Basic Image Manipulation

Contents

- Basics: images, binary operations, filtering, edge operators
- Color, texture, segmentation
- Interest operators: detectors and descriptors

Intro

- What is computer vision?
  - The analysis of digital images by a computer

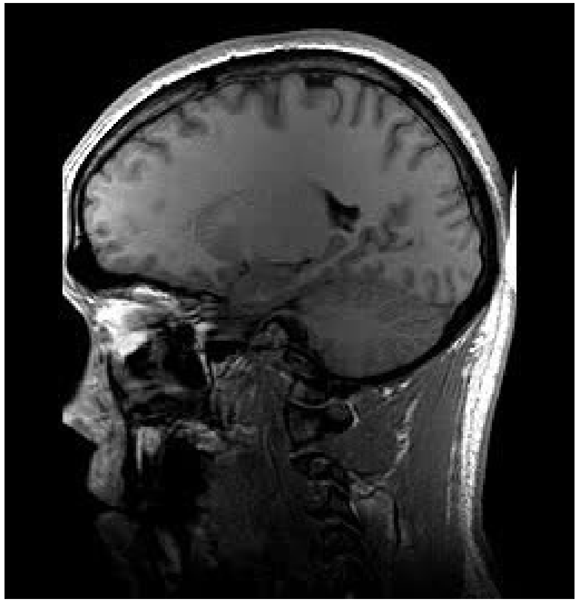
- Where do images come from?

Applications

- Medical Imaging

CT

MRI



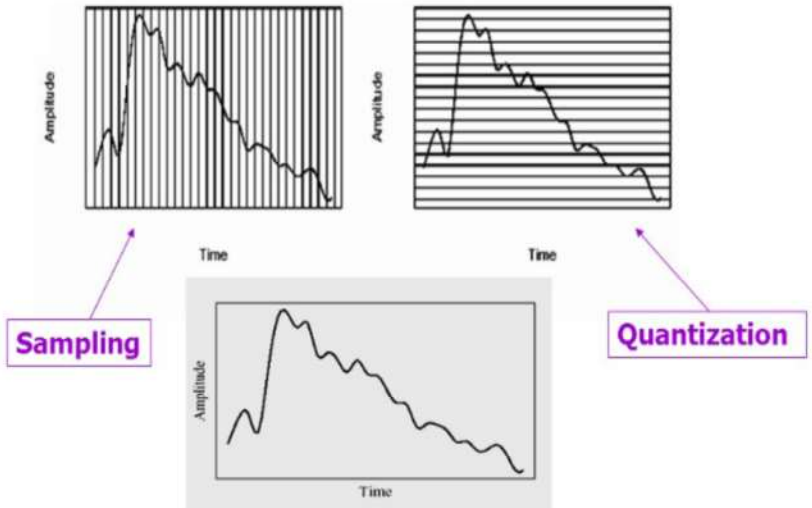
Ultrasound

Intro

Digitization = Sampling + Quantization

The process is called **digitization** and involves two other processes: **sampling** and **quantization**.

- An analog image must be converted into an array of numbers before it can be processed by a computer.



0.2826	0.3822
0.1789	0.2051
0.3344	0.3344
0.3344	0.3344

Digital Image Terminology:

0	0	0	0	1	0	0
0	0	1	1	1	0	0
0	1	95	96	94	93	92
0	0	92	93	93	92	92
0	0	93	93	94	92	93
0	1	92	93	93	93	93
0	0	94	95	95	96	95

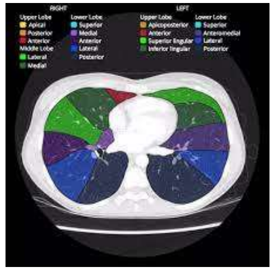
pixel with value 94

its 3x3 neighborhood

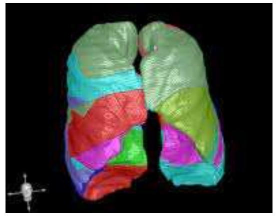
region of medium intensity

- binary image
- gray-scale (or gray-tone) image
- color image
- multi-spectral image
- range image
- labeled image

Aims of Image and Video Analysis



- Segment an image into useful regions
- Perform measurements on certain areas
- Determine what object(s) are in the scene
- Calculate the precise location(s) of objects
- Visually inspect a manufactured object
- Construct a 3D model of the imaged object
- Find "interesting" events in a video



The Three Stages of Computer Visions

- low-level
  - image -> image (processing)
- mid-level
  - image -> features
- high-level
  - features -> analysis

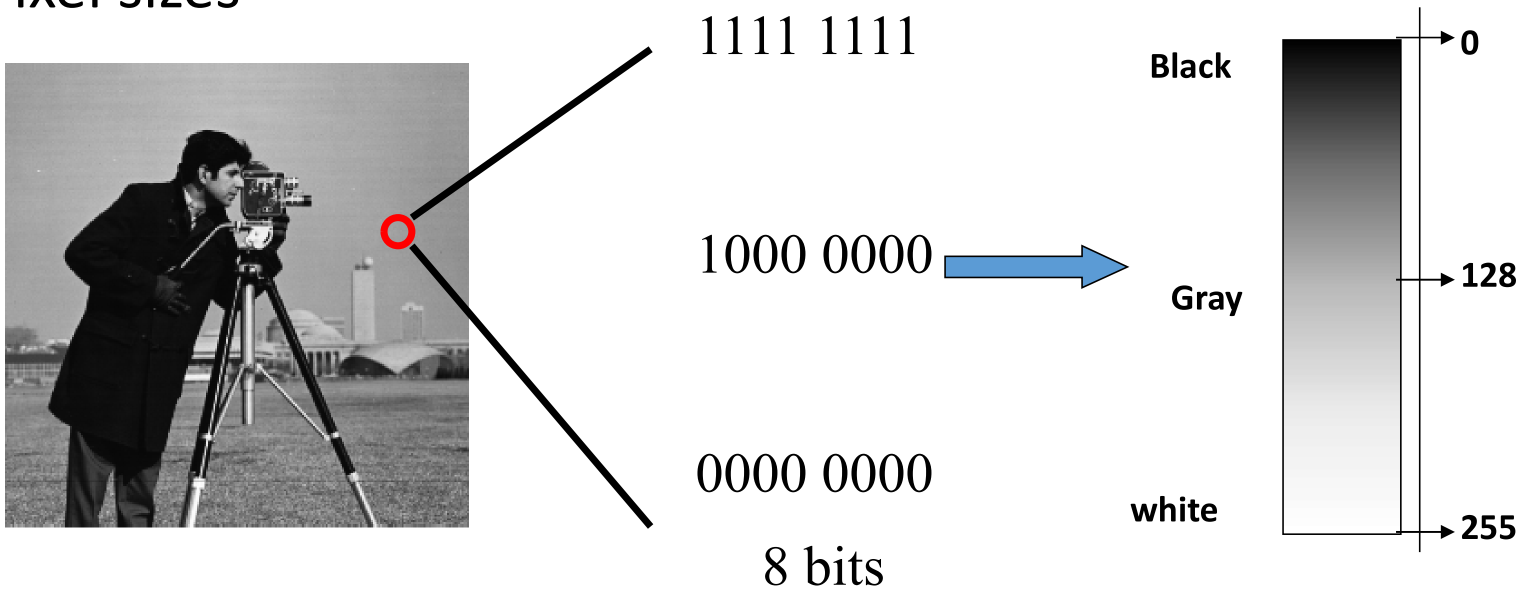
Images

Image sizes

- Physical sizes
- Pixel sizes

Photometric Resolution

- How many bits of data the display system can handle?
- How many gray-levels does the display system have?

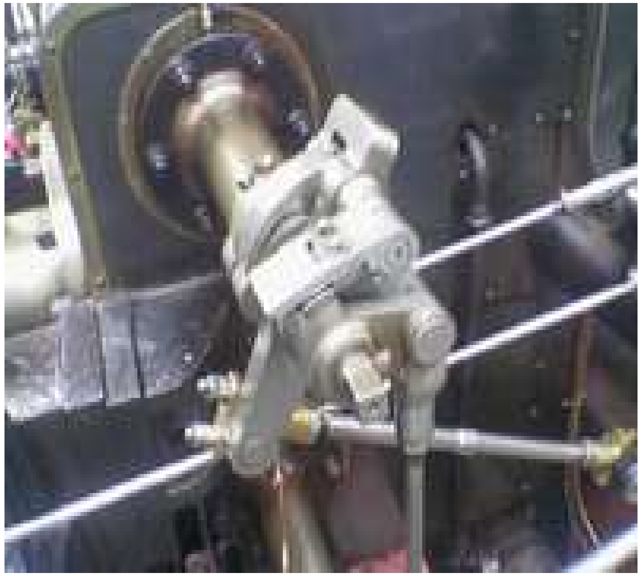


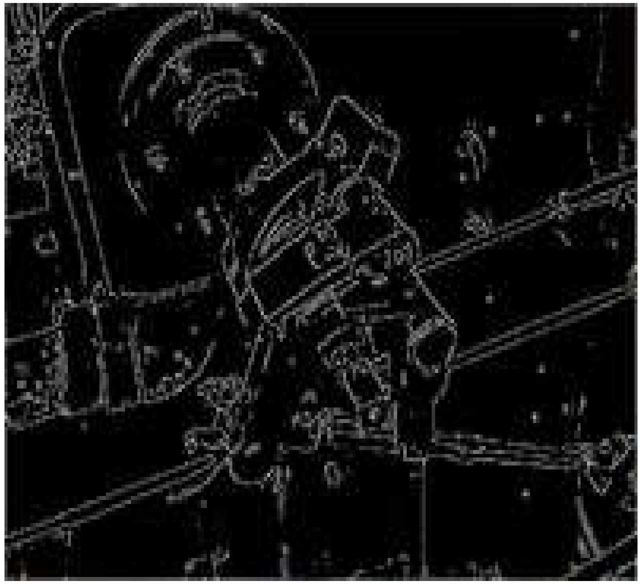
Low level

blurring

sharpening

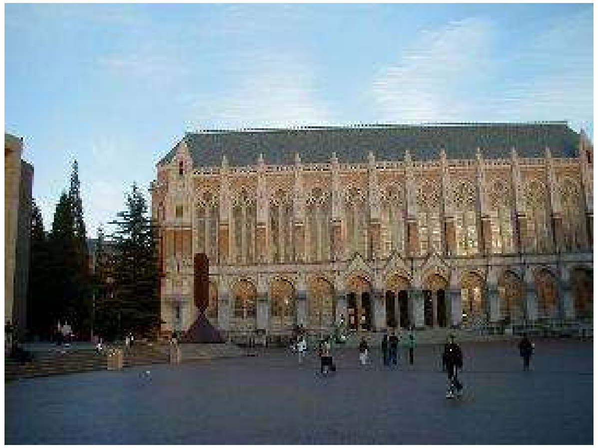
Low-level





The Canny edge detector

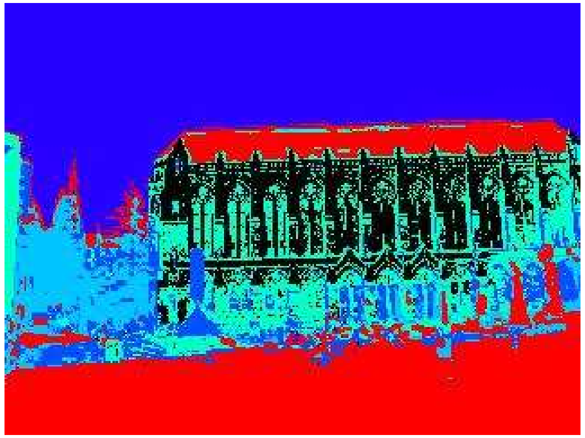
Mid-level



K-means clustering

(followed by connected component analysis)

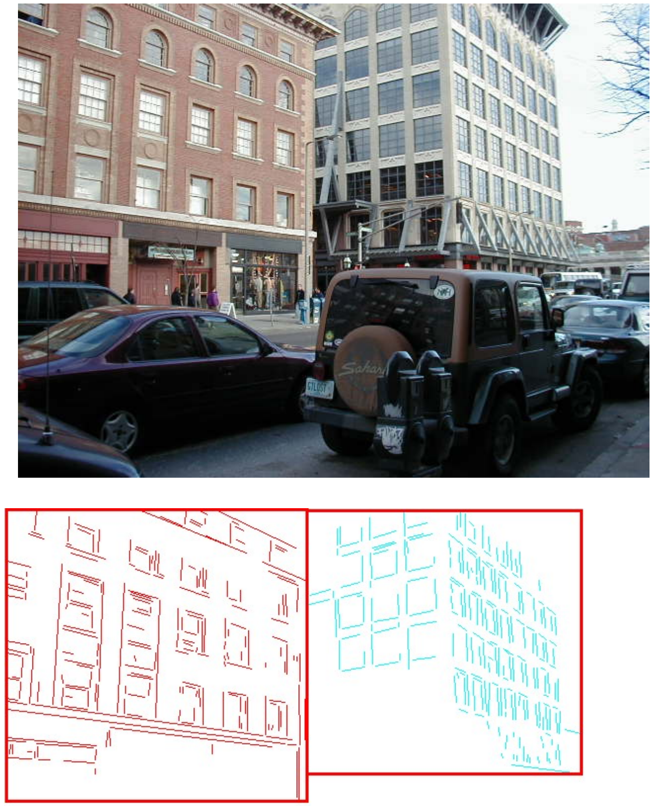
data structure

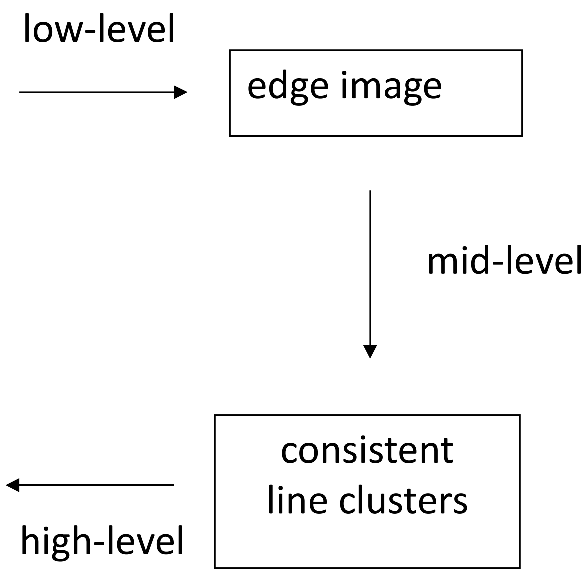


original color image

regions of homogeneous color

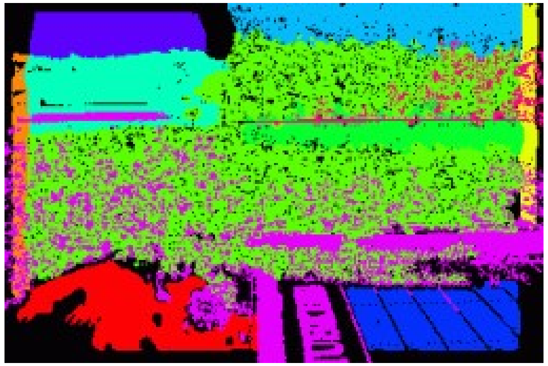
Low- to High-Level





Imaging and Image Representation

- **The 3D world has color, texture, surfaces, volumes, light sources, objects, motion, …**
- **A 2D image is a projection of a scene from a specific viewpoint.**



Images as Functions

- **gray-tone image is a function:**

  g(x,y) = val or f(row, col) = val

- **A color image is just three functions or a vector-valued function:**

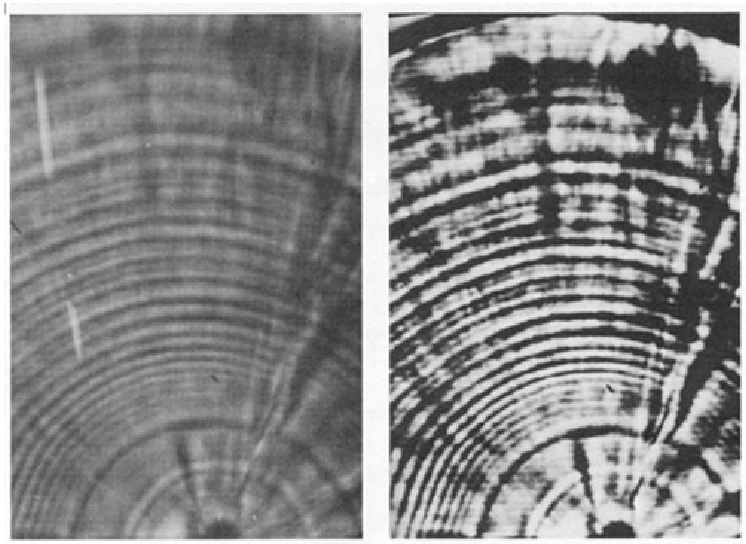
  f(row,col) =(r(row,col), g(row,col), b(row,col))

Digital Images

- Samples = pixels
- Quantization = number of bits per pixel
- Example: if we would sample and quantize standard TV picture (525 lines) by using VGA (Video Graphics Array), video controller creates matrix 640x480pixels, and each pixel is represented by 8 bit integer (256 discrete gray levels)

Image Representations

- Black and white image
  - single color plane with 2 bits
- Grey scale image
  - single color plane with 8 bits
- Color image
  - three color planes each with 8 bits
  - RGB, CMY, YIQ, etc.
- Indexed color image
  - single plane that indexes a color table
- Compressed images
  - TIFF, JPEG, BMP, etc.

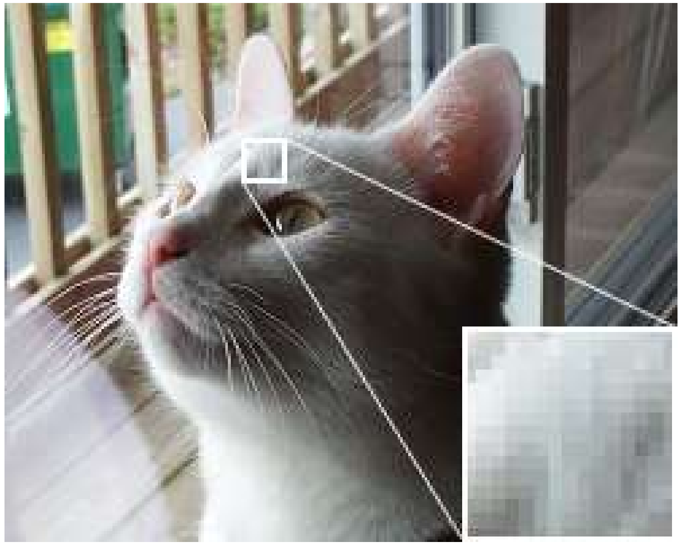


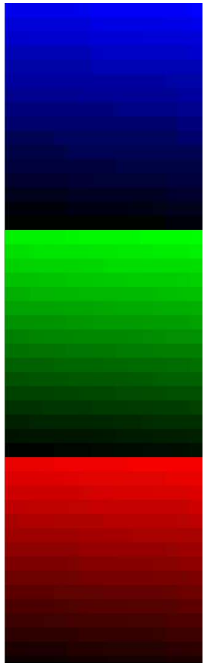
4 gray levels 2gray levels

Digital Image Representation (3 Bit Quantization)

111	111	011	011	011	011	111	111
111	011	111	111	111	111	011	111
000	111	001	111	111	001	111	000
010	111	111	111	111	111	111	010
000	111	100	111	111	100	111	000
000	111	111	100	100	111	111	000
111	000	111	111	111	111	000	111
111	111	000	000	000	000	111	111

Color Quantization
Example of 24 bit RGB Image



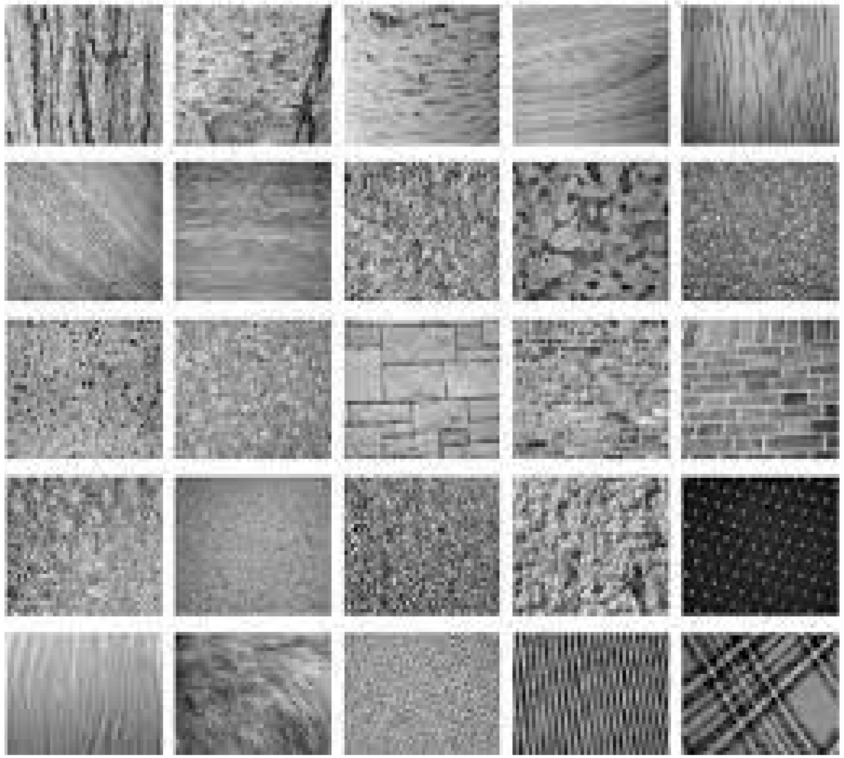


24-bit Color Monitor

Image Properties (Texture)

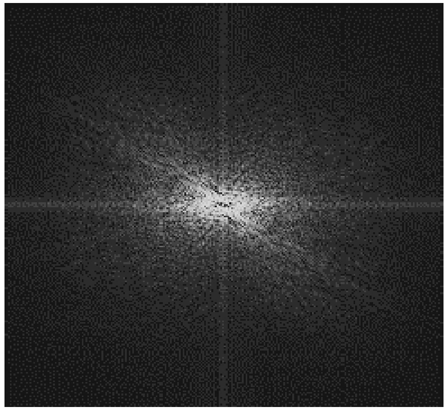
- Texture – small surface structure, either natural or artificial, regular or irregular
- Texture Examples: wood barks, knitting patterns
- Statistical texture analysis describes texture as a whole based on specific attributes: regularity, coarseness, orientation, contrast, …

Texture Examples



Spatial and Frequency Domains

- Spatial domain
  - refers to planar region of intensity values at time t
- Frequency domain
  - think of each color plane as a sinusoidal function of changing intensity values
  - refers to organizing pixels according to their changing intensity (frequency)



Edge detection

- What is edge

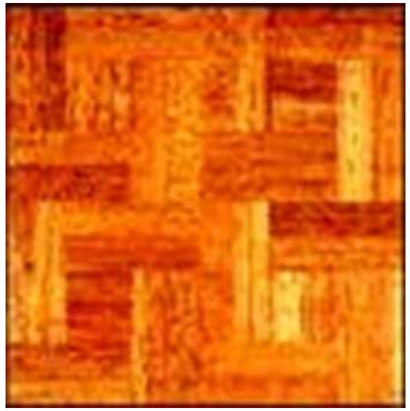
– Edge exists within objects, between objects, between object and

- background…

– Caused by varieties in color, brightness, texture, etc

color

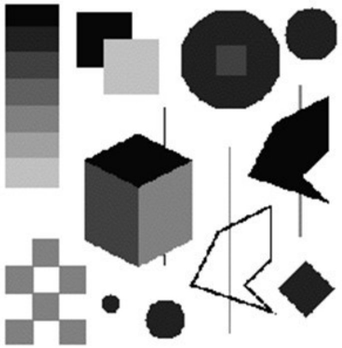
brightness



texture

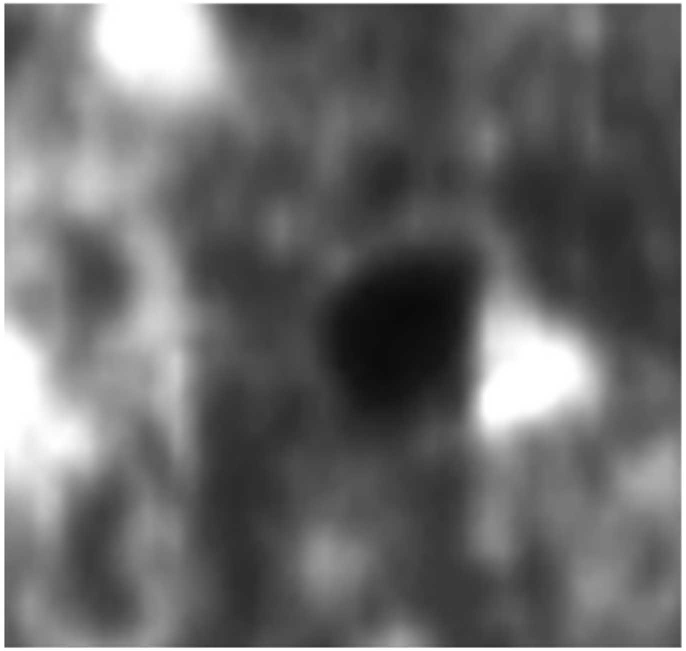
Edge detection

- What is edge detection
  - Edge is where intensity change rapidly in images
  - Convert 2D images to a set of curves

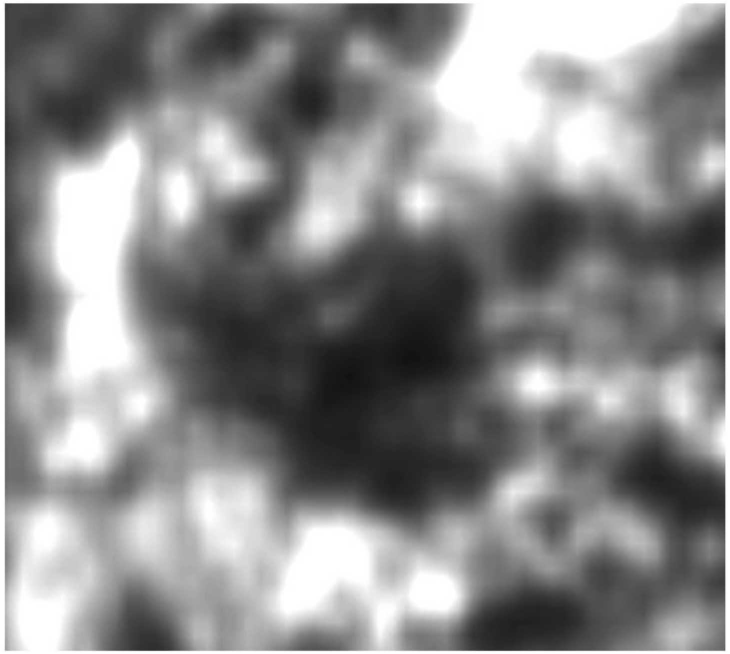


Edge detection

- Edge is important for image segmentation, morphological analysis….



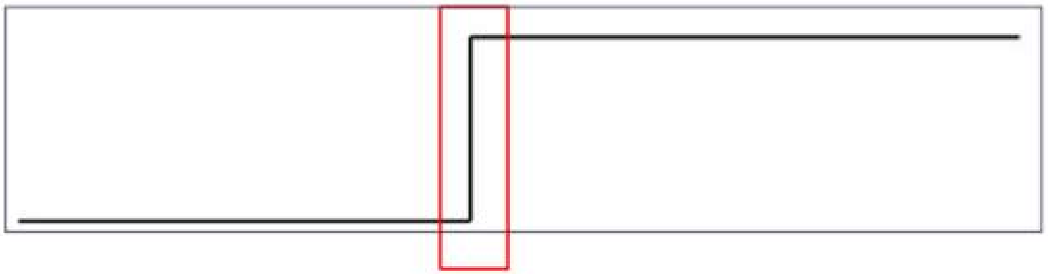
Benign



Malignant

Edge detection

- Edge is where change occurs:
  - Change is measured by derivative in 1D



Simple Edge Detection

- Example: Let assume single line of pixels

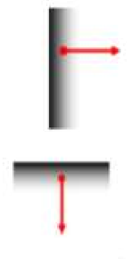
5	7	6	4	152	148	149

- Calculate 1st derivative (gradient) of the intensity of the original data
  - Using gradient, we can find peak pixels in image
  - *I(x)* represents intensity of pixel *x* and
  - *I'(x)* represents gradient (in 1D),
  - Then the gradient can be calculated by convolving the original data with a mask (-1/2 0 +1/2)
  - $I'(x) = -1/2 * I(x-1) + 0*I(x) + ½*I(x+1)$

Gradient operations

- • The gradient points in the direction of most rapid change in intensity. For image I(x,y) at location (x,y), the gradient is the vector:

$$\overline{\nabla I} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial I(x,y)}{\partial x} \\ \dfrac{\partial I(x,y)}{\partial y} \end{bmatrix}$$

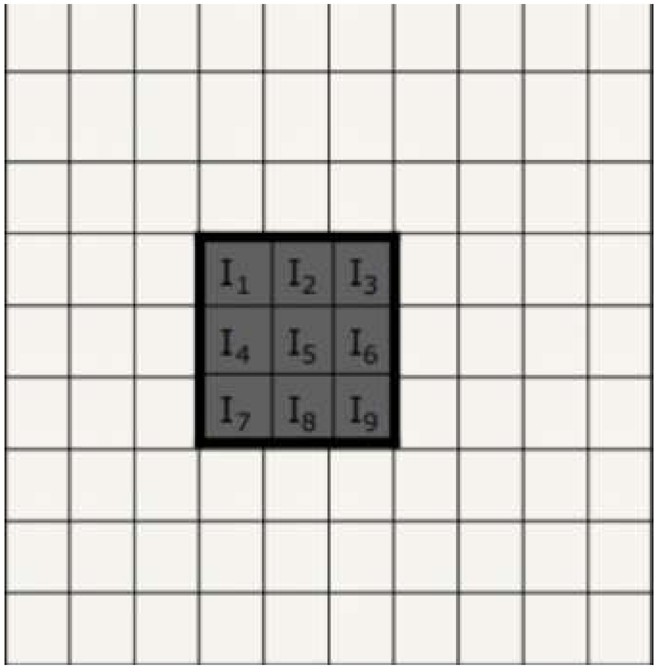


$$\theta = \tan^{-1}\left(G_y / G_x\right)$$

- The orientation of the gradient is given by
- The strength of the edge is given by magnitude of the gradient:

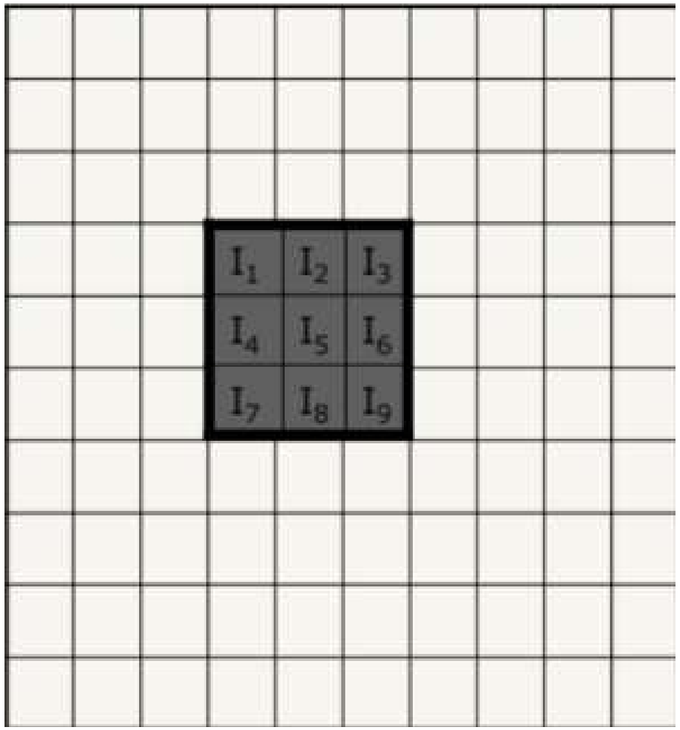
$$\nabla I = \left\|\overline{\nabla I}\right\| = \sqrt{\left[G_x^2 + G_y^2\right]}$$

Edge detection



- For each pixel the gradient can be calculated, based on a 3x3 neighborhood around this pixel.

The Prewitt edge detection



$$G_x \approx (I_3 + I_6 + I_9) - (I_1 + I_4 + I_7)$$

$$G_y \approx (I_7 + I_8 + I_9) - (I_1 + I_2 + I_3)$$

The Prewitt edge detection

$$G_x \approx (I_3 + I_6 + I_9) - (I_1 + I_4 + I_7) \qquad G_y \approx (I_7 + I_8 + I_9) - (I_1 + I_2 + I_3)$$

I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

•

kernel

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

I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

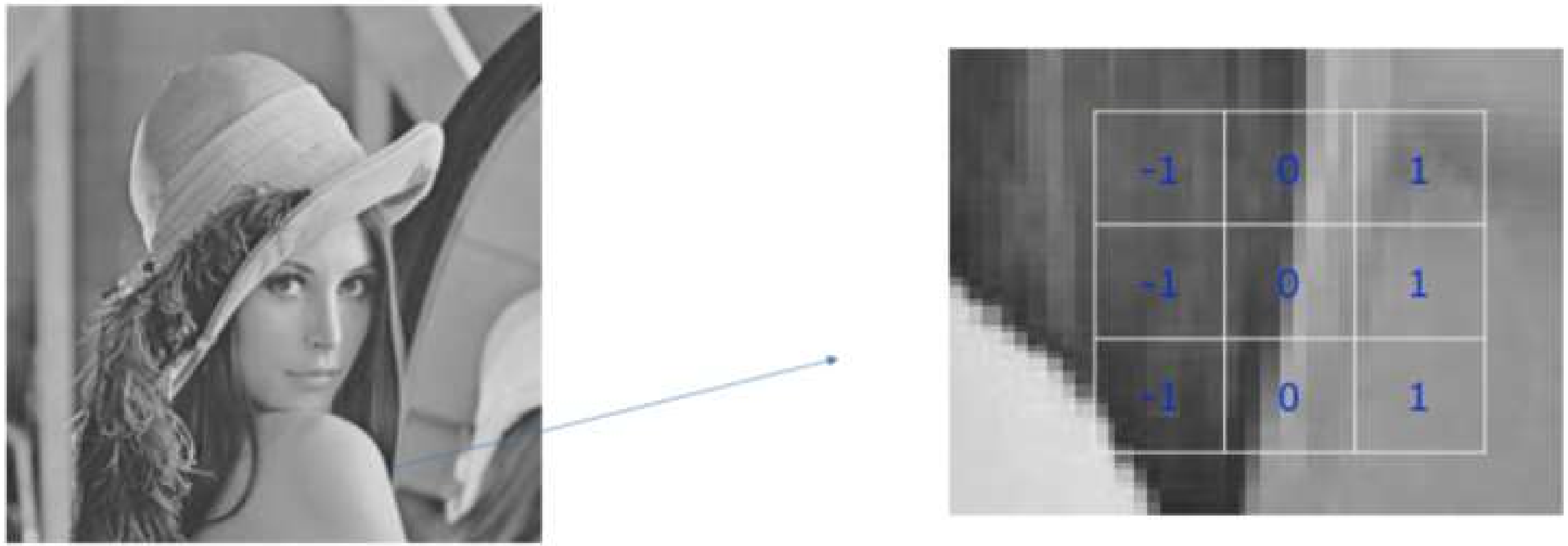
•

kernel

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

Edge Detection

The Prewitt edge detection



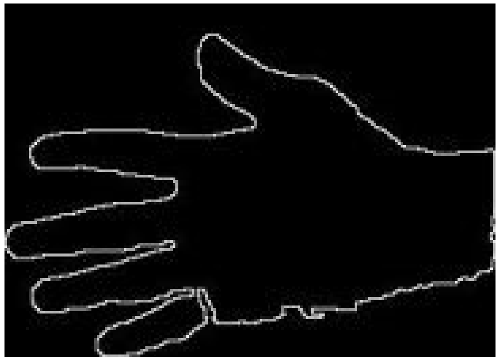
Edge Detection

Identify areas of strong intensity contrast

- filter useless data; preserve important properties

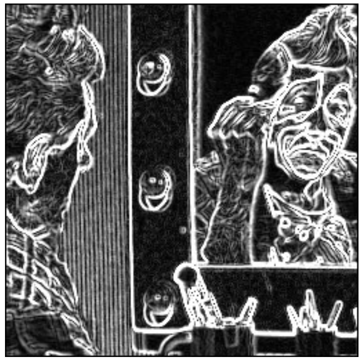
Fundamental technique

- e.g., use gestures as input
- identify shapes, match to templates, invoke commands



Mark Edge Points

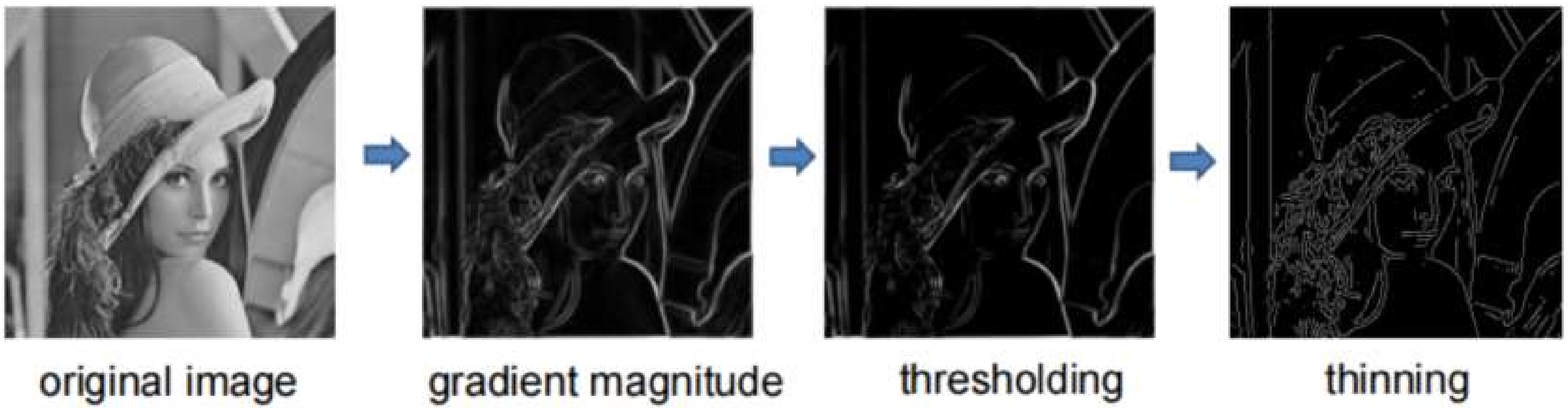
- Given gradient at each pixel and threshold
  - mark pixels where *gradient > threshold* as edges



Edge detection

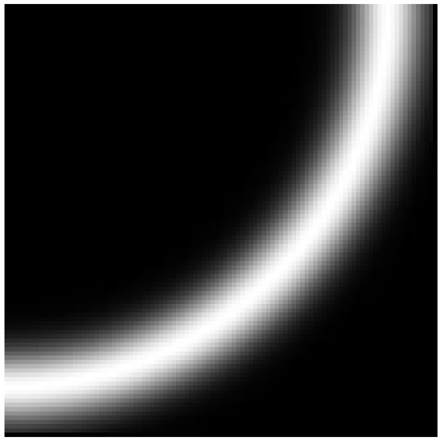
- Steps
  - Calculate gradient magnitude
  - Thresholding
  - Thinning (non-maximum suppression)

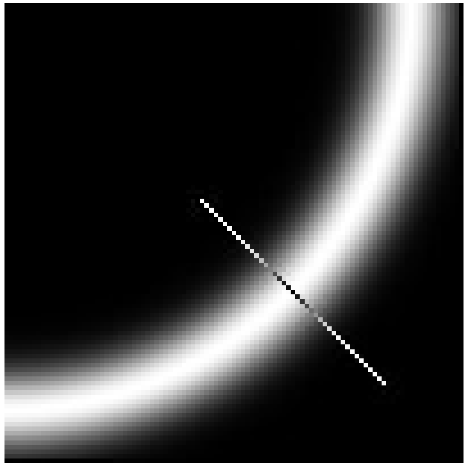
Different edge detectors:
- **Sobel edge detector**
- **Prewitt edge detector**
- **Laplacian edge detector**
- **Canny edge detector**



Edge detection

- Thinning
- Find the local maxima of the gradient magnitude
- All values along the direction of the gradient that are not peak values of a ridge are suppressed





The Prewitt edge detection

- Calculate edge for each pixel in an image

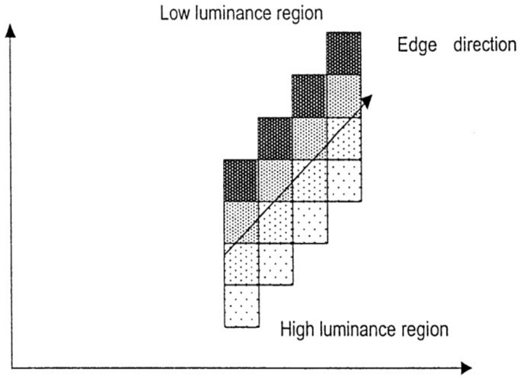
Other operators exist:

Basic Method of Edge Detection

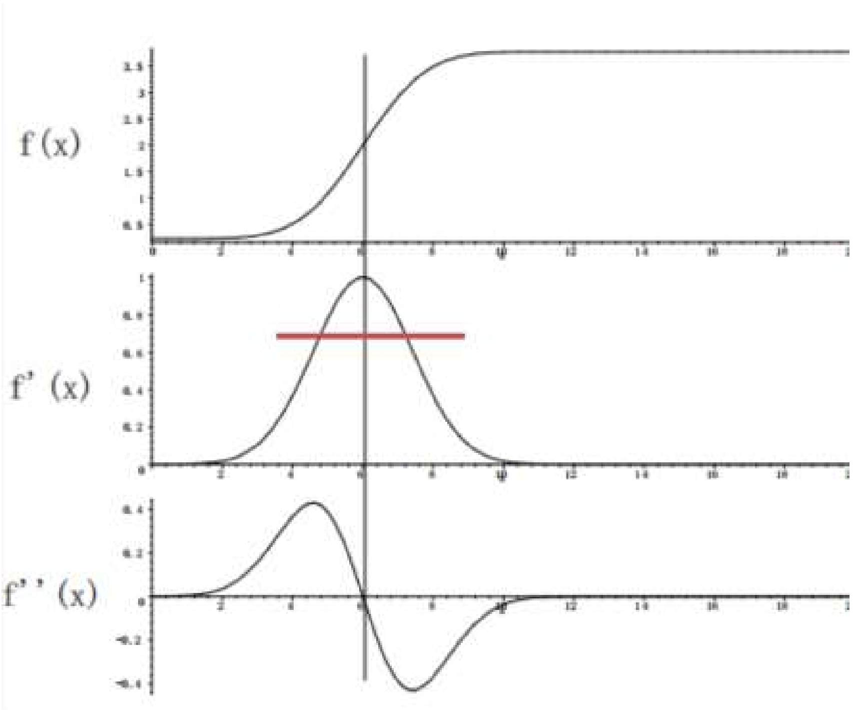
- Step 1: filter noise using mean filter
- Step 2: compute spatial gradient
- Step 3: mark points > *threshold* as edges

Compute Edge Direction

- Calculation of Rate of Change in Intensity Gradient
- Use 2nd derivative
- Example: **(5 7 6 4 152 148 149)**
- Use convolution mask (+1 -2 +1)
- I''(x) = 1*I(x-1) -2*I(x) + 1*I(x+1)
- **Peak detection in 2nd derivative is a method for line detection**.



Edge detection



First order derivative： need to set a threshold; sensitive to noise; produce thicker edges

Second order derivative: compute zero-crossing

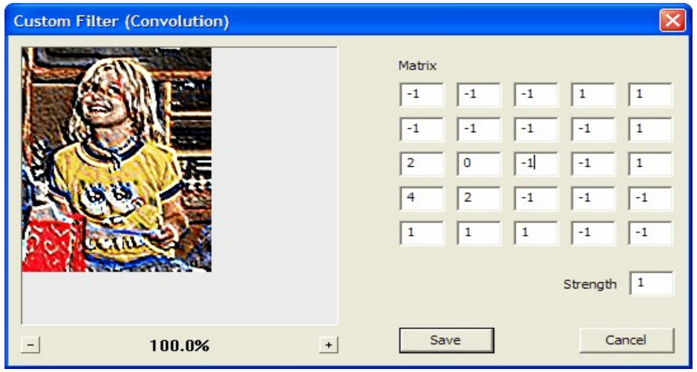
Summery

- Edge definition
- Edge extraction
  - Gradient, gradient orientation, gradient magnitude
  - First order operators (kernels)
  - Thresholding, thinning
  - Second order derivative

Image Processing Function: Filtering

- Filter an image by replacing each pixel in the source with a weighted sum of its neighbors
- Define the filter using a *convolution mask*, also referred to as a *kernel*
  - non-zero values in small neighborhood, typically centered around a central pixel
  - generally have odd number of rows/columns

Convolution Filter



100	100	100	100	100
100	100	50	50	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100

X

	0	1	0	
	0	0	0	
	0	0	0	

=

100	100	100	100	100
100	100	50	50	100
100	100	50	100	100
100	100	100	100	100
100	100	100	100	100

Mean Filter

20	12	14	23
45	15	19	33
55	34	81	22
8	64	49	95

Subset of image

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convolution filter

Common 3x3 Filters

Low/High pass filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Blur operator

$$\frac{1}{13}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

H/V Edge detector

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Mean Filter

20	12	14	23
45	15	19	33
55	34	81	22
8	64	49	95

Subset of image

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Convolution filter

Example

Exercise

- https://rubikscode.net/2022/05/16/introduction-to-opencv-and-image-processing-with-python/

- https://learnopencv.com/edge-detection-using-opencv/

Future

- Other Important Image Processing Functions
  - Image segmentation
  - Image recognition
    - Formatting
    - Conditioning
    - Marking
    - Grouping
    - Extraction
    - Matching
  - Image synthesis