School of Engineering and Physical Sciences
Electrical Electronic and Computer Engineering

Image Processing

Matlab tutorial 3

Fourier Analysis and Manipulations

1. 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

2. Resources required:

In order to carry out this session, you will need to download images and matlab files from the Image Processing module on vision.

`http://vision.hw.ac.uk`

Please download the following images:

- Image of Lena
- Landsat image
- sonar image
- sar 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`

3. **Phase manipulation:**

Download the `fft2d` program, the `ifft2d` program and the `phase_only` programs.

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?

The function `random_phase` calculates the FFT of an image, and leaves the magnitude unchanged but replaces the phase with random values. The inverse Fourier transform is then performed to re-create the image. Use this function to investigate the effect of replacing the phase with random values.
For example, try:

```
ima = imread('lena.tif');
figure(1);
colormap(gray);
imagesc(ima);
figure(2);
colormap(gray);
ima_out=random_phase(ima);
```

What happens if you repeat the above, but use the images `sonar.tif` or `sar.tif`? What can you conclude about these images?

The function `random_magnitude` calculates the FFT of an image, and leaves the phase unchanged but replaces the magnitude with random values. The inverse Fourier transform is then performed to re-create the image. Use this function to investigate the effect of replacing the magnitude with random values

For example, try:

```
ima = imread('lena.tif');
figure(1);
colormap(gray);
imagesc(ima);
figure(2);
colormap(gray);
ima_out=random_magnitude(ima);
```

What can you conclude from the above experiments?

What happens if you repeat the above, but use the images `sonar.tif` or `sar.tif`? What can you conclude about these images?

4. 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. We will now investigate whether this affects the results and how.
Compression algorithms tend to quantify the spectrum or some features related to the spectrum (for example Discrete Cosine Transform for JPEG).

Therefore, we will investigate how quantification affects the image.
Is 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?

4.1 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.
Now download the program called `move_image.m` and see how you can use it.
Comments?

4.2 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?

5. 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.