

Computer Vision_HW10_R09922093 楊子萱

In this homework, I use python3 and import cv2 to read and write images and use cv2 function to translate image into the binary image. To run my code, use command line and enter python3 hw10.py (<http://hw10.py>). .

In this homework, I used two functions with different masks values and thresholds to complete all pictures.

```
1  def mask(image, i, j, kernel):
2      result = 0
3      for k in range(kernel.shape[0]):
4          for l in range(kernel.shape[1]):
5              result += image[i+k][j+l] * kernel[k][l]
6      return result
7  def laplacian(image, threshold, kernel):
8      laplacian_result = np.full((512,512),0)
9      for i in range(512):
10         for j in range(512):
11             value = mask(image, i, j, kernel)
12             if value >= threshold:
13                 laplacian_result[i][j] = 1
14             elif value <= threshold*(-1):
15                 laplacian_result[i][j] = -1
16             else:
17                 laplacian_result[i][j] = 0
18     return laplacian_result
```

And I used two functions to check neighbor value and do the zero cross edge detecting.

```

1  def check(image,i,j):
2      for a in range(3):
3          for b in range(3):
4              if image[i+a-1][j+b-1] == -1:
5                  return 1
6      return -1
7  def zerocross(image):
8      zerocross = np.full((512,512),0)
9      for i in range(1,image.shape[0]-1):
10         for j in range(1,image.shape[1]-1):
11             if image[i][j] == -1 or image[i][j] == 0:
12                 zerocross[i-1][j-1] = 255
13             elif check(image,i,j) == 1:
14                 zerocross[i-1][j-1] = 0
15             else:
16                 zerocross[i-1][j-1] = 255
17         return zerocross

```

Finally, I input different thresholds and masks to get the result pictures below:

(a) Laplace Mask1 : 15



(b) Laplace Mask 2: 15



(c) Minimum variance Laplacian: 20



(d) Laplace of Gaussian: 3000



(e) Difference of Gaussian: 1

