RRY025 -- Projects

Version 14th Sept. 2023

Rules

The project is a mandatory part of the course and part of the examination. The project can give up to 10 points (25%) of the course's total point amount.

The projects should be done in small groups. Ideal is some 3-4 students. The choice of the project is free for the group.

The group prepares one written report about their project. The written report will be assessed. There is no strict template for the report. Typically, some roughly 10 page document is needed to describe the work. If including extensive code is necessary, it can be included in an additional appendix.

The report should contain an introduction of the problem, relevant theoretical background, a description of how it was approached and worked on (and solved?) and a conclusion that draws together the most important aspects of the work. The report can be written in English or Swedish (English is recommended. The choice of language does not affect the grading).

Each group member should also write a roughly half-a-page reflection about their own role and contribution in the group work. The reflections will not be assessed, nor do they affect the grade of the project, but they are mandatory.

Email the examiner the composition of your group and the choice of your project **at latest Thursday 21st September**.

The deadline for the written reports is **20th October**. Note that the last organized exercise/consultation session is 16th October.

List of Projects

The descriptions below give the basics of each project. Help and hints will be available during the "individual consultation" sessions in late September/Early October. The goal is that before getting more hands-on help (if needed), the group tries to make progress and solve possible problems by itself.

1. Wavelet compression of images

Wavelet and wavelet-packet image compression promises state of the art image compression ratios (e.g. 100:1) and is the basis of the next generation of compression standards (e.g. JPEG2000). Critically important in achieving the highest possible compression ratios is to use the most suitable wavelet basis functions. In this project you should **investigate how the compression ratio and image quality depends on the choice of wavelet basis functions**.

Hints: Use the image of two superimposed galaxies: galax2.mat (in the course's Matlab files) as your starting point. Type 'wavemenu' to get started. The Matlab Wavelet Toolbox will be introduced during the course. Extensive documentation about the toolbox can be found at

http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/

2. Comparison of wavelet and discrete cosine transform compression

Wavelet image compression promises state of the art image compression ratios (e.g. 100:1) and is the basis of the next generation of compression standards (e.g. JPEG2000). In this project, **compare the compression ratios and root mean square errors of different types of grayscale images using Cosine and Wavelet based methods**. For cosine compression choose a blocksize of 8, and set to zero all but say the largest 8 cosine coefficients in each block to get 8:1 compression. Compare with wavelet compression, for a similar compression ratio, using a biorthogonal wavelet basis. At what compression ratio is the image quality from wavelets the same as the 8:1 compressed cosine image?

Hints: Block processing can be achieved via the matlab command **blkproc** 3. Lossy waveform based image compression

Take a grayscale image or set of images, and **examine the properties of lossy waveform based compression**. Use a simple predictor and quantizer, then use a Digital Pulse Code Modulation. What degree of compression is possible? Are the results visually acceptable?

4. Homomorphic filtering

Many photographs are affected by strong variations in illumination. One way to enhance such images, so that details are visible in both the bright and dark areas, is via homomorphic filtering. In this processing technique we first take the logarithm of the image before filtering. The method is described in chapter 4.9.6 of the textbook (3rd Ed., 2008). **Apply homomorphic filtering technique** to the image forest.mat (in Course's Matlab files).

Use *load('forest.mat')* to load the the *forestgray* image. Attempt to choose either interactively or automatically the best parameters for the filter function to give the best possible enhancement of the image.

5. Local image enhancement

Many photographs are affected by strong variations in brightness. Standard methods of global histogram equalisation do a poor job of enhancing such images. One way to improve such images so that details are visible in both the bright and dark areas is via local enhancement, as described in chapters 3.3.3 and 3.3.4 of the textbook (3rd Ed., 2008). Both **local histogram equalisation and simpler methods using local mean and variance** can be used to advantage to improve such images. **Implement these image enhancement algorithms and apply them** to the image forest.mat (in Course's Matlab files).

Use *load('forest.mat')* to load the the *forestgray* image. Attempt to choose either interactively or automatically the best parameters for the local histogram box or for the transfer function based on the local mean and variance.

6. Movie restoration via scratch removal

A recent active area of image processing is the restoration of old movie films. Many of these old films are deteriorating and there are active efforts to convert them into digital form and remove distortions.

One major problem affecting old movies is the presence of scratches. **Devise a simple algorithm to identify scratches and 'repair' such scratches by replacing affected pixels by**

adjacent ones. In the basic project choose one colour and apply your scratch removal algorithm to the individual frames separately. An example of a movie affected by scratches can be found in the file film1_big.jpg (in Course's Matlab files).

In this figure the left column shows the original frames. The middle column shows the frames after colour correction. The final column shows the frames after applying a scratch removal algorithm. Load this image into matlab using *imread*, then extract a frame from the *middle column*. This will have three colour layers (red, green, blue). Apply initially your algorithm to one frame of one of the colour layers. Use the multi-colour and multi-frame data set to aid the scratch removal algorithm.

7. Restoring a rotationally blurred image

An image is taken from an orbiting satellite which is rapidly spinning on its axis. The result is that even during the shortest exposure time the satellite rotates by 10 degrees and the images are **rotationally blurred**. **Design and implement an algorithm to restore the image** and save the mission. Hints: Create a suitable rotationally blurred image via using *imrotate*, selecting the highest possible quality interpolation (bicubic). This project is harder than it looks (in fact, it is almost non-solvable perfectly). The goal is to learn by figuring out why easy solutions do *not* work. The grading is not dependent on reaching a good result (but rather on the discussion on what was done/tried/learned during the process).

8. Restoring an image affected by constant speed camera motion using pseudo inverse and Wiener filter

A distant object is photographed through a telephoto lens from a moving car, such that during the exposure time the image is convolved by a space-invariant horizontal boxcar of several pixels. In addition a small amount of Gaussian noise is added. **Implement and test Pseudo - Inverse and Wiener filter restoration**. Which method produces the best results?

Hints: For Wiener filtering assume an image power spectral density function typical for the class of images being restored, or assume it to be power law with some inner cutoff and white noise.

9. Build a "portrait" mode image enhancement tool

A typical "portrait" mode in modern cameras identifies a face and creates a depth effect ("bokeh") using image enhancement techniques. **Build a "portrait" mode tool** that 1) identifies an object (head/face) using simple edge detection; 2) smooths the background of the photo to emphasize object; 3) possibly manipulates (histogram equalizes/sharpens) the object for further enhancement.

Hints: Steps 1 and 2 can be done using multiple ways, including simple averaging tools, Fourier domain, or Wavelets.

Consider what kind of photos are "easy" to use for this purpose. Try to build a tool that works for an "easy" image. (it is *really* hard for *any* image with a face on it)

Note: This project is aimed to be exploratory. It is quite hard to build a well-functioning portrait mode (as you perhaps can imagine). The point of this project is not to get to an excellent result, but to learn while doing it.