ECE-178 Homework #5

Due on Wednesday, November 13, 2024, 11:59 PM

Name:

Exploration of Naive Deconvolution

You are given a filter $h[k]$ and a signal $y[n]$ such that $y[n] = h[k] * x[m]$ (where $x[m]$ is not known) where $k < n$ and $k < m$.

1. Derive an equation which relates the lengths n, k and m. In other words, find m in terms of k and n.
2. Derive a formula which "reconstructs" a possible value of $x[n]$ (undoing the effect of the filter), assuming that $n = k = m$ (ignore the contradiction with (1) for now).
3. Is this value of $x[n]$ unique?
4. Does $x[n]$ always exist? If not, under which conditions does $x[n]$ not exist? Provide your answer in terms of y and h (and potentially their Fourier Transforms).
5. Let's generalize this to cases where n , k and m are not necessarily equal.

   a. Propose a transformation to $h[k]$ which makes the length of the filter equal to the length of your output. In one sentence, why is this important for the reconstruction formula derived in (2)?

   b. Mathematically justify why this transformation still generates a valid solution for $x[n]$. If you need to modify your result in (2), justify that choice here.
6. In the context of a 2D signal (an image) and a 2D filter (kernel), explain the intuition behind your answer behind (3). *Hint: What happens if your kernel is a LPF/HPF?*

[Programming] Motion Blur Correction

Motion blur occurs when subjects move (relative to the camera's frame of reference) during a camera exposure. For this assignment, we will consider the case of a still scene with a camera moving at constant velocity parallel to the imaging plane.

In our highly simplified blur model, the blurred image is the unweighted average of N images, where each image is taken at a slightly different location (due to the moving camera). You are given the final captured image, and some processed accelerometer data which contains the position of the camera (x_n, y_n) for each of the N frames.

1. Using `train_1.csv`, compute the convolution kernel which can represent the motion blurred image. Check your work by convolving your kernel with `train_1_clean.png` and verifying that it looks similar to the result `train_1.png` (it's ok if it doesn't match exactly).
2. Choose a deconvolution technique (refer to discussion) to undo the blurring of `train_1.png` and `train_2.png` (and check your result against the provided clean images). Apply the same technique to `test_1.png` and `test_2.png`, and include the deblurred images in your submission. Does it look like you expected?

Bonus (not graded): Extend your technique to cases of variable velocity (`bonus.png`).