**ISyE 8803 – Topics on High Dimensional Data Analytics**

**Exam I**

* For all questions, you are required to clearly state all assumptions you make and show all necessary details of your solutions.
* You are not allowed to discuss the exam content with your fellow students or receive aid on this exam.
* You are expected to observe the Georgia Tech Honor Code throughout the exam.
* Exam is due on June 24 at 11:59 pm. Late submission is NOT accepted. Please submit your solutions via Canvas.

**Question 1: Kernels and splines: brothers, or distant cousins? (35 points)**

The wheat dataset contains 100 wheat samples with specified protein and moisture content. Samples were measured by diffuse reflectance as log (1/R) from 1100 to 2500 nm in 20 nm intervals (70 data points). We want to use Kernel regression with the Epanechnikov kernel and smoothing splines to build two regression models to predict the protein and moisture content. For this purpose, we randomly split the data into training and test sets. The training set includes 80 functional data in “yTrain.csv” and their corresponding responses in “proteinTrain.csv”. The test data set includes 20 observations in “yTest.csv” and “proteinTest.csv”.

1. Code your own Kernel regression with Epanechnikov kernel and find the optimal bandwidth fitted to the mean signal of the training data. Report the optimal bandwidth and plot the estimated mean function along with the sample average signal.

Epanechnikov kernel is defined as:

1. Develop a prediction model to predict the protein and moisture content based on the kernel you found in part (1) and the training data.
2. Find the optimal lambda using GCV for the smoothing splines fitted to the mean signal of the training data. Report the optimal lambda and the number of spline coefficients corresponding to the optimal lambda.
3. Develop a prediction model to predict the protein and moisture content based on the extracted features from the training data using smoothing spline.
4. Evaluate and compare the performance of the estimated prediction models in terms of mean-squared error on the test set. Which model do you recommend?

**Question 2. Denoising and Pyramid Blending (35 points)**

This question has three parts, the first two parts are regarding to the denoising an image and second part is about blending a source image into the denoised image you got from part 1.

(1) *2D Bspline basis:* Denoise the image “space.jpg” using a 2D Bspline basis, and show the resulting image.

(2) *Threshold-Averaging Method:* Denoise the image using “space.jpg” the following steps, and show the resulting image.

Let denote the original intensity of the pixel located at of the image.

(a) Denoise the image using the following blurring mask: . Let

represent the resulting intensity of the pixel located at of the image.

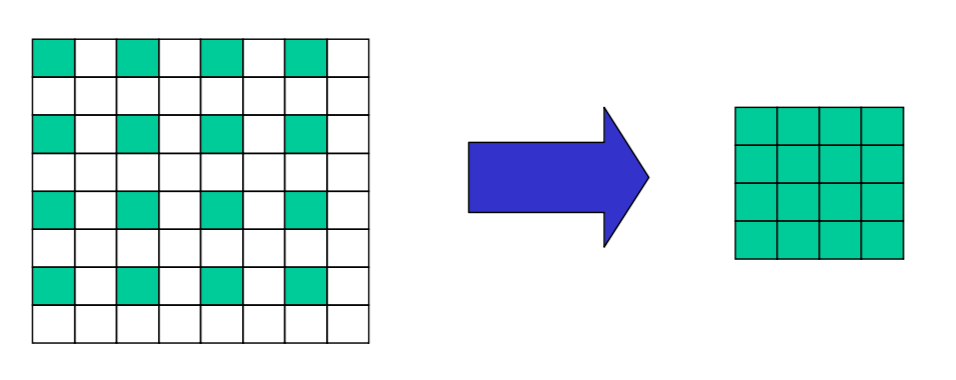
(b) Given the threshold , the final intensity of the pixel located at of the resulting image, can be obtained as follows:

(3) *Pyramid Blending:* The primary goal of this part is to seamlessly blend an object or texture from a source image (the “Image\_source.jpg”) into a background image (the denoised “space.jpg”). The simplest method would be to just copy and paste the pixels from one image directly into the other. Unfortunately, this will create very noticeable seams, even if the backgrounds are well-matched. How can we get rid of these seams without doing too much perceptual damage to the source region? The insight is that people often care much more about the gradient of an image than the overall intensity. Laplacian pyramid blending is a technique for blending, that decomposes the source image and background image using a Laplacian pyramid and then would combine and collapsed them to get the final image.We want to blend the “Image\_source.jpg” in to the denoised image of “space.jpg” you got from previous parts (Pick one on your choice from part 1 or 2). To make your job easier, we provided the mask for you, file “Mask\_source”. This image is aligning the source image and background image and you need it for part of blending algorithm. To do the Pyramid Blending, follow the below steps:

(a) Build the Laplacian pyramids of “Image\_source.jpg” and denoised “space.jpg”.

To build the one layer of pyramid, you have to pass both images to a Gaussian filter, then subtract the original images from their corresponding smoothed images and you will have one layer of Laplacian Pyramid.

(b) To get the next level of the pyramid, you should down sample the Gaussain filter passed images from the previous part. The general method for down sampling is:



**Original Image Down Sample Image**

And then pass the down sampled images to the Gaussian filter and subtract the smoothed images from down sampled ones to get the Laplacian layer. Repeat this process to build the 6 layers Laplacian pyramid.

(c) To do the Pyramid blending, you also have to build the Gaussian pyramid of “Mask\_source” image. The process is the same as part (b), except that you should not subtract from original image (Just keep the Gaussian smoothed one)

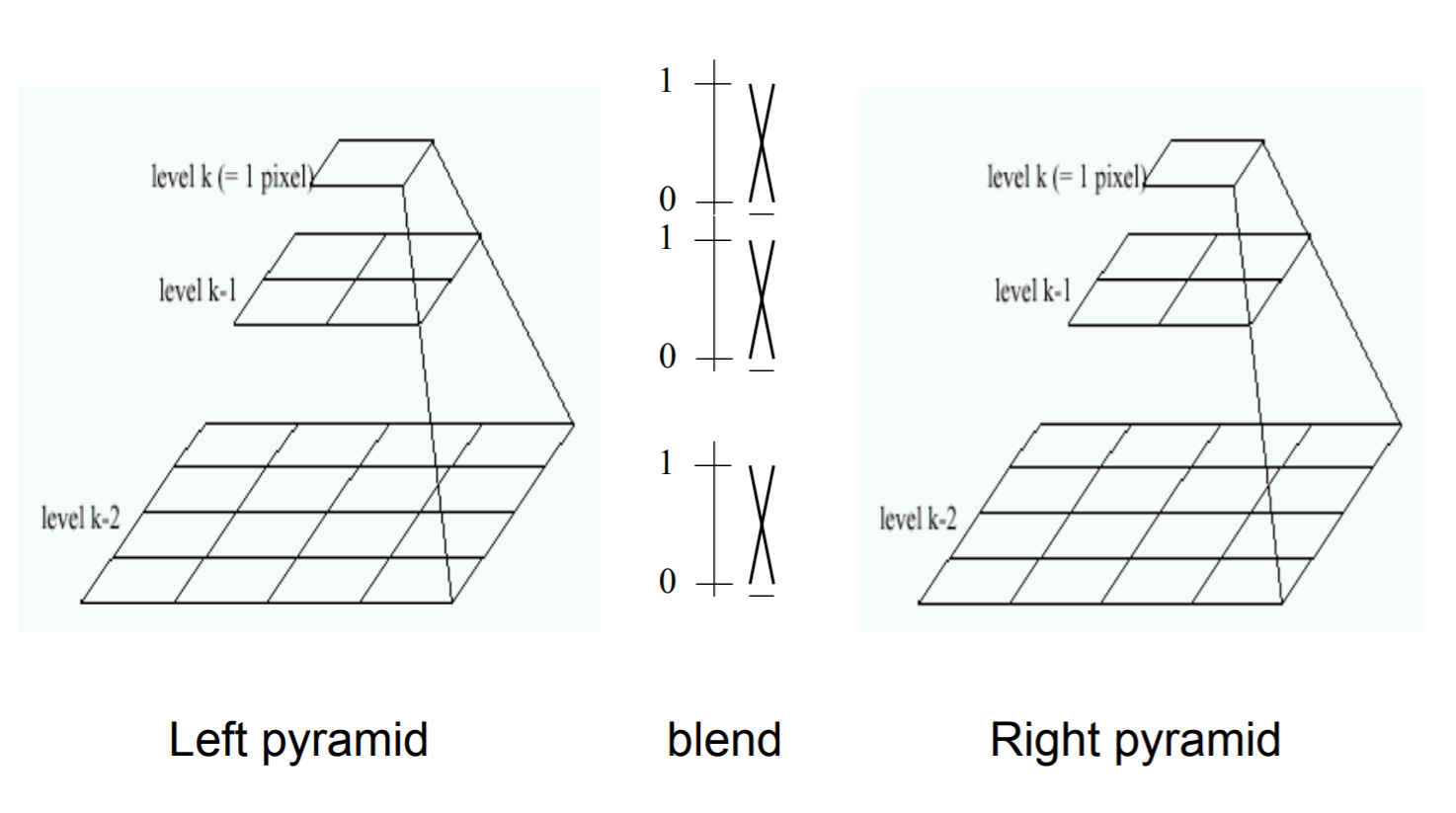
(d) Form a combined pyramid from Laplacian pyramids of denoised space and image source, using nodes of the Gaussian pyramid of mask as weights:

Combined pyramid = (mask pyramid)\*( Laplacian pyramids of image source) +

(1- mask pyramid)\*( Laplacian pyramids of denoised space)

Then collapse the combined pyramid to get the final blended image.

For this question, please show the result of Gaussian pyramids of the denoised space image, source image and the final pyramid blended image. Note that due to the low frequency of images, images in Laplacian pyramids would be mostly dark, especially in lower levels, that is why you do not need you to show them.



Left Pyramid Blend Right Pyramid

**Question 3. Tensor decomposition (30 points)**

1. Consider the following optimization problem:

Where denotes the column of matrix .

Find the closed-form solution for this optimization problem. Note: denotes the outer product. You may assume that and are known.

(Hint: rewrite the optimization model in a vector form by using Khari-Rao product).

1. In this part, a neuron activity dataset is given in the file ‘neurondata.mat’. Consider a recording of N neurons over K experimental trials. We assume neural activity is recorded at T time points within each trial. This dataset is naturally represented as a T×N×K array of firing rates, which is known in mathematics as a third-order tensor. Each element in this tensor, , denotes the firing rate of neuron n at time t within trial k. Here, the indices t, n, and k are in the range from 1 to 500, 50, and 100, respectively.

Perform CP decomposition on the dataset (use AIC criteria to find the optimal rank). Plot the first three column vectors for each factor matrix and interpret them.