HERIOT-WATT UNIVERSITY

DEPARTMENT OF COMPUTING AND ELECTRICAL ENGINEERING

B39SD2 – Matlab tutorial 3
Fourier Analysis and manipulations

Objectives:

We will investigate the notion of spectrum and simple filtering in the frequency domain.
As we will see, in all cases, we can interpret the effect of these filters in the frequency domain easily while their effect is not obvious in the time domain.

During this session, you will learn & practice:

1- More programming in matlab
2- More examples of functions and scripts
3- Fourier transform and its applications
4- Simple Filtering techniques
5- Phase information and its relation to image structure
6- Correlation and template matching in imaging

Ressources required:

In order to carry out this session, you will need to download images and matlab files from the following location:

http://www.ece.eps.hw.ac.uk/~ceeyrp/home/Pages/Teaching/B39SD2/B39SD2.html

Please download the following images:

- Image of Lena
- Landsat image
- Fractal image

We will also use the circuit and flowers images (circuit.tif, peppers.png).
You are of course encouraged to try these programs out on images of your choice.

Please download the following functions and script:

- fft2d.m (same as last week)
- ifft2d.m (same as last week).
- phase_only.m
- random_phase.m
- random_magnitude.m
- quant_fft.m
- move_image.m
- SimpleFiltering.m
- StandardCorrelation.m
- TemplateMatching.m

Phase manipulation:

Download the fft2d program, the ifft2d program and the phase-only programs.

As you have already seen in class, the phase of the spectrum is tightly related to the structure of the image. This will be illustrated in the following.

Please load an image (for example circuit.tif).
Display the image in a figure;
Now use the phase_only function (help phase_only).
Comments?

Modify the phase_only function program to write a program which replaces the phase of the spectrum by a random phase. Hints: use the size function to get the size of the image and the rand function to generate the random phase.

Do the same with the amplitude of the spectrum. Comments?

Try it with various images

FFT Manipulations:

As we now know, the FFT of an image (generally real) is a complex number. We can extract the phase and the magnitude of the spectrum. The numbers representing them normally have a finite precision on computers. Does that affects the results and how?
Compression algorithms tend to quantify the spectrum or some features related to the spectrum (Discrete Cosine Transform).
How does quantification affects the image?
It the phase or the magnitude more sensitive to the quantification?
You can explore this using the quant_fft program. Load an image (for instance circuit.tif) and apply the algorithm to this image varying the parameters of the function. For example, play with the number of levels on which you quantify the magnitude and the phase of the spectrum. Comments?

Displace an image using the FFT:

As we have seen in class, a linear phase shift in the Fourier domain corresponds to a translation in the space domain. This could be used to translate an image in the space domain using simple multiplication in the Fourier domain. Try to write a program that does that.
Now download the program called move_image.m and see how you can use it.
Comments?

Simple Fourier Filtering:

Again as seen in class, the high frequencies of the spectrum corresponds to the edges while the low frequencies corresponds to the 'filling'. An easy way to demonstrate that is to take the Fourier transform of an image, cut the high (resp low) frequencies and Fourier transform back the result. Try it using the SimpleFiltering program.

For those of you who have done filtering before, what are the issues involved with this type of filtering? Is the fact the we are using only N samples to calculate the Fourier transform an issue?
You will have seen in class the convolution theorem. Can we use it to perform filtering efficiently? Try to write a program that does this…

Template Matching

Template matching is a technique used to find a specific shape (template) in a scene (image). The idea is to compare a small part of the scene to the template using a similarity metric to define a score part of the scene under consideration and the template. A high score is associated to a high probability that the template is the scene at this particular location.

Correlation is a very standard way to perform template matching. It is very fast when implemented using FFTs and will be demonstrated during this lab on a simple example.

You will need to download the following files from the web:

- StandardCorrelation.m
- TemplateMatching.m
- Letters.bmp (Our scene)
- E.bmp, H.bmp (Our templates)

First analyse the code to understand what operations are performed here. Then run it on our example image. Now add noise to the image and try again (to add noise use the randn function). Comments?
Do you think this technique is:

- Rotation invariant?
- Scale Invariant?
- Can cope with shape deformations?
- Robust to noise

Compression and DCT:

Launch the matlab demo (type demo) and go to the image processing toolbox demo.
Play with the DCT demo and try to understand the effect of the parameters and coefficients.
Try it on various images with different spectral content.