ECE7410/ENGI9804 Image Processing and Applications

Laboratory 3

Frequency domain filtering

Introduction

The objective of this laboratory exercise is to get familiarized with frequency domain filtering concepts.

Frequency Domain Procedure Review

The frequency domain image filtering process can be performed as follows:

1. Given the input image $f(x, y)$ of size $M \times N$, obtain the padding sizes $P = 2M$ and $Q = 2N$.
2. Form a padded image of size $P \times Q$ using zero padding.
3. Multiply $f(x, y)$ by $(-1)^{x+y}$ to centre the Fourier transform (FT) on the $P \times Q$ frequency rectangle.
4. Compute the Discrete Fourier Transform (DFT), $F(u, v)$ of the centred image
5. Construct a real, symmetric filter transfer function, $H(u, v)$, of size $P \times Q$ with centre at $(P/2, Q/2)$.
6. Form the product $G(u, v) = H(u, v)F(u, v)$ using element-wise multiplication.
7. Obtain the Inverse Fourier transform of $G(u, v)$, ignoring parasitic complex components: $g_p(x, y) = (real[\mathfrak{F}^{-1}G(u, v)])(-1^{x+y})$
8. Obtain the final result, $g(x, y)$, of the same size as the input image by extracting the $M \times N$ region from the top left quadrant of $g_p(x, y)$.

Note that centering the transform helps to visualize the filtering process and to generate the filter functions, but centering is not a fundamental requirement.

Low pass and high pass filter implementation

1. Download the test image (puppy.jpg) and *lp_hp_filters.m* from Brightspace under *Lab 03*. Look at the image, and read the code and understand the implementation.
2. [2] Read the image, convert the image to grayscale, and display.

```
F_rgb=imread('puppy.jpg'); % read the gs image
F=rgb2gray(F_rgb);
```

3. [4] Obtain the padding parameters. Display them.

```
im_size=size(F); % Obtain the size of the image
P=2*im_size(1);Q=2*im_size(2); % Optaining padding parameters as 2*image size
```

4. [4] Obtain and display the FT of the image.

```
FTIm=fft2(double(F),P,Q); % FT with padded size
```

5. Design the filter

```
Filter = lp_hp_filters('ideal','lp', P, Q, D0,n); %Specify the filter
%The function calculates the LP or HP filters based on the parameters given. %
    Filter_type=('ideal', 'btw', or 'gaussian')
%lp_or_hp=('lp' for low pass, 'hp' for high pass)
%P, Q are the padded image size
%D0 is the cutoff frequency (e.g., D0 = 0.1*im_size(1) for 0.1 times image
    height), and
%n is the order of a Butterworth filter (e.g., n=0)
```

6. Implement the following filters ((a) to ((f))): multiply the FT of the image with filter, undo the padding, move the origin of frequency spectrum to the centre and display the results.

```
Filtered_image=real(ifft2(Filter.*FTIm)); % multiply the FT of image by the
    filter and apply the IDFT
Filtered_image=Filtered_image(1:im_size(1), 1:im_size(2)); % Resize the image (
    undo padding)
Fim=fftshift(FTIm); % move the origin of the FT to the center
FTI=log(1+abs(Fim)); % compute the magnitude  (log to brighten display)
Ff=fftshift(Filter); % move the origin of the FT to the center
FTF=log(1+abs(Ff)); % compute the magnitude  (log to brighten display)

subplot(2,2,1), imshow(F,[]), title('Original Image'); % show the image
subplot(2,2,2), imshow(FTI,[]), title('FT of Original'); % show the image
subplot(2,2,3), imshow(FTF,[]), title('Filter in frequency domain'); % filter
subplot(2,2,4), imshow(Filtered_image,[]), title('Filtered Image'); % show the
    image
```

   (a) $[3 \times 2 = 6]$ *Ideal* low pass filters for cut-off frequencies (0.3 and 0.7 of image height).
   (b) $[3 \times 6 = 18]$ *Butterworth* low pass filters for cut-off frequencies (0.1 and 0.5 of image height) and order $n = 1, 5, 20$.
   (c) $[3 \times 3 = 9]$ *Gaussian* low pass filters for cut-off frequencies (0.1, 0.3 and 0.7 of image height).
   (d) $[3 \times 3 = 9]$ *Ideal* high-pass filters for cut-off frequencies (0.1, 0.3 and 0.7 of image height).
   (e) $[3 \times 6 = 18]$ *Butterworth* high pass filters for cut-off frequencies (0.1 and 0.5 of image height) and order $n = 1, 5, 20$.
   (f) $[3 \times 3 = 9]$ *Gaussian* high pass filters for cut-off frequencies (0.1, 0.3 and 0.7 of image height).

7. [5] What can you observe when increasing the cut off frequency radius of the *Ideal* filter?

8. [5] What can you observe when increasing the order of the *Butterworth* filter when the cut-off frequency remains the same?

9. [10] Discuss the *general* performance of the *Gaussian* filters in comparison to the *Ideal* and *Butterworth* filters. You do not need to discuss each possible comparison case. Only discuss any interesting phenomenon or performance observations.

Important notes [20 points]

1. [10] You ***MUST*** include all of your code (such that it can be easily run to verify your results), results and discussion in your lab report, including also any images that are not included with the lab.

2. [10] You ***MUST*** name your lab report G#.pdf using your group number (e.g., G03.pdf), and produce a high-quality report.

3. Submit your work to Brightspace in the appropriate dropbox folder.