Assignment A4: Frequency Domain Filtering

CS 6640
Fall 2020

Assigned: 28 September 2020

Due: 15 October 2020

For this problem, handin a report (PDF file as described in Lab Report Format) as required below as well as the Matlab .m files for the functions described by the headers below, and any help functions you write.

None of the functions should read or write files, write to the interpreter, draw, plot, etc. unless explicitly required by the header.

1. In A3, you compared two different ways of filtering an image with a $k \times k$ Prewitt filter. In this problem, you will investigate yet another way to implement image filtering. For this problem, we will do *fourier domain filtering* on `cell.tif`.

   1. Zero-pad your 3x3 Prewitt filter such that it has the same dimensions as `cell.tif`. Use the `padarray` function to help with this.
   2. Transform your padded Prewitt filter to the Fourier domain using the `fft2` function in MATLAB. Let us call this filter `P`.
   3. Using `fft2` and `ifft2`, perform Prewitt filtering on `cell.tif`. Confirm that the results are very similar to your 3x3 Prewitt filtering results from A3. Include the filtered image results from both A3 and the Fourier-domain filtering in your report.
   4. Repeat the performance experiment that you did in A3 with different sizes of Prewitt filter, but this time you do not need to fit a polynomial to your data points. Instead, plot your data points for this experiment against the polynomial best-fit curves that you obtained in A3. (Hint: use the `hold` feature in Matlab to overlay multiple plots.) Comment on the relative performance of these three ways to filter an image.

2. In this problem, we will investigate high-pass and low-pass filtering in the frequency domain.

1. Create a Gaussian filter using the provided `GaussianKernel` function. Visualize this filter using the `imagesc` and `surf` functions in MATLAB, and include these images in your report. Why is this considered to be a "low-pass" filter?

2. Let `G` denote the Gaussian filter that you created in the previous problem. Call `fftshift`, which is built into MATLAB, on `G` in order to produce an "uncentered" version of `G`. Let `G_uncentered` denote the uncentered version of `G`. Visualize `G_uncentered` using `imagesc`. (The reason why we create `G_uncentered` is because `G`, as returned by the `GaussianKernel` function, is a *centered* frequency-domain filter, but MATLAB assumes uncentered input into the `fft2` and `ifft2` functions. See Example 5.2 on p.138 of the text for an example of the role played by `fftshift`, and Ch. 5.14 of the text for discussion of the *centered* Discrete Fourier Transform.)

3. Filter `map1.jpg` using `G_uncentered` and `fft2` and `ifft2`. Give a qualitative description of what this filter does.

4. Create a filter `H = 1 - G` in MATLAB. Again, visualize using `imagesc` and `surf`. Why is this considered to be a "high-pass" filter?

5. Filter `map1.jpg` using the high-pass filter you just developed. Uncenter and use `fft2` and `ifft2` like you did before. Give a qualitative description of what this filter does.

3. Develop a texture feature analysis tool based on the 2D FFT power spectrum. For every 5x5 region in the image, compute the 2D FFT, compute the power spectrum, and use that as a 25-element feature vector. Produce a texture feature array, M*N by 25, and then use kmeans as in A2 to explore the usefulness of this for semantic region analysis (based on texture) of document images franklin and metro. Normalize the power spectrum by the value of the (0,0) component and see if this helps. Propose a performance measure (in terms of labeling semantic components) and report results. Develop the function *CS6640_FFT_texture* described below. This function should do the one thing the header says!

```
function T = CS6640_FFT_texture(im)
% CS6640_FFT_texture - compute FFT texture parameters
```

```
% On input:
%     im (MxN array): input gray level image
% On output:
%     T (M*Nx25 array): texture parameters
%        each texture parameter is a column vector in T
% Call:
%     T = CS6640_FFT_texture(im);
% Author:
%     <Your name>
%     UU
%     Fall 2020
%
```