Department of Computer Science
Digital Image Processing

N. Komodakis, Course instructor

Autumn 2022

Playing with Images in the Frequency Domain

The goal of this assignment is to get you familiarized with the use of Fourier Transforms on images, including Discrete Space Fourier Transform (DSFT), and, the most commonly used Discrete Fourier Transform (DFT).

1 Discrete Space Fourier Transform & Image Segmentation

1.1 Discrete Space Fourier Transform

Your are given two one-dimensional filters $h(m)$ and $g(m)$ with values as follows:

m	-5	-4	-3	-2	-1	0	1	2	3	4	5
$h(m)$	0.009	0.027	0.065	0.122	0.177	0.2	0.177	0.122	0.065	0.027	0.009
$g(m)$	0.013	0.028	0.048	0.056	0.039	0	0.039	0.056	0.048	0.028	0.013

Plot the magnitude of the Fourier Transform of the two 1D-filters and the 2D-filter $h(m)g(n)$, where h with subscript m refers to rows and $g(n)$ with subscript n refers to columns. Implement and utilize the Discrete Space Fourier Transform *(DSFT)* to plot the magnitude in the interval $[-\frac{1}{2}, \frac{1}{2}]$ for both spatial frequencies. In your computation, assume that the values outside the filter boundaries are set to 0.

1.2 Image Segmentation

A building facade image I is given in the file:
http://www.csd.uoc.gr/~hy371/images/build_neoclassic.png

Compute the filter response $y_1(m.n)$ of image I by utilizing filter $h_1 = h(m)g(n)$ and filter response $y_2(m.n)$ of image I by utilizing filter $h_2 = g(m)h(n)$. Compute each filter response by implementing image filtering in the spatial domain in the rows and the columns separately.
Calculate the square magnitude:

$$A = y_1^2(m.n) + y_2^2(m.n)$$

and the unsigned direction of the vector $(y_1(m.n), y_2(m.n))$:

$$\theta(m,n) = \left| arctan\frac{y_2(m.n)}{y_1(m.n)} \right|, \theta(m,n) \in \left[0, \frac{\pi}{2}\right]$$

We cluster the image pixels into five groups $L = \{0, 1, 2, 3, 4\}$. The first group includes all pixels that satisfy the relation $A(m, n) \leq \mu$, where μ is the mean value of A. The rest of the image pixels (which satisfy $A(m, n) > \mu$) will be classified into four groups based on the direction θ. For the grouping you will utilize the k-means clustering algorithm and you will use the criterion of absolute value deviation. The four cluster centers that will be computed by the k-means algorithm are denoted by $\Theta_k (k = 1, 2, 3, 4)$ (their initial values that are given as input to the k-means algorithm will be: $[0, 0.15\pi, 0.35\pi, 0.5\pi]$). Based on the resulting clustering, we obtain an image L as follows:

1. If $A(m, n) \leq \mu$, then $L(m, n) = 0$.
2. If $A(m, n) > \mu$ and $\theta(m, n) \leq 0.5(\Theta_1 + \Theta_2)$, then $L(m, n) = 1$.
3. If $A(m, n) > \mu$ and $\theta(m, n) \geq 0.5(\Theta_3 + \Theta_4)$, then $L(m, n) = 2$.
4. If $A(m, n) > \mu$ and $\theta(m, n) > 0.5(\Theta_1 + \Theta_2)$ and $\theta(m, n) \leq 0.5(\Theta_2 + \Theta_3)$, then $L(m, n) = 3$.
5. If $A(m, n) > \mu$ and $\theta(m, n) > 0.5(\Theta_2 + \Theta_3)$ and $\theta(m, n) < 0.5(\Theta_3 + \Theta_4)$, then $L(m, n) = 4$.

Plot the resulting image L where group labels are color coded:
0:'black', 1:'yellow', 2:'red', 3:'green', 4:'Blue'.

Useful functions: *abs, meshgrid, surf, imfilter, atan, kmeans, label2rgb*

2 Discrete Fourier Transform & Image Compression

For this part of the assignment you are required to apply the Discrete Fourier Transform (DFT) in human face images, in order to investigate (a) the role of the phase in the frequency space, and, (b) the possibility of compressing the image under this transformation space. As a first step, you have to compute the 2-D DFT of the image using the built-in Matlab function *fft2*, which can be expressed in the polar coordinate system using the built-in Matlab functions of *abs* and *angle*.

2.1 The role of the phase and magnitude

Part 1: You are requested to apply the inverse DFT, using the built-in Matlab function *ifft2(.,'symmetric')*, by maintaining the phase as is and only altering the magnitude using the following formula:

$$A(u, v) = \frac{1}{1 - \alpha(cos(2\pi u) + cos(2\pi v))}$$

The (u, v) variables take values in the range $[0, 1)$ as follows:

$$(u, v) = (\frac{m}{M}, \frac{n}{N}), m = 0, ..., M - 1 \text{ and } n = 0, ..., N - 1$$

where (M, N) denote the image dimensions.

You will consider three different cases for the magnitude A, corresponding to the values of $\alpha = 0.45, 0.49, 0.495$.

The image Y_i that results from applying the inverse DFT for the magnitude case i will then be linearly transformed based on the following formula:

$$Z_i = max(X)\frac{Y_i - min(Y_i)}{max(Y_i) - min(Y_i)} + min(X)$$

where X denotes the original image.

For each transformed image Z_i you will then adjust its pixel values so that its histogram matches the one of the original image (for that you will use the built-in Matlab function *histeq* and the histogram counts of the original image). Finally, compute the mean absolute error between the resulting image and the original one.

Part 2: Uniformly quantize the phase for a set of intervals $K_\phi \in \{5, 9, 17, 33, 65\}$. For each quantized phase interval configuration perform an inverse DFT using (a) the magnitude A with $\alpha = 0.495$, and, (b) the magnitude of the original image. Compute the mean absolute error between all transformed images and the original one.

2.2 Image Compression

For this part of the assignment you will be investigating the effect of retaining only a subset of the spatial frequencies on the compression and reconstruction of an image. More specifically, you will retain only p% of the spatial frequencies, choosing the ones with the largest magnitude, and setting the rest of the spatial frequencies to zero. You will be examining three scenarios for values of $p = 2.5, 5, 7.5$. The images that result from applying inverse DFT will be compared to the original image by measuring the mean absolute error.

Image Dataset: apply the functions of the second part of the assignment on the following images:
http://www.csd.uoc.gr/~hy371/images/einstein.png
http://www.csd.uoc.gr/~hy371/images/obama.png
http://www.csd.uoc.gr/~hy371/images/barbara.png

Remark: To display the logarithm of the magnitude or the phase of the Fourier transform, we recommend that you use the built-in Matlab function *fftshift*, that rearranges a Fourier transform of the image by shifting the zero-frequency component to the center of the array.

Useful functions: *fft2, abs, angle, ifft2, fftshift, colormap, imhist, histeq, imquantize, multithresh, mean2, min, im2double*