Computational Photography
Project 3, Fall 2014

October 16, 2014

This project is about linear filtering and the discrete Fourier transform. The Matlab excercise needs to be turned in by uploading on ILIAS. The pen and paper exercises need to be turned in during the exercise session. The deadline is October 30, 14:00.

Pen and Paper Exercises

1. **(1 point)** Determine which of these filters is (a) linear and (b) shift invariant. Justify your answers.
   (a) $\mathcal{F}(f(x)) = e^{f(x)}$
   (b) $\mathcal{F}(f(x)) = f(x)f(x-1)$
   (c) $\mathcal{F}(f(x)) = \sum_{k=x-4}^{x+2} f(k)$
   (d) $\mathcal{F}(f(x)) = f(2x)$
   (e) $\mathcal{F}(f(x)) = [\sin(2x)]f(x)$
   (f) $\mathcal{F}(f(x)) = xf(x)$
   (g) $\mathcal{F}(f(x)) = f(x) - f(x-5)$
   (h) $\mathcal{F}(f(x)) = f(x-3) - 2f(x-12)$

2. **(1 point)** Write pseudo code for a moving average (box) filter of size $n \times n$. You can assume that n is odd. Boundary issues can be ignored. Give the complexity of your algorithm in terms of the filter size n and the number of image pixels m.

3. **(1 point)** What is the spatial filter and its frequency transform that corresponds to applying the box filter twice? Explain your answer.

   Explain the output that you would obtain by repeatedly applying the moving average filter an infinite number of times. Also explain this effect by considering the corresponding Fourier domain operation.

4. **(0.5 points)** Give a 3×3 mask for performing unsharp masking in a single pass. Assume that the blurring step is performed by a 3×3 box filter.

5. **(1 point)** The 2D continuous Fourier transform pair is given by the equations

$$F(\mu, \nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-j2\pi(\mu t + \nu z)} dt dz,$$

and

$$f(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t+\nu z)}d\mu d\nu.$$

Show that the Gaussian lowpass filter in the frequency domain,

$$H(\mu,\nu) = Ae^{-(\mu^2+\nu^2)/2\sigma^2},$$

corresponds to the spatial domain filter

$$h(t,z) = A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)}.$$

6. (**0.5 points)** Prove that the continuous 2D Fourier transform and its inverse (see above) are linear operations.

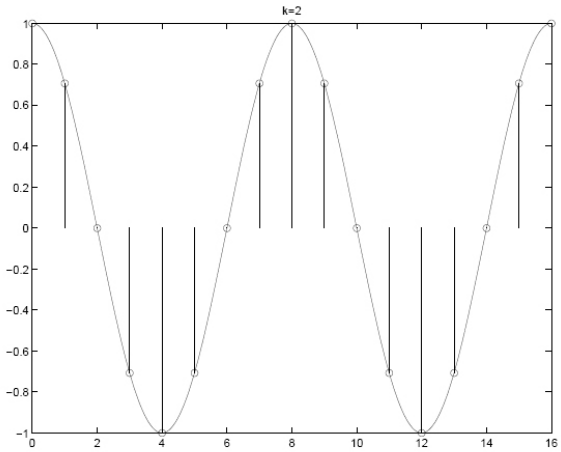
Matlab Exercises

Solve all assignments in a single Matlab script. Use cells to organize the script. Use the *Publish* function in the *File* menu to execute the script and save its outputs. Turn in the Matlab script and the published document in html format. For more information about publishing Matlab scripts (m files) look for "publishing M files" in the Matlab help. Note that the publishing functionality does not work correctly if the filename of your Matlab script contains special characters such as spaces!

For questions that need to be answered in words, include your text in the code using `disp` or by writing a comment.

1. (**0.5 point)** Sampling and aliasing.

   (a) Consider the function $f(x) = \cos(\omega_o x)$ with $\omega_o = 2\pi k/N$. Here, k is considered the *frequency* of the cosine wave, and ω_o is also called the *angular frequency*. Let $N = 16$ and consider the interval $x \in [0, 16]$. Let $f(n)$ denote $f(x)$ sampled at the integers $0, 1, \ldots, 16$. In the same graph, plot the continuous function $f(x)$ (using `fplot`) and the sampled version $f(n)$ (using `stem`) for a frequency $k = 1, \ldots, 16$. Use `hold` to plot both graphs in the same figure. Use `subplot` to arrange the plots for all k in a grid. As an illustration, here is the plot for $k = 2$.



(b) Indicate the value of ω_o and k at which $f(x)$ hits the Nyquist frequency. Note that at this point we can write $f(n) = (-1)^n$.

2. (**0.5 point)** 1D discrete Fourier Transform (DFT).

(a) Implement your own discrete Fourier transformation according to the formula from the class slides. For $n, m \in \{0 \dots M-1\}$ the DFT is given as

$$F_m = \sum_{n=0}^{M-1} f_n e^{-i2\pi \frac{mn}{M}},$$

where f_n are the samples of the input function, and F_m are the Fourier coefficients. Create an $M \times M$ matrix for the DFT containing the entries given by the formula. The resulting matrix has the structure of a *Vandermonde matrix*.

(b) A unitary matrix U has $\det(U) = 1$ and its complex conjugate is its inverse: $U^*U = UU^* = I$. By dividing the DFT matrix by $\sqrt{M}$ we get a unitary matrix. Show that your matrix has this property.

(c) Plot the real and the imaginary part of the rows (or columns) of your DFT matrix. Use `real` and `imag` to get the real an imaginary parts. What do you observe? **Hint:** Plot also $\sin(kx)$ and $\cos(kx)$ for different integers k for comparison.

(d) Take M samples at $\frac{2n\pi}{M}$, $n = 0, ..., M-1$ of $\cos(kx)$ and store them in a vector. Multiply this vector with your DFT matrix. Compare your result to what you get using `fft`, Matlab's fast Fourier transform. They should be the same. Apply `fftshift` to center the outputs. Give an intuitive explanation in words of the result you get.

(e) Read about the fast Fourier transformation (FFT) algorithm on Wikipedia. What is the asymptotic complexity of the naive DFT and the FFT algorithm? What is the speed-up factor that you get with the FFT compared to the naive DFT for an input vector with one million elements (i.e., about the size of a typical raster image)?

3. (**1 point)** 2D discrete Fourier Transform (DFT).

(a) Compute the discrete Fourier transform of a grey-scale image of your choice using `fft2`.

(b) Verify that element $(1, 1)$ of the Fourier transform corresponds to the sum over all pixel values.

(c) Verify Parseval's theorem. You can compute the power of the DFT $|F|^2$ using `F.*conj(F)`, where F is the DFT and `conj` computes the complex conjugate.

(d) Center the Fourier transform using `fftshift`.

(e) Visualize the power spectrum. To obtain a useful visualization, use the logarithmic values $\log(1 + |F|^2)$, which reduces the contrast. Then normalize the values to lie in the range $[0, 1]$ by scaling linearly.

(f) Visualize the phase angle. You can use the Matlab function `angle` to obtain the phase angle. Make sure to map the resulting values to the range $[0, 1]$ for visualization.

4. **(1 point)** Constructing a filter directly in the frequency domain.

   (a) Compute the discrete Fourier transform of a grey-scale image of your choice using `fft2`. Since filtering in the Fourier domain corresponds to circular convolution, you need to zero-pad the image to avoid boundary artifacts. A safe choice is to zero-pad using twice the size of the image itself. You can specify zero-padding as parameters of `fft2`.

   (b) Construct a Butterworth highpass filter and a Butterworth bandpass filter directly in the frequency domain. Remember the transfer functions (that is, impulse responses) for the lowpass filter

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}},$$

   and the bandpass filter

$$H(u,v) = \frac{1}{1 + \left[\frac{WD(u,v)}{D^2(u,v) - D_0^2}\right]^{2n}}.$$

   Use the `meshgrid` function to generate coordinate matrices and use these to compute a matrix storing the distance function D.

   (c) Visualize the power spectrum of the Butterworth filters for different parameters D_0, and for the bandpass filter for different W. Describe the effects of these parameters.

   (d) Apply the filter by multiplying the DFT with the filter. Crop the result to the original size and show the image. Also show the power spectrum of the filtered image. Experiment with different parameters and show all results.

5. **(1 point)** Image restoration.

   (a) Generate a blurred image using a Gaussian lowpass filter. You can construct the filter using `fspecial` and apply it in the spatial domain using `imfilter`.

   (b) Add Gaussian noise of zero mean and variance 0.05 using `imnoise`.

   (c) Try to recover the original image using direct inverse filtering. Show the result. What do you observe? Try a blurred image without noise. Why do you get a better result?

   (d) Try to recover the image (with noise) using Wiener filtering, which is provided in Matlab by the function `deconvwnr`. Experiment with different constant values for the noise-to-signal power ratio. Show the results. What happens when you set the noise-to-signal power ratio to zero, and why?

6. **(1 point)** Image compression.
   In this exercise you will experiment with a simple method for image compression based on the Fourier transform. The basic idea of our method is to set a certain percentage of the Fourier coefficients of the input image to zero, and then write the resulting matrix to a file using lossless compression. Note that saving .mat files includes lossless

compression similar to "zip" files, so we do not have to worry about this part. The trick here is that as there are more zeros there in the data, the lossless compression algorithm becomes more effective and can reduce file size further. To compress color images, you can initially treat each color channel separately.

(a) Assume that the user specifies a certain number of coefficients (or a percentage of the total number) that should be set to zero. How would you select the coefficients that will be set to zero? Argue with Parseval's Theorem.

(b) Based on your strategy from (a), implement a Matlab script that sets a given percentage of the coefficients in the Fourier transformed image to zero.

(c) Save the matrix in a .mat file. If you use a Matlab Version 7 (2004 or newer), Matlab saves the file using lossless compression similar to "zip" files. Otherwise you should manually "zip" the file. Read out the size of the compressed file.

(d) Load the .mat file and apply the inverse Fourier transformation to your compressed image. Display the result.

(e) Calculate the L_2 error between the compressed image and the original image. How is this number linked to the original values of the coefficients that you set to zero? Argue again with Parseval's Theorem.

(f) Do experiments with various percentages of coefficients set to zero. Gather your results in a table that shows the percentage of zeroed Fourier coefficients, the file size and the L_2 error.

Bonus (Matlab) Exercise

Read about jpeg compression (e.g. Wikipedia). Describe the similarities and differences between jpeg compression and the procedure you have implemented in the previous assignment in a few sentences. Try to improve your algorithm by implementing one or more of the ideas you have read about.