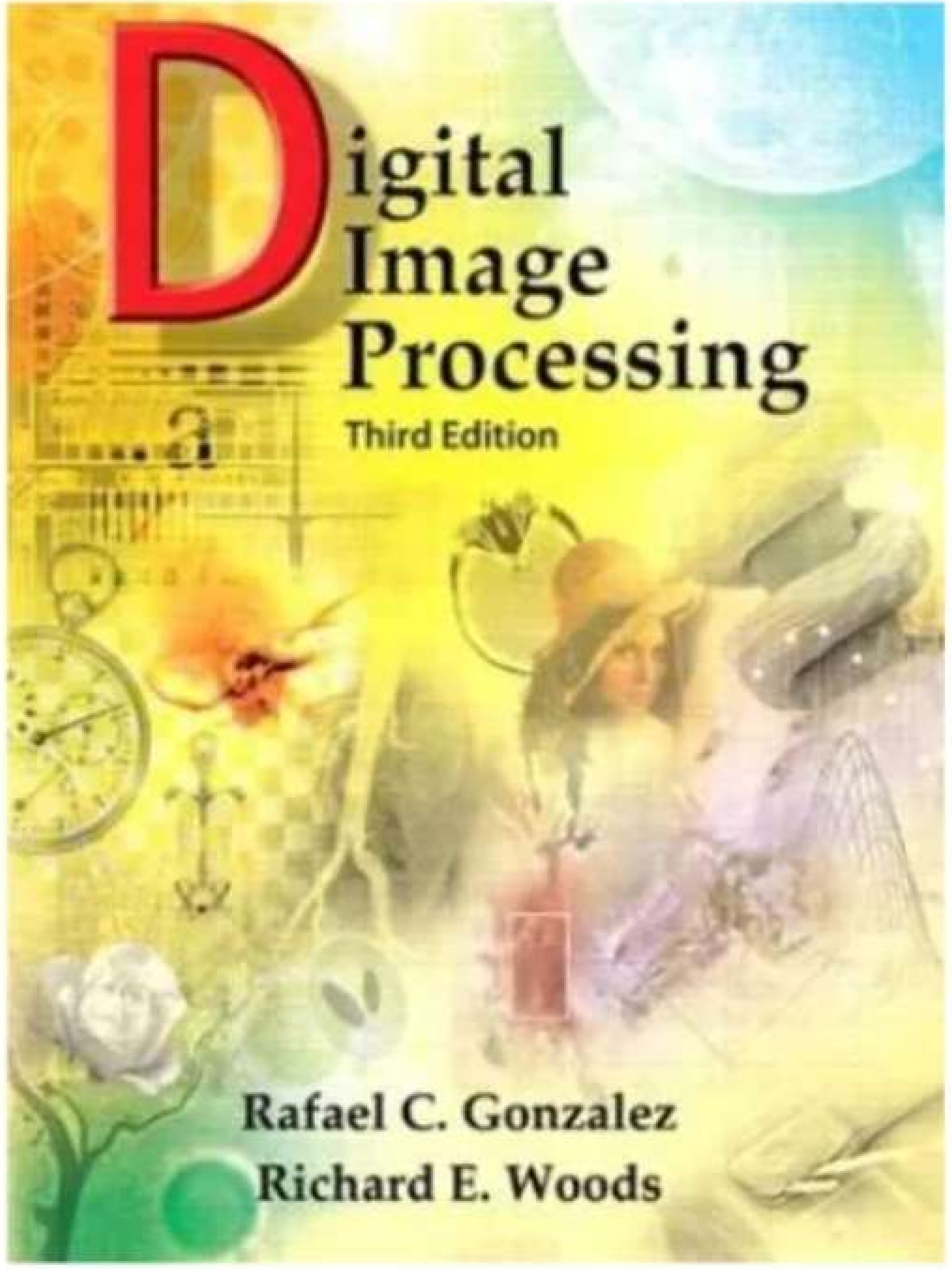
Contents…

- Smoothing spatial filters:
  - Smoothing linear,
  - Order statistic filters,
- Sharpening spatial filters:
  - Use of second derivatives for enhancement,
  - Use of first derivatives for enhancement



Introduction

Introduction

- Smoothing filters are used for blurring and for noise reduction.
- Blurring is used in preprocessing tasks, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves.
- Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

Smoothing Linear Filters

Smoothing Linear Filters

- Smoothing, linear spatial filter computes the average of the pixels contained in the neighborhood of the filter mask.

- These filters sometimes are called *averaging filters and sometimes they are also referred as lowpass filters.*

- During such filtration, the value of every pixel in an image is replaced by the average of the intensity levels in the neighborhood defined by the filter mask.

Smoothing Linear Filters

- This results in an output image with reduced “sharp” transitions in intensities.
- Random noise typically consists of sharp transitions in intensity levels, hence smoothing linear filter is used for noise reduction.
- Averaging filters have the undesirable side effect that they blur edges.

Smoothing Linear Filters

- Figure shows 3 x 3 smoothing filter.

- Use of this filter yields the standard average of the pixels under the mask.

- Equation:

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

which is the average of the intensity levels of the pixels in the 3 x 3 neighborhood defined by the mask.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

- The coefficients of the filter are all 1s to make it computationally more efficient.

- At the end of the filtering process the entire image is divided by 9.

- An m x n mask would have a normalizing constant equal to $1/mn$.

- A spatial averaging filter in which all coefficients are equal is called a box filter.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

- This mask uses *weighted average, (pixels are multiplied by* different coefficients i.e. giving more importance (weight) to some pixels at the expense of others).

- In the mask shown, the pixel at the center is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation.

- The other pixels are inversely weighted as a function of their distance from the center of the mask.

- The diagonal terms are further away from the center than the orthogonal neighbors and, thus, they are weighed less than the immediate neighbors of the center pixel.

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

Smoothing Linear Filters

- Highest weighing given to the center and the value of the coefficients are decreased as a function of increasing distance is an attempt to reduce blurring in the smoothing process.

- The sum of all the coefficients in the mask of figure is equal to 16, an attractive feature for computer implementation because it is an integer power of 2.

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

- With reference to earlier equation, the general implementation for filtering an M x N image with a weighted averaging filter of size m x n (*m and n odd) is* given by the expression

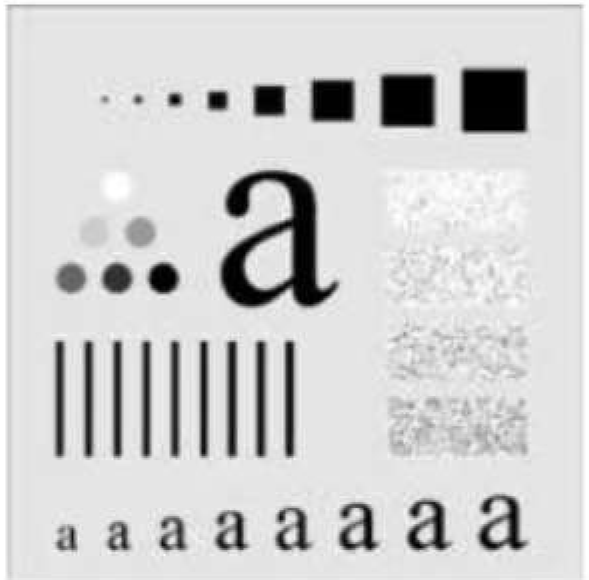
$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}$$

Where, $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$

The denominator in equation is simply the sum of the mask coefficients and, therefore, it is a constant that needs to be computed only once.

Smoothing Linear Filters

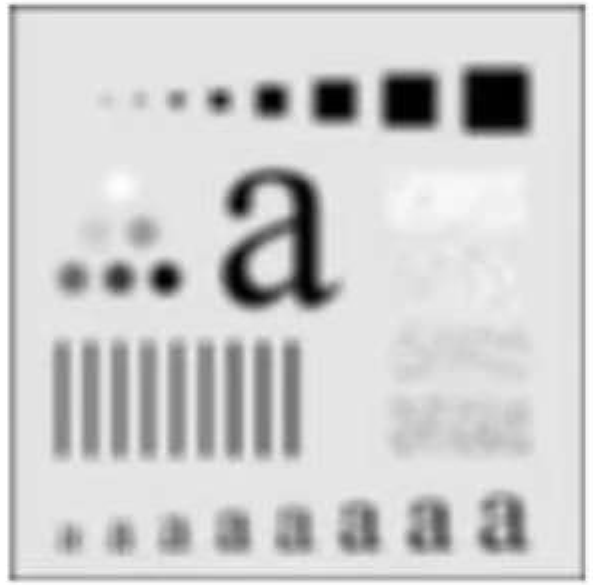
Original image, of size 500 x 500 pixels



Result of averaging filter masks of size m = 3

Result of averaging filter masks of size m = 5

Result of averaging filter masks of size m = 9



Result of averaging filter masks of size m = 15

Result of averaging filter masks of size m = 35

Smoothing Linear Filters

Image of size
528 x 485 pixels from
the Hubble Space
Telescope

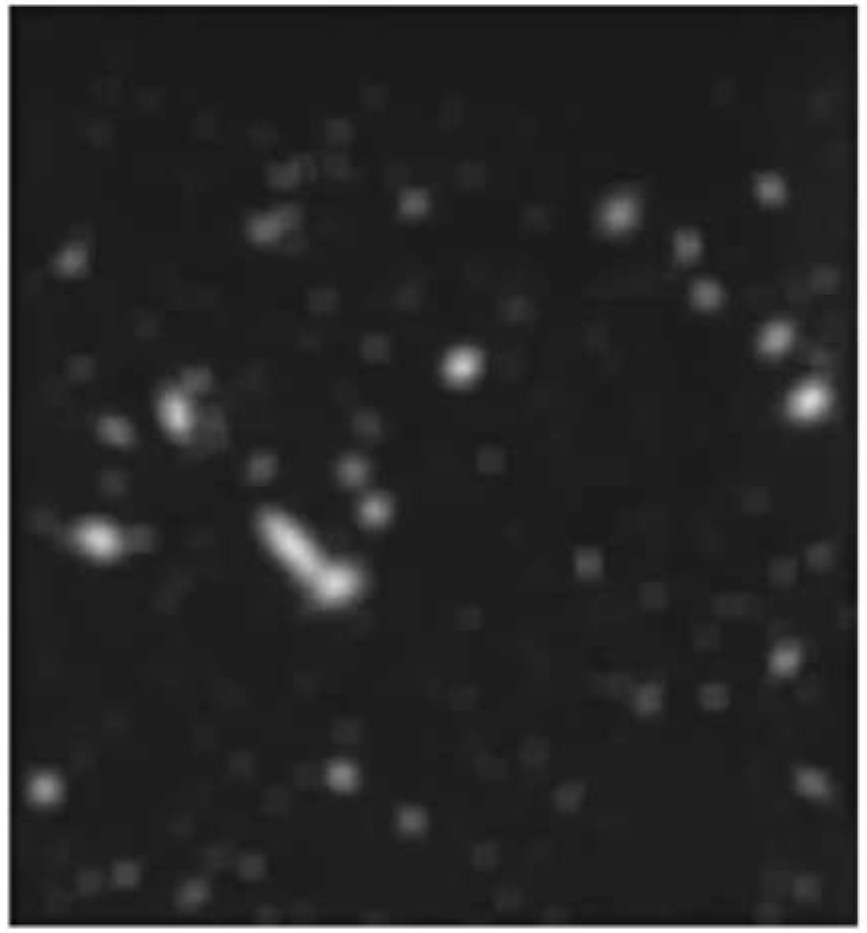


Image filtered with a
15 x 15 averaging mask

Original image courtesy of NASA

Smoothing Nonlinear Filters

Smoothing Nonlinear Filters

- Order-statistic filter is nonlinear spatial filter.

- The response is based on ordering (ranking) the pixels contained in the image encompassed by the filter and then replacing the value of the center pixel with the value determined by the ranking result.

- The best-known filter in this category is the *median filter,* which replaces the value of a pixel by the median of the intensity values in the neighborhood of that pixel.

- Median filters are popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring.

Smoothing Nonlinear Filters

- Median filters are particularly effective in the presence of *impulse noise, (salt-and-pepper noise) because of its appearance as white and black* dots superimposed on an image.

- The median ξ, of a set of values is such that half the values in the set are less than or equal to ξ and half are greater than or equal to ξ .

- In order to perform median filtering at a point in an image, we first sort the values of the pixel in the neighborhood, determine their median, and assign that value to the corresponding pixel in the filtered image.

Smoothing Nonlinear Filters

- **For example:**
- In a 3 x 3 neighborhood the median is the 5th largest value, in a 5 x 5 neighborhood, 13th largest value, and so on.
- When several values in a neighborhood are the same, all equal values are grouped.
- For example, suppose that a 3 x 3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20.
- The principal function of median filters is to force points with distinct intensity levels to be more like their neighbors.

Smoothing Nonlinear Filters

- The median filter is the most useful order-statistic filter in image processing.

- The median represents the 50^{th} percentile of a ranked set of numbers.

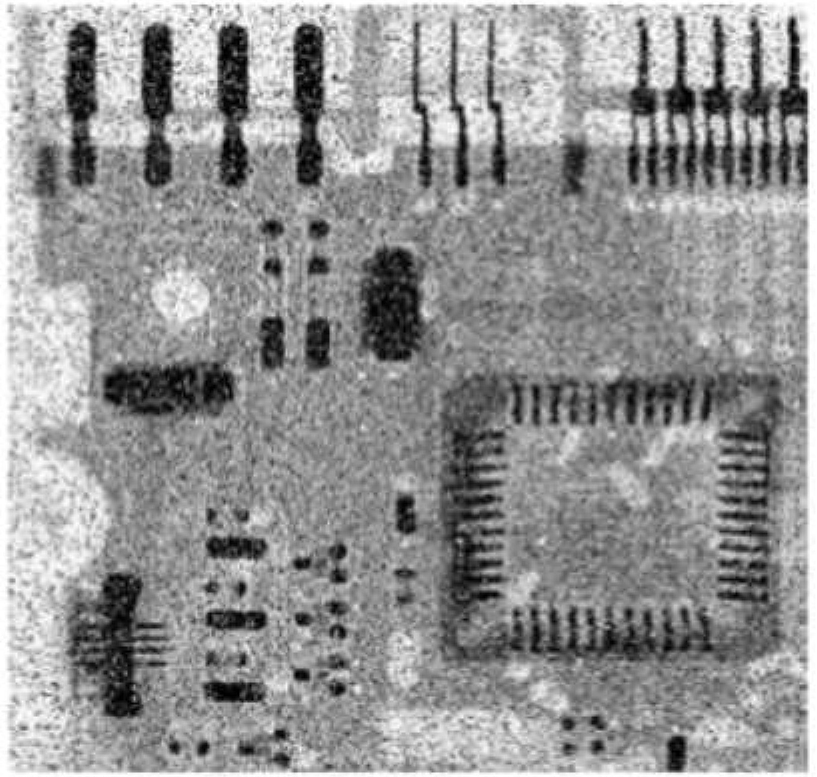
- For example, using the 100^{th} percentile results in the so-called *max filter, which is useful for finding the* brightest points in an image.

- The response of a 3 x 3 max filter is given by

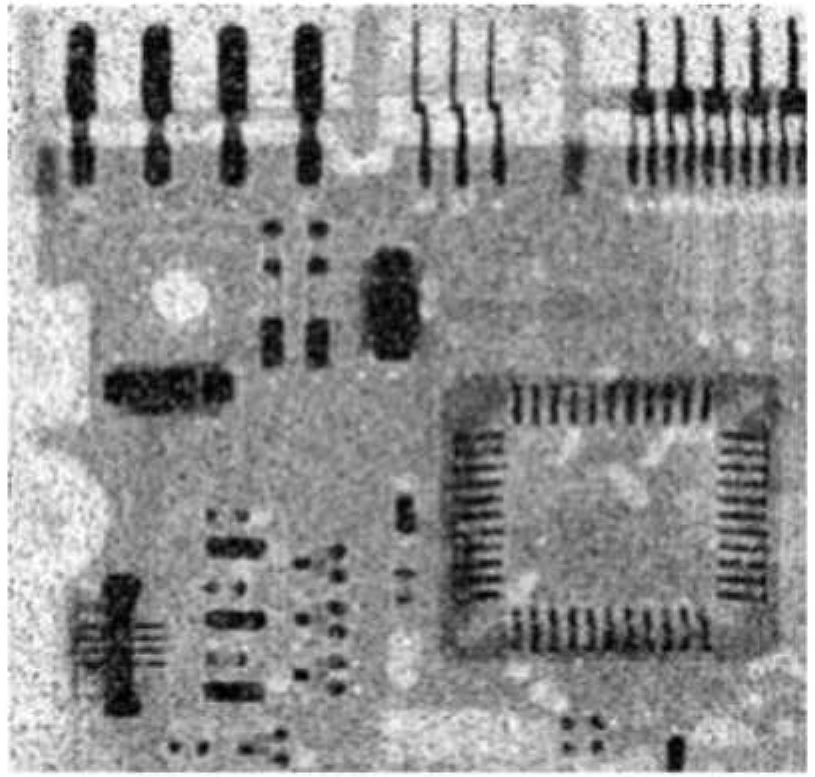
$$R = max\{z_k | \ k = 1, 2, ..., 9\}.$$

- The 0^{th} percentile filter is the *min filter*.

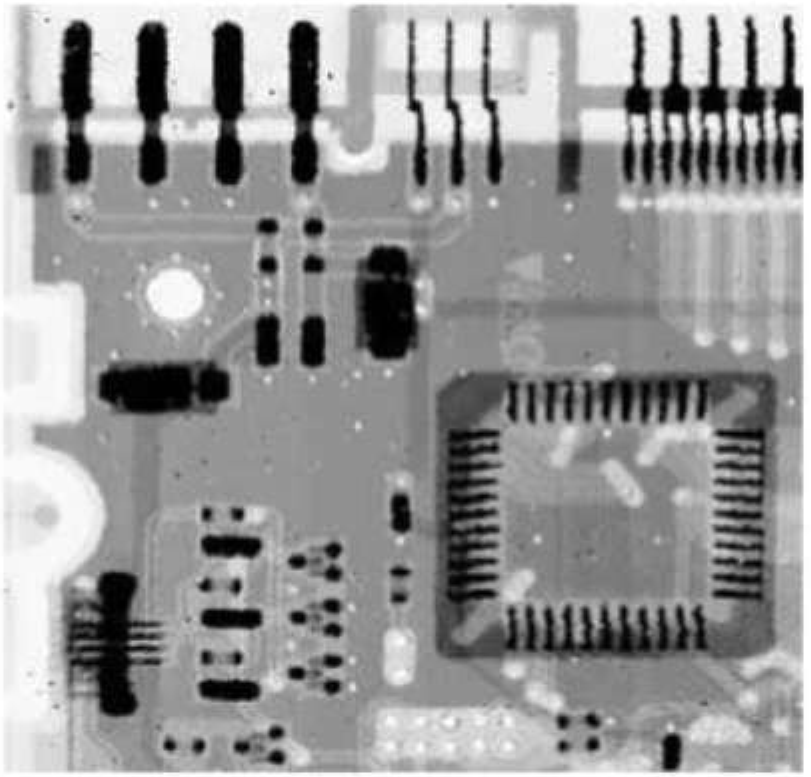
Smoothing Nonlinear Filters



X-ray image of circuit board corrupted by salt-and-pepper noise



Noise reduction with 3 x 3 averaging mask



Noise reduction with a 3 x 3 median filter

The averaging filter blurred the image and its noise reduction performance was poor.
The superiority of median over average filtering is quite good.
In general, median filtering is much better than averaging for the removal of salt-and-pepper noise.

image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.

Sharpening Filters

Sharpening Filters

- Sharpening filters are used is to highlight transitions in intensity.
- Sharpening applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.
- Image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood.
- As averaging is analogous to integration, it is logical to conclude that sharpening can be accomplished by spatial differentiation.
- Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying intensities.

Sharpening Filters

- **Derivatives:**
- The derivatives of a digital function are defined in terms of differences.
- There are various ways to define these differences.
- *First derivative*
  - *(1) must be zero in areas of constant* intensity;
  - (2) must be nonzero at the onset of an intensity step or ramp; and
  - (3) must be nonzero along ramps.

- *Second derivative*
  - (1) must be zero in constant areas;
  - (2) must be nonzero at the onset *and* end of an intensity step or ramp; and
  - (3) must be zero along ramps of constant slope.

Sharpening Filters

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

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

- Second-order derivative of $f(x)$ as the difference

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

Sharpening Filters

- **Second Derivative for Image Sharpening:**

- This approach consists of defining a discrete formulation of the second-order derivative and then constructing a filter mask based on that formulation.
- We are interested in *isotropic filters, whose response is independent of the direction of* the discontinuities in the image to which the filter is applied.
- Isotropic filters are *rotation invariant, in the sense that rotating the image and* then applying the filter gives the same result as applying the filter to the image first and then rotating the result.
- The simplest *isotropic derivative* operator is the Laplacian, which, for a function (image) $f(x,y)$ of two variables, is defined as

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

Sharpening Filters

- **Second Derivative for Image Sharpening:**

- Because derivatives of any order are linear operations, the Laplacian is a linear operator.
- To express this equation in discrete form, 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, similarly, in the *y-direction we have*

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

Therefore, discrete Laplacian of two variables is

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

Sharpening Filters

- **Second Derivative for Image Sharpening:**

0	1	0
1	−4	1
0	1	0

Filter mask used to implement discrete Laplacian of two variables

Isotropic result for rotations in increments of 90^0

1	1	1
1	−8	1
1	1	1

Mask used to implement an extension of this equation that includes the diagonal terms

Isotropic results in increments of 45°

0	−1	0
−1	4	−1
0	−1	0

Implementations of the Laplacian

−1	−1	−1
−1	8	−1
−1	−1	−1

Implementations of the Laplacian

Sharpening Filters

- **Second Derivative for Image Sharpening:**

- As Laplacian is a derivative operator, its use highlights intensity discontinuities in an image and deemphasizes regions with slowly varying intensity levels.

- This will produce images that have grayish edge lines and other discontinuities.

- If the definition used has a negative center coefficient, then we *subtract, rather than add, the Laplacian image to obtain a* sharpened result.

- Thus, the basic way in which we use the Laplacian for image sharpening is

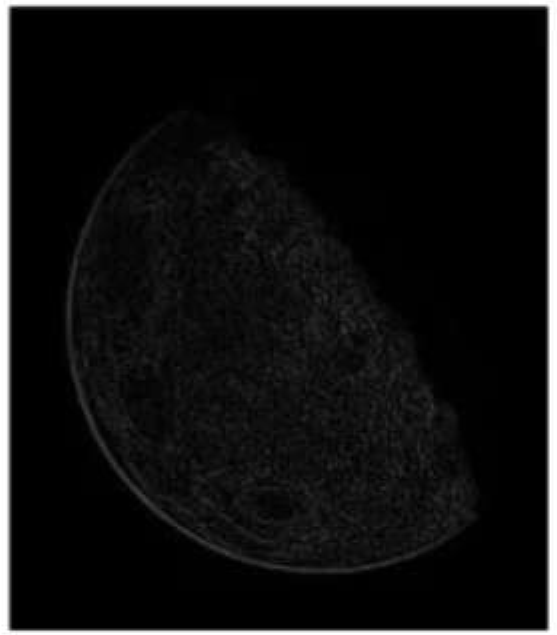
$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

The constant is c = -1, if the Laplacian filters are mask 1 or 2 used, and c = 1 if either of the other two filters is used.

Sharpening Filters

- **Second Derivative for Image Sharpening:**

Blurred image of the moon



Laplacian without scaling

Laplacian with scaling

Image sharpened using the mask 1

Result of using the mask 2

0	1	0
1	−4	1
0	1	0

Mask 1

1	1	1
1	−8	1
1	1	1

Mask 2

Sharpening Filters

- **Unsharp Masking and Highboost Filtering:**

- Printing and publishing sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image.

- This process, called *unsharp masking,*

- Consists of the following steps:

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the *mask.)*
3. Add the mask to the original.

Sharpening Filters

- **Unsharp Masking and Highboost Filtering:**

- Letting $\bar{f}(x, y)$ denote the blurred image, unsharp masking is expressed in equation form as follows. First we obtain the mask:

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

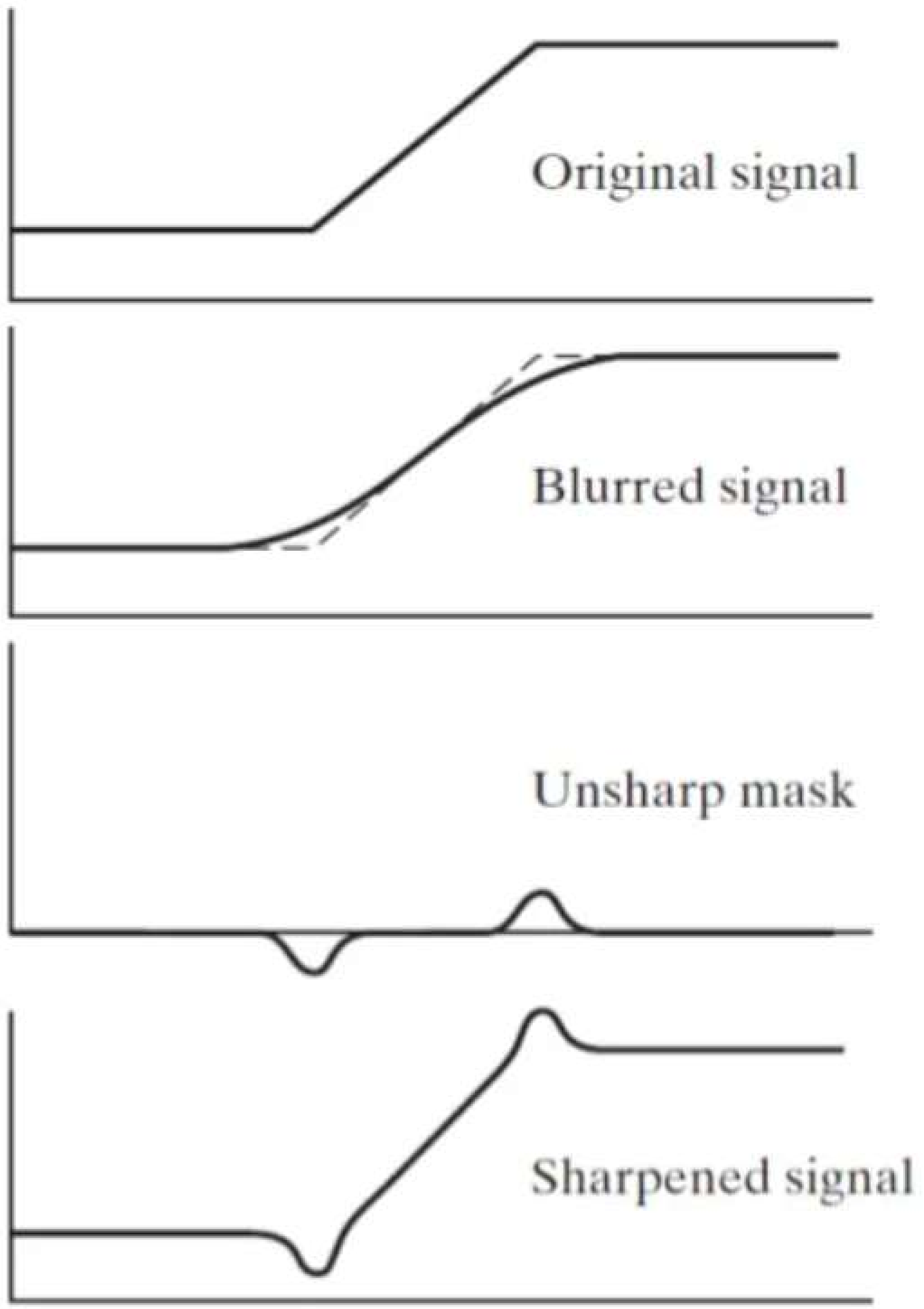
Then we add a weighted portion of the mask back to the original image

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$

where we included a weight, k (k ≥ 0), for generality. When k = 1, we have unsharp masking, as defined above. When k ≥ 1, the process is referred to as *highboost filtering. Choosing* k < 1,*de-emphasizes the contribution of the unsharp* mask.

Sharpening Filters

- **Unsharp Masking and Highboost Filtering:**



Example

Sharpening Filters

- **First-Order Derivatives (Nonlinear) Image Sharpening:**

- First derivatives in image processing are implemented using the magnitude of the gradient. For a function $f(x, y)$, the gradient of *f at coordinates* (x, y) *is defined* as the two-dimensional column *vector*

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

This vector has the important geometrical property that it points in the direction of the greatest rate of change of *f at location (x, y).*

Sharpening Filters

- **First-Order Derivatives (Nonlinear) Image Sharpening:**

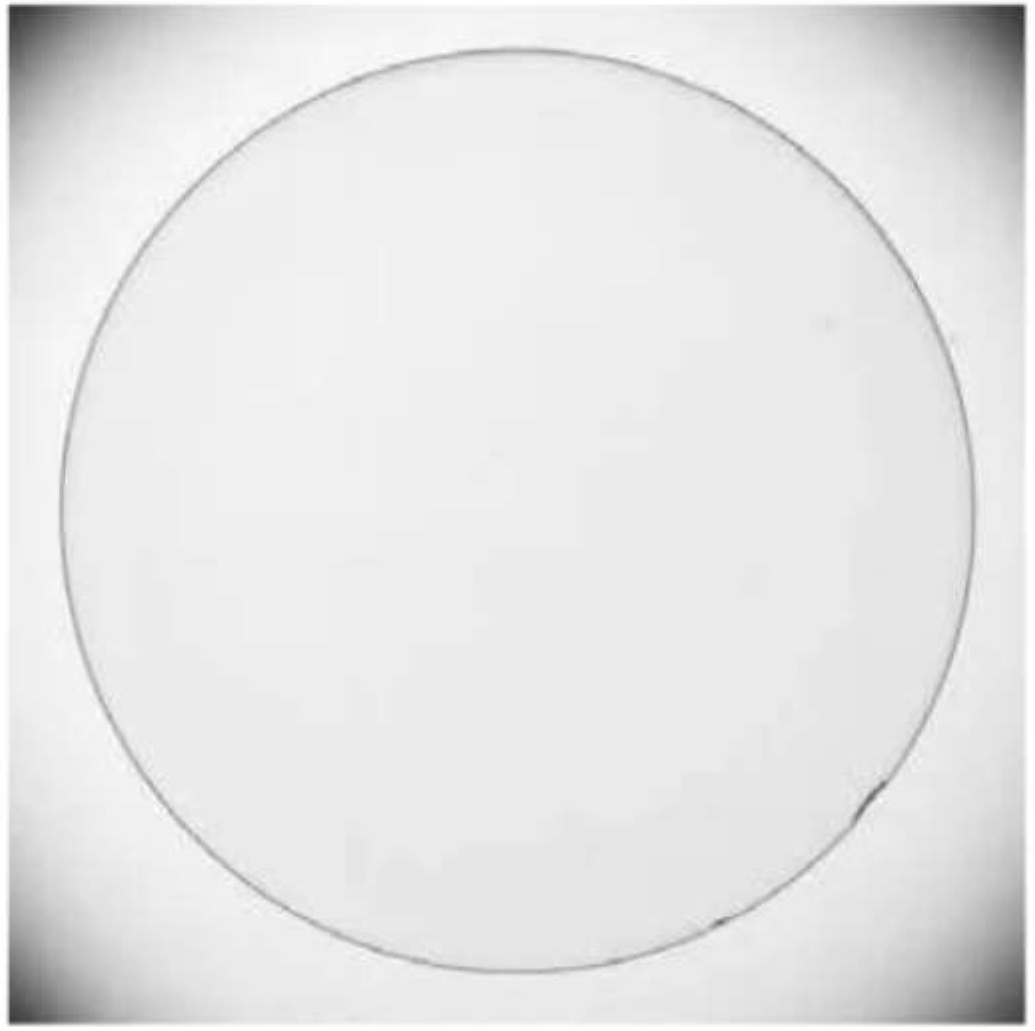
- The *magnitude (length) of vector denoted as M(x, y), where*

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

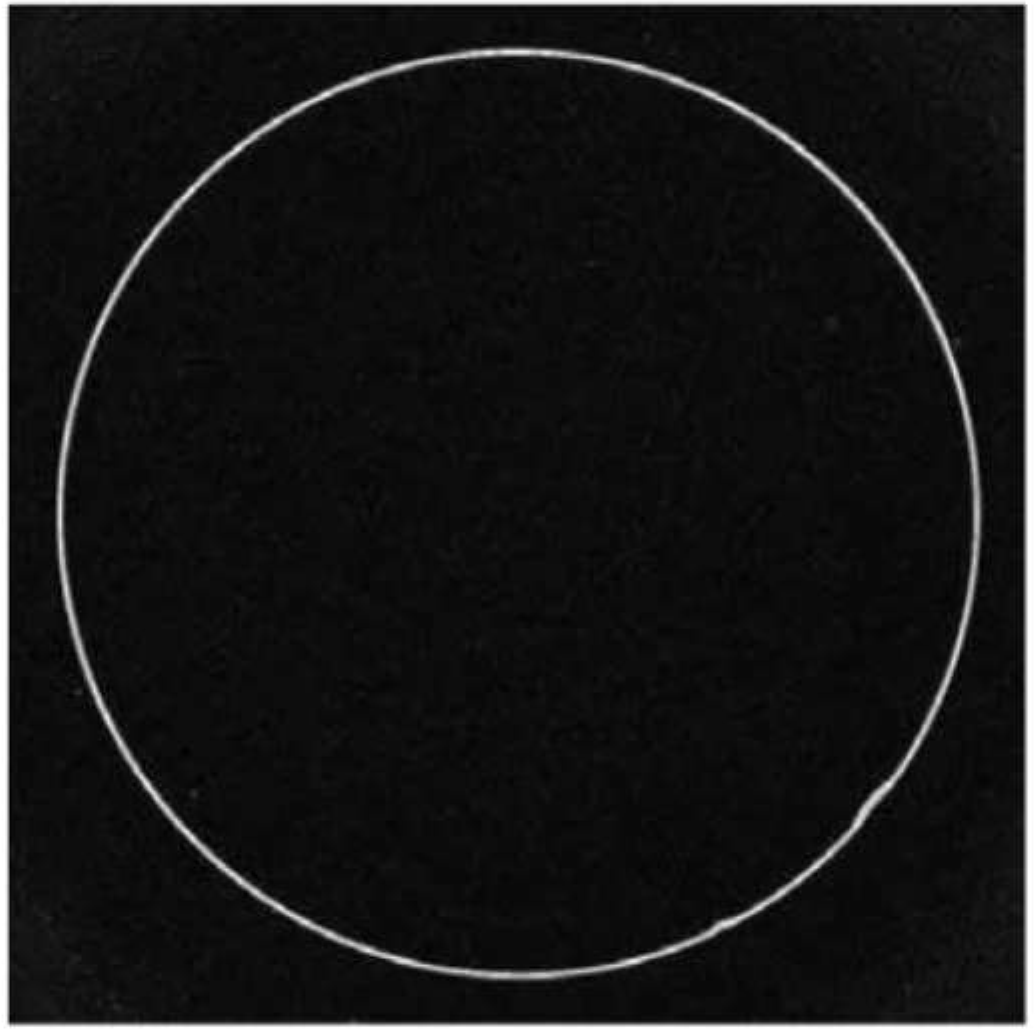
is the *value at (x, y) of the rate of change in the direction of the gradient vector.* Note that *M(x, y) is an image of the same size as the original, created when x and y are allowed to vary over all pixel locations in f. It is common practice* to refer to this image as the *gradient image (or simply as the gradient when the* meaning is clear).

Sharpening Filters

- **First-Order Derivatives (Nonlinear) Image Sharpening:**



Optical image of contact lens
(with defects)



Sobel gradient.