Today

Early vision on a single image

Reminder

Project 1 is due 11:59pm EST, Sunday, Feb 16th, 2025.

Requirement: you need to submit your project as **a SINGLE zipped file named as Lastname-Firstname-Proj1** through **Blackboard** including

- A written project report includes a brief introduction on the addressed problem, a succinct description on the methods you implemented with the major steps, the experimental results and analysis, conclusion, and reference.

- Code with appropriate comments.

Your report must be well organized and be easy to follow. There should be no spelling and grammar errors!

Early Vision on One Image

So far, we talked about image formation. Next, we will discuss early vision on one image

- **Linear and nonlinear filters**
  - Linear and nonlinear filters for noise reduction
  - Linear filters for differentiation
- **Edge detection**
- **Features**
  - Edges, lines, curves, corners etc.

Recall: Early Vision on One Image: Two Important Topics

- **Image noise: intrinsic property of the sensor (CCD) and independent of scene**
  - Intensity noise – quantization and sensor
  - Positional noise – spatial sampling

- **Features: characterizing the shape/appearance of the objects in the image**
  - Edges, lines, curves, corners etc.
  - Image statistics: mean, variance, histogram etc.
  - Complex features
  - Widely used in many problems of computer vision including camera calibration, stereo, object detection/tracking/recognition, etc.

Properties of Noise

- **Spatial properties**
  - Spatially periodic noise
  - Spatially independent noise
- **Frequency properties**
  - White noise – noise containing all frequencies within a bandwidth

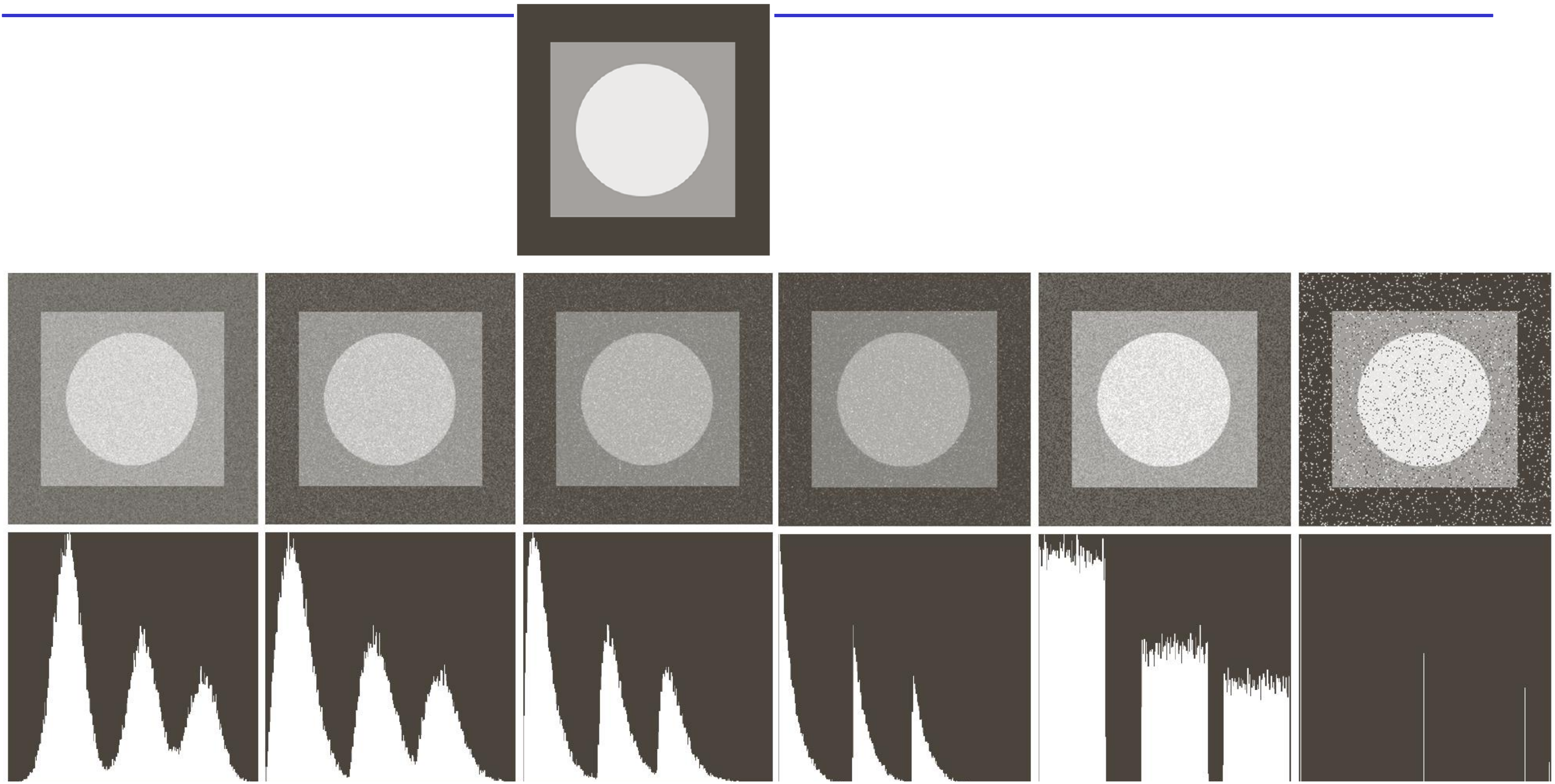
Image Noise

Observed image intensity   Ideal image intensity

$$\hat{I}(x,y) = I(x,y) + \eta(x,y)$$

Image noise

Recall: An Example



Gaussian Rayleigh Gamma Exponential Uniform Salt & Pepper

Figure from "Digital Image Processing", Gonzalez and Woods

Recall: Estimation of Noise Parameters

Take a small stripe of the background, do statistics for the mean and variance.

Figure from "Digital Image Processing", Gonzalez and Woods

Periodical Noise

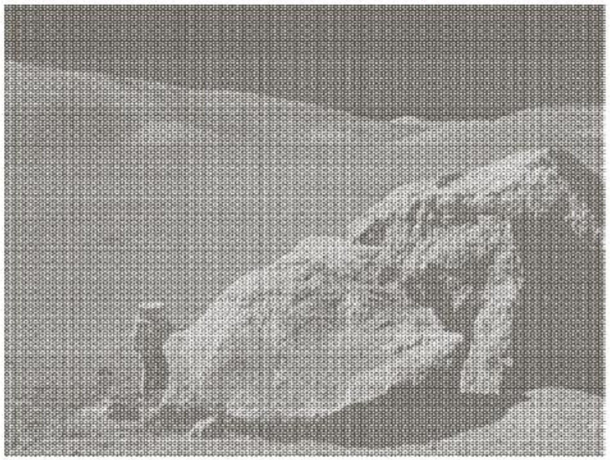


Image is corrupted by a set of sinusoidal noise of different frequencies

Figure from “Digital Image Processing”, Gonzalez and Woods

Image Denoising by Filtering

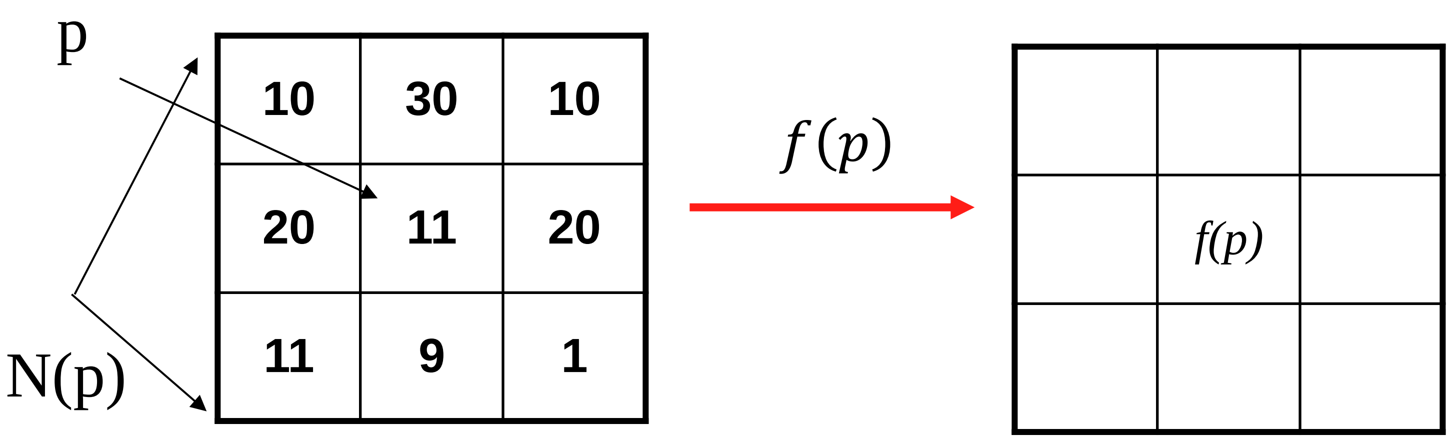
Gaussian Noise

Averaging

Gaussian Smoothing

Image Filtering

Modifying the pixels in an image based on some function of a local neighborhood of the pixels



$f(p)$:

- Linear function
  - Correlation
  - Convolution
- Nonlinear function
  - Order statistic (median)

Linear Filters

General process:

- Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.

Properties

- Output is a linear function of the input
- Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)

Example: smoothing by averaging

- form the average of pixels in a neighborhood

Example: smoothing with a Gaussian

- form a weighted average of pixels in a neighborhood

Example: finding an edge

Linear Filtering - Correlation

The output is the linear combination of the neighborhood pixels

$$f(p) = \sum_{q_i \in N(p)} a_i q_i$$

The coefficients of this linear combination combine to form the "filter-kernel"

1	3	0
2	10	2
4	1	1

Image

$\otimes$

1	0	-1
1	0.1	-1
1	0	-1

Kernel

$=$

	5	

Filter Output

Linear Filtering - Convolution

$$f(i,j) = I * H = \sum_{k}\sum_{l} I(k,l)H(i-k,j-l)$$

$I = \text{Image}$
$H = \text{Kernel}$

H

H_1	H_2	H_3
H_4	H_5	H_6
H_7	H_8	H_9

$X - flip$

H_7	H_8	H_9
H_4	H_5	H_6
H_1	H_2	H_3

$Y - flip$

I

I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

$\circledast$

H_9	H_8	H_7
H_6	H_5	H_4
H_3	H_2	H_1

$$\begin{aligned} I * H = {} & I_1H_9 + I_2H_8 + I_3H_7 \\ & + I_4H_6 + I_5H_5 + I_6H_4 \\ & + I_7H_3 + I_8H_2 + I_9H_1 \end{aligned}$$

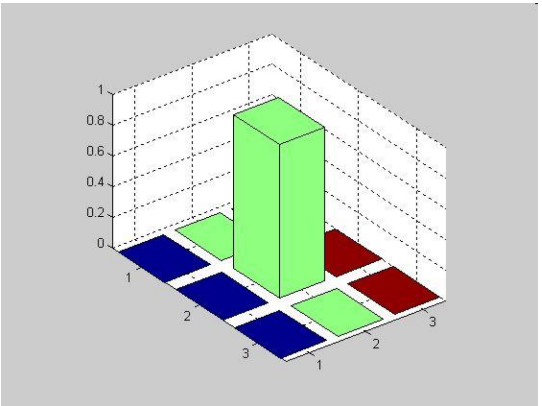
Linear Filtering - Convolution

$$f(i,j) = I * H = \sum_{k} \sum_{l} I(k,l) H(i-k, j-l)$$

$$\underset{\text{Image}}{\begin{array}{|c|c|c|}\hline 1 & 3 & 0 \\ \hline 2 & 10 & 2 \\ \hline 4 & 1 & 1 \\ \hline \end{array}} \circledast \underset{\text{Kernel}}{\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0.1 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array}} = \underset{\text{Filter Output}}{\begin{array}{|c|c|c|}\hline & & \\ \hline & -3 & \\ \hline & & \\ \hline \end{array}}$$

$$\underset{\text{Image}}{\begin{array}{|c|c|c|}\hline 1 & 3 & 0 \\ \hline 2 & 10 & 2 \\ \hline 4 & 1 & 1 \\ \hline \end{array}} \otimes \underset{\text{Kernel}}{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0.1 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}} = \underset{\text{Filter Output}}{\begin{array}{|c|c|c|}\hline & & \\ \hline & 5 & \\ \hline & & \\ \hline \end{array}}$$

Linear Filtering

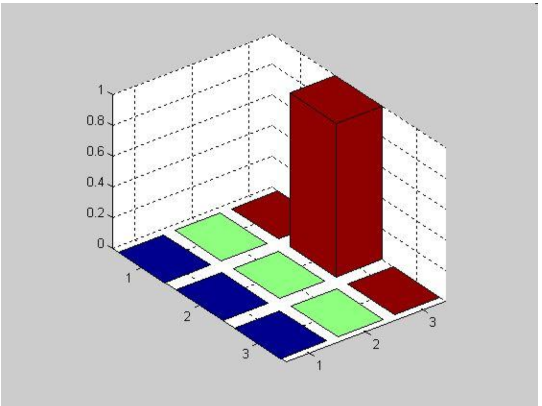


*

0	0	0
0	1	0
0	0	0

=

Linear Filtering

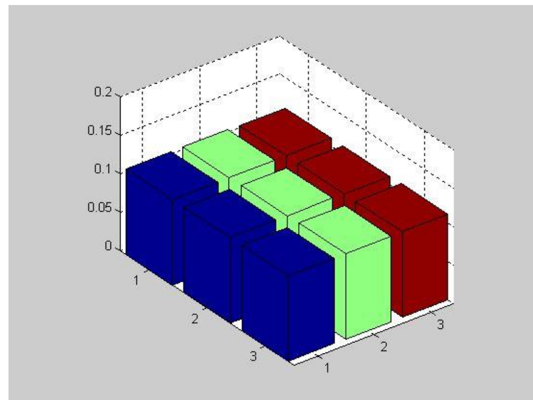


*

0	0	0
0	0	1
0	0	0

=

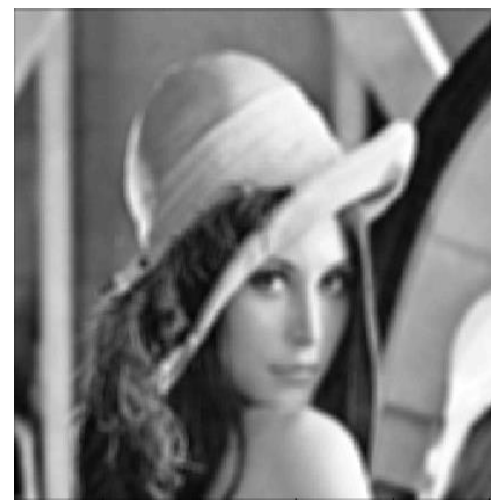
Linear Filtering (Smoothing)



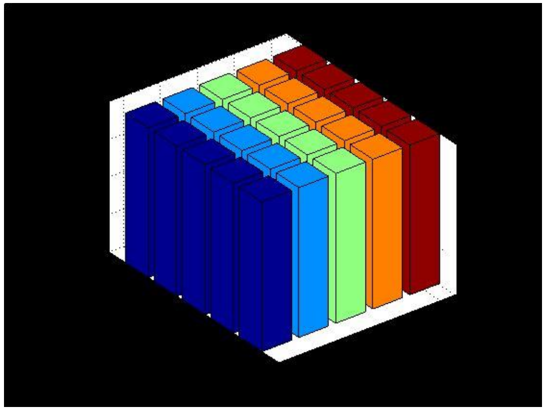
$$* \frac{1}{9}$$

1	**1**	**1**
1	**1**	**1**
1	**1**	**1**

=



Linear Filtering (Blurring)

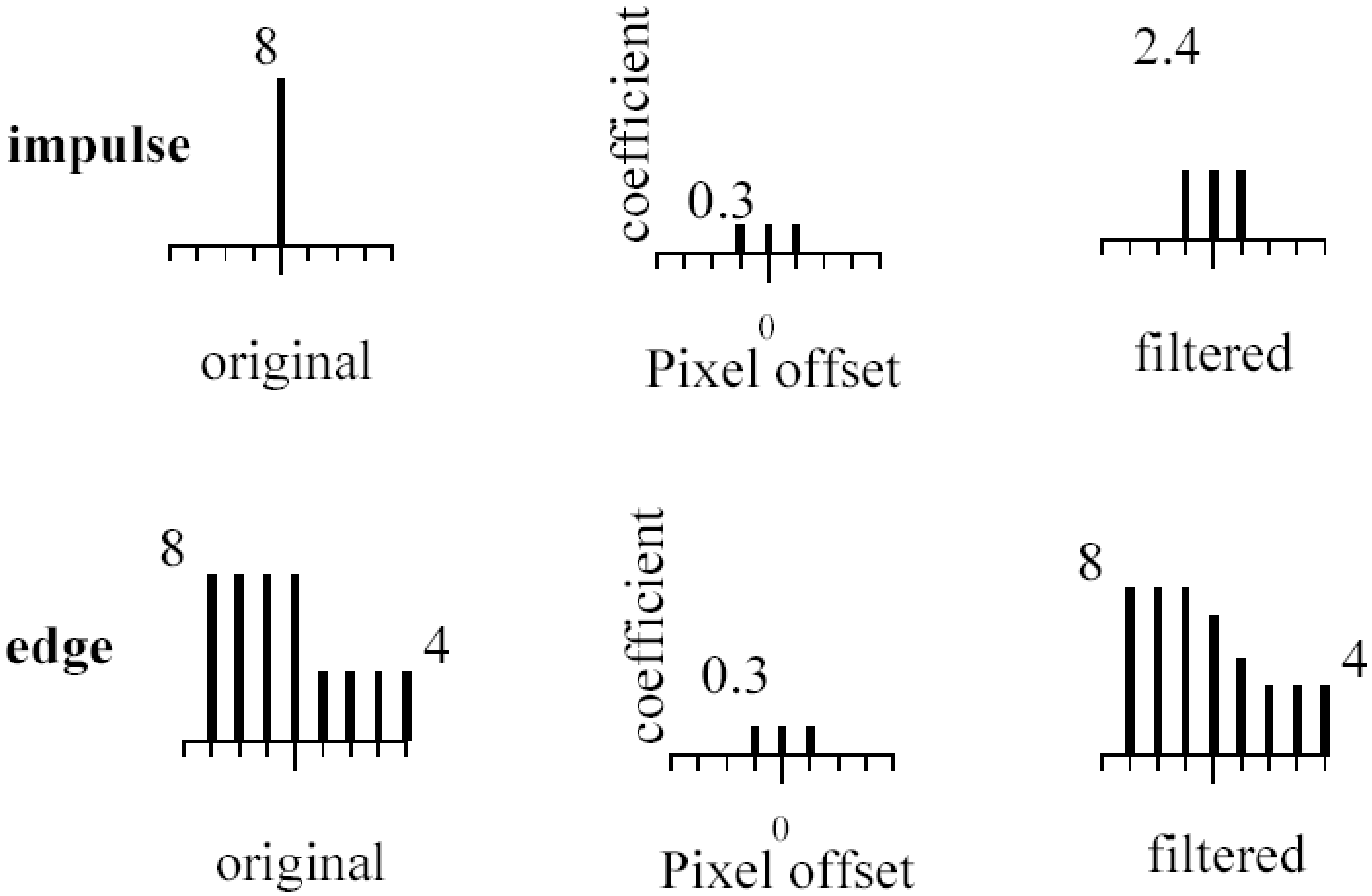


$$* \frac{1}{25}$$

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

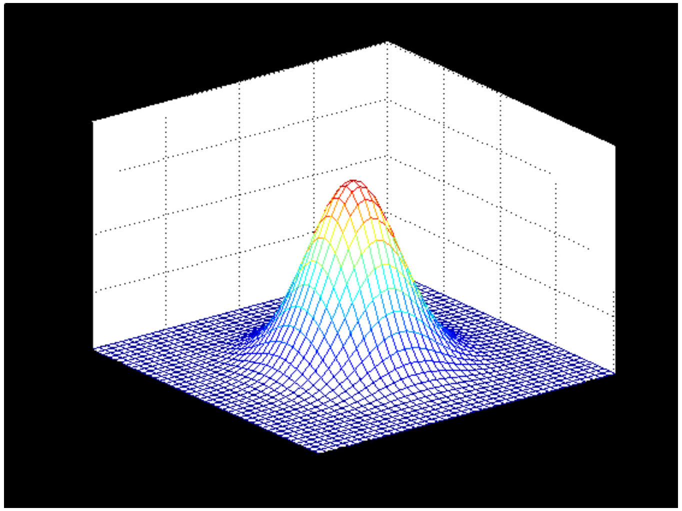
=

Blur examples

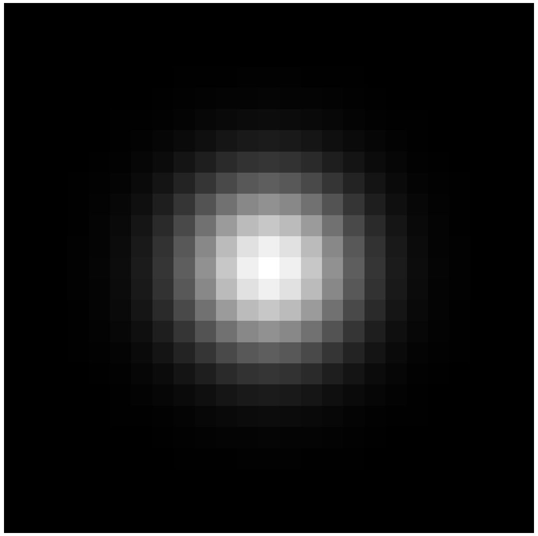


Gaussian Filter

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$



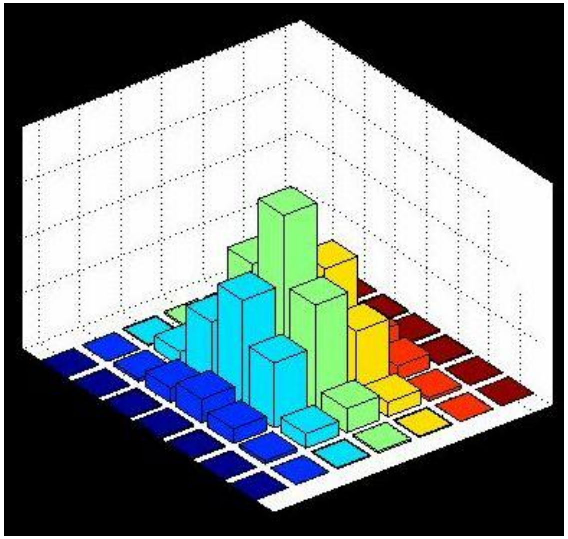
$$H(i,j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left((i-k-1)^2+(j-k-1)^2\right)}{2\sigma^2}\right)$$



where $H(i,j)$ is $(2k+1)\times(2k+1)$ array

Linear Filtering (Gaussian Filter)

*



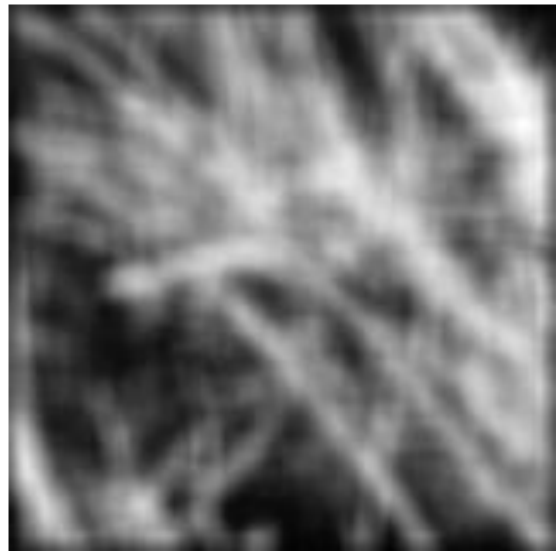
=

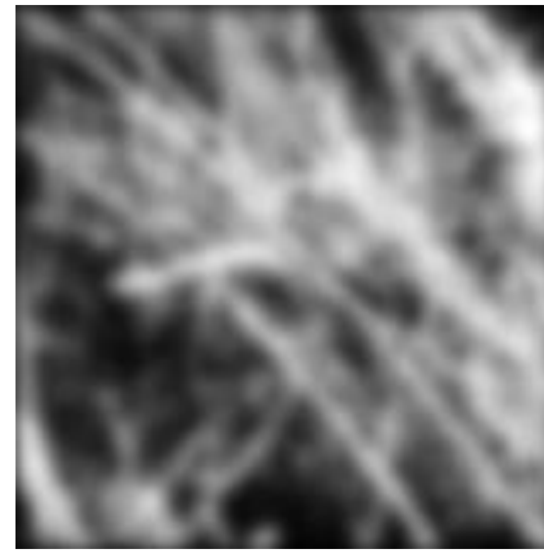
Gaussian vs Average

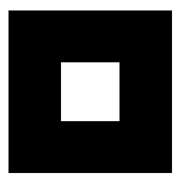
Gaussian Smoothing

Smoothing by Averaging

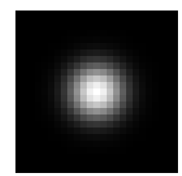
Gaussian vs Average







Average

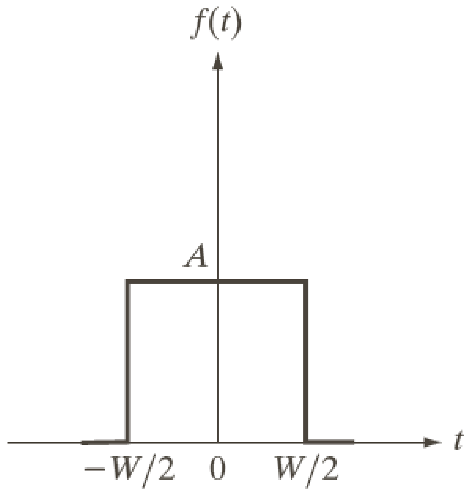


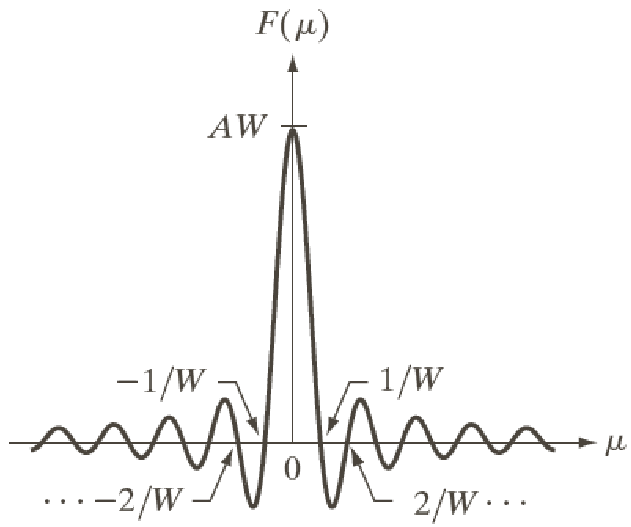
Gaussian

Ringing artifacts near sharp transitions by averaging

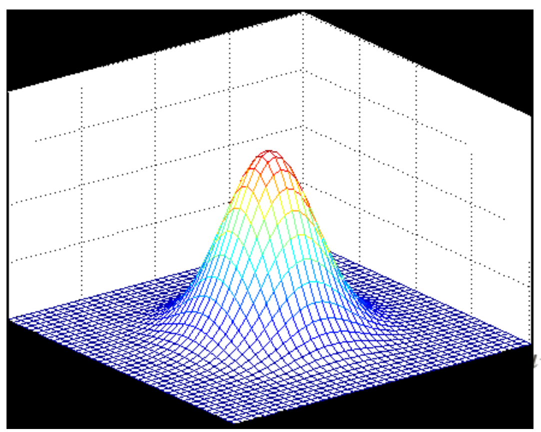
Gaussian vs Average

Averaging





Gaussian



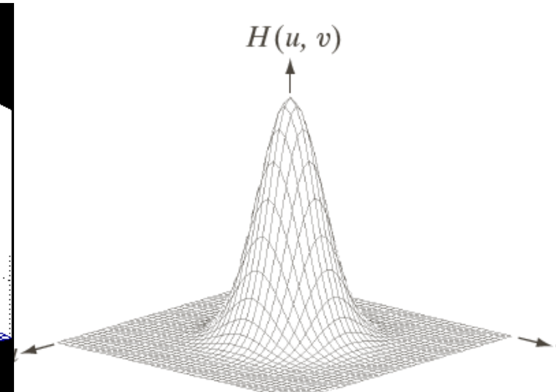


Figure from “Digital Image Processing”, Gonzalez and Woods

Smoothing Reduces Noise

Generally expect pixels to "be like" their neighbors

- surfaces turn slowly
- relatively few reflectance changes

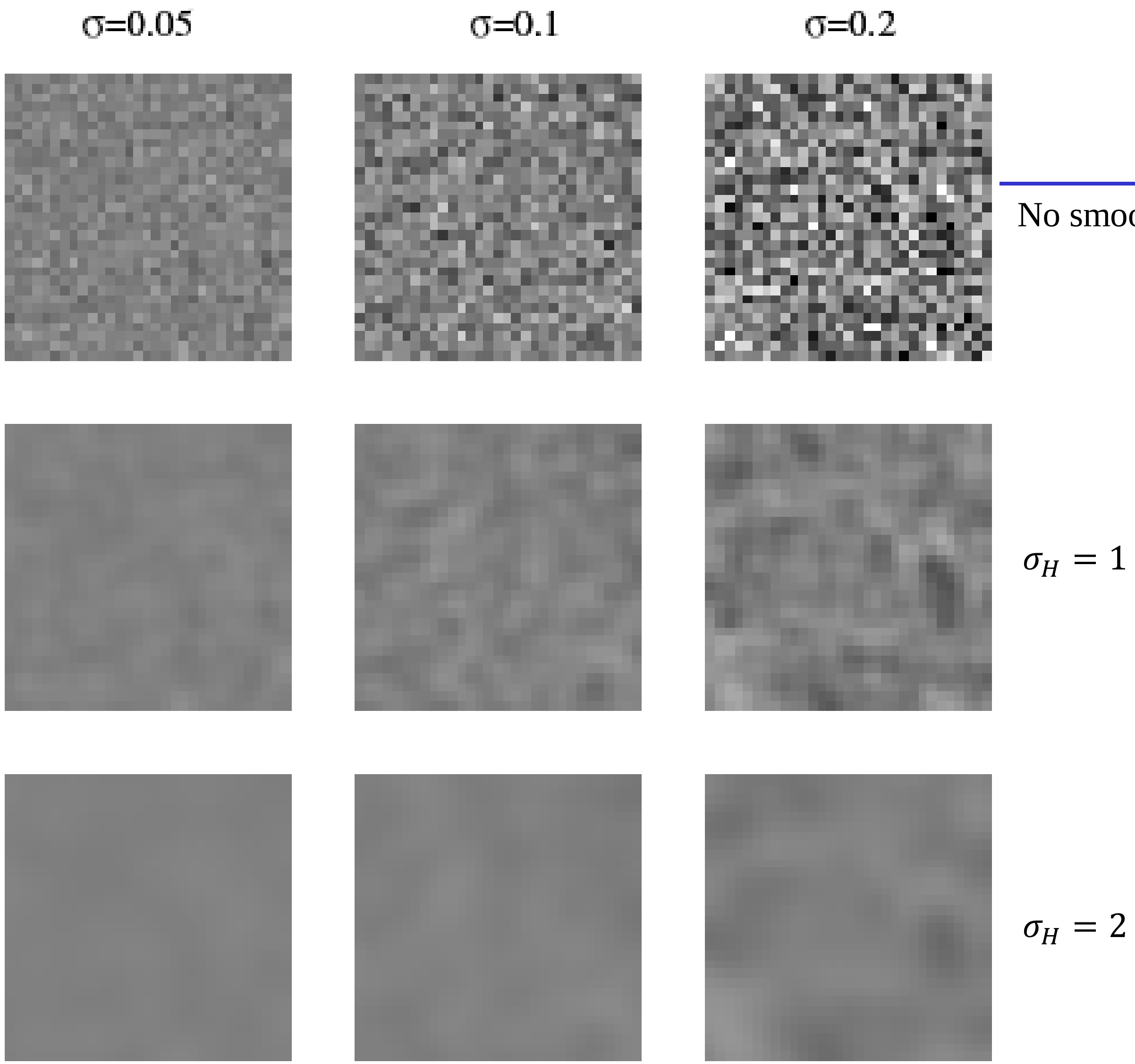
Generally expect noise processes to be independent from pixel to pixel

Implies that smoothing suppresses noise, for appropriate noise models

Scale

- the parameter in the symmetric Gaussian
- as this parameter goes up, more pixels are involved in the average
- and the image gets more blurred
- and noise is more effectively suppressed

Gaussian Noise



The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realisations of an image of gaussian noise.

Other Nonlinear Filters for Noise Reduction

- **Geometric mean filter**
- **Harmonic mean filter**
- **Contraharmonic mean filter**

Continuous noise model

- **Order statistic filters**
  - Median filter
  - Max/min filter

Impulse noise model

Reading Assignments

Chapter 5, Digital Image Processing, Rafael C. Gonzalez and Richard E. Woods, 3rd Edition, Prentice Hall

Image Features

- **Gradient and edges**
- **Corners**
- **Line and curves**
- **Textures**
- **Advanced features**
  - Haar
  - Gabor
  - SIFT
  - HOG
  - Etc.

Gradients and Edges

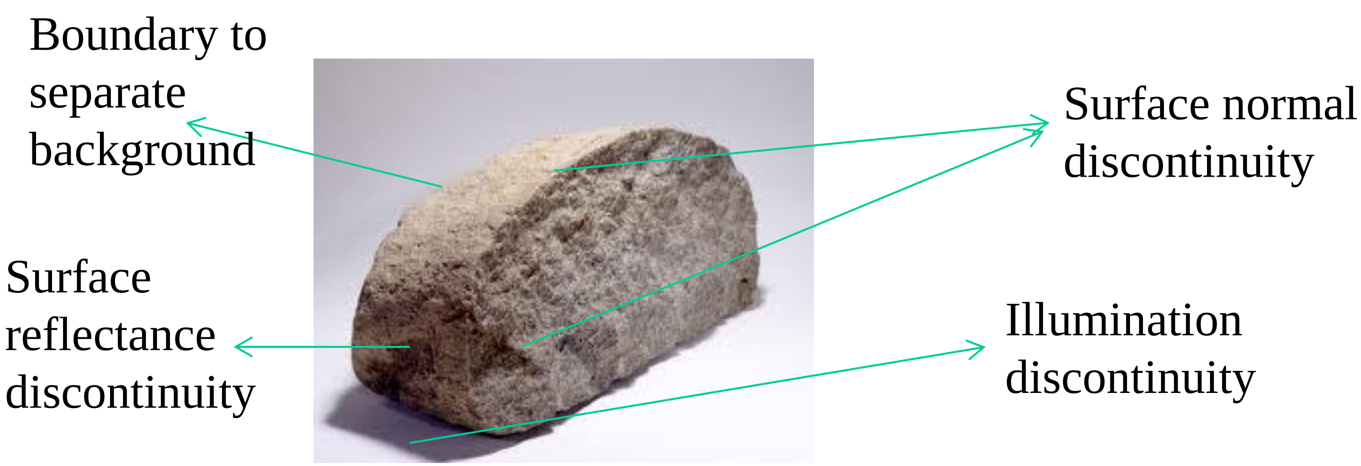
Points of sharp change in an image are interesting:

- change in reflectance
- change in object
- change in illumination
- noise

Sometimes called edge points

General strategy

- determine image gradient
- now mark points where gradient magnitude is particularly large w.r.t neighbors (ideally, curves of such points).



Basic Edge Detection

The first−order derivative

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

The second−order derivative

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

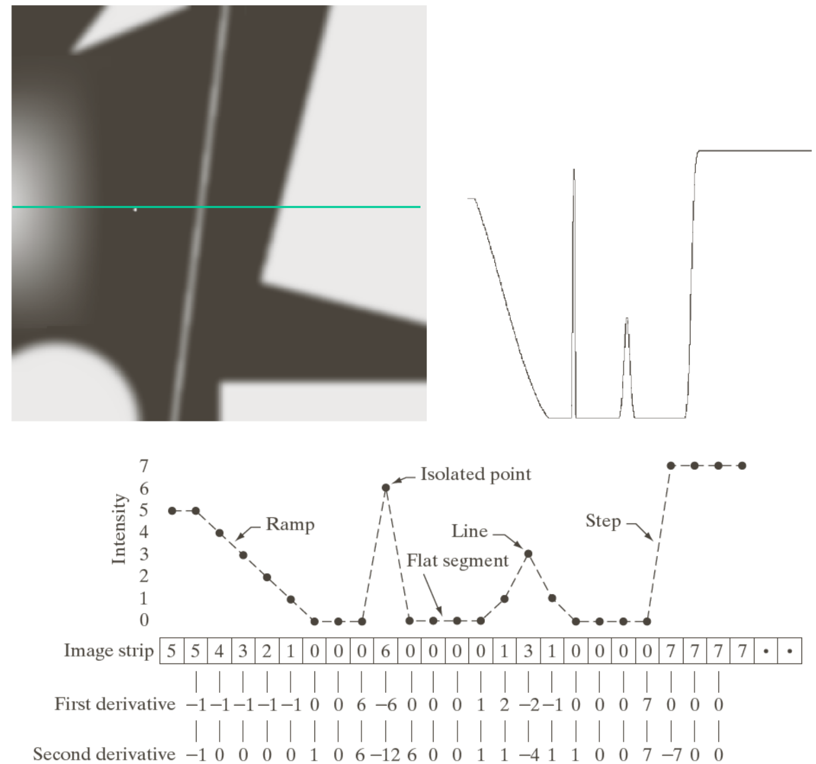


Figure from “Digital Image Processing”, Gonzalez and Woods

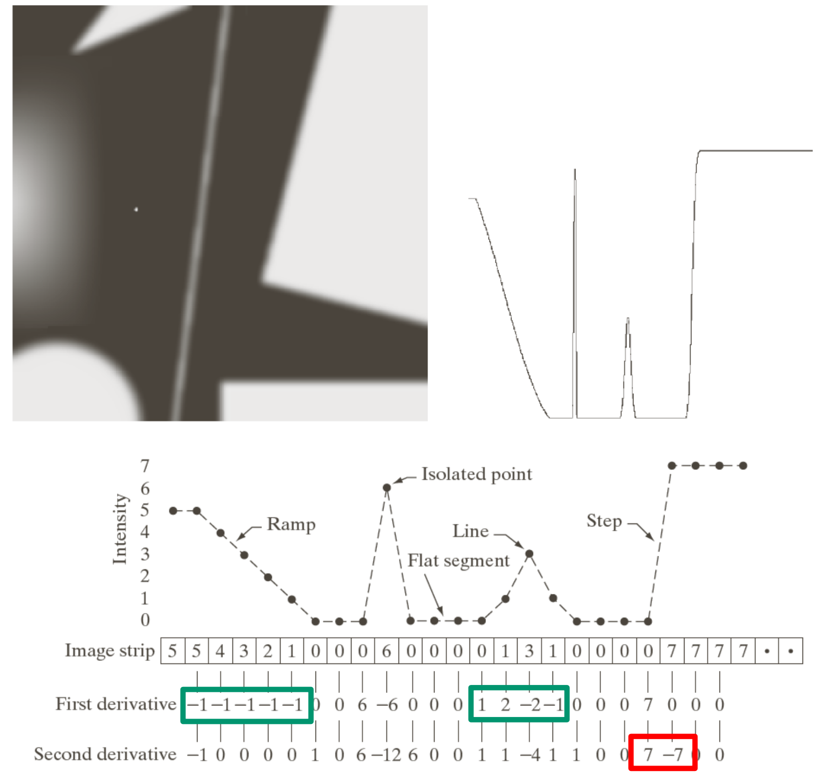
Some Observations

1. **First-order derivatives generally produce thicker edges**

2. **Second-order derivatives have a stronger response to fine details, such as thin lines, isolated points, and noise**

3. **Second-order derivatives produce a double-edge response at ramp and step transitions in intensity**

4. **The sign of the second-order derivative can be used to determine whether a transition into an edge is from light to dark or dark to light**



First-order Derivative based Edge Detection

Gradient
$$\nabla f(x,y) = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

Edge strength
$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$
$$\approx |g_x| + |g_y|$$

Edge direction
$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

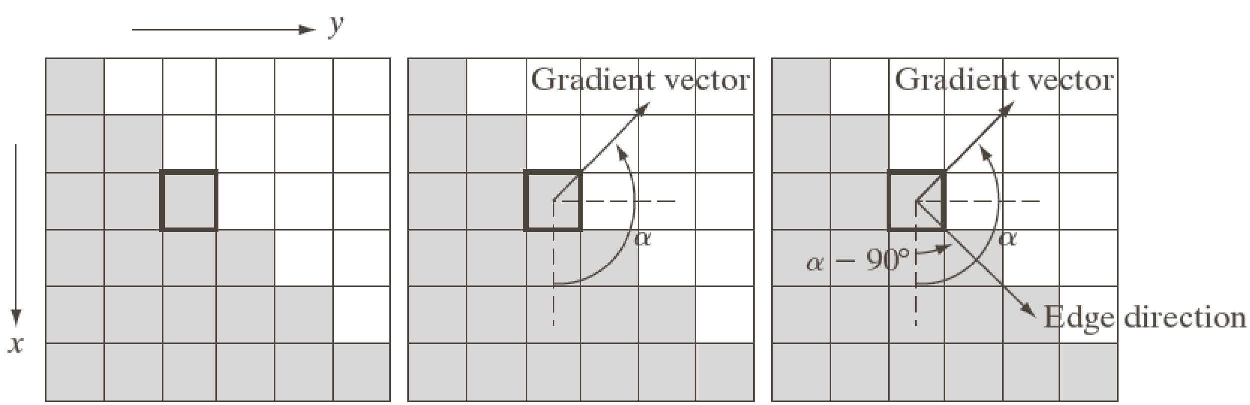
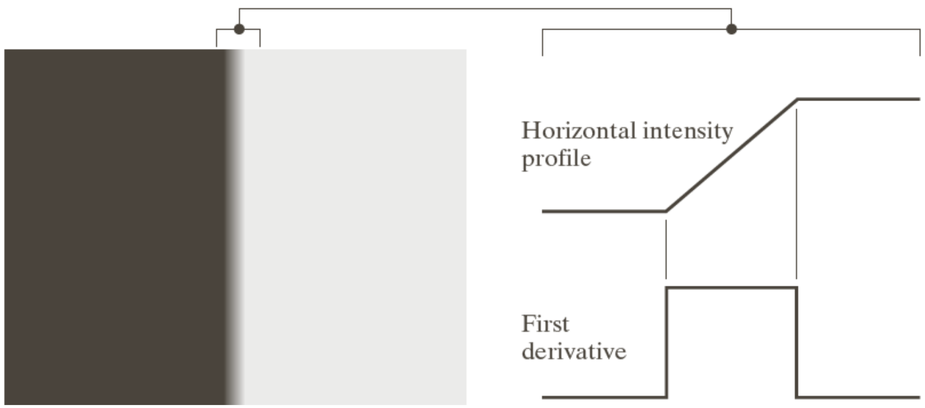


Figure from "Digital Image Processing", Gonzalez and Woods

First-order Derivative



General strategy

- determine image gradient
- now mark points where gradient magnitude is particularly large w.r.t neighbors (ideally, curves of such points).

Figure from "Digital Image Processing", Gonzalez and Woods