# Outline

- Image Representation and Description
  - Boundary representations: Chain codes, polygonal approximations, signatures, and skeletons
  - Boundary descriptors: Basic descriptors, Fourier descriptors, statistical moments

## Boundary Representations: Chain Codes

- We can describe a boundary by starting from a point and forming a vector with directions for each pixel using 4 or 8 connectivity
- The disadvantages of this direct implementation is that, the chain can be very long and not robust to noise
- Therefore, we usually use larger grids then pixels
- In this way, both robustness is increased amd the chain is shorter

# Boundary Representations: Chain Codes (Cont.)

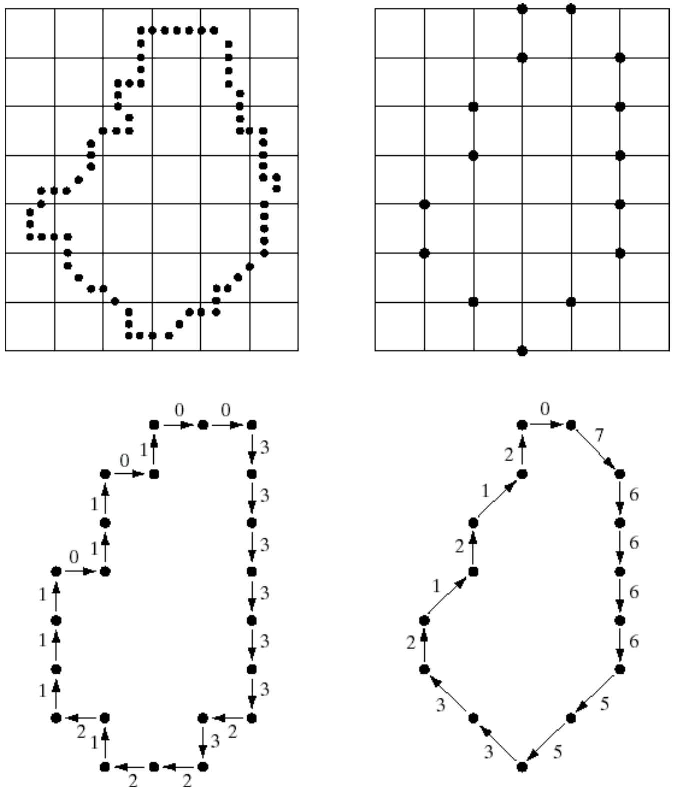



Illustration of a chain code

## Boundary Representations: Polygonal Approximations

- The goal is to find polygons that approximates the shape of the boundary
- Large number of polygons have a more accurate representation but it is time consuming to find these
- Small number of polygons catch only the basic information of the boundary (less accurate), but it is easier to find these
- Now let us have a look at a few methods to find these polygons

# Boundary Representations: Polygonal Approximations - Minimum perimeter polygons

- In this case we force the boundary to have the minimum length by keeping it in the same pixel

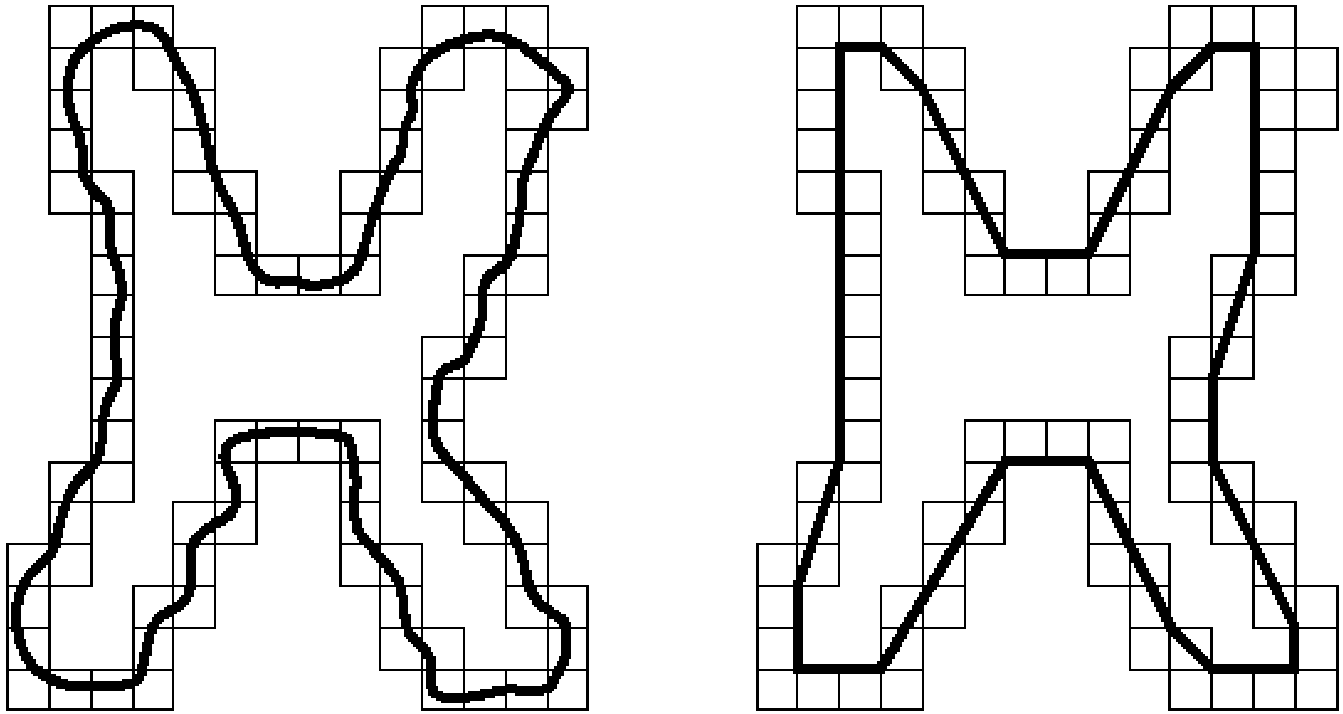


Illustration of a minimum parameter polygon

# Boundary Representations: Polygonal Approximations - Merging and Splitting

- Merging: We go through each of the boundary pixels, and merge them as long as the error between the representation and the real boundary is less than a threshold
- Splitting: We split the original boundary into two parts until a certain criterion is satisfied, such as the distance between the representation segment and the original boundary being less than a certain value

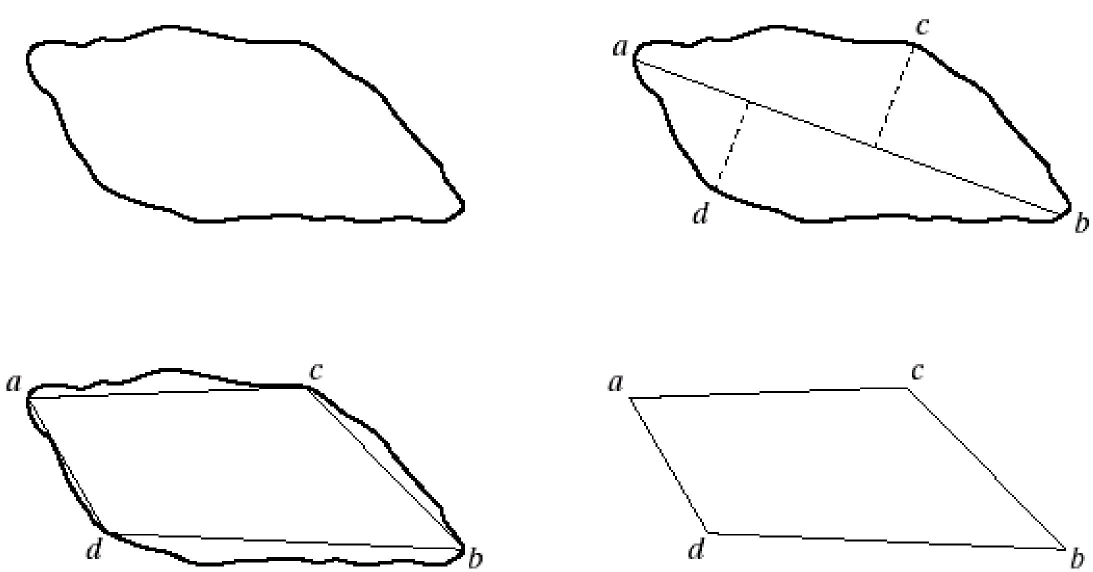



Illustration of splitting

# Boundary Representations: Signatures

- Signatures are 1-D representations of 2-D boundaries
- One straightforward way to obtain is to form a 1-D function which corresponds to the distance of the boundary pixel to its centroid

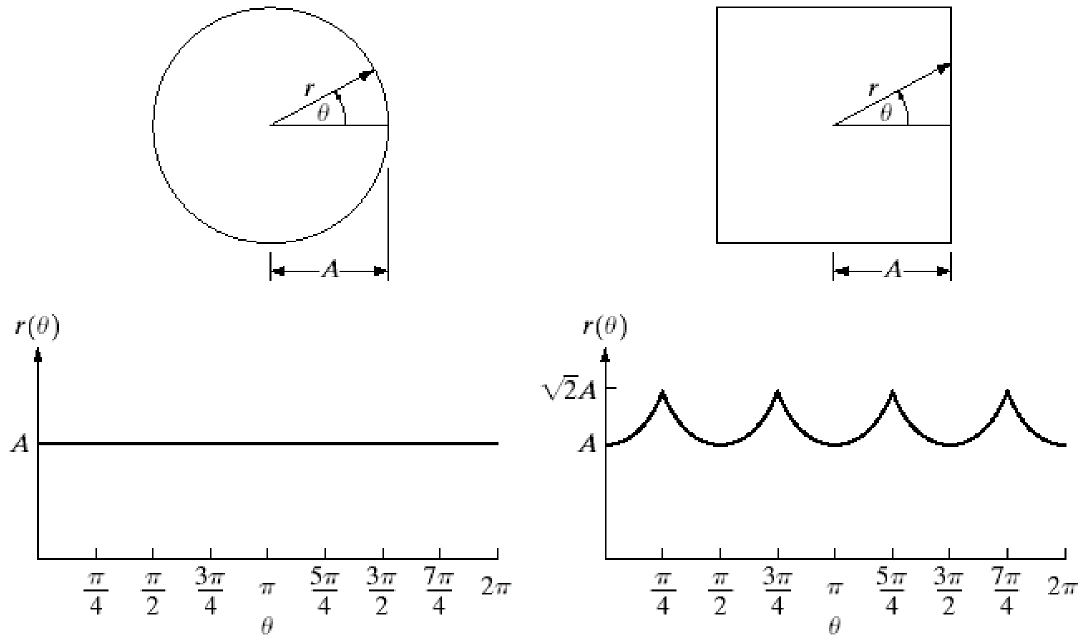



Illustration of signature

## Boundary Representations: Signatures (Cont.)

- Signatures obtained in this way are variant to rotation and scaling
- We can however obtain the same signature regardless of the orientation of the shape if we start from a point which is unique in the boundary (such as its distance from the center)
- Scaling basically just changes the amplitude of the shape, so we can obtain same signature by mapping the amplitude range of the signature to a fixed interval

# Boundary Representations: Skeletons

- We can form a skeleton as described earlier, however skeletons are not guaranteed to be connected
- We define another form of skeleton called the medial axis transformation (MAT) of a region
- MAT of a region $R$ with border $B$ is formed by all points in $R$ that has more than one closes point on $B$

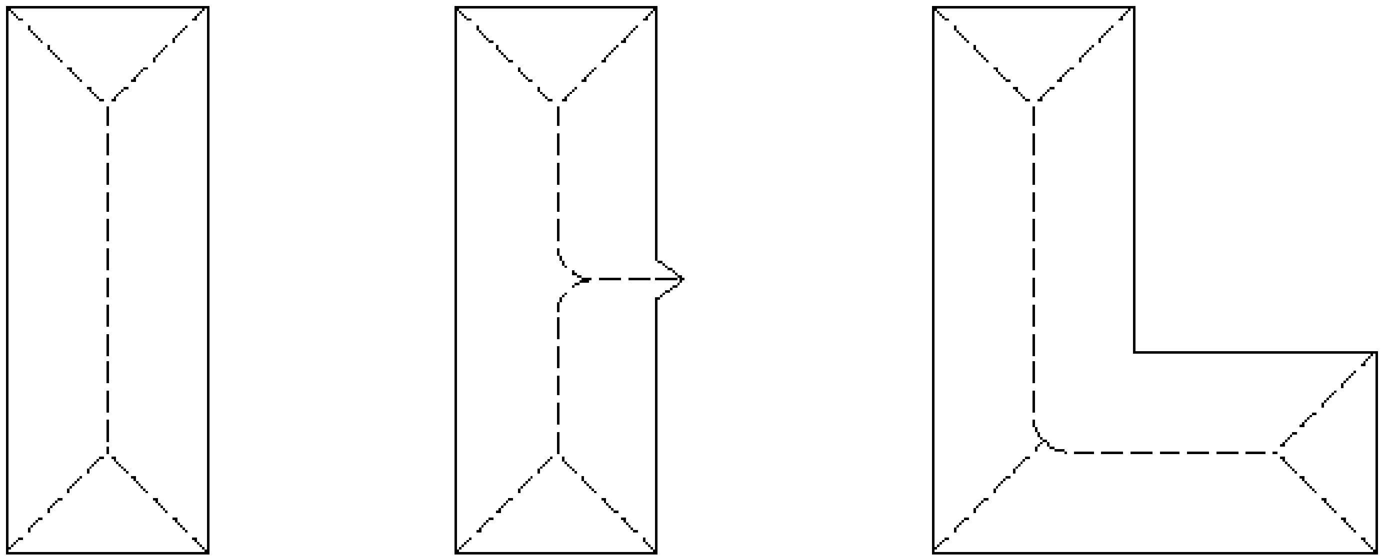


Illustration of MAT

# Boundary Descriptors: Basic Descriptors

- Length: we can calculate the length of a boundary by adding number of pixels with the horizontal and vertical directions and $sqrt 2$ times the diagonal directions
- Diameter: the diameter of a boundar is defined as the maximum of the distance between all points on boundary

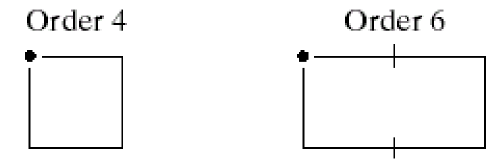
$$\text{Diam}[B] = \max_{i,j}[D(p_i, p_j)]$$

- Major and minor axis: Major axis is the line that connects the diameter points, and minor axis is the axis perpendicular to the major axis
- Eccentricity: ratio of the length of major and minor axes
- Curvature: rate of change of slope

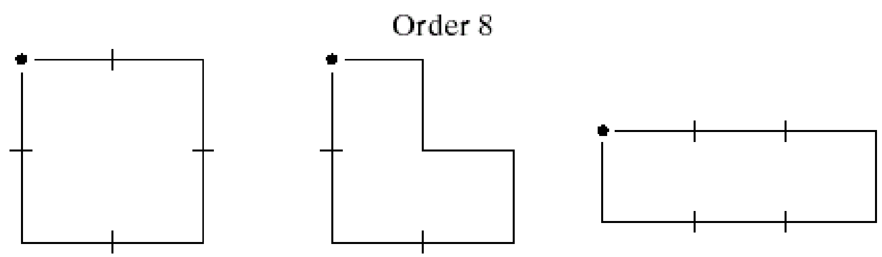
## Boundary Descriptors: Shape Numbers

- Chain code representations depend on the starting pixel
- We can remove this problem if we choose the pixel that results in a chain code that is minimum when viewed as an integer number
- Difference of a chain code is defined as simply the difference between the successive chain code elements in a circular fashion and in mod 4 or mod 8 depending on the connectivity used
- Shape number is the shifted version of the difference code so that it has the smallest value possible when viewed as an integer

# Boundary Descriptors: Shape Numbers (Cont.)




| | 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 |

Illustration of shape numbers

# Boundary Descriptors: Fourier Descriptors

- Now consider the pairs ($x$ and $y$ coordinates) of points that form a boundary
- Let us define a complex number $s(k)$

$$s(k) = x(k) + jy(k)$$

- The Fourier transform of this series is called the Fourier descriptor of the boundary

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

- The boundary can be obtained from its Fourier descriptor as follows

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

# Boundary Descriptors: Fourier Descriptors (Cont.)

- If we truncate the summation, we just keep the low pass components of the boundary resulting in a more smooth boundary

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

where $P < K$

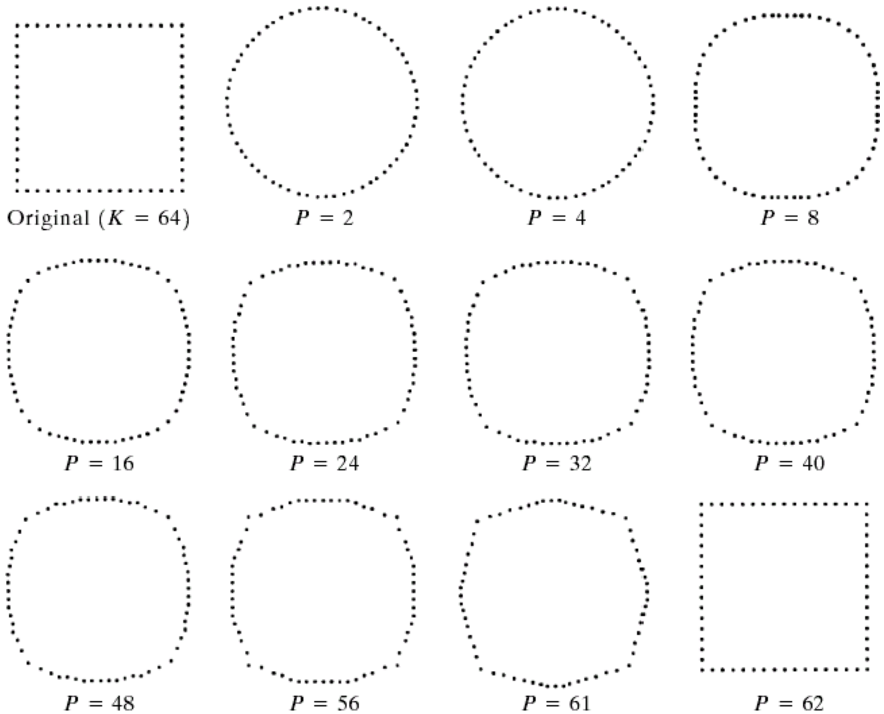



Illustration of truncation

## Boundary Descriptors: Fourier Descriptors (Cont.)

- Although Fourier descriptors are not invariant to certain changes in the boundary such as scaling rotation etc, these operations have simple corresponding changes in its Fourier descriptor

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

Illustration of shape numbers

# Boundary Descriptors: Statistical Moments

- After obtaining a 1-D representation of a boundary, we can treat this 1-D function as a PDF, and calculate the moments
- These moments can be used as the descriptors
- Let $g(r)$ represent the 1-D function, and $p(v_i)$ the PDF of its amplitude
- The mean is then

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

- The more general $n$th order moment about its mean is defined as

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