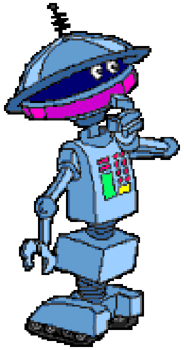
Electrical/Computer Engineering Design Project



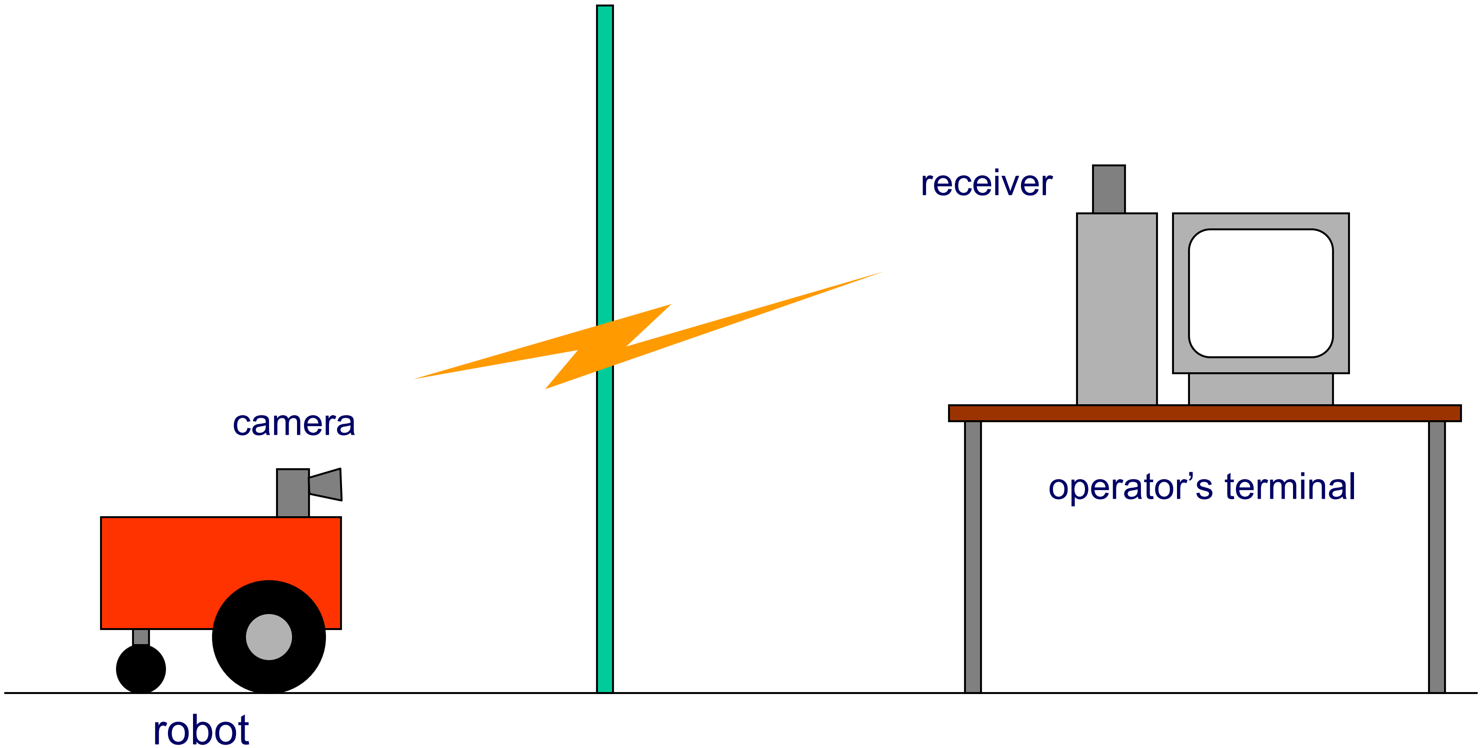
Fall 2002

Lecture 3 – Computer Vision

Topics

- Computer vision system
- Image enhancement
- Image analysis
- Pattern Classification

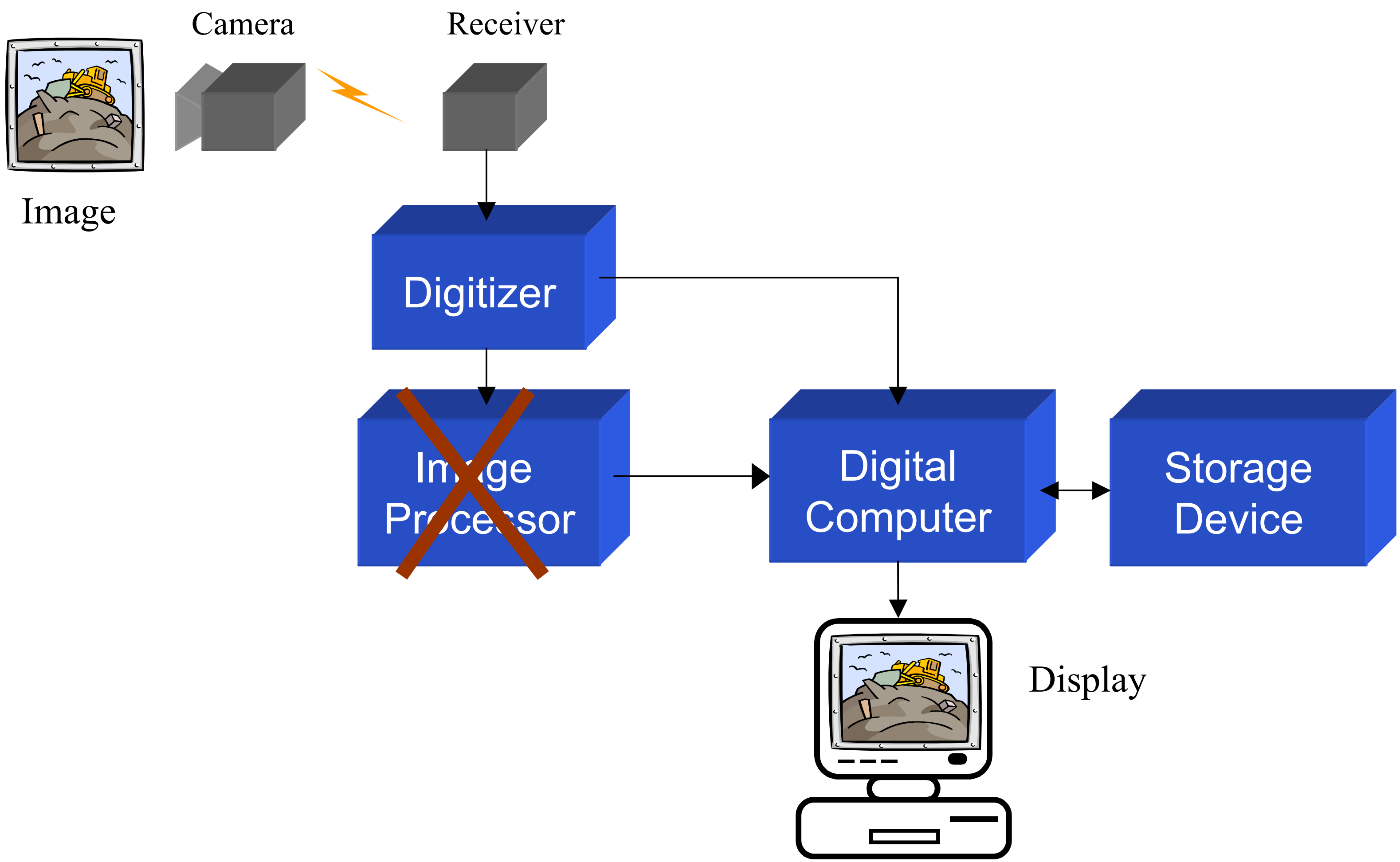
Image Processing System



Dual purpose:

1. Provide feedback for operator
2. Recognize victims (must be automated)

Image Processing System



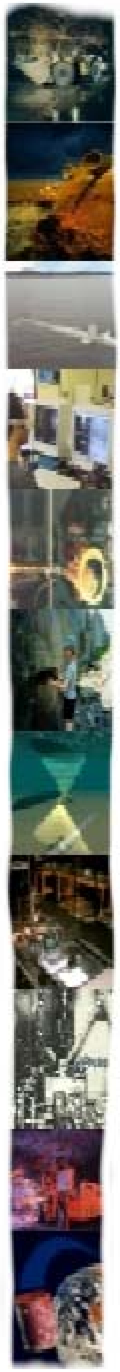


Image representation

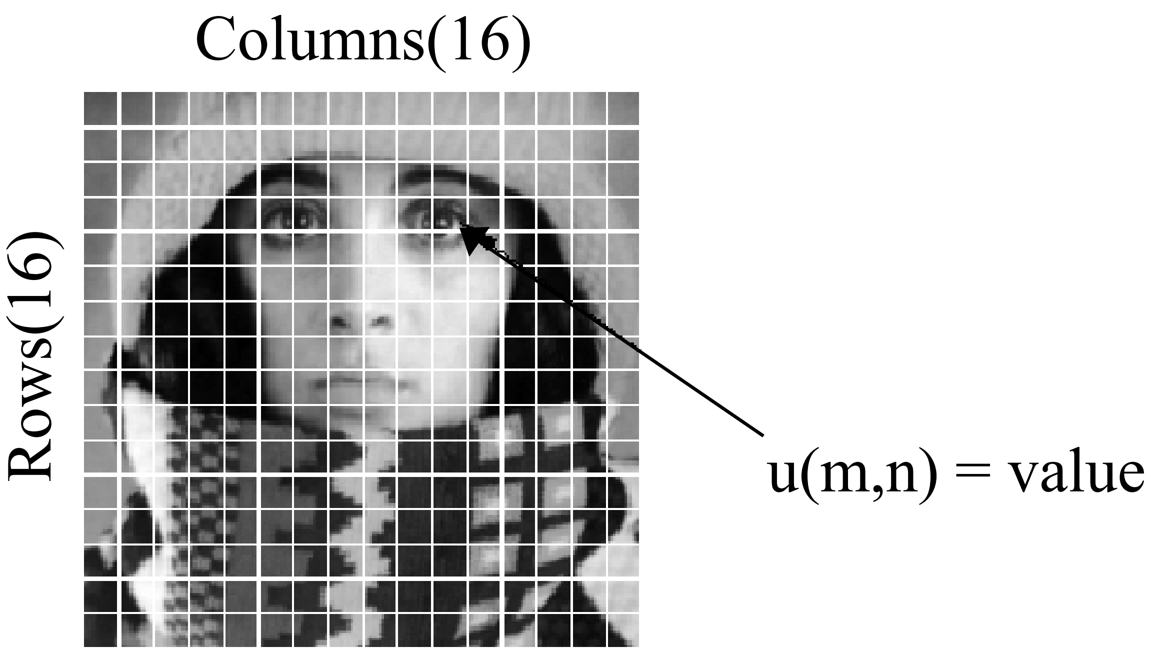
- Images are usually represented by a 2-D intensity function f(x,y) where
  - x and y represent spatial coordinates
  - The value of *f* is proportional to the brightness (gray level) of the image.

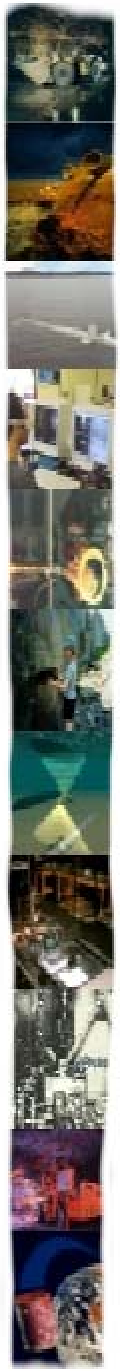
- A digital image u(m,n) is represented by a matrix whose rows and columns identify a point in an image and matrix element value identifies the gray level at that point.

- Each point is referred to as a picture element or “pixel”

Image representation

- For computer storage, an array with the number of gray levels being a power of 2 is selected.
- A typical gray level image contains 256 shades of gray (8 bit)
- Values are stored between 0 - 255





Objective of Computer Vision

- The field of computer vision can be divided into two areas
  - Image enhancement
  - Image analysis

Image enhancement

- Refers to the accentuation or sharpening of image features such as edges, boundaries, or contrast to make an image more useful for display and analysis.

Image Enhancement

Point Operations

- Contrast stretching
- Noise clipping
- Window Slicing
- Histogram modeling

Spatial Operations

- Noise smoothing
- Median Filtering
- Unsharp masking
- Low-pass, band-pass, high-pass filtering

Transform Operations

- Linear filtering
- Root filtering
- Homomorphic filtering

Point Operations

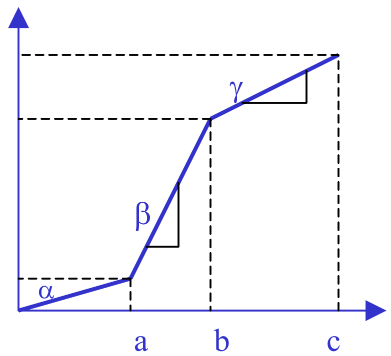
- Mapping a given gray level u $\in [0, L]$ into a gray level $v \in [0, L]$ according to $v = f(u)$

<u>Contrast Stretching</u>

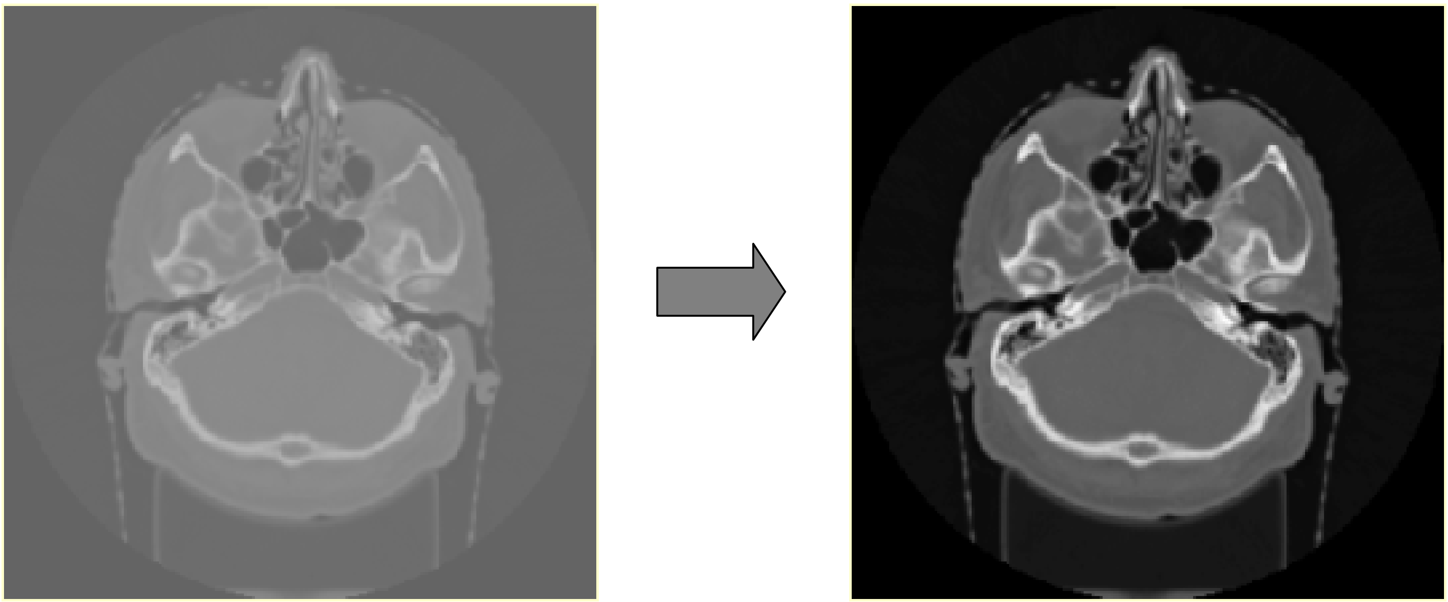
For low-contrast images that occur because of:

- Poor or non-uniform lighting
- Non-linearity or small dynamic range of sensors.

$$v = \begin{cases} \alpha u & 0 \le u \le a \\ \beta(u - a) & a \le u \le b \\ \gamma(u - b) & b \le u \le L \end{cases}$$

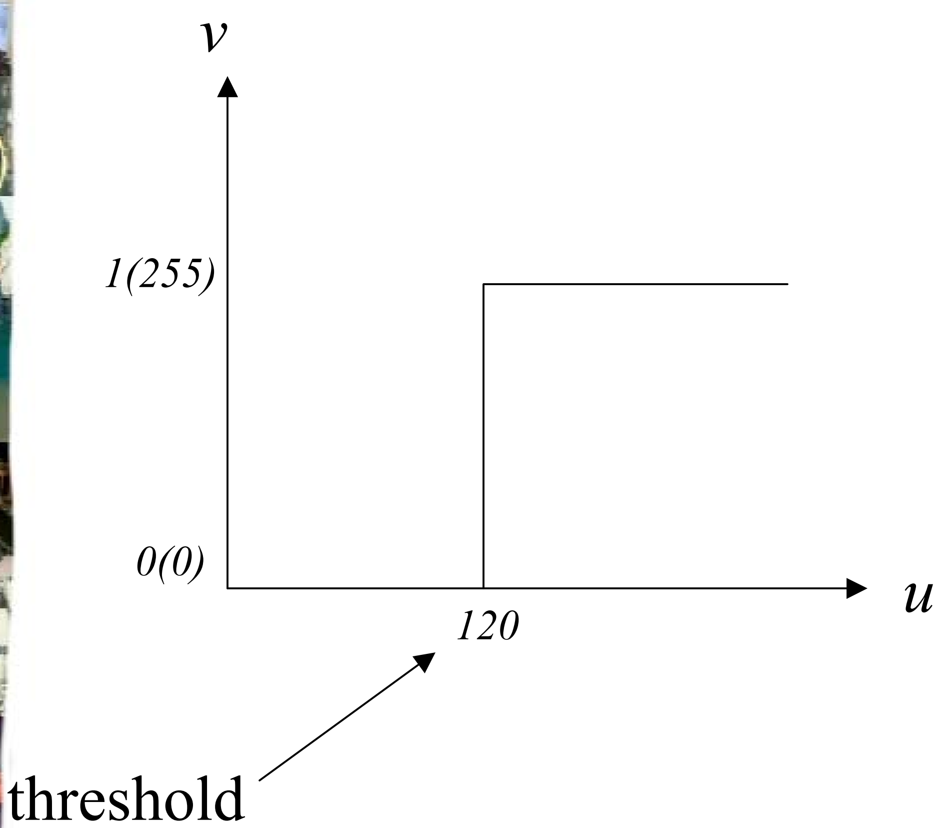


Contrast Stretching



Thresholding

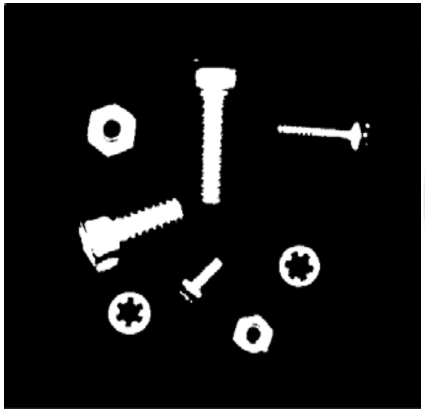
Output is a binary image.



Example:

$u(m,n) =$

$v(m,n) =$



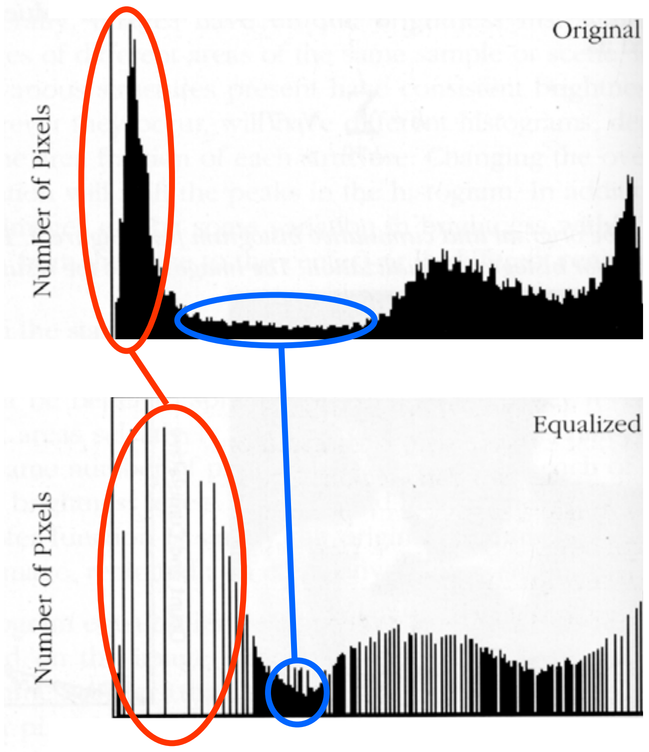
Histogram Equalization

The effect of performing histogram equalization on an image is to produce a uniform distribution of gray levels in an image.

The equalization process spreads out the peaks in an image and while compressing the range of intensities in regions of the histogram that have relatively few pixels.

The histogram represents the relative frequency of occurrence of the various gray levels in an image.

Equalization provides a way to enhance minor intensity variations in an apparently uniform image - method to emphasize dim features.



Histogram Equalization Examples

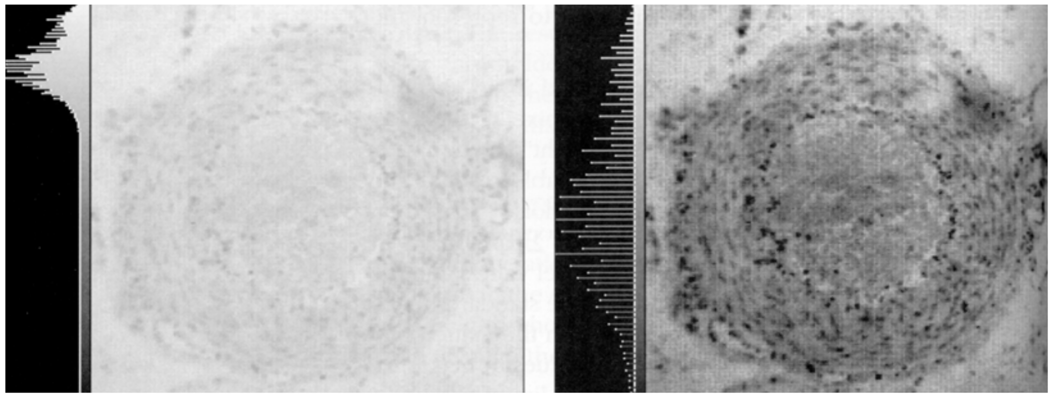
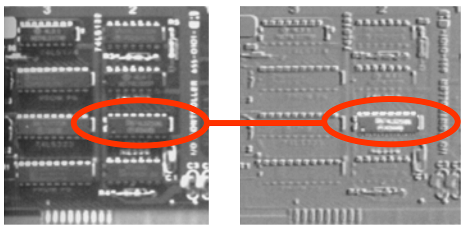
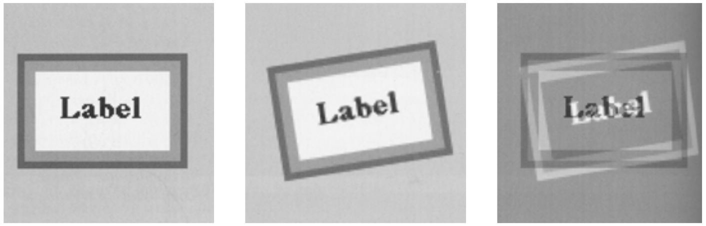
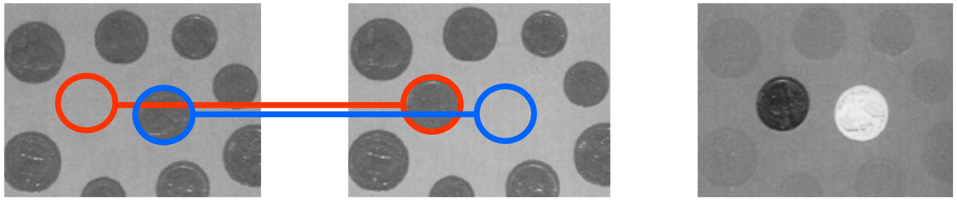


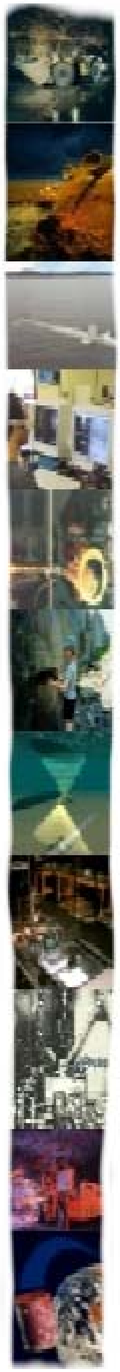
Image Math

Image math operations (addition, subtraction, multiplication and division) use two images to create a third image using pixel by pixel application of the mathematical operator.



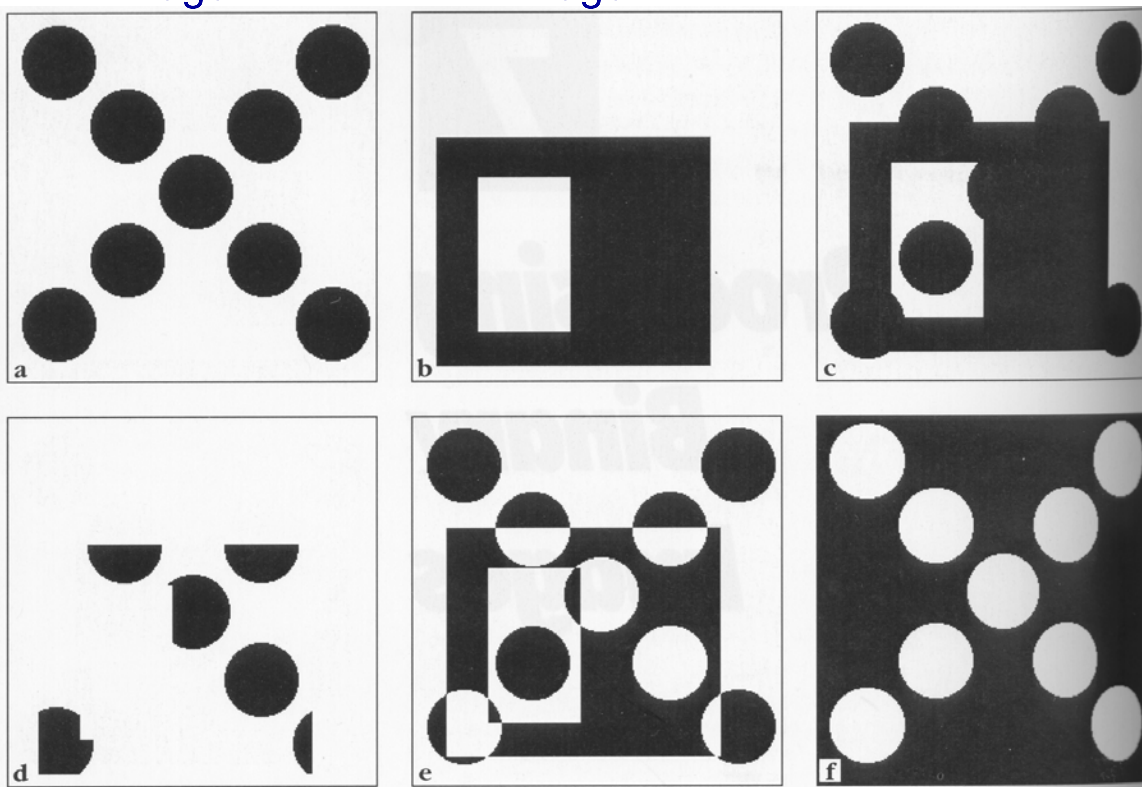






Boolean Operations

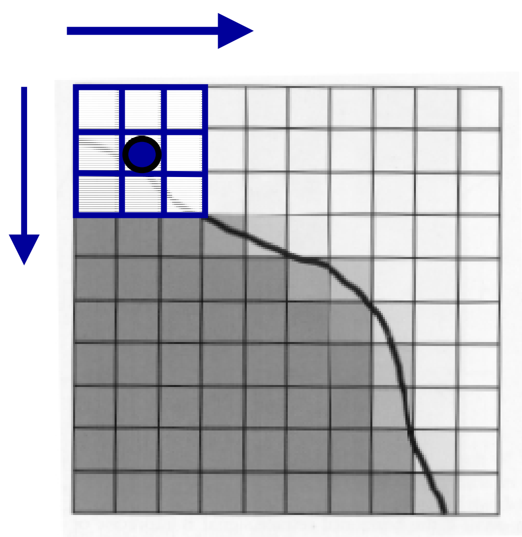
Image A   Image B   A OR B

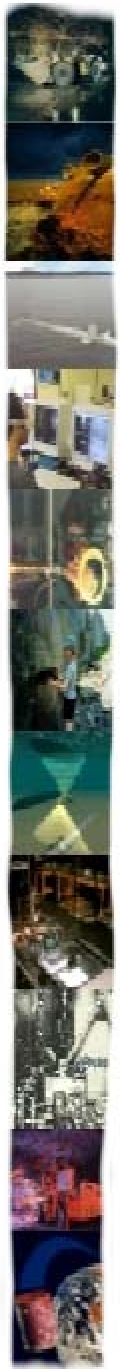


A AND B   A XOR B   NOT A

Spatial Operations

Spatial filtering involves passing a weighted mask, or 'kernel', over the noisy image and replacing the original image pixel value corresponding to the centre of the kernel with the sum of the original pixel values in the region corresponding to the kernel multiplied by the kernel weights.





Spatial Operations

- Performed on local neighborhoods of input pixels. Image is convolved with a finite response filter called a spatial mask (window).

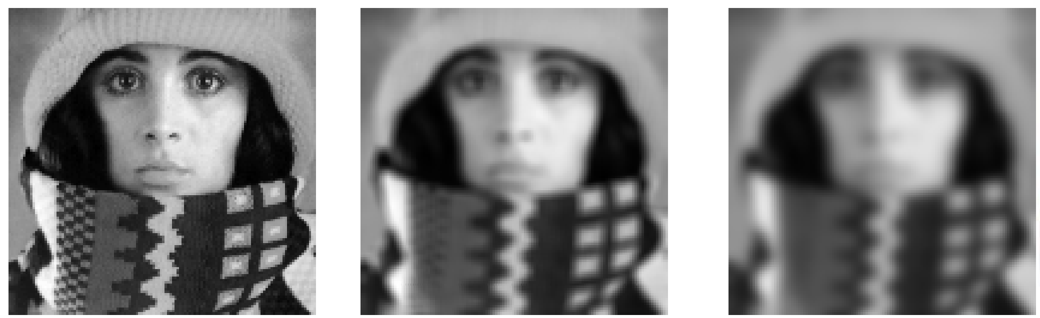
<u>Spatial average</u>

- each pixel is replaced by a weighted average of its neighborhood pixels.

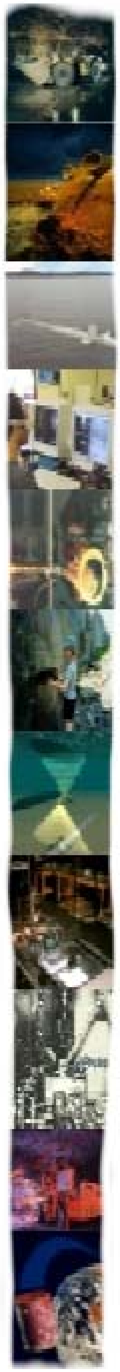
- average filter- used for noise smoothing, low-pass filtering

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3 window



Has a blurring effect.



Masks – edge detection

Prewitt

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

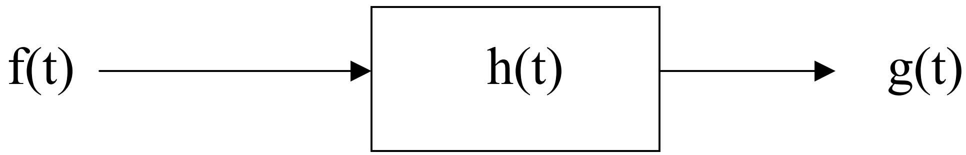
Sobel

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Transformations

- Images can be transformed using techniques similar to those in systems and signals.



Except now the signal is represented in 2-D space instead of 1-D time.

Examples of transformations include:

- Fourier transforms

- Cosine transforms

- Walsh transform

Image Analysis

- Goal is to extract important features from image data, from which a description, interpretation, or understanding of the scene can be provided.

Image Analysis

Feature Extraction

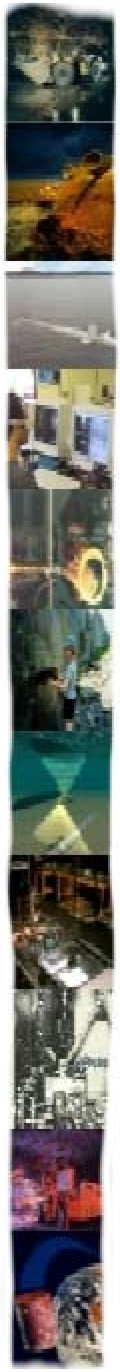
- Spatial features
- Transform features
- Edges and boundaries
- Shape features
- Moments
- Texture

Segmentation

- Template matching
- Thresholding
- Boundary detection
- Clustering
- Quad-trees
- Texture matching

Classification

- Clustering
- Statistical
- Decision trees
- Similarity Measures
- Min. spanning tree



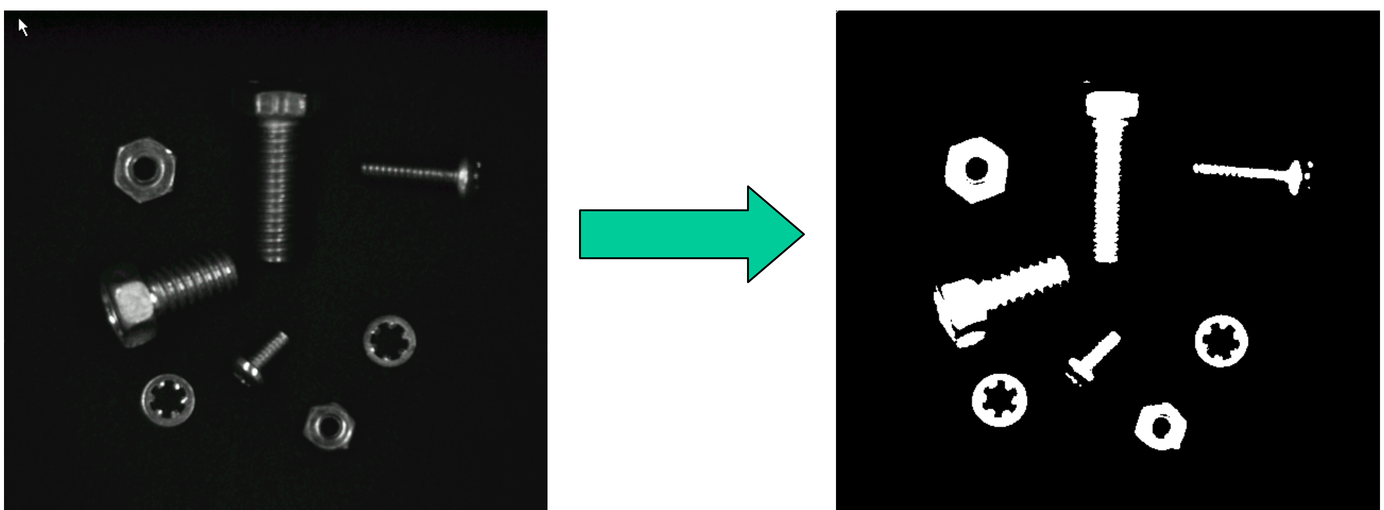
Segmentation

Includes:

1. Thresholding
2. Edge detection

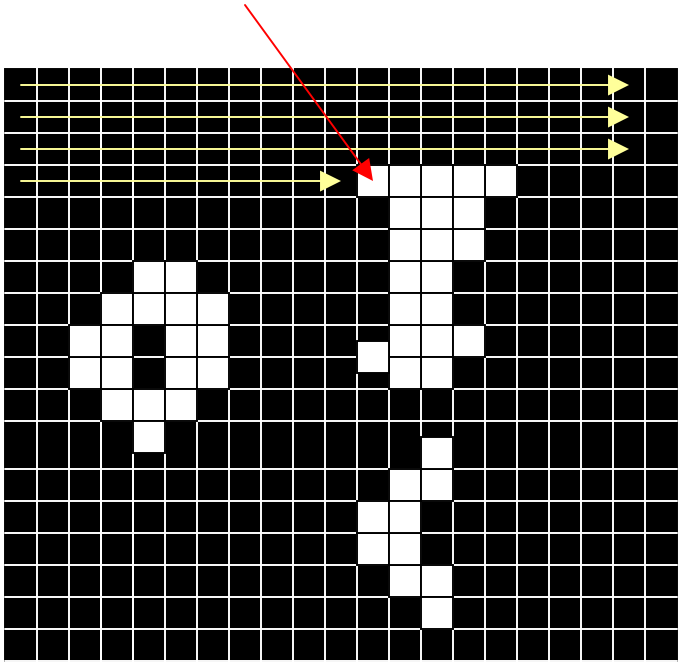
Thresholding

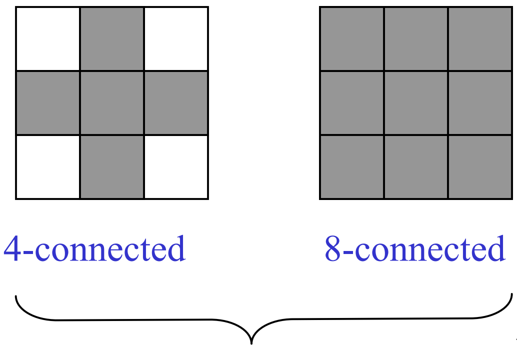
- separate different portions of the image based on the gray level values.

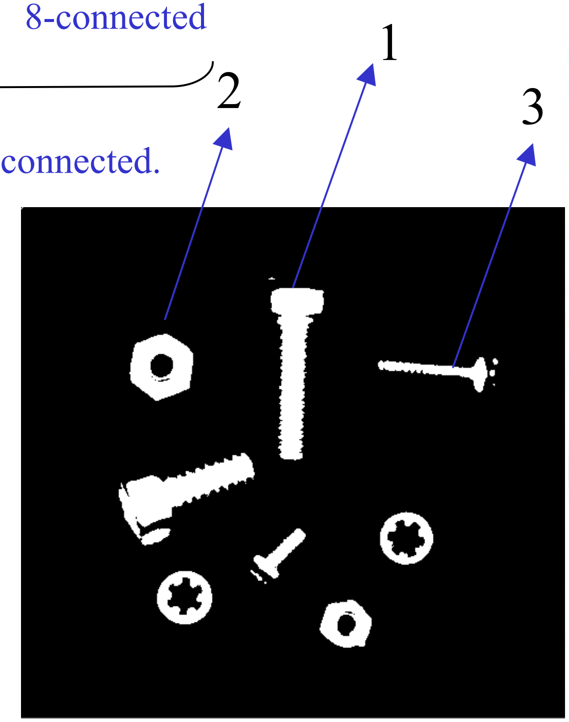


Blob Labeling

- Each blob is labeled with a unique value.
- Scan image until a “on” pixel is encountered. Label every pixel connected to the pixel with the same label value



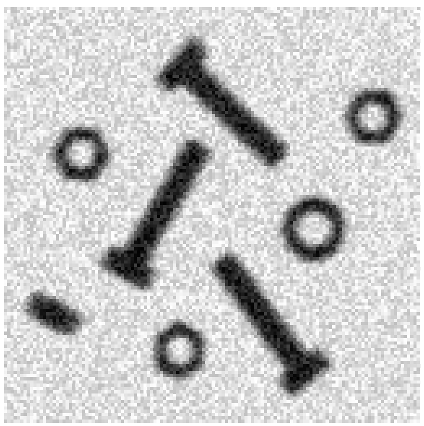


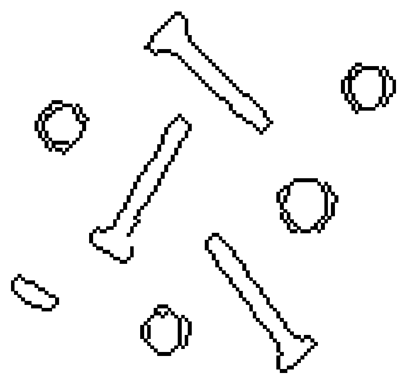


Edge Detection

Edge detection

- segment the image by finding sharp edges.
- Can be performed using a spatial mask

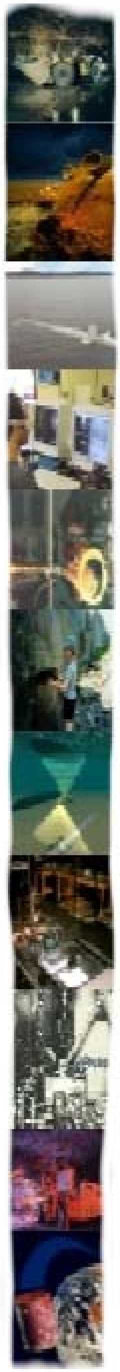




Prewitt

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1



Feature extraction

Feature extraction involves finding features of the segmented image.

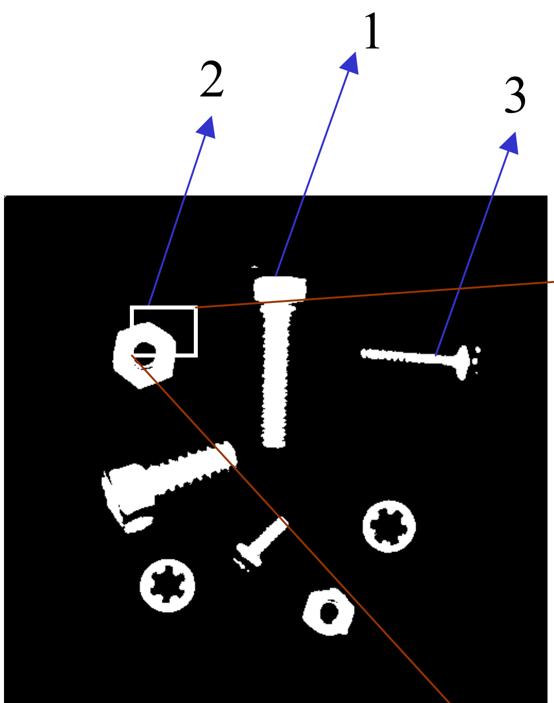
Usually performed on a binary image produced from a thresholding operation.

Common features include:

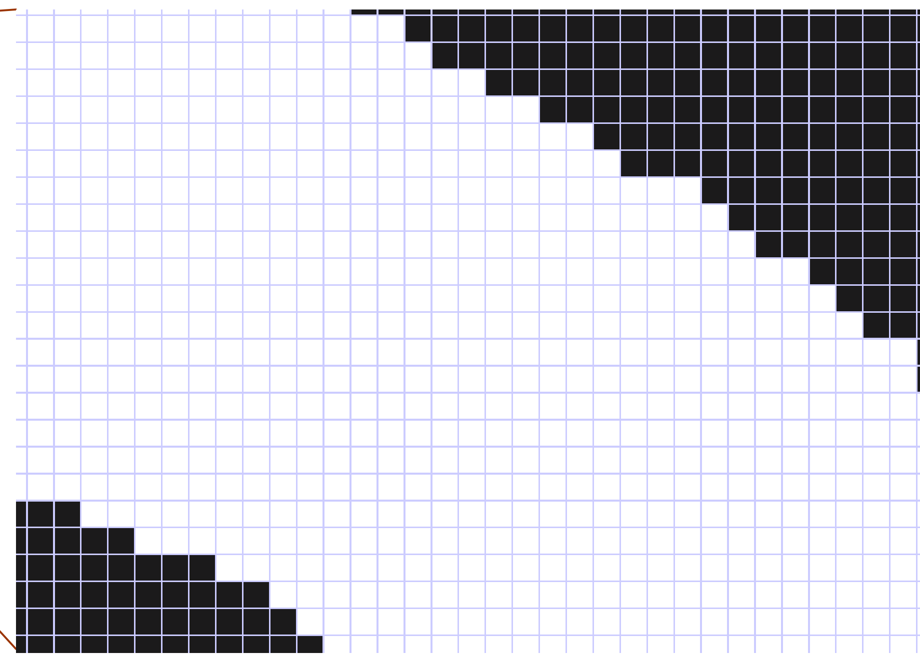
1. Area.
2. Perimeter.
3. Center of mass.
4. Compactness.

Feature extraction - area

Count the number of pixels in each labeled blob.



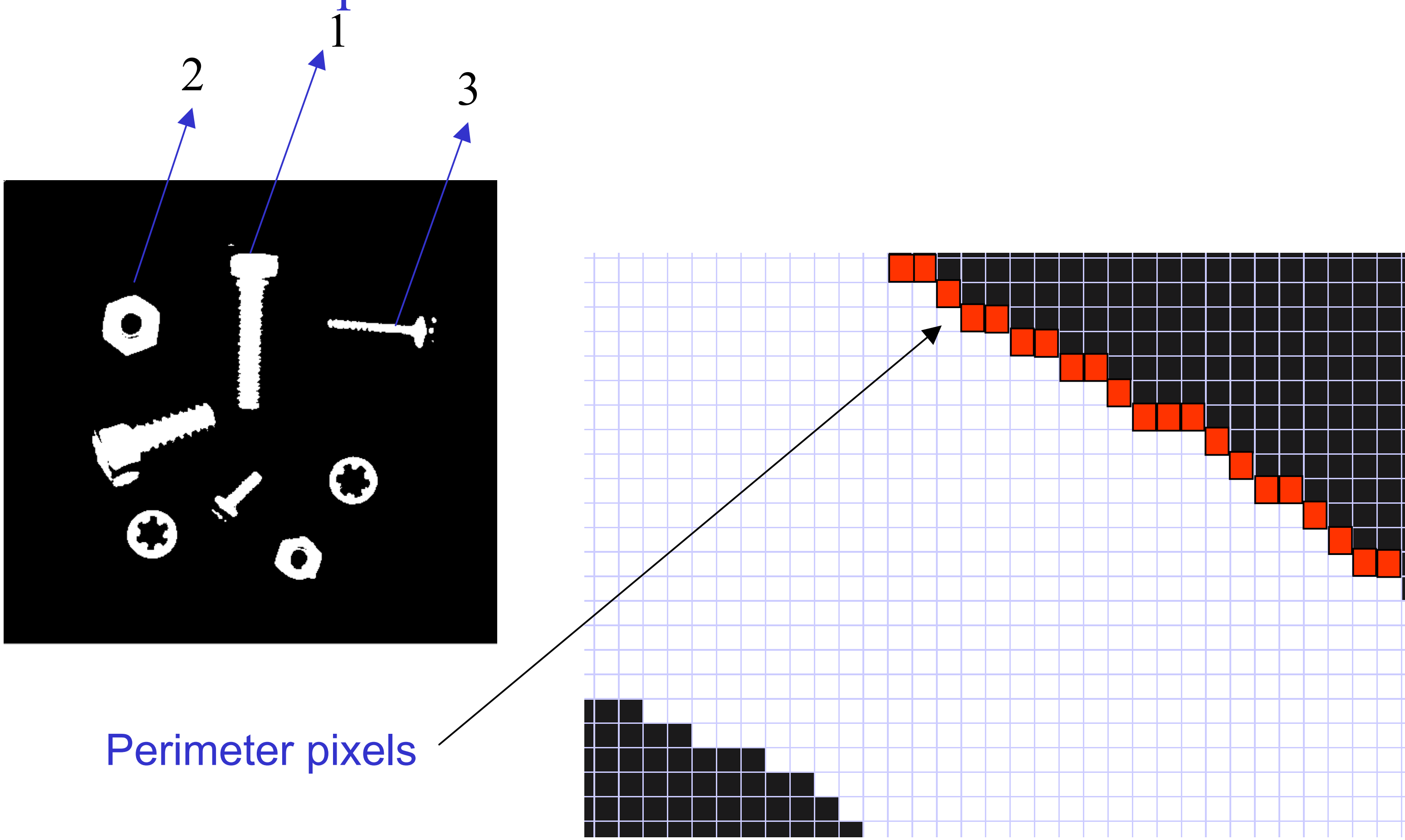
Simple algorithm – scan images and count the number of pixels labeled with a “2”

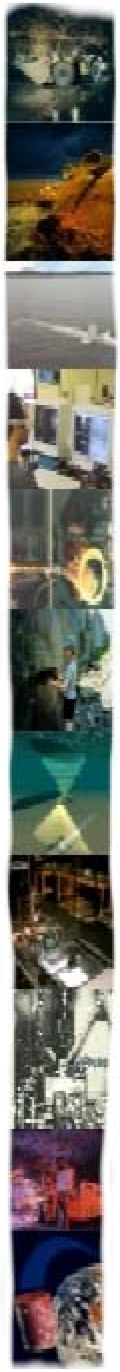


```
for(i=0;i<width;i++){
    for(j=0;jheight;j++){
        if(image[i][j]==2
            area[2] += 1;
    }
}
```

Feature extraction - perimeter

The number of pixels in a blob that border a “0”





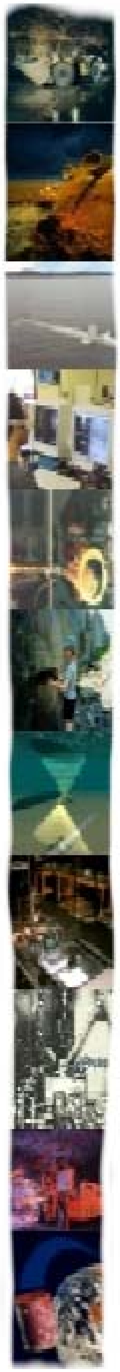
Feature extraction - compactness

- Other features can be obtained from simple measurements.

- Compactness = permiter^2/4πarea
- For a perfect circle
  - Compactness $= (2\pi r)^2/4\ \pi(\pi r^2)$

    $= 4\pi^2 r^2 / 4\pi^2 r^2 = 1$

- Long skinny blobs will have a high value for compactness.



Other Features

Table 1. Representative shape descriptors

$$Formfactor = \frac{4\pi \cdot Area}{Perimeter^2}$$

$$Roundness = \frac{4 \cdot Area}{\pi \cdot Max\ Diameter^2}$$

$$Aspect\ Ratio = \frac{Max\ Diameter}{Min\ Diameter}$$

$$Elongation = \frac{Fiber\ Length}{Fiber\ Width}$$

$$Curl = \frac{Length}{Fiber\ Length}$$

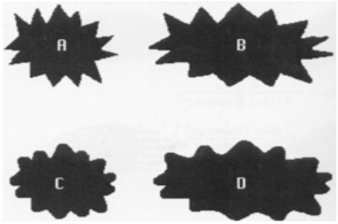
$$Convexity = \frac{Convex\ Perimeter}{Perimeter}$$

$$Solidity = \frac{Area}{Convex\ Area}$$

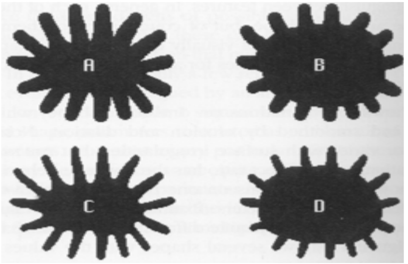
$$Compactness = \frac{\sqrt{\left(\frac{4}{\pi}\right) Area}}{Max\ Diameter}$$

$$Modification\ Ratio = \frac{Inscribed\ Diameter}{Maximum\ Diameter}$$

$$Extent = \frac{Net\ Area}{Bounding\ Rectangle}$$

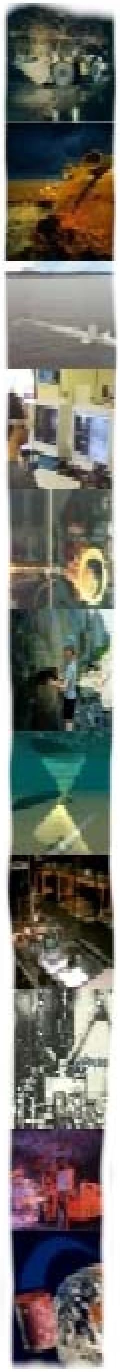


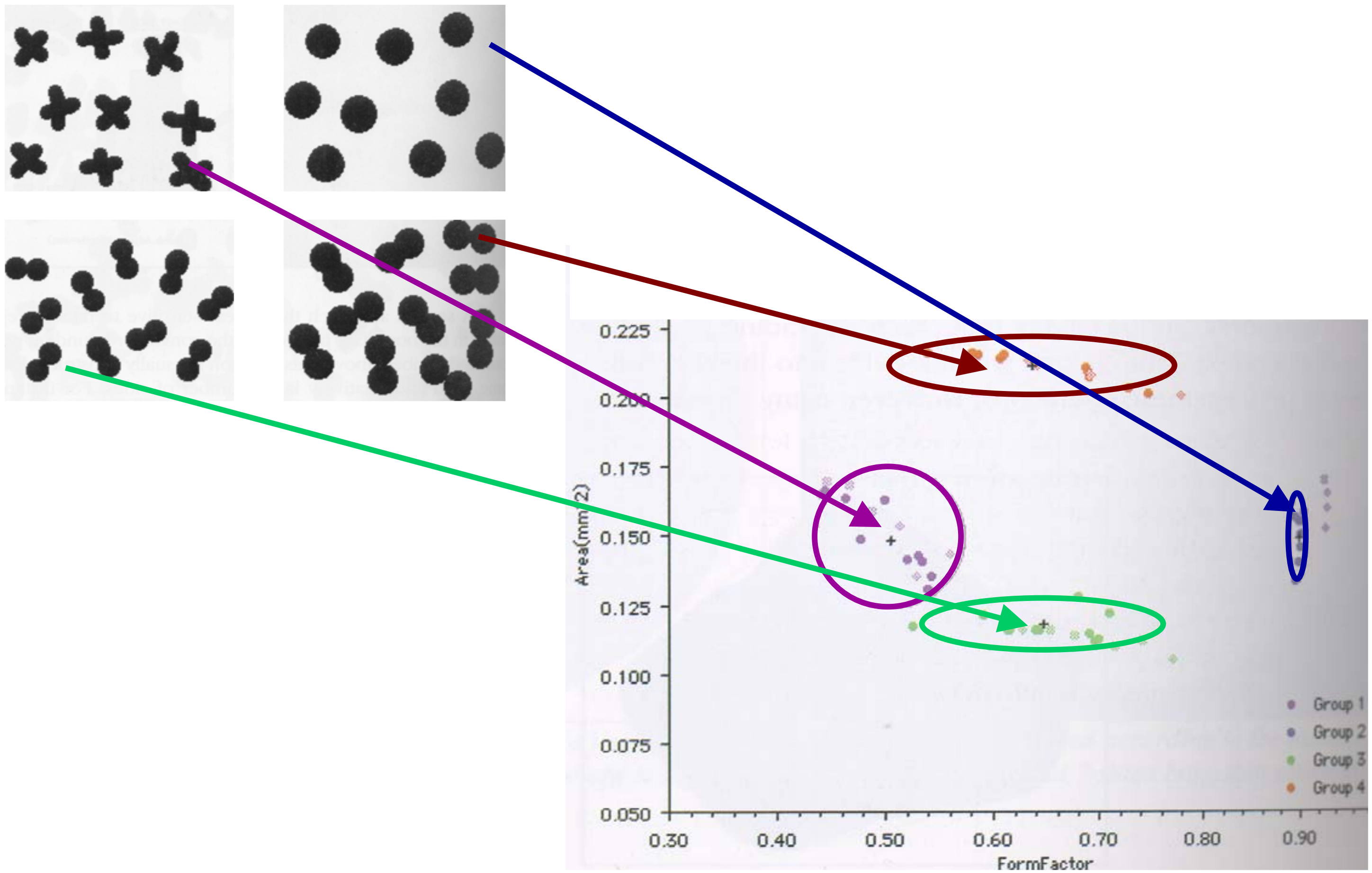
	Formfactor	Aspect Ratio
A	0.257	1.339
B	0.256	2.005
C	0.459	1.294
D	0.457	2.017



	Roundness	Convexity	Solidity	Compactness
A	0.587	0.351	0.731	0.766
B	0.584	0.483	0.782	0.764
C	0.447	0.349	0.592	0.668
D	0.589	0.497	0.714	0.768

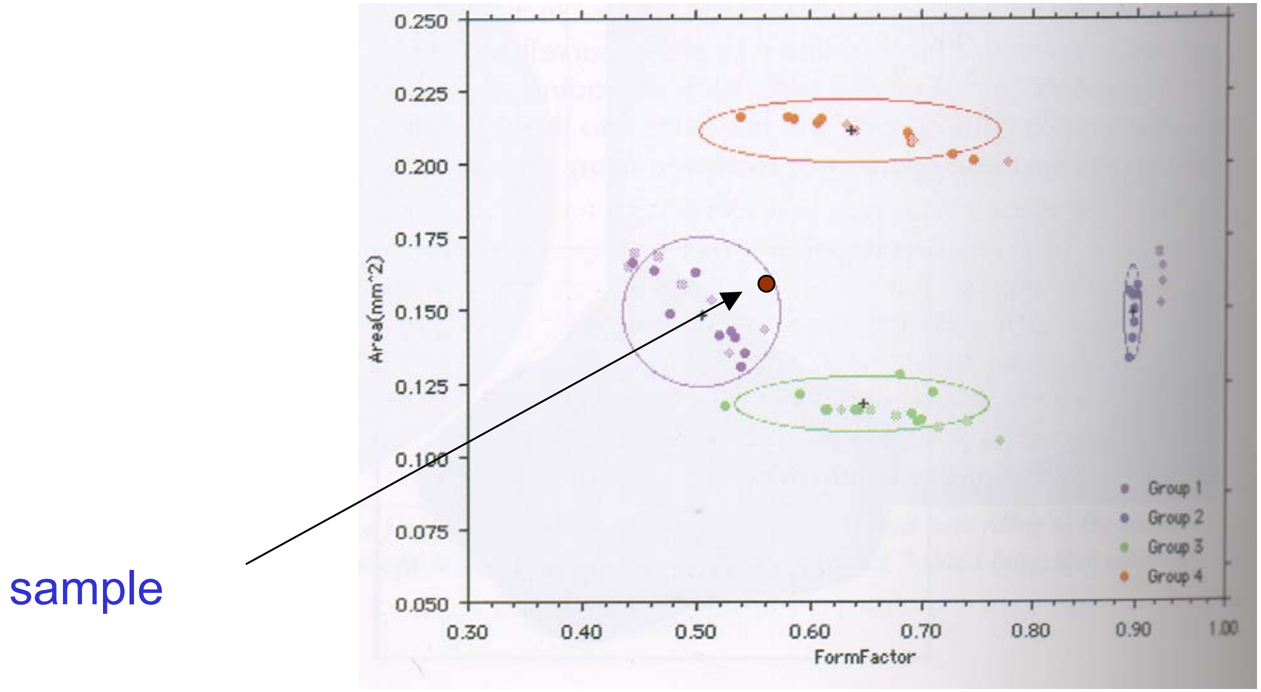
Pattern Classification





Simple Pattern Classification

- Nearest neighbor
- N nearest neighbors



Next Day - Robotics