Exercises for the lecture on Image Enhancement

A. Kaestner
Neutron Imaging and Activation Group, Paul Scherrer Institut
email:anders.kaestner@psi.ch phone: 056 310 42 86

February 26, 2014

Introduction

The following exercises are made to get a practical training on the topics covered by the image enhancement lecture. They will also serve an introduction to the use of matlab and the image processing and wavelet toolboxes. It is intended that the image enhancement exercises are solved using matlab.

Some scripts are provided for your convenience as well a collection of test images that you can use for the exercises. In this exercise we will only work with 2D data, 3D would improve the performance due larger neighborhoods but also take more time to process.

Data

Experimental data

scroll.tif A slice image from a neutron CT of a lead scroll. Noisy and many ring aritfacts.

wood.tif A slice image from a neutron CT of a piece of petrified wood. Sampling artifacts and ring artifacts.

asphalt_gray.tif A slice image from an X-ray CT of an asphalt sample.

Phantom data

testpattern.tif A test image with geometric structures. Intended for degraded phantom experiments.

asphalt_bilevel.tif A binarized slice image from an X-ray CT of an asphalt sample. Intended for phantom experiments.

1 Images and Noise

In this exercise we are measuring and generating image noise. The resulting images can be used in later exercises.

Useful matlab functions

Use the help command for more information about the syntax for each function

imread Load an image to the matlab workspace.

imagesc Displays an image with intensities scaled between min and max value.

subplot Divides the figure into several sub-panels.

linkaxes A convenient way to zoom and pan in all subplots simultaneously.

hist Computes a histogram of a variable. Note that the (:) notation is needed.

mean Computes the mean value of a variable. Note that the value is only computed for one direction, i.e. mean(a) where a is an image will result in a vector of mean values. Use mean(a(:)) to compute a single mean value for the image.

std Computes the standard deviation of the variable. Note that the (:) notation is needed.

var Computes the variance of the variable. Note that the (:) notation is needed.

rand Generates a random field with uniform distribution.

randn Generates a random field with Gaussian distribution.

1.1 Measure signal to noise ratio

1. Load the experiment images using imread. Often it is needed to cast the data type of the image to double, e.g. `a=double(imread('file.png'))`
2. Display the images in sub-panels of a figure. `imagesc` displays the images stretched to fit the figure window, use the command axis image to display isotropic pixels. It is useful to annotate the panels using `title`

3. Identify different interesting regions and measure the signal to noise ratio. Regions are chosen by extracting sub-images from the original image, e.g. `b=a(20:100,50:70)` selects a sub image in the region rows 20-100 and columns 50-70. Note that `imagesc` displays the image in x and y coordinates i.e. transposed.

1.2 Generate noise

1. Load the image 'testpattern.png'.
2. Generate images with Gaussian noise and different variances.
3. Add Gaussian noise (SNR=1,2, 5, 10, and 100) to the phantom image and display them on sub-panels using `linkaxes` to connect the zoom and pan. This type of images will be used the following exercises.

2 Convolution

Useful matlab commands

filter2 filters an image with a convolution kernel

medfilt2 filters an image using a median filter

fft Computes the Fourier transform of an image

ifft Computes the inverse Fourier transform of an image

2.1 Real space

Filter noisy phantom images with SNR 1, 2, 5, 10, and 100 and show them in subplots with the noise free phantom as well as noisy input and difference between phantom and filtered. Create a new figure window per filter.

1. Box filter 3x3, 5x5, and 7x7.
2. Median filter 3x3, 5x5, and 7x7.
3. *Extra:* Gauss filter with σ=1, 2, 3. Implement this filter with separate kernels for x and y.

Observe filter performance near edges and how well the noise is suppressed. By linking all subpanels you can benefit from the simultaneous zoom and pan.

3 Scale spaces

Useful matlab functions

wavemenu A GUI for testing different aspects of wavelet processing. In this exercise we will use *SWT denoising 2D*

dwt2 one step of wavelet decomposition.

wavedec2 multi level wavelet decomposition.

detcoef2 computes statistics of the wavelet components created by wavedec2.

wbmpen estimates threshold levels for denoising.

wdencmp denoises or compresses an image.

3.1 Wavelets

1. Test different number of levels, wavelet families using wavemenu. The input images must be scaled to the interval 0-255 for the tool to work well. Which settings are suited for the test images.

2. Implement a matlab script that uses the same settings.

3.2 Diffusion filters

The correct parameterization is important for the success when using PDE filters.

1. Run the two scripts nldemo1 and nldemo2 in the diffusion folder.

2. Try to find good filter parameters for the example images using the function `nldif`. Important parameters to tune are:

   sigma The strength of gradient smoothing, larger value needed for low SNR.

   lambda Gradient threshold that tunes the sensitivity to low contrast structures. Must be set lower when sigma increase.

   stepsize and steps relaxation time of the solution. The greater the product of the stepsize and step the more smoothing is to be expected. The step size is usually decreased for improved quality of the solution especially for low SNR data.

4 Performance evaluation

Useful matlab functions

loglog, semilogx, semilogy Plots data with either logarithmic scale or a mix of linear and logarithmic axes.

mse Computes the mean squared error between two variables (any dimension).

ssim Computes the mean structural similarity index between two images (2D).

for loops Controlled loops over a predefined set of values.

tic, toc Measures the execution time between tic and toc.

Design a performance evaluation experiment using phantom images degraded by different noise levels. In this exercise the results from the previous exercises are summarized and provides an overview of how different filters perform. Measure the error and likeliness using mse and ssim (matlab scripts provided with the exercises). Also measure the execution time for each test. The execution time may not be so important for a single slice but consider the case when you have several Gvoxel images to process. Then you have to decide which method you can use in the given time.

Test conditions:

- Use SNR=100, 50, 20, 10, 5, 2, and 1
- 10 test runs per noise level
- Select 2-3 filter methods

Present the results with linear or logarithmic plot if that makes sense.