Assignment - 2

* Name = Parthkumar Patel
* Student ID = A20416508
* Course number = Fall 2018 CS - 512
* Semester = Fall 2018.

(1)

(a)

→ SNR is calculated by, using below formula,

$$SNR = \frac{E_S}{E_N} = \frac{\sigma_s^2}{\sigma_n^2} = \frac{\frac{1}{n}\sum_{ij}\left(I(i,j)-\bar{I}\right)^2}{\sigma_n^2}$$

where SNR is the signal to noise ratio

→ To calculate variance in image first we calculate the difference between intensity at all pixels minus the average intensity and then square it.

→ Variance in the image is calculated using two ways.

(1) Calculating variance for multiple frames of a static view.

(2) Calculating variance in uniform region.

श्री रामकृष्ण ...
Dt: / / | Pg no:

(b)

→ # Impulsive noise -

→ They have two types : (1) positive impulsive
(2) Negative impulsive

→ They take specific pixels ~~of~~ and add very different values instead off adding values at every pixels.

→ It includes instantaneous and not needed sharp sounds.

Gaussian Noise -

→ It ~~has~~ is a ~~p~~ statistical noise having a PDF equals to that of the normal distribution which is known as a gaussian distribution.

→ If the image has a gaussian noise then the profile cross section has peeks.

→ Median Filter handles better impulsive noise.

(c)

→ Image =

2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2

→ Filter =

1	1	1
1	1	1
1	1	1

→ After applying 3×3 convolution filter

0	0	0	0	0	0	- - - .
0	18	18	18	18	18	- —
0	18	18	18	18	18	- - - -
0	18	18	18	18	18	- - -
0	18	18	18	18	18	- - . .

(d)

→ We need the derivative of an image convolved with a filter. To make it more efficient first do the derivative of filter then convolve with an image. It is more efficient compersed with the derivative of an image convolved with a filter.

(e)

→ Three different ways to handle boundries during convolution is =

(1) zero ~~perdeting~~. padding.
(2) Ignore the boundries
(3) Mirror/ Replicate.

(f)

→ smoothing filter = $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

→ In this filters, from left to right sum of filters are 9.

→ Intensity becomes 9 times higher, so to normalize we have to multiply by 1/9.

(g)

→ separable implementation =

$$I_G = I * G = \sum_i \sum_j I(i,j)\, e^{-\frac{i^2+j^2}{2\sigma^2}}$$

$$= \sum_i e^{-i^2/2\sigma^2} \sum_j I(i,j)\, e^{-j^2/2\sigma^2}$$

$$= (I * G_y) * G_x$$

$$\cancel{= I * G_x * G_y}$$

$$= I * G_x * G_y$$

2 1D Gaussian.

→ For, $M * N$ image, one 2D convolution requires $M * N * m^2$ operations.

→ Two 1D convolution requires $2 * M * N * m$ operations.

→ $2 * M * N * m$ is less then $M * N * m^2$ so, two 1D convolution is more efficient.

(h)

→ $\sigma \le m/5$

$\therefore m \geq \sigma \cdot 5$

here, $\sigma = 2$ is given,

so, $m \geq 2 \cdot 5$

$\therefore m \geq 10.$

→ Size of the filter should be 10.

(i)

→ Gaussian image pyramid is produced by shrinking the image.

→ Instead off changing the window size, we change the size of image and for that we produced gaussian image pyramid.

→ Total amount of additional processing done in a pyramid is,

$$m^2 + \frac{1}{4}m^2 + \frac{1}{16}m^2 + \cdots < \frac{4}{3}m^2$$

$$\therefore \frac{4}{3}m^2 - m^2 = \frac{1}{3}m^2$$

→ So, we require 30% extra additional image processing compared to a single image.

(j)

→ Laplacian pyramid is produced by convolving with the current image., then sample it and after that unsample (deconvolute) it and then again convolve with gaussian and then consider difference between original image and later image which leads to residuals. It is formed by residuals.

→ It is useful for compression. (small residuals).

(2)
(a)
→ Edge detection consendrate on the important part of the image and it allows to understand the image by looking the edges.

→ Desirable properties of edge detection.
(1) scene elements.
(2) invariant (pose, illumination, scale, viewpoint).
(3) reliable detection.

(b)
→ Edge detection steps are =

(1) Smooth to reduce noise (without affecting edges).
(2) Enhance edges (not the blur image)
(3) detect edges.
(4) localize edges (precise location of edge).

→ When we are performing smoothing, if we take derivative of image, it is sensitive to noise. so starts with convolution to remove noise.

(c)

→ Two filters used for image gradient is
(1) Sobel filter
(2) Gaussian filter

→ Image gradient means computing image derivatives.

→ Image gradient is a directional change in the intensity or color in an image

→ It is used to detect edges.

(d)

→ First smooth and then take derivative

$$\Delta x = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{smooth}$$

smoothing derivative

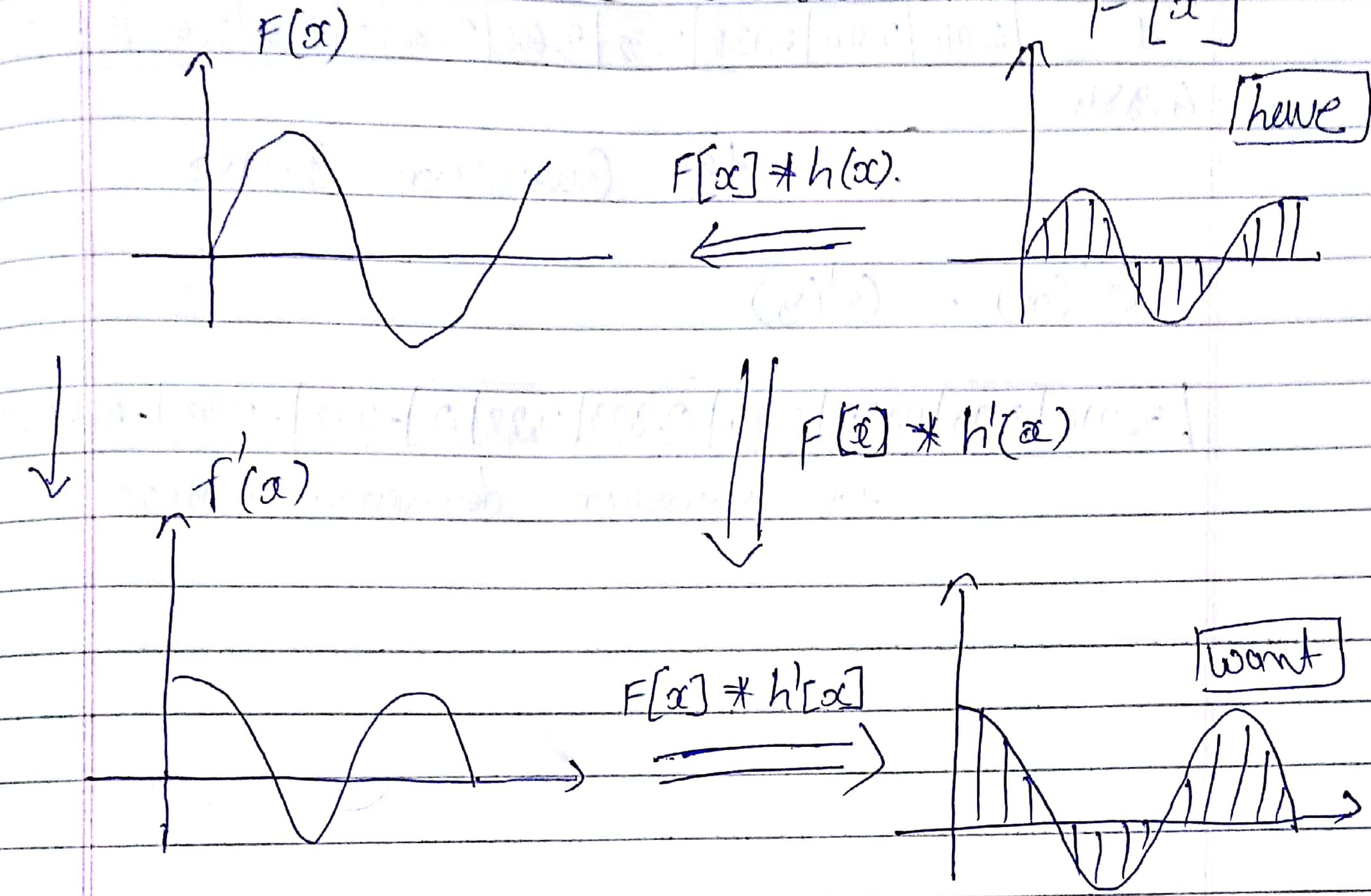
derivative

$$\Delta Y = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

smoothing derivative.

(e)

* More accurate derivative.



→ we have discrete sample of image. So, first have to make it continious by convolving using filter.

→ Then find the derivative of filter & convolve it with discrete function.

→ Then take sample of the derivative & convolve with the function to get the accurate derivative

→ Elements of a fiter for more accurate derivative computation with $\sigma = 2$.

then $M = 11$ $\left[\begin{array}{l} m \geq 5\sigma \\ m \geq 10 \end{array}\right.$

\# $G(x) = G(y)$

$\frac{1}{4.984}$

0.011	0.04	0.135	0.32	0.606	0.88	1	0.88	0.600	0.32	0.135

1D Gaussian Filter

$G'(x) = G'(y)$

0.016	0.05	0.135	0.24	0.303	0.22	0	-0.22	-0.303	-0.24	-0.135

1D Gaussian derivative filter.

श्री स्वामिनारायण विज्ञान विहार
Dt: / / Pg.no.:

(f)

→ It is difficult to localize an edge using first order derivative.

→ In the second order derivative by zero crossing you can find localize edge.

(g)

→ smooth with a gaussian before aplying laplacian.

$$H \Rightarrow \nabla^2 (I * G) = \underbrace{\nabla^2 G}_{LoG} + I$$

where $G = e^{-r^2/2\sigma^2}$

Follow the river and will reach the sea

Here, $\sigma = 1$ is given

$$\nabla^2 G = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-r^2/2\sigma^2}$$

$$= \frac{r^2 - 2}{1} e^{-r^2/2}$$

$$= r^2 - 2e^{-r^2/2}$$

→ Edge detection using LOG :

(1) Compute LOG : $H = (\nabla^2 G) * I$

(2) Threshold : $E(i,j) = \begin{cases} 0 & \text{if } H(i,j) < 0 \\ 1 & \text{if } H(i,j) \geq 0 \end{cases}$

(3) Mark edges at transitions $0 \to 1$, $1 \to 0$
scan left to right and top to bottom.

(h)

→ * Standard edge detection :-

→ It uses 3x3 fiters, where each pixels get the number giving the greatest rate of change in light intensity in the direction where intensity is changing fastest by calculating gradient magnitude for each pixel.

* Canny edge detection :

→ It removes noise with a low pass filter first then applying a sobel filter and then doing non maximum suppression to pick out the best pixel for edges when there are multiple possibilities in a local neighbourhood.

* Conditions for detecting edge in canny =

(1) Attempt detection only if gradient magnitude is large enough.

(2) Smooth along the edges to preserve edges.

(3) Alternate to zero crossing of $\frac{d^2}{dG^2}(I * G)$ is maximum of $\frac{d}{dG}(I * G)$.

(I)

→ Non-maximum suppression =

→ Local maximum of gradient magnitude in direction of gradient.

$$\nabla(I * G) = (I_x, I_y)$$

$$\theta = \tan^{-1}\left(\frac{I_y}{I_x}\right).$$

$$\theta^* = \text{round}\left(\frac{\theta}{45}\right) * 45.$$ (directize angle by 45)

$$E(i,j) = \begin{cases} 1, & \text{if } \nabla(I * G) \text{ is a local maximum} \\ 0 & \text{otherwise} \end{cases}$$

* Hysteresis Thresholding =

→ ~~We~~ Use, τ_H to start tracking and τ_c to confine ($\tau_H > \tau_c$)

(1) intialize array of visited pixels $v(i,j) = 0$.

(2) scale image T–B, L–R.

if, $!v(i,j)$ && $|\nabla I| > \tau_H$ start tracking on edge.

↓

visited pixel is zero.

(3) search for additional neighbours in direction orthogonal to ∇I such that.

$$|\nabla I| > \tau_L$$

Meditation is the best mode of worship.