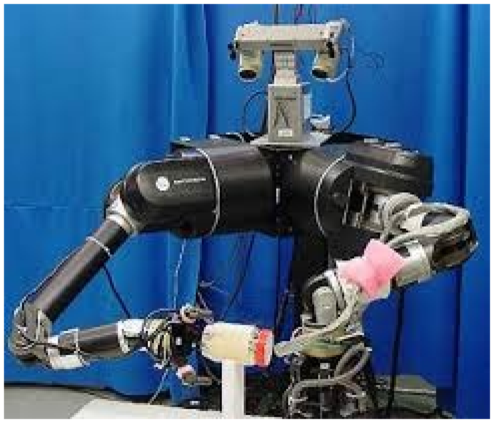
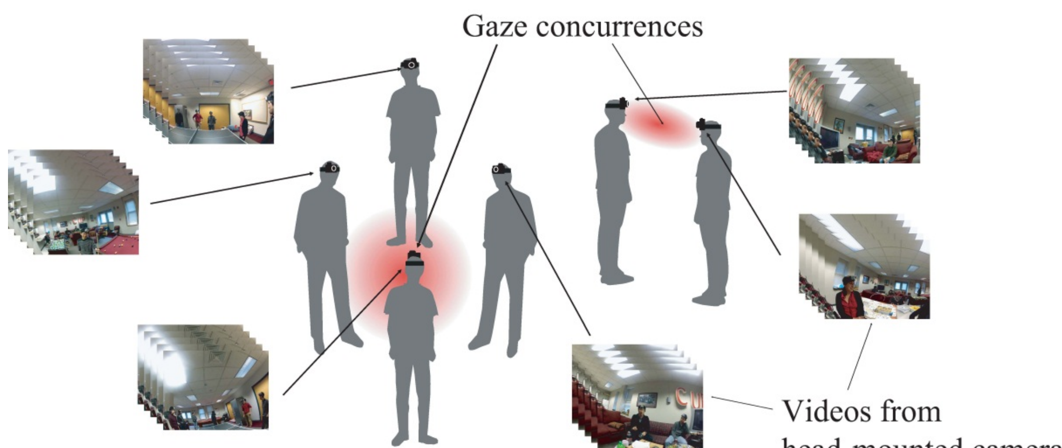
Computer Vision

Howie Choset
http://www.cs.cmu.edu.edu/~choset

Introduction to Robotics
http://generalrobotics.org

What is vision?





What is this a picture of?

Is it snowing?

What is this a picture of?

Is it snowing?

What is this a picture of?

Is it snowing?

What is this a picture of?

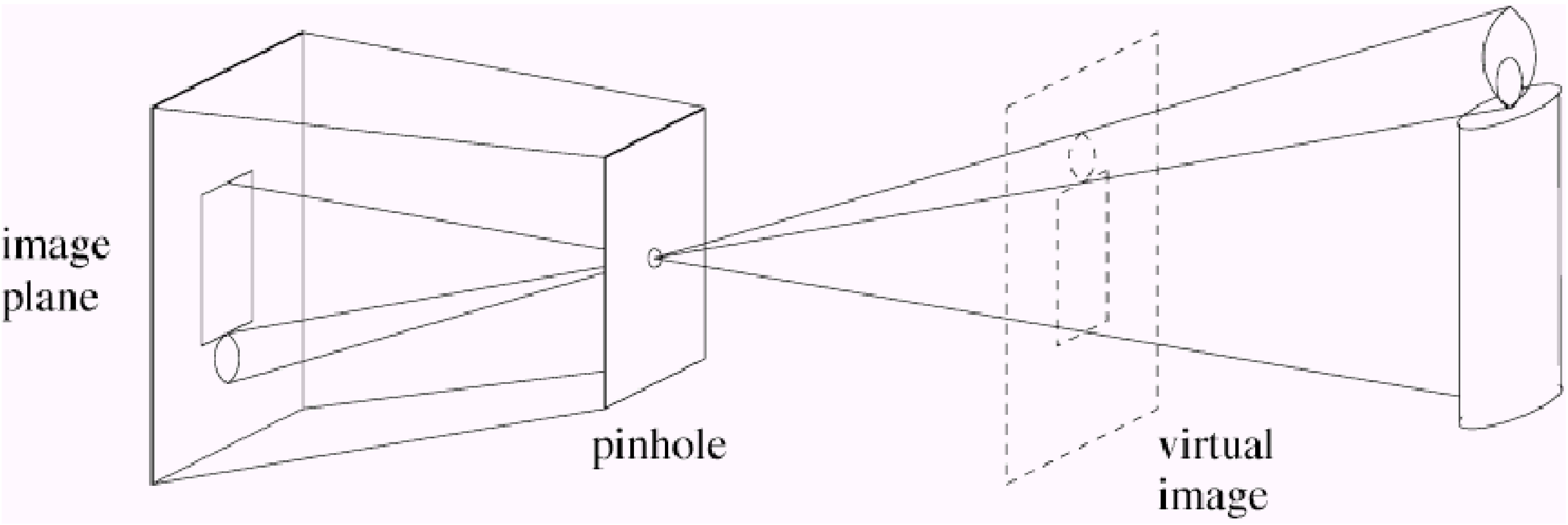
Is it snowing?

Occlusion

Baxter is occluding the stop sign

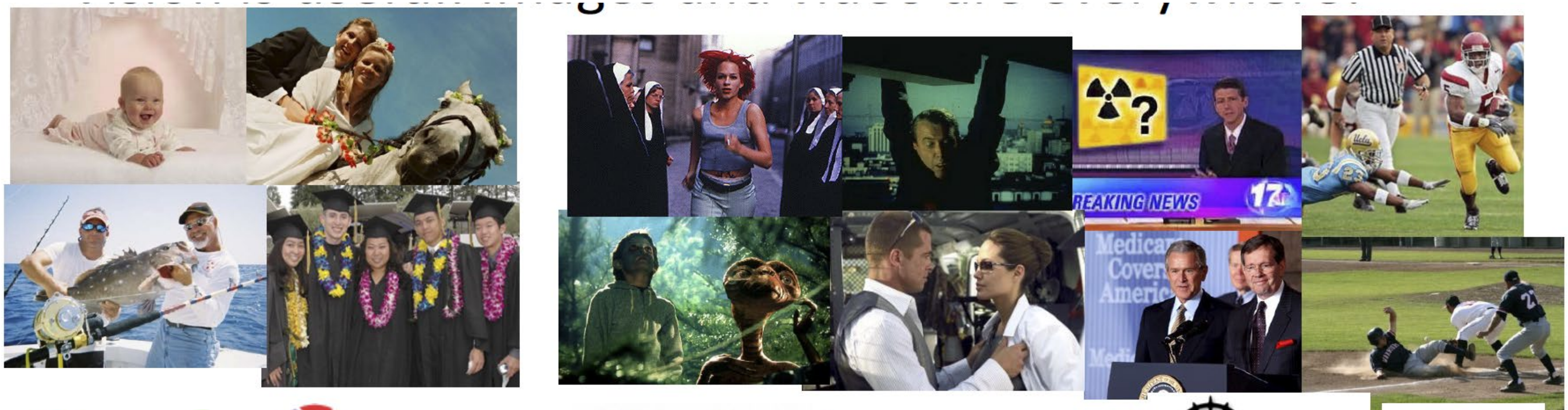
What is computer vision?

What is computer vision?



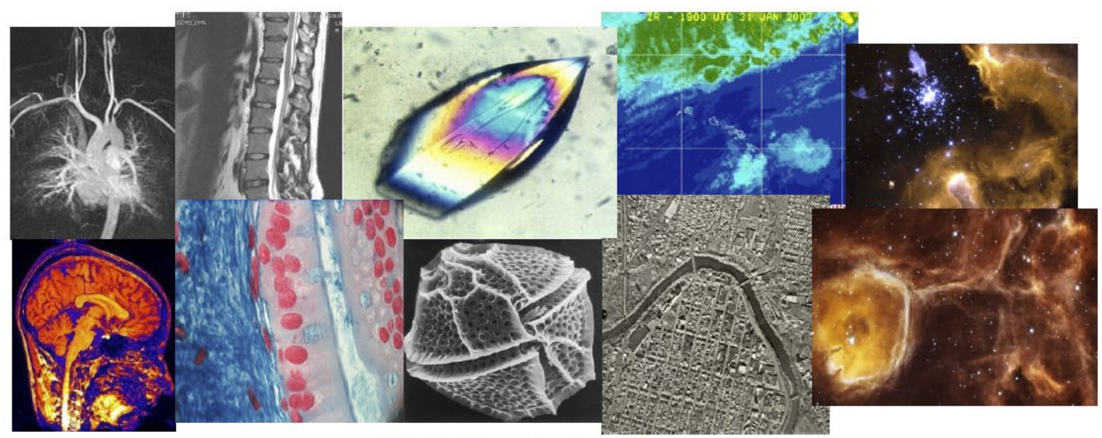
Slide credit: Fei-Fei Li

Why study vision?



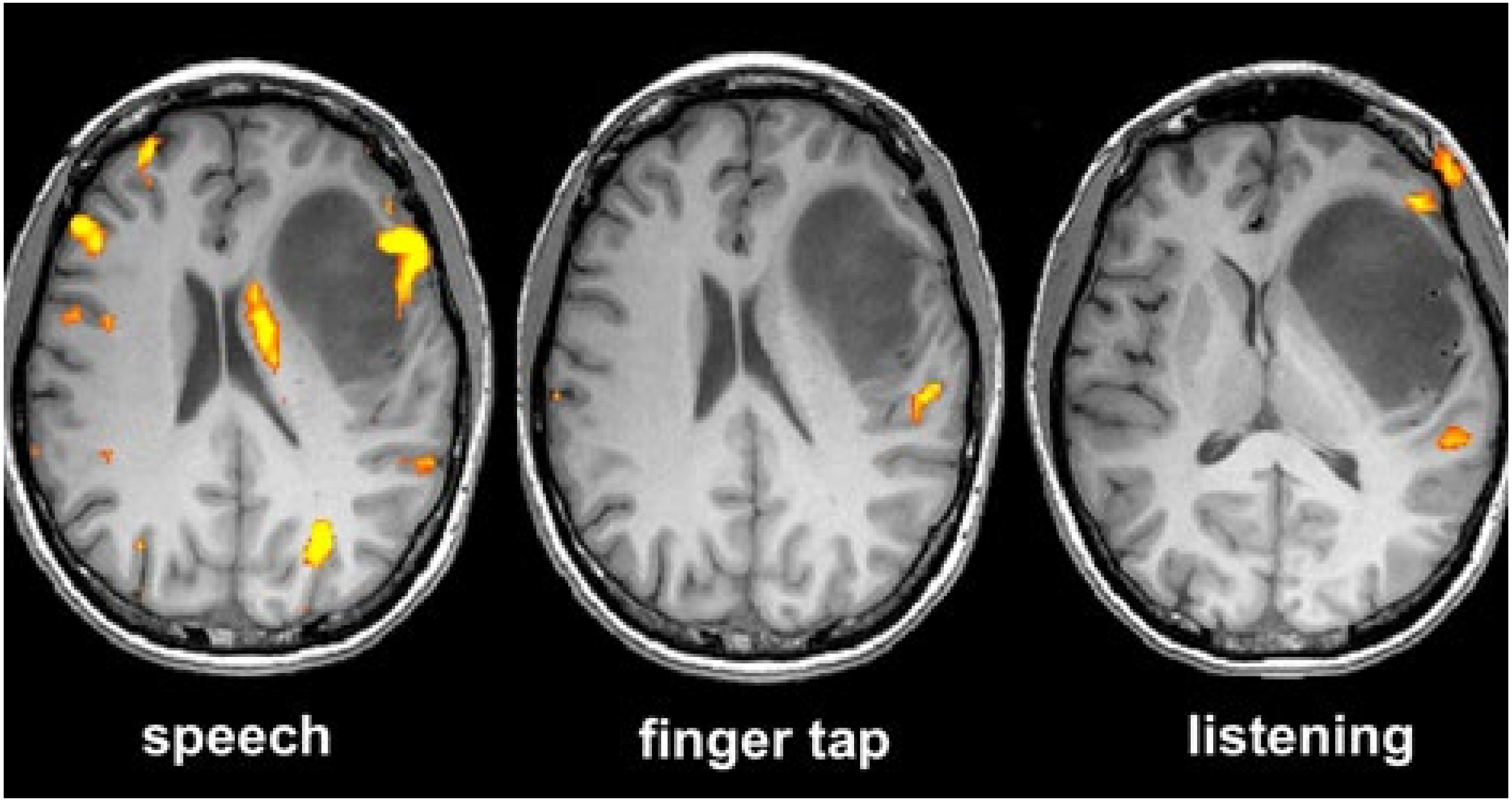
Google Image Search
Picasa
flickr
webshots
picsearch
YouTube Broadcast Yourself

Surveillance and security



Medical and scientific images

Computer vision is not only for things a person could see



Is Vision Hard?

https://dspace.mit.edu/bitstream/handle/1721.1/6125/AIM-100.pdf?sequence=2

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

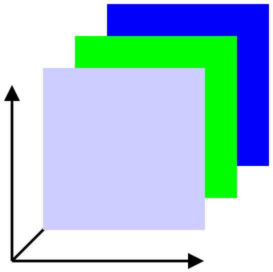
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Images

- Discrete representation of a continuous function
  - Pixel: Picture Element – cell of constant color in a digital image
  - An image is a two dimensional array of pixels
  - Pixel: numeric value representing a uniform portion of an image
- Resolution
  - Number of pixels across in horizontal
  - Number of pixels in the vertical
  - Number of layers used for color
    - Often measured in bits per pixel (bpp) where each color uses 8 bits of data
  - Ex: 640x480x24bpp

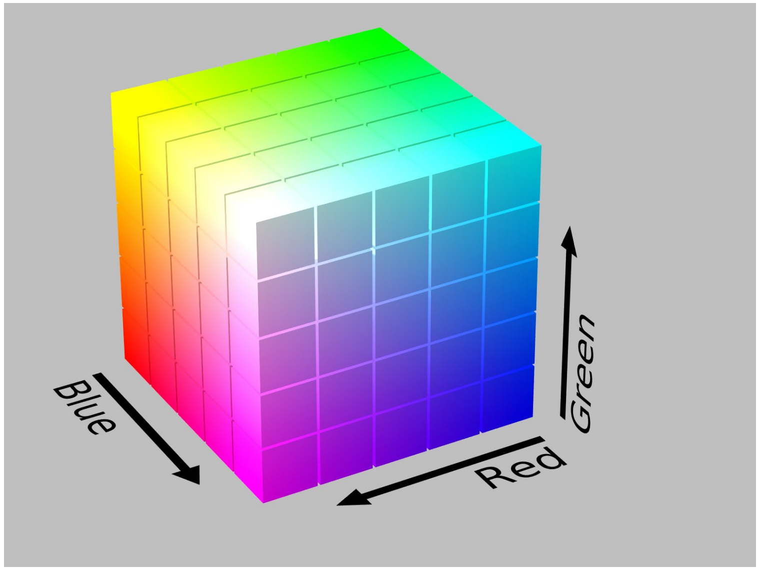
Images

- Binary images: Two color image
  - Pixel is only one byte of information
  - Indicates if the intensity of color is above or below some nominal value
  - Thresholding
- Grayscale
  - All pixels represent the intensity of light in an image, be it red, green, blue, or another color
    - Like holding a piece of transparent colored plastic over your eyes
  - Intensity of light in a pixel is stored as a number, generally 0..255 inclusive
- Color
  - Three grayscale images layered on top of eachother with each layer indicating the intensity of a specific color light, generally red, green, and blue (RGB)
  - Third dimension in a digital image

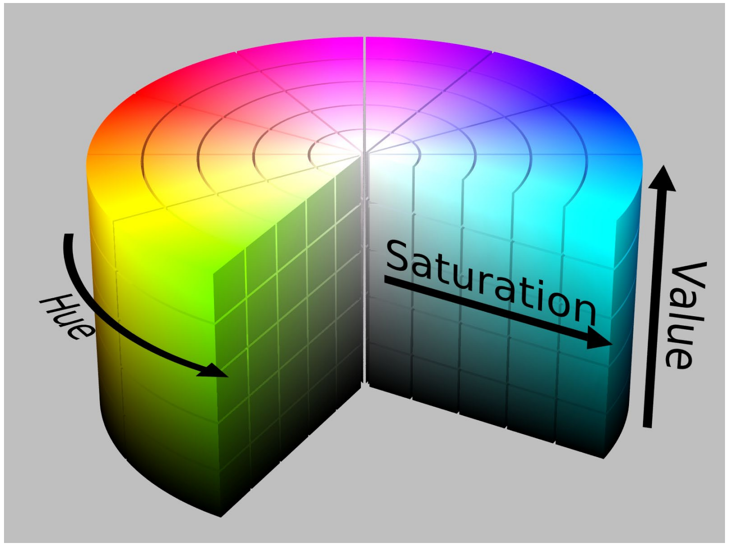


Common color image formats

- RGB



- HSV



Many different formats

- How the pixels are stored
  - jpg, gif, png, you can even make your own format
- What does that numebr represents
  - Does it encode a color, an intensity, binary value

Goal of Vision

What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

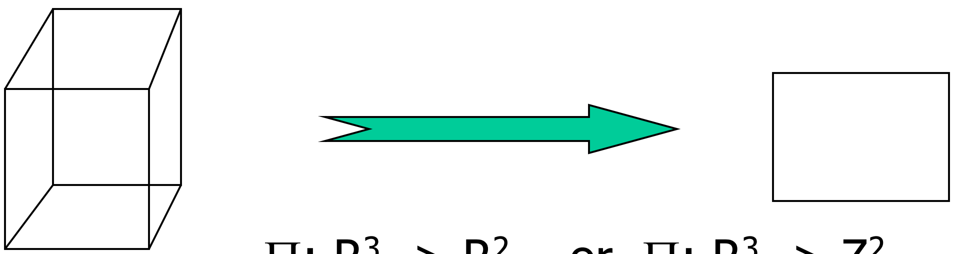
Source: S. Narasimhan

THE ROBOTICS INSTITUTE

Reconstructing 3D

Who is taller?

Recover Projection

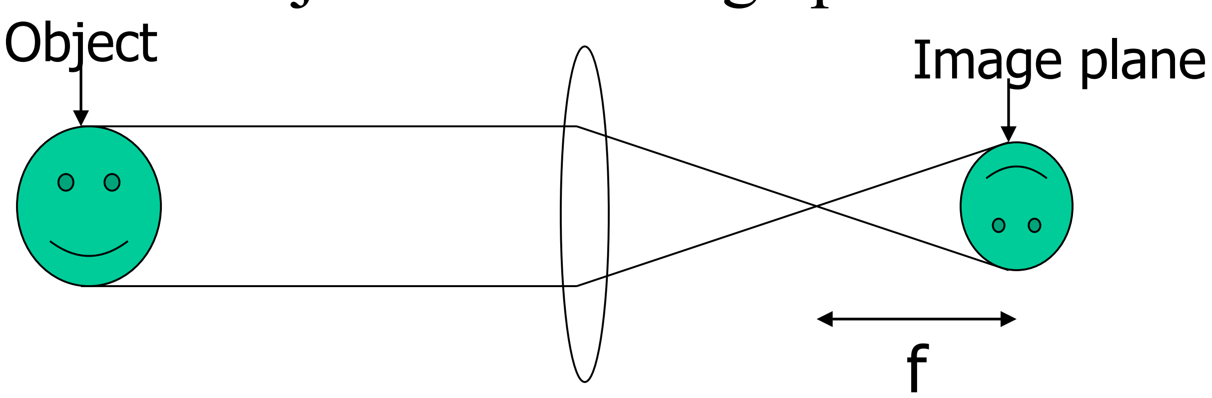


$\Pi: R^3 \to R^2$ or $\Pi: R^3 \to Z^2$

Recover third dimension or just infer stuff

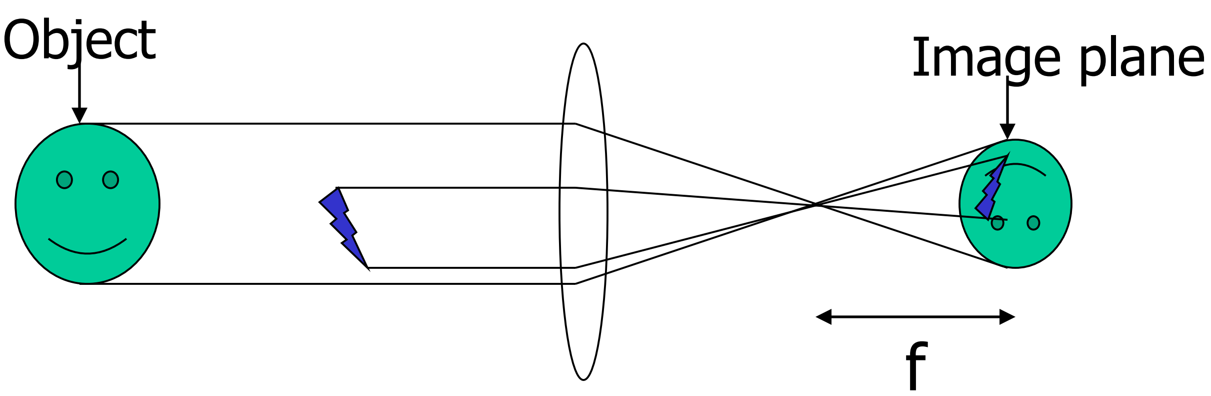
Optics

- Focal length
  - Length f of projection through lens on image plane
- Inversion
  - Projection on image plane is inverted



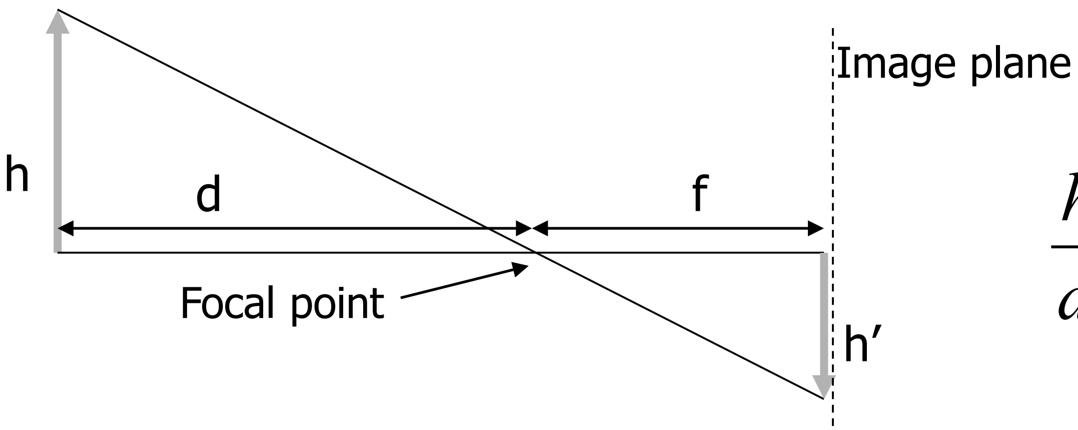
Perspective

- 1 Point Perspective
  - Using similar triangles, it is possible to determine the relative sizes of objects in an image
  - Given a calibrated camera (predetermine a mathematical relationship between size on the image plane and the actual object)



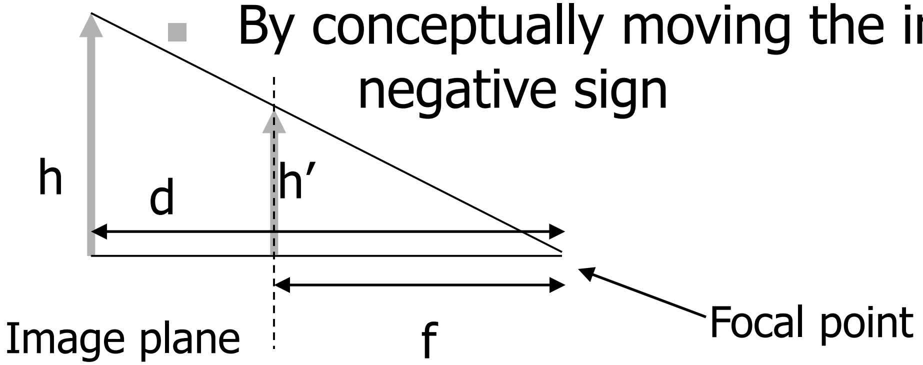
Projection on the image plane

- Size of an image on the image plane is inversely proportional to the distance from the focal point



$$\frac{h}{d} = \frac{-h'}{f}$$

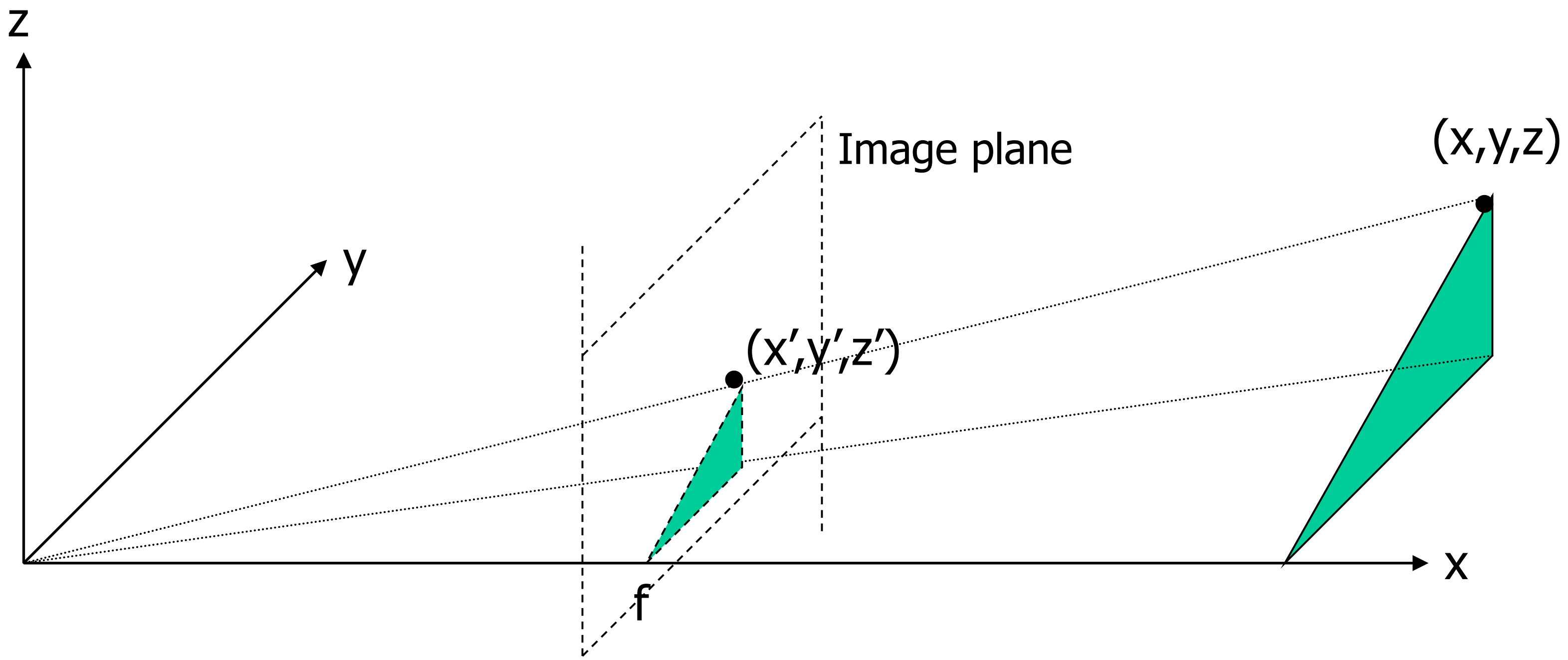
- By conceptually moving the image plane, we can eliminate the negative sign



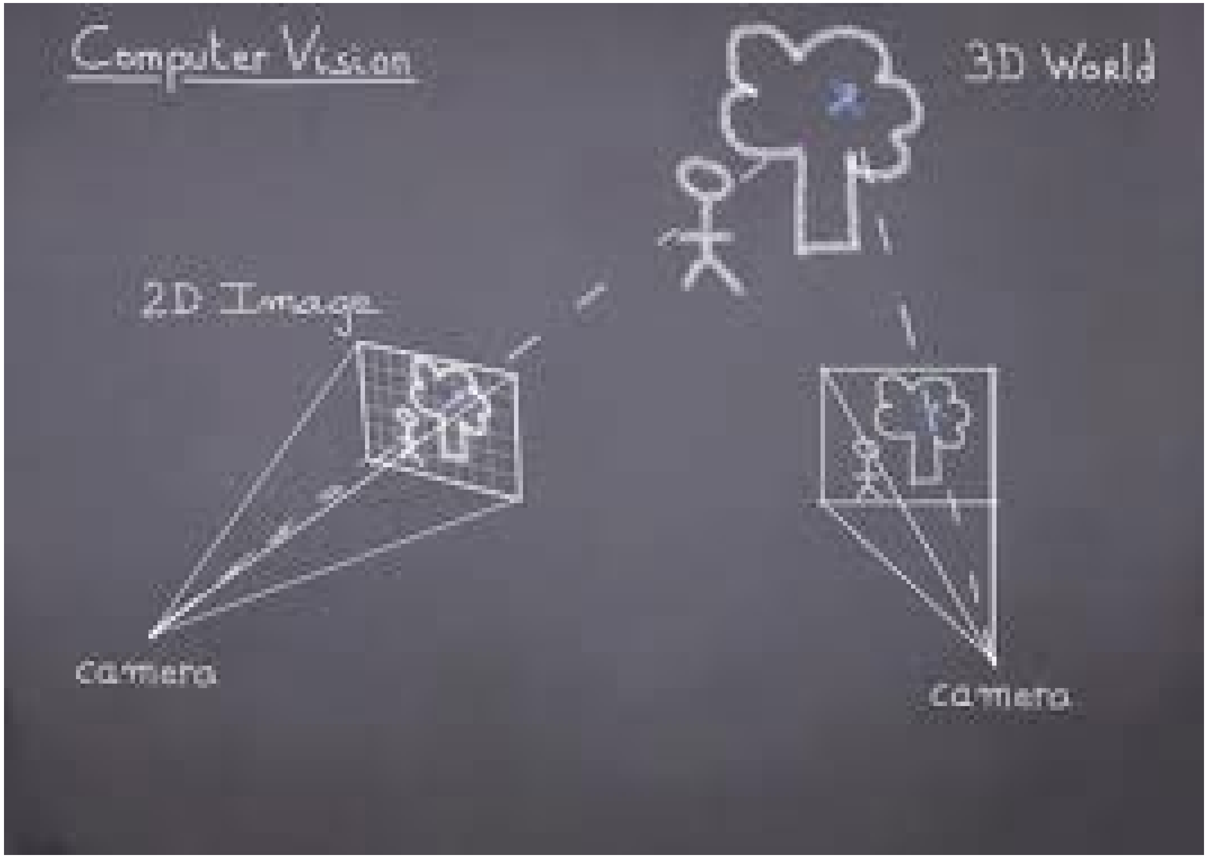
$$\frac{h}{d} = \frac{h'}{f}$$

Move to three dimensions

$$\frac{x}{x'} = \frac{x}{f} = \frac{y}{y'} = \frac{z}{z'} \Rightarrow x' = f\frac{x}{z}, y' = f\frac{y}{z'}$$

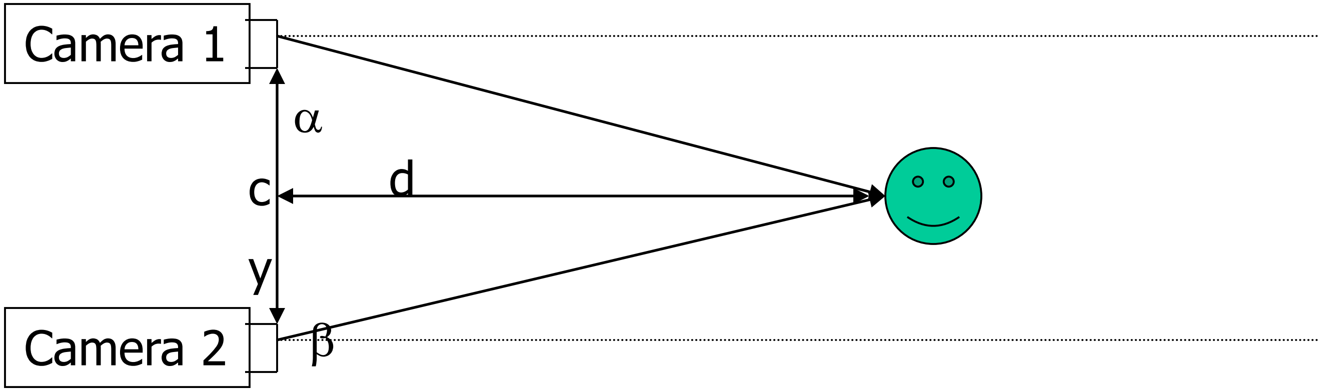


Stereo



Stereo Vision

- Way of calculating depth from two dimensional images using two cameras



- d and y are unknowns, α and β can be determined processing and c is known

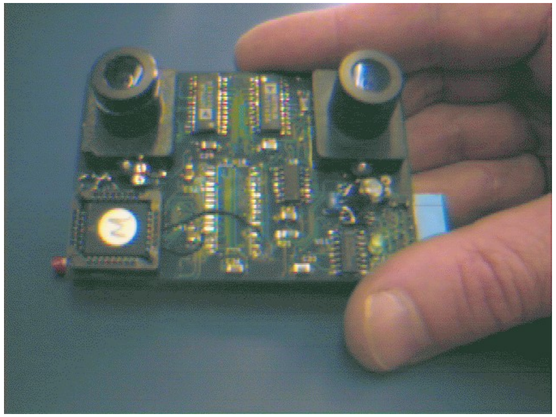
$$\tan\alpha = \frac{d}{c-y}$$

$$\tan\beta = \frac{d}{y}$$

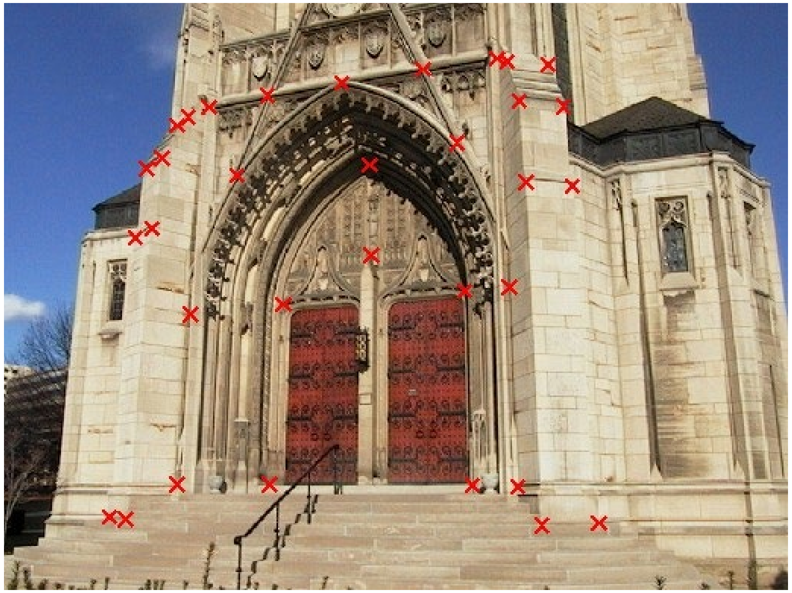
$$y = \frac{c\tan\alpha}{\tan\alpha - \tan\beta}$$

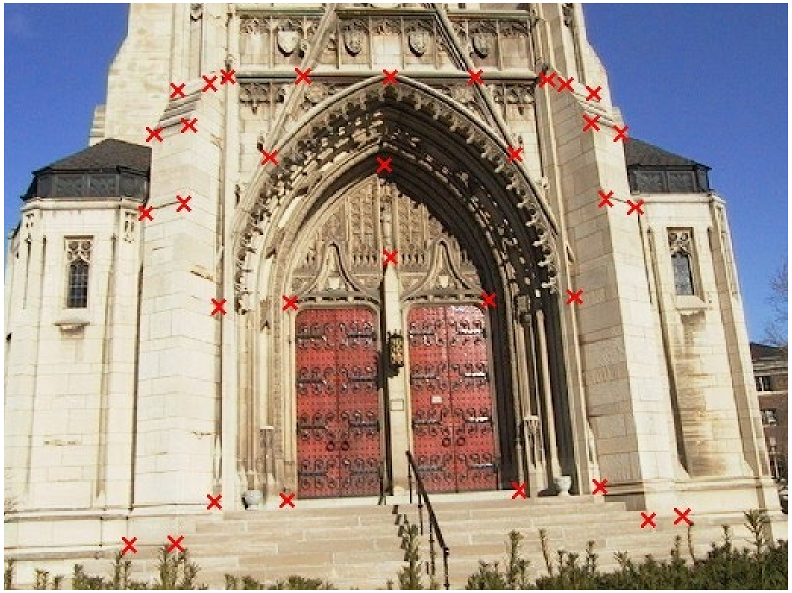
$$d = y\tan\beta$$

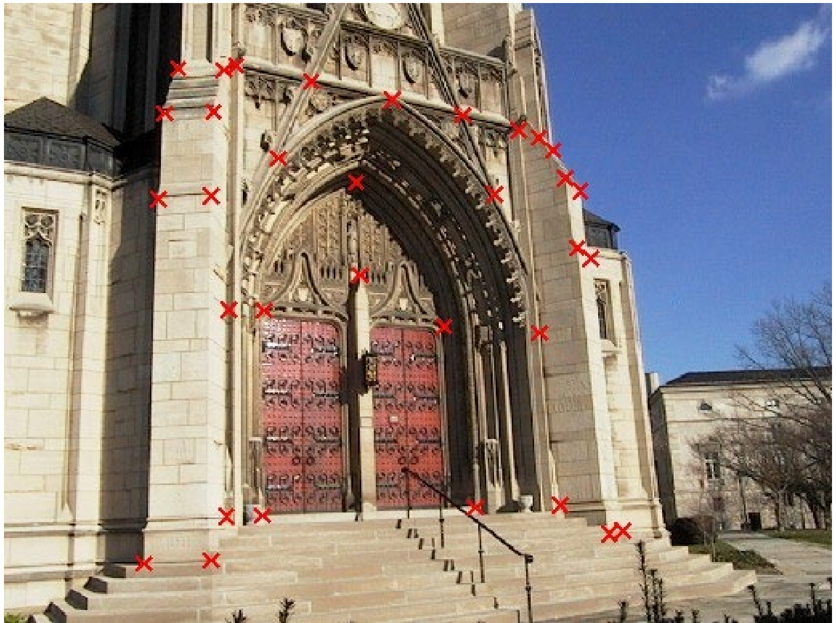
Multi-Camera Geometry: Stereo



Multi-Camera Geometry

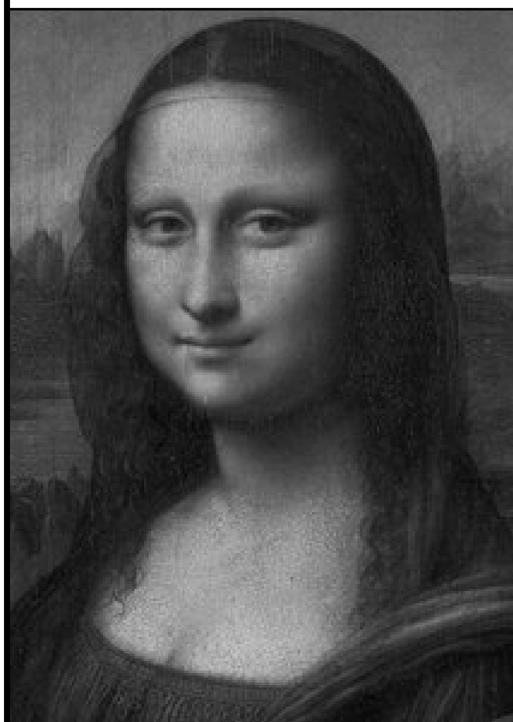






Getting less obvious information from an image

Grayscale vs. Binary image

Grayscale	Binary threshold
	

Thresholding

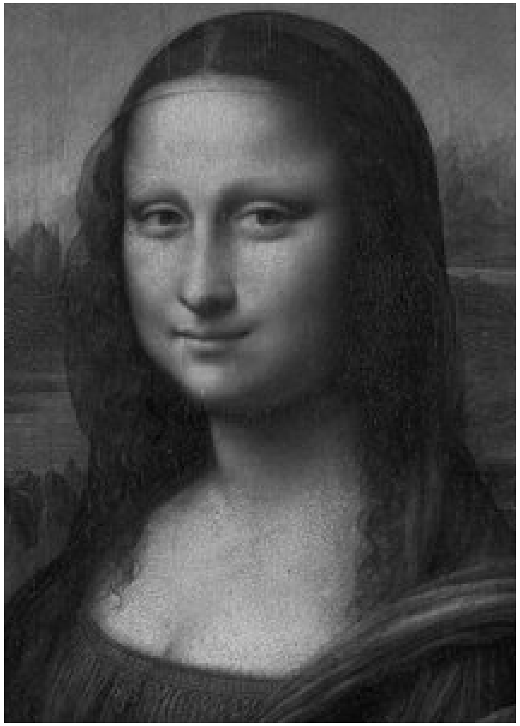
- Purpose
  - Trying to find areas of high color intensity
  - Highlights locations of different features of the image (notice Mona's eyes)
  - Image compression, use fewer bits to encode a pixel
- How done
  - Decide on a value μ
  - Scan every pixel in the image
    - If it is greater than μ, make it 255
    - If it is less than μ, make it 0
  - Picking a good μ
    - Often 128 is a good value to start with
    - Use a histogram to determine values based on color frequency features

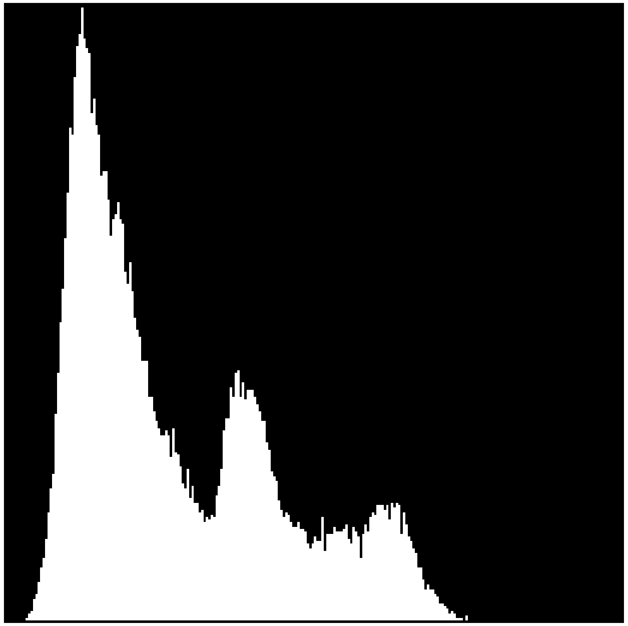
Histogram

- Measure the number of pixels of different values in an image.
- Yields information such as the brightness of an image, important color features, possibilities of color elimination for compression

1. FOR $i = 0$ TO $N - 1$
2. $\quad H[i] \leftarrow 0$
3. FOR $r = 0$ TO $N_{rows} - 1$
4. $\quad$ FOR $c = 0$ TO $N_{cols} - 1$
5. $\quad\quad Index \leftarrow Image(r, c)$
6. $\quad\quad H[Index] \leftarrow H[Index] + 1$

Mona's Histogram





How to Choose a Threshold

Probability that a pixel has a gray value *z*

$$P(z) = \frac{H(z)}{N_{Rows} \times N_{Cols}}$$

We denote the mean by μ, and compute it by

$$\mu = \sum_{z=0}^{N-1} zP(z)$$

Following Copied from Robot Motion and Control
By
Spong, Hutchinson, Vidyasagar

Variance

- What is the mean for an image
  - Half pixels 127, half pixels 128
  - Half pixel 0, half pixels 255
- 127.5
  - Same mean, Different images
- Average deviation or variance
  - Variance(X) = E[(X - μ)^2]
  - standard deviation

$$\sigma = \sqrt{\mathrm{E}[(X-\mu)^2]}$$
$$= \sqrt{\mathrm{E}[X^2] + \mathrm{E}[(-2\mu X)] + \mathrm{E}[\mu^2]} = \sqrt{\mathrm{E}[X^2] - 2\mu\,\mathrm{E}[X] + \mu^2}$$
$$= \sqrt{\mathrm{E}[X^2] - 2\mu^2 + \mu^2} = \sqrt{\mathrm{E}[X^2] - \mu^2}$$
$$= \sqrt{\mathrm{E}[X^2] - (\mathrm{E}[X])^2}$$

Automatic Threshold Selection

$q_i(z_t)$ as the probability that a pixel in the image will belong to group i for a particular choice of threshold, z_t.

i = 0,1

Less than z_t More than z_t

$$q_0(z_t) = \frac{\sum_{z=0}^{z_t} H[z]}{(N_{rows} \times N_{cols})}, \quad q_1(z_t) = \frac{\sum_{z=z_t+1}^{N-1} H[z]}{(N_{rows} \times N_{cols})}$$

Approach: Pick a threshold z_t that minimizes the variance of the two resulting groups

More threshold selection

$$\mu_i = \sum_{z=0}^{N-1} z \frac{H_i[z]}{\sum_{z=0}^{N-1} H_i[z]} = \sum_{z=0}^{N-1} z \frac{H_i[z]/(N_{rows} \times N_{cols})}{\sum_{z=0}^{N-1} H_i[z]/(N_{rows} \times N_{cols})}$$

$$\mu_0(z_t) = \sum_{z=0}^{z_t} z \frac{P(z)}{q_0(z_t)},$$

$$\mu_1(z_t) = \sum_{z=z_t+1}^{N-1} z \frac{P(z)}{q_1(z_t)}$$

$$H_0[z] = \begin{cases} P(z) & z \leq z_t \\ 0 & \text{otherwise} \end{cases} \qquad q_0(z_t) = \frac{\sum_{z=0}^{z_t} H[z]}{(N_{rows} \times N_{cols})}$$

$$H_1[z] = \begin{cases} 0 & z \leq z_t \\ P(z) & \text{otherwise} \end{cases} \qquad q_1(z_t) = \frac{\sum_{z=z_t+1}^{N-1} H[z]}{(N_{rows} \times N_{cols})}$$

Two groups,
Background – below
Foreground - above

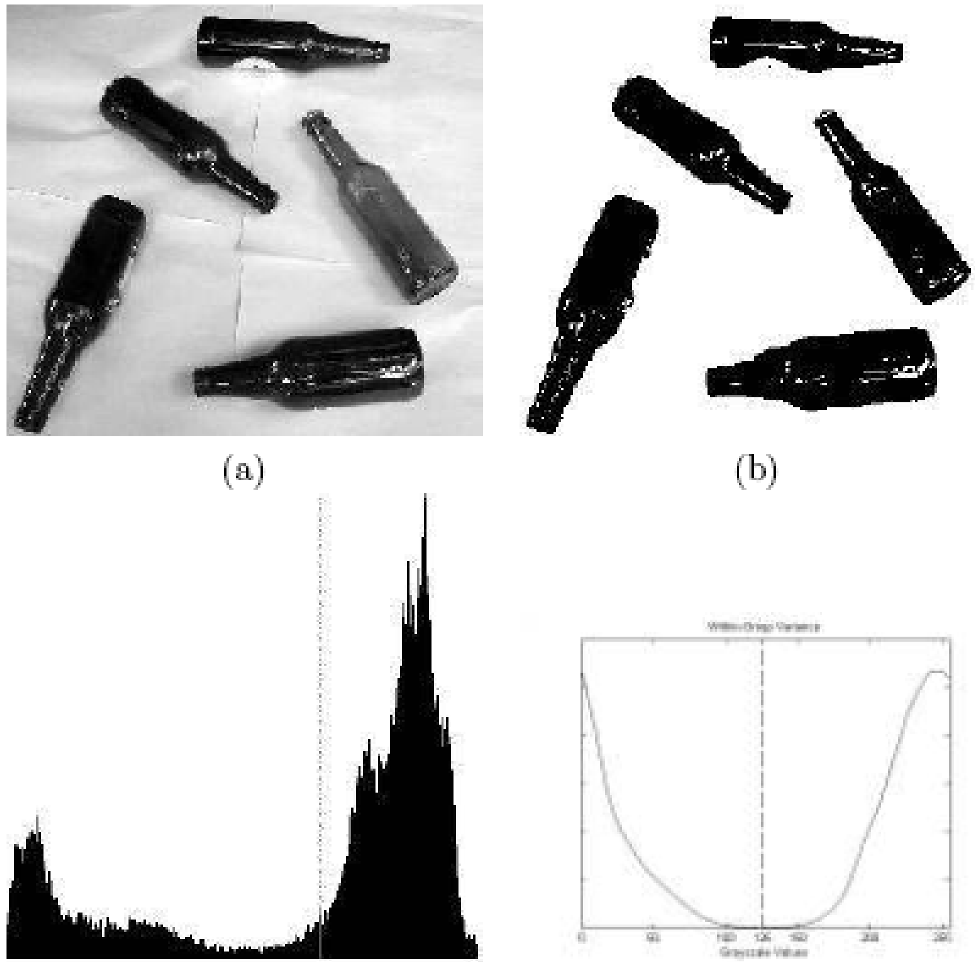
More threshold selection

$$\sigma_0^2(z_t) = \sum_{z=0}^{z_t} (z - \mu_0(z_t))^2 \frac{P(z)}{q_0(z_t)}$$

$$\sigma_1^2(z_t) = \sum_{z=z_t+1}^{N} (z - \mu_1(z_t))^2 \frac{P(z)}{q_1(z_t)}$$

We could pick z_t to minimize sum of variances $\sigma_i^2(z_t)$

BAD, because assumes equal number of pixels in back and foreground

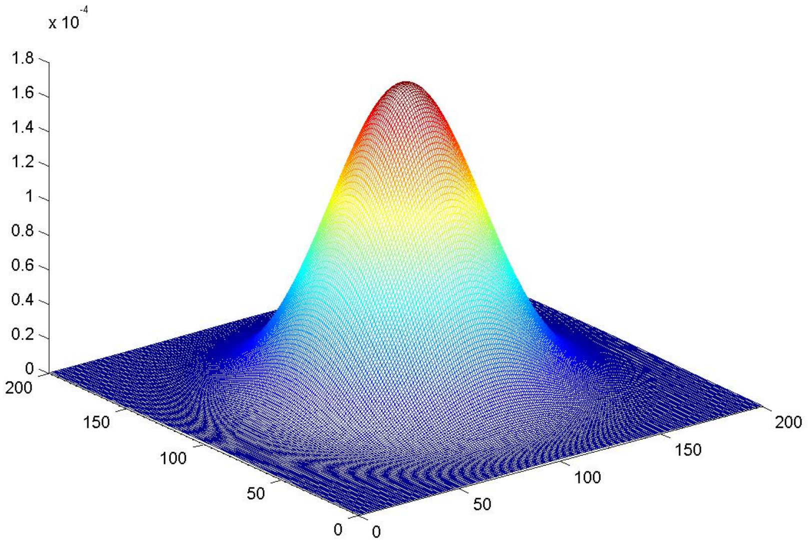


(a) (b) (c) (d)

(a) An image with 256 gray levels. (b) Thresholded version of the image in (a). (c) Histogram for the image shown in 11.2a (d) Within-group variance for the image shown in 11.2a

Gaussian Masks

- Used to smooth images and for noise reduction
- Use before edge detection to avoid spurious edges



Johann Carl Friedrich Gauss

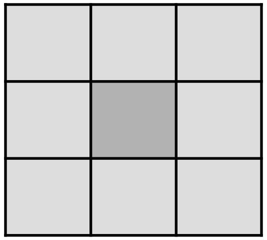
April 30, 1777 – Feb 23, 1855
number theory, statistics, analysis, differential geometry, geodesy, electrostatics, astronomy, and optics.

Connectivity

- Two conventions on considering two pixels next to each other

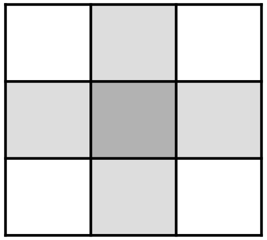
8 point connectivity

All pixels sharing a side or corner are considered adjacent



4 point connectivity

Only pixels sharing a side are considered adjacent



- To eliminate the ambiguity, we could define the shape of a pixel to be a hexagon

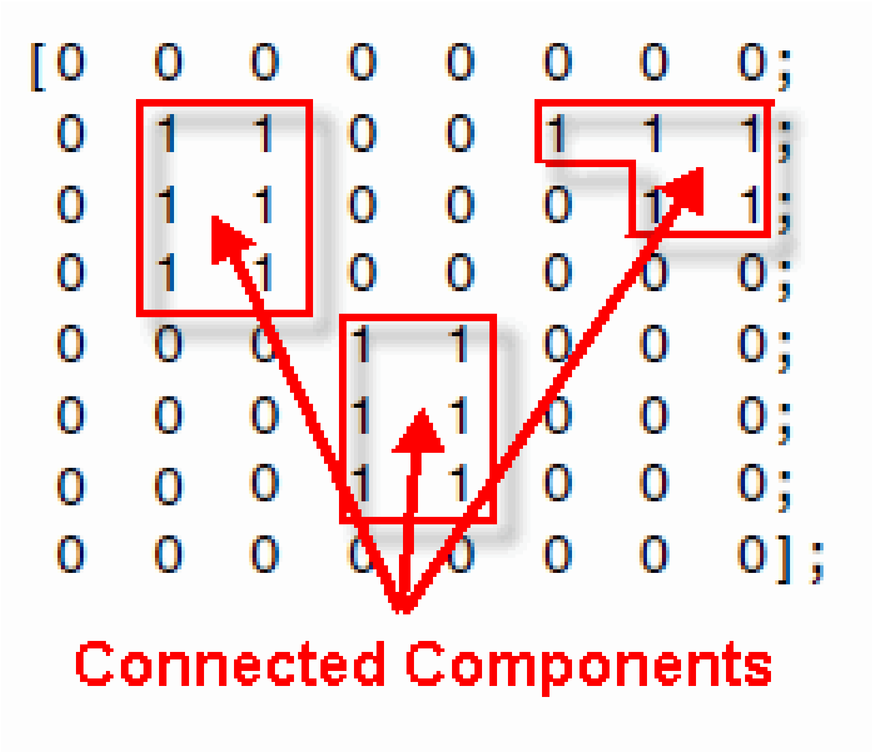
Segmentation: Wavefront

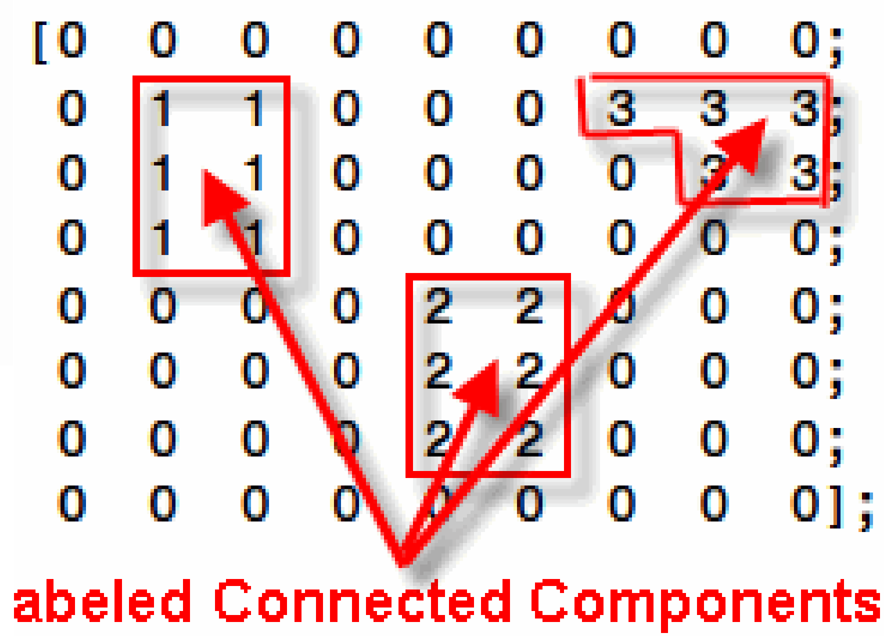
Assume a binary image with values of 0 or 1

- 1. Choose 1st pixel with value 1, make it a 2
- 2. For each neighbor, if it is also a 1, make it a 2 as well
- 3. Repeat step two for each neighbor until there are no neighbors with value 1
- 4. All pixels with a value 2 are are a continuous object

Or look for pixels below a threshold (instead of a 0) and otherwise above (instead of 1)

Input and Output





Segmentation: Double Raster

Assume a binary image with values of 0 or 1
Initialize cntr to 0

- 1. Perform a raster scan – across and down
  - a. Encounter a pixel with a 1
  - b. Look up, look left
    - If both 0
      - Increment cntr by 1
      - assign pixel P a value cnt
    - If either is 1, assign P the label of the 1
    - If both are 1
      - Note equivalence
      - Assign P's label as minimum of 2
- 2. Perform second raster scan to align equivalences

Or look for pixels below a threshold (instead of a 0) and otherwise above (instead of 1)

Double Raster Example

Copied from Robot Motion and Control
By
Spong, Hutchinson, Vidyasagar

0	0	0	0	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	X	0	0	X	X	0
0	0	0	0	X	0	0	X	X	0
0	0	0	0	X	X	X	X	X	0
0	X	X	X	X	X	X	X	X	0
0	X	X	X	X	X	X	X	X	0
0	0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	3	3	0
0	0	0	0	2	0	0	3	3	0
0	0	0	0	2	2	2	2	2	0
0	4	4	4	2	2	2	2	2	0
0	4	4	4	2	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(b)

0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	3	3	0
0	0	0	0	2	0	0	3	3	0
0	0	0	0	2	2	2	X	2	0
0	4	4	4	X	2	2	2	2	0
0	4	4	4	2	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(c)

0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	2	2	0
0	0	0	0	2	0	0	2	2	0
0	0	0	0	2	2	2	2	2	0
0	2	2	2	2	2	2	2	2	0
0	2	2	2	2	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(d)

Centroids

- Use the region filled image from above
- Compute the area of the region
  - Number of pixels with the same number value (n)
- Sum all of the x coords with the same pixel value. Do the same for y coords
- Divide each sum by n and the resulting x, y coord is the centroid