2D Signal Processing[1]

Dr. François Pitié

Electronic and Electrical Engineering Dept.

pitief@tcd.ie www.mee.tcd.ie/~sigmedia

THIS LABORATORY SESSION COUNTS FOR 5% OF THE TOTAL 4C8 MARK

1 Lab Information

This is the second of 4 lab handouts in 4c8 and accounts for 5% of the overall 4C8 mark. In this lab, you will be required to compile a report that answers specific instructions included in the report. There are other instructions that do not require answers in the report but are necessary to complete the lab.

IN THIS LABORATORY HANDOUT SPECIFIC INSTRUCTIONS ARE INDICATED WITH THIS SYMBOL : □ .

ENUMERATED INSTRUCTIONS REQUIRE ANSWERS IN YOUR SUBMITTED REPORT.

1.1 Submission Requirements

Labs Reports should be typed up and submitted electronically using the **PDF file format** via the module's blackboard page by the specified deadline. Remember to put your name and student number on the top of your reports.

Penalties for Late Submission

- less than 15 minutes past the deadline - no penalty.
- greater than 15 minutes and less than 2 hours past the deadline - loss of 25% of the final mark.
- greater than 2 hours past the deadline - loss of 50% of the final mark.

Plagiarism Policy

Any submitted code or answers that can be seen to be plagiarised from other sources will result in 0 marks being awarded for that section of the lab and may result in you being awarded 0 marks for the entire assignment. Anti-plagiarism software will be used on submitted materials.

[1] This lab was originally written by Prof. Anil Kokaram and ammended by Dr. David Corrigan.

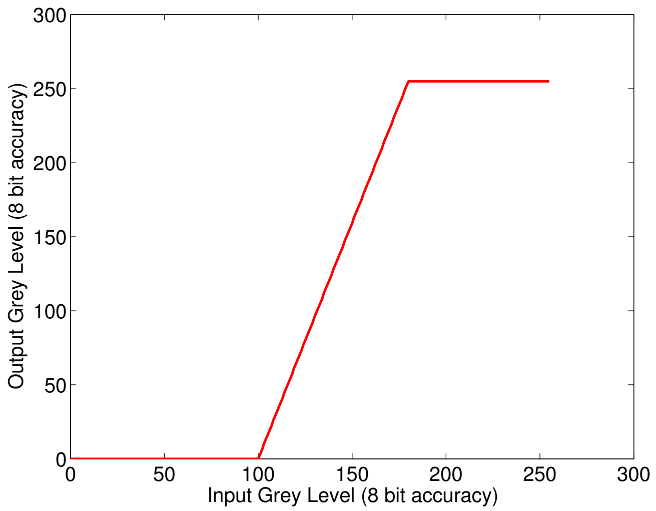


Figure 1: A pixelwise transformation mapping input grey levels 100:180 to 0:255 in the output picture.

2 Contrast enhancement

In poorly contrasted images, the pixel values do not occupy all the available levels of grey. Other times, some parts of the image can be well contrasted and in other parts the image is poorly contrasted. Contrast stretching is a simple linear operation for making a certain range of input grey levels map onto the whole available range. Consider an image in which most of its greylevels are between 100 and 180. Figure 1 shows a mapping from input to output which ensures that values below 100 get mapped to zero on the output, values between 100 and 180 get linearly mapped or *stretched* to occupy 0 - 255 levels on the output and values greater than 180 get clipped to 255 on the output.

1. Write an .m script to do a linear contrast stretch on the jfk image from lab 1 so that the image values in the range **55 to 200** are stretched to occupy the whole 8 bit grayscale range. The code will have to ensure that any intensities outside of the range 0:255 are appropriately clipped. Display the output image in a separate figure window when you are done. In your report, include your script along with the image you generate. Explain clearly how the code implements all of the listed requirements.

   Note: The code that you submit for this section will be partially assessed on its neatness, clarity and computational efficiency. Comments are unnecessary as you already have to explain how you implement contrast stretching in your answer.

3 The Gaussian Filter

The Gaussian filter is a popular low pass filter. It is defined according to the equation

$$f[h,k] = \exp\left(-\frac{h^2+k^2}{2\sigma^2}\right)$$

where σ^2 is the variance of the filter. The gaussian filter is usually truncated with a rectangular window to form an FIR filter. As a result, the number of coefficients/taps in the filter is an important parameter. Also, the sum of the filter coefficients is normalised to 1.

The filter can easily be implemented separably for images. The vertical and horizontal filter masks are

$$f_1[h] = \exp\left(-\frac{h^2}{2\sigma^2}\right)$$

and

$$f_2[k] = \exp\left(-\frac{k^2}{2\sigma^2}\right).$$

In this lab, because we use the same variance and we are interested in masks that have the same number of rows and columns then $f_1 = f_2$ after normalisation of their sums to 1.

1. Write a matlab function that implements a gaussian low pass filter according to the above criteria. Include the code in your submission and explain what each line of your function does. The function should meet the following criteria.
   - The function should take 3 inputs. The first input specifies the variance of the filter mark (positive float/double), while the second specifies the size (positive integer). The third input is a string which either has the value 'combined' or 'separable'. This parameter is a flag which returns a 1D gaussian filter if the value is separable and a 2D filter if the value is combined.
   - There is one output which contains the filter mask/kernel. The size of the ouput is $N\times 1$ (or $1 \times N$) if the separable mode is specified or is $N \times N$ if the combined mode is specified. (N is the value of size specified by the 2^{nd} input parameter).
   - The function should cater for the possibilitiy of an invalid value of the 3^{rd} parameter. You should define some default behaviour (either return the separable or combined version of the filter mask).
   - The maximum value of the filter mask should be at the centre of the array.
   - The filter must be symmetric for both odd and even filter lengths.
2. Explain why it is necessary that all the coefficients of the low pass filter sum to 1.

3. Changing the value of `N` to 21 and using the matlab functions `tic` and `toc`, record the time that it takes to apply the system to the image 1000 times. The filter should be applied to both the rows and the columns of the image. In your answer include the code you use and write down the time taken.

4. Change your m-file to use the full 2D implementation of the filter (ie. not separably). Include your code and write down the time it takes to apply this separable implementation 1000 times as before.

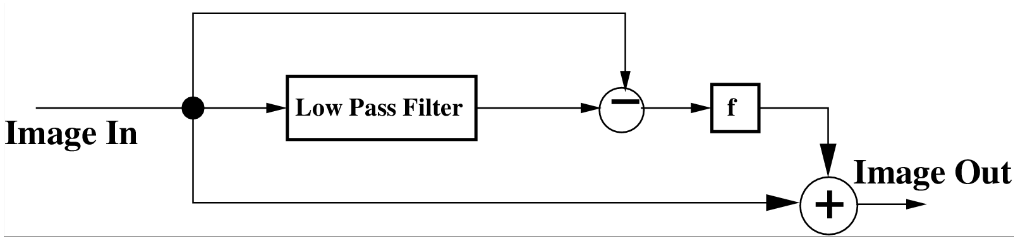
5. Is there any difference in execution time between these two implementations? Explain your findings.

4 Unsharp Masking

It is possible to use a low pass filter to generate high pass information by subtracting the output of the low pass filter from the input signal.

1. Write an m-file that loads up the `sigmedia06907.tif` image and extracts the three colour planes. Then using the Gaussian filter above as a low pass filter, implement the system in the figure below and apply it to each channel of the input image. The Gaussian filter (Low Pass Filter) has user defined variance σ^2 and N taps. The fraction of high pass information that is added back into the image is f. Use $N = 15$ and $\sigma^2 = 2.5^2$.

   Along with your code the answer should explain how the unsharp masking filter is implemented. The code that you submit for this section will be partially assessed on its neatness, clarity and computational efficiency.



2. By applying the system and observing the output image in colour, pick a value of f that gives the best looking result. Write the value of f.

3. Describe the appearance of your output compared with the input.

4. Justify the use of the selected value.

5. Now process the `pool.01.bmp` image with these same settings. Is the result as good? Explain your findings.