Spatial Filtering

Background

- Filter term in "Digital image processing" is referred to the subimage
- There are others term to call subimage such as mask, kernel, template, or window
- The value in a filter subimage are referred as coefficients, rather than pixels.

Basics of Spatial Filtering

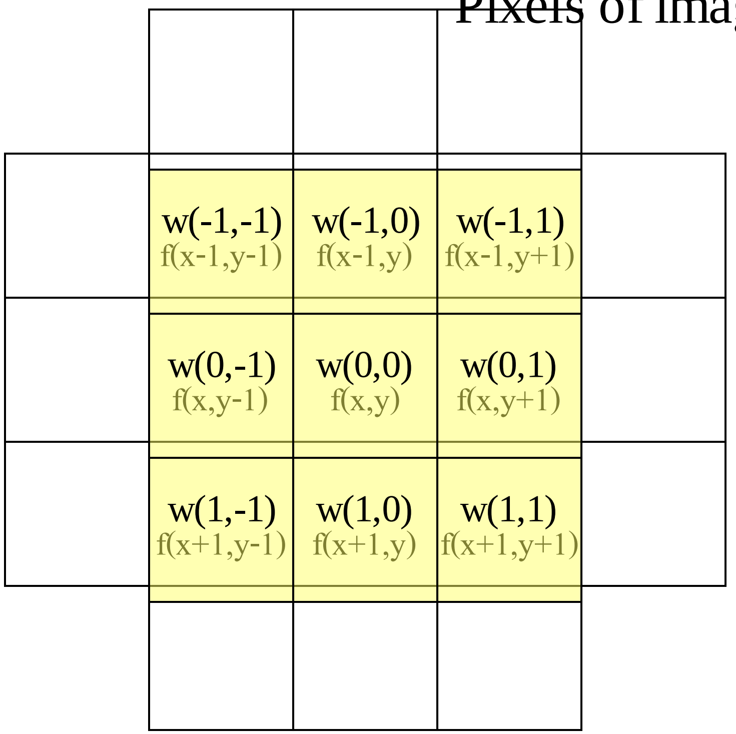
- The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.
- Spatial filtering term is the filtering operations that are performed directly on the pixels of an image

Mechanics of spatial filtering

- The process consists simply of moving the filter mask from point to point in an image.
- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship

Linear spatial filtering

Pixels of image



The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask

Mask coefficients

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

$$
\begin{aligned}
f(x,y) = \; & w(-1,-1)\, f(x-1, y-1) + w(-1,0)\, f(x-1, y) + w(-1,1)\, f(x-1, y+1) + \\
& w(0,-1)\, f(x, y-1) + w(0,0)\, f(x, y) + w(0,1)\, f(x, y+1) + \\
& w(1,-1)\, f(x+1, y-1) + w(1,0)\, f(x+1, y) + w(1,1)\, f(x+1, y+1)
\end{aligned}
$$

Note: Linear filtering

- The coefficient w(0,0) coincides with image value f(x,y), indicating that the mask is centered at (x,y) when the computation of sum of products takes place.
- For a mask of size mxn, we assume that m-2a+1 and n=2b+1, where a and b are nonnegative integer. Then m and n are odd.

Linear filtering

- In general, linear filtering of an image f of size MxN with a filter mask of size mxn is given by the expression:

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

Discussion

- The process of linear filtering similar to a frequency domain concept called "*convolution*"

Simplify expression

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i$$

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Where the w's are mask coefficients, the z's are the value of the image gray levels corresponding to those coefficients

Nonlinear spatial filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration

Smoothing Spatial Filters

- Smoothing filters are used for blurring and for noise reduction.
  - Blurring is used in preprocessing steps, such as removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves
  - Noise reduction can be accomplished by blurring

Type of smoothing filtering

- There are 2 way of smoothing spatial filters
  - Smoothing Linear Filters
  - Order-Statistics Filters

Smoothing Linear Filters

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- Sometimes called "averaging filters".
- The idea is replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

Two 3x3 Smoothing Linear Filters

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

Standard average

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Weighted average

5x5 Smoothing Linear Filters

$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Smoothing Linear Filters

- The general implementation for filtering an MxN image with a weighted averaging filter of size mxn is given by the expression

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

Result of Smoothing Linear Filters

Original Image

[3x3]

[5x5]

[7x7]

Order-Statistics Filters

- Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- Best-known "median filter"

Process of Median filter

10	15	20
20	100	20
20	20	25

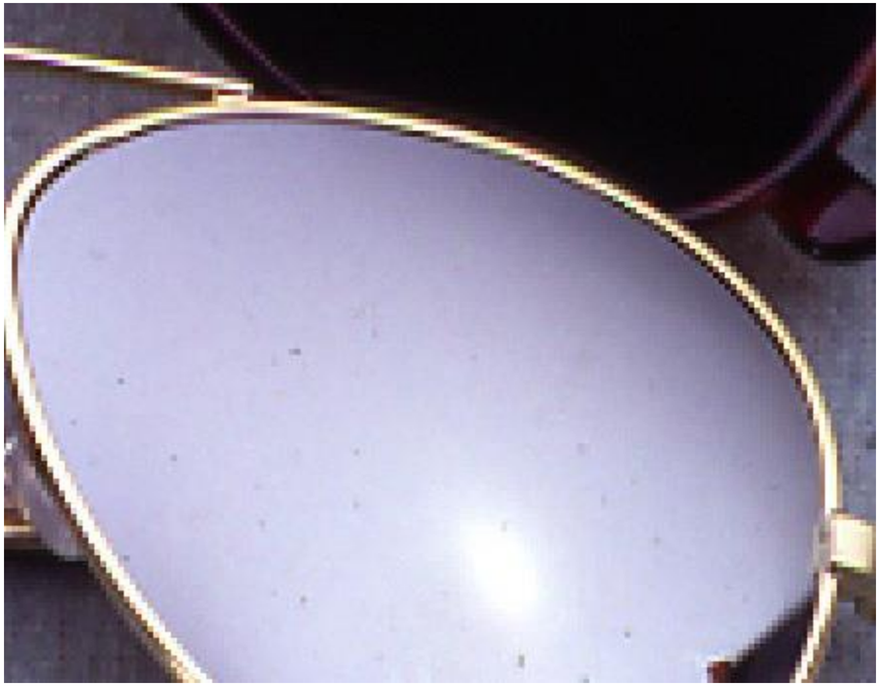
10, 15, 20, 20, 20, 20, 20, 25, 100

↑

5th

- Sort the values of the pixel in the mask.
- In the MxN mask find the median.

Result of median filter



Noise from Glass effect Remove noise by median filter

Sharpening Spatial Filters

- The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as an natural effect of a particular method of image acquisition.

Introduction

- The image blurring is accomplished in the spatial domain by pixel averaging in a neighborhood.
- Since averaging is analogous to integration.
- Sharpening could be accomplished by spatial differentiation.

Foundation

- We are interested in the behavior of these derivatives in areas of constant gray level(flat segments), at the onset and end of discontinuities(step and ramp discontinuities), and along gray-level ramps.
- These types of discontinuities can be noise points, lines, and edges.

Definition for a first derivative

- Must be zero in flat segments
- Must be nonzero at the onset of a gray-level step or ramp; and
- Must be nonzero along ramps.

Definition for a second derivative

- Must be zero in flat areas;
- Must be nonzero at the onset and end of a gray-level step or ramp;
- Must be zero along ramps of constant slope

Definition of the 1st-order derivative

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

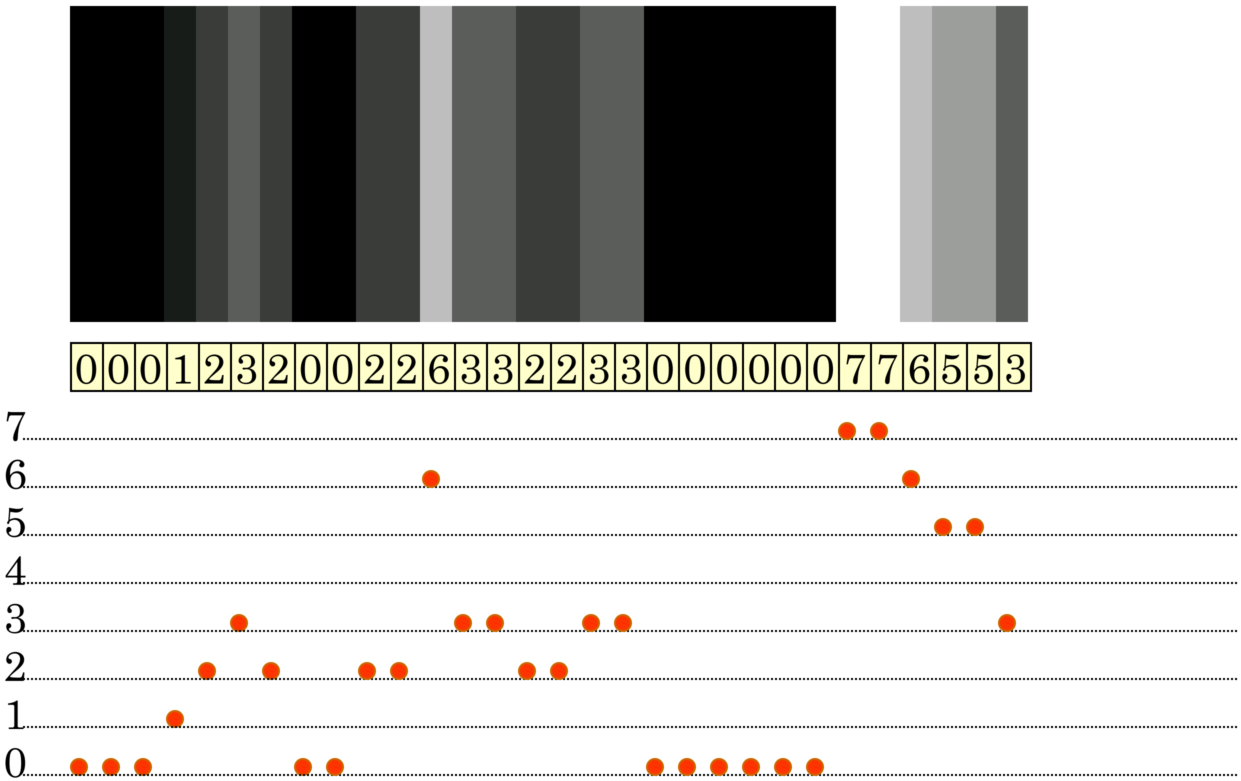
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Definition of the 2nd-order derivative

- We define a second-order derivative as the difference

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

Gray-level profile



Derivative of image profile

	0 0 0 1 2 3 2 0 0 2 2 6 3 3 2 2 3 3 0 0 0 0 0 0 7 7 6 5 5 3
first	0 0 1 1 1 -1 -2 0 2 0 4 -3 0 -1 0 1 0 -3 0 0 0 0 0 -7 0 -1 -1 0 -2
second	0 -1 0 0 -2 -1 2 2 -2 4 -7 3 -1 1 1 -1 -3 3 0 0 0 0 -7 7 -1 0 1 -2

Analyze

- The 1^{st}-order derivative is nonzero along the entire ramp, while the 2^{nd}-order derivative is nonzero only at the onset and end of the ramp.

  1^{st} make thick edge and 2^{nd} make thin edge

- The response at and around the point is much stronger for the 2^{nd}- than for the 1^{st}-order derivative

The Laplacian (2nd order derivative)

- Shown by Rosenfeld and Kak[1982] that the simplest isotropic derivative operator is the Laplacian is defined as

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

Discrete form of derivative

f(x-1,y)	*f(x,y)*	*f(x+1,y)*

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

f(x,y-1)
f(x,y)
f(x,y+1)

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

2-Dimentional Laplacian

- The digital implementation of the 2-Dimensional Laplacian is obtained by summing 2 components

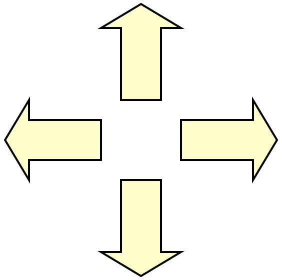
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

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

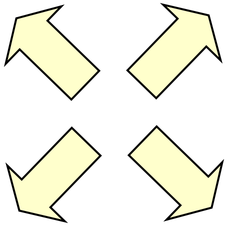
	1	
1	-4	1
	1	

Laplacian

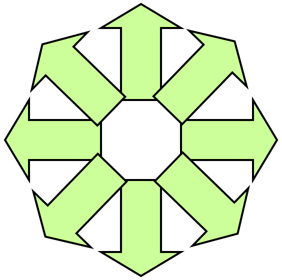
0	1	0
1	-4	1
0	1	0



1	0	1
0	-4	0
1	0	1

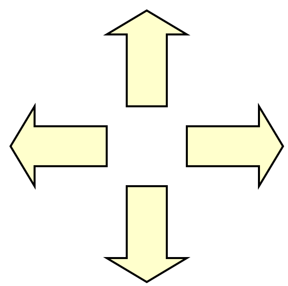


1	1	1
1	-8	1
1	1	1



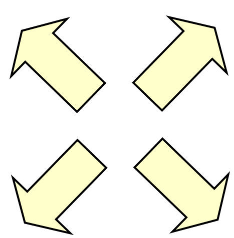
Laplacian

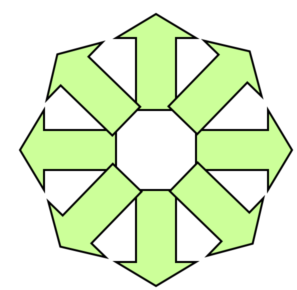
0	-1	0
-1	4	-1
0	-1	0



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

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





Implementation

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

Where f(x,y) is the original image

$\nabla^2 f(x,y)$ is Laplacian filtered image

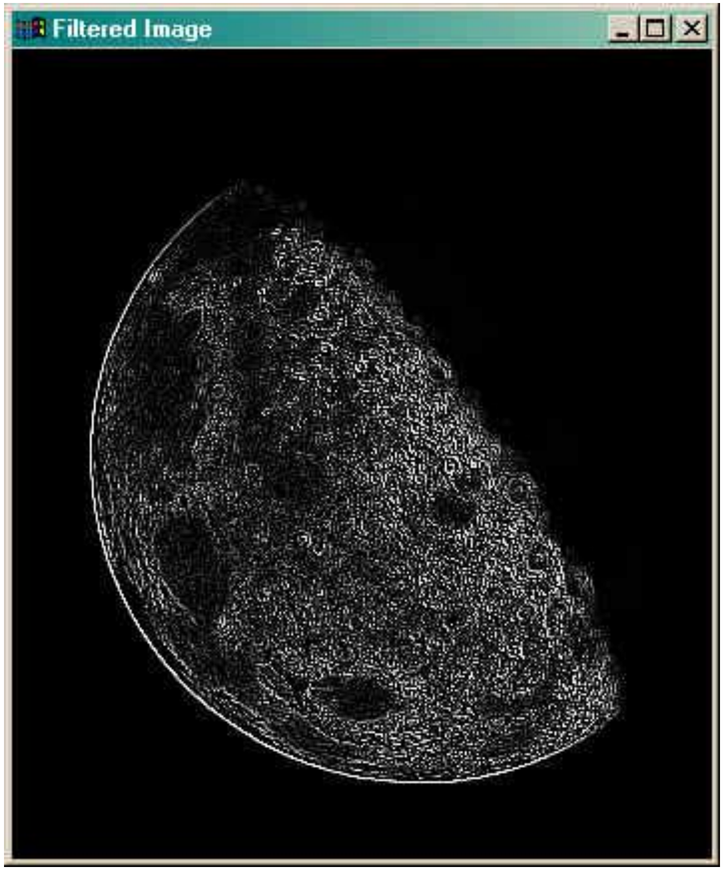
g(x,y) is the sharpen image

Implementation

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

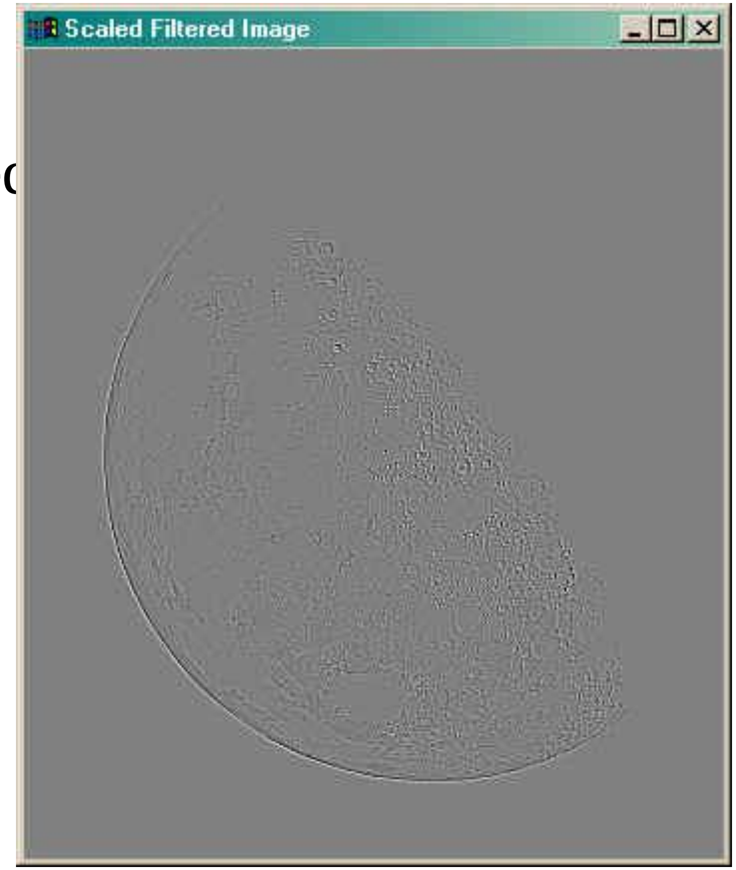
Implementation

Filtered = Conv(image,mask)



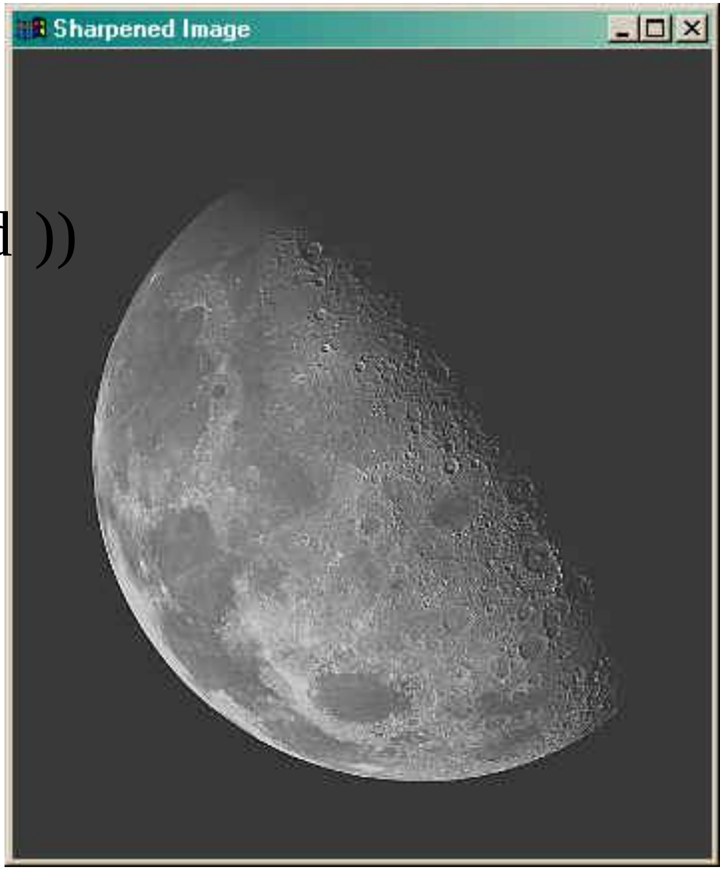
Implementation

filtered = filtered - Min(filtered)
filtered = filtered * (255.0/Max(filtered



Implementation

sharpened = image + filtered
sharpened = sharpened - Min(sharpened)
sharpened = sharpened * (255.0/Max(sharpened))



Algorithm

- Using Laplacian filter to original image
- And then add the image result from step 1 and the original image

Simplification

- We will apply two step to be one mask

$$g(x,y) = f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) + 4f(x,y)$$

$$g(x,y) = 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$$

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Unsharp masking

- A process to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

Where $f_s(x, y)$ denotes the sharpened image obtained by unsharp masking, and $\bar{f}(x, y)$ is a blurred version of $f(x, y)$

High-boost filtering

- A high-boost filtered image, f_{hb} is defined at any point (x,y) as

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \quad where\ A \geq 1$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f_s(x,y)$$

This equation is applicable general and does not state explicity how the sharp image is obtained

High-boost filtering and Laplacian

- If we choose to use the Laplacian, then we know $f_s(x,y)$

$$f_{hb} = \begin{cases} Af(x,y) - \nabla^2 f(x,y) & \textit{If the center coefficient is negative} \\ Af(x,y) + \nabla^2 f(x,y) & \textit{If the center coefficient is positive} \end{cases}$$

0	-1	0
-1	A+4	-1
0	-1	0

-1	-1	-1
-1	A+8	-1
-1	-1	-1

The Gradient (1st order derivative)

- First Derivatives in image processing are implemented using the magnitude of the gradient.
- The gradient of function f(x,y) is

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient

- The magnitude of this vector is given by

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

G_x

-1	1

This mask is simple, and no isotropic. Its result only horizontal and vertical.

G_y

1
-1

Robert's Method

- The simplest approximations to a first-order derivative that satisfy the conditions stated in that section are

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$G_x = (z_9 - z_5)$ and $G_y = (z_8 - z_6)$

$$\nabla f = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Robert's Method

- These mask are referred to as the Roberts cross-gradient operators.

-1	0
0	1

0	-1
1	0

Sobel's Method

- Mask of even size are awkward to apply.
- The smallest filter mask should be 3x3.
- The difference between the third and first rows of the 3x3 image region approximate derivative in x-direction, and the difference between the third and first column approximate derivative in y-direction.

Sobel's Method

- Using this equation

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

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

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