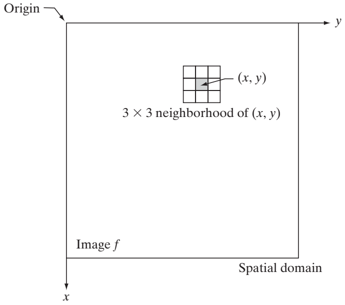
Introduction to Digital Image Processing

Dr. Kundan Kumar
Ph.D. (IIT Kharagpur)
Assistant Professor
ECE Department (Cabin - E139)

Institute of Technical Education & Research (ITER)
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India-751030

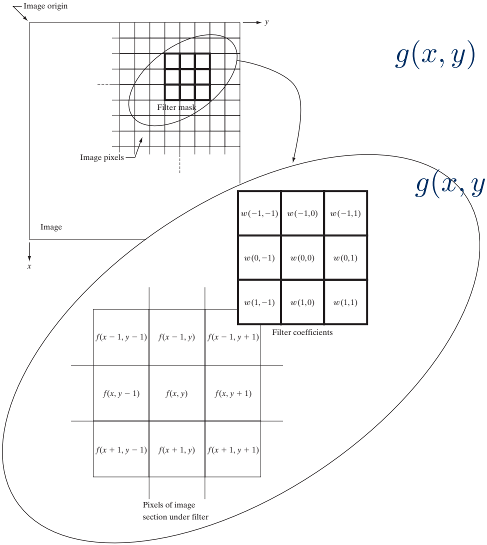
Spatial Filtering

- Define a center point (x, y)
- Perform an operation that involves only the pixels in a predefined neighborhood
- Result of the operation response of the process at that point
- Repeat the process for every pixel in the image



- Output is a function of a pixel value and its neighbors.
- Possible operation are: sum, weighted sum, average, weighted average, min, max, median, etc.

Spatial Filtering

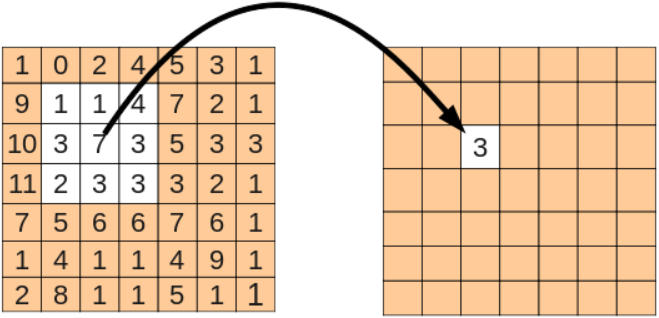


$$g(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) \cdot f(x+i, y+j)$$

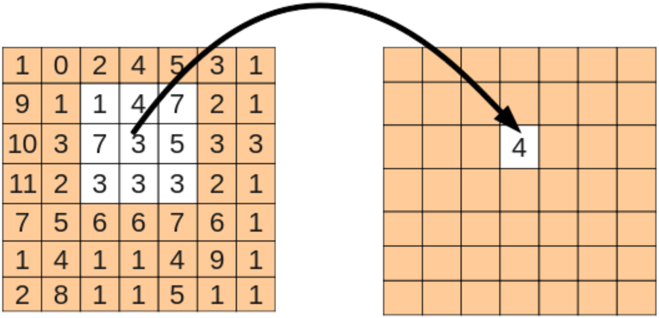
$$\begin{aligned} g(x,y) &= w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} \\ &= \sum_{k=1}^{mn} w_k z_k \\ &= \mathrm{w}^T \mathrm{z} \end{aligned}$$

(called inner product)

Spatial Filtering - Example

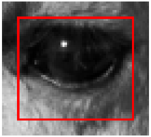


Spatial Filtering - Example



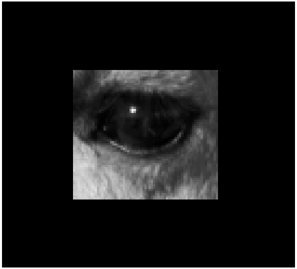
Handling boundary pixels

- Crop



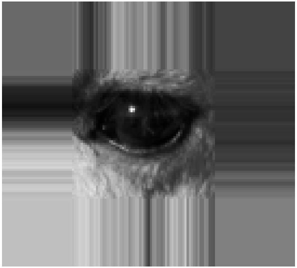
Handling boundary pixels

- Crop
- Padding



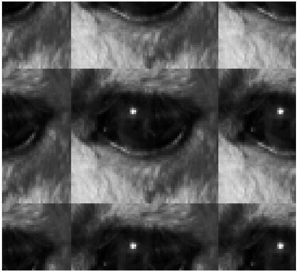
Handling boundary pixels

- Crop
- Padding
- Extend



Handling boundary pixels

- Crop
- Padding
- Extend
- Wrap



Correlation and Convolution

Origin $f(x, y)$

$$\begin{matrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{matrix} \qquad w(x,y)\;\begin{matrix} 1&2&3\\ 4&5&6\\ 7&8&9 \end{matrix}$$

(a)

Padded f

$$\begin{matrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(b)

Initial position for w

$$\begin{matrix} 1&2&3&0&0&0&0&0&0\\ 4&5&6&0&0&0&0&0&0\\ 7&8&9&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(c)

Full correlation result

$$\begin{matrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&9&8&7&0&0&0\\ 0&0&0&6&5&4&0&0&0\\ 0&0&0&3&2&1&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(d)

Cropped correlation result

$$\begin{matrix} 0&0&0&0&0\\ 0&9&8&7&0\\ 0&6&5&4&0\\ 0&3&2&1&0\\ 0&0&0&0&0 \end{matrix}$$

(e)

Rotated w

$$\begin{matrix} 9&8&7&0&0&0&0&0&0\\ 6&5&4&0&0&0&0&0&0\\ 3&2&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(f)

Full convolution result

$$\begin{matrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&1&2&3&0&0&0\\ 0&0&0&4&5&6&0&0&0\\ 0&0&0&7&8&9&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

(g)

Cropped convolution result

$$\begin{matrix} 0&0&0&0&0\\ 0&1&2&3&0\\ 0&4&5&6&0\\ 0&7&8&9&0\\ 0&0&0&0&0 \end{matrix}$$

(h)

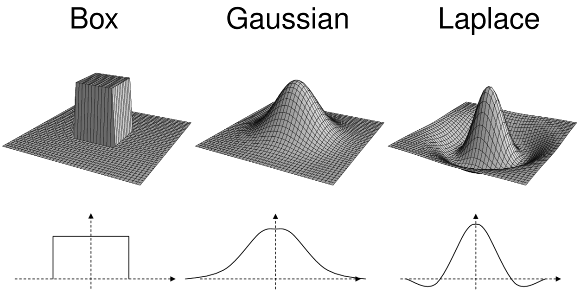
correlation

$$g(x, y) = \sum_{j=-k}^{k} \sum_{j=-k}^{k} w(i, j) \cdot f(x + i, y + j)$$

convolution

$$g(x, y) = \sum_{j=-k}^{k} \sum_{j=-k}^{k} w(i, j) \cdot f(x - i, y - j)$$

Filter masks



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

Linearly Separable filter

Definition

A kernel H is called separable if it can be broken down into the convolution of two kernels:

$$H = H_1 * H_2 \tag{1}$$

More generally, we might have:

$$H = H_1 * H_2 * \cdots * H_n \tag{2}$$

Example: The "shift by ten" kernel is 10 copies of the "shift by one" kernel convolved together.

Linearly Separable filter

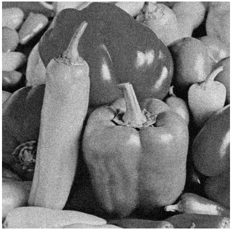
- Prewitt and Sobel filters are examples of linearly separable filters
- The filter kernel matrix can be separable as the product of a column vector with a row vector.

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

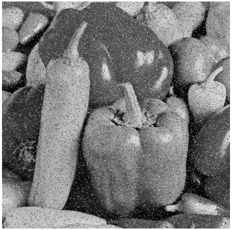
- 2D filtering process can actually be carried out by two sequential 1-D filtering operations.
- Thus, the rows of the image are first filtered with the 1-D row filter and the resulting filtered image is then filtered column-wise by the 1-D column filter.

Types of noise

- What kind of noise?
  - Additive
  - Multiplicative
  - Impulsive
- Image smoothing suppress the noise by using redundancy in the image data.



(a)



(b)

Figure: (a) Random noise, (b) Impulse noise

Spatial Filtering

- Smoothing
- Sharpening

A B C

Figure: A: Input image; B: Smoothing filter output; C: Sharpening filter output

Spatial filtering

Linear Filters:

- Smoothing filters
  - Mean filter
  - Gaussian filter
- Edge enhancing filters (Sharpening)
  - Sobel filter
  - Prewitt filter
  - Laplace filter

Non-linear filters:

- Min, Max
- Median, Percentile filters.

Smoothing filter

- Referred to low-pass filter
- Reduce sharp transition in intensities
- Reduce noise
- Reduce “irrelevant” details in the image
- Blur edges

Mean and Weighted Averaging Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

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

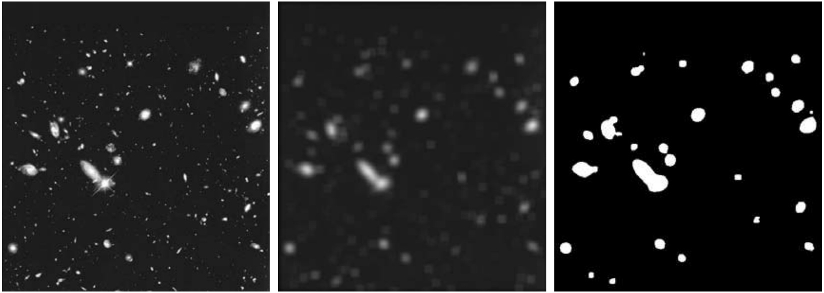
Mean Filter

Figure: Input image; Mean image with 3×3 kernel and Mean filter with 5×5 kernel.

Mean Filter

Figure: Input image image; Mean image with 3×3 kernel and Mean filter with 11×11 kernel.

Application



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Gaussian Filtering

- The mean (box) filter gives "blocky" effect in blurring
- Radially symmetric filter is more preferable
- Blurring looks better if the weighting dies off gradually, rather than all of a sudden
- The Gaussian is radially symmetric and dies-off gradually.

In 1D:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

In 2D:

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

Gaussian filter

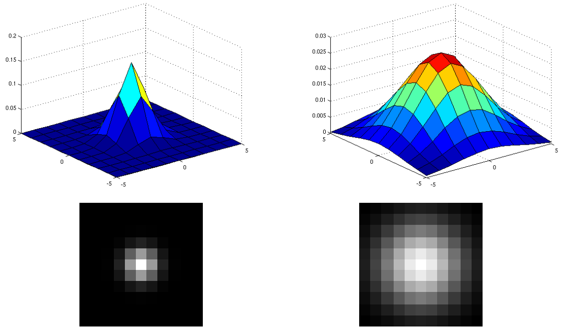


Figure: Gaussian distribution for $\sigma = 1$ and $\sigma = 2.5$ and corresponded Gaussian kernels of the size 11×11.

Gaussian filter

A B C

Figure: A: Input image; B: Gaussian filter size 11×11 with $\sigma = 1$; C: Gaussian filter size 11×11 with $\sigma = 2.5$.

Separability of 2D Gaussian

- A 2D Gaussian is just the product of two 1D Gaussians:

$$\begin{aligned} G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right) \\ &= g_\sigma(x)\cdot g_\sigma(y) \end{aligned} \tag{3}$$

- As a result, convolution with a Gaussian is separable

$$f * G = f * G_x * G_y \tag{4}$$

where G_x is the “horizontal” and G_y the “Vertical” 1D discrete Gaussian kernels.

Gaussian Filter Mask with $\sigma = 1$

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Order-Statistic (Nonlinear) Filters

- Takes the median of the values in some neighbourhood
  - Better than mean (linear filters) for some noise models
  - Preserves edges slightly better than linear filters
- Generalizes to percentile filter
  - Median is 50%
  - 0% is Min filter (= erosion, morphological operation)
  - 100% is Max filter (= dilation, morphological operation)
  - Other useful filters: 5%, 95% (like min and max, but less sensitive to noise)

Max and Min filter

- Max filter - The maximum value replaces the current pixel $\Rightarrow$ brighter image
- Min filter - The minimum value replaces the current pixel $\Rightarrow$ darker image

Figure: Original image; Max filter with 3×3 kernel and Min filter with 3×3 kernel.

Median filter

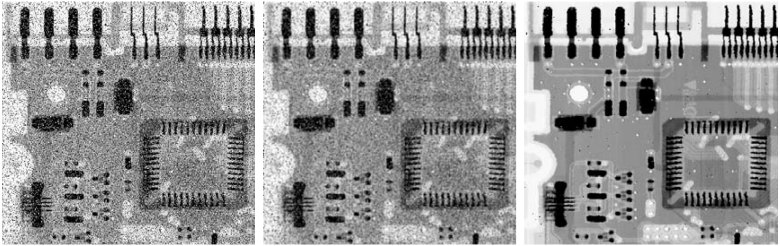
- Good for impulsive noise (salt and pepper)
- Sharp edges are kept as no new values are created
- Dominating areas become more dominating; fine details are erased as very thin lines and sharp corners are damaged.
- Time-consuming to sort values compared to mean values
- Applying median filtering is an efficient smoothing.

Median filter

A B C

Figure: A: Input image; B: Median filter size 3×3; C: Median filter size 5×5.

Comparison between Mean and Median filter



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening spatial Filtering

- Highlight transitions in intensities
- Edges are mainly used in image analysis for finding region boundaries
- Edges are pixels where values change abruptly
- Sharpening - Image details can be enhanced by adding the edges to the input image
- For image functions $f(x, y)$ partial derivatives may be approximated by differences
- A change can be described by a gradient that pointing in the direction of the largest growth of the image function

Partial derivatives

- Continuous domain

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}$$

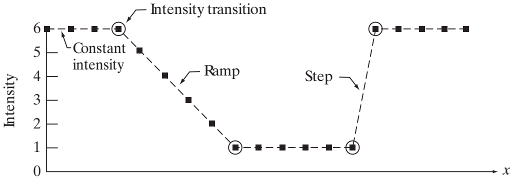
- In Discrete domain

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

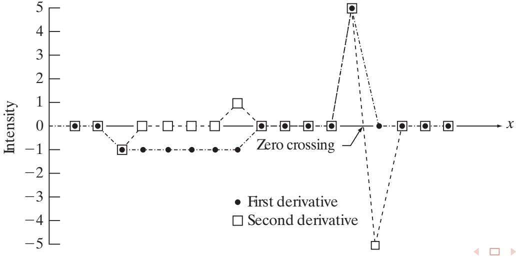
- In the discrete case, we approximate the first-order derivative by finite differences
- Second-order derivative of $f(x,y)$

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

Discrete Derivatives - Example



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6 → x
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Gradient Operator

- The gradient of an image $f(x,y)$ is the vector $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}) = (g_x, g_y)$
- The magnitude of ∇f is the important quantity in edge detection
- Often approximated with

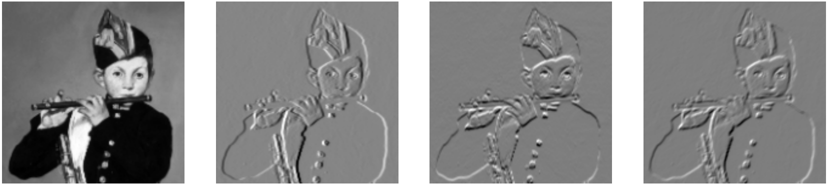
$$\nabla f \approx |g_x| + |g_y|$$

or

$$\nabla f = \max(|g_x|, |g_y|)$$

- Direction of ∇f is $\arctan \frac{g_x}{g_y}$

Gradient Operator



A B C D

Figure: Input image; g_x, g_y, $|g_x| + |g_y|$.

Gradient operators detect a direction of the edge

- Approximations of first derivatives use a collection of masks, each corresponding to a certain direction
- 8 directions for a 3×3 neighbourhood
- The response of each filter represents the magnitude of the edge in that direction
- The mask resulting in the greatest magnitude indicates the edge direction at a certain pixel
- Well known operators:
  - Roberts
  - Prewitt
  - Sobel

Roberts operator

-1	0
0	1

0	-1
1	0

- Due to its small neighbourhood:
  + Easy to compute.
  - High sensitivity to noise.

Figure: Roberts operator.

Prewitt operator

-

1	0	-1
1	0	-1
1	0	-1

1	1	1
0	0	0
-1	-1	-1

Figure: Prewitt operator for vertical and horizontal direction; Gradient magnitude.

Sobel operators

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

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

Sobel operator for horizontal edge

Sobel operators

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

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

Sobel operator for horizontal edge

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

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

Sobel operator for vertical edge

Sobel operators

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

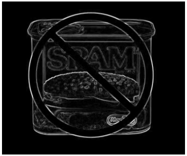
-1	-2	-1
0	0	0
1	2	1

Sobel operator for horizontal edge

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

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

Sobel operator for vertical edge



Roberts, Prewitt and Sobel operator

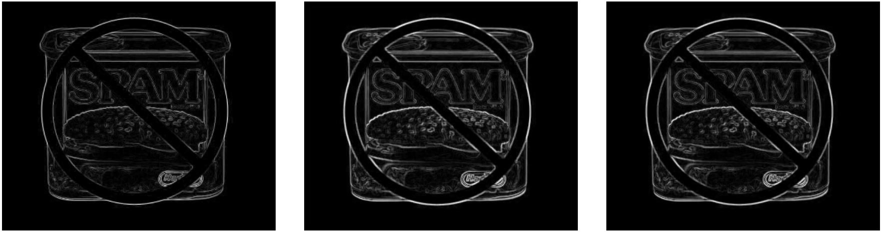


Figure: Comparison between Roberts, Prewitt and Sobel operator.

Image sharpening using second derivative: Laplacian Operator

- Simplest second derivative operator is the Laplacian

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

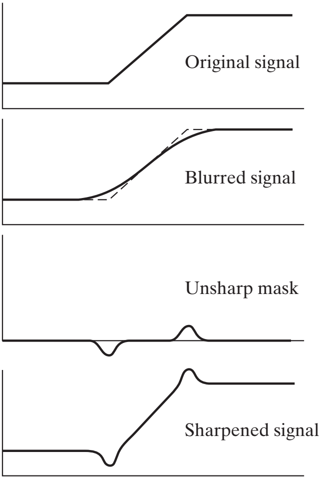
- To express the expression in discrete form

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

- Therefore, the discrete Laplacian of two variables is

$$\begin{aligned}\nabla^2 f(x,y) =& f(x+1,y) + f(x-1,y) + f(x,y+1) \\ &+ f(x,y-1) - 4f(x,y)\end{aligned}$$

Laplacian operator



- The Laplacian is a well-known linear differential operator approximating the second derivative.
- Rotation invariant; need only one convolution mask
- Response doubly to some edges
- Noise sensitive

Laplacian Mask

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Laplacian Mask Operation

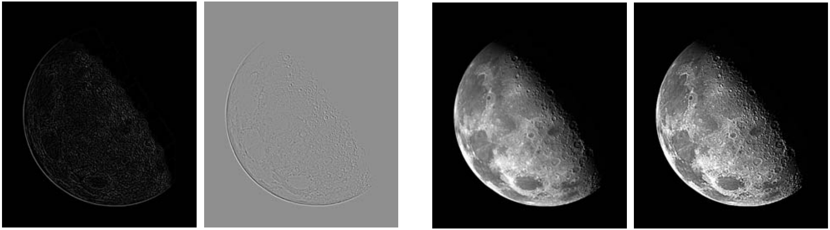
- Because the Laplacian is a derivative operator, its use highlights intensity discontinuities in an image and de-emphasizes regions with slowly varying intensity levels.
- This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background.
- Background features can be recovered while still preserving the sharpening effect of the Laplacian simply by adding the Laplacian image to the original.

$$g(x, y) = f(x, y) + k(\nabla^2 f(x, y))$$

Laplacian Mask Operation

a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)



Unsharp Masking and Highboost Filtering

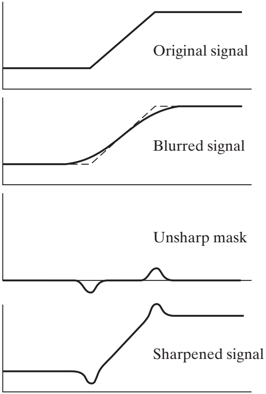
- A process of substracting an unsharp (smoothed) version of an image from the original image is called unsharp masking.
- Consist of the following step:
  1. Blur the original image
  2. Subtract the blurred image from the original (resulting difference is called the mask)
  3. Add the mask to the original.
- Letting $\bar{f}(x, y)$ denote the blurred image, unsharp masking is expressed in equation form as follows.
  - First we obtain the mask:

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

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

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

Unsharp Masking and Highboost Filtering



- Where a weight, k $(k \geq 0)$, for generality.
- When $k = 1$, we have *unsharp masking*.
- When $k > 1$, the process is referred to as *highboost filtering*.
- Choosing $k < 1$ de-emphasizes the contribution of the unsharp mask.

Thank You

Queries?