Digital Image Processing

Image Enhancement - Filtering

Derivative

- Derivative is defined as a rate of change.

Discrete Derivative Finite Distance

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

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

Forward difference

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

Central difference

Example

$$
\begin{array}{lllllllll}
f(x) = & 10 & 15 & 10 & 10 & 25 & 20 & 20 & 20 \\
f'(x) = & 0 & 5 & -5 & 0 & 15 & -5 & 0 & 0 \\
f''(x) = & 0 & 5 & -10 & 5 & 15 & 20 & 5 & 0
\end{array}
$$

Derivative Masks

Backward difference	[-1 1]
Forward difference	[1 -1]
Central difference	[-1 0 1]

Derivatives in 2-dimension

Given function $f(x,y)$

Gradient vector $\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

Gradient magnitude $\left|\nabla f(x,y)\right| = \sqrt{f_x^2 + f_y^2}$

Gradient direction $\theta = \tan^{-1} \dfrac{f_x}{f_y}$

Derivatives of Images

Derivative masks $f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $\quad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of Images

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Correlation

$$f \otimes h = \sum_{k} \sum_{l} f(k,l)\,h(i+k, j+l)$$

$$f = \text{Image}$$

$$f = \text{Kernel}$$

$$\begin{array}{c} f \\ \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \end{array} \otimes \begin{array}{c} h \\ \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \end{array} \rightarrow$$

$$\begin{aligned} f * h = {} & f_1 h_1 + f_2 h_2 + f_3 h_3 \\ & + f_4 h_4 + f_5 h_5 + f_6 h_6 \\ & + f_7 h_7 + f_8 h_8 + f_9 h_9 \end{aligned}$$

Convolution

$$f * h = \sum_k \sum_l f(k,l)\,h(i-k,\,j-l)$$

$f = \text{Image}$

$h = \text{Kernel}$

$$\begin{bmatrix} h_7 & h_8 & h_9 \\ h_4 & h_5 & h_6 \\ h_1 & h_2 & h_3 \end{bmatrix} \xleftarrow{X-flip} \overset{h}{\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}}$$

$$\Big\downarrow\, Y-flip$$

$$\overset{f}{\begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}} \otimes \begin{bmatrix} h_9 & h_8 & h_7 \\ h_6 & h_5 & h_4 \\ h_3 & h_2 & h_1 \end{bmatrix} \longrightarrow$$

$$\begin{aligned} f * h = {} & f_1 h_9 + f_2 h_8 + f_3 h_7 \\ & + f_4 h_6 + f_5 h_5 + f_6 h_4 \\ & + f_7 h_3 + f_8 h_2 + f_9 h_1 \end{aligned}$$

Averages

- Mean

$$I = \frac{I_1 + I_2 + \ldots I_n}{n} = \frac{\sum_{i=1}^{n} I_i}{n}$$

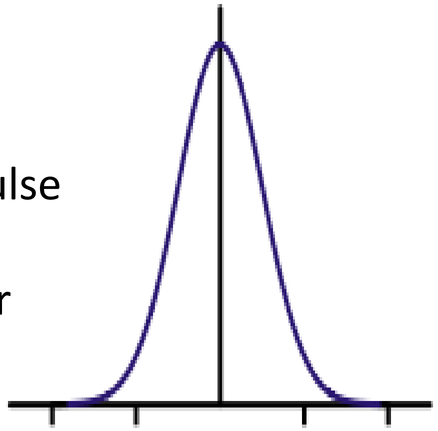
- Weighted mean

$$I = \frac{w_1 I_1 + w_2 I_2 + \ldots + w_n I_n}{n} = \frac{\sum_{i=1}^{n} w_i I_i}{n}$$

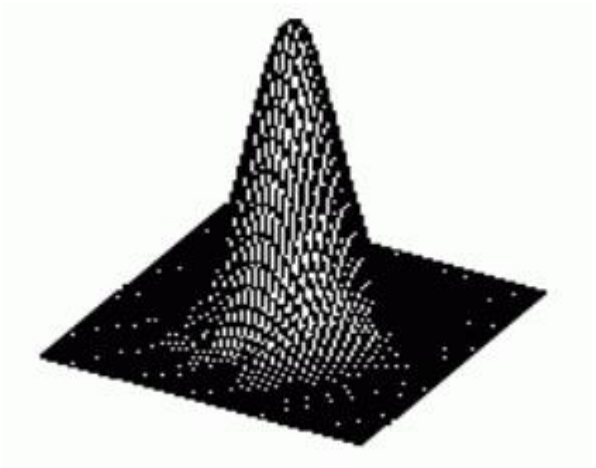
Gaussian Filtering

- The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump.

Shape of Impulse Response for Gaussian Filter



2D Gaussian



Scale of Gaussian

- As sigma increases, more pixels are involved in averaging
- As sigma increase, image is more blurred
- As sigma increase, noise is more effectively suppressed

Gaussian Filter

- Gaussian smoothing

$$\overbrace{S}^{\text{smoothed image}} = \overbrace{g}^{\text{Gaussian filter}} * \overbrace{I}^{\text{image}} \qquad g = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{1}{2\pi\sigma^2} . e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Consider sigma = 0.6 and kernel size = 3x3

$$\frac{1}{2\pi\sigma^2} = \frac{1}{2 \times 3.14 \times 0.6 \times 0.6} = \frac{1}{2.2619}$$

Gaussian Smoothing

Kernel width; X= 3, Kernel Height; Y = 3

$$X = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } Y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$e^{\frac{-(x^2+y^2)}{2\sigma^2}} = \begin{bmatrix} -2.7778 & -1.3889 & -2.7778 \\ -1.3889 & 0 & -1.3889 \\ -2.7778 & -1.3889 & -2.7778 \end{bmatrix}$$

The Gaussian kernel's center part (Here 0.4421) has the highest value and intensity of other pixels decrease as the distance from the center part increases.

Gaussian Smoothing

The Gaussian kernel takes the form as:

$$\begin{bmatrix} 0.0275 & 0.1102 & 0.0275 \\ 0.1102 & 0.4421 & 0.1102 \\ 0.0275 & 0.1102 & 0.0275 \end{bmatrix}$$

Convolve the kernel with the region in the given image

$$\begin{bmatrix} 72 & 68 & 88 & 159 \\ 69 & 66 & 87 & 162 \\ 70 & 66 & 83 & 161 \\ 70 & 66 & 78 & 154 \end{bmatrix}$$

Gaussian Smoothing

Performing Convolution:

$$\begin{bmatrix} 68 & 88 & 159 \\ 66 & 87 & 162 \\ 66 & 83 & 161 \end{bmatrix} .* \begin{bmatrix} 0.0275 & 0.1102 & 0.0275 \\ 0.1102 & 0.4421 & 0.1102 \\ 0.0275 & 0.1102 & 0.0275 \end{bmatrix} = \begin{bmatrix} 1.8692 & 9.7009 & 4.3706 \\ 7.2757 & 38.4624 & 17.8585 \\ 1.8142 & 9.1497 & 4.4256 \end{bmatrix}$$

On convolution of the local region and the Gaussian kernel gives the highest intensity value to the center part of the local region **(38.4624)** and the remaining pixels have less intensity as the distance from the center increases.
Sum up the result and store it in the current pixel location **(Intensity = 94.9269)** of the image.

Gaussian Smoothing

$$\begin{bmatrix} [] & [] & [] & [] \\ [] & [] & 94.9269 & [] \\ [] & [] & [] & [] \\ [] & [] & [] & [] \end{bmatrix}$$

Performing calculations for each pixel, the resultant image is:

$$\begin{bmatrix} 48.7478 & 59.2645 & 79.7865 & 100.2444 \\ 57.1176 & 69.7512 & 94.9296 & 121.1870 \\ 57.1740 & 68.9526 & 92.2220 & 119.6981 \\ 47.7534 & 59.9750 & 74.1254 & 96.7113 \end{bmatrix}$$

The end