**IPV Assignment 3 (week 3)**

# **Goal: Image filtering(non-linear filtering) and applications**

## (un)sharping

☞ <ftp://qiftp.tudelft.nl/DIPimage/docs/Introduction_to_DIPimage.pdf> Finish the following parts:

* the part 5((un)sharpening & local maximum and minimum filters)
* the part 6 point operations (Histogram-based operations & thresholding)

**Q1: for Part 5 (un)sharpening:** The two (un)sharping filters (“**a-laplace(a)**” and “**2\*a - gaussf(a)**”) are not the same. Can you explain the difference of them? How are they alike?

## Remove Noise Using an Averaging Filter and a Median Filter

This example shows how to remove salt and pepper noise from an image using an averaging filter and a median filter to allow comparison of the results. These two types of filtering both set the value of the output pixel to the average of the pixel values in the neighborhood around the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called outliers). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image.

Note: Median filtering is a specific case of order-statistic filtering, also known as rank filtering. For information about order-statistic filtering, see the reference page for the ordfilt2 function.

Read image into the workspace and display it.

I = imread('eight.tif');

figure

imshow(I)

For this example, add salt and pepper noise to the image. This type of noise consists of random pixels being set to black or white (the extremes of the data range).

J = imnoise(I,'salt & pepper',0.02);

figure

imshow(J)

Filter the noisy image, J, with an averaging filter and display the results. The example uses a 3-by-3 neighborhood.

Kaverage = filter2(fspecial('average',3),J)/255;

figure

imshow(Kaverage)

Now use a median filter to filter the noisy image, J. The example also uses a 3-by-3 neighborhood. Display the two filtered images side-by-side for comparison. Notice that medfilt2 does a better job of removing noise, with less blurring of edges of the coins.

Kmedian = medfilt2(J);

imshowpair(Kaverage,Kmedian,'montage')

**Q2**: Try out the above example and understand it.

**Goal: Make your own applications and demo for image filtering part.**

Requirements:

* you choose **at least 2** applications: **(un)sharpening, smoothing, blurring, noise reduction**

Must have:

1. choose your own images
2. indicate which type of filter you used:

* linear --- explain how/why you design **mask/kernel**
* non-linear filter--- explain how/why you choose it

1. show the comparison of the original image and filtered image to demonstrate the applications

**Q3:** you choose **at least 2** applications and implement them in matlab.

Must have:

1. you need to explain the steps you use to achieve the goal
2. choose your own images
3. show the comparison of the original image and filtered image to demonstrate the applications

**Application Goal: edge detection.**

|  |  |
| --- | --- |
| Input image | Output image |
| http://www.cs.uregina.ca/Links/class-info/425-nova/Lab3/Exercise/two_cats/two_cats.jpg | http://www.cs.uregina.ca/Links/class-info/425-nova/Lab3/Exercise/two_cats/two_cats_edge.jpg |

**Application Goal: image thresholding**.

|  |  |
| --- | --- |
| Input image | https://nl.mathworks.com/help/examples/images/win64/SegmentImageIntoThreeLevelsUsingTwoThresholdsExample_01.png |
| Output image | https://nl.mathworks.com/help/examples/images/win64/SegmentImageIntoThreeLevelsUsingTwoThresholdsExample_02.png |

## **Application Goal: Image Enhancement**

|  |
| --- |
| https://nl.mathworks.com/help/examples/images/win64/LowLightImageEnhancementExample_06.png |