ECE/OPTI 532, Fall 2020

Homework 4 Assignment: Histogram Thresholding

Due Thu Oct 22 at 5:00 pm

Write a computer program to threshold a grayscale image to obtain a bi-level image. Write your own code for the steps of the algorithm; don't use existing thresholding-related functions. Use Kittler and Illingworth's Kullback information minimization approach. You do not have to apply the correction terms for tail truncation. Here's a possible command-line user interface:

```
USAGE
    thresh [-t T] infile outfile

    T: user-specified threshold; 0 <= T <= 255;
       default: T is automatically computed using Kittler's method
    infile: the input grayscale image
    outfile: the output bi-level image

DESCRIPTION
    If a fixed threshold is not specified, then it is automatically
    computed by minimizing the Kullback information measure.
    Specifically, the image is thresholded as follows:
        y[n] = 255 if x[n] >= T
             = 0   if x[n] <  T
```

Submit the following items:

- Your commented source code files.
- Run your program on the `address.png` image, and submit the thresholded image. (You should also try your program on `graybook.png`, but don't submit that image.)
- Show the threshold value that Kittler's algorithm found for the `address.png` image.

Reference

J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41-47, 1986.

Recursive Update Formulas

```
mu1  = (q1prev*mu1prev+t*P[t])/q1;
var1 = (q1prev*(var1prev+(mu1prev-mu1)*(mu1prev-mu1))
       +P[t]*(t-mu1)*(t-mu1))/q1;
mu2  = (mu-q1*mu1)/q2;
var2 = (q2prev*(var2prev+(mu2prev-mu2)*(mu2prev-mu2))
       -P[t]*(t-mu2)*(t-mu2))/q2;

H = (q1*log(var1)+q2*log(var2))/2-q1*log(q1)-q2*log(q2);
```

H Values for **address.png** Image

gray	H Value
27	2.76137
28	2.76137
29	2.76132
30	2.76121
31	2.7611
32	2.76099
33	2.76078
34	2.76054
35	2.7604
36	2.76018
...	
137	2.79171
138	2.78289
139	2.77582
140	2.76994
141	2.76562
142	2.76363
143	2.76249
144	2.76179
145	2.76169
146	2.7616