ECE5554 – Computer Vision
Lecture 6b – Curvature and Morphology

Creed Jones, PhD

Today’s Objectives

Curvature

- Curvature definition
- Discrete Curve Evolution (DCE)

Binary Morphology

- Erosion and Dilation
- Zhang-Suen skeletonization
- Discrete Skeleton Evolution (DSE)

NOTE!

- We are halfway through the course!
- Well done
- Keep it up

CURVATURE

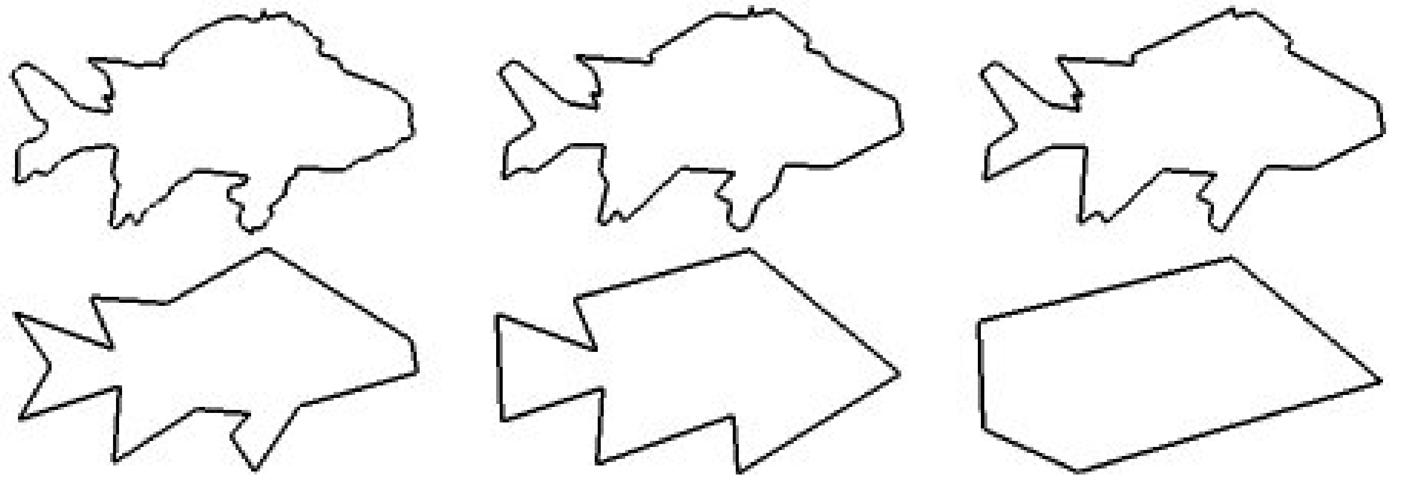
Contours and other curves have a property called *curvature* – it's intuitively attractive but not simple to define

- It's most often defined in terms of first and second derivatives of a curve, expressed parametrically

- A common definition for curvature at the i^{th} point in the curve is:

$$\kappa(i) = \sqrt{\frac{\left(\frac{d^2y}{di^2}\frac{dx}{di} - \frac{d^2x}{di^2}\frac{dy}{di}\right)^2}{\left(\frac{d^2x}{di^2} + \frac{d^2y}{di^2}\right)^3}}$$

Discrete Curve Evolution is a method of simplifying curves (such as contours) by removing segments of noise or detail



L.J. Lateck, iDaniel De Wildt, Jianying Hu - Extraction of key frames from videos by optimal color composition matching and polygon simplification, IEEE Fourth Workshop on Multimedia Signal Processing, January 2001

Every vertex in a polygon can be assigned a relevance that is a measure of how much it “adds” to the overall shape

Consider a vertex v_i at which two line segments s_i and s_{i+1} meet

- Define $\theta(i)$ as the *turn angle* from s_i to s_{i+1}
- We write the length of s_k as $l(s_k)$
- The relevance measure is then:

$$K(i) = \frac{abs\big(\theta(i)\big)l(i)l(i+1)}{l(i)+l(i+1)}$$

Every vertex in a polygon can be assigned a relevance that is a measure of how much it "adds" to the overall shape

Consider a vertex v_i at which two line segments s_i and s_{i+1} meet

- Define $\theta(i)$ as the *turn angle* from s_i to s_{i+1}
- We write the length of s_k as $l(s_k)$
- The relevance measure is then:

$$K(i) = \frac{abs\big(\theta(i)\big)l(i)l(i+1)}{l(i) + l(i+1)}$$

- For a 2D contour,

$$\theta(i) = \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) - \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$

The Discrete Curve Evolution algorithm iteratively removes the vertex that contributes least to the overall shape

VIRGINIA TECH. ece BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

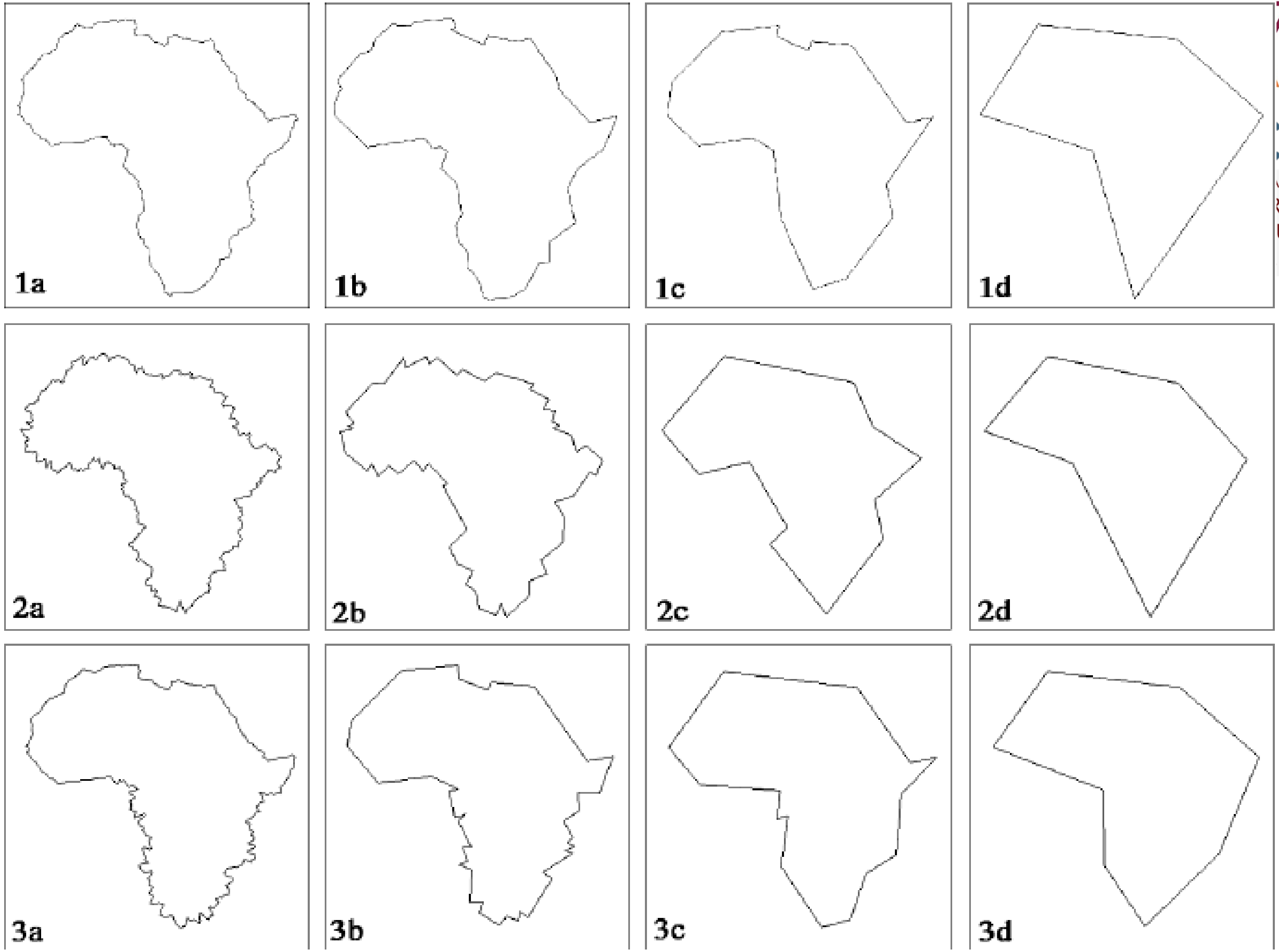
The relevance measure is:

$$K(i) = \frac{abs\big(\theta(i)\big)l(i)l(i+1)}{l(i)+l(i+1)}$$

The Discrete Curve Evolution algorithm is as follows:

- do
  - Evaluate $K(i)$ for all points in the curve
  - Remove the point v_{min} with lowest K
  - It's two line segments are replaced by a new segment joining v_{min-1} and v_{min+1}
- until the resulting curve is "satisfactory"
  - Satisfactory might mean:
    - "having no vertices with relevance less than some K_{min}"
    - "having no fewer than *n* vertices"
    - other criteria related to perimeter or other shape feature

Examples of applying DCE to the outline of Africa, with noise added to all and to part of the contour



Barkowsky, Latecki and Richter, Schematizing Maps: Simplification of Geographic Shape by Discrete Curve Evolution, 2000

BINARY MORPHOLOGY, BRIEFLY

Binary morphology

- Most morphological operations are binary in nature
  - The source image is foreground / background
  - foreground = 1, background = 0
- Binary morphological operations are then:

$$I_{out}(x,y) = \underset{k \in N, l \in N}{LogicalOp}\left[I_{in}(x+k, y+l) \right]$$

- Most often, the neighborhood is 3 by 3
- As with kernel operations, we operate from a source to a destination image

Erosion will only allow foreground pixels to remain if all of the pixels in the neighborhood are foreground

- Pixels around the perimeter of objects become background, since some of their neighborhood is background
- Called Erosion, since the edges of foreground areas are reduced or eroded away

There are three common kinds of erosion operation, with different conditions for retaining the center pixel in the neighborhood

1. **4-neighbor:**
   zero the pixel if any of the 4 orthogonal neighbors are zero

1	1	1
1	1	0
1	1	0

→ 0

2. **8-neighbor:**
   zero the pixel if any of the 8 neighbors are zero

1	1	1
1	1	1
1	1	0

→ 0

3. **Directional:**
   for example, zero the pixel if the neighbor to the right is zero

1	1	0
1	1	0
1	1	0

→ 0

- 8-neighbor erosion is thinner – more *eroded*
- Require that a pixel have <u>all 8</u> neighbors be in the foreground in order to remain foreground in the output

Dilation will only allow background pixels to remain background if <u>none</u> of the pixels in the neighborhood are foreground

- Pixels just outside the perimeter of objects become foreground, since some of their neighborhood is foreground
- Called Dilation, since foreground areas get larger

There are three common kinds of dilation operation, with different conditions for making a dark center pixel bright

1. **4-neighbor:**
   make the pixel 1 if any of the 4 orthogonal neighbors are 1

0	0	0
1	0	0
0	0	0

→ 1

2. **8-neighbor:**
   make the pixel 1 if any of the 8 neighbors are 1

1	0	0
0	0	0
0	0	0

→ 1

3. **Directional:**
   for example, make the pixel 1 if the neighbor to the right is 1

0	0	0
0	0	1
0	0	0

→ 1

- 8-neighbor output is “fatter” – more dilated
- More pixels have at least 1 8-neighbor in the foreground

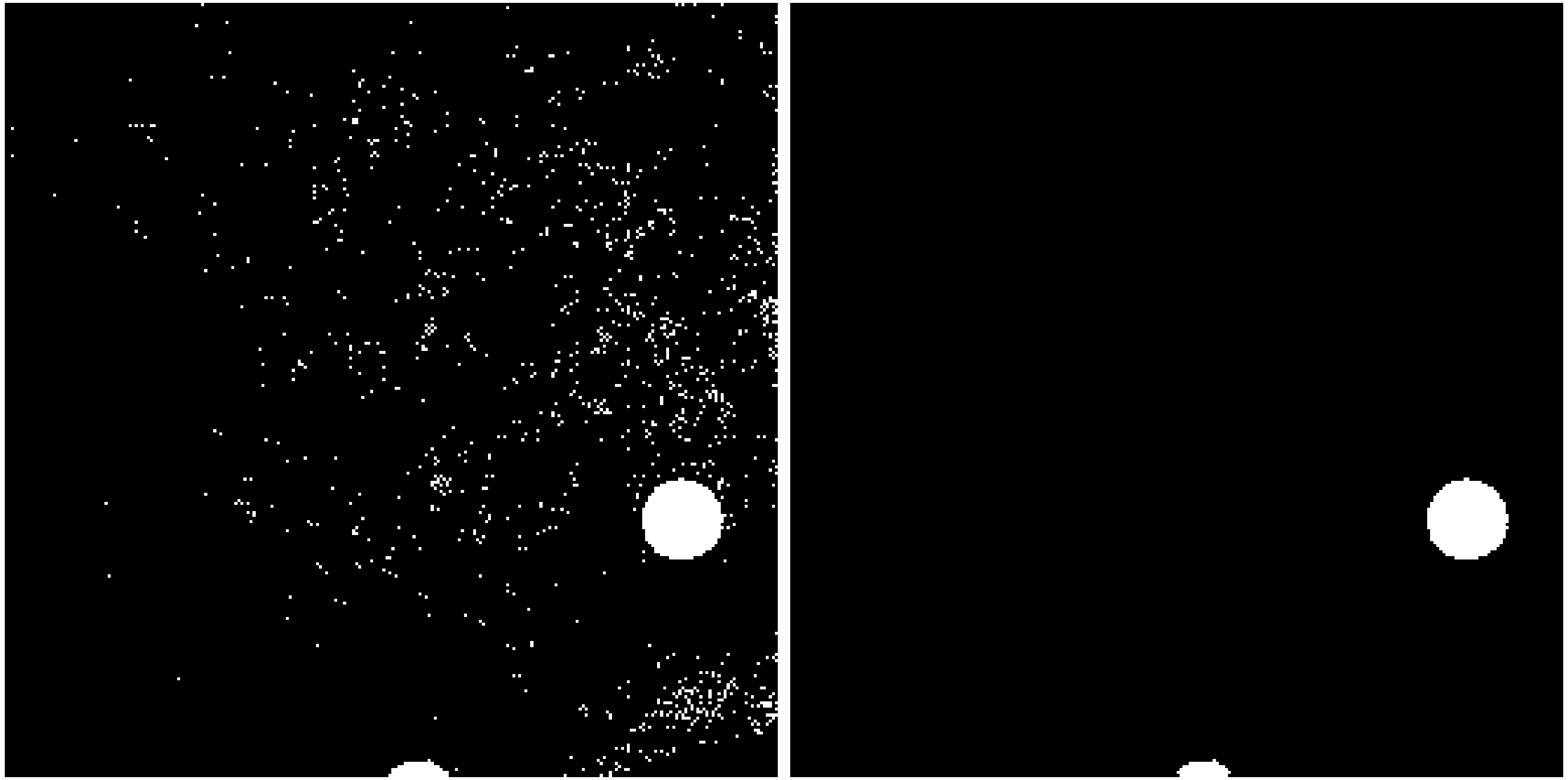
We can summarize some of the types of erosion and dilation by their action on different types of neighborhoods

CENTER PIXEL IN OUTPUT IMAGE	Erosion		Dilation	
	input image center pixel = 0	input image center pixel = 1	input image center pixel = 0	input image center pixel = 1
4-neighbor operation	0	I(c-1, r) & I(c, r-1) & I(c+1, r) & I(c, r+1)	I(c-1, r) | I(c, r-1) | I(c+1, r) | I(c, r+1)	1
8-neighbor operation	0	I(c-1,r-1) & I(c-1,r-1) & I(c-1,r+1) & I(c,r-1) & I(c,r+1) & I(c+1,r-1) & I(c+1,r) & I(c+1, r+1)	I(c-1,r-1) & I(c-1,r-1) & I(c-1,r+1) & I(c, r-1) & I(c,r+1) & I(c+1, r-1) & I(c+1,r) & I(c+1, r+1)	1
directional operation	0	examine certain neighborhood pixels	examine certain neighborhood pixels	1

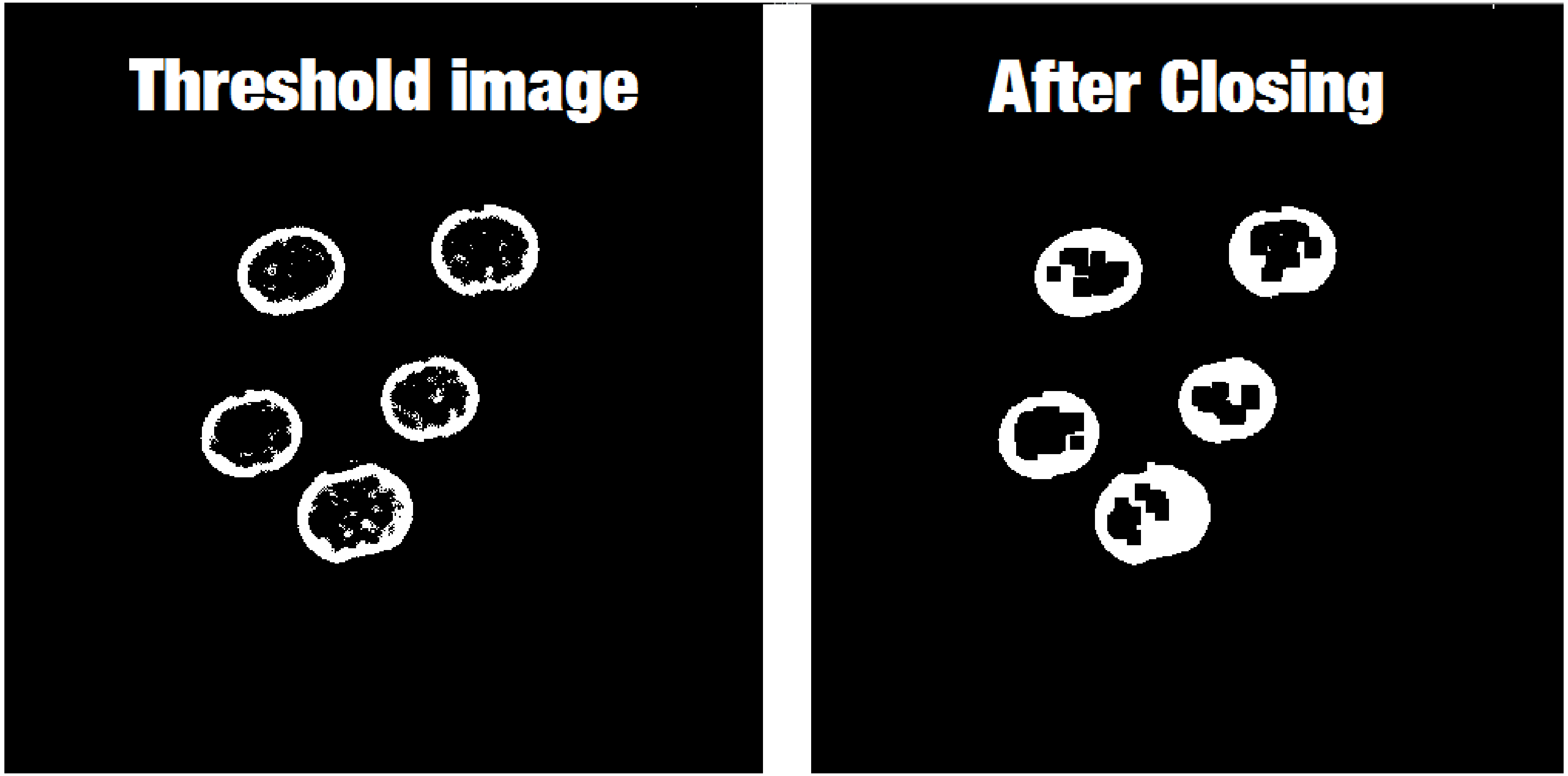
What are erosion and dilation good for?

- removing single-pixel and small objects
  - morphological noise removal
- Checking minimum dimensions on objects
  - If a certain number of erosions splits one object in two, maybe it was too thin
  - (this idea sounds great, but no one ever really uses it!)
- Cleaning up an object contour for further processing

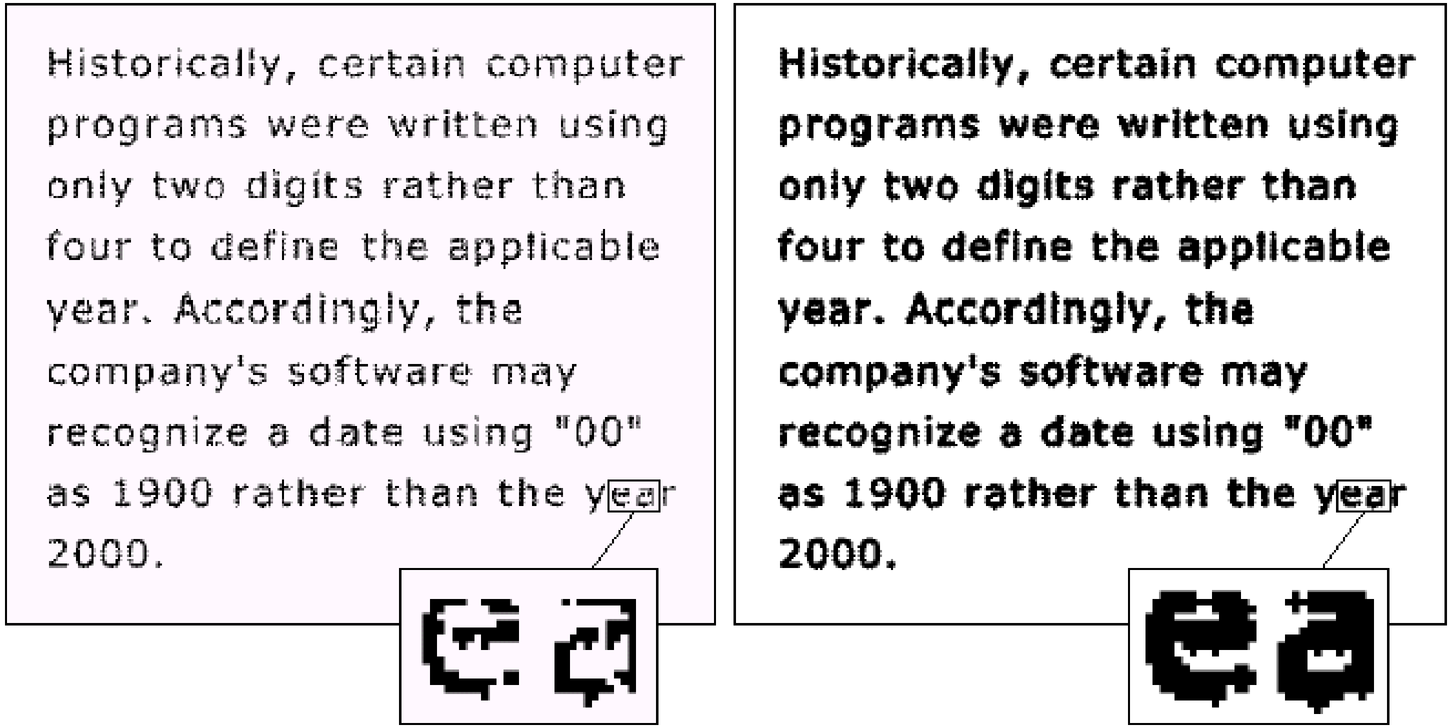
Opening (erosion followed by dilation) will remove small foreground objects



Closing (dilation followed by erosion) will fill in small gaps and holes in foreground objects



Object repair using dilation



0	1	0
1	1	1
0	1	0

a c
b

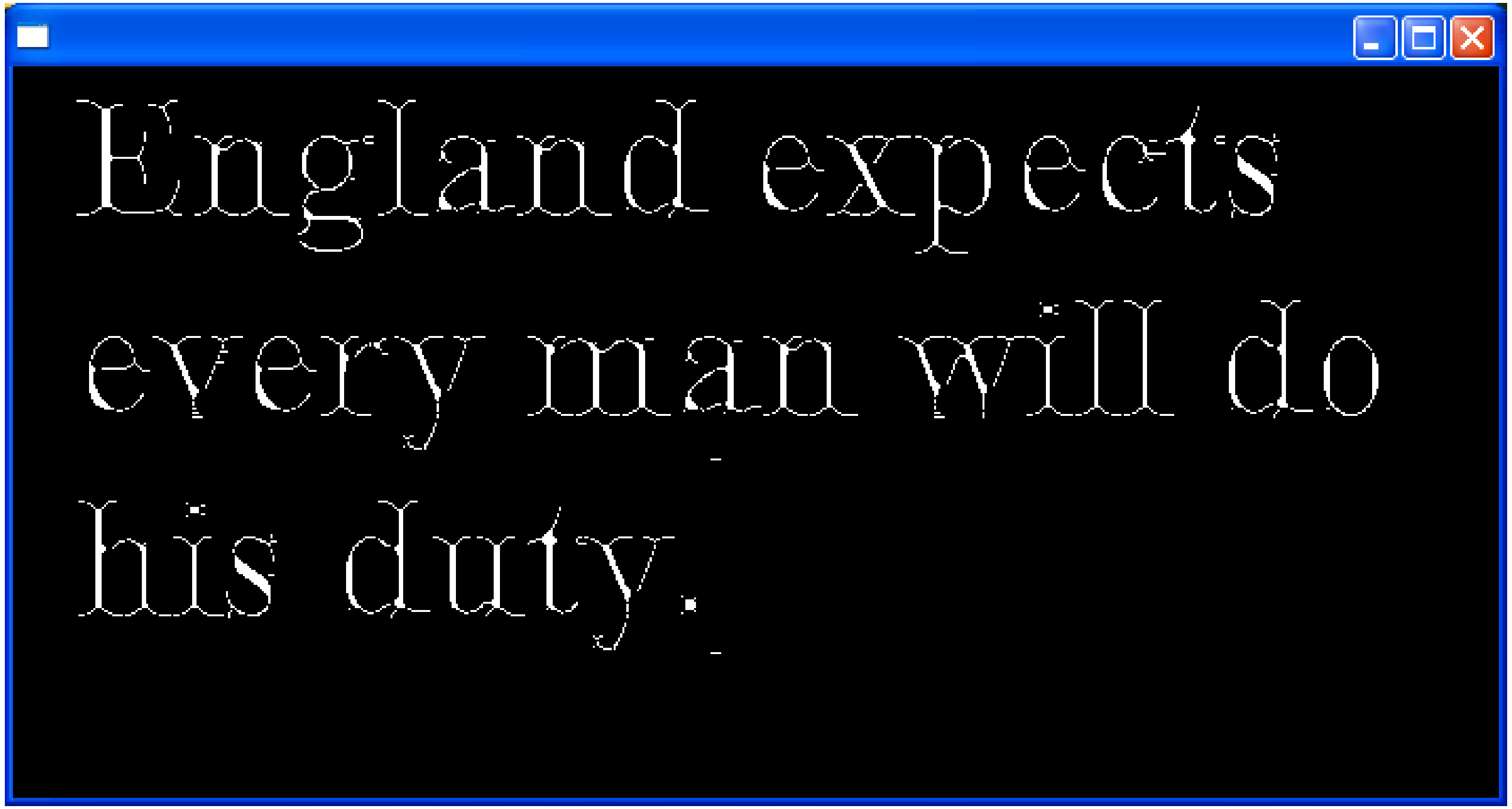
FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

A series of specialized erosion operations can reduce a binary object to its median axis, or “skeleton”

- Repeated erosion will completely remove all objects
- However, if we put some conditions on it –
  - don’t remove pixels that have black on either side
  - don’t snip off ends
- then we can extract the “skeleton” of the object
  - 1-pixel wide line along the “backbone” of objects
- There are many ways to do this, differing in the details…

An example of skeletonization

- This image is not quite fully skeletonized…



Skeletonization or thinning is also known as *center line detection* or *medial axis transformation*

- This is an important binary image process
- Many different algorithms exist
  - With variations and enhancements

- The classic algorithm was first described by Zhang and Suen
  - Zhang, T. Y., and Ching Y. Suen. "A fast parallel algorithm for thinning digital patterns." *Communications of the ACM* 27.3 (1984): 236-239.
  - http://www-prima.inrialpes.fr/perso/Tran/Draft/gateway.cfm.pdf

In the first pass of Zhang and Suen skeletonization, the following steps are taken

In the output image, a foreground point is deleted if all of the following are true:

1. $2 \leq B(P_1) \leq 6$
   (between 2 and 6 of its neighboring pixels are foreground)
2. $A(P_1) = 1$
   (exactly one pair of neighbor pixels, considered in order, is 01 – bg:fg)
3. Any of $\{P_2, P_4, P_6\}$ is background
4. Any of $\{P_4, P_6, P_8\}$ is background

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

$A(P_1) =$ the number of 01 patterns in the ordered set $\{P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

$B(P_1) =$ the number of foreground pixels in the set $\{P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

The second pass of Zhang and Suen skeletonization is similar, with slightly different criteria in steps 3 and 4

In the output image, a foreground point is deleted if <u>all</u> of the following are true:

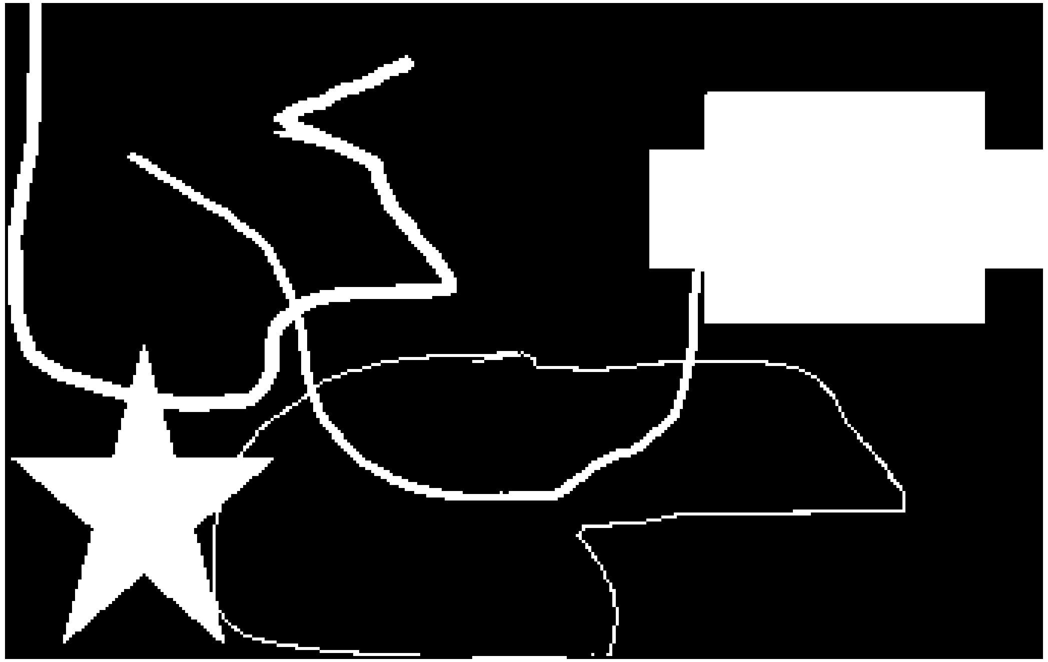
1. $2 \leq B(P_1) \leq 6$
   (between 2 and 6 of its neighboring pixels are foreground)
2. $A(P_1) = 1$
   (exactly one pair of neighbor pixels, considered in order, is 01 – bg:fg)
3. Any of $\{P_2, P_4, \boldsymbol{P_8}\}$ is background
4. Any of $\{\boldsymbol{P_2}, P_6, P_8\}$ is background

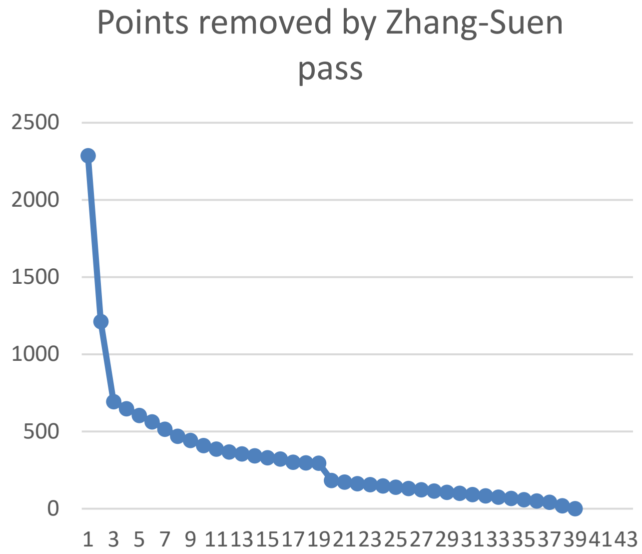
P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

$A(P_1) =$ the number of 01 patterns in the ordered set $\{P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

$B(P_1) =$ the number of foreground pixels in the set $\{P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$

The Zhang and Suen skeletonization preserves spikes and projections, which many other methods tend to either remove or to trim




Points removed by Zhang-Suen pass
2500
2000
1500
1000
500
0
1 3 5 7 9 11131517192123252729313335373941 43

DISCRETE SKELETON EVOLUTION

Discrete Skeleton Evolution simplifies a *skeleton* by removing less relevant branches

- A *skeleton* is a curve that follows the "medial axis" of an object
- Skeletons contain three types of points:
  - Endpoints have only one adjacent point
  - Connection points have exactly two adjacent points
  - Junction points have three or more adjacent points
- An *end branch* is the portion of the skeleton from an endpoint to the nearest junction point

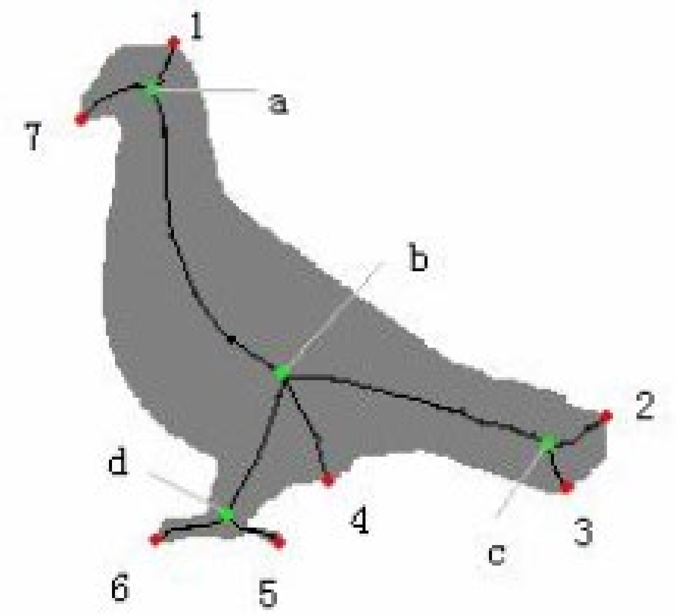


Fig. 2. The endpoints (red) and junction points (green) on the skeleton in Fig. 1(e)

Bai and Latecki, *Discrete Skeleton Evolution*, 2007

Discrete Skeleton Evolution iteratively removes the end branch that contributes least to the ability to reconstruct the original shape

Define the following:

- An end branch from endpoint l_i to junction $f(l_i)$ is written as $P\left(l_i, f(l_i)\right)$
- For a skeleton point s, $B(s)$ is the *maximal disk* centered at s
  - This is the largest disk that will entirely fit within the original shape
- The reconstruction for a given skeleton S is then $R(S) = \bigcup_{s \in S} B(s)$
- The weight of a given end branch is then:

$$w(i) = 1 - \frac{A\left(R\left(S - P\left(l_i, f(l_i)\right)\right)\right)}{A\left(R(S)\right)}$$

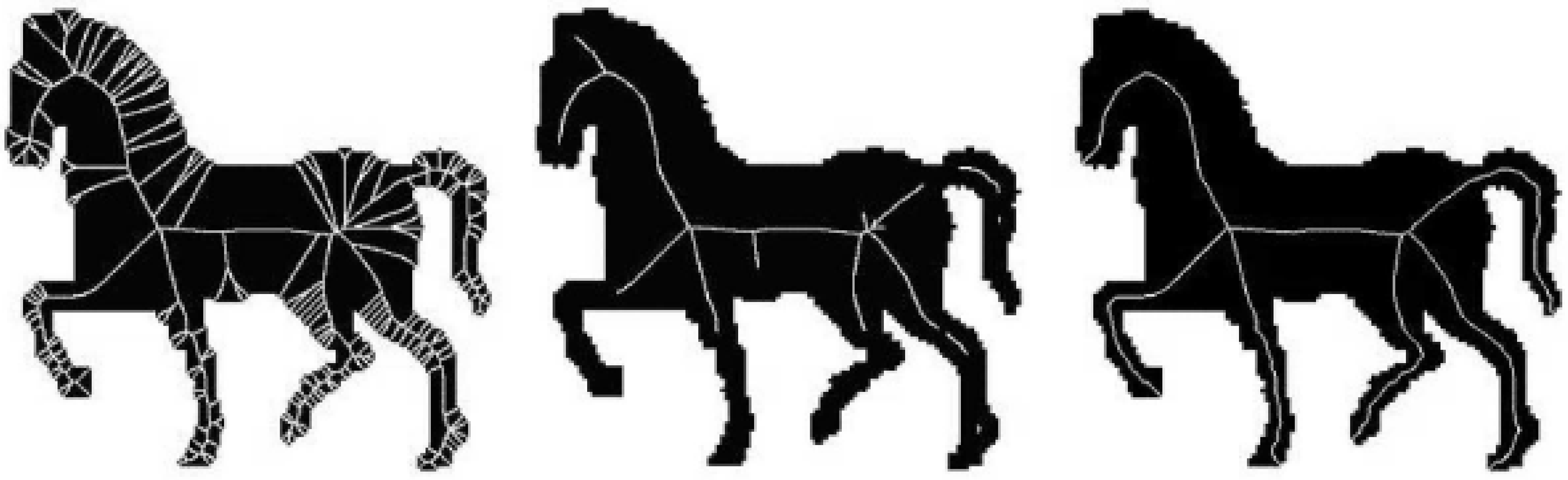
Discrete Skeleton Evolution iteratively removes the end branch that contributes least to the ability to reconstruct the original shape

The weight of a given end branch is:

$$w(i) = 1 - \frac{A\left(R\left(S - P\left(l_i, f(l_i)\right)\right)\right)}{A\left(R(S)\right)}$$

The Discrete Skeleton Evolution algorithm is as follows:

- do
  - Evaluate w(i) for all end branches in the curve
  - Remove the end branch P_{min} with lowest w
  - Form the new skeleton
- until the resulting skeleton is “satisfactory”
  - Satisfactory might mean:
    - “having no end branches with relevance less than some K_{min}”
    - other criteria related to reconstruction area or other metric



(a) (b) (c)

Fig. 1. The skeleton in (a) has many redundant branches. To remove them, usually skeleton pruning is applied. (b) Illustrates the problems of actual pruning approaches (it is generated by a method in [7]). In particular, observe that pruning may change the topology of the original skeleton. (c) Illustrates the pruning result of the proposed method that is guaranteed to preserve topology.

Bai and Latecki, *Discrete Skeleton Evolution*, 2007

Today’s Objectives

Curvature

- Curvature definition
- Discrete Curve Evolution (DCE)

Binary Morphology

- Erosion and Dilation
- Zhang-Suen skeletonization
- Discrete Skeleton Evolution (DSE)