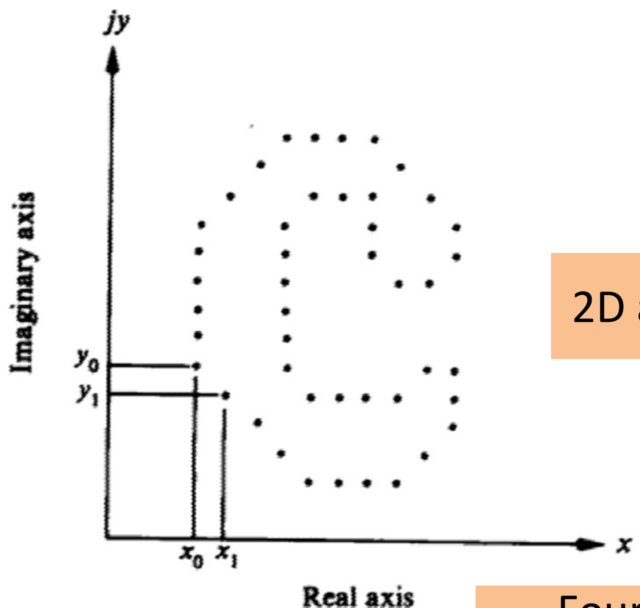
Boundary descriptor: Signature



Slope Density Function

Boundary description: Fourier Descriptors

- Boundary as a set of points



K point boundary (starting at x_0, y_0):
$(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_{K-1}, y_{K-1})$

Can be expressed as $\mathrm{x(k)} = x_k$ and $\mathrm{y(k)} = y_k$

or $\mathrm{s(k)} = [\mathrm{x(k)}, \mathrm{y(k)}],\ k = 0,1,2, \ldots, K-1$

2D as 1D

Treat as a complex number:

$$\mathrm{s(k)} = \mathrm{x(k)} + j\ \mathrm{y(k)}$$

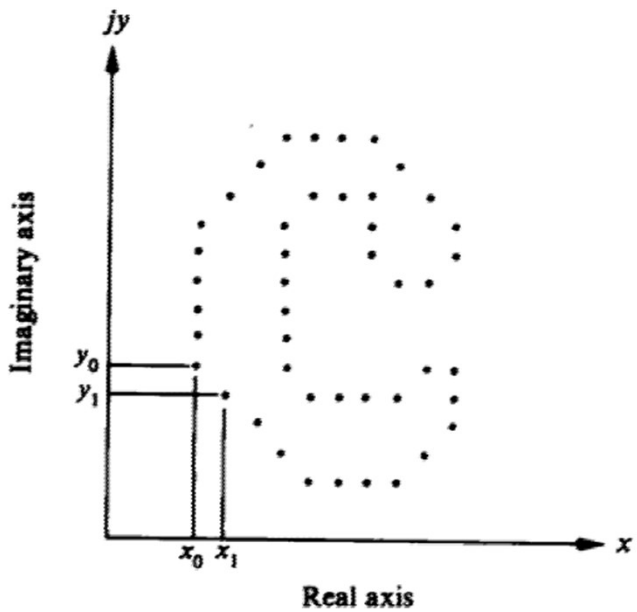
DFT of $\mathrm{s(k)}$:

$$\mathrm{a(u)} = \sum_{k=0}^{K-1} s(k) e^{-j2 \qquad /K}$$

Fourier Descriptor

Fourier Descriptors

- Boundary as a set of points



DFT of $\mathrm{s(k)}$:

$$\mathrm{a(u)} = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K}$$

Inverse DFT to restore $\mathrm{s(k)}$:

$$\mathrm{s(k)} = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K}$$

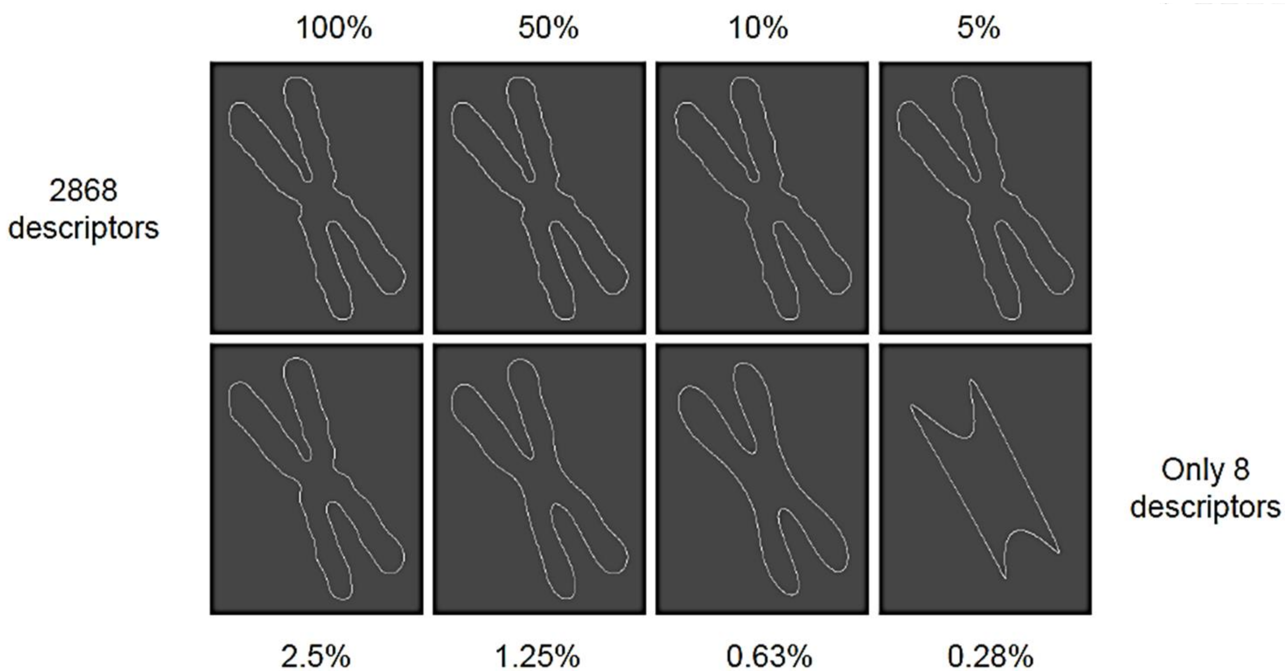
Use only first P coefficients in inverse DFT

$$\hat{s}(\mathrm{k}) = \frac{1}{P} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/P}$$

Fourier Descriptors

- Use only P coefficients for inverse DFT

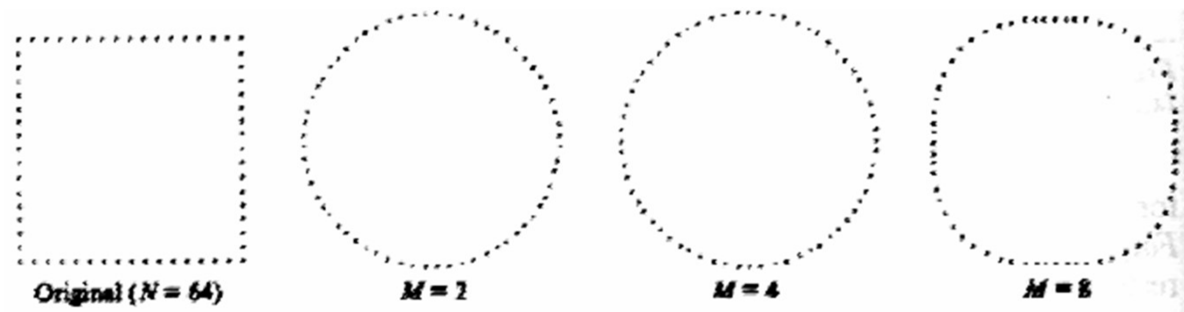
$$\hat{s}(\mathrm{k}) = \frac{1}{P}\sum_{u=0}^{P-1} a(u)e^{j2\pi uk/P}$$



Fourier Descriptors (take away)

1. We only need a few descriptors to capture the gross shape

2. Low order coefficients can be compared to measure the similarity of shapes

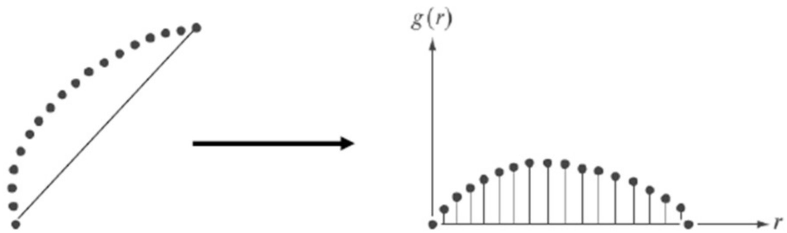


Fourier Descriptors (take away)

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

Boundary Description using Statistical Moments

- Boundary as 1D function



$$\mu_n(v) = \sum_{i=0}^{A-1} (v_i - m)^n \, p(v_i)$$

$$m = \sum_{i=1}^{A-1} v_i \, p(v_i)$$

Consider amplitude of g(*r*) as a discrete random variable v, with amplitude histograms $\mathrm{p}(v_i)$, i=0,1,2,…,A-1

Today's Lecture

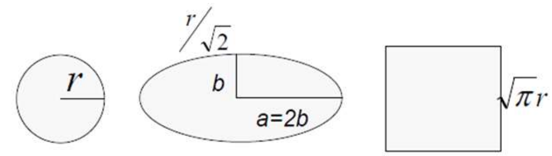
- Connected components algorithm
- Shape descriptors
  - Shape Number
  - Signature
  - Fourier descriptors
  - Moments
- **Region descriptors**
  - Simple descriptors
  - Statistical
  - Spectral

Region Descriptors- Simple

- Area (A)
- Perimeter (P)
- Compactness
- Circularity ratio
- Mean/Median intensity
- Max/Min intensity
- Normalized area

$$C = \frac{(P)^2}{area}$$

$$R_c = \frac{(4\pi A)}{P^2} = \frac{A}{P^2/4\pi}$$



C :

R_c :

Perimeter of ellipse: $p \approx 2\pi\sqrt{\frac{a^2+b^2}{2}}$

Region Descriptors- normalized area

Region no. (from top)	Ratio of lights per region to total lights
1	0.204
2	0.640
3	0.049
4	0.107

Region Descriptors: Topological

- Topology: study of properties of a figure that are unaffected by any deformations, twisting and stretching

- Who is a topologist? A: Someone who cannot distinguish between a doughnut and a coffee cup.

Topological descriptors

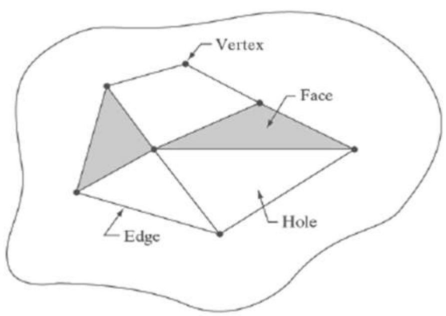
- Properties of a region that are unaffected by any deformations, twisting and stretching

H: # holes in the image

C: # connected components

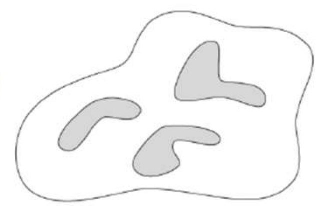
E = C-H: Euler Number

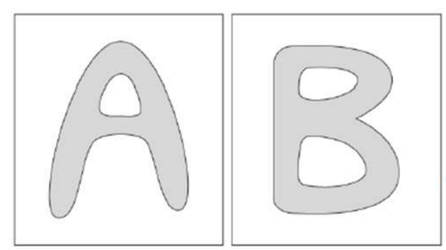
$$V - Q + F = C - H = E$$



H=2, C=1, E=-1

H=0, C=3, E=3

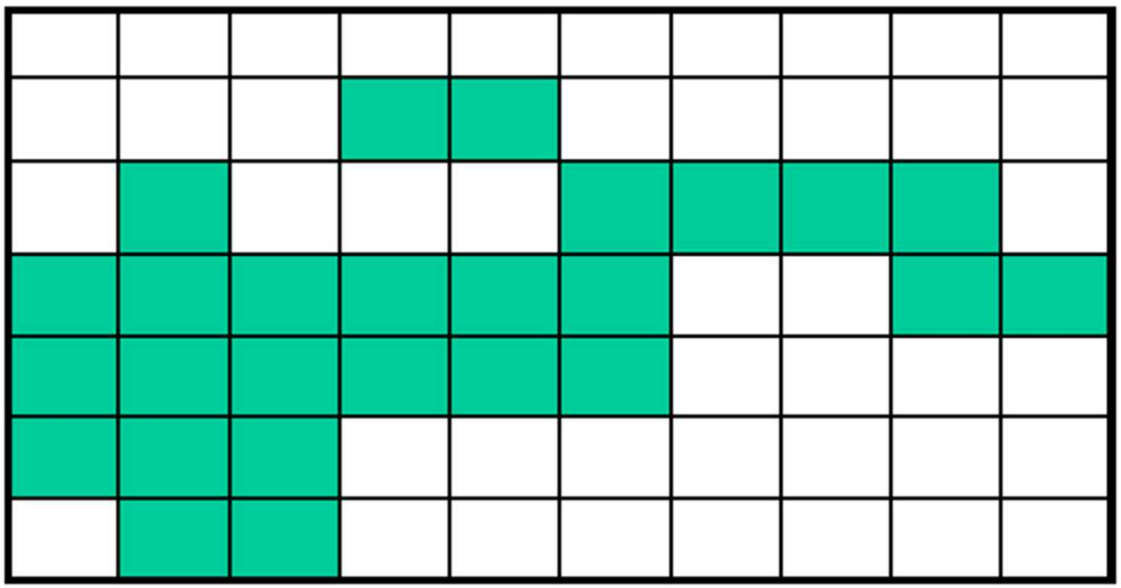




Euler number?

Topological descriptors

- Can you distinguish images of characters: 0,1,8,6,Z



Region descriptors: Texture

- Quantify texture content to describe the region in terms of smoothness, coarseness and regularity.

- We will discuss two approaches
  - Statistical → smooth, coarse, grainy
  - Spectral → properties of the fourier spectrum (periodicity, energy, peaks)

Quantifying Texture: Statistical approaches

- Statistical moments of the intensity histogram

$$\boxed{\text{image}} \quad f \rightarrow h_f \quad \boxed{\text{histogram}}$$

- Obtain statistics of the histogram:

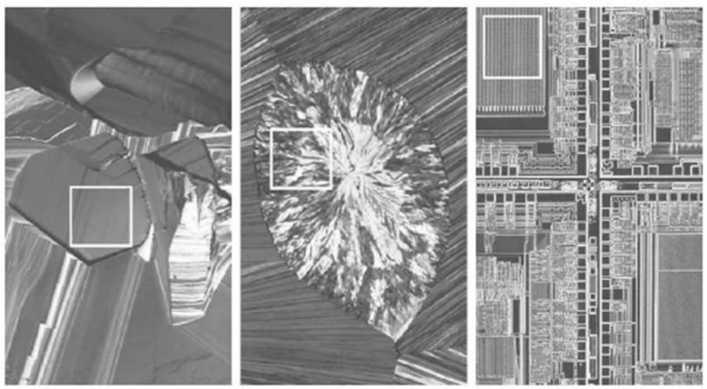
<u>Mean:</u>

σ^2 <u>Variance:</u>

<u>Skewness:</u>

<u>Entropy:</u>

Quantifying Texture: Statistical approaches



$$R_{Norm} = 1 - \frac{1}{1 + \frac{\sigma^2}{(L-1)^2}}$$

$$U = \sum_{i=0}^{L-1} h(i)^2$$

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

Quantifying Texture: Statistical approaches

- What is the main limitation of using histograms?

- Co-occurrence matrix ($\boldsymbol{G}$)
  - Position operator ($\boldsymbol{Q}$)

$\boldsymbol{Q}$ is “one pixel immediately to the right”

$$p_{ij} = \frac{q_{ij}}{n}$$

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image f

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix **G**

Quantifying Texture: Statistical approaches

$$p_{ij} = \frac{q_{ij}}{n}$$

$$P(i) = \sum_{j=1}^{K} p_{ij}$$

$$P(j) = \sum_{i=1}^{K} p_{ij}$$

$$m_r = \sum_{i=1}^{K} i\, P(i)$$

$$m_c = \sum_{j=1}^{K} j\, P(j)$$

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image *f*

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix **G**

$$\sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2\, P(i)$$

$$\sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2\, P(j)$$

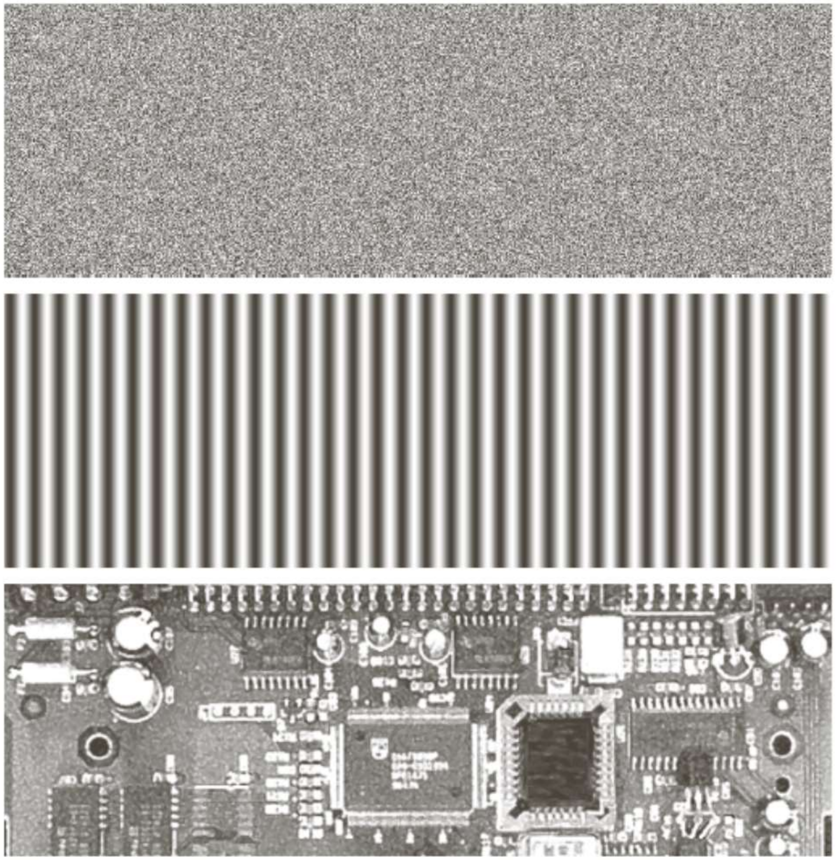
Quantifying Texture: Statistical approaches

Descriptor	Explanation	Formula
Maximum probability	Measures the strongest response of **G**. The range of values is [0, 1].	$\max_{i,j}(p_{ij})$
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image. Range of values is 1 to −1, corresponding to perfect positive and perfect negative correlations. This measure is not defined if either standard deviation is zero.	$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\sigma_c}$ $\sigma_r \neq 0;\ \sigma_c \neq 0$
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image. The range of values is 0 (when **G** is constant) to $(K-1)^2$.	$\sum_{i=1}^{K}\sum_{j=1}^{K}(i-j)^2 p_{ij}$

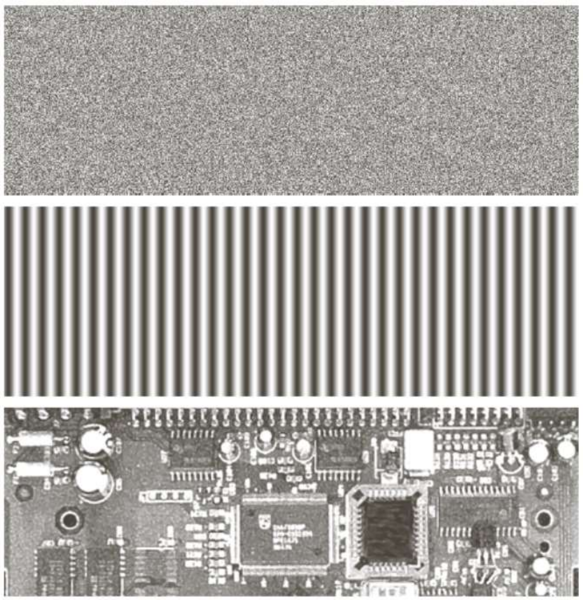
Quantifying Texture: Statistical approaches

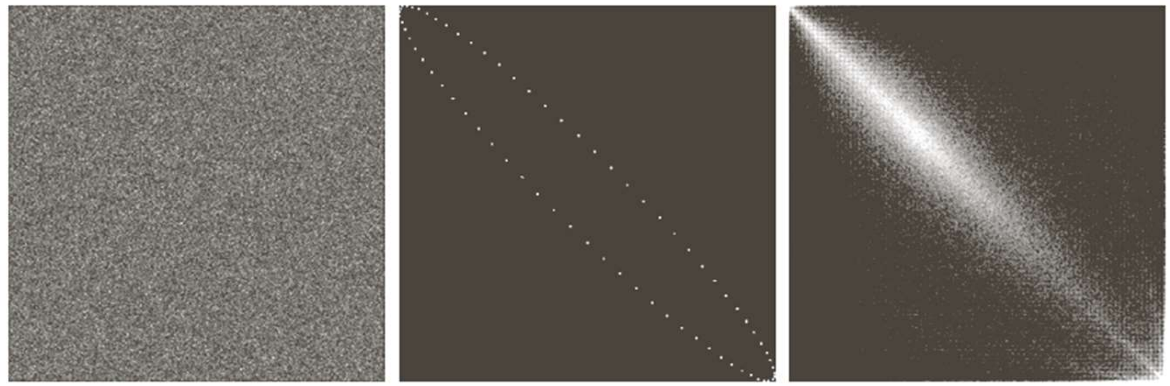
Uniformity (also called Energy)	A measure of uniformity in the range [0, 1]. Uniformity is 1 for a constant image.	$\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij}^2$
Homogeneity	Measures the spatial closeness of the distribution of elements in **G** to the diagonal. The range of values is [0, 1], with the maximum being achieved when **G** is a diagonal matrix.	$\sum_{i=1}^{K}\sum_{i=1}^{K} \frac{p_{ij}}{1 + \lvert i - j\rvert}$
Entropy	Measures the randomness of the elements of **G**. The entropy is 0 when all p_{ij}'s are 0 and is maximum when all p_{ij}'s are equal. The maximum value is $2\log_2 K$. (See Eq. (11.3-9) regarding entropy).	$-\sum_{i=1}^{K}\sum_{i=1}^{K} p_{ij}\log_2 p_{ij}$

Quantifying Texture: Statistical approaches



Quantifying Texture: Statistical approaches





Normalized Co-occurrence Matrix	Descriptor					
	Max Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	−0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	570	0.01230	0.0824	6.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	1356	0.00480	0.2048	13.58

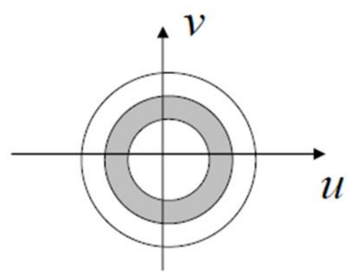
Quantifying Texture: Spectral approach

$$f(x,y) \leftrightarrow F(u,v)$$

Power Spectrum

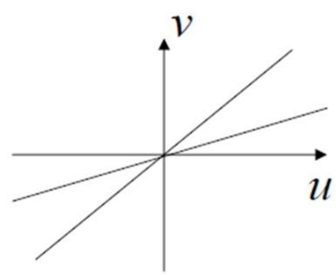
$$P(u,v) = |F(u,v)|^2$$

$$P(r) = 2\sum_{\theta=0}^{\pi} P(r,\theta)$$



Indicator for size of dominant texture element or texture coarseness

$$P(\theta) = \sum_{r=0}^{L/2} P(r,\theta)$$



Indicator for the directionality of the texture

source: Shahram Ebadollahi

Quantifying Texture: Spectral approach

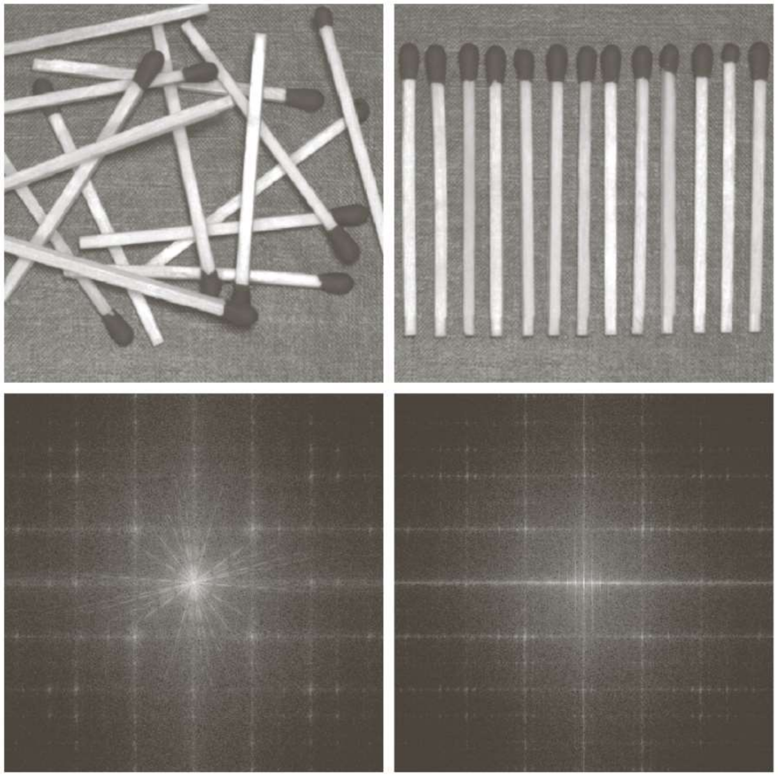
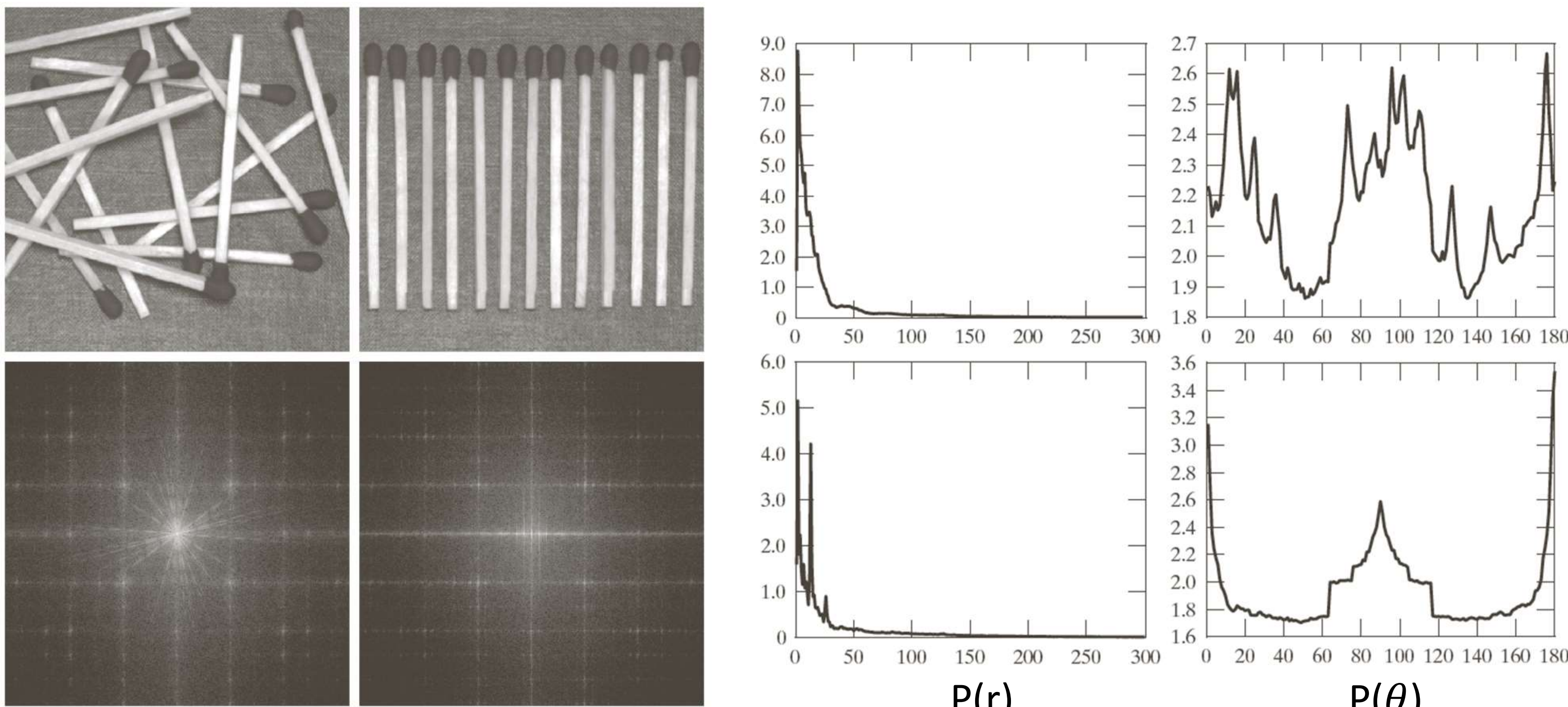


FIGURE 11.35 (a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600×600 pixels.

Quantifying Texture: Spectral approach



P(r) P(θ)