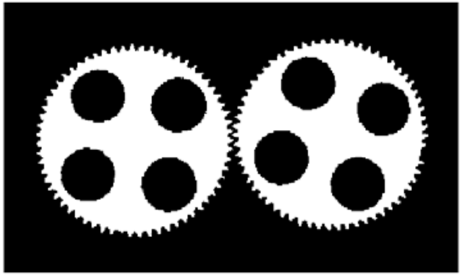
CS 4495 Computer Vision
Binary images and Morphology

Aaron Bobick
School of Interactive Computing



Administrivia

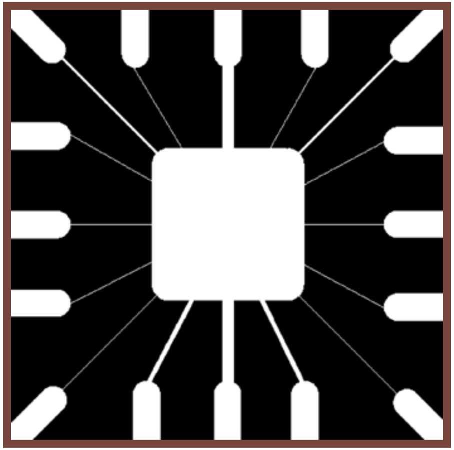
- PS7 – read about it on Piazza
- PS5 – grades are out.
- Final – Dec 9
    - Study guide will be out by Thursday hopefully sooner.

Binary Image Analysis

Operations that produce or process binary images, typically 0's and 1's

- 0 represents background
- 1 represents foreground

```
00010010001000
00011110001000
00010010001000
```

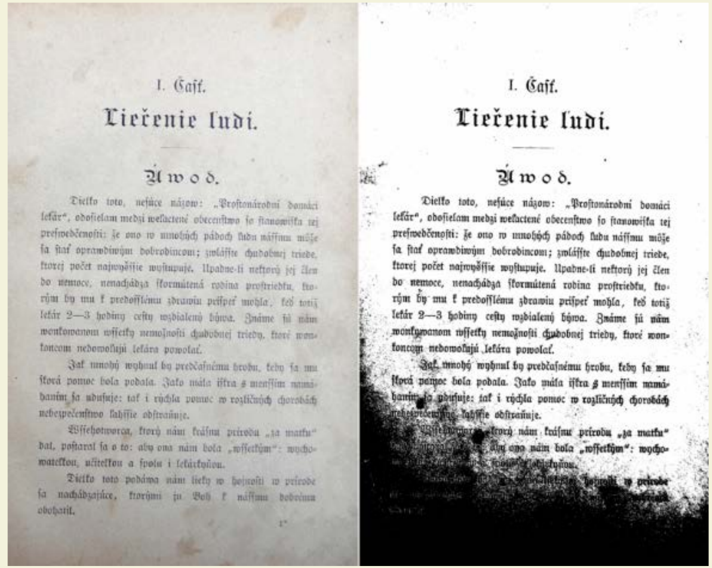


Slides: Linda Shapiro

Binary Image Analysis

Used in a number of practical applications

- Part inspection
- Manufacturing
- Document processing



I. Časť.

Liečenie ľudí.

Úwod.

Dielko toto, nesúce názow: „Prostonárodní domáci lekár“, odosielam medzi wełactené obecenstwo so stanowiska tej preswedčenosti: že ono w mnohých pádoch ľudu nášmu može sa stať oprawdiwým dobrodincom; zwlášte chudobnej triede, ktorej počet najwyšše wystupuje. Upadne-li nektorý jej člen do nemoce, nenachádza skormútená rodina prostriedku, ktorým by mu k predošlému zdrawiu prispeť mohla, keď totiž lekár 2—3 hodiny cesty wzdialený býwa. Známe sú nám wonkowanom wšetky nemožnosti chudobnej triedy, ktoré wonkoncom nedowoľujú lekára powolať.

Jak mnohý wyhnul by predčasnému hrobu, keby sa mu skorá pomoc bola podala. Jako málá iskra s menším namáhaním sa uduśuje: tak i rýchla pomoc w rozličných chorobách nebezpečenstwo ľahšie odstraňuje.

Wšehotworca, ktorý nám krásnu prírodu „za matku“ dal, postaral sa o to: aby ona nám bola „wšetkým“: wychowatelkou, učitelkou a spolu i lekárkyňou.

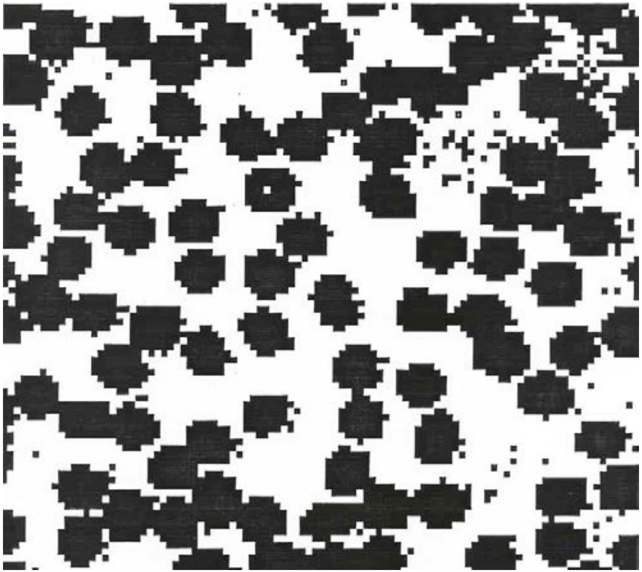
Dielko toto podáwa nám lieky w hojnosti w prírode sa nachádzajúce, ktorými ju Boh k nášmu dobrému obohatil.

1*

What kinds of operations?

- Separate objects from background and from one another
- Aggregate pixels for each object
- Compute features for each object

Example: Red blood cell image



- Many blood cells are separate objects
- Many touch – bad!
- Salt and pepper noise from thresholding
- How useable is this data?

Results of analysis

- 63 separate objects detected
- Single cells have area about 50
- Noise spots
- Gobs of cells

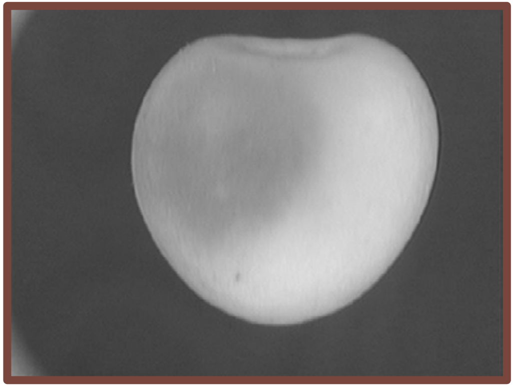
Object	Area	Centroid	Bounding Box	
1	383	(8.8 , 20)	[1 22 1 39]	
2	83	(5.8 , 50)	[1 11 42 55]	
3	11	(1.5 , 57)	[1 2 55 60]	
4	1	(1 , 62)	[1 1 62 62]	
5	1048	(19 , 75)	[1 40 35 100]	gobs
32	45	(43 , 32)	[40 46 28 35]	cell
33	11	(44 , 1e+02)	[41 47 98 100]	
34	52	(45 , 87)	[42 48 83 91]	cell
35	54	(48 , 53)	[44 52 49 57]	cell
60	44	(88 , 78)	[85 90 74 82]	
61	1	(85 , 94)	[85 85 94 94]	
62	8	(90 , 2.5)	[89 90 1 4]	
63	1	(90 , 6)	[90 90 6 6]	

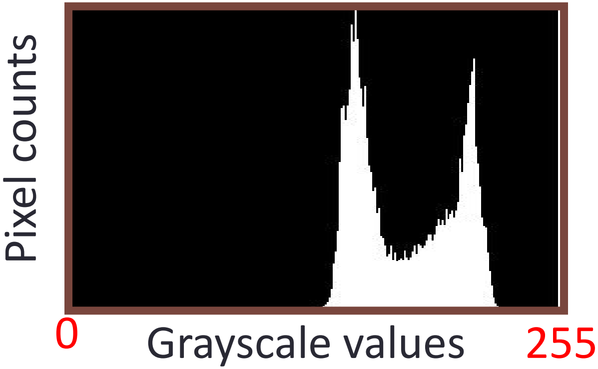
Useful Operations

- Thresholding a gray-scale image
- Determining good thresholds
- Connected components analysis
- Binary mathematical morphology
- All sorts of feature extractors, statistics (area, centroid, circularity, …)

Thresholding

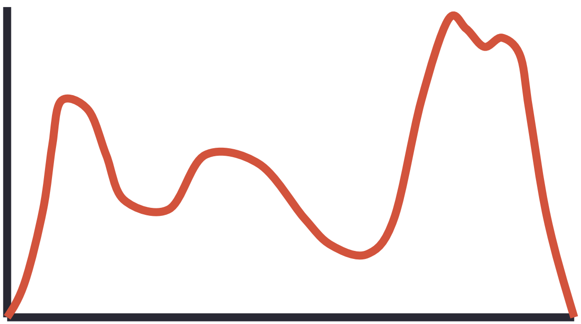
- Background is black
- Healthy cherry is bright
- Bruise is medium dark
- Histogram shows two cherry regions (black background has been removed)





Histogram-Directed Thresholding

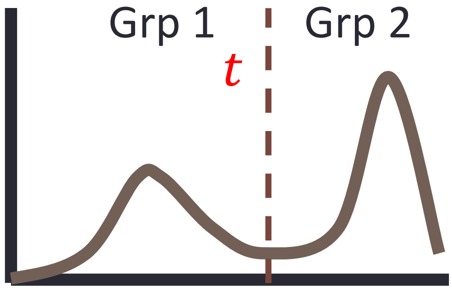
- How can we use a histogram to separate an image into 2 (or several) different regions?



Is there a single clear threshold? 2? 3?

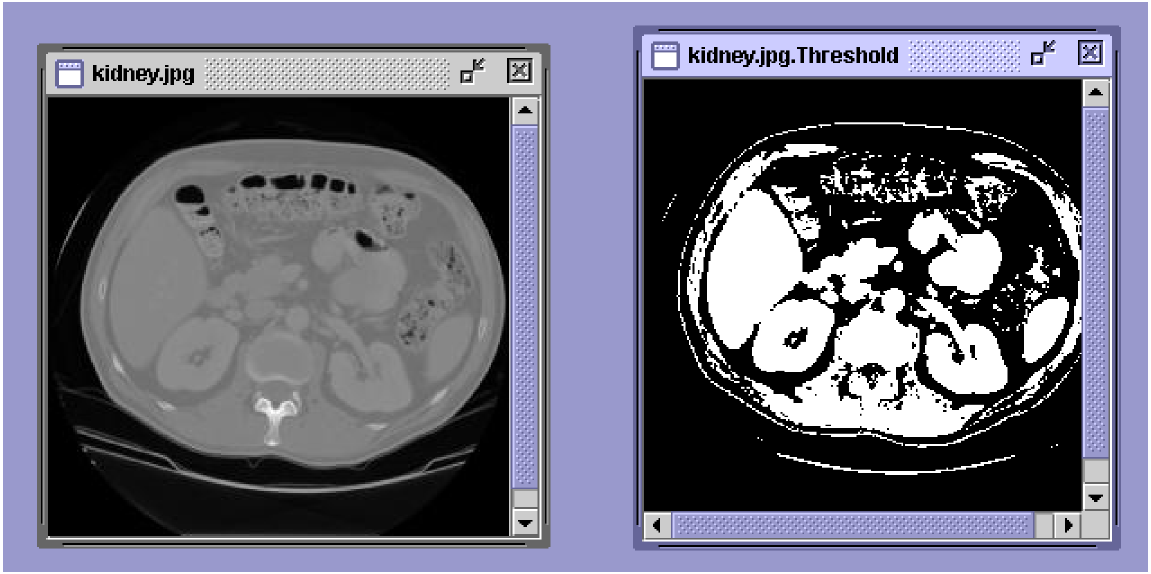
Automatic Thresholding: Otsu's Method

Assumption: The histogram is bimodal



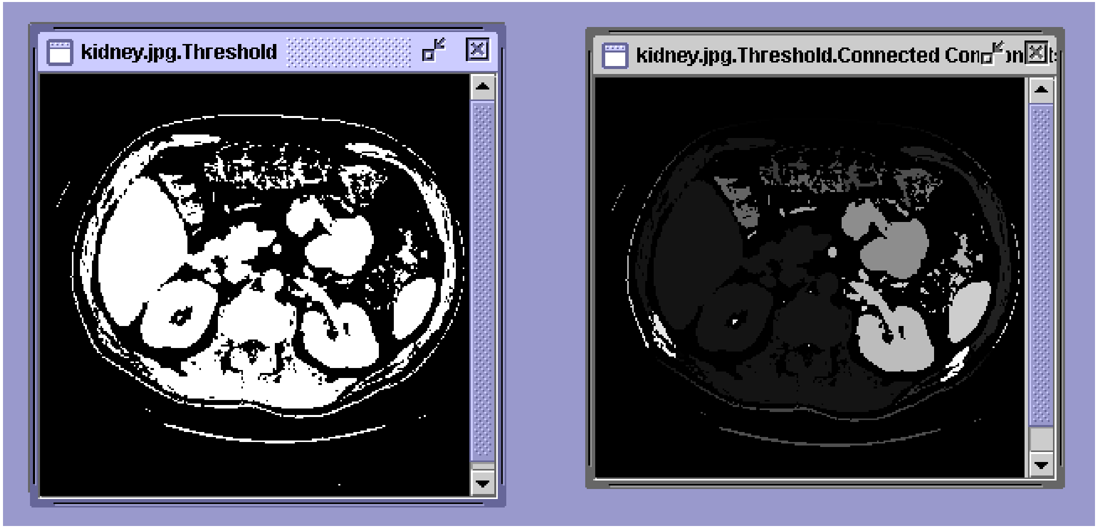
Method: Find the threshold t that minimizes the weighted sum of within-group variances for the two groups that result from separating the gray tones at value t

Thresholding Example



Connected Components Labeling

Once you have a binary image, you can identify and then analyze each connected set of pixels



Connected Components Methods

- Recursive Tracking (almost never used)
- Parallel Growing (needs parallel hardware)
- Row-by-Row (most common)
  - Classical Algorithm - 2 pass
  - Efficient Run-Length Algorithm (developed for speed in real industrial applications)

Equivalent Labels

Original Binary Image

0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1

Equivalent Labels

- CC = 0
- Scan across rows:
  - If 1 and connected:
    - Propgate lowest label behind or above (4 or 8 connected)
    - Remember conflicts
  - If 1 and not connected:
    - CC++ and label CC
  - If 0:
    - Label 0
- Re-label based on table

0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1

Equivalent Labels

- CC = 0
- Scan across rows:
  - If 1 and connected:
    - **Propgate lowest label behind or above (4 or 8 connected)**
    - Remember conflicts
  - **If 1 and not connected:**
    - **CC++ and label CC**
  - If 0:
    - Label 0
- Re-label based on table

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

Equivalent Labels

- CC = 0
- Scan across rows:
  - If 1 and connected:
    - Propgate lowest label behind or above (4 or 8 connected)
    - Remember conflicts
  - If 1 and not connected:
    - CC++ and label CC
  - If 0:
    - Label 0
- Re-label based on table

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

$1 \equiv 2$

$1 \equiv 3$

Equivalent Labels

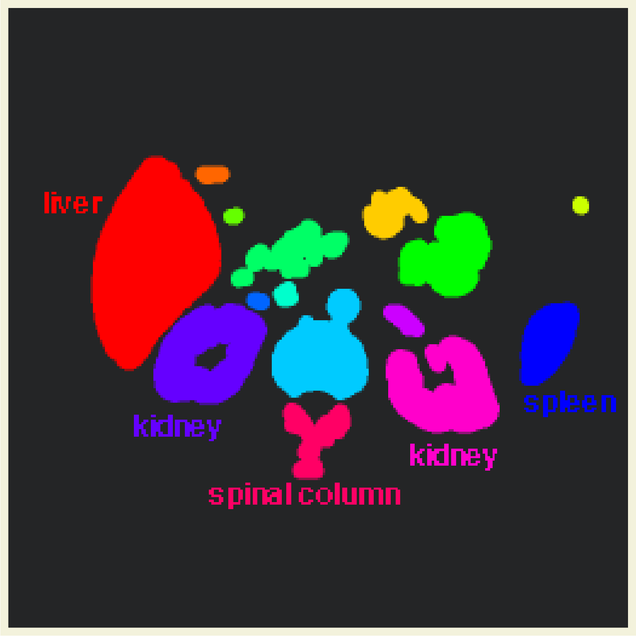
- CC = 0
- Scan across rows:
  - If 1 and connected:
    - Propgate lowest label behind or above (4 or 8 connected)
    - Remember conflicts
  - If 1 and not connected:
    - CC++ and label CC
  - If 0:
    - Label 0
- Re-label based on table

```
0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 1
0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 1 1 1
0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1
0 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1
```

$1 \equiv 2$

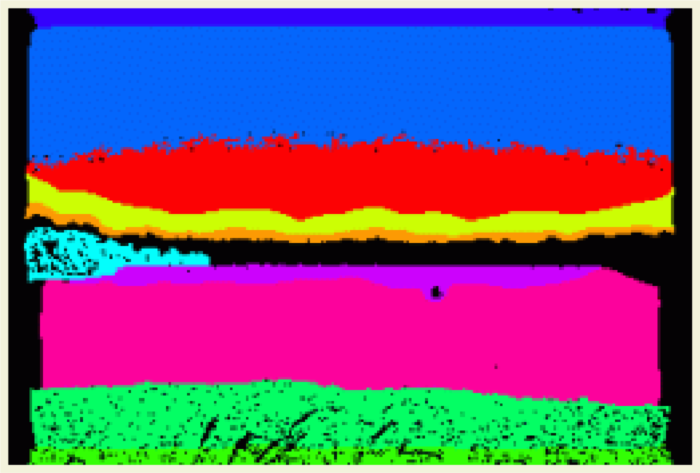
$1 \equiv 3$

Labeling shown as Pseudo-Color



Connected components of 1's from thresholded image

Labeling shown as Pseudo-Color



Connected components of cluster labels

Mathematical Morphology

Two basic operations

- Dilation
- Erosion

And several composite relations

- Closing and opening
- Thinning and thickening

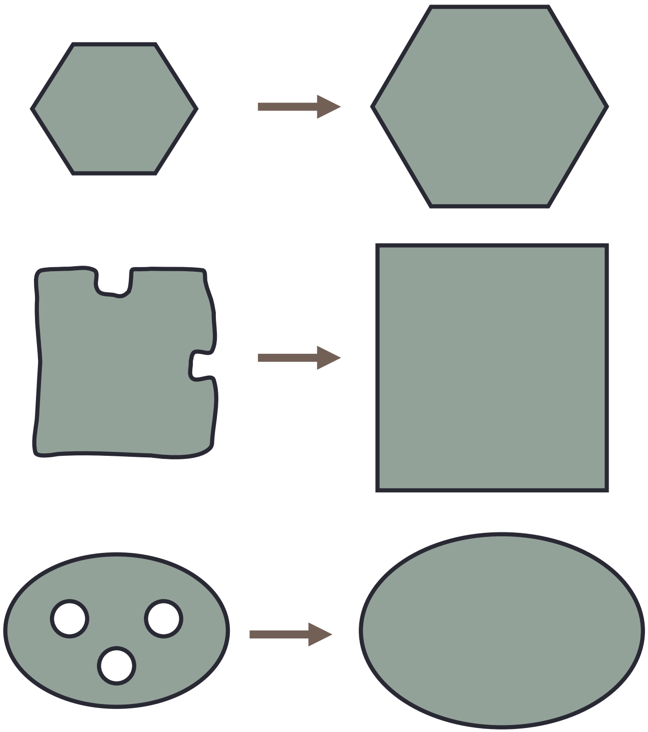
. . .

Dilation

Dilation expands the connected sets of 1s of a binary image.

It can be used for:

- Growing features
- Filling holes and gaps

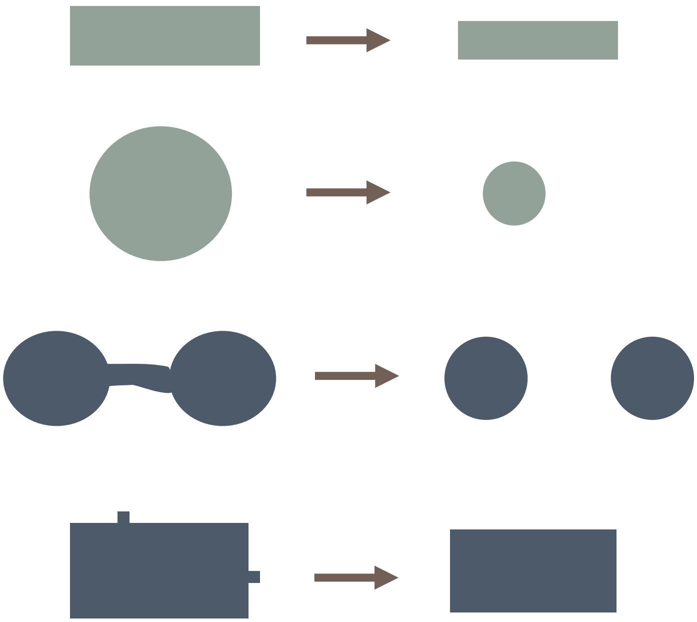


Erosion

Erosion shrinks the connected sets of 1s of a binary image.

It can be used for:

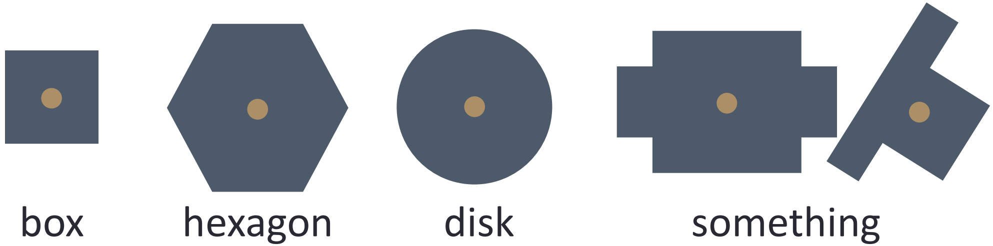
- Shrinking features
- Removing bridges, branches, protrusions



Structuring Element

A shape mask used in basic morphological ops.

- Any shape, size that is digitally representable
- With a defined origin

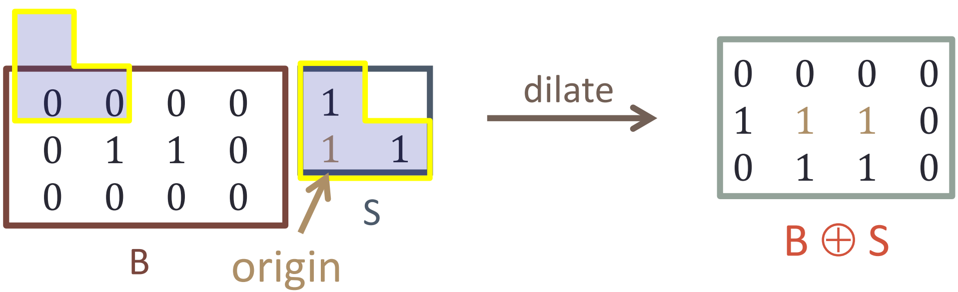


Box (length,width)   Disk (diameter)

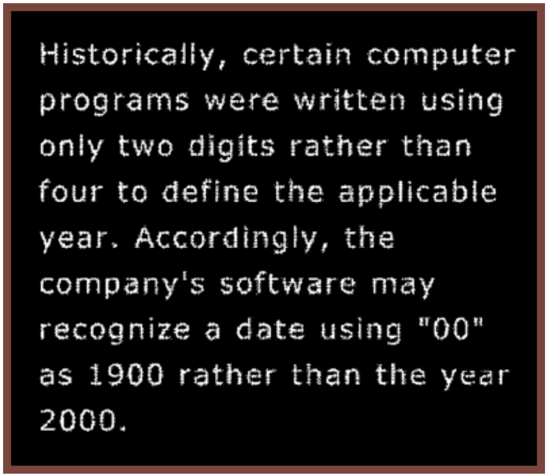
Dilation

Input: Binary image B, structuring element S

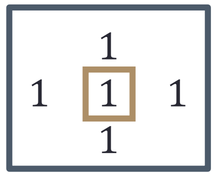
- Move S over B, placing origin at each pixel
- Considering only the 1-pixel locations in S, compute the binary *OR* of corresponding elements in B



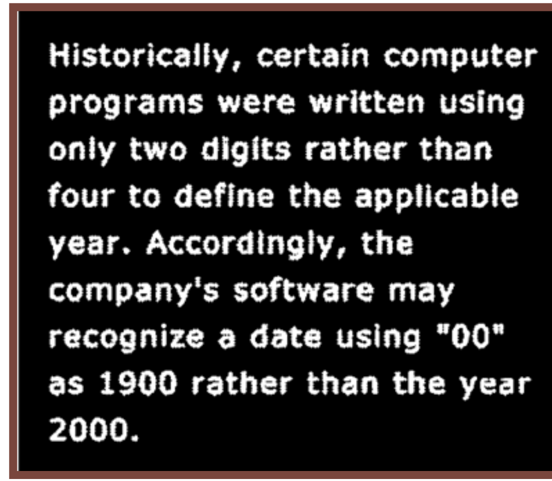
Binary text example


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Original



Structuring Element S


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Dilated by S

Erosion

Input: Binary image B, structuring element S

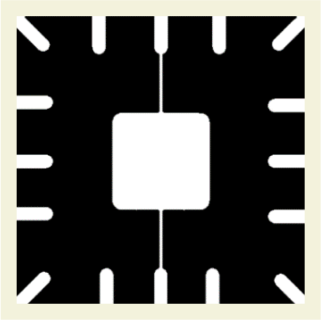
- Move S over B, placing origin at each pixel
- Considering only the 1-pixel locations in S, compute the binary *AND* of corresponding elements in B

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{B} \qquad \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{S} \xrightarrow{\text{erode}} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{B} \ominus \mathbf{S}}$$

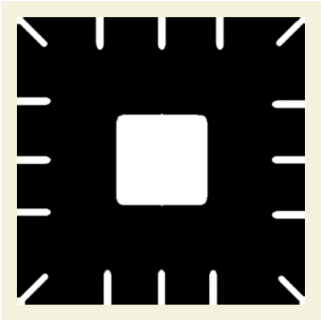
origin

Effect of disk size on erosion

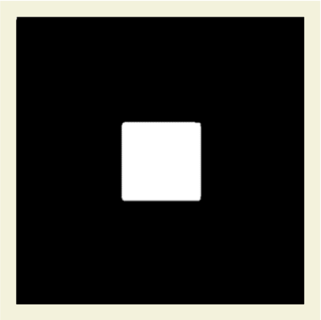
Original image



Erosion with a disk of radius 5



Radius 10



Radius 20

Slide: Ioannis Ivrissimtzis

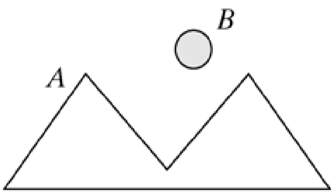
Opening and Closing

- The two most useful binary morphology operations are Opening and Closing.

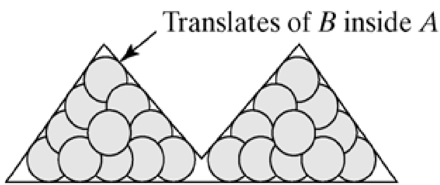
Opening

- Opening is the compound operation of erosion followed by dilation (with the same structuring element)
  - Can show that the opening of A by B is the union of all translations of B that fit entirely within A.

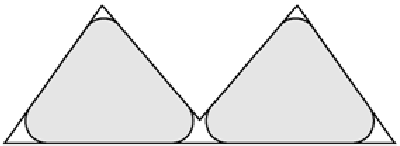
Opening



Binary image A and structuring element B



Translations of B that fit entirely within A



The opening of A by B is shown shaded

Intuitively, the opening is the area we can paint when the brush has a footprint B and we are not allowed to paint outside A.

Slide: Ioannis Ivrissimtzis

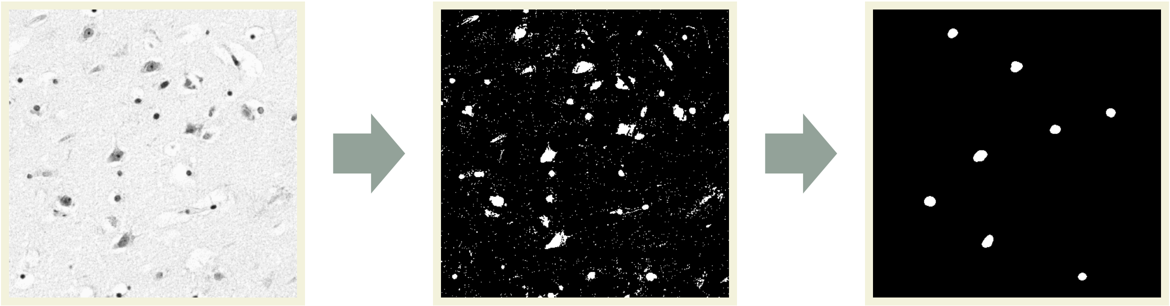
Opening

- Opening is the compound operation of erosion followed by dilation (with the same structuring element)
  - Can show that the opening of A by B is the union of all translations of B that fit entirely within A.
  - Opening is **idempotent:** Repeated operations has no further effects!

Opening example – cell colony

Use large structuring element that fits into big objects

- Structuring Element: 11 pixel disc

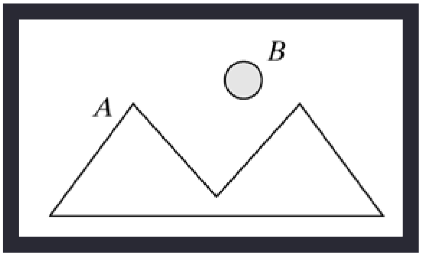


Slide: Thomas Moeslund

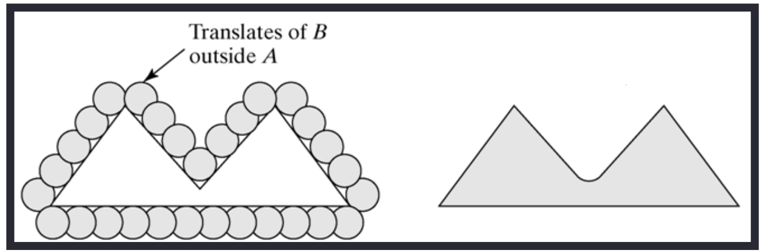
Closing

- Closing is the compound operation of dilation followed by erosion (with the same structuring element)
  - Can show that the closing of A by B is the complement of union of all translations of B that do not overlap A.

Closing



Binary image A and structuring element B



Translations of B that do not overlap A

The closing of A by B is shown shaded

Intuitively, the closing is the area we can not paint when the brush has a footprint B and we are not allowed to paint inside A.

Slide: Ioannis Ivrissimtzis

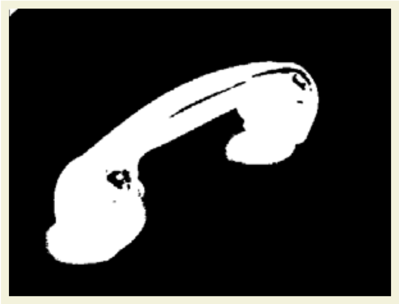
Closing

- Closing is the compound operation of dilation followed by erosion (with the same structuring element)
  - Can show that the closing of A by B is the complement of union of all translations of B that do not overlap A.
  - Closing is **idempotent**: Repeated operations has no further effects!

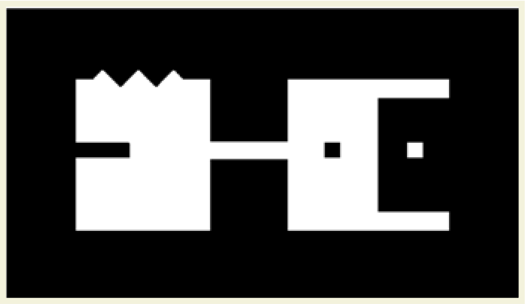
Closing Example - Segmentation

Simple segmentation:

1. Threshold
2. Closing with disc of size 20


Slide: Thomas Moeslund



Original image

Opening

Closing

Opening followed by closing

Real example – Fingerprint analysis

Original image | Opening | Opening following by closing

Slide: Ioannis Ivrissimtzis

Some Basic Morphological Algorithms

- $\boxed{\text{Boundary extraction}}$
- Region filling
- Extraction of connected components
- Convex Hull
- $\boxed{\text{Thinning}}$
- Skeletons
- Pruning

Boundary extraction

Let $A \oplus B$ denote the dilation of A by B and let $A \ominus B$ denote the erosion of A by B.

The boundary of A can be computed as:

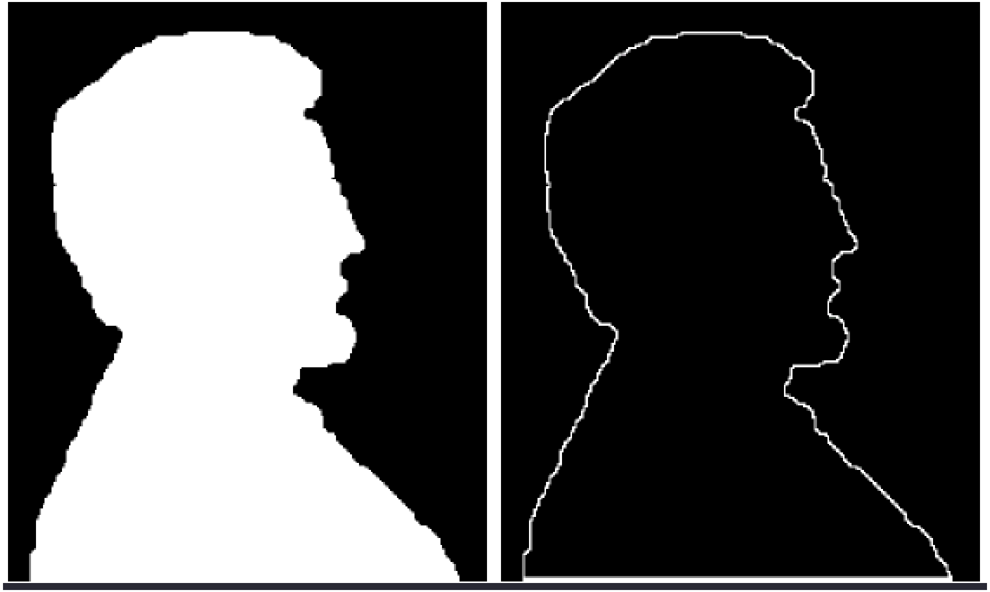
$$A - (A \ominus B)$$

where B is a 3x3 square structuring element.

That is, we subtract from A an erosion of it to obtain its boundary.

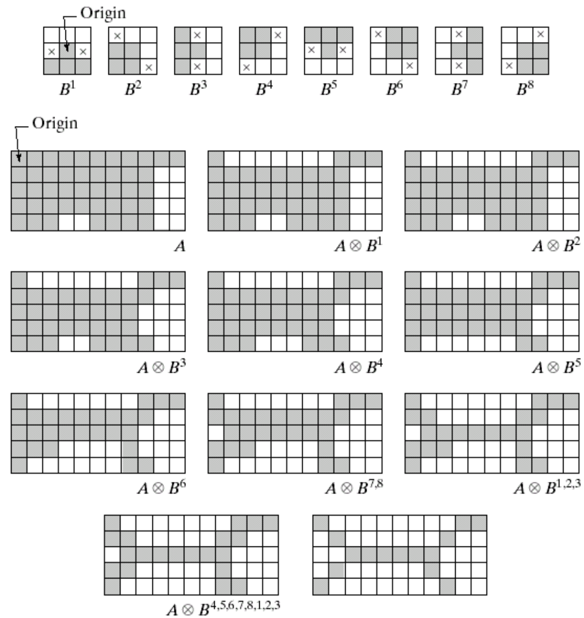
Slide: Ioannis Ivrissimtzis

Example of boundary extraction



Thinning

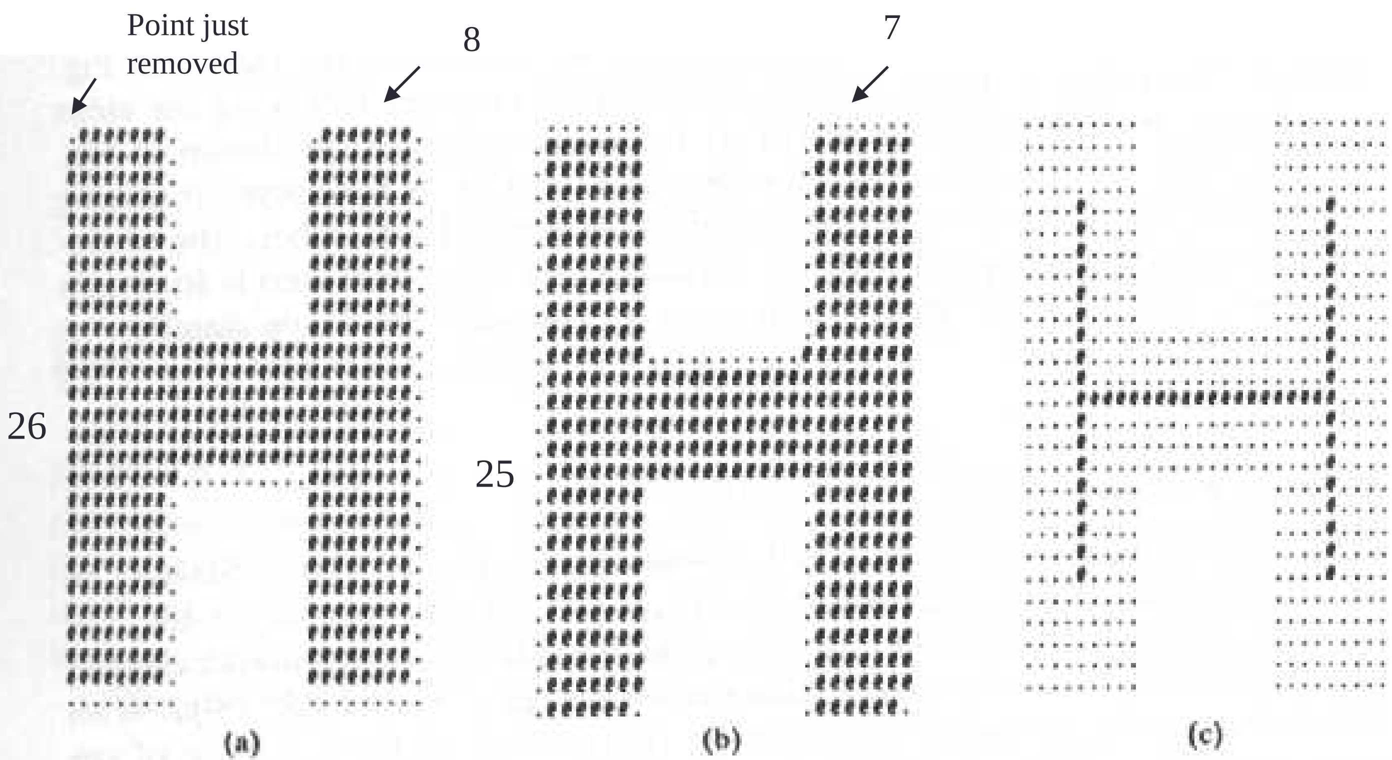
$$
\begin{aligned}
A \otimes B &= A - (A \circledast B) \\
&= A \cap (A \circledast B)^c
\end{aligned}
$$



a
b c d
e f g
h i j
k l

FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m-connectivity.

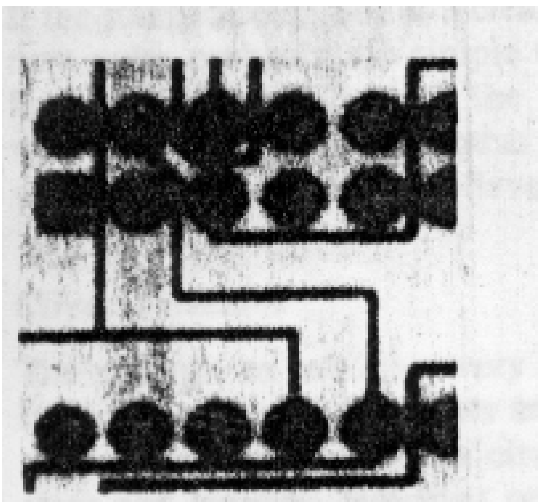
Thinning example

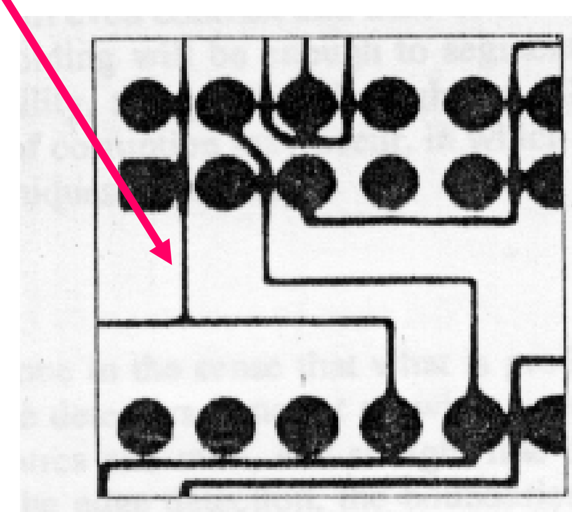


results of the first pass **results of the second pass** **final results**

Semi-real thinning

All lines are thinned to one pixel width
Now you can check connectivity

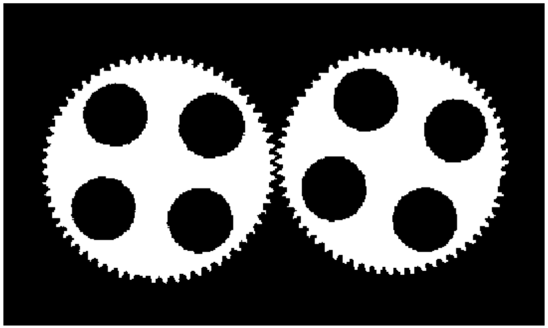




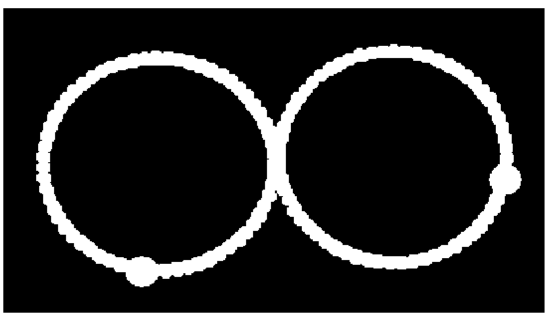
How powerful is morphology

- It depends…
- If almost “clean” binary images then very powerful to both clean up images and to detect variations from desired image.
- Example…

Gear Tooth Inspection



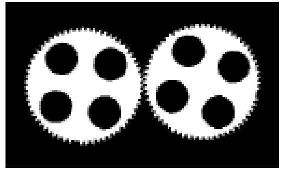
original binary image



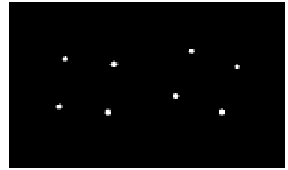
detected defects

How did they do it?

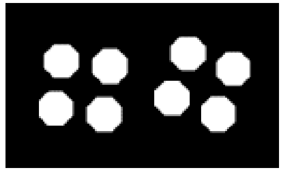
Slide by Linda Shapiro



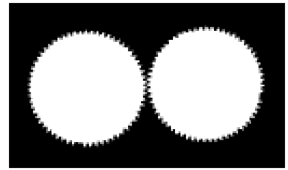
a) original image B



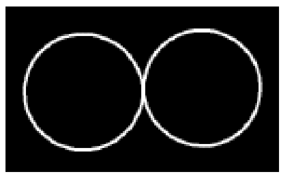
b) $B1 = B \ominus$ hole_ring



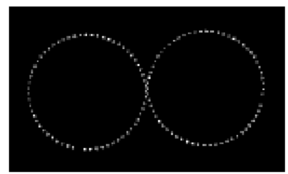
c) $B2 = B1 \oplus$ hole_mask



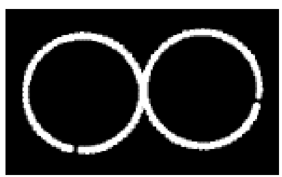
d) B3 = B OR B2



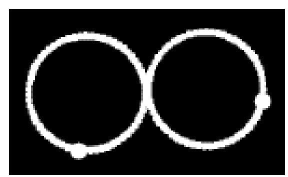
e) B7 (see text)



f) B8 = B AND B7



g) $B9 = B8 \oplus$ tip_spacing



h) RESULT = $((B7 - B9) \oplus$ defect_cue) OR B9

Geometric and Shape Properties

- area
- centroid
- perimeter
- perimeter length
- circularity
- elongation
- mean and standard deviation of radial distance
- bounding box
- extremal axis length from bounding box
- second order moments (row, column, mixed)
- lengths and orientations of axes of best-fit ellipse

Slide by Linda Shapiro

Morphology today

- Binary morphology still has a science behind it – it’s almost like advanced algebra because it’s all about set operations and mathematical relations.
- It doesn’t appear much in research in computer vision but we all do it. Almost all the time. It’s simply critical to making image analysis work.
- Not sexy, but very important.