Chennai Mathematical Institute
Computer Vision
January 17, 2020
Due Date: January 27, 2020 @ 6:00PM

Assignment 1: **Image Filtering**

Note: Use Python for this assignment. There are many libraries available in Python including: Python Imaging Library (PIL), Scikit-image, and Mathplotlib.

Instructions:

1. Write your code in Jupyter Notebook.
2. Write your answers and observations in TeX or Word and submit a PDF document on Moodle.
3. There are a few images provided in a folder for you to use to test your code. Feel free to test your code on additional images to support your findings.

Good luck!

In this assignment, you will:

1. Convert a color image into a grayscale image: (3 points)
   - Read the input image "hill.png". Convert it to a grayscale image, display and save it as "hill_gray.png"

2. Read in a grayscale image and linearize the intensity values: (3 + 5 = 8 points)
   a) Load the image in "Einstein.jpg" into a Numpy array. Originally, it will be in the form of a 2D-array of unsigned integers. Check and report how many bits per integer there are in the image, and what its width and height is. Then, convert the image into a double-precision array.
   b) Convert the image into an array within the range [0,1]. Do this by applying a linear transform (shift and scale) to the image, so that the minimum intensity value is mapped to 0, and the maximum intensity value is mapped to 1. Now, multiply the intensity values by the maximum value for the representation (eg. 255 for 8-bits). Display and save the image after linearization.
   c) What do you observe in the linearized image?

3. (8 points) Write down the formulas for computing the padded pixel values $f_p(i,j)$ as a function of the original pixel values $f(k,l)$ and the image width and height (M,N) for each of the padding modes:

   - Zero

- Constant
- Clamp
- Mirror

Note: See exercise 3.8, page 196 in Richard Szeliski's text.

4. (16 points) Write a function that will take as input an image (grayscale or color), a padding specification and width of padding in each dimension and returns the appropriately padded image. Test your function on the given input images for widths of 1, 2 and 3 in each dimension. Display and store your results.

5. (25 points) Implement convolution with a separable kernel. The input should be an image (grayscale or color), along with the vertical and horizontal kernels. Utilize the padding functions implemented above. Realize the box filter and the Gaussian filter for different kernel sizes (5x5, 7x7, 11x11) using your convolution function.

6. (10 points) Apply the box and Gaussian filters of different sizes to smooth, and then sharpen the given input images. Store your results.

7. (20 points) Implement the median filter using the same filter sizes as with the Gaussian and box filters. As with the linear filters above, apply the median filters of different sizes to smooth, and then sharpen the given input images. Test and store your results.

8. (10 points) Compare your results from tasks 7 and 8 above and identify the filters that provide the best results for smoothing and sharpening on the given input images.