# Binary shape analysis

Our current focus: geometrical properties of objects in the image

Last week:

- Mathematical morphology
  - Operators
  - Applications
    - image cleaning (dilation, erosion, opening, closing)
    - Binary ‘region growing’ (conditional dilation)
    - Binary template matching (hit-and-miss operator)
    - Shape representations: convex hull, skeleton

This week:

  - Shape description and representation
    - Contour-based (Sonka 8.2)
    - Region-based (Sonka 8.3)

Note: the following slides are from the lecture notes of Dr. Morse (Computer Vision 1) http://morse.cs.byu.edu/650/home/index.php

# Shape Description

- Once you have segmented an image into objects, how do you describe them?
- Applications:
  - Object identification
  - Content-based image retrieval
  - Object searching/finding/tracking
  - Image registration (especially multimodality)
- Often used as inputs to
  - matching algorithms
  - machine learning algorithms

# Shape Description

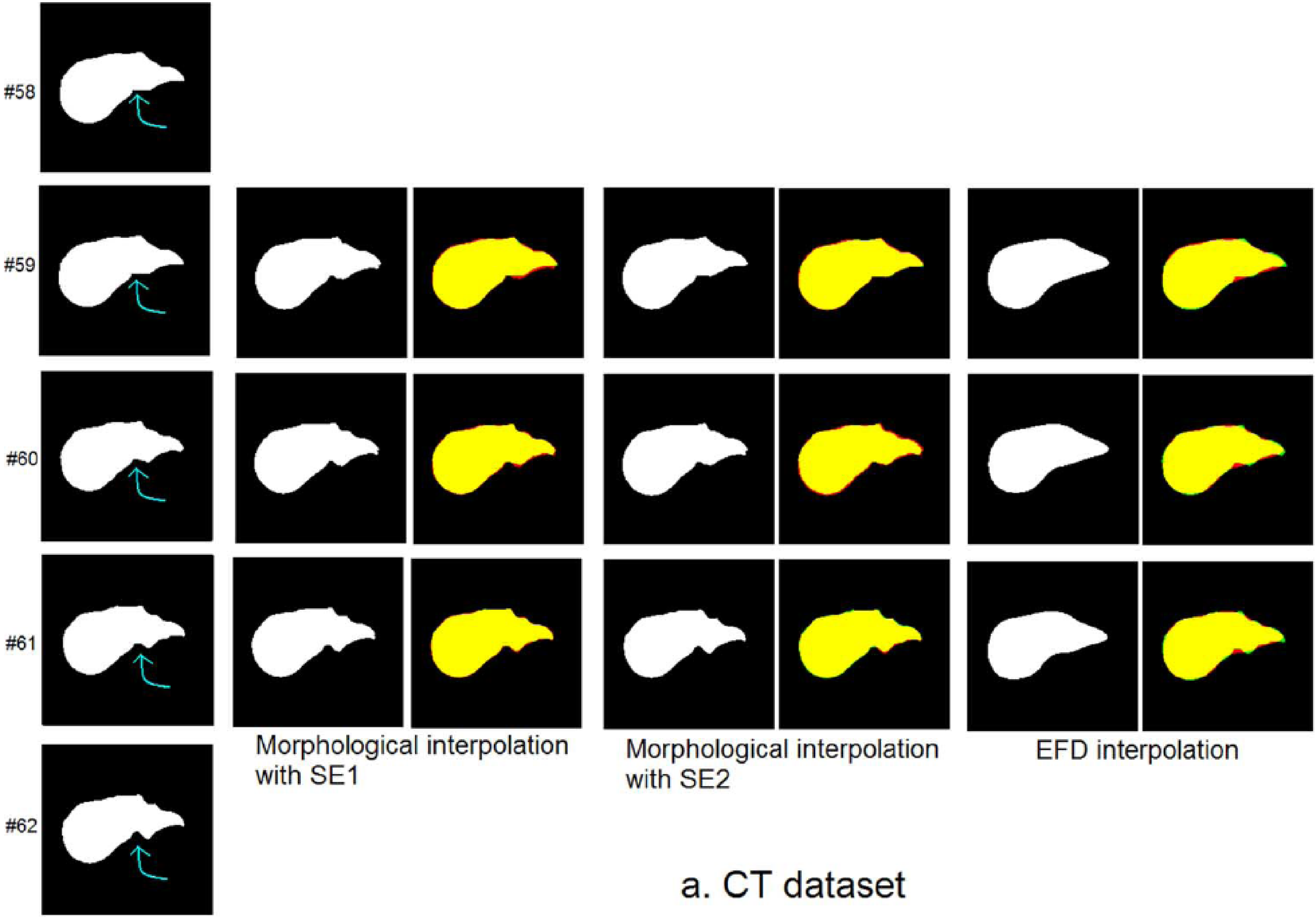
- Once you have segmented an image into objects, how do you describe them?
- Applications:
  - Object identification
  - Content-based image retrieval
  - Object searching/finding/tracking
  - Image registration (especially multimodality)
- Often used as inputs to
  - matching algorithms
  - machine learning algorithms

# Shape Description

- Once you have segmented an image into objects, how do you describe them?
- Applications:
  - Object identification
  - Content-based image retrieval
  - Object searching/finding/tracking
  - Image registration (especially multimodality)
- Often used as inputs to
  - matching algorithms
  - machine learning algorithms

# Shape Description (cont'd)

- Desirable properties:
  - Compact representation
  - Invariant to as many transformations as possible (translation, rotation, scale, etc.)
  - Useful for matching—
    relatively insensitive to small variations
- Approaches:
  - Describe the boundary (contour) of the object
  - Describe the region occupied by the object (next lecture)


#58
#59
#60
#61
#62
Morphological interpolation with SE1
Morphological interpolation with SE2
EFD interpolation

a. CT dataset

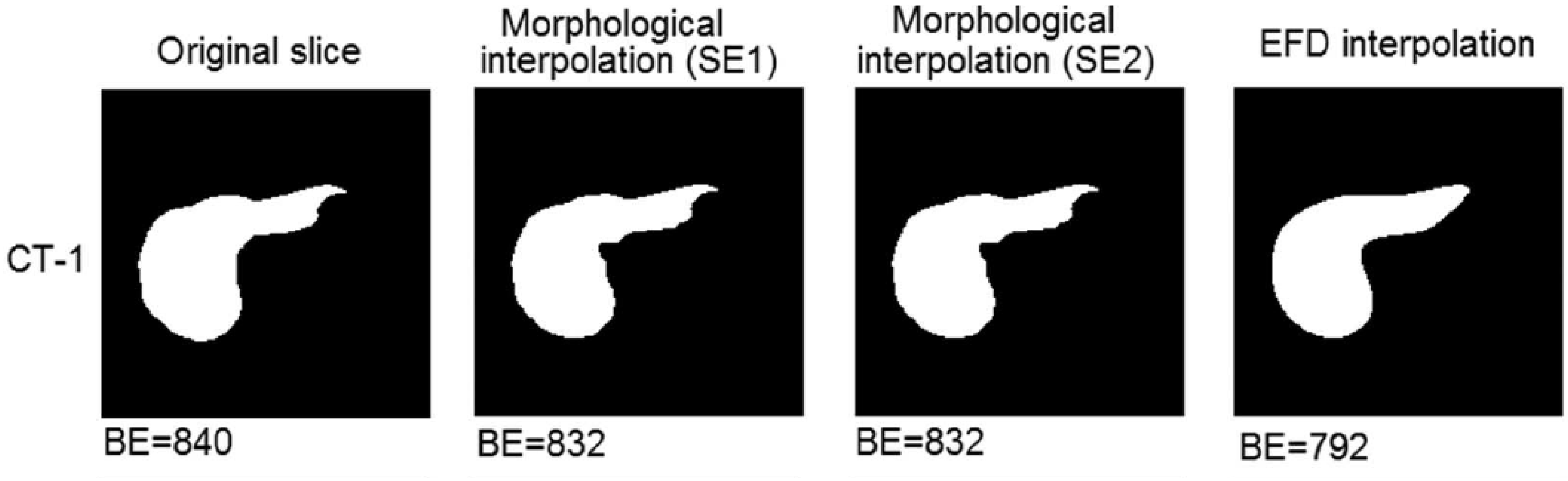
# Shape Description (cont'd)

- Desirable properties:
  - Compact representation
  - Invariant to as many transformations as possible (translation, rotation, scale, etc.)
  - Useful for matching—
    relatively insensitive to small variations
- Approaches:
  - Describe the boundary (contour) of the object
  - Describe the region occupied by the object (next lecture)

# Simple Geometric Descriptors

- Boundary length (perimeter)
- Bending energy
  - Basic Idea: How much work do you have to do to bend a straight line into the shape?
  - Calculation:
    $$\int_0^1 \kappa(s)^2 \; ds$$
    where $\kappa(s)$ is the curvature at point $s$
- Histograms of geometric properties
  - Orientations
  - Curvature
  - ...

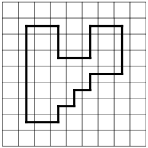
# Example of using bending energy for contour description



# Chain Codes and Variations

- Chain code: encode direction around the border between pixels
- Differential chain code: *change* in direction around the border (differences between chain code numbers modulo 4 or 8)
- Shape number: differential CCs normalized for starting point– rotate differential chain code to be as small a number as possible

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



| | |
|---|---|
| Chain | 0 0 6 6 0 0 2 2 0 0 6 6 6 4 4 6 4 6 4 6 4 4 2 2 2 2 2 2 |
| Differential | 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 0 0 0 0 0 |
| Shape No. | 0 0 0 0 0 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 |

# Contour Encoding: Chains

- Encode the border of the region only.
- Only need to encode relative direction around the border.

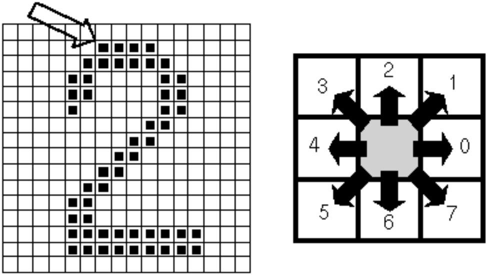
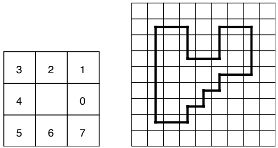



**Figure 3.1** An example chain code; the reference pixel is marked by an arrow: 00007766555555660000000644444444222111111223444565221 1.

# Chain Codes and Variations

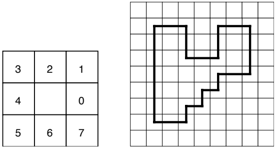
- Chain code: encode direction around the border between pixels
- Differential chain code: *change* in direction around the border (differences between chain code numbers modulo 4 or 8)
- Shape number: differential CCs normalized for starting point– rotate differential chain code to be as small a number as possible




| | |
|---|---|
| Chain | 0 0 6 6 0 0 2 2 0 0 6 6 6 4 4 6 4 6 4 6 4 4 2 2 2 2 2 2 |
| Differential | 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 0 0 0 0 0 |
| Shape No. | 0 0 0 0 0 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 |

# Chain Codes and Variations

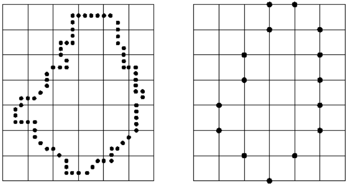
- Chain code: encode direction around the border between pixels
- Differential chain code: *change* in direction around the border (differences between chain code numbers modulo 4 or 8)
- Shape number: differential CCs normalized for starting point– rotate differential chain code to be as small a number as possible




| | |
|---|---|
| Chain | 0 0 6 6 0 0 2 2 0 0 6 6 6 4 4 6 4 6 4 6 4 4 2 2 2 2 2 2 |
| Differential | 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 0 0 0 0 0 |
| Shape No. | 0 0 0 0 0 6 0 6 0 2 0 2 0 6 0 6 0 0 6 0 2 6 2 6 2 6 0 6 |

# Chain Codes: Smoothing and Resampling

- Problem:
  Pixel grid and noise cause change in chain code (and its length)
- Approach:
  Smooth the shape and/or resample to some fixed number of points (code length)

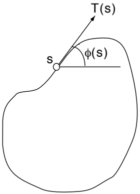


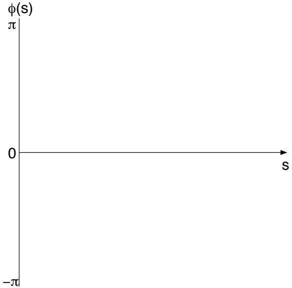
a b
c d

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

# Tangential Representations: $\psi$-$s$ curves

- Encodes the tangent angle as a function of arclength
- Similar to differential chain codes, but not limited to grid (average orientation over small sections of the boundary)
- Sample a fixed number of points around the boundary



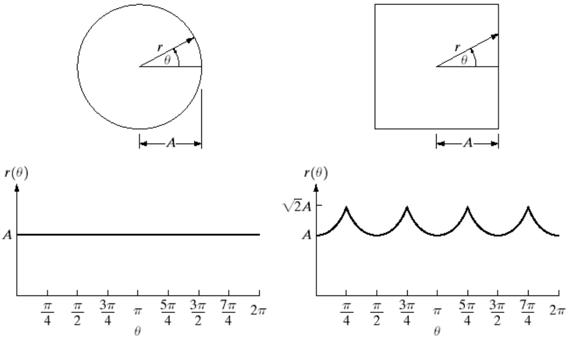



# Radial Representations: *r*-*s* Curves

- Encodes the distance from the center as a function of arclength

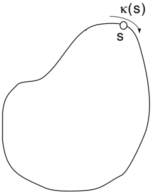
a b

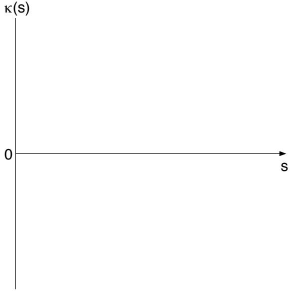
**FIGURE 11.5** Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A \sec\theta$ for $0 \leq \theta \leq \pi/4$ and $r(\theta) = A \csc\theta$ for $\pi/4 < \theta \leq \pi/2$.



# Curvature Representations

- Encodes the curvature $\kappa(s)$ as a function of arclength
- Differentiates the tangent vector (not quite the same as differentiating the $\psi$-$s$ curve, but close)



# Signatures

- In general, a *signature* is a 1-dimensional function describing a shape (chain code, differential chain code, $\psi$-$s$ curve, $r$-$s$ curve, etc.)
- Here's another: orthogonal distance $d(s)$ to opposing side as a function of arclength $s$

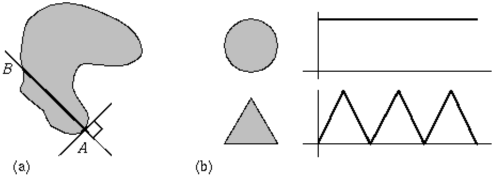



**Figure 6.9** *Signature: (a) Construction, (b) signatures for a circle and a triangle.*

# Chord Distribution

- Measure length of all chord from one point on the boundary to another.
- Histogram of lengths: invariant to rotation
- Histogram of angles: invariant to scale

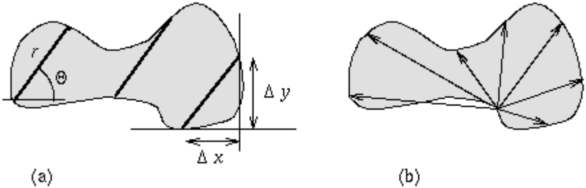


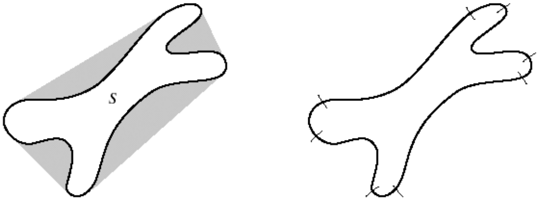

**Figure 6.10** *Chord distribution.*

# Points of Extreme Curvature

- Another approach to describing a shape is to decompose it into parts based on *extrema of boundary curvature*
- Decompose into parts
  $\longrightarrow$ compare parts
  $\longrightarrow$ build relationship graphs
- Some evidence that this corresponds to human perception

# Convex Hulls

- Build the convex hull of the shape, look at where the shape touches its convex hull—gives you an idea of the "extremal" points or sections of the curve.




a b

**FIGURE 11.6**
(a) A region, $S$, and its convex deficiency (shaded).
(b) Partitioned boundary.

# Fourier Descriptors

- Can think of the boundary pixels $(x_k, y_k)$ as a curve in the complex plane:

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

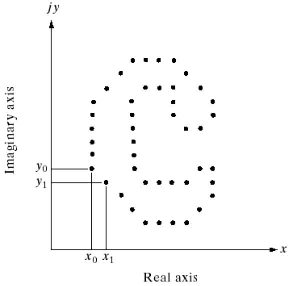



**FIGURE 11.13** A digital boundary and its representation as a complex sequence. The points $(x_0, y_0)$ and $(x_1, y_1)$ shown are (arbitrarily) the first two points in the sequence.

# Fourier Descriptors (cont'd)

- A *Fourier descriptor* is the Fourier Transform of the complex-valued boundary curve:

$$a(\nu) = \mathcal{F}(s(k))$$

- Relatively insensitive to transformations:

| 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.

- Most of the time, just
  - use the magnitude (invariant to start point)
  - ignore the zero-frequency term (invariant to translation)
  - normalize sum (invariant to overall size)

# Why Fourier Descriptors?

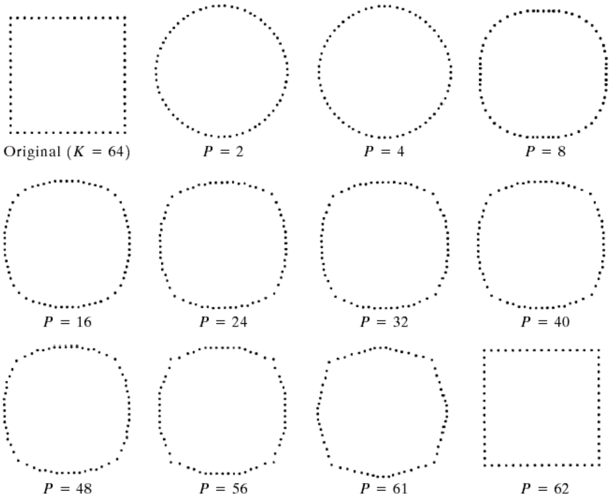
- Separates
  - low-frequency components of shape (general properties)
  - high-frequency ones (detail, small perturbations)
- Can filter shape!
  - Low-pass filtering the Fourier descriptor smooths the shape

# Why Fourier Descriptors?

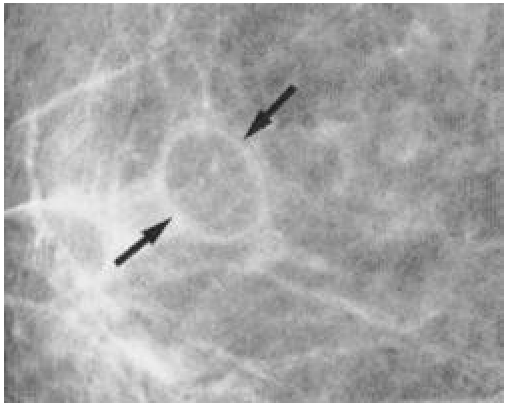
- Separates
  - low-frequency components of shape (general properties)
  - high-frequency ones (detail, small perturbations)
- Can filter shape!
  - Low-pass filtering the Fourier descriptor smooths the shape

# Why Fourier Descriptors?

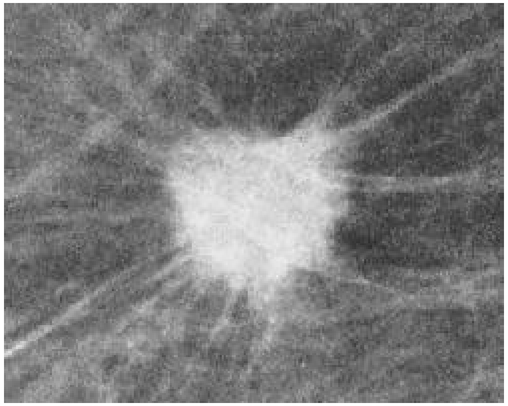
**FIGURE 11.14** Examples of reconstruction from Fourier descriptors. $P$ is the number of Fourier coefficients used in the reconstruction of the boundary.



# Computer-aided diagnosis: Comparing two contours



Circumscribed (benign) lesions in digital mammography



Spiculated lesions in (digital mammography)

The feature of interest: regularity of contour

-The circumscribed shape will have its Fourier coefficients at lower frequencies than the spiculated shape

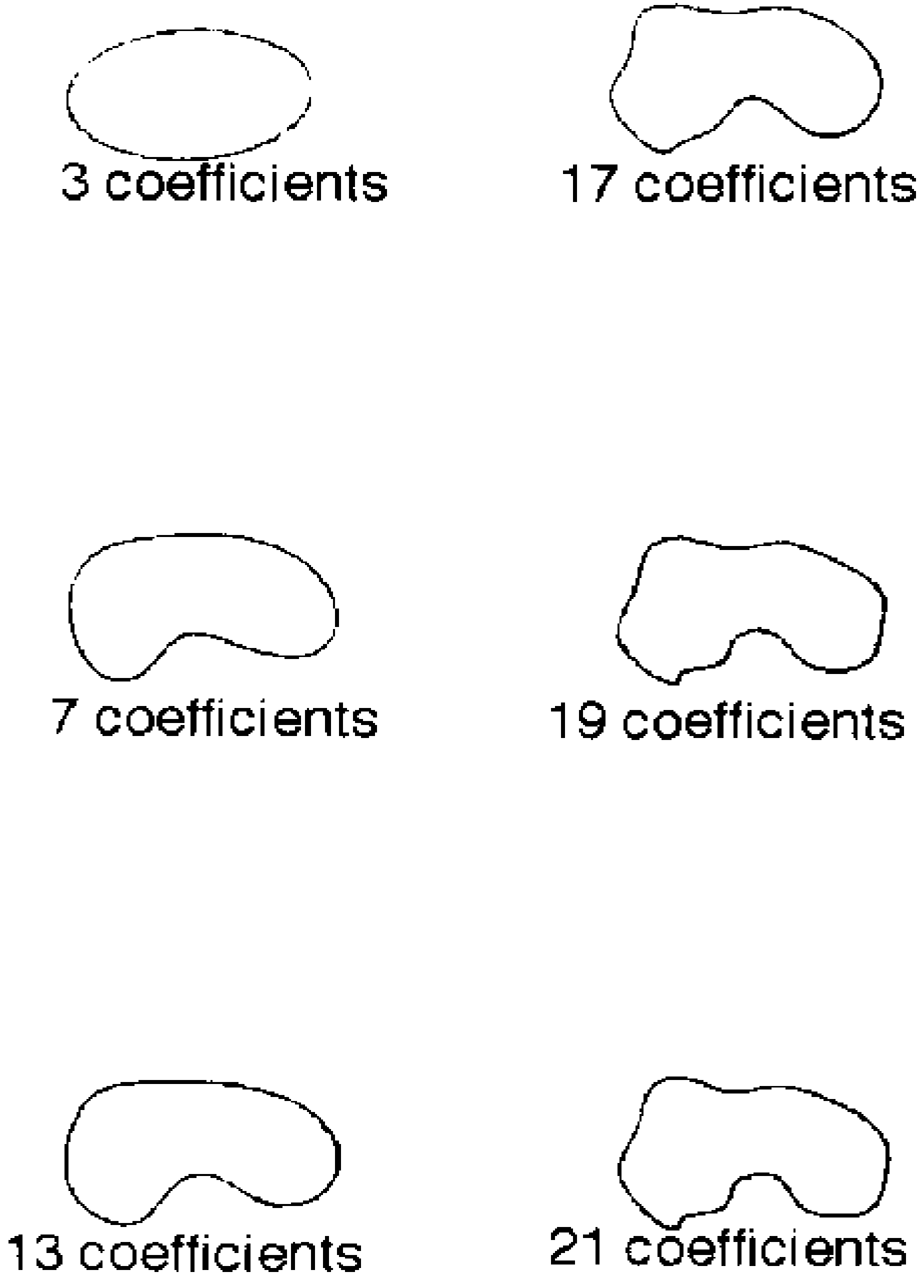



Fig. 8. Progressive approximations to a contour with increasing number of terms in the reconstruction.

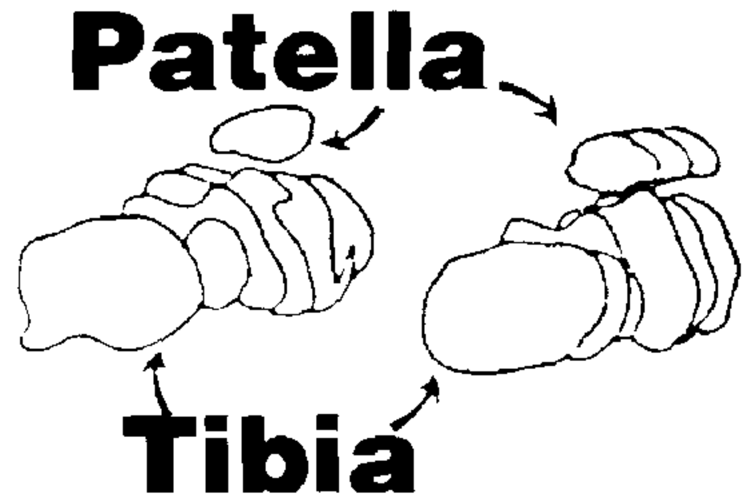



Fig. 6. Reconstruction of knee region anatomy with truncated Fourier series.

# Summary

- There are *lots* of ways to try to describe/quantify the shape of an object from points on the boundary
- Remember the objectives:
  - Compact representation
  - Invariant to as many transformations as possible (translation, rotation, scale, projection, etc.)
  - Relatively insensitive to small variations

## Summary

- There are *lots* of ways to try to describe/quantify the shape of an object from points on the boundary
- Remember the objectives:
  - Compact representation
  - Invariant to as many transformations as possible (translation, rotation, scale, projection, etc.)
  - Relatively insensitive to small variations

# Example of using contour descriptors

Reading:

- ELEC 536
  - Jeong and Radke, “Reslicing axially sampled 3D shapes using elliptic Fourier descriptors”, Medical Image Analysis 2007

Main idea:

- Contour-based interpolation
- Interpolation between parallel slices from a 3D shape is necessary for reslicing and putting into correspondence organ shapes acquired from volumetric medical imagery

# Rationale

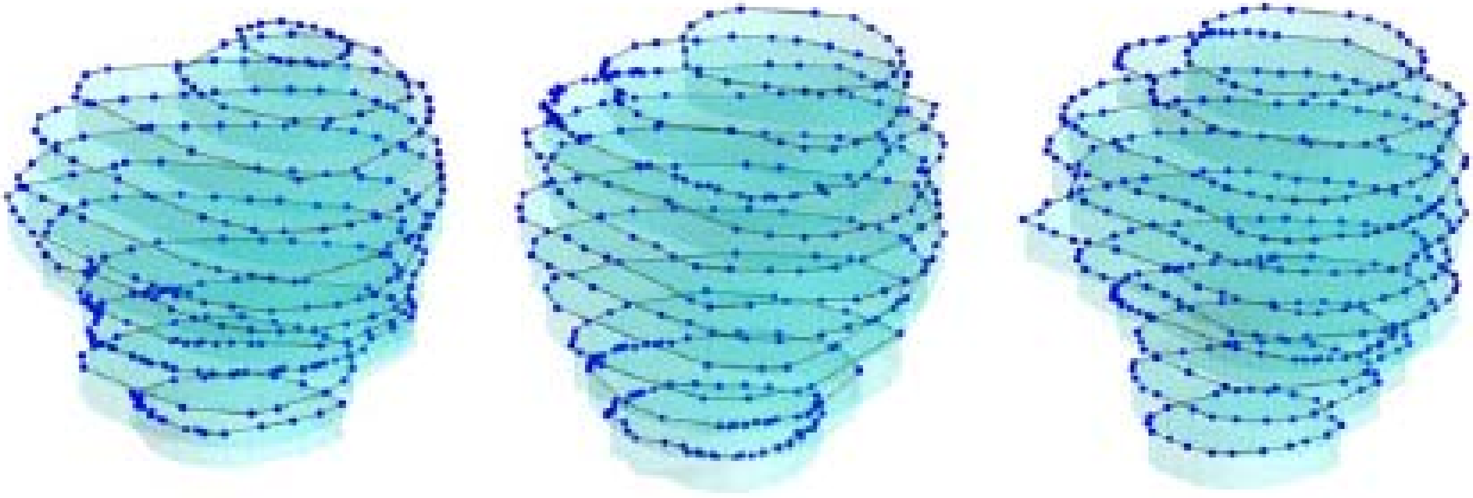


Fig. 1. Axial slices of one patient's prostate acquired on three different days of radiation treatment, contoured from CT imagery. The number of axial slices for each dataset and the number of sample points around each contour generally vary between datasets, and the spacing of the sample points around each contour is usually nonuniform.

# Interpolation using Elliptic Fourier descriptors

- The main goal of the elliptical Fourier analysis is to approximate a closed contour as the sum of elliptical harmonics.

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix}$$

$$a_0 = \frac{1}{2\pi} \sum_{j=0}^{R-1} x(t_j), \qquad c_0 = \frac{1}{2\pi} \sum_{j=0}^{R-1} y(t_j),$$

$$a_k = \frac{1}{\pi} \sum_{j=0}^{R-1} x(t_j) \cos kt_j, \qquad b_k = \frac{1}{\pi} \sum_{j=0}^{R-1} x(t_j) \sin kt_j,$$

$$c_k = \frac{1}{\pi} \sum_{j=0}^{R-1} y(t_j) \cos kt_j, \qquad d_k = \frac{1}{\pi} \sum_{j=0}^{R-1} y(t_j) \sin kt_j.$$

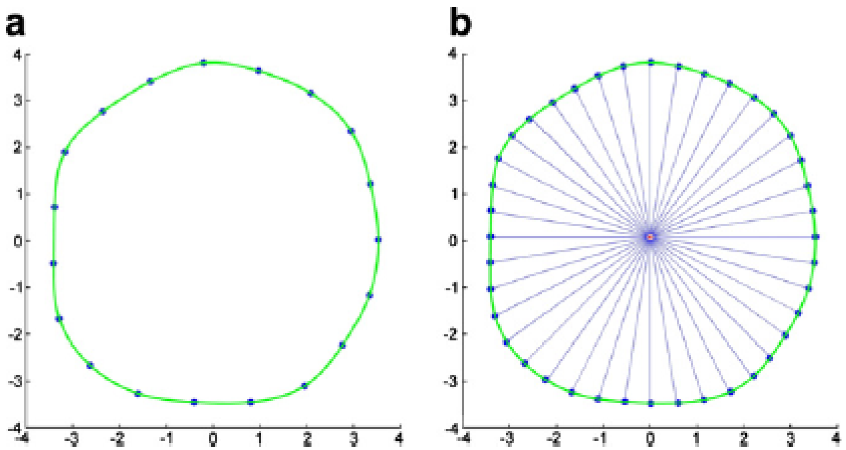
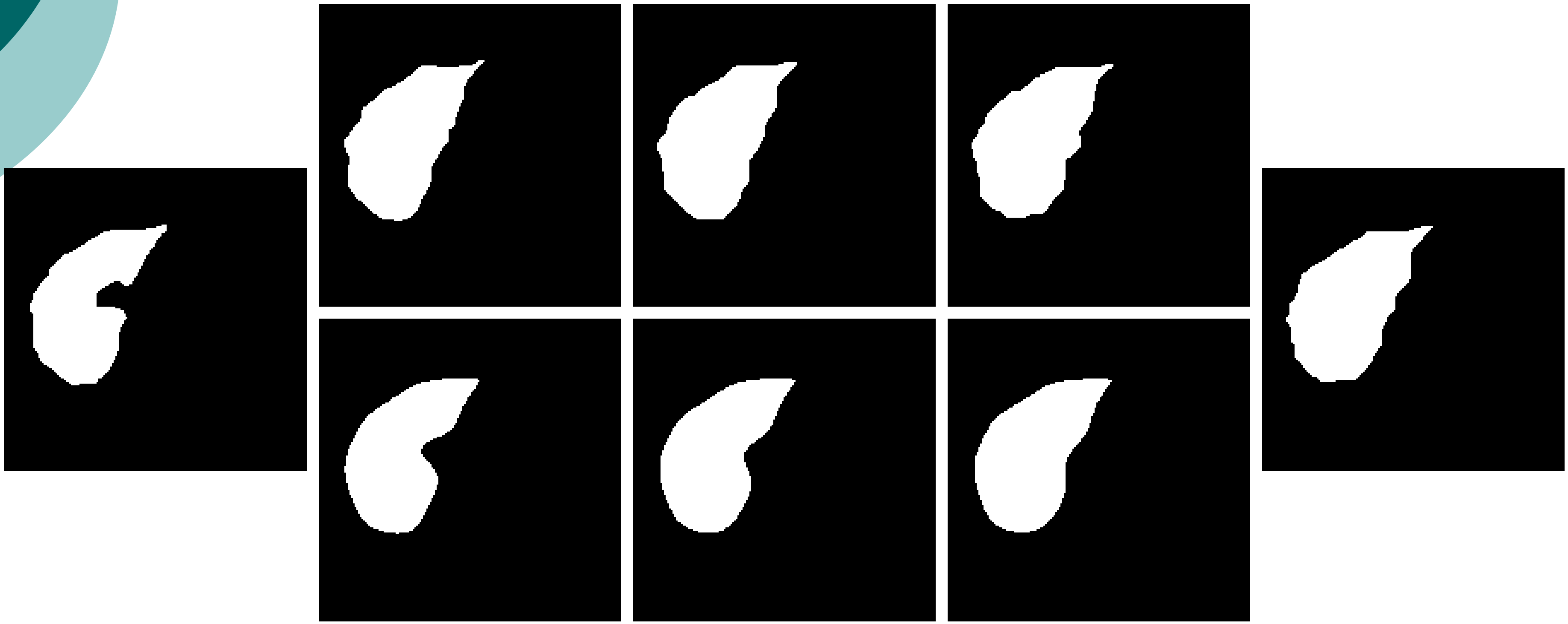



Fig. 3. (a) The original set of points on a slice. (b) The set of points after spline interpolation and resampling.

# Contour Interpolation via EFD descriptors

- Input: 2 or more contour images
- Convert each contour to EFDs
- Assign a z-value to each contour (frame #, or distance between MRI slices)
- Choose z-value(s) at which to interpolate
- Use bicubic interpolation for EFDs at those values of z
- Convert interpolated EFDs back into x,y contour images

# Experimental validation



# Disadvantages

EFD algorithm does not account for cases where multiple contours are present in the same image