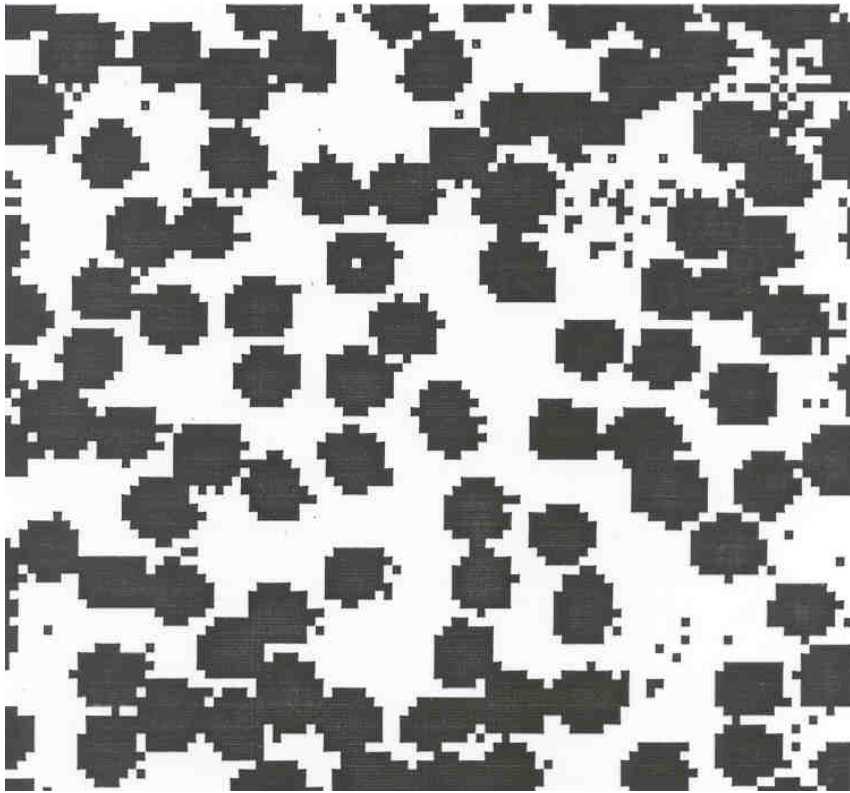


Binary Image Analysis

- used in a variety of applications:
 - part inspection
 - riveting
 - fish counting
 - document processing
- consists of a set of image analysis operations that are used to produce or process binary images, usually images of 0's and 1's.

```
00010010001000  
00011110001000  
00010010001000
```

Example: red blood cell image



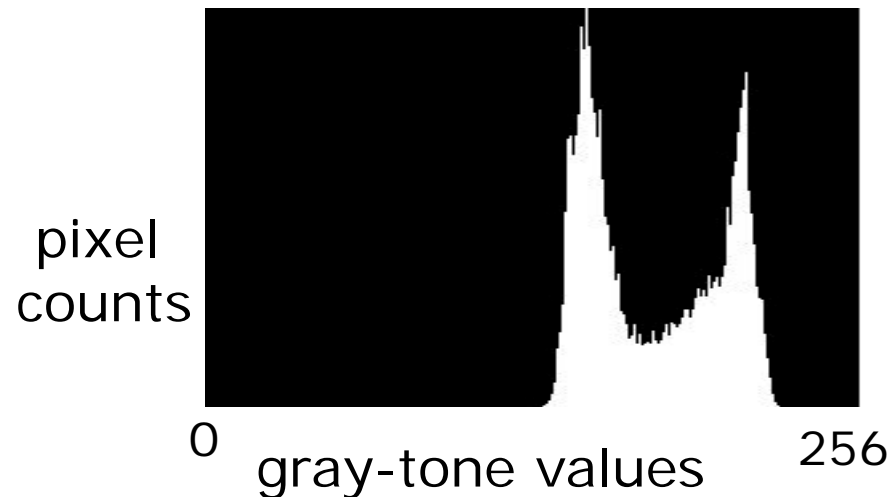
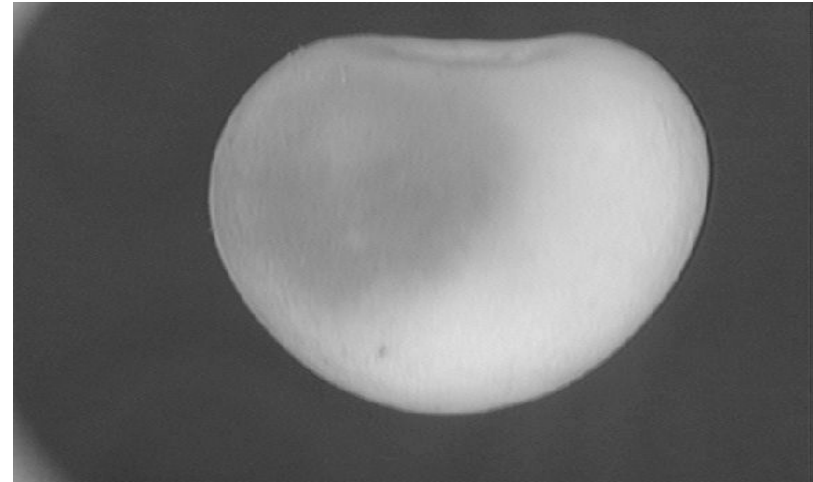
- Many blood cells are separate objects
- Many touch – bad!
- Salt and pepper noise from thresholding
- What operations are needed to clean it up?

Useful Operations

1. **Thresholding a gray-tone image**
2. **Determining good thresholds**
3. **Filtering with mathematical morphology**
4. **Connected components analysis**
5. **Numeric feature extraction**
 - **location features**
 - **gray-tone features**
 - **shape features ...**

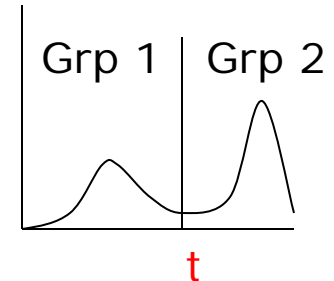
Thresholding

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



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 .

Works well **if** the assumption holds.

Thresholding Example



original image



pixels above threshold

Mathematical Morphology

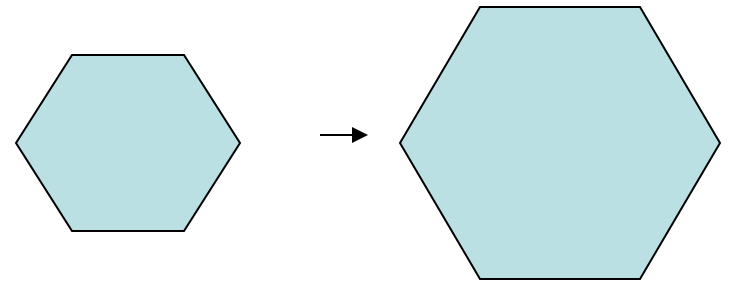
(Dilation, Erosion, Closing, Opening)

- **Dilation**

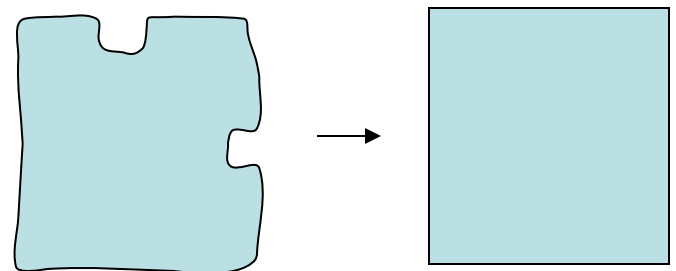
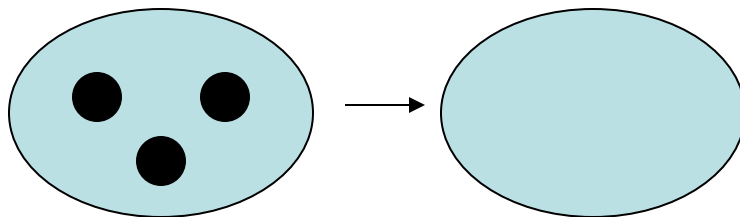
Dilation **expands** the connected sets of 1s of a binary image.

It can be used for

1. growing features



2. filling holes and gaps

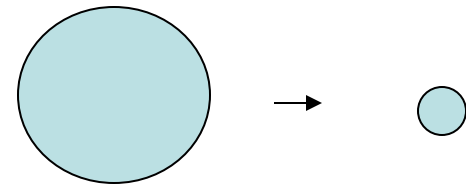
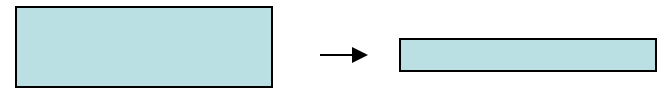


• Erosion

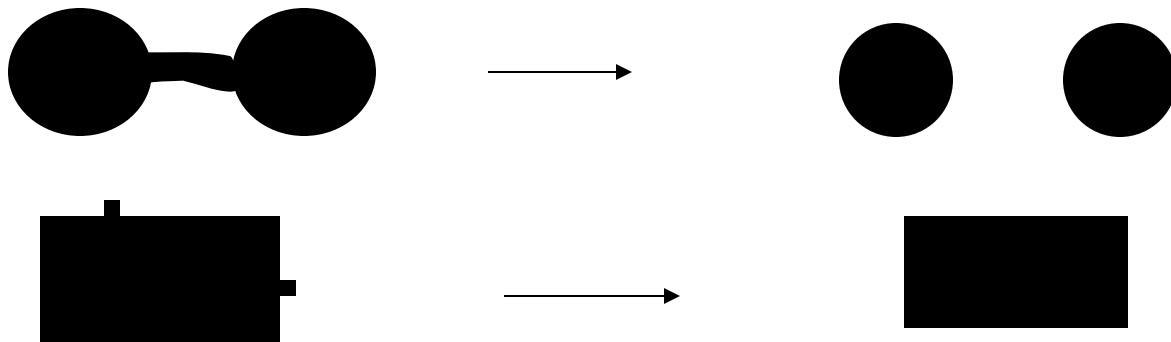
Erosion **shrinks** the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



2. Removing bridges, branches and small protrusions



Structuring Elements

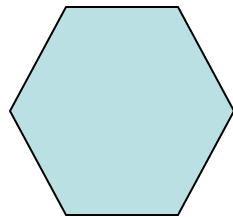
A **structuring element** is a shape mask used in the basic morphological operations.

They can be any shape and size that is **digitally representable**, and each has an **origin**.

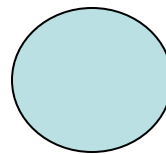


box

box(length,width)

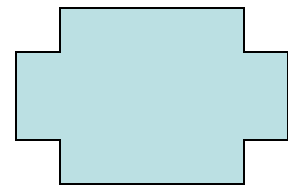


hexagon



disk

disk(diameter)



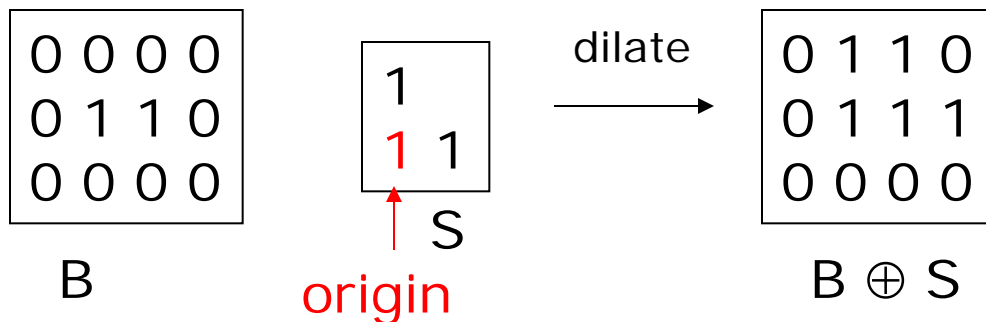
something

Dilation with Structuring Elements

The arguments to dilation and erosion are

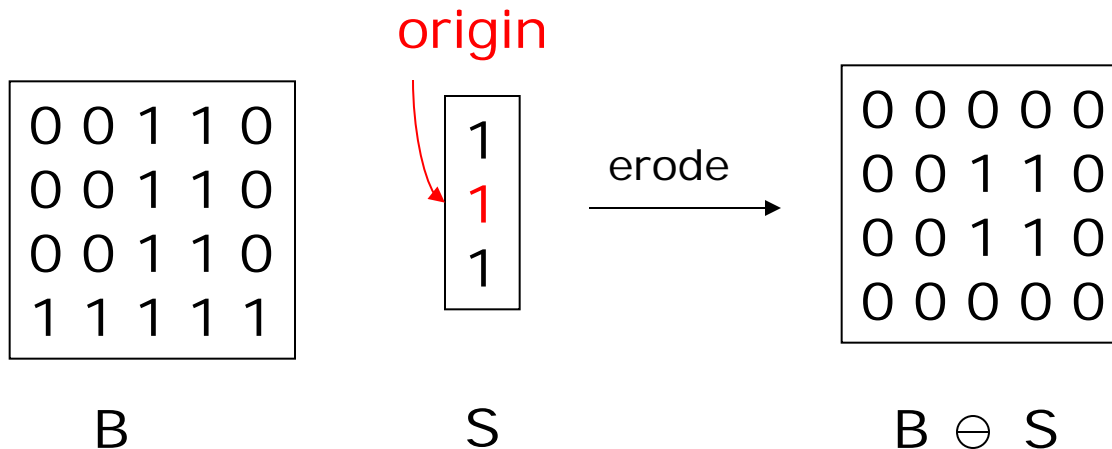
1. a binary image **B**
2. a structuring element **S**

`dilate(B,S)` takes binary image **B**, places the origin of structuring element **S** over each 1-pixel, and ORs the structuring element **S** into the output image at the corresponding position.



Erosion with Structuring Elements

$\text{erode}(B, S)$ takes a binary image B , places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B .



Opening and Closing

- **Closing** is the compound operation of dilation followed by erosion (with the same structuring element)
- **Opening** is the compound operation of erosion followed by dilation (with the same structuring element)

| | | | | | | | |
|---|---|---|---|---|---|---|--|
| | | | | | | | |
| 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 | | | | |
| | | | | | | | |

a) Binary image B

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

b) Structuring Element S

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 | 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 | | | |

c) Dilation $B \oplus S$

| | | | | | | | |
|--|--|--|---|---|--|--|--|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | 1 | 1 | | | |
| | | | 1 | 1 | | | |
| | | | 1 | 1 | | | |
| | | | | | | | |
| | | | | | | | |

d) Erosion $B \ominus S$

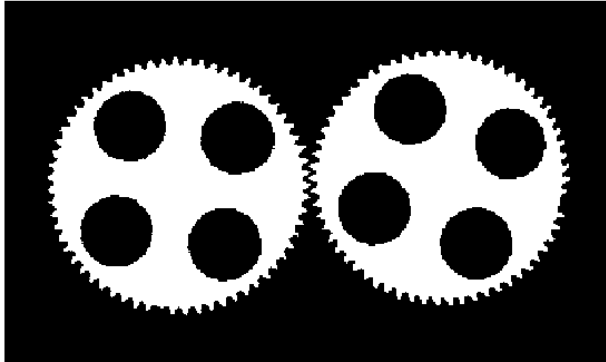
| | | | | | | | |
|--|---|---|---|---|---|---|--|
| | | | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | |
| | | 1 | 1 | 1 | 1 | 1 | |
| | | 1 | 1 | 1 | 1 | 1 | |
| | | 1 | 1 | 1 | 1 | 1 | |
| | | 1 | 1 | 1 | 1 | 1 | |
| | | 1 | 1 | | | | |
| | | | | | | | |

e) Closing $B \bullet S$

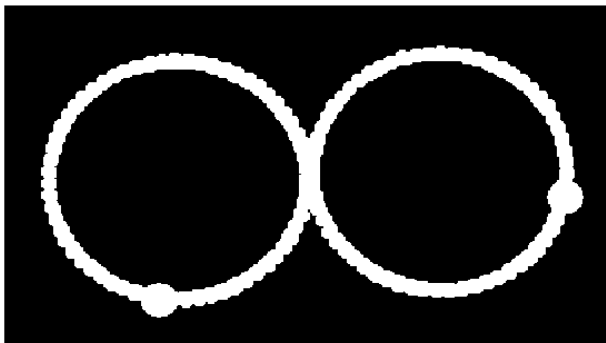
| | | | | | | | |
|--|--|--|---|---|---|---|--|
| | | | | | | | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | | | | | |
| | | | | | | | |

f) Opening $B \circ S$

Application: Gear Tooth Inspection



original
binary
image



detected
defects

Connected Components Labeling

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

The connected components operation takes in a binary image and produces a **labeled image** in which each pixel has the integer label of either the background (0) or a component.

original



thresholded



opening+closing



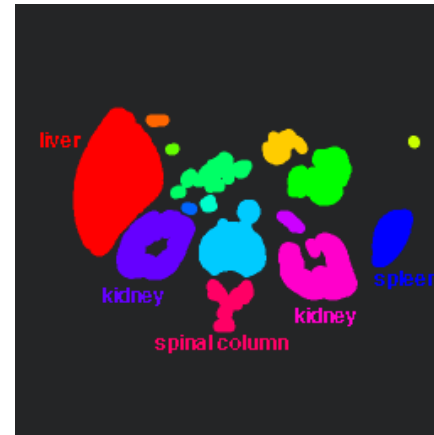
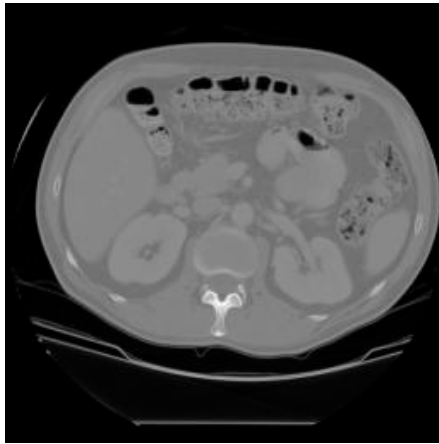
components



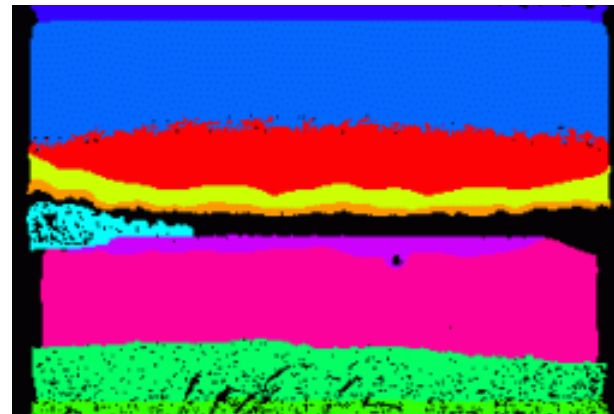
Methods for CC Analysis

1. Recursive Tracking (almost never used)
2. Parallel Growing (needs parallel hardware)
3. Row-by-Row (most common)
 - a. propagate labels down to the bottom, recording equivalences
 - b. Compute equivalence classes
 - c. Replace each labeled pixel with the label of its equivalence class.

Labelings shown as Pseudo-Color



connected components of 1's from cleaned, thresholded image



connected components of cluster labels