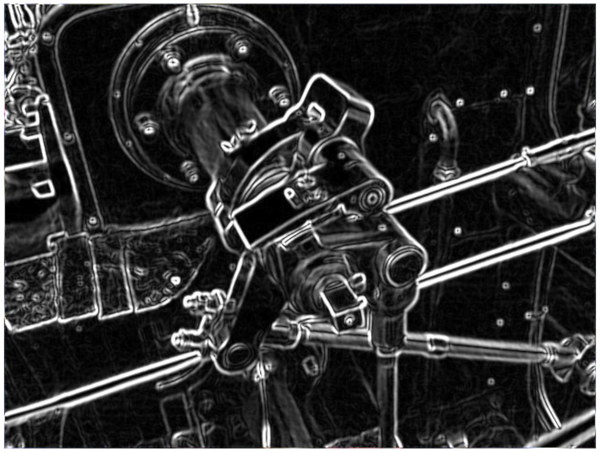
Filters

Computer Vision and Pattern Recognition

CS 4243

S1-Y2023/24

National University of Singapore | School of Computing



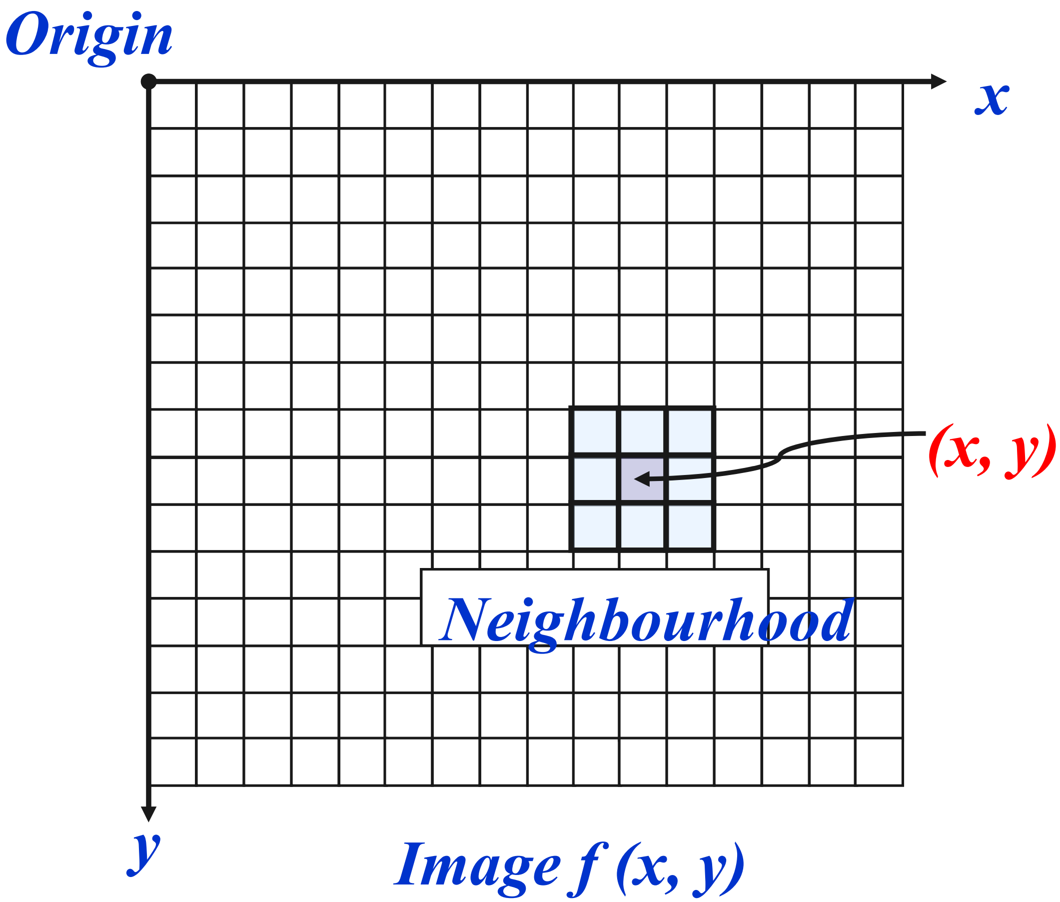
Lesson 2

Local Operations, Filtering, and Convolution

- Where we see how we can filter and image
- Every system in this big universe is a filter

Local Operations

- An image in the spatial domain
- A local operator, like T, uses the pixel and its neighbors to compute the new value of that pixel.
- E.g., replace each pixel f(x,y), with the average of that and its 8 neighbors.
- $g(x,y)=T[f(x\pm\Delta x, y \pm\Delta y)]$



Filters

- **Lowpass**
  - Passes the low spatial frequency components of the image and filters the high-frequency ones.
  - a kind of averaging op
- **Highpass**
  - Passes the high spatial frequency components of the image and filters the low-frequency ones.
  - a kind of derivative, difference op
- **Bandpass**
  - Passes spatial frequency components between 2 frequencies
- **Bandreject**
  - filters spatial frequency components between 2 frequencies

Lowpass Filters

- Averaging tends to be lowpass filtering
- Q: Why do we need those coefficients?
- Make the image less noisy but blur

$$\boldsymbol{h_{l1}} = {}^{1}\!/_{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$$

$$\boldsymbol{h_{l2}} = {}^{1}\!/_{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\boldsymbol{h_{l3}} = {}^{1}\!/_{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

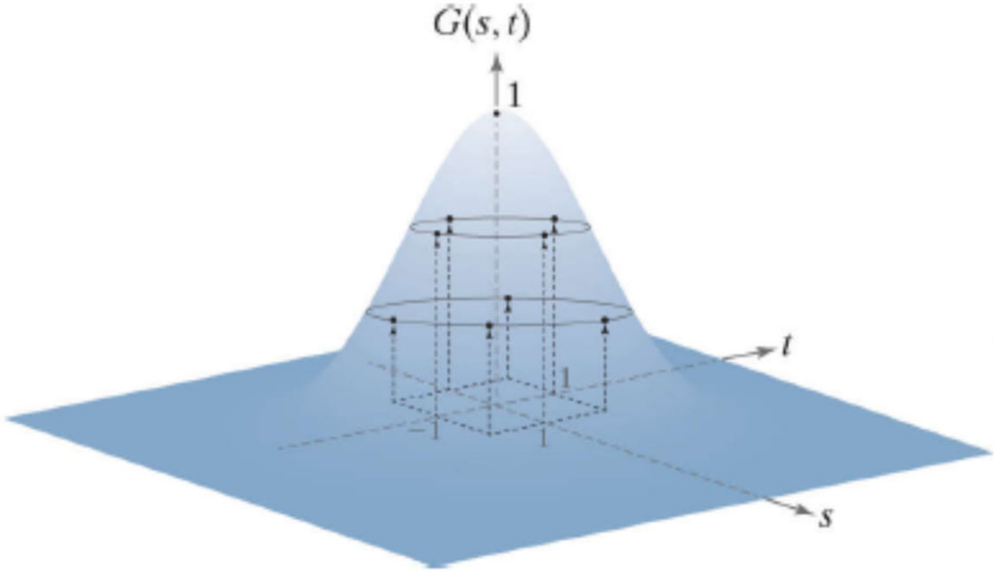
Gaussian Filter

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Mean is 0 and Std is σ

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

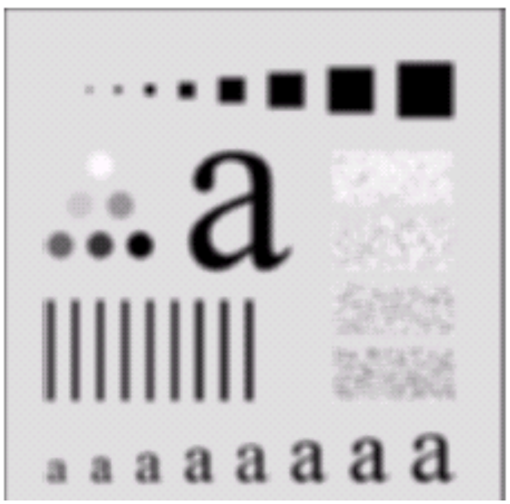
3x3: $\sigma^2 = 1$

Lowpass Filters

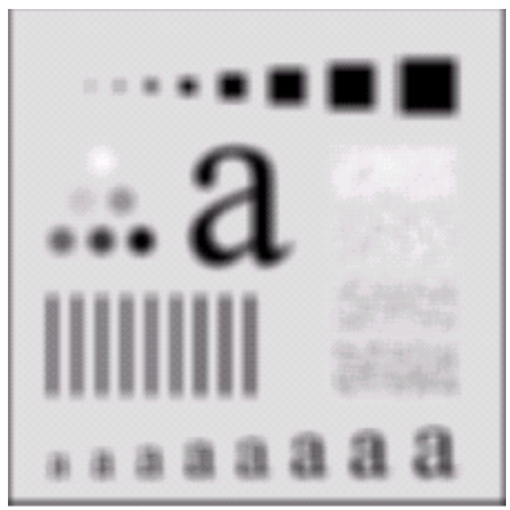
Original image 500x500

filtered with 3x3

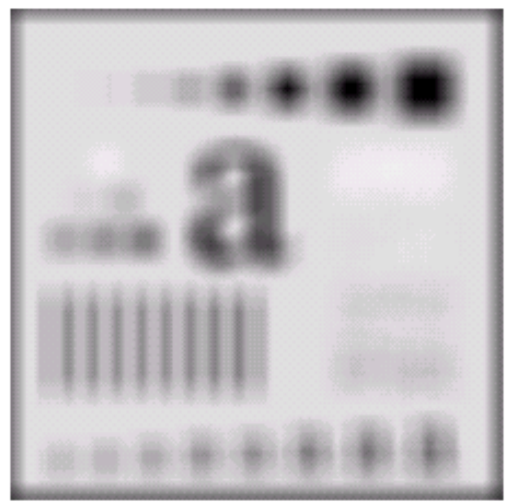
filtered with 5x5



Filtered with 9x9



filtered with 15x15



filtered with 35x35

High Pass Filters

- Typically, they tend to be a gradient/ differentiation operator
- Sign-changing can be seen in the filter matrix
- Could be used to detect lines and edges in different directions or symmetrically.
- Make the image sharper
- Can be based on the 1st or 2nd order derivative.

High Pass Filters

$$\boldsymbol{h_{h1}} = \begin{bmatrix} \mathbf{1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{h_{h2}} = \begin{bmatrix} \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{8} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \end{bmatrix}$$

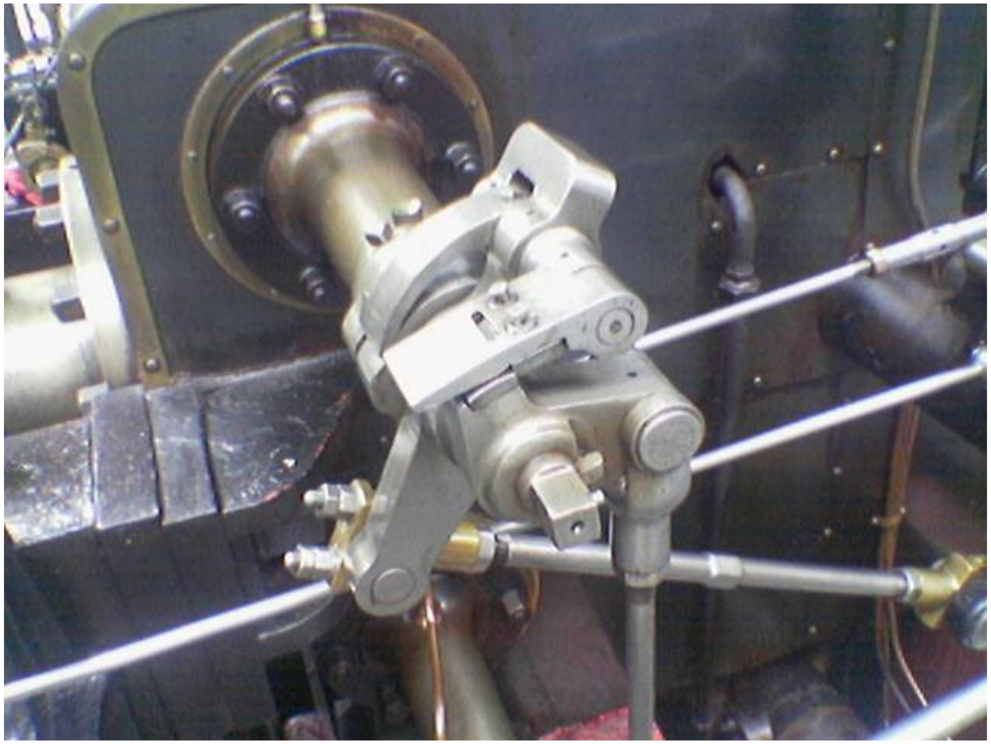
$$\boldsymbol{h_{h3}} = \begin{bmatrix} \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{9} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{-1} & \mathbf{-1} \end{bmatrix}$$

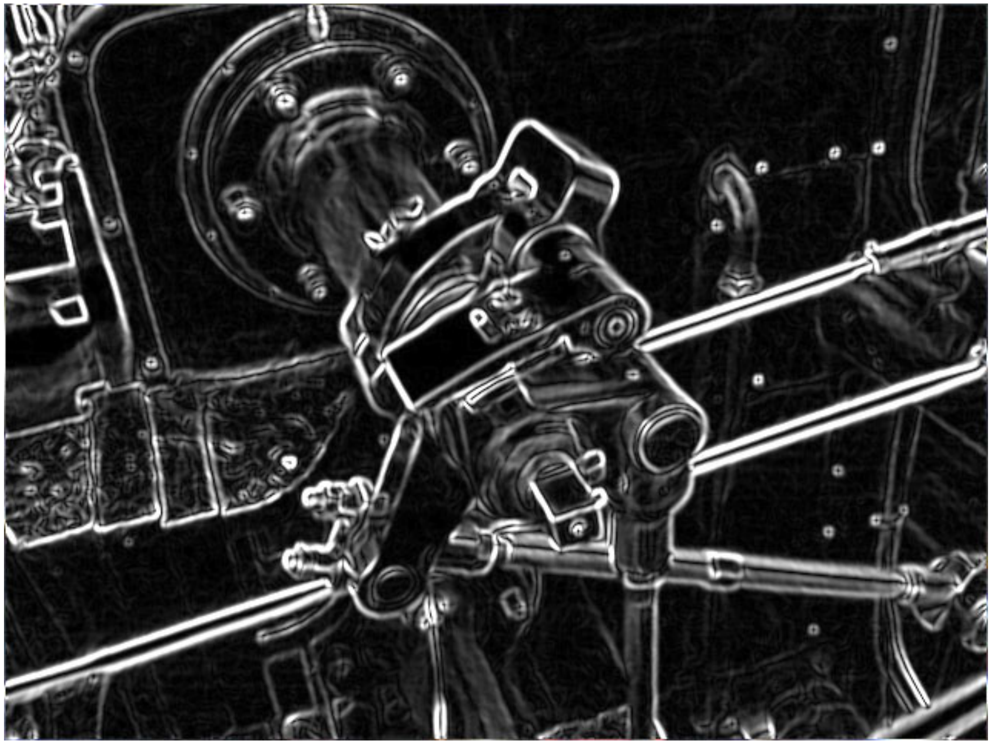
$$\boldsymbol{h_{h4}} = \begin{bmatrix} \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{-1} & \mathbf{4} & \mathbf{-1} \\ \mathbf{0} & \mathbf{-1} & \mathbf{0} \end{bmatrix}$$

High Pass Filters

Guernica and filtered Guernica with hh2

High Pass Filters





An image and result of edge detection using the Sobel algorithm

1^{st} Derivative

- **Discrete 1^{st} derivative / gradient:**

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- **Actually, it computes the differences between neighbor samples**
- **And result is:**

2^{nd} Derivative

- **Discrete 2^{nd} derivative:**

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

- **In fact, it is the derivative of the 1^{st} derivative**
- **And result is:**

The Laplacian

- **Definition:**

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

- **X derivation:**

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

- **Y derivation:**

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The Laplacian (cont…)

Finally, the Laplacian operator can be rewritten as:

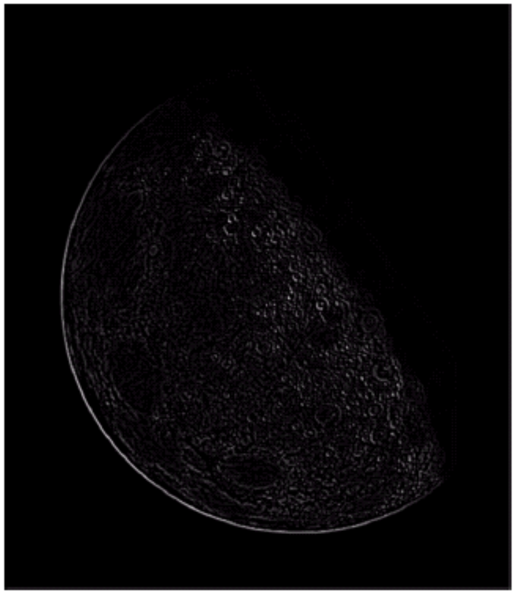
$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

And a Laplacian filter simply is:

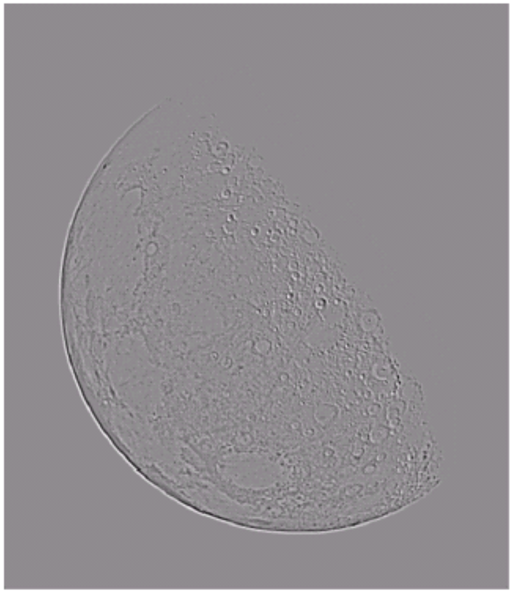
0	1	0
1	-4	1
0	1	0

The Laplacian example

Original
Image



Laplacian
Filtered Image



Laplacian
Filtered Image
Scaled for Display

Sobel Operator

- Irwin Sobel and Gary Feldman, colleagues at the Stanford Artificial Intelligence Laboratory (SAIL).
- Based on the previous equations we can derive the *Sobel Operators*
- We use these 2 horizontal and vertical edge detectors, **hsx** and **hsy**, and convolve our image with both and obtain 2 detail images **Gx** and **Gy**.

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

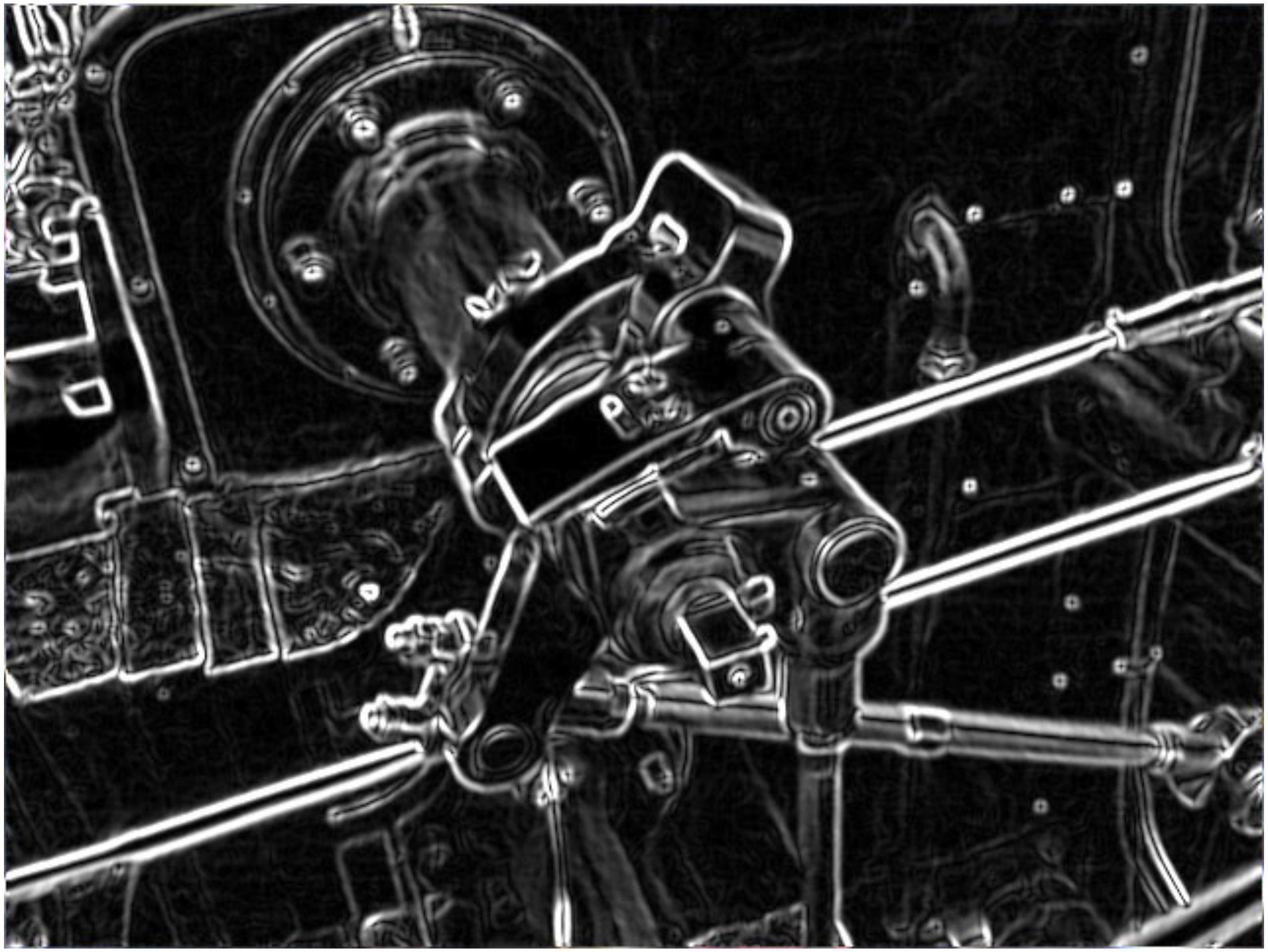
Sobel Operator

- $G_x = a * h_{sx}$
- $G_y = a * h_{sy}$
- Then we obtain the intensity (or edges strength map), and the edge direction map.
- $|G| = \sqrt{G_x^2 + G_y^2}$
- Some times $|G| = |G_x| + |G_y|$
- $< G = \mathrm{atan}(\frac{G_y}{G_x})$
- Then we apply a thresholding on |G|

Sobel Operator

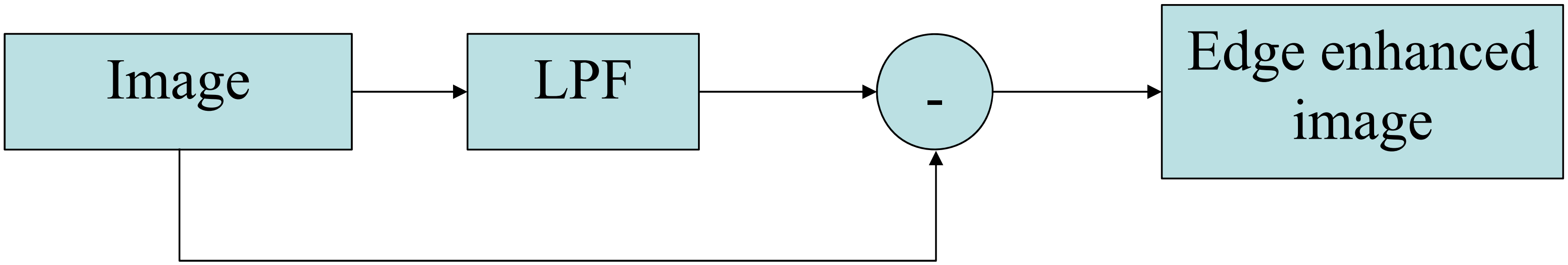
- $|\boldsymbol{G}| = \begin{cases} \mathbf{1} & |\boldsymbol{G}(\boldsymbol{i},\boldsymbol{j})| \geq \boldsymbol{T} \\ \mathbf{0} & \boldsymbol{else} \end{cases}$
- T is the threshold
- Many things can be done with |G| and <G, e.g.:
  - Finding the strongest edges
  - And their directions
  - Border tracking
  - And vectorization

Sobel Operator



Unsharp Masking

- A lowpass filter, filters the edges and lines from the image.
- So, what if we subtract a lowpass filtered version of an image from the original one?
- This is **unsharp masking**
- A reliable edge enhancement algorithm



Median Filter

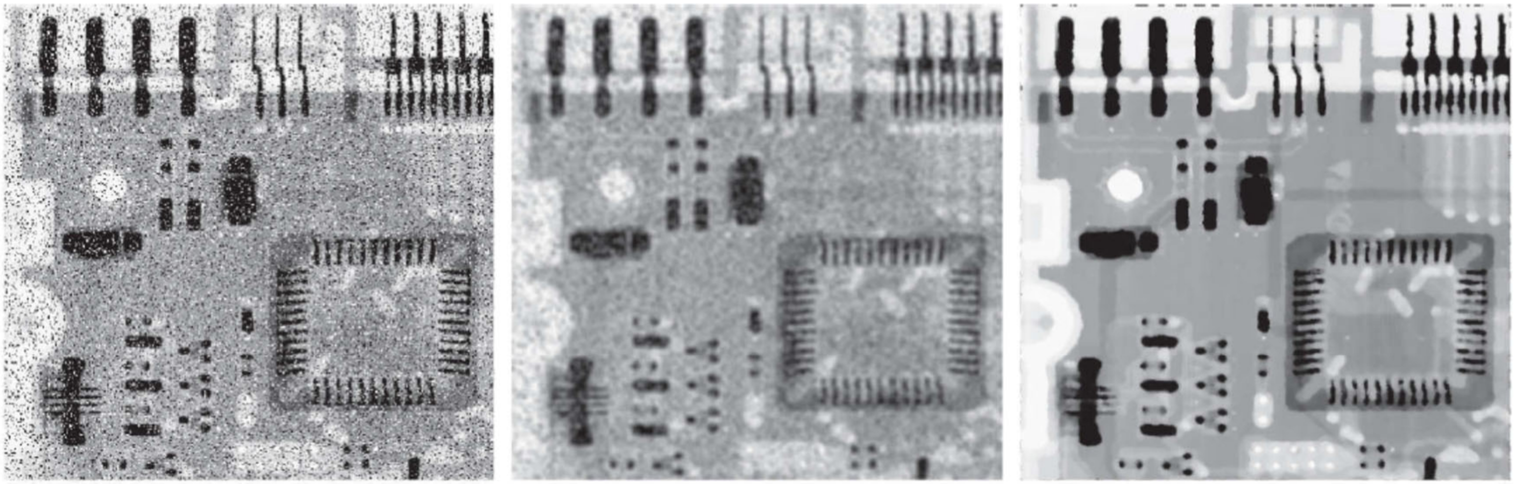
1- all over your image

2- in a small *nxn* patch/neighborhood

3- sort the pixels and find the median, *M*

4- replace the central pixel of your patch with *M*

5- slide one pixel to the next neighborhood ;

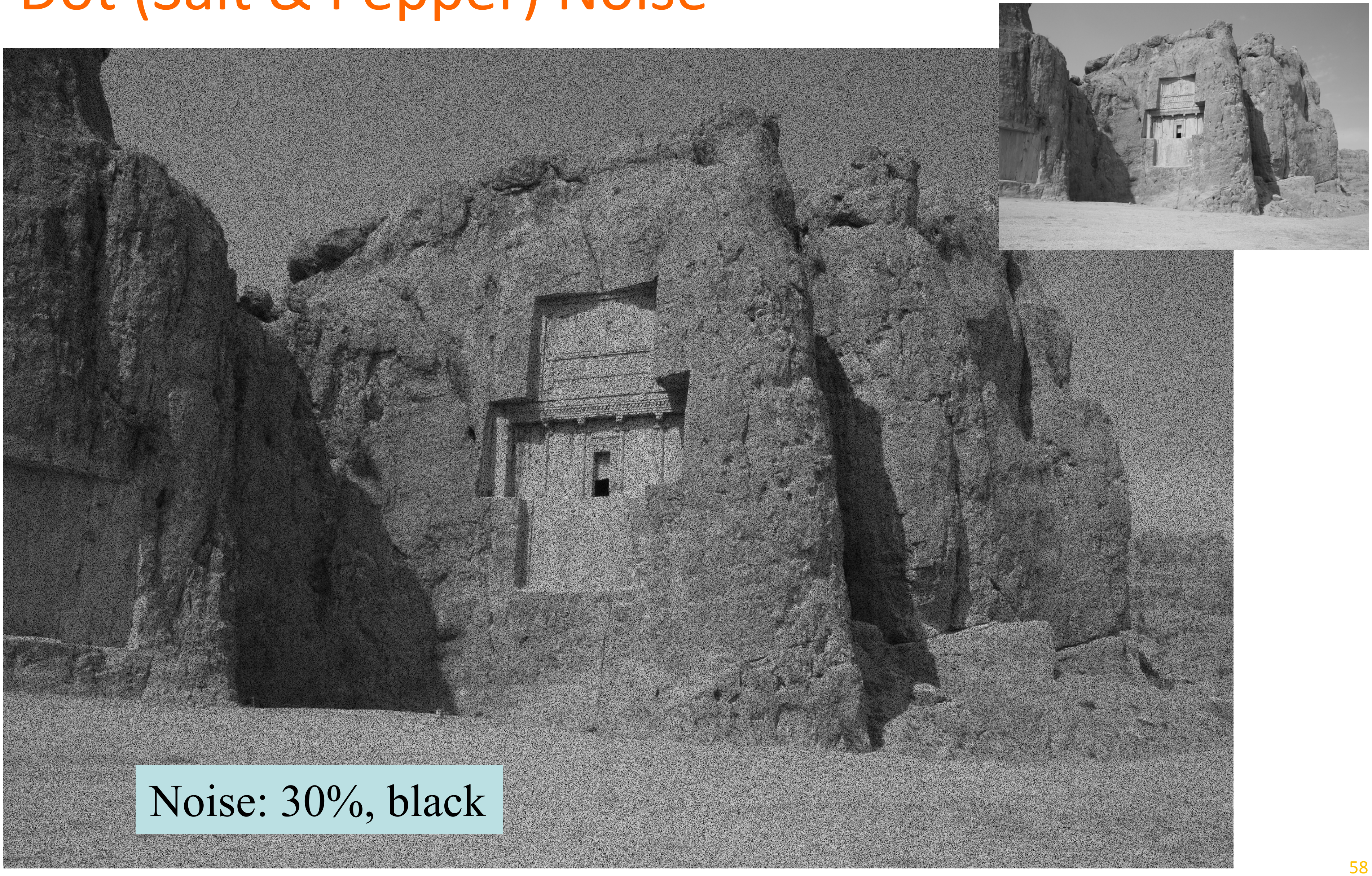


a b c

FIGURE 3.49

(a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Dot (Salt & Pepper) Noise



Saltpepper.ipynb , pic679_gl.jpg , noisy.jpg

Noisy Image and Median Filter

Salt’n Pepper noise 5%, Median with window size=5x5

Noise

- In signal processing, **noise** is a general term for unwanted (and, in general, unknown) modifications that a signal may suffer during capture, storage, transmission, processing, or conversion.
- **White noise:** Is a random signal having equal intensity at different frequencies, giving it a constant power spectral density.
- The presence of noise in an image might be **additive** or **multiplicative**. In the Additive Noise Model, an additive noise signal is added to the original signal to produce a corrupted noisy signal that follows the following rule: $w(x, y) = s(x,y) + n(x,y)$
- **Dot noise:** See the figures. Mostly result of the multiplication of a [0,1] or [1,255] matrix with your image. Could be due to channel disconnection or saturation.

Noise

- **SNR: Signal to Noise Ratio**, is used in imaging to characterize image quality. The sensitivity of a (digital or film) imaging system is typically described in the terms of the signal level that yields a threshold level of SNR.
- $SNR_{dB}(a) = 10log_{10}(\frac{power(a)}{power(n)})$
- Typically and usually, power(n)=$\sigma^2(n)$
- In simulations, noise is a matrix of random numbers. For instance, if we generate a matrix of normal distribution random numbers, power of noise is the variance.