Fundamentals of Spatial Filtering
(Sharpening Filters)

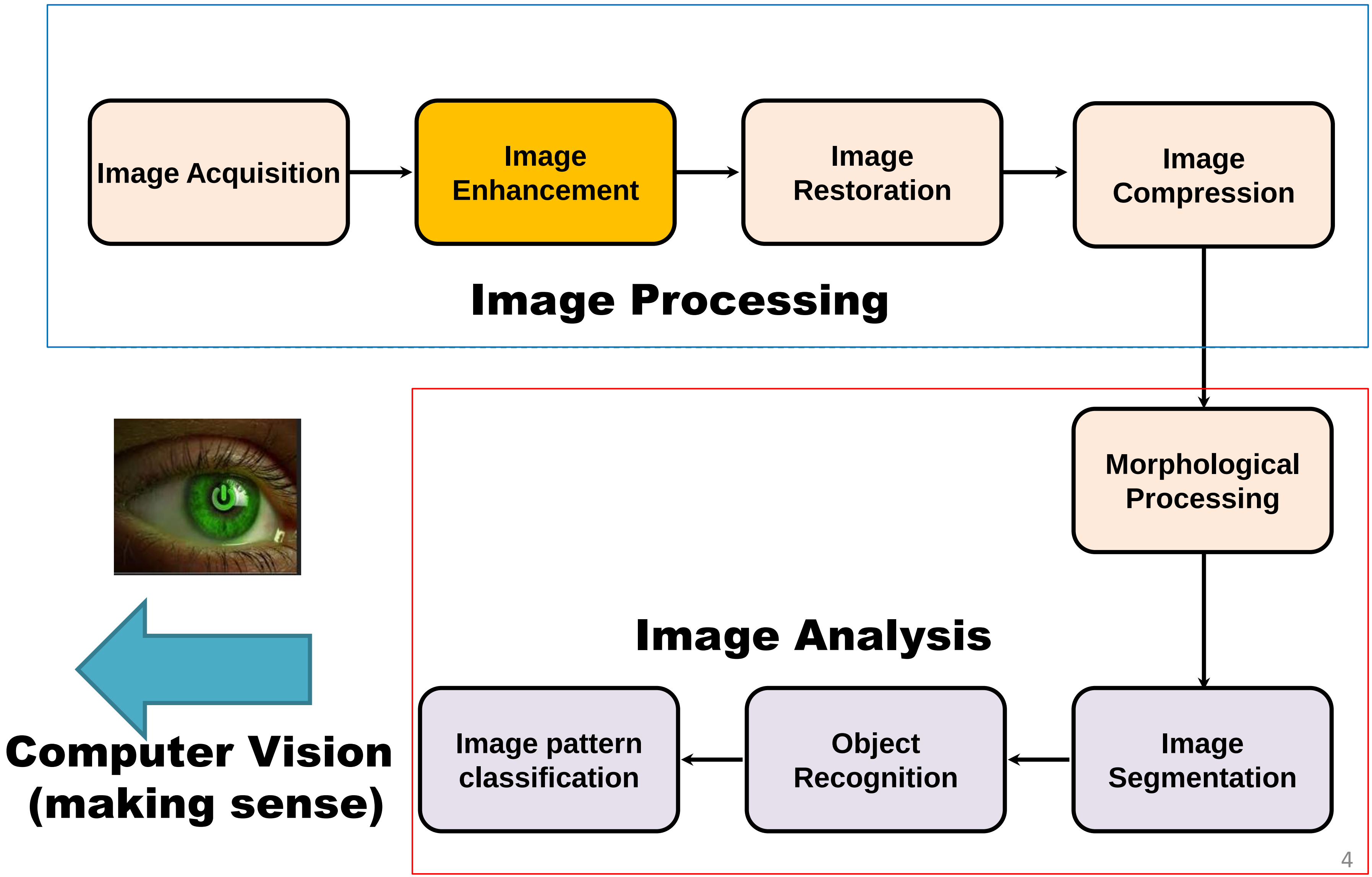
Recap

- Fundamentals of Spatial Filtering
- Correlation and Convolution
- How to construct Spatial filter masks?
- Smoothing (Lowpass) spatial filters
  - Box filter kernels
  - Gaussian filter kernels
  - Smoothing Non-linear Filters

Lecture Objectives

- Sharpening spatial filters
  - Foundation of image sharpening
  - Using second-order derivative for image sharpening (Laplacian)
  - Unsharp masking and high boost filtering
  - Using first-order derivative for image sharpening (Gradient)
  - Highpass, Bandreject, and Bandpass filters from lowpass filters
  - Combining special enhancement methods

Key Stages in DIP



Foundation of Image Sharpening

Basics

- Basic idea:
  - **Highlight** transitions in intensity
  - Opposite behavior to averaging/blur filters

- Mathematical operation:
  - Averaging ≈ Integration
  - Sharpening ≈ Spatial differentiation (**The strength of the *response* of a derivative operator is proportional** to the ***magnitude*** **of the intensity discontinuity at the point at which the operator is applied**)

- Image differentiation **enhances** *edges and other discontinuities* (noise) and **de-emphasize** areas with *slowly varying intensities*.

- Image sharpening is often referred to as *highpass filtering*.

Basics

- We try to understand the *digital differentiation* with 1D signals and then extend our understanding to 2D signals (images).

- Primary areas of interest are:
  - Constant intensity (flat regions)
  - Onset and end of discontinuities (*step* and *ramp* discontinuities)
  - Along Intensity ramps

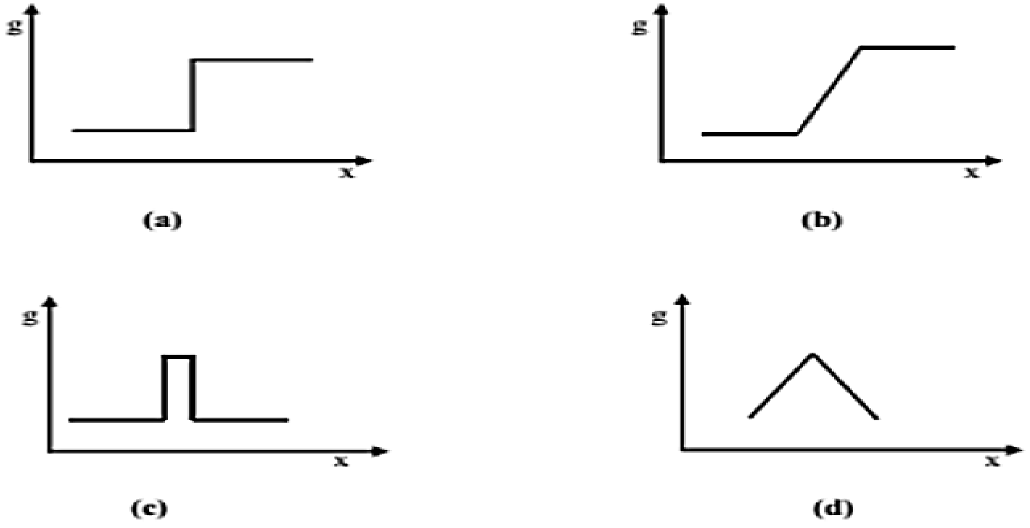
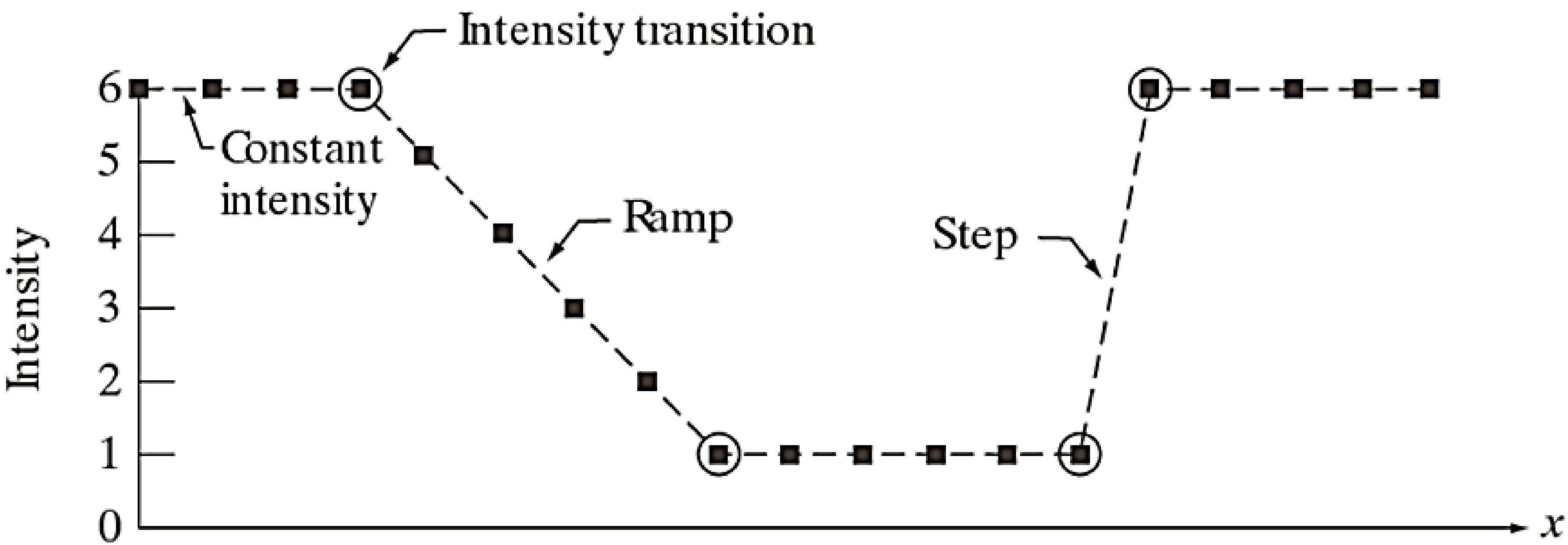


Figure : Edge Types (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge.

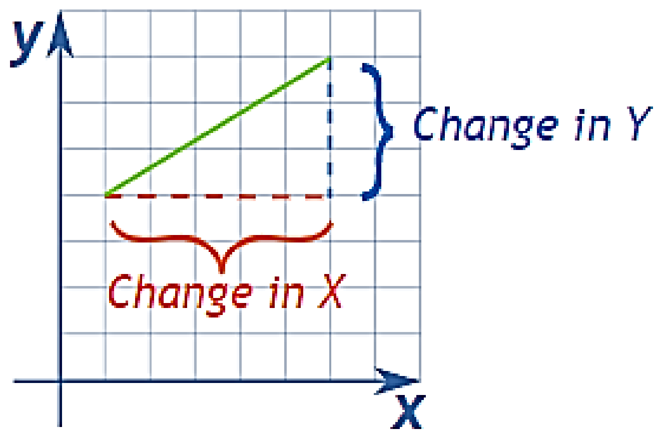
Basics


Intensity transition
Constant intensity
Ramp
Step
Intensity
x
0
1
2
3
4
5
6

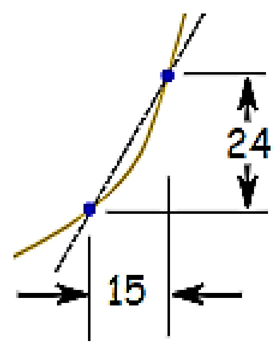
Introduction to Derivatives

- It is all about **slope !!!**

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}}$$



We can find an **average** slope between two points.

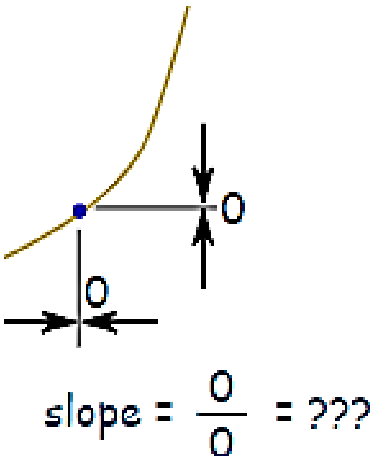


$$\text{average slope} = \frac{24}{15}$$

Introduction to Derivatives

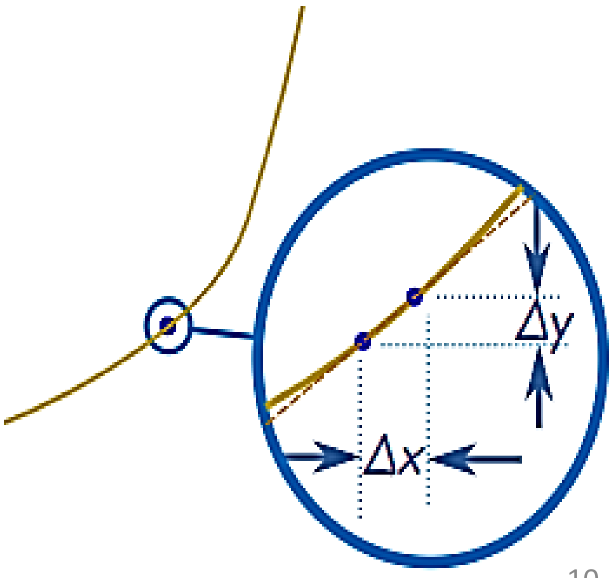
But how do we find the slope **at a point**?

There is nothing to measure!



But with derivatives we use a small difference ...

... then have it **shrink towards zero**.



Introduction to Derivatives

Let us Find a Derivative!

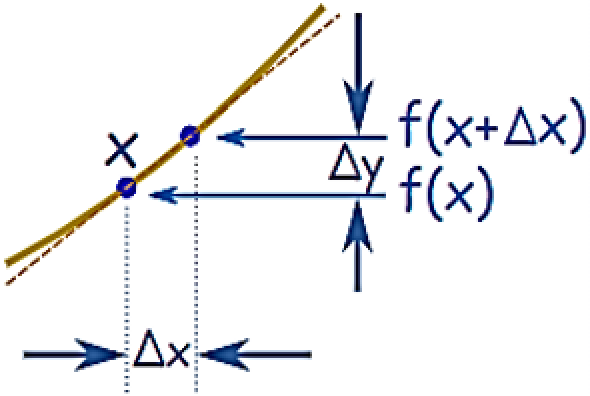
To find the derivative of a function $y = f(x)$ we use the slope formula:

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}} = \frac{\Delta y}{\Delta x}$$

And (from the diagram) we see that:

x changes from x to $x+\Delta x$

y changes from $f(x)$ to $f(x+\Delta x)$



Now follow these steps:

- Fill in this slope formula: $\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$
- Simplify it as best we can
- Then make $\mathbf{\Delta x}$ shrink towards zero.

Introduction to Derivatives - example

Example: the function $\mathbf{f(x)} = \mathbf{x^2}$

We know $\mathbf{f(x)} = \mathbf{x^2}$, and we can calculate $\mathbf{f(x+\Delta x)}$:

Start with: $\mathbf{f(x+\Delta x) = (x+\Delta x)^2}$

Expand $(x + \Delta x)^2$: $\mathbf{f(x+\Delta x) = x^2 + 2x\, \Delta x + (\Delta x)^2}$

The slope formula is: $\dfrac{f(x+\Delta x) - f(x)}{\Delta x}$

Put in $\mathbf{f(x+\Delta x)}$ and $\mathbf{f(x)}$: $\dfrac{x^2 + 2x\, \Delta x + (\Delta x)^2 - x^2}{\Delta x}$

Simplify (x^2 and $-x^2$ cancel): $\dfrac{2x\, \Delta x + (\Delta x)^2}{\Delta x}$

Simplify more (divide through by Δx): $= 2x + \Delta x$

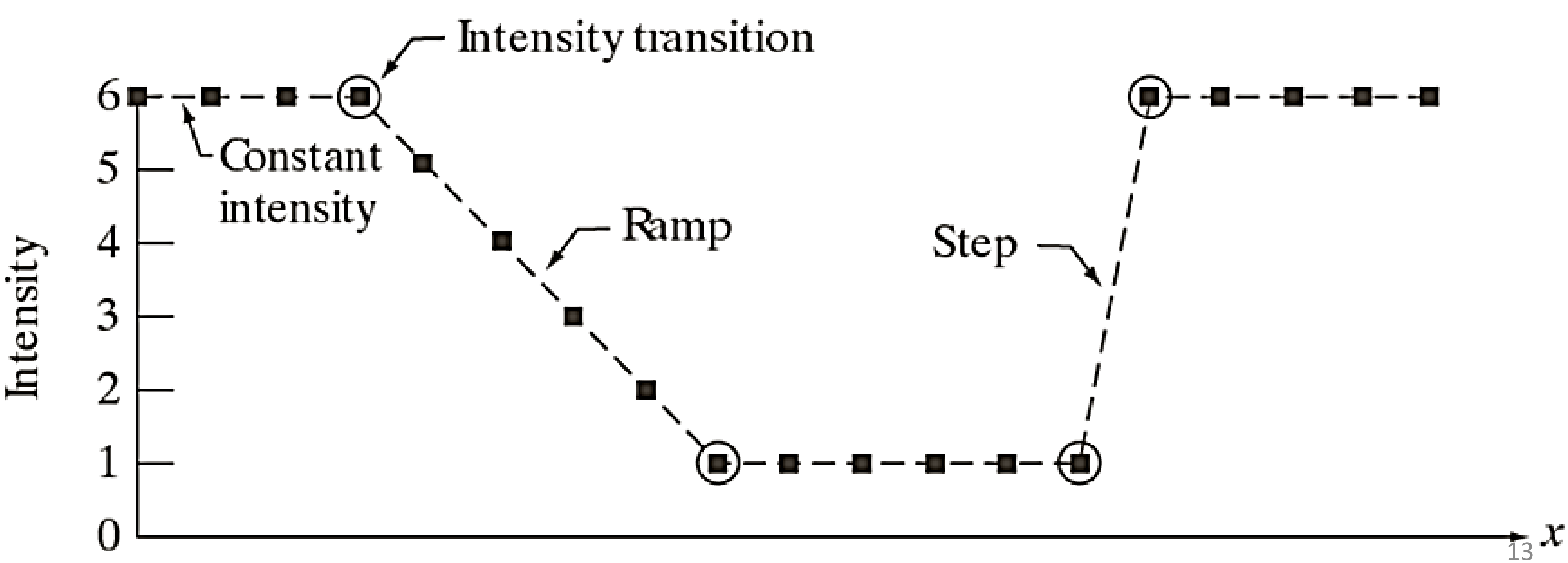
Then, **as Δx heads towards 0** we get: $= 2x$

Result: the derivative of $\mathbf{x^2}$ is $\mathbf{2x}$

In other words, the slope at x is $\mathbf{2x}$

First Order Derivative

- Properties of digital differentiation - **first order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at the onset of an intensity step or ramp
  3. **Must be nonzero** along ramps



First Order Derivative

- A basic definition of the *first-order derivative* of a *one-dimensional function* **f(x)** is the difference:

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

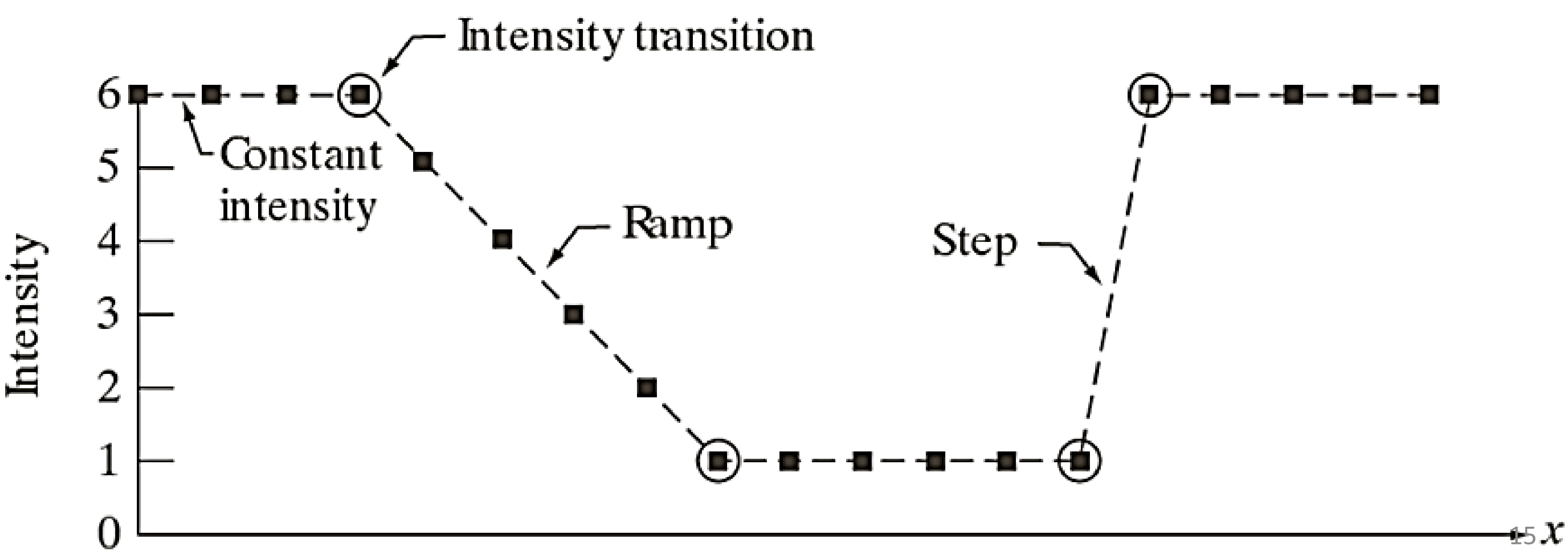
- Example: **f(x)=[1,1,2,3,4,5,8,7,7,7,7,4,4,4,1,1,1]**

What is the **first order derivative** of f(x)=?

To avoid a situation in which the *previous or next points are outside the range of the scan line*, we calculate derivatives from the second through the second last points in the sequence

Second Order Derivative

- Properties of digital differentiation - **second order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at onset and end of an intensity step or ramp
  3. **Must be zero** on a ramp of constant slope



Second Order Derivative

- A basic definition of the *second-order derivative* of a *one-dimensional function* **f(x)** is the difference

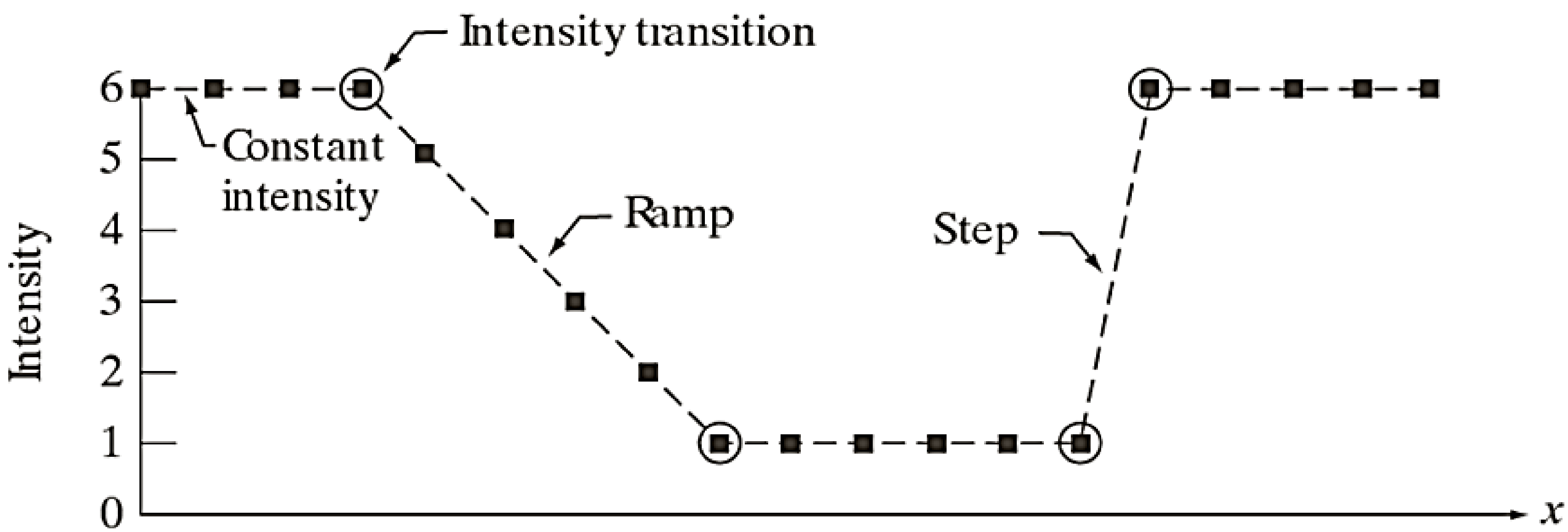
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

- Example: **f(x)=[1,1,2,3,4,5,8,7,7,7,7,4,4,4,1,1,1]**

What is the **second order derivative** of f(x)=?

To avoid a situation in which the *previous or next points are outside the range of the scan line*, we calculate derivatives from the second through the second last points in the sequence

First Order Vs. Second Order Derivative

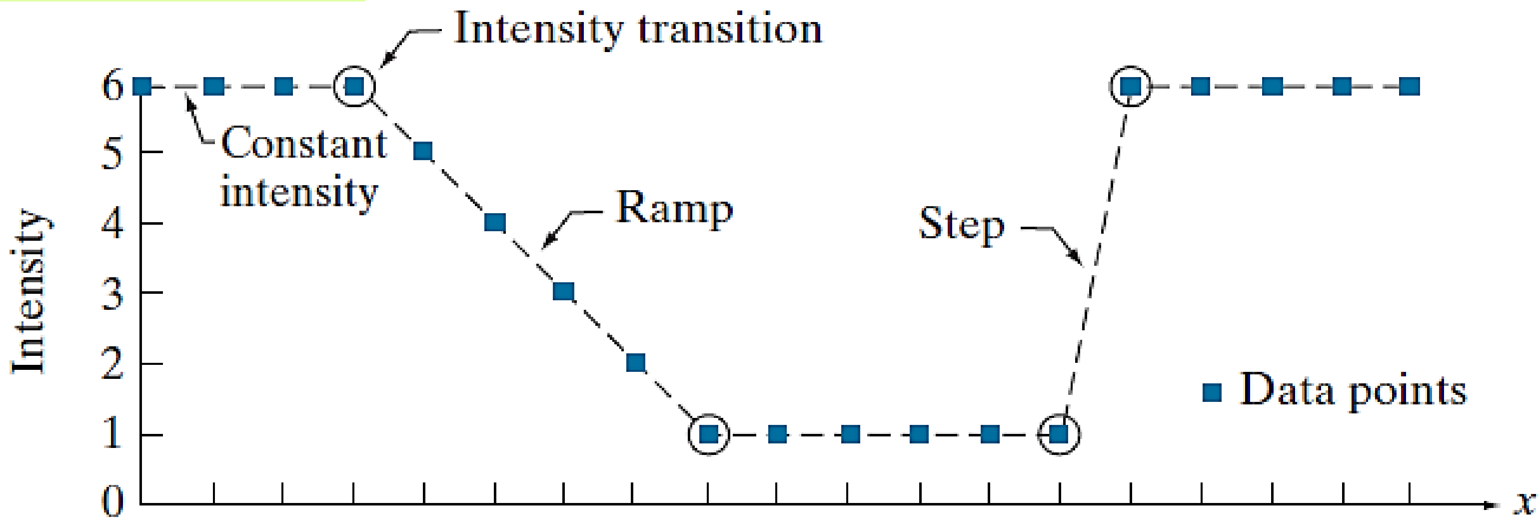


- Properties of digital differentiation - **first order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at the onset of an intensity step or ramp
  3. **Must be nonzero** along ramps

- Properties of digital differentiation - **second order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at onset and end of an intensity step or ramp
  3. **Must be zero** on a ramp of constant slope

Illustration of 1st Order Derivative

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

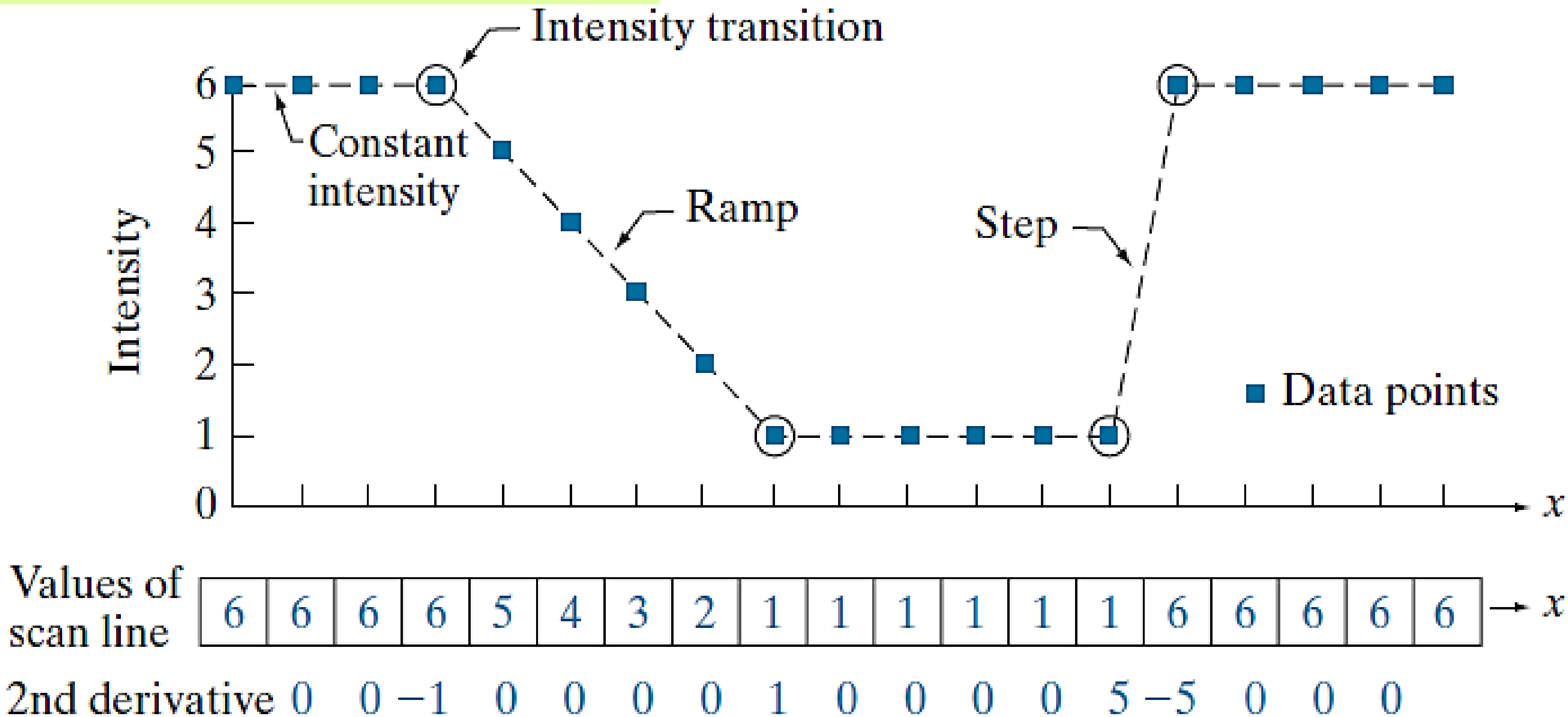


Values of scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	

- Properties of digital differentiation - **first order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at the onset of an intensity step or ramp
  3. **Must be nonzero** along ramps

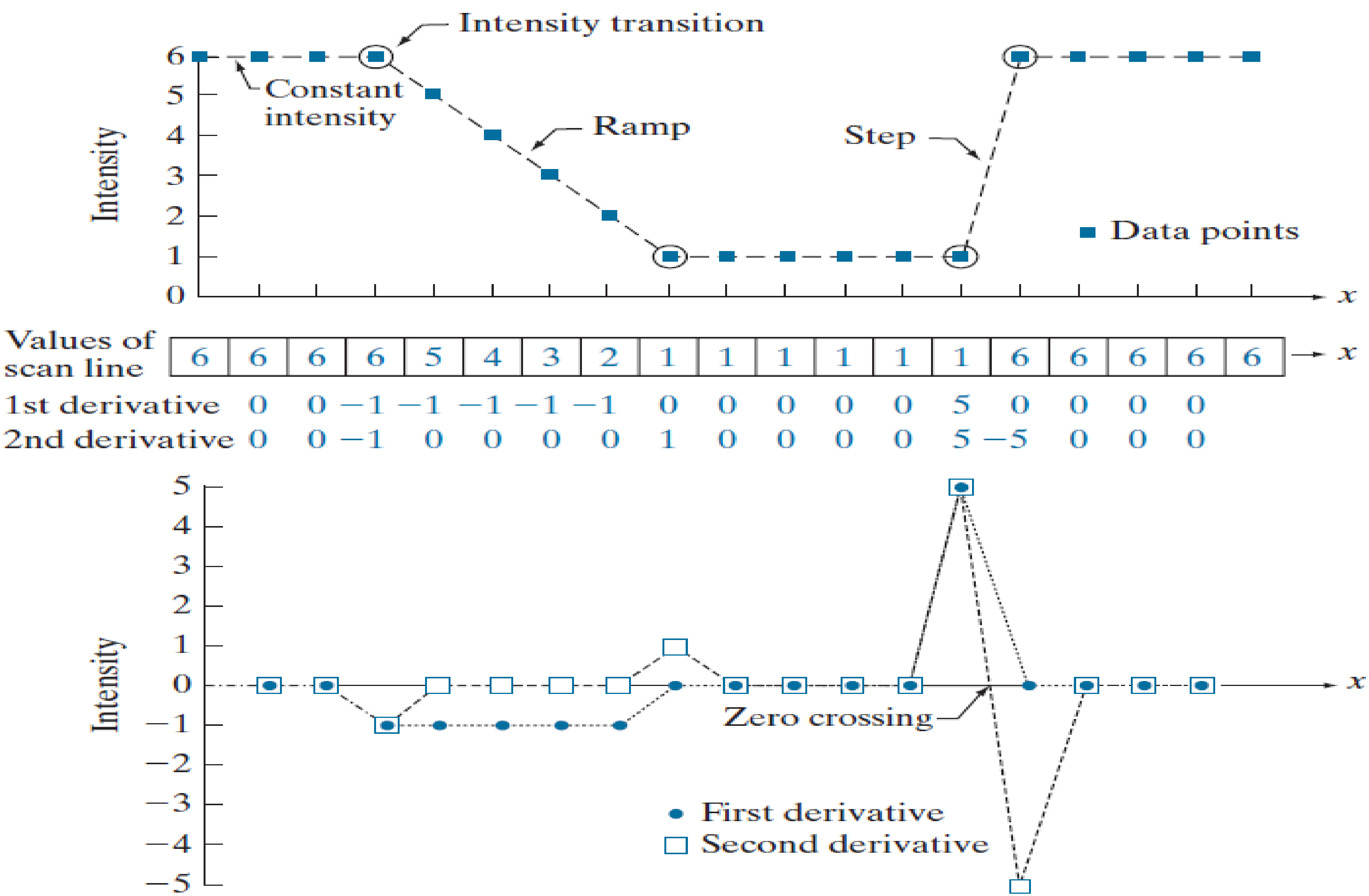
Illustration of 2nd Order Derivative

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



- Properties of digital differentiation - **second order derivative:**
  1. **Must be zero** in the areas of constant intensity
  2. **Must be nonzero** at onset and end of an intensity step or ramp
  3. **Must be zero** on a ramp of constant slope

Illustration of 1D Derivatives



The ***zero crossing*** property is quite useful for locating edges

Derivatives - Important Observations

- Edges in digital images often are ramp-like transitions in intensity.

- The **first-order derivative** is nonzero along the entire ramp (*leads to thick edge*), while the **second-order derivative** is nonzero only at the onset and end of the ramp (*leads to double edge one pixel thick, separated by zeros*).

- The response at and around the onset and end point of the intensity transition is **much stronger** for the second-order derivative than for the first-order derivative.

- **Second-order derivative** enhances fine details much better than first-order derivative.

Using the Second-order Derivative for Image Sharpening

The Laplacian Kernel

Second-order Derivative for Image Sharpening

- Isotropic filters
  - Possess uniform(invariant) behavior *irrespective of the orientation of the image*.
  - Response is independent of the *direction of the discontinuities in the image*.

- The Laplacian kernel
  - Simplest isotropic second-order derivative operator (kernel)
  - Defines discrete formulation of a function
  - Use the discrete values to construct a filter mask

The Laplacian operator(kernel)

For a function (image) ***f* (*x*, *y*)** of two variables, Laplacian is defined as:

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

Recall that in 1D:

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

So, in the **x-direction**, we have:

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

and, in the **y-direction**, we have:

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

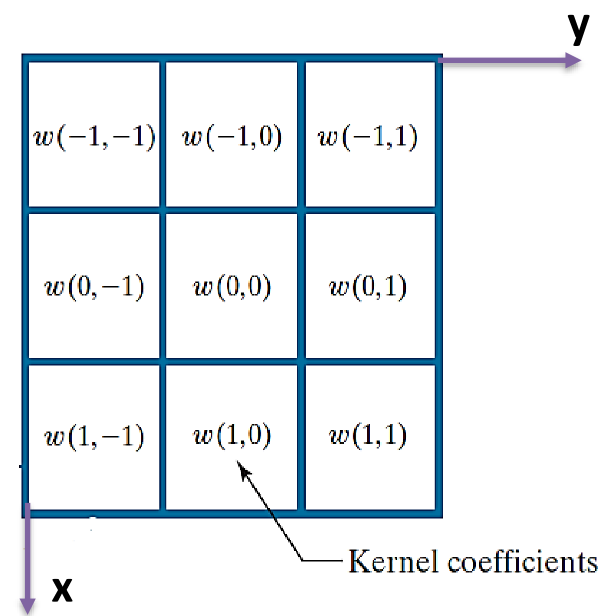
The Laplacian operator(kernel)

So, from the preceding two equations, the discrete Laplacian for a function (image) ***f* (*x*, *y*)** of two variables, it is defined as:

$$\boxed{\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)}$$

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

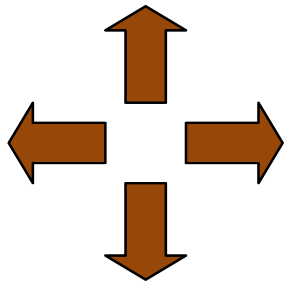
$$w(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



The Laplacian operator(kernel)

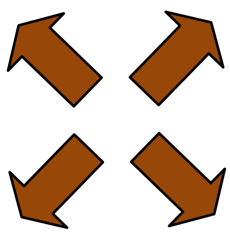
0	1	0
1	-4	1
0	1	0

Isotropic for rotation
in increments of **90**°

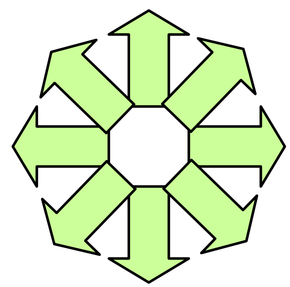


1	0	1
0	-4	0
1	0	1

Isotropic for rotation
in increments of **90**°



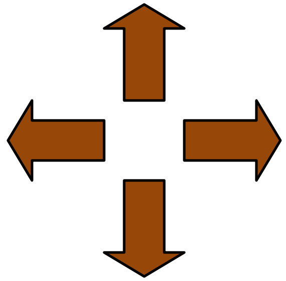
1	1	1
1	-8	1
1	1	1



Isotropic for rotation
in increments of **45**°

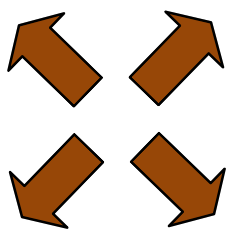
The Laplacian operator(kernel)

0	-1	0
-1	4	-1
0	-1	0



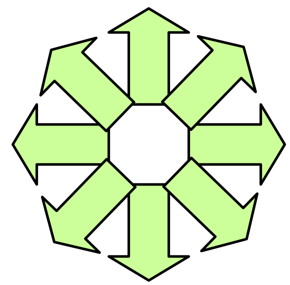
Isotropic for rotation
in increments of **90**°

-1	0	-1
0	4	0
-1	0	-1



Isotropic for rotation
in increments of **90**°

-1	-1	-1
-1	8	-1
-1	-1	-1



Isotropic for rotation
in increments of **45**°

Obtained from definitions of the second derivatives that are the **negatives** of the ones we discussed before

The Laplacian operator(kernel)
How to use ??

- Because the Laplacian is a derivative operator,
  - Laplacian operator *highlights sharp intensity transitions(edges) in an image* and de-emphasizes regions of slowly varying intensities.
  - Laplacian operator tend to produce images that have grayish edge lines and other discontinuities, all *superimposed* on a dark, featureless background.
  - **<u>Background features</u> of an original image can be *"recovered"* while still preserving the <u>sharpening effect</u> of the Laplacian by adding the Laplacian image to the original image.**

The Laplacian operator(kernel) - Facts

- It is important to keep in mind *which definition of the Laplacian* is used:

 1. If the definition used has a *negative center coefficient*, then we ***subtract the Laplacian image from the original image*** *to obtain a sharpened result*.

 2. If the definition used has a *positive center coefficient*, then we ***add the Laplacian image to the original image*** *to obtain a sharpened result*.

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient is positive} \end{cases}$$

where $f(x,y)$ and $g(x,y)$ is the original image and the sharpened image, respectively

The Laplacian operator(kernel)

Image Sharpening

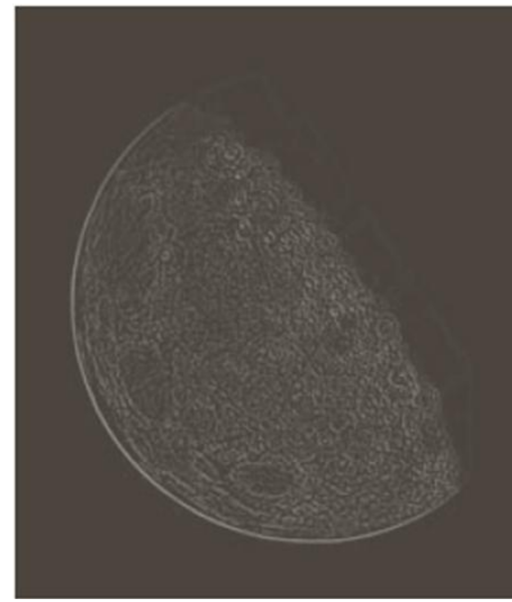
North pole of the moon

★

0	1	0
1	-4	1
0	1	0

Laplacian kernel

=



Laplacian image

The Laplacian operator(kernel)

Image Sharpening

Original

Original ✚

0	1	0
1	4	1
0	1	0

Original ✚

1	1	1
1	8	1
1	1	1

The Sum of Coefficients of Smoothing & Sharpening filters

- The sum of normalized **Smoothing kernel** coefficients is usually One so that:
  - The **Constant areas** in the original image filtered with these kernels would be constant in the filtered image also.
  - The **sum of the pixels** in the original and filtered images will be the same, thus preventing a bias from being introduced by filtering.

- The sum of **Sharpening kernel** coefficients is usually Zero so that:
  - when a derivative kernel encompasses a constant region in a image, the result of convolution in that location **must be zero**.

Unsharp Masking and Highboost Filtering

Unsharp Masking

- Traditional process used since the 1930s by the *printing* and *publishing* industry.

- **Procedure**
  1. Blur the original image
  2. Subtract the blurred image from the original image. The resulting difference is called the **"mask"**
  3. Add a weighted portion of the mask to the original image

Unsharp Masking - Procedure

Let $f(x, y)$ be the image and
let $\bar{f}(x, y)$ denote the blurred image.

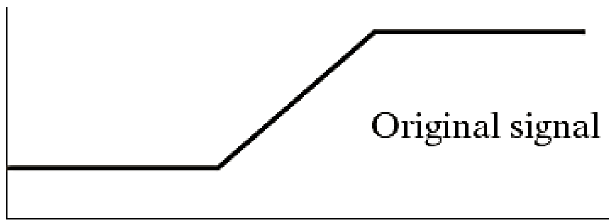
First obtain the mask:
$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$

Add a "weighted" portion of the mask to the original image:
$g(x, y) = f(x, y) + k * g_{mask}(x, y)$

$k \geq 0$, in general.

$\mathrm{k} = 1$: Unsharp masking
$\mathrm{k} > 1$: Highboost filtering

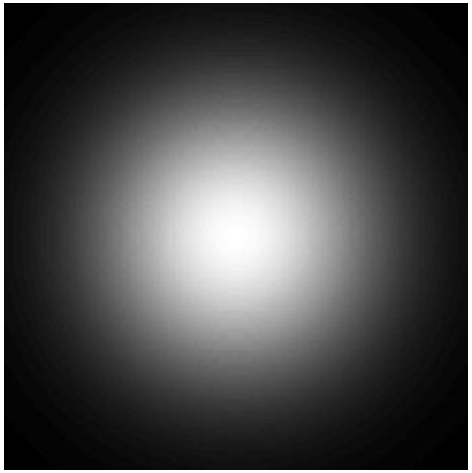
Unsharp Masking - Illustration

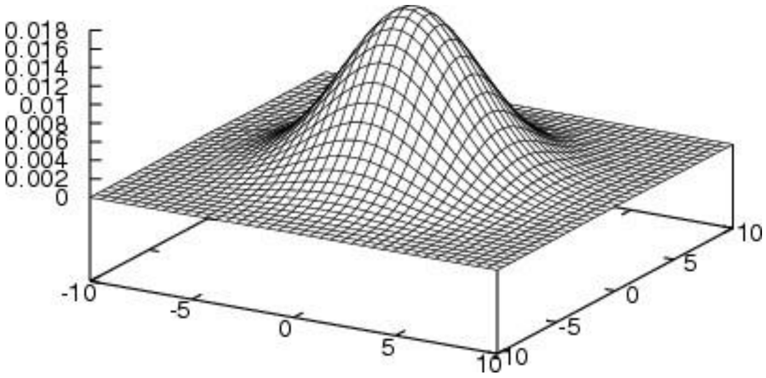


Unsharp Masking - Illustration

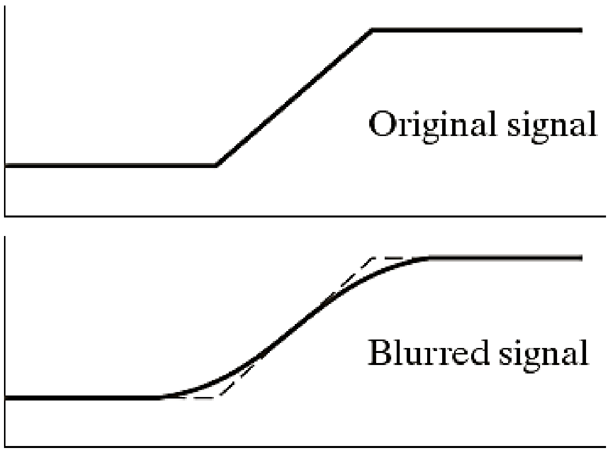
Use 2D Gaussian kernel for image blurring (smoothing)

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$



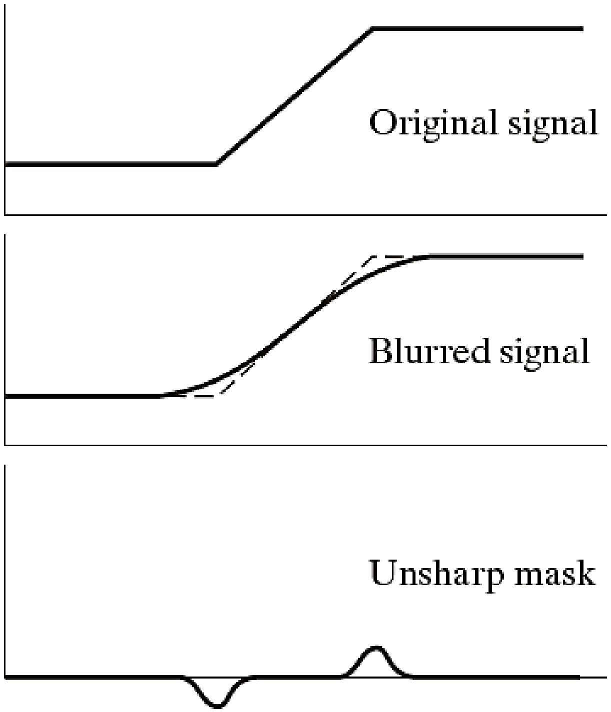


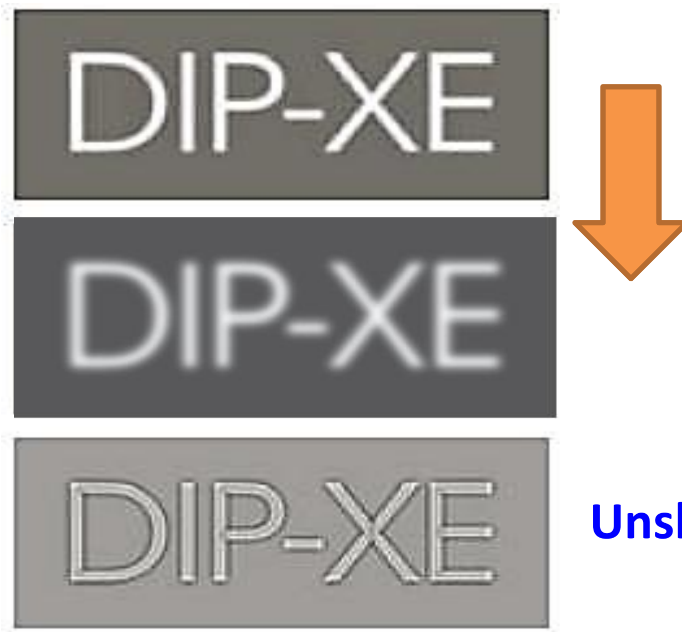
Unsharp Masking - Illustration



Gaussian filter
σ=3

Unsharp Masking - Illustration

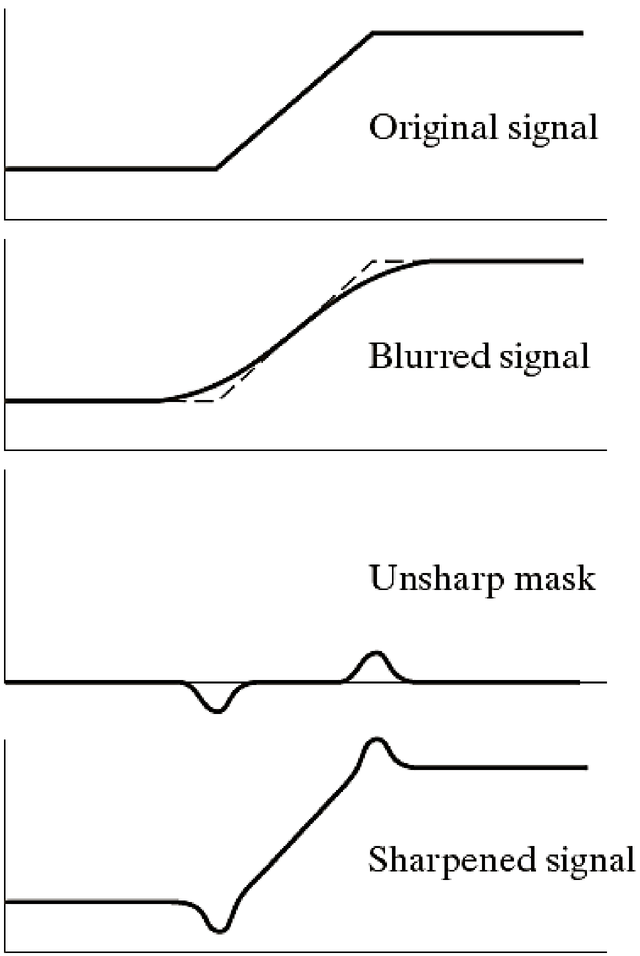




Gaussian filter
$\sigma=3$

Unsharp Mask

Unsharp Masking - Illustration



Gaussian filter
σ=3

Unsharp Mask

Result of **Unsharp Masking** **K=1**

Result of **Highboost Filtering** **K=4.5**

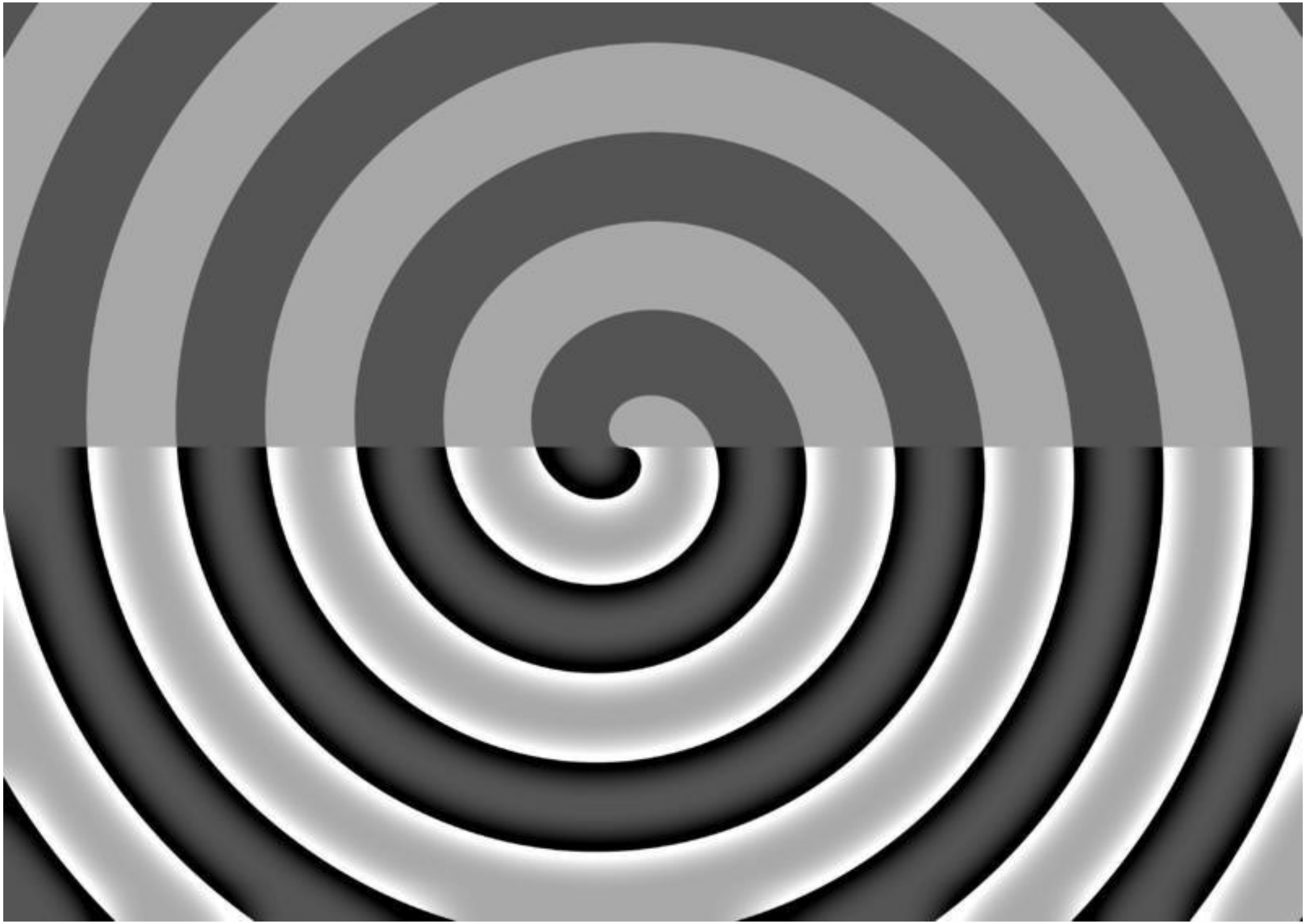
Unsharp Masking - Illustration

Original image

Unsharp Masking

Highboost Filtering

Unsharp Masking - Illustration



Unsharp Masking - Illustration

Original

Sharpened Image, k=1

Highboost Image, k>1

Using First-order Derivative for Image Sharpening

The Gradient

Gradient in 2D

- For an image f(x,y), the **gradient** of f(x,y) at coordinates (x,y) is defined as the two-dimensional column vector:

$$\nabla f \equiv \operatorname{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\[2mm] \dfrac{\partial f}{\partial y} \end{bmatrix}$$

- The gradient points in the **direction** of the **maximum rate of change of *f*** at location (*x*, *y*).

- The **First derivatives** in image processing are *implemented* using the **magnitude of the gradient**.

Magnitude of the Gradient

- The **magnitude** (length) of the vector **∇f**, denoted as **M(x, y),** which is also denoted as **|| ∇f ||** is defined as:

$$M(x,y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

- **Magnitude** -> scalar at location (x, y).

- **M(x, y)** is referred to as the **gradient image**.

Magnitude of the Gradient - Approximation

- **Derivative** -> linear operator

- **Magnitude** contains squares and square-root
  - Non-linear
  - Hence, the gradient is a non-linear operator

- **Approximation**:

$$M(x, y) \approx |g_x| + |g_y|$$

Magnitude of the Gradient - Approximation

- The **simplest approximation** of a *first-order derivatives* that satisfy all the following conditions are:
  1. Must be zero in the areas of constant intensity
  2. Must be nonzero at the onset of an intensity step or ramp
  3. Must be nonzero along ramps

$g_x = f(x+1,y) - f(x,y)$

$g_y = f(x,y+1) - f(x,y)$

0	0	0
0	-1	0
0	1	0

0	0	0
0	-1	1
0	0	0

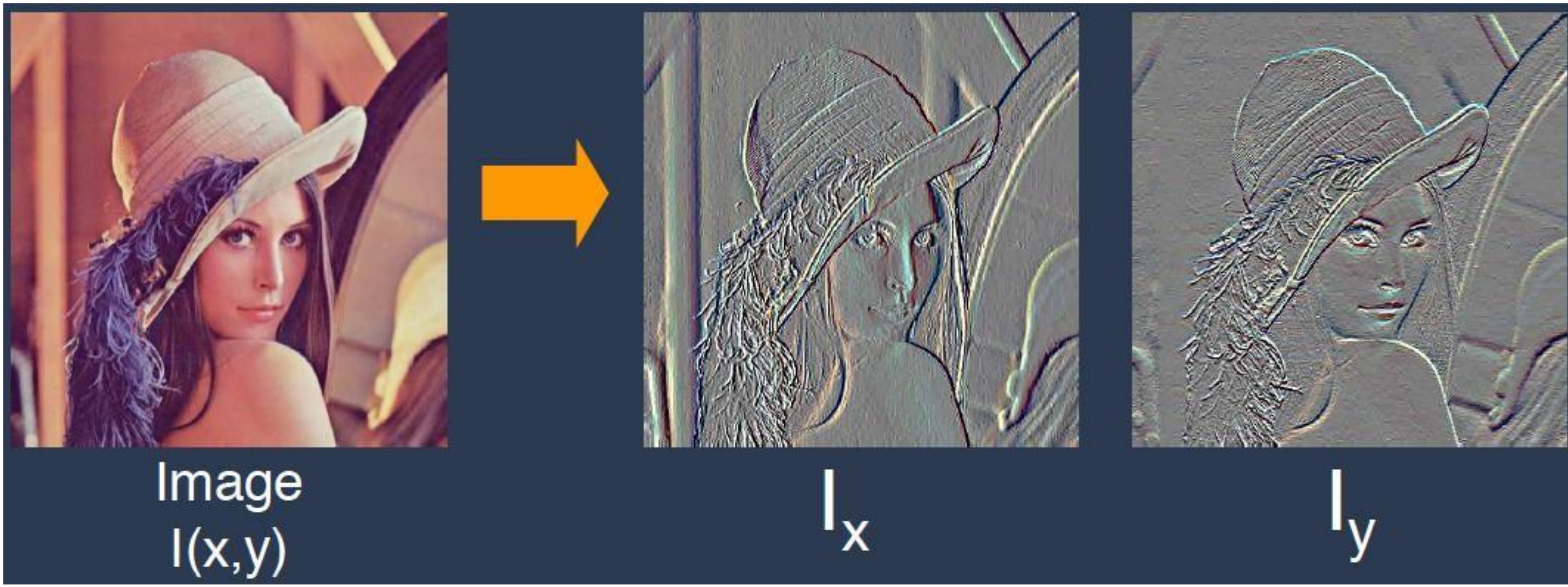
y

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

x

$$g_x = (z_8 - z_5) \text{ and } g_y = (z_6 - z_5)$$

Visualizing Gradient



0	0	0
0	-1	0
0	1	0

0	0	0
0	-1	1
0	0	0

Magnitude of the Gradient - Approximation

- **Robert's cross-gradient Operator**: $M(x,y)=\sqrt{g_x^2+g_y^2}$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$g_x=(z_9-z_5) \quad \text{and} \quad g_y=(z_8-z_6)$$

So,

$$M(x,y)=\left[(z_9-z_5)^2+(z_8-z_6)^2\right]^{1/2}$$

OR

$$M(x,y)\approx |z_9-z_5|+|z_8-z_6|$$

- **Even-sized** filter masks:

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

Robert's cross-gradient operators

- **But, we want odd size filter masks !!!.**

Magnitude of the Gradient - Approximation

Sobel's Operator:

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

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

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

$$M(x,y) = \left[g_x^2 + g_y^2\right]^{\frac{1}{2}} = \left[\left[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right]^2 + \left[(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right]^2\right]^{\frac{1}{2}}$$

We use a weight value of **2** in the center coefficient to achieve some smoothing by giving more importance to the center point.

Application of Sobel's Operator

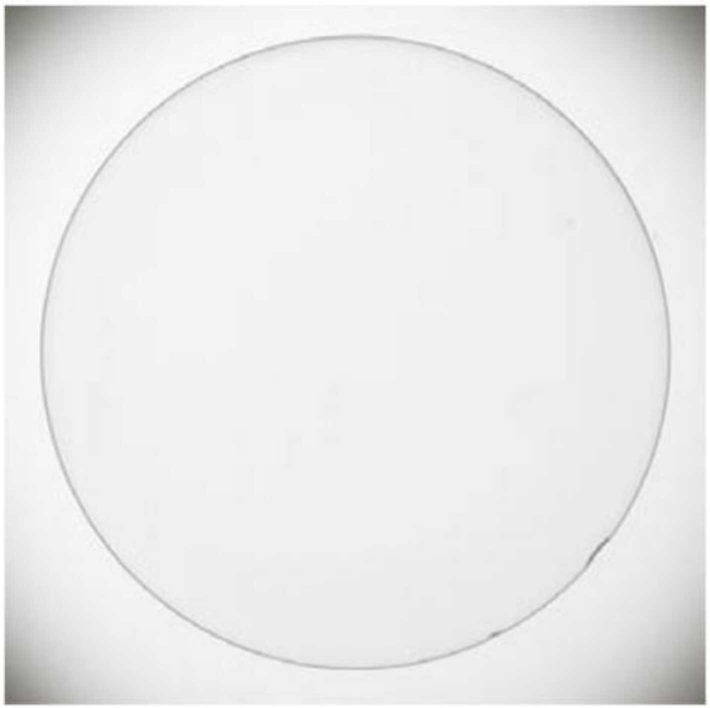
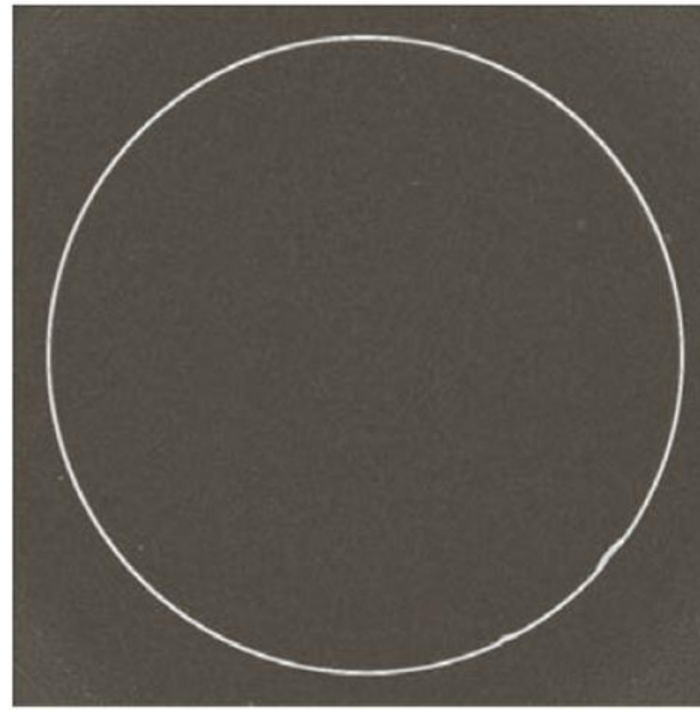


Image of a contact lens (note defects on the boundary at 4 and 5 o'clock)



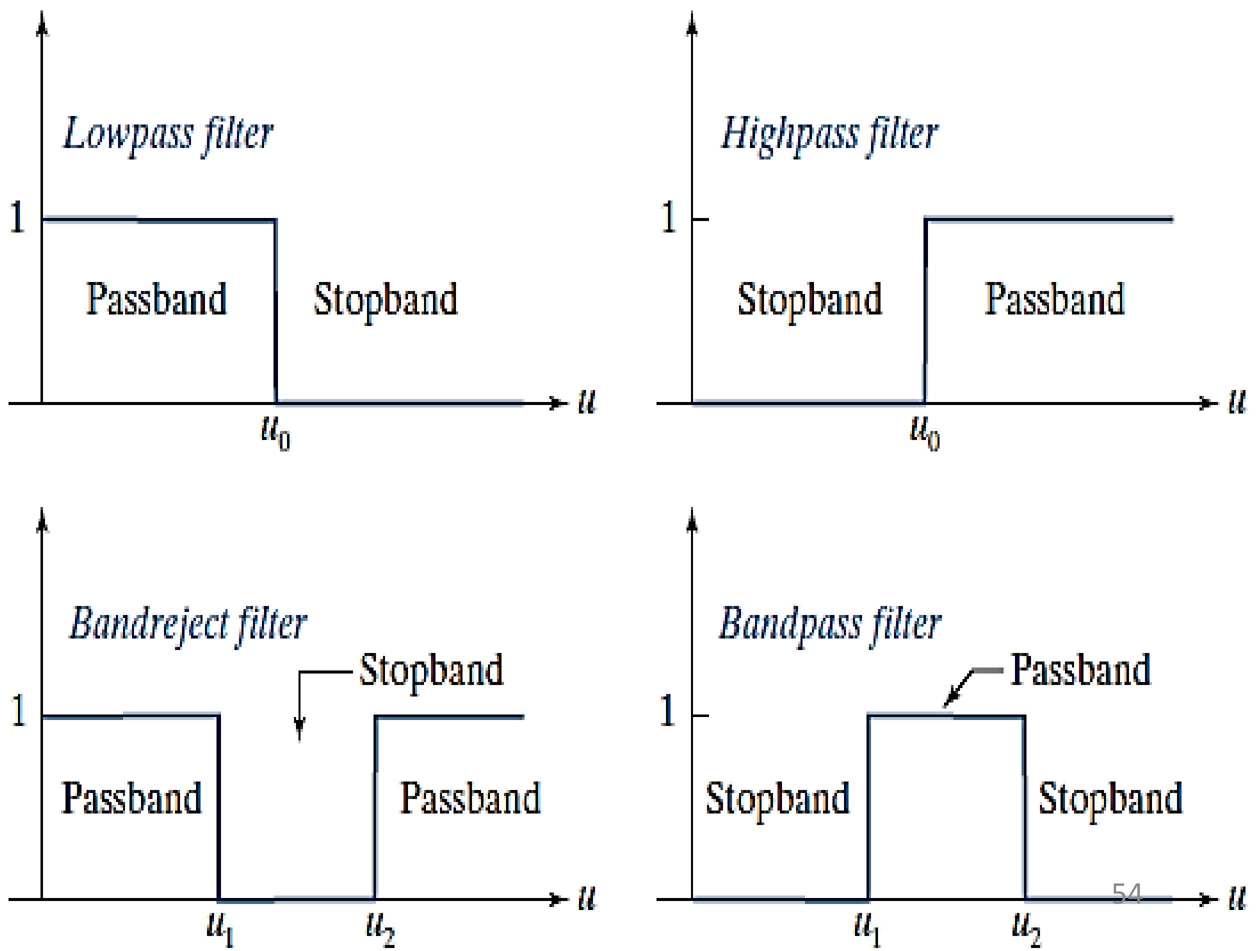
Sobel gradient

Deriving Highpass, Bandreject, and Bandpass filters from lowpass filters

Linear Filter Classifications

- The *lowpass filters* can be used to construct all the following filters without requiring any extra efforts:

1. Highpass filters
2. Bandpass filters
3. Bandreject filters



Construction of *Highpass* and *Lowpass* filters

- A *highpass filter* behaves in exactly the opposite manner of a *lowpss filter* by **attenuating** or **deleting** *low frequencies.*

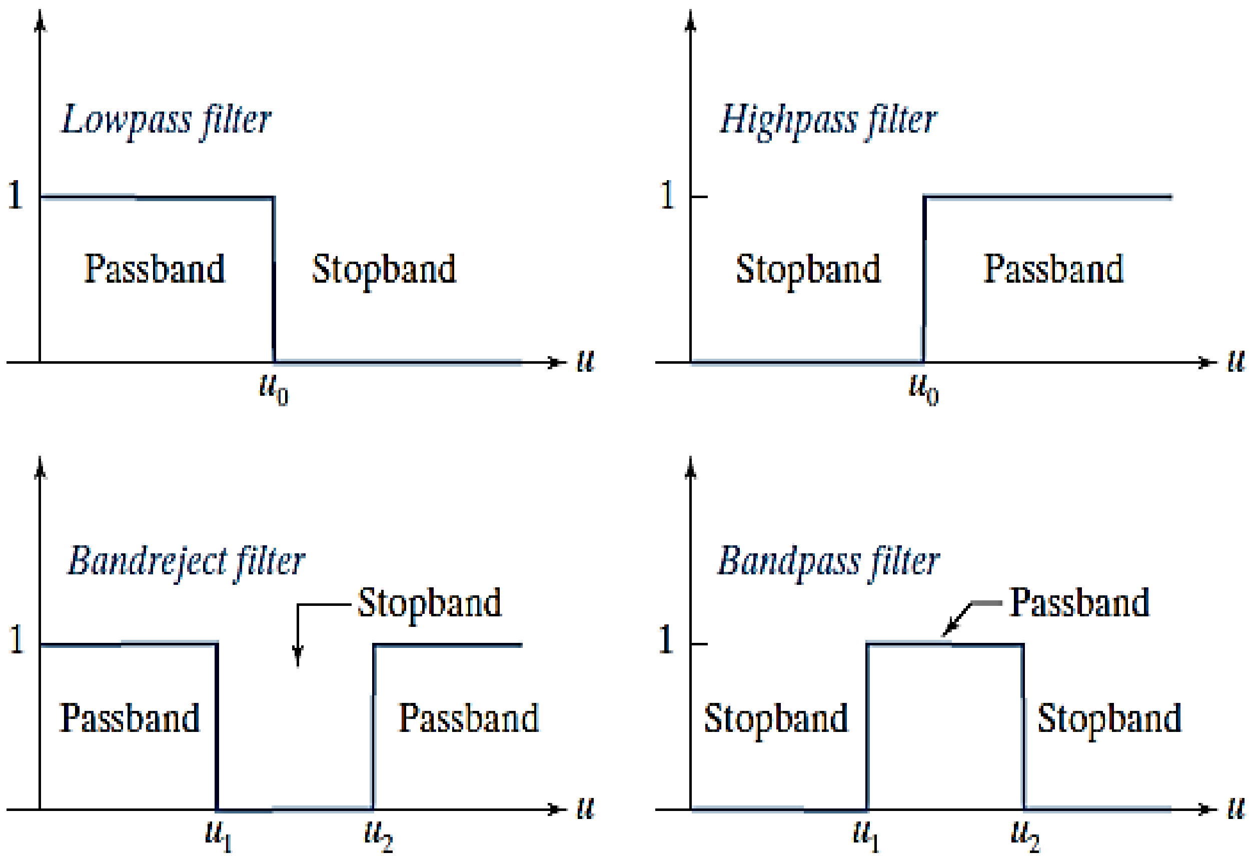
- We obtain a *highpass filter* in the spatial domain by **subtracting** a *lowpass filter* from a **unit impulse** with the *same center as the kernel* and **Vice versa**.

$$\underbrace{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}}_{\text{Unit impulse}} - \underbrace{\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}}_{\text{Highpass filter}} = \underbrace{\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}}_{\text{Lowpass filter}}$$

$$\underbrace{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}}_{\text{Unit impulse}} - \underbrace{\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}}_{\text{Lowpass filter}} = \underbrace{\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}}_{\text{Highpass filter}}$$

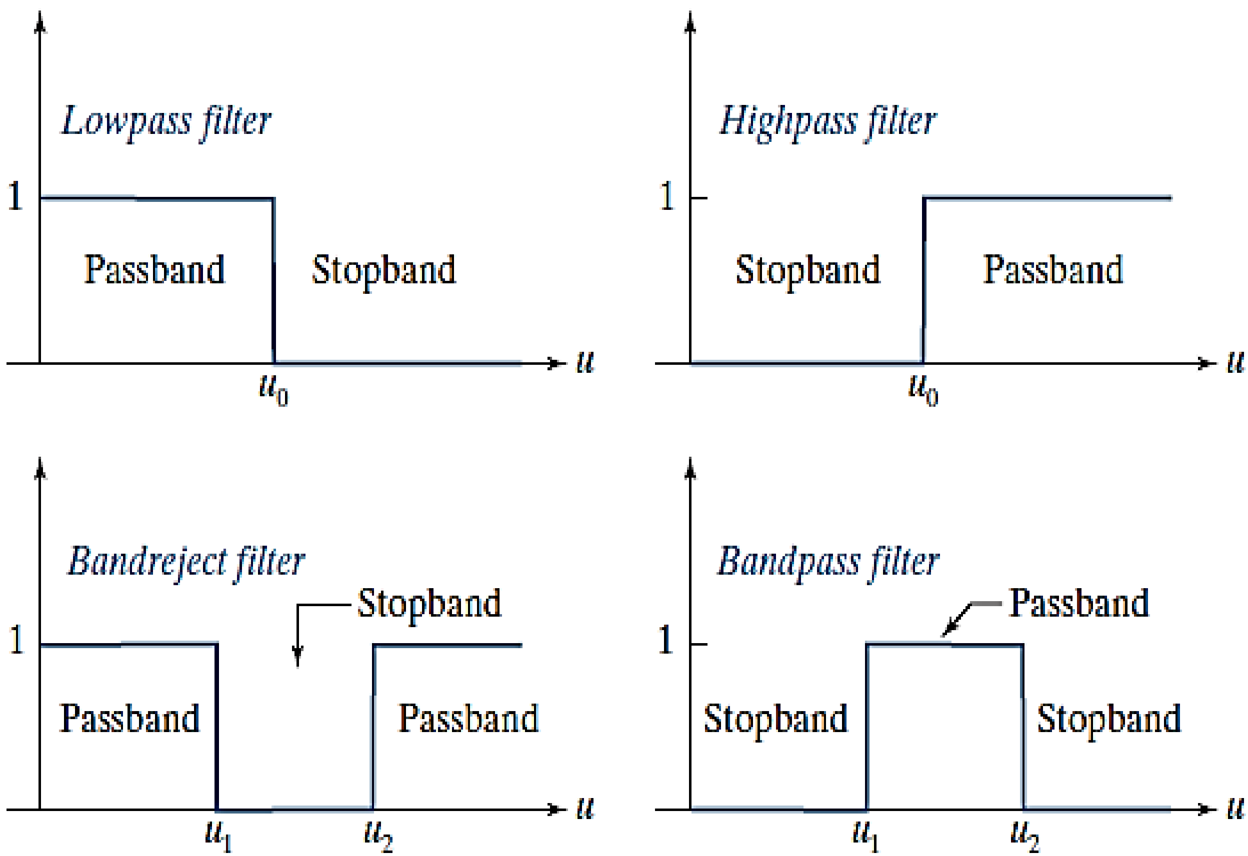
Construction of *Bandreject* filter

- A *bandreject (notch) filter* can be constructed from the **sum of** a *lowpass filter* and a *highpass filter* with ***<u>different cut-off frequencies</u>***.



Construction of *Bandpass* filter

- A *bandpass filter kernel* can be constructed from **subtracting** a **unit impulse** from a *bandreject filter* with the ***same center as the kernel***.

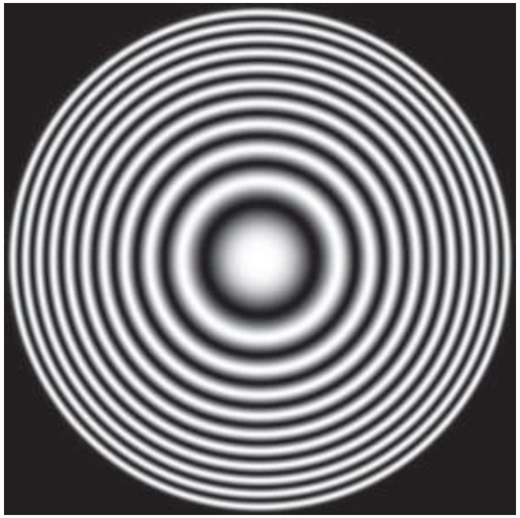


Spatial filters expressed in terms of *Lowpass filters*

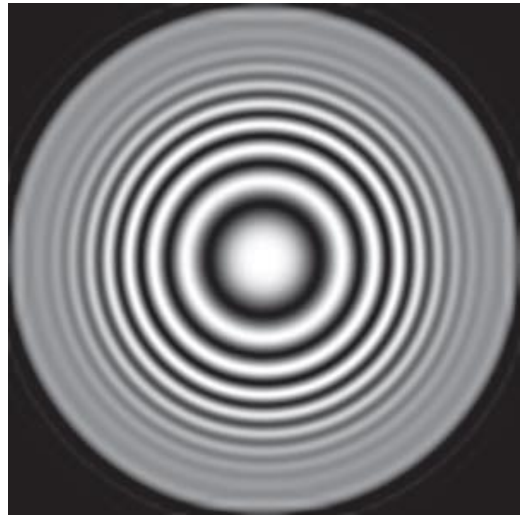
Filter type	Spatial kernel in terms of lowpass kernel, *lp*
Lowpass	$lp(x,y)$
Highpass	$hp(x,y) = \delta(x,y) - lp(x,y)$
Bandreject	$br(x,y) = lp_1(x,y) + hp_2(x,y)$ $= lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]$
Bandpass	$bp(x,y) = \delta(x,y) - br(x,y)$ $= \delta(x,y) - [lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]]$

- $\boldsymbol{\delta}(\boldsymbol{x},\boldsymbol{y})$ is a unit impulse function.
- The centres of the unit impulse and the filter kernels **must coincide** to derive other filters.

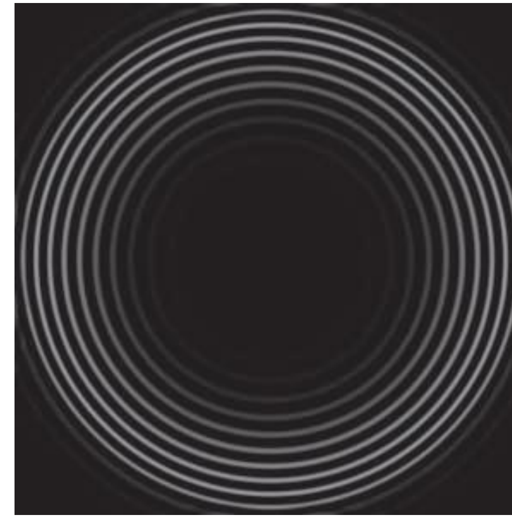
Spatial filters - Illustration



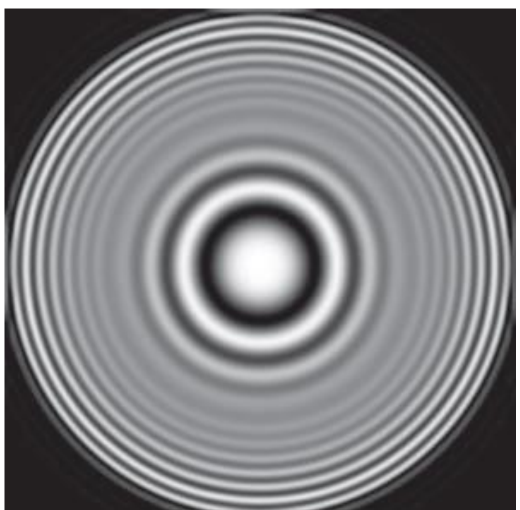
Input image (Zone plate)



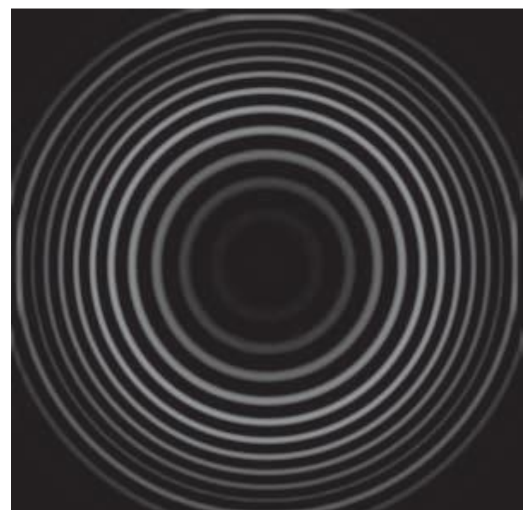
Lowpass result



Highpass result



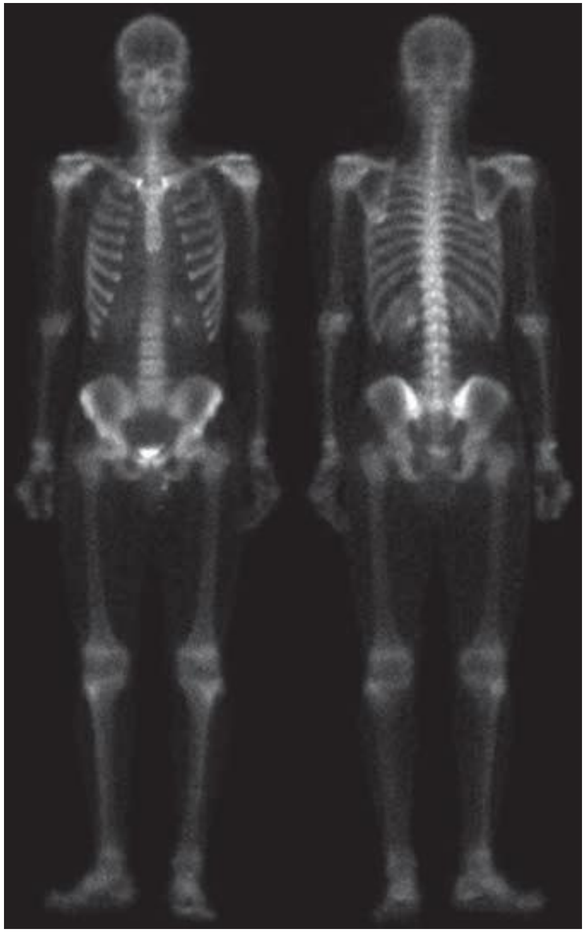
Bandreject result



Bandpass result

Combining Special Enhancement Methods

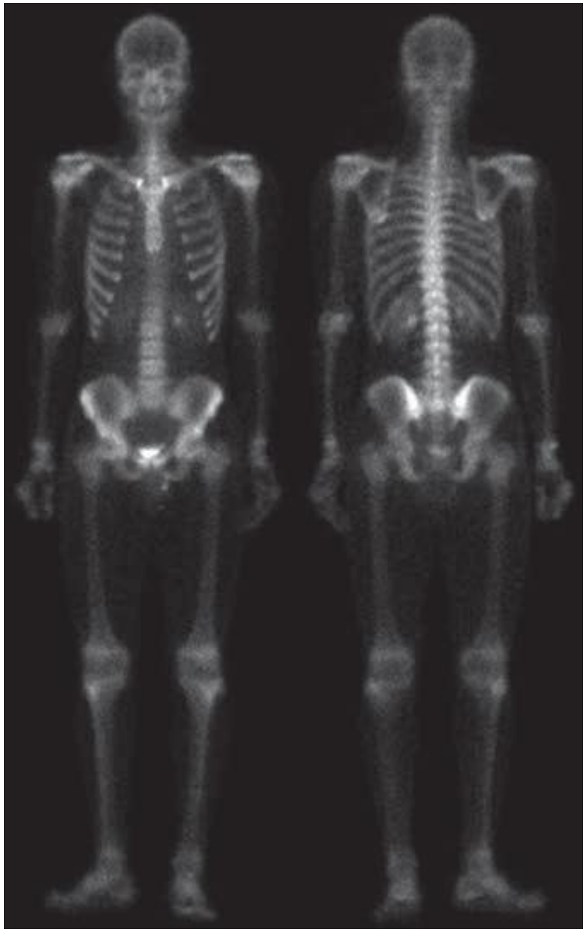
Combining Special Enhancement Methods



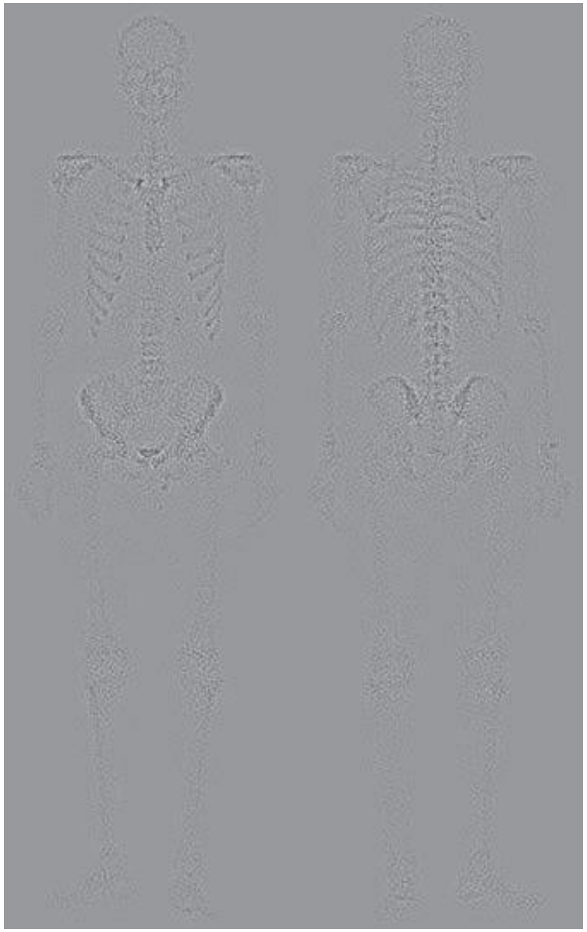
Nuclear whole body bone scan, used to detect diseases such as bone infections and tumors

- Our objective is to enhance this image by sharpening it and by *bringing out more details*.
- The *narrow dynamic range of the intensity levels* and *high noise content* make this image difficult to enhance.
- We use the following strategy to enhance this image:
  1. Utilize the *Laplacian* to highlight fine detail.
  2. Use *Gradient* to enhance prominent edges.
  3. Use the *smoothed version* of the gradient image to *mask* the Laplacian image.
  4. Finally, *increase the dynamic range* of the intensity levels by using an *intensity transformation.*

Combining Special Enhancement Methods

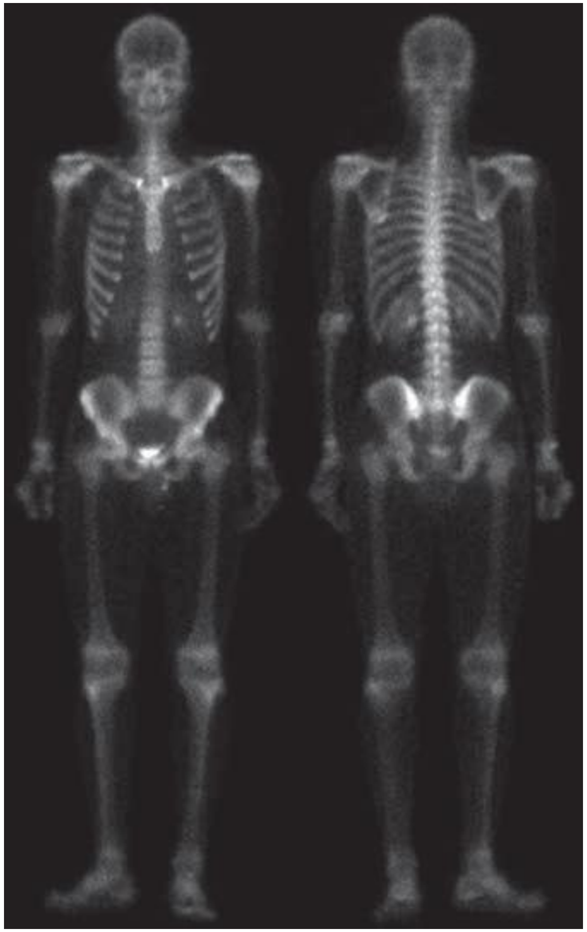


(a) Nuclear whole body bone scan



(b) Laplacian of (a)

Combining Special Enhancement Methods

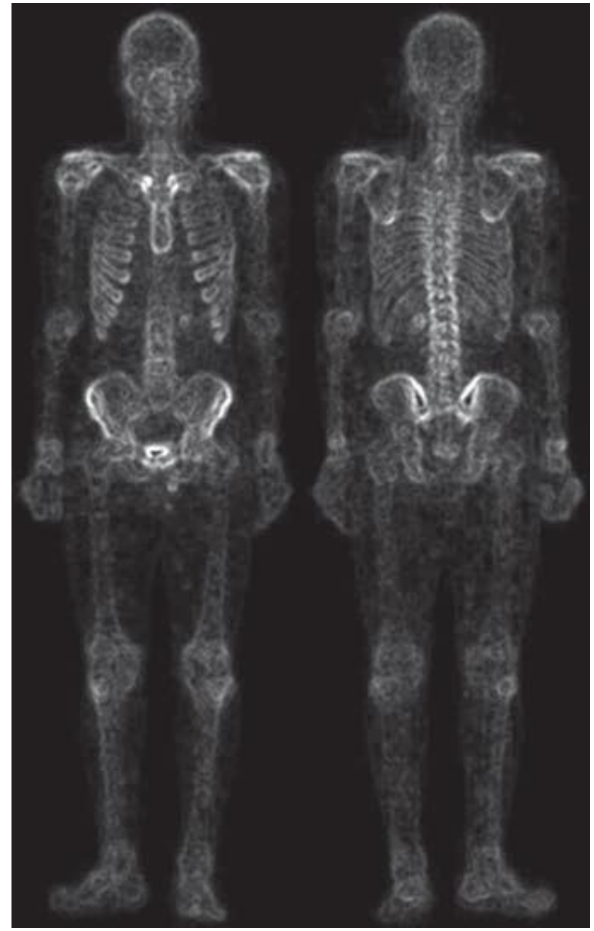


(a) Nuclear whole body bone scan

(c) Sobel gradient of image (a)

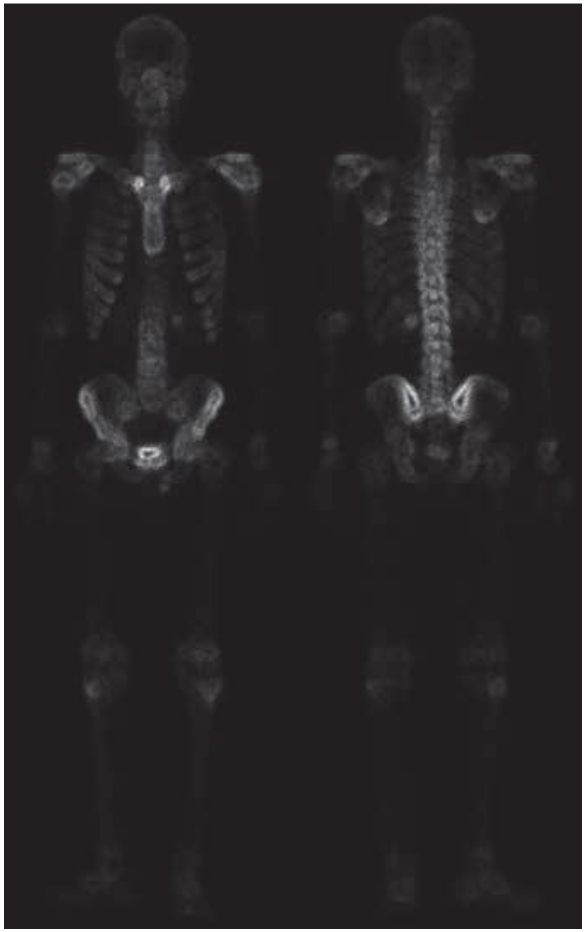
Combining Special Enhancement Methods

(c) Sobel gradient of image (a)

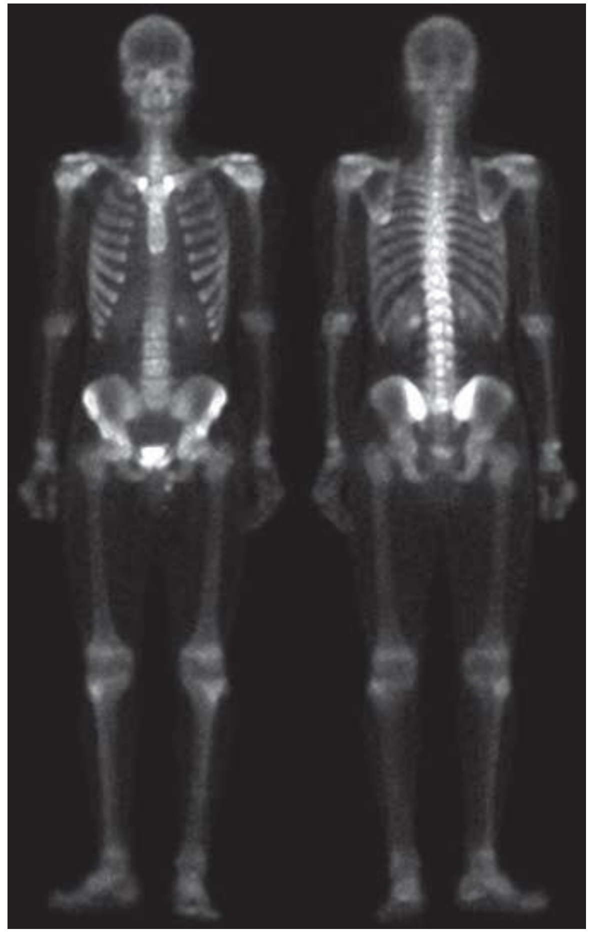


(d) Sobel gradient image (c) smoothed with a 5 × 5 box filter

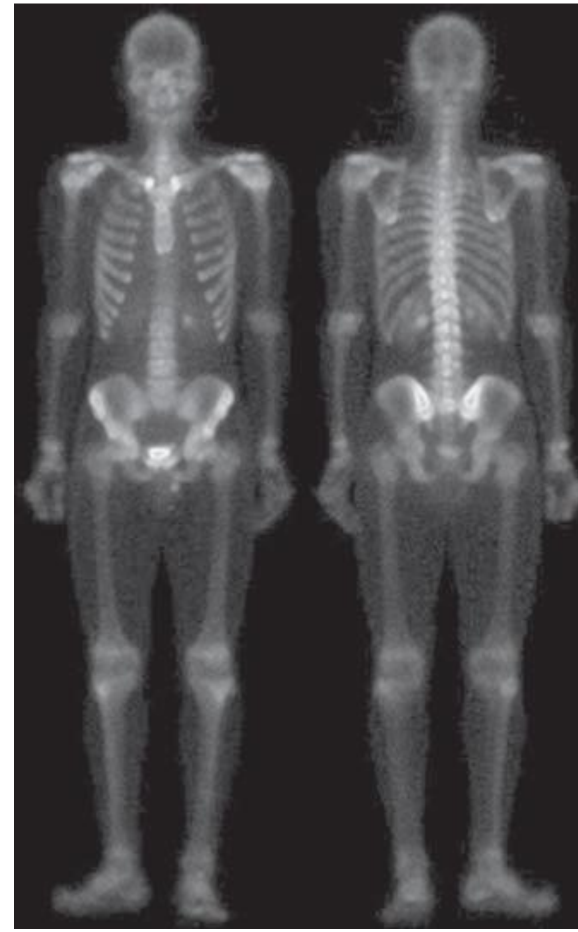
Combining Special Enhancement Methods



(e) Mask image formed by the product of (b) and (d)

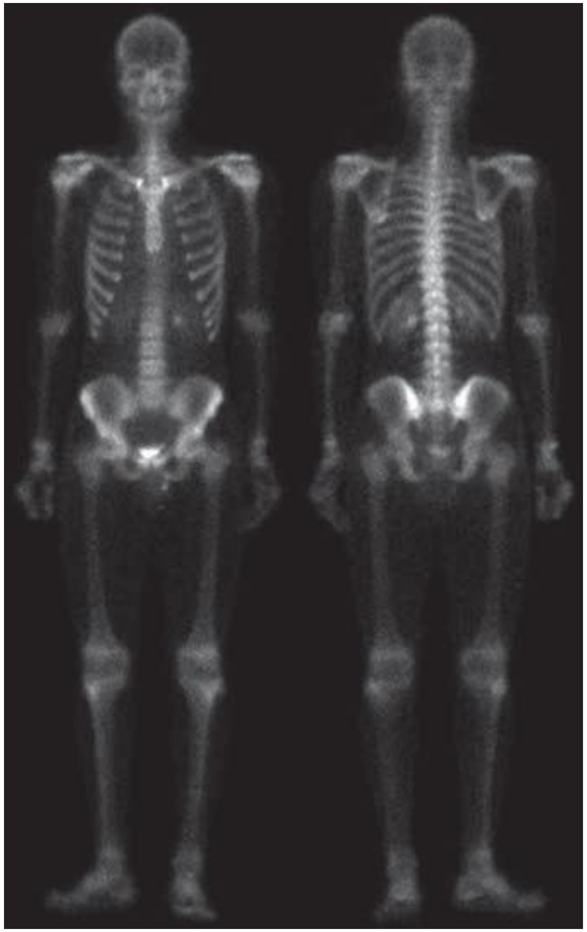


(f) Sharpened image obtained by the adding images (a) and (e)

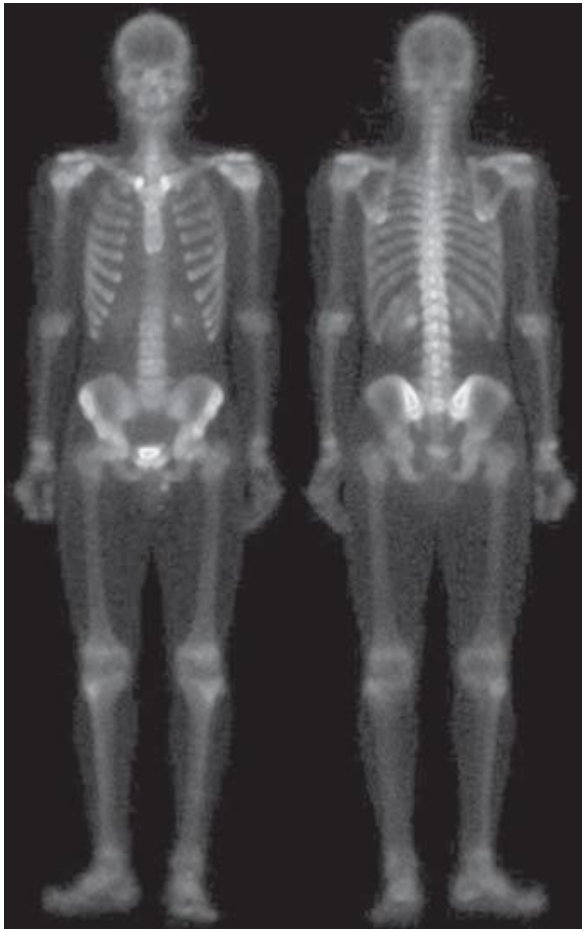


(h) Final result obtained by applying a powerlaw transformation to (f) using γ = 0 5 . and c = 1

Combining Special Enhancement Methods



(a) Original image



(h) Final result

Next Lecture

- Introduction to Frequency Domain
  - Background
  - Sinusoidal Waves
  - Complex Numbers
- Fourier Series
- Impulse
- Fourier Transform
- Convolution of Continuous Functions