# COMP 478 DD

# Piyush Pokharkar

# 40120654

# Assignment 3

# Theoretical Questions

# 1)

Convert to polar coordinates:

# 2)

# 3)

# Programming Questions

# 1)

pkg load image;

I = imread('cameraman.tif');

subplot(1, 2, 1);

canny\_edge = edge(I, 'Canny', [0.1, 0.2], 2);

imshow(canny\_edge);

title('Canny Edge');

subplot(1, 2, 2);

LoG = edge(I, 'log', 40, 0.5);

imshow(LoG);

title('Laplacian of Gaussian');

## 1)

1. Smoothing the image to remove noise. Can be done using gaussian filter
2. Calculate the intensity gradient of the image to identify regions of rapid intensity change (edges). Can be done using sobel operators.
3. Calculate magnitude (strength of edge) and direction (edge’s orientation) of the gradient of each pixel.
4. Apply non-maximum suppression to thin the edges by keeping only the local maxima in the gradient direction. This removes the weaker edge detections.
5. Apply a double threshold, so pixels with magnitude T1 are an edge point and if pixel magnitude is between T1 and T2, and a connected pixel value is above T1, it is an edge point. Otherwise, non-edge.
6. Edge tracking by Hysteresis to connect weak edges to strong ones. Connects high-threshold pixels to form edges and includes some weak pixels if they are connected to strong edges.

## 2)

Edge linking is implemented to connect weak edged to strong edges. Identify weak pixels that are spatially connected to strong edges. Follow the weak edges using connectivity criterion and mark them as part of the edge. This is typically implemented using recursive algorithms (like depth-first search) or data structures like stack/queue to efficiently track and link edges.

No, Laplacian of Gaussian edge detector doesn’t need this step.

## 3)

LoG: threshold (numeric scalar), sigma.

edge(I, 'log', 40, 0.5);

A higher threshold minimized the

Canny: lower\_threshold, upper\_threshold, sigma

edge(I, 'Canny', [0.1, 0.2], 2);

## 4) A black and white image of a person and a camera Description automatically generated

After experimenting with different threshold and sigma values, canny edge has less noise and prominent connecting edges to form outlines. While Laplacian of Gaussian has more accurate, thinner edges which are closer to the shape of the original image. And it didn’t detect the background as much as the main object.

# 2)

A person with a camera on tripod

Description automatically generatedA black squares with white dots

Description automatically generated

u = 5;

v = 5;

T = 1;

a = 0.1;

b = 0.1;

H = zeros(u, v);

% Create a uxv filter

for i = 1:u

for j = 1:v

if(i==1 && j ==1)

H(i,j) = T;

else

H(i,j) = filter\_func(T,a,b,i-1,j-1);

endif

endfor

endfor

I = imread('cameraman.tif');

% Apply filter function using conv2

motion = conv2(I, real(H), 'same') + 1i \* conv2(I, imag(H), 'same');

% Plot

figure;

subplot(1, 2, 1), imshow(I), title('Original Image');

subplot(1, 2, 2), imshow(abs(motion), []), title('Motion Blur');

magnitudeImage = abs(H);

% Display the magnitude image

figure;

imshow(magnitudeImage, []);

title('Magnitude Image');

filter\_func.m

function y = filter\_func(T, a, b, u, v)

y = (T / pi\*(u\*a + v\*b)) \* sin(pi\*(u\*a + v\*b)) \* exp(-j\*pi\*(u\*a + v\*b));

endfunction