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”

텍스트, 사람이(가) 표시된 사진

자동 생성된 설명

**Figure 1. results of lena and lena\_filtered**

**텍스트, 실내, 진드기, 새각류의갑각류이(가) 표시된 사진

자동 생성된 설명**

**Figure 2. results of moon and moon\_filtered**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**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 ;

}