Assignment 4

COMP 478 Image Processing

Etienne Pham Do

40130483

Etienne Pham Do 40130483

COMP 478 Assignment 4

1.

Text, letter

Description automatically generated

2.

The Hough transform for lines cannot be carried out in the Cartesian system because a line that is expressed as b = -ax + y can have its slope a approach infinity when the value of b is constant (line is vertical), when expressing the line in the parameter space. Instead of representing the line as b = -ax + y, the representation is used, where is the perpendicular distance from the line to the origin and being the angle to which is perpendicular to the line. This translates to sinusoidal curves in the plane instead of relying on a straight line in the ab plane. The plane is then divided into multiple bins and for each time a sinusoid crosses a bin, a sum is calculated whose value is thresholded to detect lines.

3.

We use the Hough transform to detect and count all circle centers whose radius is equal to the big polka dot. In fact, instead of the line equation, we use the circle equation (x-a)­­­­2+(y-b)2 = r2, where (a,b) is the circle center and the r being the radius. The parameter space would consist of a,b and r. Knowing the value of the radius, we can trace a circle in the parameter space for each point in the original circle. The intersection of the traced circles would make up one circle center. When said intersection is found, a variable should keep count of the total number of circle centers whose radius is equal to r.

**Programming**

1.

a)

A picture containing ground

Description automatically generated

import cv2

import numpy as np

image = cv2.imread('tools\_noisy.png',0)

x, output\_image = cv2.threshold(image, 100, 255, cv2.THRESH\_OTSU)

cv2.imshow('Otsu', output\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

b)

A picture containing text

Description automatically generated

image = cv2.GaussianBlur(image, (5,5), 0)

x2, output\_image2 = cv2.threshold(image, 100, 255, cv2.THRESH\_OTSU)

cv2.imshow('Otsu', output\_image)

cv2.imshow('Otsu with denoise', output\_image2)

cv2.waitKey(0)

cv2.destroyAllWindows()

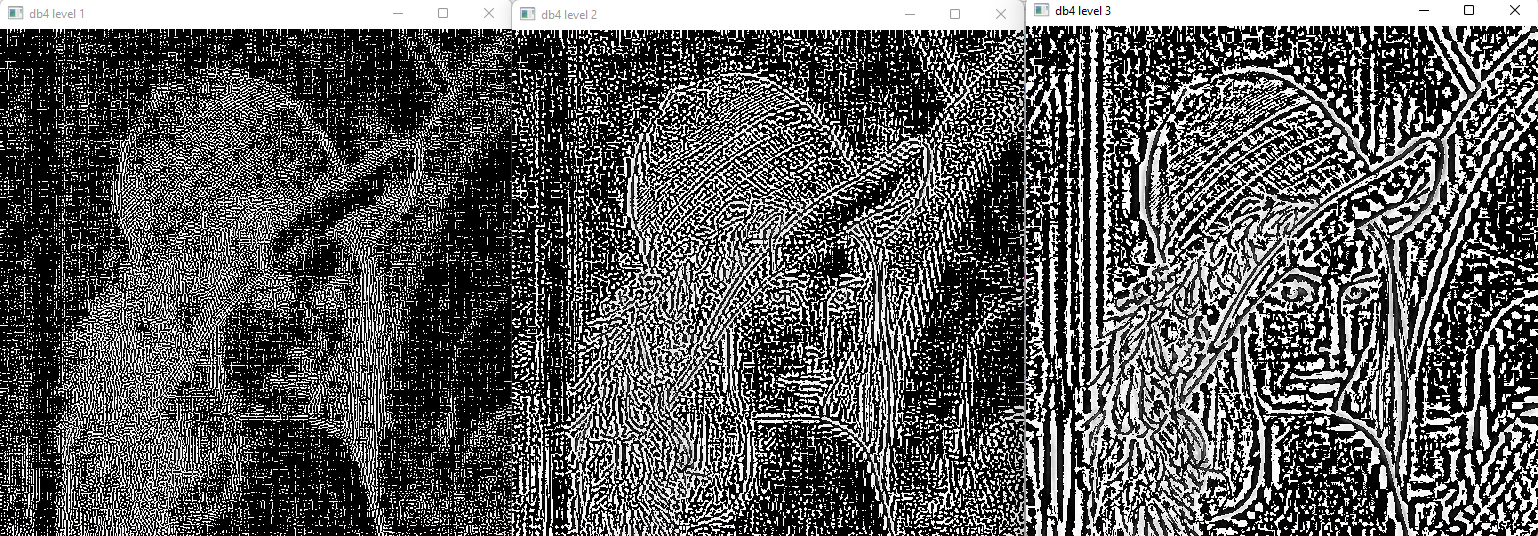
2.

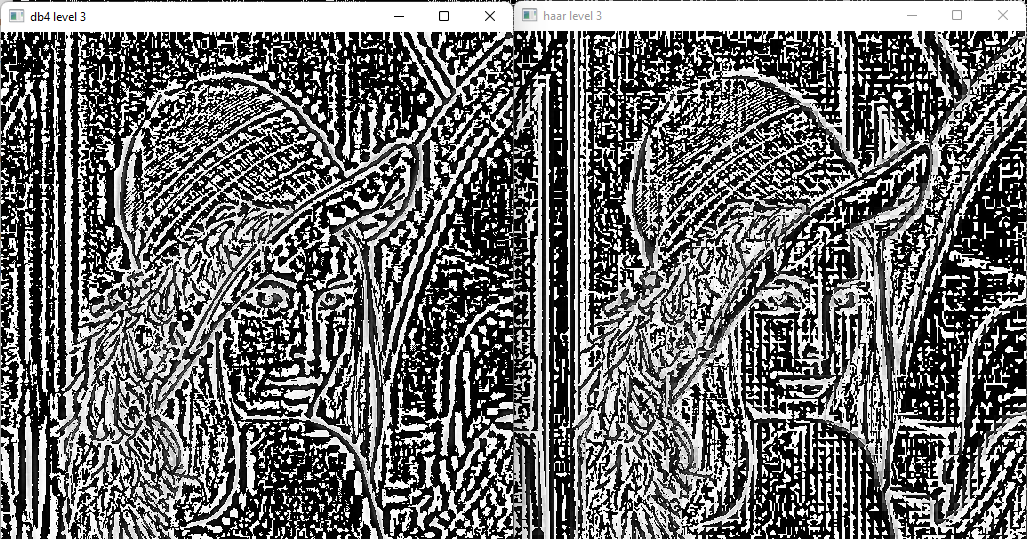
a)

A picture containing text, building

Description automatically generated

b)



c) 

The Daubechies 4 level 3 has a zoomed-in version of the portrait, whereas the Haar level 3 appears smaller than the other one.

import pywt

def wavelet\_transform(imagef32, mode, level):

coeffs = pywt.wavedec2(imagef32, mode, level=level)

coeffs\_list = list(coeffs)

coeffs\_list[0] \*= 0

#reconstructing image to opencv format

new\_image = pywt.waverec2(coeffs\_list, mode)

new\_image \*= 255

new\_image = np.uint8(new\_image)

return new\_image

image2 = cv2.imread('lena.tif', 0)

image2 = np.float32(image2)

image2 /= 255

#haar

new\_haar\_lvl1 = wavelet\_transform(image2, 'haar', 1)

new\_haar\_lvl2 = wavelet\_transform(image2, 'haar', 2)

new\_haar\_lvl3 = wavelet\_transform(image2, 'haar', 3)

#daubechies 4

new\_db4\_lvl1 = wavelet\_transform(image2, 'db4', 1)

new\_db4\_lvl2 = wavelet\_transform(image2, 'db4', 2)

new\_db4\_lvl3 = wavelet\_transform(image2, 'db4', 3)

cv2.imshow('haar level 1', new\_haar\_lvl1)

cv2.imshow('haar level 2', new\_haar\_lvl2)

cv2.imshow('haar level 3', new\_haar\_lvl3)

cv2.imshow('db4 level 1', new\_db4\_lvl1)

cv2.imshow('db4 level 2', new\_db4\_lvl2)

cv2.imshow('db4 level 3', new\_db4\_lvl3)

cv2.waitKey(0)

cv2.destroyAllWindows()