ÉCOLE CENTRALELYON

Image processing - frequency domain

BE 2h - MATLAB

A. Image filtering

Given an image $f(x,y)$ defined for $x = 0,1,\ldots,M-1$ and $y = 0,1,\ldots,N-1$, the 2-D discrete Fourier transform (DFT) of $f(x,y)$ is denoted by $F(u,v)$ and is given by:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi ux/M}e^{-j2\pi vy/N}$$

and is defined for $u = 0,1,\ldots,M-1$ and $v = 0,1,\ldots,N-1$.

Similarly, the inverse discrete Fourier transform is given by:

$$f(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)e^{j2\pi ux/M}e^{j2\pi vy/N}$$

again for $x = 0,1,\ldots,M-1$ and $y = 0,1,\ldots,N$. The Matlab routines for computing the 2-D DFT and the inverse 2-D DFT are the routines fft2 and ifft2.

Exercise 1 – 2-D Fourier Transform

Using the image files Moon.tif, Fig_woman.tif, and Fig_xray.tif, read in each of the images and compute the 2-D DFT magnitude for each of the images.

A.1.1. On a single plot (using a grid of 2 x 2 images), plot the following:

- the original gray scale image in the upper left cell
- the image representation of the 2-D DFT magnitude of the image being studied in the upper right cell
- a clipped and scaled version of the 2-D DFT magnitude that is clipped and scale at a level where you can see the nature of the 2-D DFT magnitude for each image; this plot should be placed in the lower left cell
- the log transformed 2-D DFT magnitude plotted again on a scale that enables you to see the structure of the transform magnitude; this plot should be placed in the lower right cell

You will need to use the Matlab routine FS=fftshift(F) to shift the DC magnitude point from the upper left corner of the image to the center of the magnitude range.

In your report, include the Matlab code along with image plots of the three images for which you repeat the analysis.

A.1.2. Can you say how the image properties are seen in the 2-D DFT magnitude plots?

Exercise 2 – Importance of DFT Phase

The phase angle of the Fourier transform of an image contains a great deal of information about the image. To see this, we will next perform a simple set of experiments. First we define the

quantities of interest precisely.

If we denote the gray scale (sampled) image as $f[m, n]$ with 2-D DFT, $F[i, k]$. then we can define the "magnitude only" image corresponding to $f[m, n]$ as the image obtained by taking the 2-D inverse DFT of the image magnitude function, i.e.,

$$f\text{mag}[m,n] \Leftrightarrow |F[i,k]|$$

i.e., $f_{mag}[m,n]$ is the inverse 2-D DFT of$|F[i,k]|$. We can also define the "phase only" image corresponding to $f\ [m,n]$ as the image obtained by taking the 2-D inverse DFT of the image phase function, i.e.,

$$f_{ph}[m,n] \Leftrightarrow e^{j\ \arg\{F[i,k]\}}$$

where $\arg\{F[i,k]\}$ is the phase angle of $F\ [i,k]$.

A.2.1. Write a Matlab program to compute the magnitude only and phase only versions of the two gray scale images, Fig_woman.tif and Fig_Barre500.tif, and plot the results in a 2 x 2 grid with the following format:

- the upper left and lower left images should be the original gray scale image files (so that you can compare them to the transformed images directly).
- the upper right image should be the image reconstructed from magnitude only infor- mation
- the lower right image should be the image reconstructed from phase only information

Include your Matlab code in the report, along with the two requested plots (one for each set of images).

A.2.2. Perform the following experiment. Compute the 2-D DFTs of the two images used in part 1 of this exercise. From these transforms, form two new DFTs in which the magnitudes and phases are interchanged. Compute the corresponding two images and display them, along with the original gray scale images, in a 2 x 2 grid of the form:

- the upper left image should be the first original image
- the upper right image should be the image reconstructed with the cross magnitude and the correct image phase signal.
- the lower left image should be the second original image
- the lower right image should be the image reconstructed with the cross magnitude and the correct image phase signal.

Include your Matlab code in the report, along with the requested plot.

A.2.3. On the basis of this exercise (as well as the results from Exercise 1) what can you say about the relative importance of magnitude and phase in preserving image properties. Why do you think the phase contributes so much more than the magnitude to preserving some of the original image properties?

Exercise 3 – Linear Filtering of Images

Using the included Matlab routine H=lpfilter(type, M, N, D0, n) for designing lowpass filters, where the input arguments are:

- type='gaussian', 'ideal' or 'btw' (for Butterworth filters) M,N = dimensionality of filter frequency response
- D0 = filter cutoff frequency (normalized to range [0, M] or [0 N])
- n = filter order for Butterworth filters

and the output H is the frequency response of the filter, write a Matlab m-file that accepts as input the filter type (ftype='gaussian','ideal','btw'), the cutoff frequency D0, and (optionally for the Butterworth filter) the filter order n. Within the Matlab program plot the filter frequency response (using the mesh command) along with the filter impulse response. Test your program for the following conditions:

I. ftype='gaussian', D0=50

II. ftype='ideal', D0=50

III. ftype='btw', n=4, D0=50

Include the graphics for the three frequency and impulse responses in your report.

A.3.1. How would you convert your Matlab code to change the design from a lowpass filter to a highpass filter? Demonstrate this capability by designing a Gaussian highpass filter with cutoff D0=50 and include the plot of the frequency and impulse response of the resulting filter.

A.3.2. Using the original input gray-scale image, Fig_test_pattern.tif, write another Matlab m-file which reads in the original image (what is the image size), then design an appropriate lowpass filter with appropriate padding, filter the image in the frequency domain, convert back to the image domain and plot the processed image (remember to crop the filtered image so that you have the same region of support as the original image).

A.3.3. Compare the processed images with the three lowpass filters used in part (a) of this Exercise. Also compare the effects of differing values of D0 between 20 and 200. What is the effect of lowpass filtering on the image?

A.3.4. Include designs for highpass filters in your Matlab code. Repeat part (c) using highpass filters in place of the lowpass filters. What is the effect of highpass filtering? What is the major difference between the three filter types?

B. Image compression

The objective of this session is to apply the DCT transformation to compress images. Then, to proceed to the decompression, by applying the inverse transformation. And finally, to calculate the error between the starting image and the images after decompression for quantization factors ranging from 1 to 25. For each quantization factor, the compression performed will be calculated by noting the number of different DCT coefficients from zero.

The stages of compression are as follows:

- Reading a bitmap image
- Recovery of the matrix of RGB coefficients
- Calculation of the DCT coefficients of the matrix by blocks of 8x8 pixels
- Quantize

- Writing DCT coefficients to a file

For decompression, simply reverse the previous steps.

```
function [duree_traitement] = main_compression(qualite, nom_source_bmp, nom_destination_jpg)
        o function [imageRGBdouble] = lecture_image_bmp(nom_fichier)
        o function [M] = conversion_spatial_frequantiel(matrice, qualite)
                ▪ function [res] = ajoute(m,valeur)
                ▪ function [Q] = mat_quant(Fq)
        o function [] = ecriture_jpg(JPG, nom, qualite)

function [duree_traitement] = main_decompression(nom_destination_jpg , nom_source_bmp)
        o function [JPG,qualite] = lecture_jpg(nom)
        o function [M] = conversion_frequentiel_spatial(matrice, qualite)
                ▪ function [res] = ajoute(m,valeur)
                ▪ function [Q] = mat_quant(Fq)
                ▪ function [a] = m_IDCT2(m)
        o function [] = ecriture_bmp(BMP, nom)
```

B.1. From the lecture and also the code review, explain how image compression works.

B.2. Calculate the difference between a bitmap image before compression and after decompression for quantization factors ranging from 1 to 25.

B.3. For each quantization factor, calculate the number of non-zero DCT coefficients.

B.4. Plot the error curves (mean and standard), as well as the compression rates, as a function of the quantization factor.

Compress your report as well as your matlab code. Drop the compressed file on the dedicated moodle link