Assignment 3

General Info

- The work can be done anywhere where MATLAB and the related toolboxes are available.
- A **written report** is required. The report is free form but should include results, figures, and any code you wrote, as well as discussions of what you observe.
- Reports should be submitted as an electronic copy in **PDF** format on **Canvas** before the due date.
- Late submissions will not be accepted.
- Better documentation and clearer discussions of your work improves our ability to fairly mark your report. Make sure your report is well structured, organized, and clear. Your report has to follow the order of questions. Give all relevant code right before the results and discussion of each part, not in an appendix.

1. Frequency-Domain Sampling

The signal $x(n) = a^{|n|}$, $-1 < a < 1$ has Fourier transform:

$$X(w) = \frac{1-a^2}{1-2a\cos w + a^2}$$

(a) Plot $X(w)$ for $0 \le w \le 2\pi$, $a = 0.8$.
(b) Reconstruct and plot $X(w)$ from its samples $X(2\pi k/N)$, $0 \le k \le N-1$ for $N = 20$.
(c) Reconstruct and plot $X(w)$ from its samples $X(2\pi k/N)$, $0 \le k \le N-1$ for $N = 100$.
(d) Compare the spectra obtained in parts (b) and (c) with the original spectrum $X(w)$ and explain the differences.
(e) Illustrate the time domain aliasing when $N = 20$.

2. Frequency Analysis of Amplitude Modulated Discrete Time Signal

The discrete time signal:

$$x(n) = \cos 2\pi f_1 n + \cos 2\pi f_2 n$$

where f_1=1/18 and f_2=5/128, modulates the amplitude of the carrier

$$x_c(n) = \cos 2\pi f_c n$$

where f_c=50/128. The resulting amplitude-modulated signal is

$$x_{am}(n) = x(n)\cos 2\pi f_c n$$

(a) Sketch the signals $x(n)$, $x_c(n)$, and $x_{am}(n)$, $0 \le n \le 255$.
(b) Compute and sketch the 128-point DFT of the signal $x_{am}(n)$, $0 \le n \le 127$.
(c) Compute and sketch the 128-point DFT of the signal $x_{am}(n)$, $0 \le n \le 99$.

(d) Compute and sketch the 256-point DFT of the signal $x_{am}(n)$, $0 \leq n \leq 179$.
(e) Explain the results obtained in parts (b) through (d), by analytically deriving the spectrum of the amplitude-modulated signal and comparing it with the experimental results.

3. Redundancy in the Color Space

In the coming weeks you will be learning about compression of digital signals. In anticipation, we will explore how we can account for information that is perceptually redundant; specifically, visually redundant color information.

The YC_BC_R colour space is derived from the RGB colour space and has the following three components: Y, Luminance or Luma component obtained from RGB after gamma correction; C_B=B-Y, how far the blue component is from Luma; and C_R=R-Y, how far the red component is from Luma. This colour space separates the luminance and chrominance components into different channels, and is mostly used in compression for TV transmission.

(a) Load 'peppers.png' (already included in default MATLAB path) into MATLAB with `imread`. Convert the image into YC_BC_R and spatially downsample the luma channel Y by a factor of 4 using a 'bilinear' interpolator. Upsample the downsampled luma channel to its original size, then compute and report the mean squared error with the original luma channel. Reconstruct the image by combining the resampled luma with the original C_B, and C_R information and converting it back to RGB. Compare this reconstruction with the original 'peppers.png' and comment on your observations.
(b) This time, rather than resampling the luma, downsample **both** chroma channels C_B, and C_R. Measure and report the mean squared error between each resampled channel and its original values separately. Finally, reconstruct the image by combining the two resampled chroma channels with the original luma information and converting it back to RGB. Compare this reconstruction with the original 'peppers.png' and comment on your observations.
(c) Which of the two iterations resulted in a better looking image? Why? How is this relevant in the context of image compression?

4. Audio Denoising and STFT

Short-time Fourier transform (STFT) is a very powerful tool for audio processing. It enables us to perform powerful Fourier analysis on a time varying signal by restricting the transform to a specific location in time with the help of windowing functions. This simple, yet effective, tool can be used to perform audio denoising.

(a) Open and run 'Q4_student.m' and observe how the included functions are used to compute the STFT. Comment on the effects of 'w' and 'q' on the resulting STFT.
(b) Use the 'perform_threshold' function with 'hard' thresholding to denoise the audio signal. Convert the denoised STFT back into the time domain using the 'perform_stft' function and play the audio signal. Compare it the original and comment on the effects of denoising.
(c) Change the thresholding method from 'hard' to 'block'. Comment on the results. Play with the 'block_size' parameter of the thresholding algorithm and find a value that improves the results.

5. Homomorphic Filtering

Homomorphic filtering is a technique for separating multiplicative components of a signal with the help of the log transform. In the image domain, the log transform may be separated into low-frequencies (representing illumination) and high-frequencies (reflectance in the scene).

(a) Run the file 'Q5_student.m'. Explore all the functions that are being called and explain, in detail, all the parameters that are being passed into homomorphic.m.
(b) Use the code to extract all the low-frequency illumination from the in_the_tunnel image.
(c) Explain how the 'highboost.m' filter operates.

End of assignment 3