Goals

The purpose of this project is to get hands-on experience with histogram based methods and understand how they can be used in conjunction with other methods such as connected components and topological denoising.

Image colorization

You will write software to perform the following functions:

1. A histogram equalizing transformation routine
   a. Input: single channel image, i.e. gray scale image
   b. Output: a 1-dimensional array of integers corresponding to the histogram equalization transformation mapping all values from 0 to L-1, where L-1 is the maximum intensity found in the input image, to new values.

2. An intensity transformation routine
   a. Input: 1) single channel image, i.e. gray scale image 2) an array of integers corresponding to an intensity transformation
   b. Output: single channel image that is the result of applying the given intensity transformation to the input image

3. An inverse histogram equalization transformation routine
   a. Input: single channel image, i.e. gray scale image or a one of the channels of a color image
   b. Output: a 1-dimensional array of integers corresponding to the inverse of the histogram equalization transformation

4. A color matching routine that uses the above routines to colorize a gray scale input image using a reference color image
   a. Input: 1) gray scale input image, 2) color reference image
   b. Output: Colorized version of input image

Use the Lake Powell images (attached to this project) to test your color matching routine, you will need to convert one of the images to gray scale and use other image to color it. Further, experiment with your own images, test what happens when use two images from different styles/contexts.

Counting cells

You will write software to perform the following functions

1. Otsu’s thresholding method
   a. Input: single channel image
   b. Output: a binary image thresholded using Otsu’s method (maximizing between class variance)

2. Adaptive thresholding method
   a. Input: 1) single channel image, 2) block size bsize, 3) minimum variance mvar
   b. Output: binary image created by dividing the input image into non-overlapping square blocks of size bsize, checking whether a given
block has variance greater than \textit{mvar}, and if it does thresholding it at an intensity level equal to the mean intensity of the block plus one standard deviation of intensity of the block.

3. Given the cell image (attached to this project) count number of cells using both thresholding methods and the connected components program you wrote for the previous project. Comment on the differences of the two thresholds in the context of this problem. Note you should use topological denoising as in the previous project if necessary to make your count more accurate.