DSAP Programming Homework 2

B06507002 材料二 林柏勳

My zipper file contains

a) A report : b06507002_report.pdf
b) Two python programs
   - b06507002.py: the implementation of assignment
   - bonus.py: the implementation of LoG and Laplacian
   - Laplacian.py: the implementation of Laplacian

One can run my program using the below method:

(Command line in windows or linux command line)

(Put the input file in the same folder as the program)

Python b06507002.py test.pgm

Python bonus.py test.pgm

Or if one want to change threshold value instead of 128 in assignment:

Python b06507002.py test.pgm threshold

(threshold can be a value from 0 to 255)

Note that the default program in bonus.py is LoG, if you want to run Laplacian, you can comment line 112~113 and uncomment line 115~117 in bonus.py, then run bonus.py again as above.

Another way to run Laplacian is use the below command.

Python Laplacian.py test.pgm

1. Assignment

First of all, I have to read the input file, so I split the input file name by "." and add "_out.pgm" and "_b_out.pgm" as my output file name.

Second, I read the file and split the file by "\n" and remove all the comments which are lines that contains "#". Then, I split the data by "\n" and append all element into a list. Now, I get a list containing the list ["P4",width,height,……(data)].If the first element in the list is not "P4", I will write "error" in the output file and exit.

After that, I further divide the data by the width. Now, I have a h*w list.

I then go through a double loop to judge if the if the pixel is on the edge.

Let h stand for height and w stand for width.

For every (i,j) element in the list.

If the pixel is on the border, i.e i=0 or i=h-1 or j=0 or j=w-1, the value for output is 255.

Else, calculate a to i for (*i,j*) element e using the below relationship.

Then, let x=(c+2*f+i2)-(a+2*d+g) and y=(g+2*h2+i2)-(a+2*b+c).

If $\sqrt{x^2+y^2} > threshold$, the value for output (i,j) is 0.

Else: the value for output (i,j) is 255.

h\w	j-1	j	j+1
i-1	a	d	g
i	b	e	h2
i+1	c	f	i2

Finally, output the result in a new pgm file.
Note that two consecutive rows are separated by “\n” and two neighboring elements in a row are separated by “   ”.

2. Bonus

The assignment is an implementation of Sobel operator, which uses the first derivative for edge detection. Below, I will implement Laplacian and Laplacian of Gaussian, a method of using the second derivative for edge detection.
First, I will talk about the mathematical background of Sobel operator and Laplacian and Laplacian of Gaussian operator.

a. The points lies on an edge are detected by
   1) Detecting the local maxima or minima of the first derivative.
   2) Detecting the zero-crossings of the second derivative.
b. Sobel operator
   For a pixel on (i,j) and consider a 3×3 neighborhood, we can compute the partial derivative $\mathrm{M_x} = \frac{\partial}{\partial \mathrm{x}}$ and $\mathrm{M_y} = \frac{\partial}{\partial \mathrm{y}}$ using the finite difference.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} and\ M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

   The number “2” in the kernel is used to emphasize the pixel is closer to the center of the mask.
   The judgement of an edge is that if $|M_x f| + |M_y f| > threshold.$
c. Laplacian operator
   The Laplacian operator is defined as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

   Where we can also compute $M_{xx} = \frac{\partial^2}{\partial x^2}\ and\ M_{yy} = \frac{\partial^2}{\partial y^2}$ using the finite difference.

$$M_{xx} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} and\ M_{yy} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

So the one common Laplacian mask is that

$$\nabla^2 = M_{xx} + M_{yy} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The judgement of edges is that

$$\begin{cases} \nabla^2 f(i,j) \times \nabla^2 f(i,j+1) < 0 \; and \; |f(i,j) - f(i,j+1)| > threshold \\ \nabla^2 f(i,j) \times \nabla^2 f(i+1,j) < 0 \; and \; |f(i,j) - f(i+1,j)| > threshold \end{cases}$$

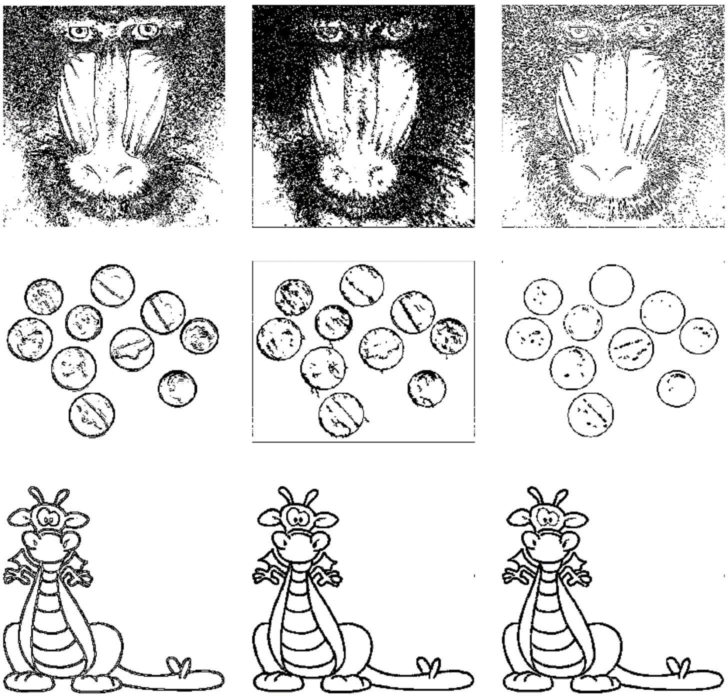
Note that I applied the periodic boundary condition.

d. However, the Laplacian operator is more sensitive to noise because we differentiate the function twice. As a result, one can use the Gaussian operator to eliminate the noise and then apply the Laplacian operator.
From linear algebra, we know that we can combine two convolution into one. Therefore, the Laplacian of Gaussian (LoG) operator in 2D can be deduced as below:

$$LoG = \nabla^2 g(x,y) = \nabla^2 \left\{ e^{\frac{-(x^2+y^2)}{2\sigma^2}} \right\} = \{ \frac{x^2 + y^2 - 2\sigma^2}{4\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \}$$

After applying LoG to the image, we can use the same criterion in 2.c to judge if an pixel is on the edge.
Note that I applied the periodic boundary condition.

e. The result is in the below, the first column is the implementation of Sobel with threshold=128 and its kernel is in b., the second column is the implementation of Laplacian with threshold=32 and its kernel is in c., and the third column is the implementation of LoG with $\sigma = 1.6$ and its kernel size is (4,4).
The three testing pictures are “baboon”, ”coins”, and “dragon” respectively.



We can see that in “baboon”, LoG is less sensitive to noise than Laplacian and it is ok compared with Sobel. However, LoG can not detect the edge completely in “coins”. In “dragon”, both Laplacian and LoG are smoother than Sobel. If you look at the lines, you can find that Sobel has hollow lines while the lines in the other two are solid.

Reference:

1. http://www.me.umn.edu/courses/me5286/vision/VisionNotes/2017/ME5286-Lecture7-2017-EdgeDetection2.pdf
2. https://www.cs.auckland.ac.nz/~rklette/CCV-CIMAT/pdfs/B03-Normalization.pdf
3. http://140.117.156.238/course/CSSV/2011/%E7%9B%A3%E6%8E%A7%E8%A6%96%E8%A8%8A_ch3.pdf
4. http://silverwind1982.pixnet.net/blog/post/250092094-laplacian-mask-%E7%9A%84%E7%B3%BB%E6%95%B8%E7%94%B1%E4%BE%86
5. https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm
6. https://dsp.stackexchange.com/questions/37673/what-is-the-difference-between-difference-of-gaussian-laplace-of-gaussian-and/47909#47909
7. http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm