Assignment 4

COMP 478 Image Processing

Etienne Pham Do

40130483

COMP 478 Assignment 4

1.

Etienne Pham Do 40130483

$$= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

$$= P_1(m_1^2 - 2m_1m_G + m_G^2) + P_2(m_2^2 - 2m_2m_G + m_G^2)$$

$$= P_1m_1^2 - 2P_1m_1m_G + P_1m_G^2 + P_2m_2^2 - 2P_2m_2m_G + P_2m_G^2$$

$$= m_G^2(P_1 + P_2) + P_1m_1^2 - 2P_1m_1m_G + P_2m_2^2 - 2P_2m_2m_G \quad (1)$$

$$= (P_1m_1 + P_2m_2)^2 + P_1m_1^2 - 2P_1m_1(P_1m_1 + P_2m_2) + P_2m_2^2 - 2P_2m_2(P_1m_1 + P_2m_2)$$

$$= P_1^2m_1^2 + 2P_1P_2m_1m_2 + P_2^2m_2^2 + P_1m_1^2 - 2P_1^2m_1^2 - 2P_1P_2m_1m_2 + P_2m_2^2 - 2P_1P_2m_1m_2 - 2P_2^2m_2^2$$

$$= -P_1^2m_1^2 - 2P_1P_2m_1m_2 - P_2^2m_2^2 + P_1m_1^2 + P_2m_2^2 \quad (2)$$

$$= P_1m_1^2(1 - P_1) + P_2m_2^2(1 - P_2) - 2P_1P_2m_1m_2$$

$$= P_1P_2m_1^2 + P_1P_2m_2^2 - 2P_1P_2m_1m_2$$

$$= P_1P_2(m_1 - m_2)^2$$

(1) Using $P_1 + P_2 = 1$ & $P_1m_1 + P_2m_2 = m_G$

(2) Using $P_1 + P_2 = 1$

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 $xcos\theta + ysin\theta = \rho$ is used, where ρ is the perpendicular distance from the line to the origin and θ being the angle to which ρ is perpendicular to the line. This translates to sinusoidal curves in the $\rho\theta$ plane instead of relying on a straight line in the ab plane. The $\rho\theta$ 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 $(x-a)^2+(y-b)^2 = r^2$, 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)


```
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)

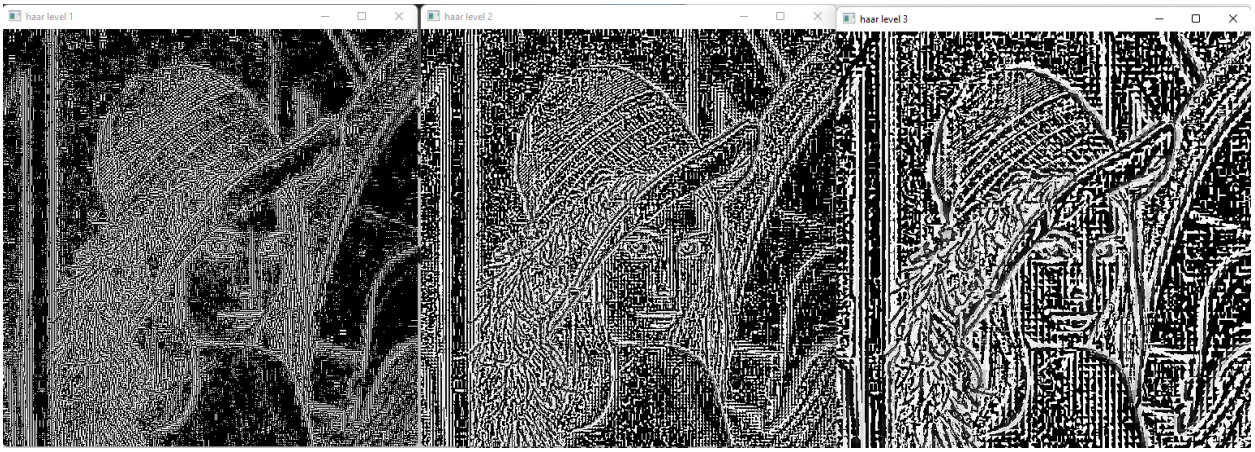

```
image = cv2.GaussianBlur(image, (5,5), 0)

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.

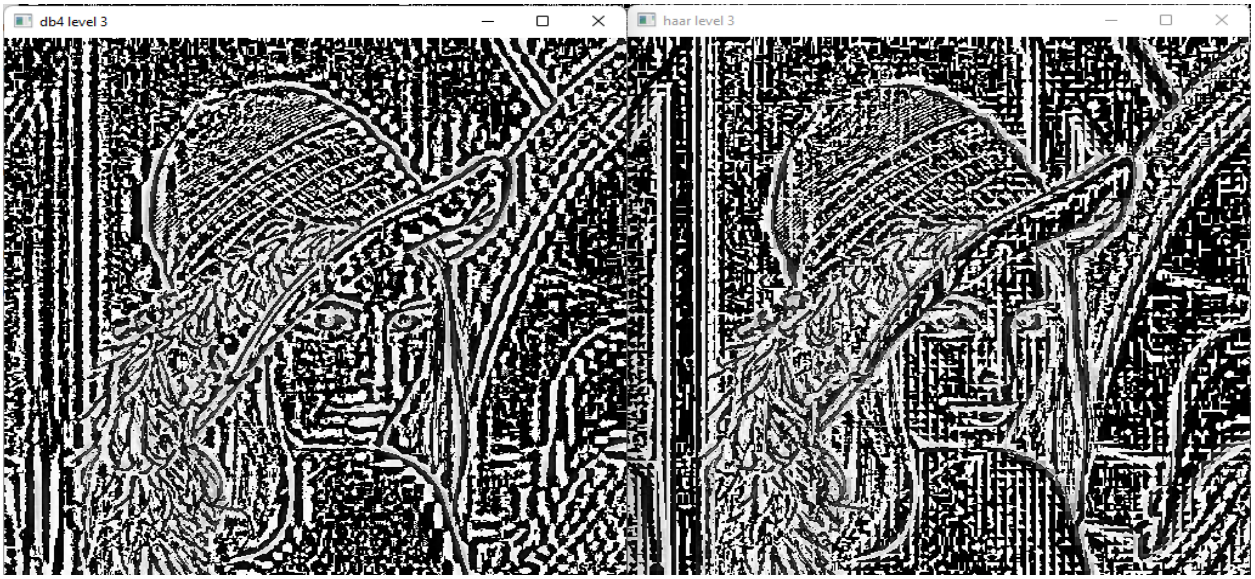
a)


haar level 1
haar level 2
haar level 3

b)


db4 level 1
db4 level 2
db4 level 3

c)


db4 level 3
haar level 3

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()
```