Name :- Yash Vijay Hoke
PRN :- 2018BTECS00101

DIP Assignment

Page No.
Date

Q.1 Explain canny edge detector in detail.

→ Canny's approach is based on 3 basic objectives:-

ⓐ Low error rate = All edges should be found and there should be no spurious responses.

ⓑ Edge points should be well localized = The edges located must be as close as possible to the true edges. That is, the distance between a point marked as an edge by the detector & center of the true edge should be as minimum as possible.

ⓒ Single edge point response = The detector should return only one point for each true edge point. That is, the number of local maxima around the true edge should be minimum. This means that the detector should not identify multiple edge pixels where only a single edge point exists.

There are 4 processes of canny edge detection algorithm.

i) Apply Gaussian filter to smooth the image in order to remove the noise. Since all edge detection results are easily affected by noise in image, it is essential to filter out noise to prevent false detection. To smooth the image a Gaussian fiter kernel is convolved with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector.

Page No.
Date

let $f(x,y)$ denotes input image and $G(x,y)$ denotes the Gaussian function, then

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

we form a smoothed image $f_s(x,y)$ by convolving f and G

$$f_s(x,y) = G(x,y) * f(x,y)$$

ii) Finding the intensity gradient of image. An edge in an image may point in a variety of directions, so the canny algorithm uses 4 filters to detect horizontal, verticle, diagonal edges in blurred image. The edge gradient & direction can be given by

$$M_s(x,y) = \|\nabla f_s(x,y)\| = \sqrt{g_x^2(x,y) + g_y^2(x,y)}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{g_y(x,y)}{g_x(x,y)}\right)$$

iii) Non-maximum suppression is an edge thinning technique. Non-maximum suppression is applied to find the locations with the sharpest change of intensity value. It contains following steps.

ⓐ Compare the edge strength of the current pixel with the edge strength of pixel in +ve & -ve gradient direction.

ⓑ If the edge strength of current pixel is greater compared to other pixels in the mask with same direction, the value will be preserved. Otherwise, the value will be supressed.

iv) Double threshold is the next step; after application of non-maximum suppression, remaining edge pixels provides a more accurate representation of real edges in an image. However, some remain that are caused by noise & color variation. In order to account for those spurious responses, it is essential to filter out edge pixels with a weak gradient value & preserve edge pixel with high gradient value. This is accomplished by selecting high & low threshold values. If edge pixel's gradient is higher than high threshold then it is marked as strong and if gradient is lower than low theshold then it is weak edge pixel.

Q.2] What is significance of using laplacian in LOG? Why not to use gradient?

→ Laplacian filters are derivative filters of used to find areas of rapid changes in images. Since derivative filters are very sensitive to noise, it is common to smooth the image before applying the laplacian like Gaussian filter. This two step process is called Laplacian of Gaussian.

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

There are different ways to find an approximate discrete convolution kernel that approximates

Page No.
Date

the effect of laplacian. To include a smoothing Gaussian filter, combine the Laplacian & Gaussian function to obtain single equation.

$$LoG(x,y) = \frac{-1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

LoG operator takes 2nd derivative of image. Where image is basically uniform, LoG will be zero. Wherever change occurs, LoG will give positive response to darker side and negative response to lighter side. At sharp edge between 2 regions, response will be

i) zero away from edge.
ii) positive just to one side.
iii) negative just to other side.
iv) zero at some point in between on edge itself.

Q.3 What are the filters used to implement laplacian?

→ The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The laplacian of an image highlights regions of rapid intensity change & is therefore used for edge detection.
The laplacian of image with pixel intensity $I(x,y)$ is given by

$$L(x,y) = \frac{\partial^2 I}{\partial^2 x} + \frac{\partial^2 I}{\partial^2 y}$$

This can be calculated using Convolution filter.

Q.4 Generate 2 × 2 haar matrix. With all steps.

→ The Haar transform is based on Haar function $h_u(x)$ defined over continuous, half open interval $x \in [0,1)$. Variable u is an integer that for $u>0$ can be decomposed as

$$u = 2^p + q$$

where p is largest power of 2 contained in u & q is the remainder - that is $q = 2^p - u$

$$h_u(x) \begin{cases} 1 & u=0 \text{ \& } 0 \le x < 1 \\ 2^{p/2} & u>0 \text{ \& } q/2^p \le x < (q+0.5)/2^p \\ -2^{p/2} & u>0 \text{ \& } (q+0.5)/2^p \le x < (q+1)/2^p \\ 0 & \text{otherwise} \end{cases}$$

The transformation matrix & basis image of discrete Haar transform can be obtained by substituting inverse transformation kernel

$$S(x,u) = \frac{1}{\sqrt{N}} h_u(x/N)$$

$$\therefore H_N = \begin{bmatrix} h_0(0/N) & h_0(0/N & \cdots & h_0(N-1/N) \\ h_1(0/N) & h_1(0/N) & \cdots & h_1(N-1/N) \\ \vdots & \vdots & & \vdots \\ h_{N-1}(0/N) & & & h_{N-1}(N-1/N) \end{bmatrix}$$

$$\therefore A_H = \frac{1}{\sqrt{N}} H_N$$

For $N=2$

$$A_H = \frac{1}{\sqrt{2}} \begin{bmatrix} h_0(0) & h_0(1/2) \\ h_1(0) & h_1(1/2) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Page No.
Date

Q.5 What is Short term Fourier transform? Write a short note.

→ The short time Fourier transform is a Fourier transform used to determine the sinusoidal frequency and phase content of local sections of signal as it changes overtime. The procedure of computing STFT is to divide longer time signal into short segments of the equal lengths and then compute the fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment. Then usually plotted the changing spectra as function of time, known as spectrogram or waterfall plot, such as commonly used in software Defined Radio based spectrum display. Full bandwidth displays covering the whole range of an SDR commonly use FFTs with 2^{24} points on desktop computers.