

HW3 Report

Computer Vision
Section 02
21800181 Kim Jisu

● HW 3

- Purpose

This program has three purpose that is performing average filtering, sharpening and median filtering. In the average filtering case, using image “lena.png” and performing filter on the left half of the image (mask size is 7x7). In the sharpening case, using image “moon.png” and performing sharpening on the right half of the image (using second derivative). In the median filtering case, using image “saltnpepper.png” and performing filter on the image (aperture size as 9).

- Principle

```
void cv::blur (InputArray src, OutputArray dst, Size ksize, Point anchor = Point(-1,-1), int borderType = BORDER_DEFAULT )  
    Blurs an image using the normalized box filter.  
void cv::GaussianBlur(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY = 0, int borderType = BORDER_DEFAULT )  
    Blurs an image using a Gaussian filter.  
void cv::Laplacian (InputArray src, OutputArray dst, int ddepth, int ksize = 1, double scale = 1, double delta = 0, int borderType = BORDER_DEFAULT )  
    Calculates the Laplacian of an image.  
void cv::medianBlur(InputArray src, OutputArray dst, int ksize )  
    Blurs an image using the median filter.
```

- Process

1. Average filtering

Deep copy from lena to lena_filtered.

shallow copy from lena_filtered to temp.

set ROI Rect(0,0,lena.cols/2,lena.rows) at temp. (to perform filtering on the left side of image)

perform blur at temp.

because perform shallow copy from lena_filtered to temp, the lena_filtered is filtered on the left half of the image.

2. Sharpening

Deep copy from moon to moon_filtered. Deep copy from moon to t_m.

Performing gaussian bluring t_m.

Do Laplacian function. (get l_matrix)

Do convertScaleAbs function. (convert 8 bit) (get a_matrix)

Re_matrix = t_m + a_

Set moon_filtered right half of image to 0.

Set re_left half of image to 0.

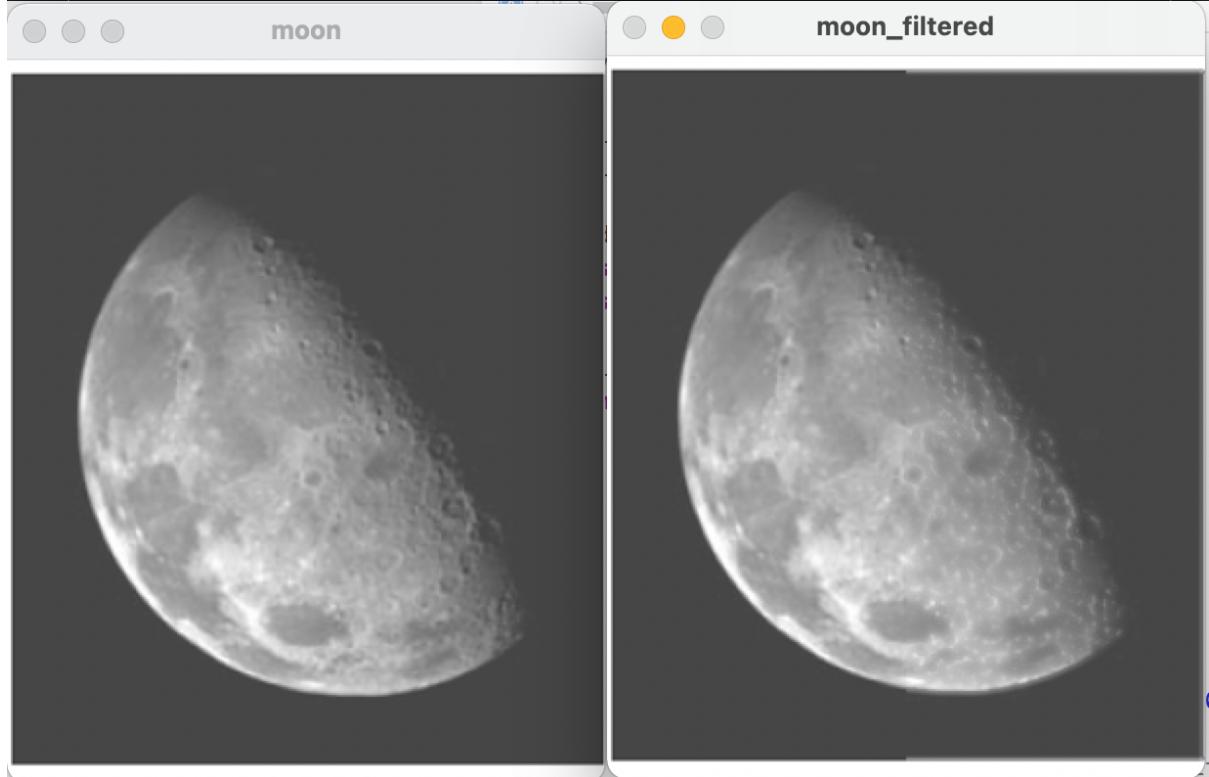
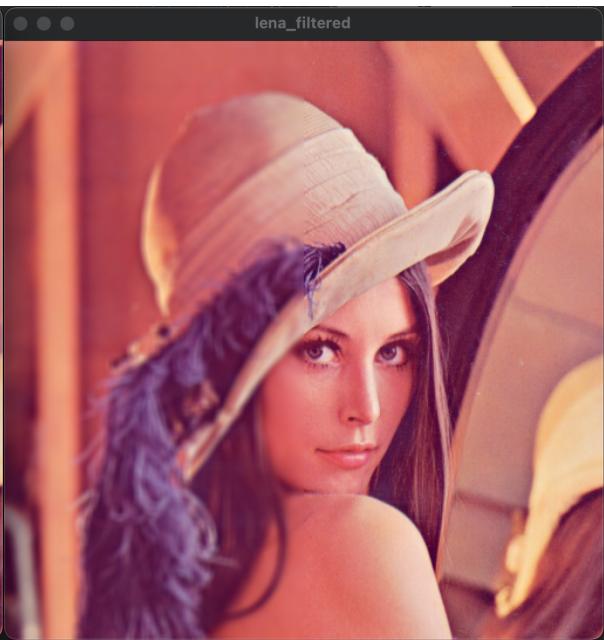
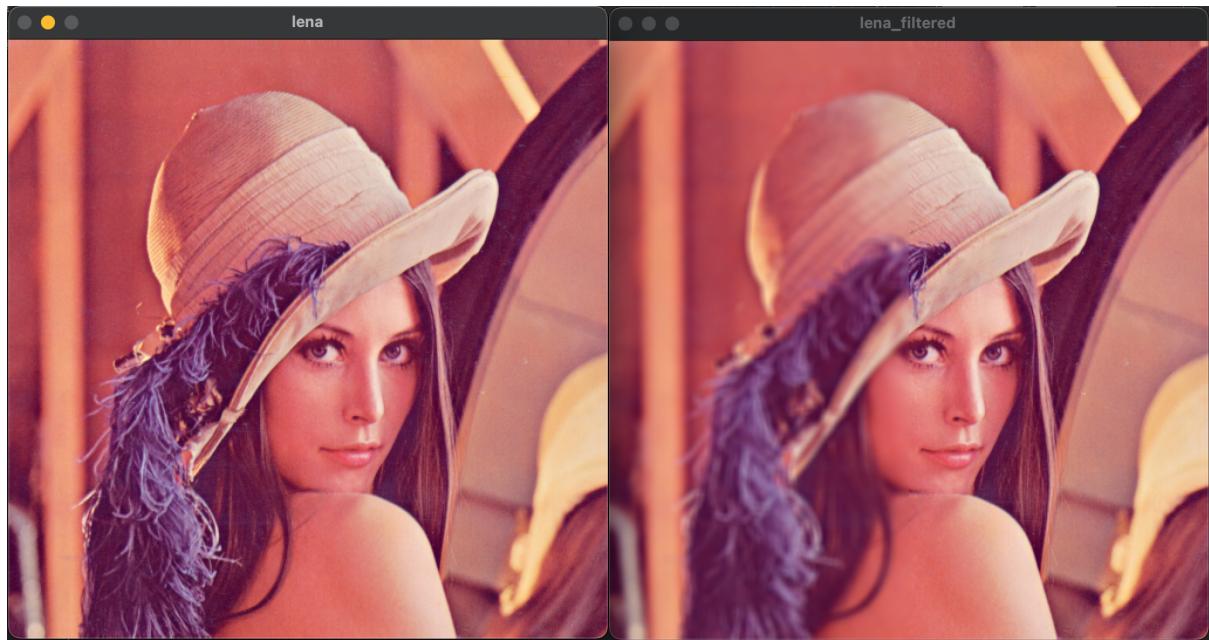
moon_filtered = moon_filtered + re_

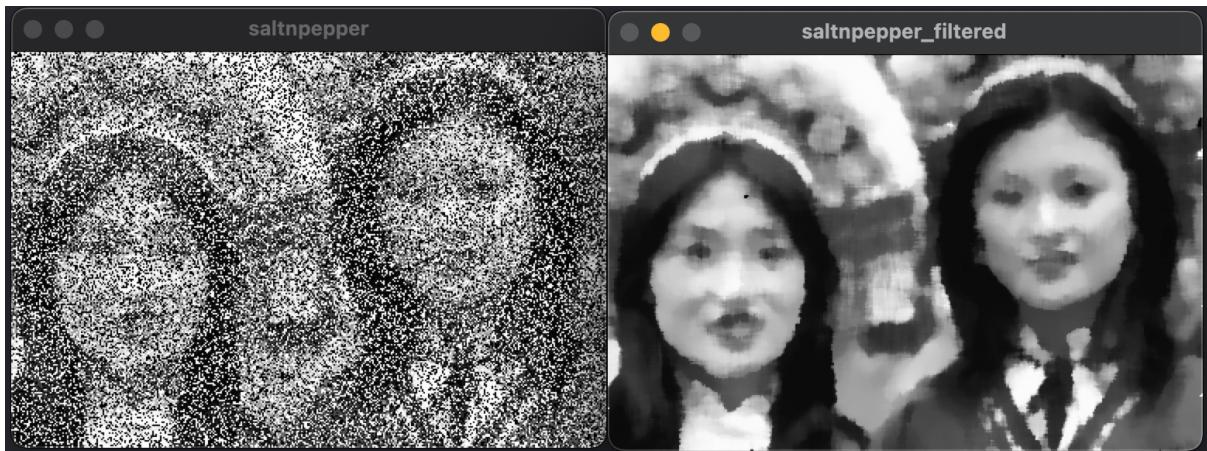
moon_filtered is filtered on the right half of the image.

3. Median filtering

Do medianBlur function (aperture size as 9)

- Result





- Code

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

using namespace std;
using namespace cv;

int main(){

    Mat lena, lena_filtered, temp;
    Mat moon, moon_filtered;
    Mat t_m,l_,a_,re_;
    Mat saltnpepper, saltnpepper_filtered;

    lena = imread("Lena.png");
    moon = imread("moon.png",0);
    saltnpepper = imread("saltnpepper.png",0);

    lena_filtered = lena.clone();
    temp = lena_filtered;
    Rect rect_a(0,0,lena.cols/2,lena.rows);
    temp = temp(rect_a);
    blur(temp,temp,Size(7,7));

    moon_filtered = moon.clone();
    t_m = moon.clone();
    GaussianBlur(t_m, t_m, Size(3,3), 0,0,BORDER_DEFAULT);
    Laplacian(t_m, l_, CV_16S);
    convertScaleAbs(l_,a_);
    re_ = a_ + t_m;

    for(int i = 0; i<moon.rows;i++)
        for(int j = moon.cols/2; j< moon.cols;j++)
            moon_filtered.at<uchar>(i,j) = 0;
    for(int i = 0; i<moon.rows;i++)
        for(int j =0; j< moon.cols/2;j++)
            re_.at<uchar>(i,j) = 0;
```

```
moon_filtered = moon_filtered + re_;

medianBlur(saltnpepper, saltnpepper_filtered, 9);

imshow("lena",lena);
imshow("lena_filtered", lena_filtered);
imshow("moon", moon);
imshow("moon_filtered", moon_filtered);
imshow("saltnpepper", saltnpepper);
imshow("saltnpepper_filtered", saltnpepper_filtered);

waitKey();
return 0;
}
```

- References

<https://docs.opencv.org/master/>