**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 12a, Week 15, due May 2, 2021**

**Global thresholding**

**Goals**

1. Convert images to binary by thresholding
2. Experiment with various thresholding algorithms

**Procedure**

Write a Processing program to convert images to binary (black and white). The goal is to isolate items of interest (foreground) from the rest of the image (background). Set the foreground to white and the background to black (some computer vision systems reverse this, with black foreground against a white background).

You may begin by using the Processing filter’s THRESHOLD mode. In that method, the threshold is a floating point value between 0 and 1. However, you will probably find that you need more control than the filter method gives you and so you will want to write your own thresholding function.

Skeleton code is provided in the Blackboard download. This code is primarily for histogram display; an image histogram can often give you a good idea of which values to try for thresholding. Most of the images in the data folder are “green screen” images intended for chroma key applications. The skeleton code converts the input image to grayscale before thresholding. These images present some challenges but you are welcome to try more difficult ones; in particular, you should try *drakearmadaletter.jpg.*

Compare the following thresholding methods:

1. Set the threshold to the average pixel value

2. Set the threshold to the median pixel value

3. Calculate the threshold using the Ridler-Calvard algorithm (given below)

4. You could try a box threshold for some images, in which the foreground is between two values

**Deliverables**

Submit your pde file on Blackboard by the due date.

Ridler-Calvard algorithm (<http://en.wikipedia.org/wiki/Balanced_histogram_thresholding>)

Algorithm findThreshold(Image src):

T0 = 0

T1 = 128

while T0 != T1:

T0 = T1

M0 = the mean of all samples < T0

M1 = the mean of all samples >= T0

T1 = (M0+M1)/2

return T0