Computer Vision for HCI

2-D Shape

Region Representations/Properties

- Once regions have been identified, properties of regions can be input to higher-level decision-making procedures
  - Recognition or inspection
- Common geometric properties of region useful for simple shape description
  - Boundary, bounding box, extremal points, area, perimeter, compactness/circularity, moments, etc.
- We will examine geometric and shape representation/properties for binary images [(0,1) or (0, 255)]
  - Assumes have segmented out pixels for the object of interest

2

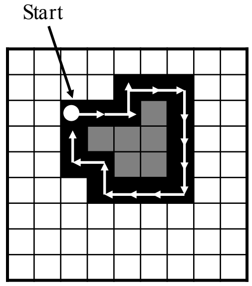
Boundary Coding

- Regions can be represented by their boundaries instead of an image
- Simplest form is linear list of border pixels of each region

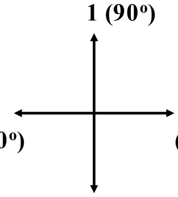
3

Chain Code Representation

- Simple technique for representing shape of contour
- Each directed line segment is assigned code
- Chain code is string of those ordered codes

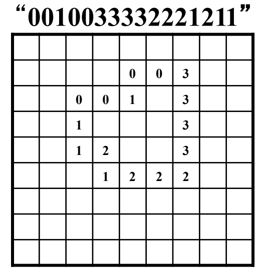


Binary image



90° quantization

Chain code:
"0010033332221211"



4

Shape Number

- Chain codes dependent upon:
  - Orientation and start point of contour
- Invariance to rotation by integer multiples of quantization (90^o)
  - Take "circular" first-difference of chain code: $f(x)$-$f(x$-1)

    Chain code: 0010033332221211

    First-diff: 3013030003003130 (1-2=3!)
  - Could also align grid to main axes of object
- Invariance to starting point
  - Circular shift first-diff number to be minimal integer

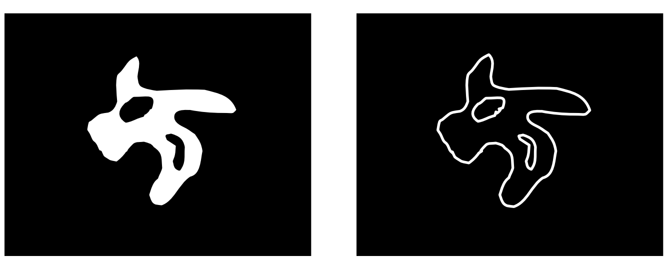
    First-diff: **301303**0003003130

    Shape #: 0003003130**301303**

5

Internal/External Boundaries

- When region has one or more hole boundaries, represented by chain code for each of them
  - Spatial relation between contours?



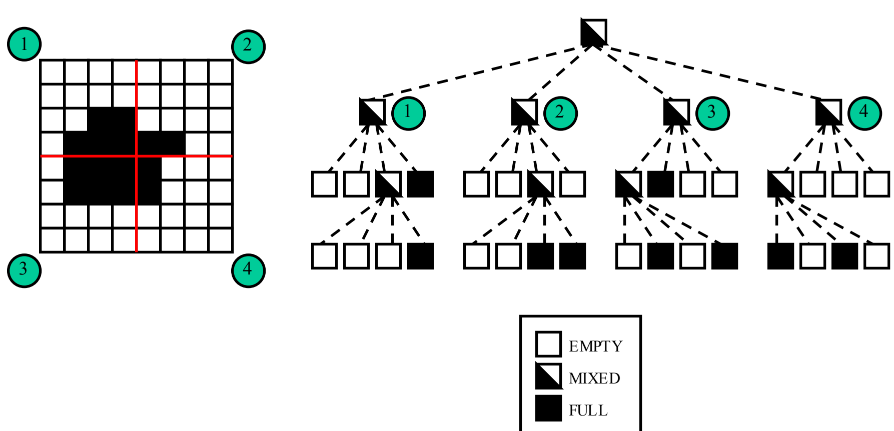
6

Quadtree Representation

- Quadtree encodes entire region (not just boundary)
  - Binary image
- Each node of quadtree represents a square region in the image
  - Has one of 3 labels: FULL, EMPTY, MIXED
- Subdivide nodes until either FULL or EMPTY

7

Quadtree Example



8

Quadtree

- Tree/graph matching for recognition
- Previous example is "blocky"
  - Small image grid
  - Need many more levels/cells (resolution) to approximate curves
- Quadtrees have been used to represent map regions in geographic information systems (GIS)

9

Medial Axis Transform

- "Skeleton" representation of binary region
- Steps
  - Original binary image is f^0
  - Iteratively compute (until no change):

$$f^k(x, y) = f^0(x, y) + \min(f^{k-1}(p, q))$$
$$\forall (p, q) \text{ 4-connected to } (x, y)$$

  - Medial axis given by all points (x,y) such that:

$$f^k(x, y) \geq f^k(p, q)$$

10

Medial Axis Example

$f^0(x,y)$

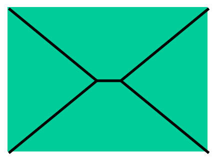
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

$f^1(x,y)$

1	1	1	1	1	1
1	2	2	2	2	1
1	2	2	2	2	1
1	2	2	2	2	1
1	1	1	1	1	1

$f^2(x,y)$

1	1	1	1	1	**1**
1	**2**	2	2	**2**	1
1	2	**3**	**3**	2	1
1	**2**	2	2	**2**	1
1	1	1	1	1	**1**



1					**1**
	1			**1**	
		1	**1**		
	1			**1**	
1					**1**

Medial axis

11

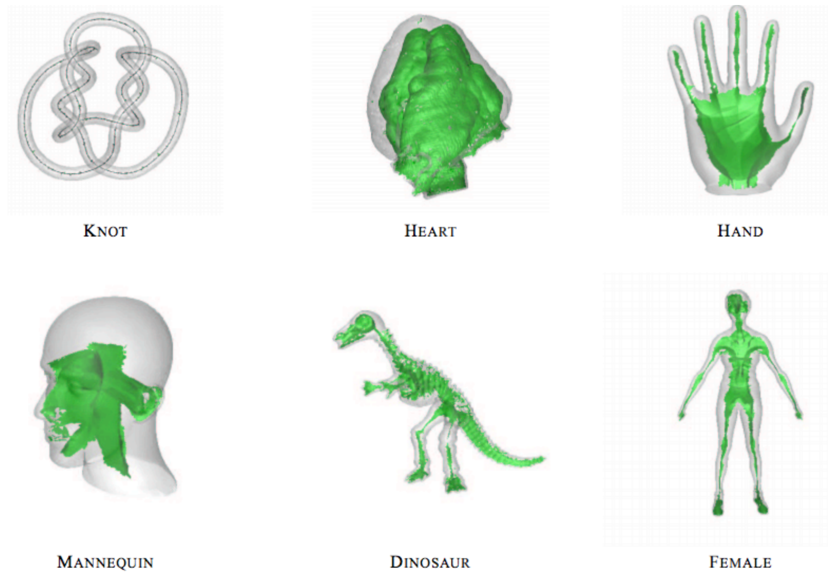
Medial Axis Extras

- Also can recover shape from medial axis with iterative algorithm
- Matlab code available for skeletonization
  - *Skel* function (bwmorph)
  - Also has morphological functions

12

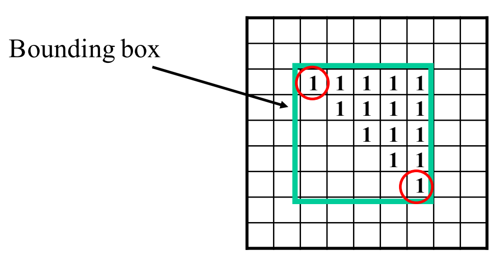
Medial Axis of 3-D Surface

(Prof. Tamal Dey)



Bounding Box

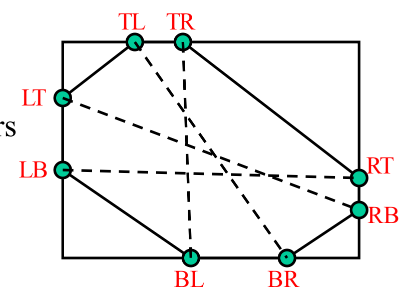
- Often useful to have rough idea of where region located (e.g., for tracking)
- Bounding box is enclosing rectangle that touches topmost, bottommost, leftmost, and rightmost points in region
- As shown, can include much of background



14

Extremal Points

- Examine where region touches bounding box
- At most 8 distinct "extremal" points/pixels for region on bounding box
  - Topmost left/right
  - Rightmost top/bottom
  - Leftmost top/bottom
  - Bottommost left/right
- Extremals occur in opposite pairs
  - Defines axis for each pair
- Compute approximations of
  - Axis length
  - Axis orientation

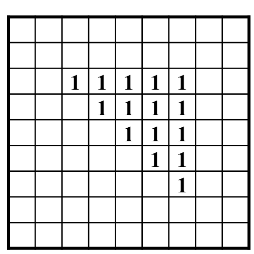


15

Area

- Area A of binary region R simply defined as:

$$A = \sum_{(x,y)\in R} 1$$



Binary image

Area = 15 (pixels)

16

Perimeter and Compactness

- <u>Perimeter</u> P of binary region R is sum of its border pixels
  - – Border pixel has at least 1 background pixel in its neighborhood
- <u>Compactness/Circularity</u> C of region is defined as follows:

$$C = 4\pi \frac{A}{P^2}$$

Circle: $A = \pi r^2$ $P = 2\pi r$

$C = 1$

Square: $A = L^2$ $P = 4L$

$C = \frac{\pi}{4}$

17

Centroid

- Centroid of binary region is <u>average location</u> of pixels in R:

$$x_c = \bar{x} = \frac{1}{N} \sum_{(x,y)\in R} x \qquad y_c = \bar{y} = \frac{1}{N} \sum_{(x,y)\in R} y$$

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

Binary image

$$x_c = \frac{1}{15}\left[2 + (2\cdot 3) + (3\cdot 4) + (4\cdot 5) + (5\cdot 6)\right]$$
$$= 4.67$$
$$y_c = \frac{1}{15}\left[(5\cdot 2) + (4\cdot 3) + (3\cdot 4) + (2\cdot 5) + 6\right]$$
$$= 3.34$$

Useful for target tracking!

18

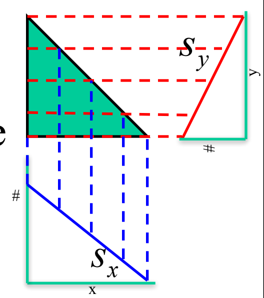
Signatures

- Useful for finding preliminary landmarks
- Horizontal signature of binary image
  - Projection of image onto the x-axis

$$s_x = \sum_y B[x, y]$$

- Vertical signature of binary image
  - Projection of image onto the y-axis

$$s_y = \sum_x B[x, y]$$



(consider figure-8 shape)

19

Spatial Moments

- Spatial moments often used to describe region shape

$$m_{pq} = \sum \sum x^p y^q I[x, y]$$

"Area"

$$A = m_{00} = \sum \sum x^0 y^0 I[x, y] \quad \text{"Zeroth order"}$$

[for binary (0,1) image]

$$m_{10} = \sum \sum x^1 y^0 I[x, y]$$

$$m_{01} = \sum \sum x^0 y^1 I[x, y]$$

"First order"

"Centroid" $\longrightarrow$ $\bar{x} = \dfrac{m_{10}}{m_{00}} \qquad \bar{y} = \dfrac{m_{01}}{m_{00}}$

20

Central Moments

- Central moments (translation invariant)

$$\mu_{pq} = \sum \sum (x - \bar{x})^p (y - \bar{y})^q I[x, y]$$

$$\bar{x} = \frac{m_{10}}{m_{00}} \qquad \bar{y} = \frac{m_{01}}{m_{00}}$$

21

Second-Order Central Moments

- Three second-order central moments

$$\mu_{20} = \sum \sum (x - \bar{x})^2 (y - \bar{y})^0 I[x, y]$$

$$\mu_{11} = \sum \sum (x - \bar{x})^1 (y - \bar{y})^1 I[x, y]$$

$$\mu_{02} = \sum \sum (x - \bar{x})^0 (y - \bar{y})^2 I[x, y]$$

22

Moment Ellipse Orientation

- If region is ellipse, second-order central moments have useful algebraic description of orientation

Let,

$$a = \mu_{20} = \sum\sum (x-\bar{x})^2 (y-\bar{y})^0 I[x,y]$$
$$b = \mu_{11} = \sum\sum (x-\bar{x})^1 (y-\bar{y})^1 I[x,y]$$
$$c = \mu_{02} = \sum\sum (x-\bar{x})^0 (y-\bar{y})^2 I[x,y]$$

23

Moment Ellipse Orientation

- Resulting orientation relationship

$$\tan(2\theta) = \frac{2b}{a-c}$$

- If $(b = 0)$ and $(a = c)$
  - Object is too symmetric to allow definition of axis
- Can also use eigenvalues and eigenvectors to determine ellipse

24

Similitude Moments

(Invariant to translation and scale)

$$\eta_{ij} = \frac{\mu_{ij}}{\left(m_{00}\right)^{\frac{i+j}{2}+1}} = \frac{\sum\sum (x-\overline{x})^i (y-\overline{y})^j I[x,y]}{\left(\sum\sum I[x,y]\right)^{\frac{i+j}{2}+1}}$$

for $2 \le (i+j) \le 3$:

$$\mathrm{N} = \begin{bmatrix}\eta_{02} & \eta_{03} & \eta_{11} & \eta_{12} & \eta_{20} & \eta_{21} & \eta_{30}\end{bmatrix}$$

25

Binary Images Only?

- NO!
- Can use equations for grayscale (or real-valued) image

26

Hough Transform

- Used to identify/represent shapes (lines, circles, etc.) in images
- Hough transform computes all possible solutions for a given shape
  - Solution with majority of votes is selected

27

Hough Transform of Straight Line

- Fit equation of straight line to edge points in image
- Equation of line:

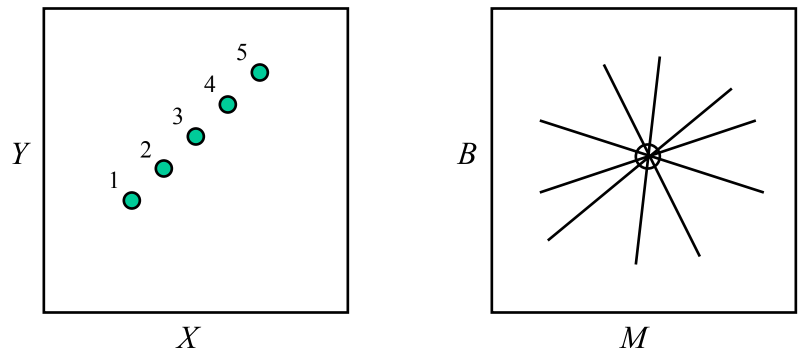
  $\boldsymbol{y} = m\boldsymbol{x} + b$ (m slope, b intercept)
- Rewrite as:

  $b = (-\boldsymbol{x})m + \boldsymbol{y}$ ($-x$ slope, y intercept)

  Line equation in (b,m) space
- Each point in (x,y) space maps to line in (b,m)
- These lines intersect at single point in (b,m) space
  - Intersection point is estimated slope and intercept of line in (x,y) space

28

Hough Transform of Straight Line



29

Hough Transform of Straight Line

- Algorithm
  - Quantize parameter space
    $P[b_{\min}, \ldots, b_{\max}][m_{\min}, \ldots, m_{\max}]$
  - For each point (x,y) do
    for $m = m_{\min} : m_{\max}$
      $b = (-x)*m + y$
      quantize b
      $P[b][m]$++
  - Find local maxima in parameter space P

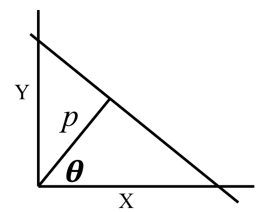
Note: multiple lines will give multiple peaks/maxima

30

Hough Transform of Straight Line

- Parameterization has problem
  - Vertical line as infinite slope *m*
- Another parameterization of line (Normal form):

$$p = x\cos(\theta) + y\sin(\theta)$$

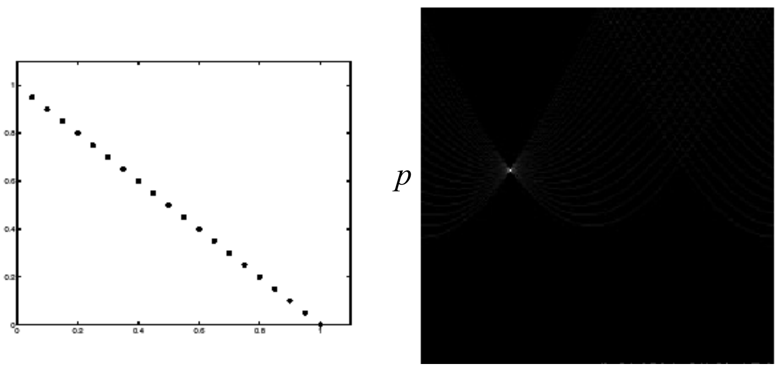


- Both parameters fully defined
- Can calculate using gradient angle
  - Compute during edge detection
  - Not have to loop through all angles $\boldsymbol{\theta}$

Hough Transform of Straight Line

- Algorithm
  - Quantize parameter space
    $P[\theta_{min},\ldots,\theta_{max}][p_{min},\ldots,p_{max}]$
  - For each "edge" point (x,y) do
    Recall gradient θ for (x,y)
    $p = x*\cos(\theta) + y*sin(\theta)$
    quantize θ, p
    $P[\theta][p]$++
  - Find local maxima in parameter space *P*

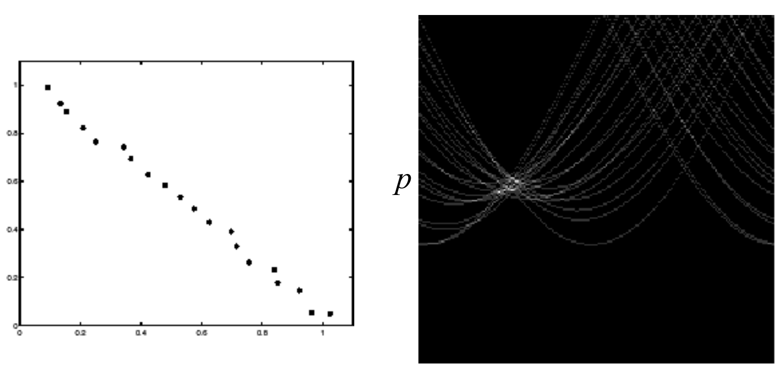
Example 1



Points drawn from a line

33

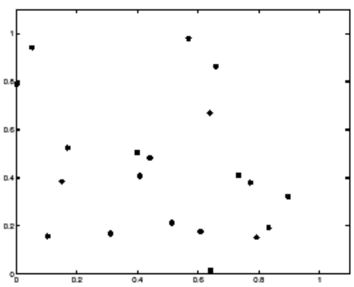
Example 2



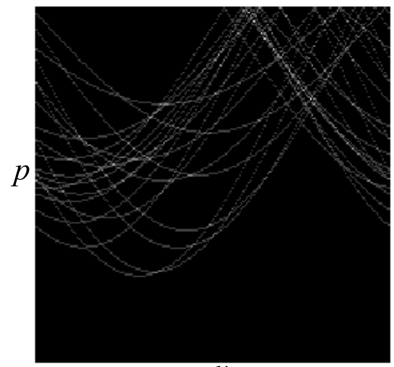
Points drawn from a line,
but offset by random noise

34

Example 3



Random points



()

Generalized Hough Transform

- If you can parameterize the shape, then can make its Hough Transform representation
- Therefore the method can be used to detect an arbitrary object (if it can be parameterized)
  - See "R-Table"

Summary

- Given a 2-D binary shape, use representations/properties to characterize the region
  - Recognition or matching
- Methods
  - Chain code
  - Quadtree
  - Medial axis
  - Bounding box, extremal points
  - Area, centroid
  - Perimeter, compactness, circularity
  - Signatures
  - Moments
- Hough Transform
  - Used to identify/represent shapes in images by searching all possible "parameters"
- Matlab
  - regionprops()
  - bwmorph: skel

37