Introduction to Digital Image Processing

— Image Features

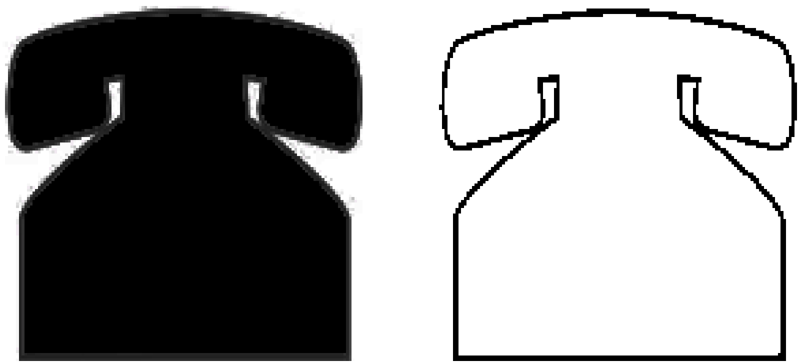
Xiaohui Yuan

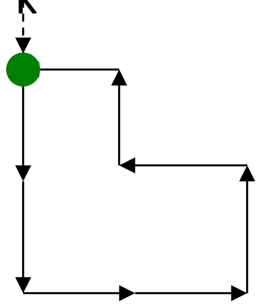
Department of Computer Science and Engineering
University of North Texas
xiaohui.yuan@unt.edu

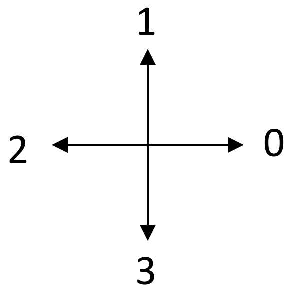
Object Boundary

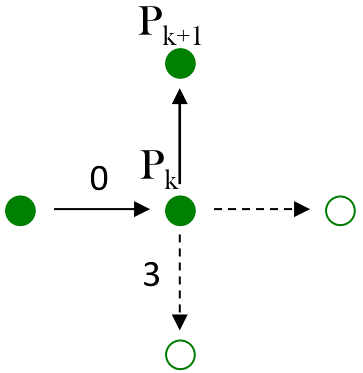
Chain Code (4-Direction)

1. Find the top-left pixel on the boundary; call this P_0. The direction property DIR is initialized as 3.
2. Traverse the four neighborhoods of the current pixel in the counter-clockwise order,
   - Begin the search in the direction calculated as (DIR+3) mod 4.
3. Stop when the current boundary pixel P_k equals to P_1 and P_{k-1} equals to P_0.



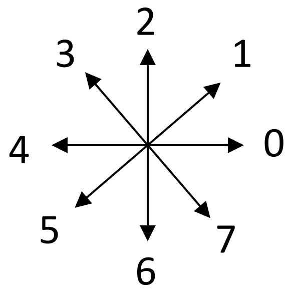


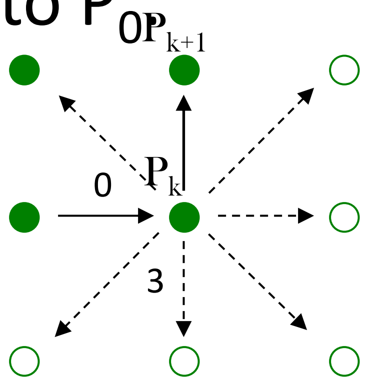


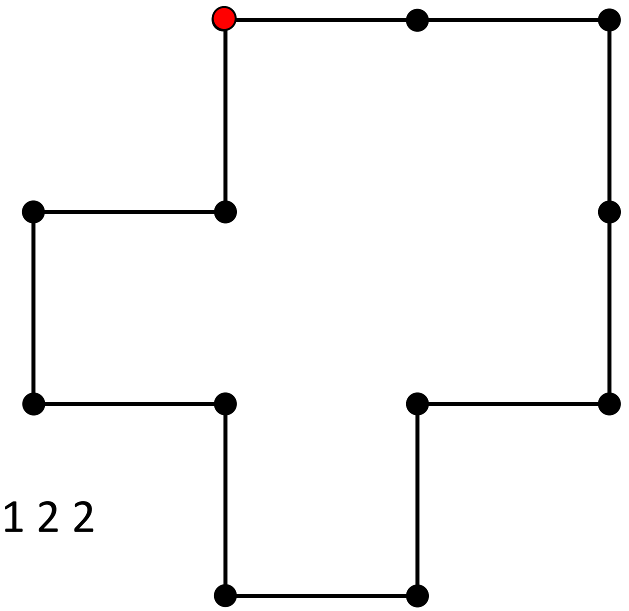


Chain Code (8-Direction)

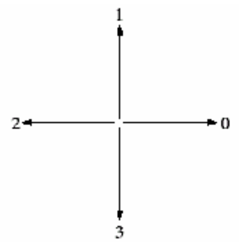
1. Find the top-left pixel on the boundary (P_0.) The direction property DIR is initialized as 7.
2. Traverse the eight neighborhood of the current pixel in a counter-clockwise order
   - beginning the search at the pixel in direction (DIR+5) mod 8 or in direction
     - (DIR+7) mod 8, if DIR is even
     - (DIR+6) mod 8, if DIR is odd
3. Stop when the current boundary pixel P_n equals to P_1 and P_{n-1} equals to P_0

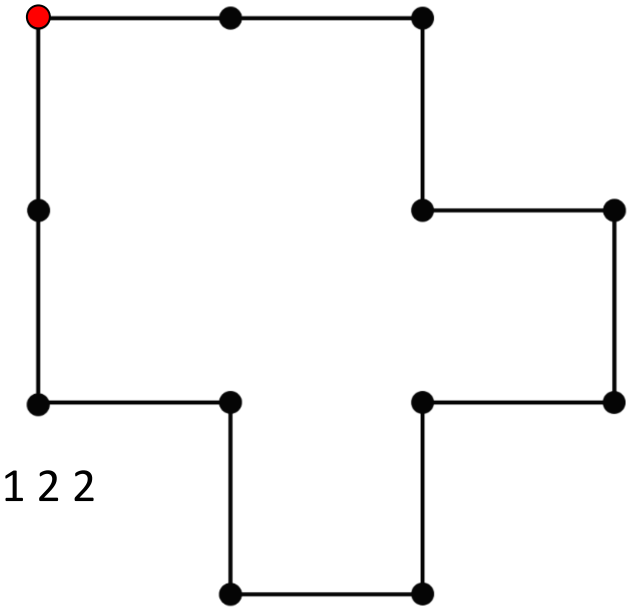






3 2 3 0 3 0 1 0 1 1 2 2





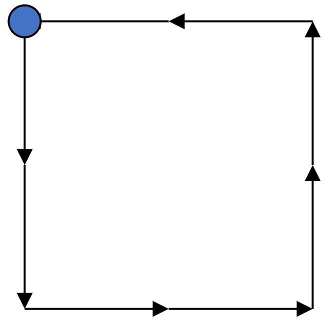
3 3 0 3 0 1 0 1 2 1 2 2

Problems with Chain Code

- Chain code representation is conceptually appealing, but has the following problems
  - Dependent on the **starting point**
  - Dependent on the **object orientation**
- To use boundary representation for recognizing objects, we need to achieve invariance to the starting point and orientation
  - Normalized codes
  - Differential codes

Normalized Chain Code

- We treat the chain code as a cyclic sequence of direction numbers and redefine the starting point so that the resulting sequence of numbers forms **an integer of minimum magnitude**.



33001122

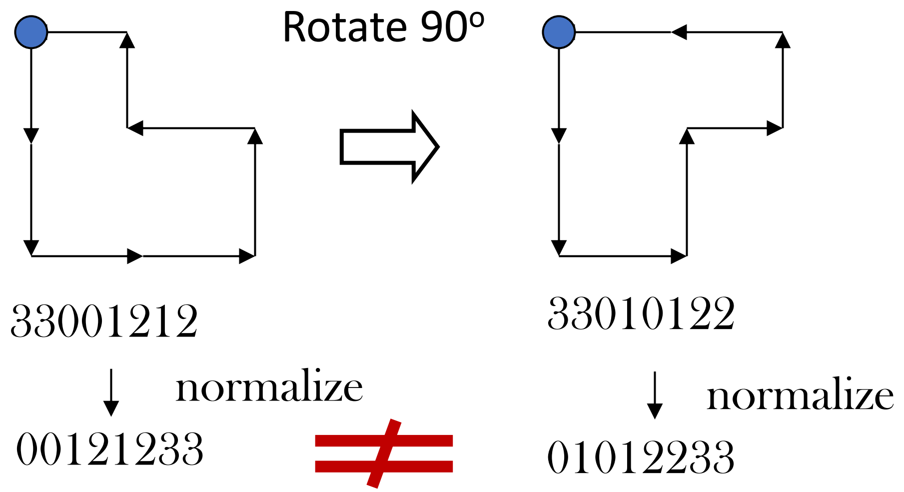
Create cyclic matrix →

33001122		00112233
30011223		01122330
00112233		11223300
01122330	Sort rows →	12233001
11223300		22330011
12233001		23300112
22330011		33001122
23300112		30011223

→ First row gives the normalized chain code

00112233

Differential Chain Code



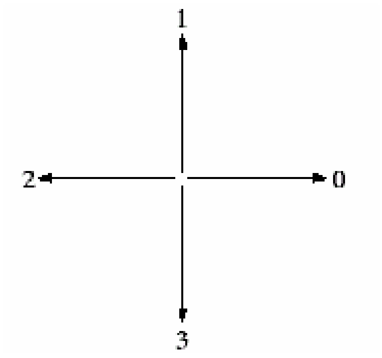
Differential coding is obtained by counting the number of direction changes of the two adjacent elements.

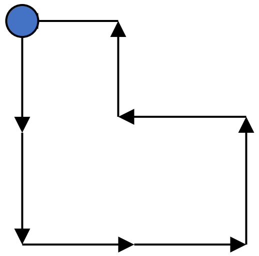
Computation:

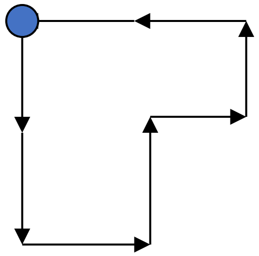
$d_k = (c_k - c_{k-1}) \bmod 4$ for 4-directional chain codes

$d_k = (c_k - c_{k-1}) \bmod 8$ for 8-directional chain codes

Normalized Differential Chain Codes







Differential code:
$d_k = (c_k - c_{k-1}) \bmod 4$

33001212
↓ differentiate
10101131
↓ normalize
01011311

33010122
↓ differentiate
10113110
↓ normalize
01011311

Fourier Descriptor

- The contour of an object is a closed curve described by pixel coordinates (x_0, y_0).
  - The sequence of pixels are encountered in traversing the object contour in the counterclockwise direction: $(x_0, y_0), \ldots, (x_{k-1}, y_{k-1})$.
  - Each coordinate pair is treated as a complex number $s = x + iy$.
- The Fourier transform of $s(k)$ results in a set of Fourier coefficients $a(u)$.

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{-j\frac{2\pi uk}{N}}$$

$a(u)$ are called **Fourier Descriptor**.

- With inverse Fourier transform, $s(k)$ can be reconstructed.
  - If the first p terms are used, the reconstruction is an approximation.
  - Two shapes are compared against their Fourier descriptors.

Contour Reconstruction using Fourier Descriptor

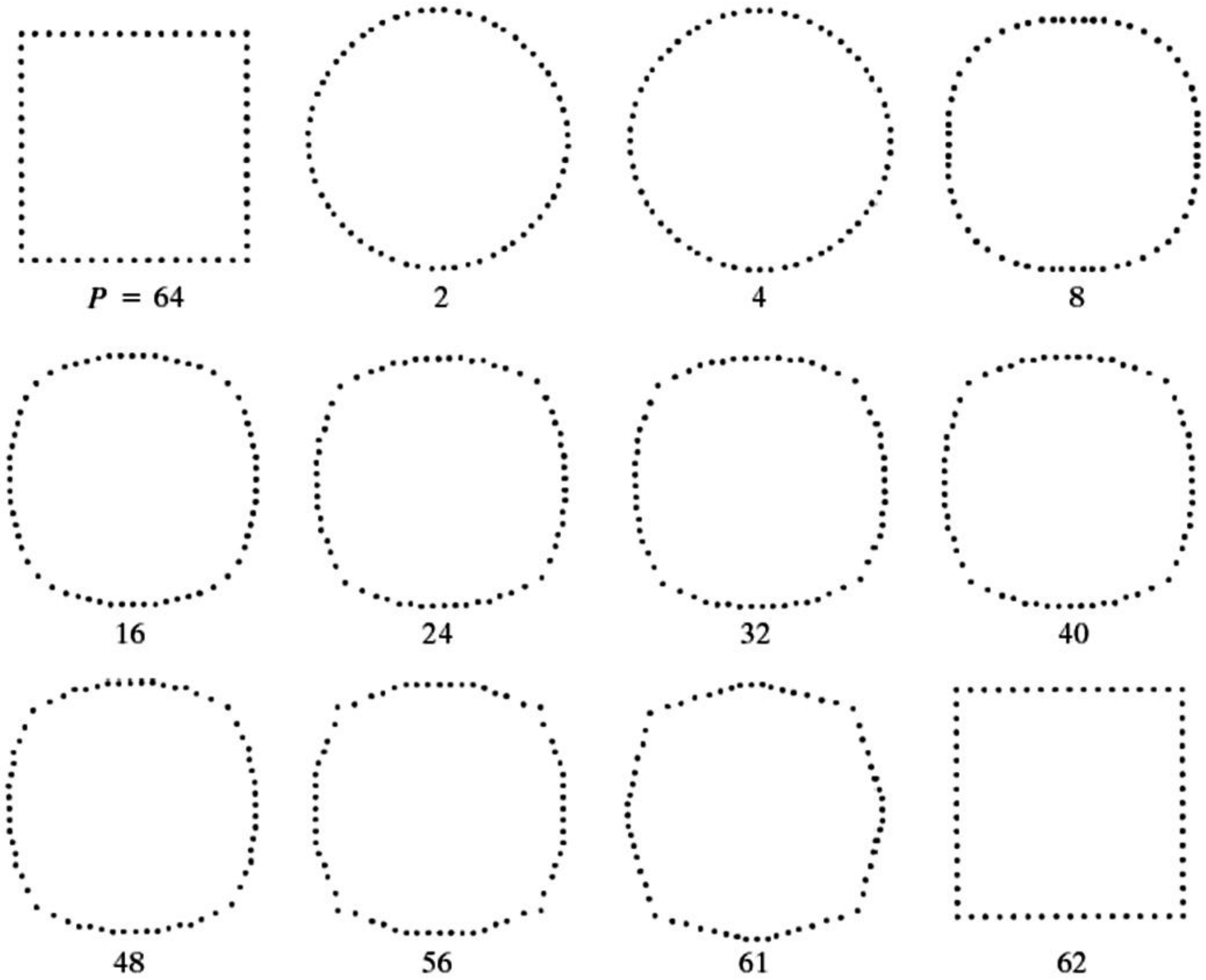
- Varying the number of coefficients used in reconstruction results in different contours.

Normalized Fourier Descriptors

- The first component of the Fourier Descriptor implies the size of the object.
- We use the first component to normalize the rest coefficients:

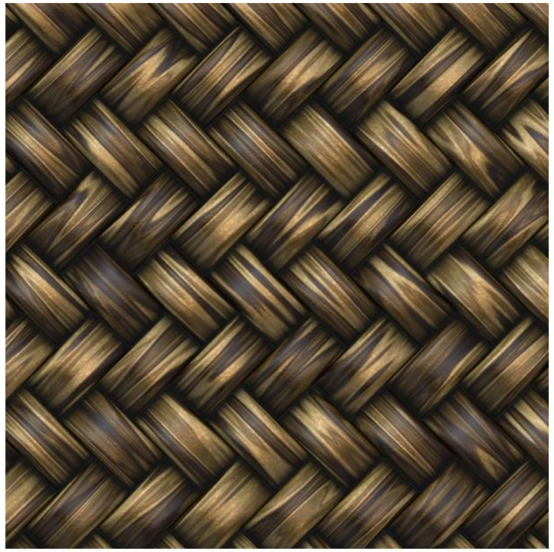
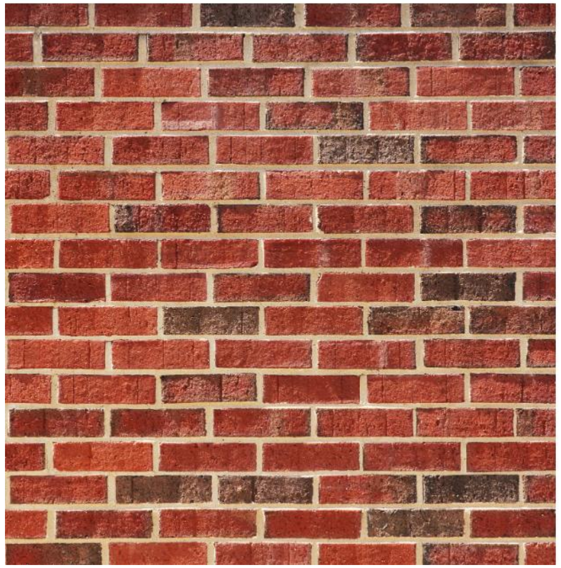
$$FD = \{\frac{FD_1}{FD_0}, \frac{FD_2}{FD_0}, \dots, \frac{FD_{N-1}}{FD_0}\}$$

- Similarity of the two objects is determined by the distance of FD.



What is Image Texture?

- What is image texture?
- How to measure texture?


Frequency Analysis …

- One possible approach is to perform local Fourier transforms of the image.
- Then we can derive information on
  - the contribution of different spatial frequencies, and
  - the dominant orientation(s) in the local texture.
- For both kinds of information, only the power (magnitude) spectrum needs to be analyzed.

Co-occurrence Matrix

- A simple and popular method for texture analysis is the computation of gray-level co-occurrence matrices.
- (optional) To compute such a matrix, we usually reduce the quantization of the image color depth (i.e., the number of color or gray-levels).
  - For example, by dividing the brightness values ranging from 0 to 255 by 64 and rounding it to the floor, we create the levels 0, 1, 2, and 3.
- By specifying a spatial relationship r (orientation and distance), the co-occurrence matrix C is obtained by counting all pairs of pixels separated by r having gray levels i and j.
  - $C_r(i, j)$ indicates how many times value i co-occurs with value j in a particular spatial relationship r.

Example

$$\begin{array}{|cccc|}\hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ \hline\end{array}$$

Image

$$r = (0, 1),\quad C_{(0,1)} = \begin{array}{c|cccc|} & 0 & 1 & 2 & 3 \\ \cline{2-5} 0 & 4 & 2 & 1 & 0 \\ 1 & 2 & 4 & 0 & 0 \\ 2 & 1 & 0 & 6 & 1 \\ 3 & 0 & 0 & 1 & 2 \\ \cline{2-5}\end{array}$$

$$r = (135, 1),\quad C_{(135,1)} = \begin{array}{c|cccc|} & 0 & 1 & 2 & 3 \\ \cline{2-5} 0 & 2 & 1 & 3 & 0 \\ 1 & 1 & 2 & 1 & 0 \\ 2 & 3 & 1 & 0 & 2 \\ 3 & 0 & 0 & 2 & 0 \\ \cline{2-5}\end{array}$$

Algorithm for Co-occurrence Matrix

Image

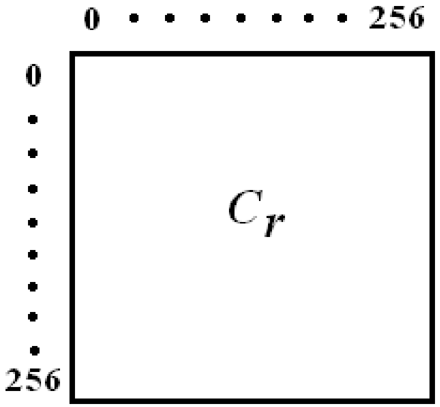
0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

r = (0, 1)

$$C_{(0,\,1)} = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 4 & 2 & 1 & 0 \\ 1 & 2 & 4 & 0 & 0 \\ 2 & 1 & 0 & 6 & 1 \\ 3 & 0 & 0 & 1 & 2 \end{array}$$

r = (135, 1)

$$C_{(135,\,1)} = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 2 & 1 & 3 & 0 \\ 1 & 1 & 2 & 1 & 0 \\ 2 & 3 & 1 & 0 & 2 \\ 3 & 0 & 0 & 2 & 0 \end{array}$$



r = (orientation, distance)

1. Assign $C_r(i, j) = 0$ for all $i, j \in [0, L]$, where L is the maximum brightness.
2. For all pixels (x_1, y_1) in the image, determine (x_2, y_2) which has the relation r with the pixel (x_1, y_1), and perform $C_r[f(x_1, y_1), f(x_2, y_2)] = C_r[f(x_1, y_1), f(x_2, y_2)] + 1$.

Using Co-occurrence Matrix

- It is often a good idea to construct a co-occurrence matrix with more than one spatial relation.
- Similar matrices of two textures indicate similar textures.
  - The difference between corresponding elements of these matrices can be taken as a similarity metric.
  - We use texture to enhance the detection of regions and contours in images.
- Additional metrics
  - Energy $\sum_i \sum_j C_r^2(i,j)$
  - Entropy $-\sum_i \sum_j C_r\ (i,j) \log_2 C_r(i,j)$
  - Homogeneity $\sum_i \sum_j \frac{C_r\ (i,j)}{1+|i-j|}$
  - Correlation $\frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) C_r(i,j)}{\sigma_i \sigma_j}$

μ_i and μ_j are the means and σ_i and σ_j are the STDs of the row and column

Local Binary Pattern

- For each pixel p, create an n-bit number (n=4 or 8) $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8$, where $b_i = 0$ if neighbor i has a value less than or equal to p's value and 1 otherwise.

	1	2	3	
	100	101	103	
8	40	50	80	4
	50	60	90	5
	7	6		

$\longrightarrow$ $\boxed{1\ 1\ 1\ 1\ 1\ 1\ 0\ 0}$

- For 4-neighbor, the range of LBP is 0 - 16
- For 8-neighbor, the range of LBP is 0 - 255
- A histogram or the decimal number is used to represent the LBP results.