CMPEN/EE455: Digital Image Processing I
Fall 2019
Project #3

assigned: 18 October 2019
due: Friday 26 October 2019 (weekend available!)

reading assignment: Same as HW #4

Filtering using Frequency-Domain Analysis

For the procedures below, you can use the following MATLAB functions and provided files:

- $F = \mathbf{fft2}(f, M, N)$: Compute the $M \times N$ DFT (FFT) of $M \times N$ image $f(x, y)$ to give $M \times N$ DFT $F(u, v)$.
- $f = \mathbf{ifft2}(F, M, N)$: Compute $M \times N$ DFT^{-1} (IFFT) of $M \times N$ DFT $F(u, v)$ to give $M \times N$ image $f(x, y)$.

  When suitable P and Q are used, both fft2 and ifft2 are useful for zero-padded quantities.
- $H = \mathbf{lpfilter}(TYPE, M, N, sig)$ (from Gonzalez, Woods, and Eddins's MATLAB-based book): defines an $M \times N$ lowpass filter $H(u, v)$ of the given TYPE ('ideal' or 'gaussian') and cutoff frequency sig. This function's MATLAB .m file is given on CANVAS along with supplemental .m file $\mathbf{dftuv}(M, N)$.
- File **mainproj3.m** on CANVAS demonstrates many of the ideas needed below.

1. *Basic DFT-based Frequency Analysis:* Let f be the $N \times N$ "checker" image.
   (a) Set $F(0, 0)$ of the DFT of $f(x, y)$ to zero and compute the inverse DFT to give a new image $g(x, y)$. What is the observable and analytical difference between g and the original f?
   (b) Using only $N \times N$ operations (i.e., you do NOT zero pad anything!), apply a Gaussian lowpass filter $H(u, v)$ with $sig = 15$ to f and denote $g(x, y)$ as the lowpass-filtered output image.
   i) Use the MATLAB function `mesh` in an appropriate way to make a 3-D plot of $|H(u, v)|$.
   ii) Give figures for $|G(u, v)|$ and $g(x, y)$.
   To display the Fourier-transform magnitude figures, be sure to do the following:
   a. modulate (multiply) $g(x, y)$ by $(-1)^{x+y}$ **per L12-11,12**
   b. scale $|G(u, v)|$ using $\log(1 + |G(u, v)|)$ **per L9-12,13**.
   (c) Repeat part (b), but now use appropriate zero padding and compute $P \times Q$ quantities.
   (d) For the results of parts (b-c), what impact does a filter have on the output image? What do the modulation and scaling operations do? Do you observe wraparound error? Discuss the nature of the zero-padded results.

2. *Filtering a Corrupted Image:* Let $f(x, y)$ be the "lake" image and create the corrupted image $c(x, y)$:

$$c(x, y) = f(x, y) + 32 \cdot \cos\left(\frac{2\pi 32 y}{N}\right)$$

   Clearly, some pixels in $c(x, y)$ will have values outside the 8-bit [0,255] range. So, you must perform all processing with sufficient precision!
   (a) Design a suitable notch filter $H(u, v)$ that when applied to $c(x, y)$ gives an image $g(x, y)$ that nearly resembles the original image $f(x, y)$. A notch filter rejects (i.e., sets to 0) a few specific frequencies while passing all others. You must describe how you designed your filter H by giving analysis to back up your design. **Lecture notes L15-17 → L15-20 are very helpful here!**
   (b) Give suitable pictures of:
   i. $f(x, y)$, $|F(u, v)|$, $c(x, y)$, $|C(u, v)|$
   (To display $c(x, y)$, be sure to clip [limit] $c(x, y)$'s values to the [0,255] range.)
   ii. $|H(u, v)|$, $g(x, y)$, $|G(u, v)|$
   iii. The image and Fourier-transform magnitude of the difference image $(f(x, y) - g(x, y))$.
   You, of course, will need to do appropriate modulation and scaling for plotting the DFT magnitudes.
   (c) Is it possible to completely recover f from c? Why or why not?