HW3

Date: 2021. 09. 27

Student ID: 21600635

Name: Bomoon JUNG

1. Homework 3

1) Please follow the instructions below.

- ✓ Read a “Lena.png”
- ✓ Perform average filtering on the left half of the image
  - Set the mask size as (7, 7)
  - blur(in, out, Size(val1, val2))
    - ➢ Blurs an image using the normalized box filter
    - ➢ in: input image, out: output image, Size(val1, val2): blurring kernel size
- ✓ Read “moon.png”
- ✓ Perform sharpening on the right half of the image
  - Perform sharpening using second derivative
  - Laplacian(in, out, CV_16S);
    - ➢ Calculates the Laplacian of an image
    - ➢ In: input, out: output, CV_16S: desire depth of output
- ✓ Read “saltnpepper.png”
- ✓ Perform median filtering on the image
  - Set aperture size a 9
  - medianBlur(in, out, val)
    - ➢ Blurs an image using the median filter
    - ➢ In:src, out: dst, val: aperture size(must be odd and greater than 1)
- ✓ Display 6 windows
  - The name of each window should be
    - ➢ “lena”
    - ➢ “lena_filtered”
    - ➢ “moon”
    - ➢ “moon_fiiltered”
    - ➢ “saltnpepper”
    - ➢ “saltnpepper_filtered”

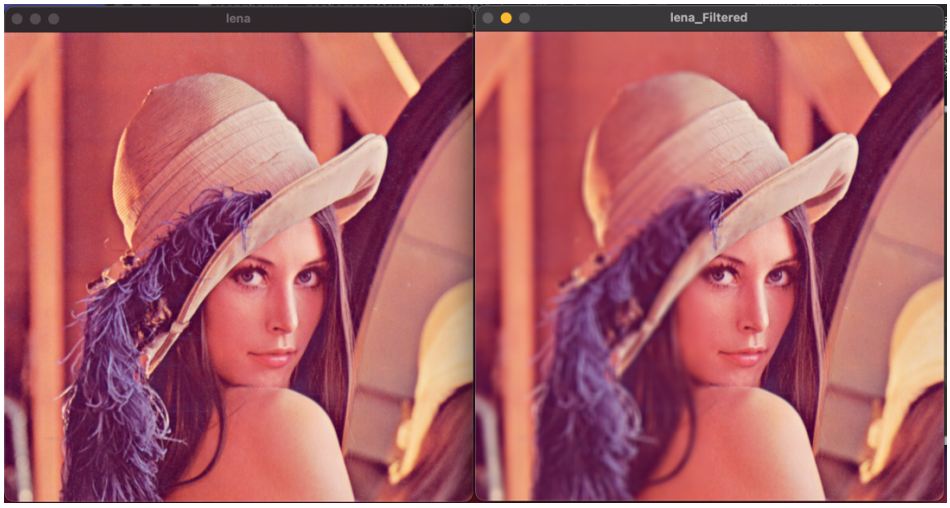

Figure 1. results of lena and lena_filtered

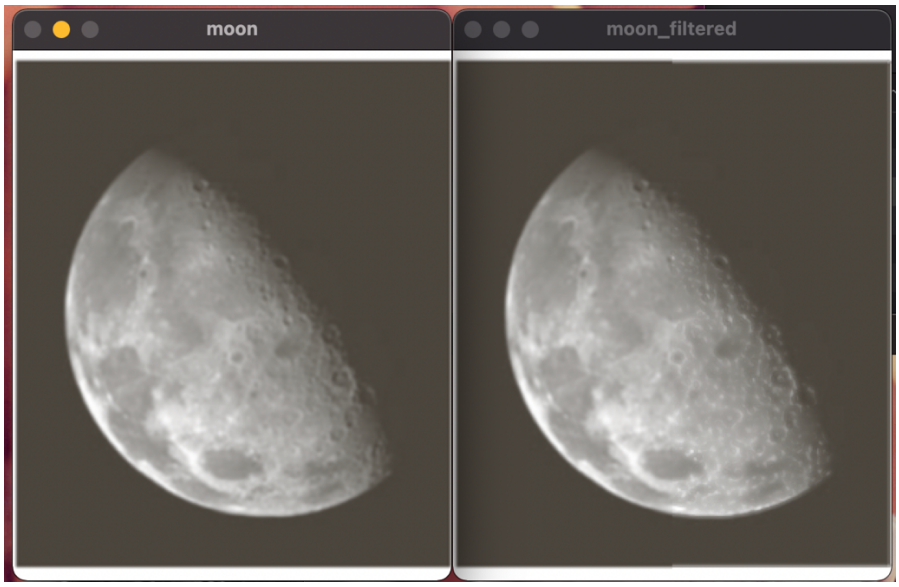

Figure 2. results of moon and moon_filtered

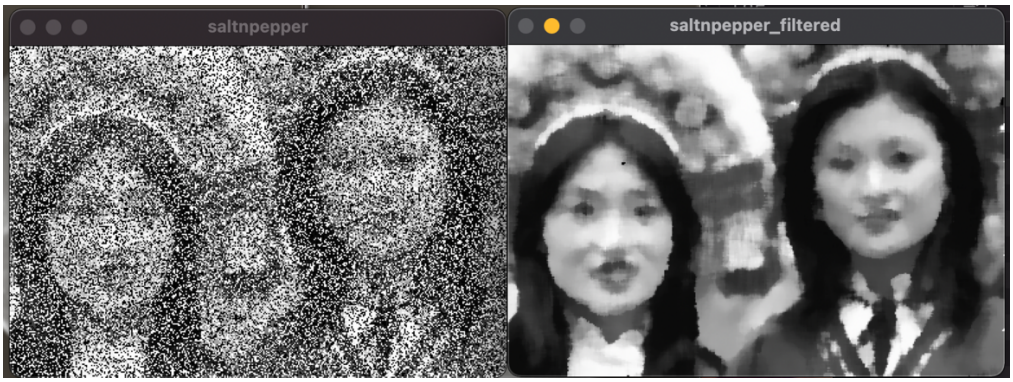

Figure 3. results of saltnpepper and saltnpepper_filtered

2) explanation

You can use ‘Rect’ to apply a filter to only the desired area of the image. Therefore, I uses ‘Rect’ to blur only the left half of the ‘lena_filtered’ image. Similarly, ‘moon_filtered’ can be sharped using ‘Rect’.

Sharping is implemented using unsharp making. Unsharp making means that after blurring the original image, if you do the 'original image – blur image', the 'unsharp mask' comes out. Finally, if you do the 'original image + unsharp mask', you can get a sharpened image.

The median fileter can reduce the noise in the image. This filter is performed through the 'medianBlur' function.

3) Source code

```
#include <opencv2/opencv.hpp>
#include <iostream>

using namespace cv ;
using namespace std ;

int
main () {
    // 1. Read "Lena.png"
    Mat lena = imread("./resources/Lena.png") ;

    if ( lena.empty() ) {
        cout << "no such file" << endl ;
        return 0 ;
    }

    Mat lena_filtered = lena.clone() ;
    int height = lena.rows ;
    int width = lena.cols ;

    // To perform filtering on the left half of the image
    Rect rect_1(0, 0, width/2, height) ;

    // Set the mask size as (7, 7)
    blur(lena(rect_1), lena_filtered(rect_1), Size(7, 7)) ;

    imshow("lena", lena) ;
    imshow("lena_Filtered", lena_filtered) ;
```

```
    // 2. Read "moon.png"
    Mat moon = imread("./resources/moon.png") ;

    if ( moon.empty() ) {
        cout << "no such file" << endl ;
        return 0 ;
    }

    Mat laplacian ;
    Mat abs_laplacian ;
    height = moon.rows ;
    width = moon.cols ;

    // To perform filtering on the right half of the image
    Rect rect_2(width/2, 0, width/2, height) ;

    // Unsharping masking
    GaussianBlur(moon, moon, Size(3, 3), 0, 0, BORDER_DEFAULT) ;
    Mat moon_filtered = moon.clone() ;

    // Second devation
    Laplacian(moon_filtered, laplacian, CV_16SC3, 1, 1, 0) ;
    convertScaleAbs(laplacian, abs_laplacian) ;

    // add second devation to origin image
    moon_filtered(rect_2) += abs_laplacian(rect_2) ;

    imshow("moon", moon) ;
    imshow("moon_filtered", moon_filtered) ;


    // 3. Read "saltnpepper.png"
    Mat saltnpepper = imread("./resources/saltnpepper.png") ;
    Mat saltnpepper_filtered ;

    if ( moon.empty() ) {
        cout << "no such file" << endl ;
        return 0 ;
    }

    // Perform median filtering(Set aperture size as 9)
    medianBlur(saltnpepper, saltnpepper_filtered, 9) ;

    imshow("saltnpepper", saltnpepper) ;
    imshow("saltnpepper_filtered", saltnpepper_filtered) ;

    waitKey(0) ;

    return 0 ;
}
```