

Computer Vision I_2018

Homework assignment #10

R07522717 機械所製造組碩一 林溫雅

```
#使用 python
#import 套件
import cv2
import numpy as np

def Laplacian(img, mode=None):
    # mode1 是第一種 kernel、mode2 是第二種、mode3 是
    # minimum-variance
    # ker = None
    if mode==1:
        ker = np.array([[0,1,0],[1,-4,1],[0,1,0]])
    elif mode==2:
        ker = np.array([[1,1,1],[1,-8,1],[1,1,1]]) / 3
    elif mode==3:
        ker = np.array([[2,-1,2],[-1,-4,-1],[2,-1,2]]) / 3

    rows, cols = img.shape
    temp_img = cv2.copyMakeBorder(src=img, top=1, bottom=1, left=1,
    right=1, borderType=cv2.BORDER_REPLICATE)
    new_img = img.copy().astype(float)
    for i in range(rows):
        for j in range(cols):
            temp = temp_img[i:i+3, j:j+3]
            new_img[i, j] = np.sum(ker * temp)
    return new_img

def Laplacian_Gaussian(img):
    ker = np.array([[ 0, 0, 0,-1,-1,-2,-1,-1, 0, 0, 0],
                    [ 0, 0,-2,-4,-8,-9,-8,-4,-2, 0, 0],
                    [ 0,-2,-7,-15,-22,-23,-22,-15,-7,-2, 0],
```

Computer Vision I _2018 Homework assignment #10

```
[ -1, -4,-15,-24,-14, -1,-14,-24,-15, -4, -1],  
[ -1, -8,-22,-14, 52,103, 52,-14,-22, -8, -1],  
[ -2, -9,-23, -1,103,178,103, -1,-23, -9, -2],  
[ -1, -8,-22,-14, 52,103, 52,-14,-22, -8, -1],  
[ -1, -4,-15,-24,-14, -1,-14,-24,-15, -4, -1],  
[  0, -2, -7,-15,-22,-23,-22,-15, -7, -2,  0],  
[  0,  0, -2, -4, -8, -9, -8, -4, -2,  0,  0],  
[  0,  0,  0, -1, -1, -2, -1, -1,  0,  0,  0]]
```

```
rows, cols = img.shape  
temp_img = cv2.copyMakeBorder(src=img, top=5, bottom=5, left=5,  
right=5, borderType=cv2.BORDER_REPLICATE)  
new_img = img.copy().astype(float)  
for i in range(rows):  
    for j in range(cols):  
        temp = temp_img[i:i+11, j:j+11]  
        new_img[i, j] = np.sum(ker * temp)  
return new_img
```

```
def Difference_Gaussian(img):  
    ker = np.array([[ -1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1],  
                    [ -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],  
                    [ -4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],  
                    [ -6, -11, -16, -16,  0, 15,  0, -16, -16, -11, -6],  
                    [ -7, -13, -17,  0, 85, 160, 85,  0, -17, -13, -7],  
                    [ -8, -13, -17, 15, 160, 283, 160, 15, -17, -13, -8],  
                    [ -7, -13, -17,  0, 85, 160, 85,  0, -17, -13, -7],  
                    [ -6, -11, -16, -16,  0, 15,  0, -16, -16, -11, -6],  
                    [ -4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4],  
                    [ -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3],  
                    [ -1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1]])
```

```
rows, cols = img.shape  
temp_img = cv2.copyMakeBorder(src=img, top=5, bottom=5, left=5,
```

Computer Vision I _2018 Homework assignment #10

```
right=5, borderType=cv2.BORDER_REPLICATE)
    new_img = img.copy().astype(float)
    for i in range(rows):
        for j in range(cols):
            temp = temp_img[i:i+11, j:j+11]
            new_img[i, j] = np.sum(ker * temp)
    return new_img

def reverse_thresholding(img, threshold=128):
    new_img = np.empty(img.shape)
    new_img.fill(255)
    mask = img >= threshold
    new_img[mask] = 0
    return new_img

original_img = cv2.imread('lena.bmp', 0)

Laplacian1 = Laplacian(original_img, mode=1)
Laplacian2 = Laplacian(original_img, mode=2)
minimum_variance_Laplacian = Laplacian(original_img, mode=3)
Laplacian_of_Gaussian = Laplacian_Gaussian(original_img)
Difference_of_Gaussian = Difference_Gaussian(original_img)

cv2.imwrite('Laplacian1_30.bmp', reverse_thresholding(Laplacian1, 30))
cv2.imwrite('Laplacian2_25.bmp', reverse_thresholding(Laplacian2, 25))
cv2.imwrite('minimum_variance_Laplacian_20.bmp',
reverse_thresholding(minimum_variance_Laplacian, 20))
cv2.imwrite('Laplacian_of_Gaussian_7000.bmp',
reverse_thresholding(Laplacian_of_Gaussian, 7000))
```