RRY025-- Image Processing

Exercises 2 – Lecture 2 – Image Enhancement II

EX 1. Image smoothing

Image smoothing can be helpful in detecting signal in noisy images, i.e., in increase their signal-to-noise ratio. Load the noisy image, **load('noisy.mat')**, which is an image an astronomer took of an object close to a nearby star. The image has low signal-to-noise ratio (of about 0.2), and it is dominated by gaussian noise.

Display with **imshow(noisy,[])**. Is any structure detectable to the eye?

When using an averaging kernel, the SNR increases with the square root of the amount of data. When using a Gaussian kernel, the "amount of data" can be computed as the number of pixels within the gaussian sigma range. How should one set the gaussian sigma parameter so that the resulting SNR = 3?

Create a 2D gaussian filter function that has the width **g_sigma** corresponding to SNR=3, and size **filtersize = g_sigma * 3,** using **g = fspecial('gaussian', filtersize, g_sigma).** Then filter the image with **filtim = filter2(g, noisy)**.

Try to change the **g_sigma**. What is the best filtersize? What sets the limits for the useful sizes to use?

EX 2. Unsharp masking; high-pass filtering; high-boost filtering

Experiment with the basics of high-pass imaging.

Load and display an image **flower = double(imread('flower.tif '))**.

Try to sharpen this image by first smoothing it and then removing that smoothed image from the original ('unsharp masking'). Use the **fspecial('average',3)** to produce a 3x3 smoothing filter. Next make a smoothed version using **filter2** and display that. Finally remove the smoothed version from the original and display it. This image is a so called 'high pass' image, because it only contains high spatial frequencies.

Which regions have large amplitude in this image and why? What is the average pixel value in this image?

The high-pass image highlights regions where the image changes, but we lose general information about the object (it's hard to recognise it as a flower). If our aim is to make simply a sharpened version of the original one can combine the high-pass image with the original ('high boost filtering'). Create (original image) + (A x High pass image)) where A is an adjustable parameter (start with A=1). Display this final image alongside the original image, is it significantly improved?. Vary the A parameter to find the best (=most pleasing) result.

EX 3. High-pass filtering – computed as a convolution

As learned during the lecture, sharpening (and high-pass filtering) is a convolution (/correlation) operation. We do not have to make a smoothed version of the image and then subtract from the original to achieve high-pass filtering. Prepare a matlab script that performs the same high-pass filtering as above using convolution.

Roadmap for the flower image:

A suitable high-pass filter can be constructed as: **hp = (1/9) .* [-1 -1 -1; -1 8 -1; -1 -1 -1]**. (do you recognize this filter?)

Convolve using **flower_convolved = filter2(hp, flower)**.

Try the same procedure for the spine4 image (**load('spine4.mat');**). Can you detect the fractures of the spine better using high-pass filtering…?

EX 4. Edge Detection

Experiment with the basics of edge detection using directional, asymmetric filters. Consider finding edges from a can of coke.

Load the image of a coke can: **load('coke2.mat');**.

Create the Sobel filters: **sbh=[1 2 1; 0 0 0; -1 -2 -1]** or vertical direction **sbv=[1 0 -1; 2 0 -2; 1 0 -1]**. Correlate the image 'coke' with these different filters using **filter2** and inspect the results.

The values in the sharpened images can be positive or negative, depening on the direction of the gradient w.r.t the filter. Isolate all edges by computing the absolute values of the high-pass images in both vertical and horizontal directions: **cokea=abs(cokeh) + abs(cokev);**.

Create an edge mask by detecting from the image features (=edges) that are brighter than some threshold value. Use the task **im2bw** to create the mask. Inspect the histogram of **cokea** image to find a good threshold value.

EX 5. Noise Reduction via image Smoothing

Find out via experimenting how well different filters remove noise from an image.

Add gaussian or 'salt and pepper' noise to the image (use, e.g., the flower image). Then filter that image using simple averaging and median filters. Use varying filter sizes (e.g., 3x3 and 7x7 pixels) Which filter removes better the noise and why? What kind of effect using the "wrong" filter has on the image?