BLG 477 E Multimedia Computing

Homework #2 : The goal of this homework is to implement Otsu's binary segmentation and K-Means Clustering. You should write your own code, it is not allowed to use any thresholding functions from other sources.

Otsu's algorithm

Otsu's segmentation algorithm automatically clusters pixels into two groups: background and foreground. The main idea of Otsu's algorihtm is to find threshold that would maximize between-class variance and minimize withinclass variance. Then, all pixels are classified into 2 classes using that threshold.

First step is to create a histogram of pixel values. Since we are taking 8-bit image, there are 256 possible values for pixels. We compute histogram ***h*** with 256 bins where the height of each bin corresponds to number of pixels that have that pixel value (from 0 to 255).

Given an image, we can estimate probability of pixel value ***i*** by simply dividing the height of bin ***i*** in the histogram by the total number of pixels *N*.

$$p_i = \frac{h_i}{N}$$

Given threshold ***k***, probability of class 0 is sum of probabilities of pixel values smaller than ***k***.

$$w_0 = \sum_{i=1}^{k} h_i \text{ and } w_1 = 1 - w_0$$

Since the goal is to maximize between class variance, we need to calculate the means and the variances of both classes.

$$\mu_1 = \frac{1}{w_0} \sum_{i=1}^{k} ip_i \text{ and } \mu_2 = \frac{1}{w_1} \sum_{i=k+1}^{256} ip_i$$

It is sufficient to increase between class variance

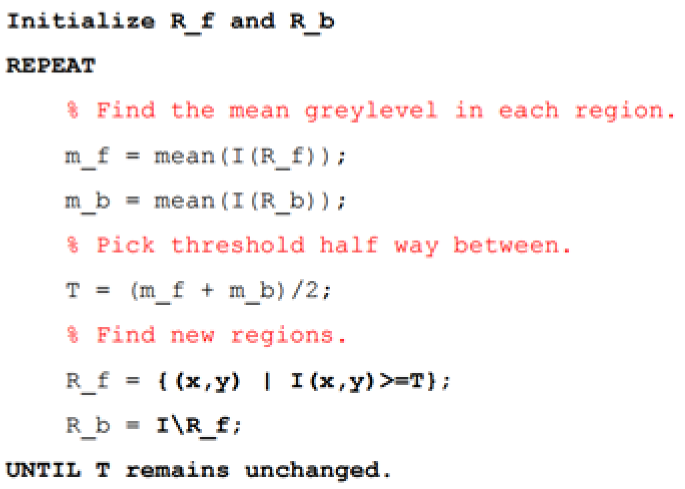
$$\sigma_b^2 = w_0 w_1 (\mu_0 - \mu_1)^2.$$

This will decrease within-class variance too. We calculate the between-class variance for each threshold ***k*** and picked the threshold that maximizes the variance. Depending on the image, sometimes Otsu's algorithm runs several times for better results.

Each of the RGB channels of color image is segmented separately. Then, we combine results from RGB channels using AND operator, where we set the final segmented image pixel to 1 only if all three segmented channels have 1 at that pixel location. We also have to specify manually which class is foreground and which is background.

K-Means Clustering

The algorithm is given below:



```
Initialize R_f and R_b
REPEAT
    % Find the mean greylevel in each region.
    m_f = mean(I(R_f));
    m_b = mean(I(R_b));
    % Pick threshold half way between.
    T = (m_f + m_b)/2;
    % Find new regions.
    R_f = {(x,y) | I(x,y)>=T};
    R_b = I\R_f;
UNTIL T remains unchanged.
```

Policy: Collaboration in the form of discussions is acceptable, but you should write your own code by yourself. Cheating is highly discouraged for it could mean a zero or negative grade from the homework. If a question is not clear, please let me know via email.

Submission Instructions: Please submit your homework through the Ninova web site. Please zip and upload all your files using filename studentID_HW2.zip. You must provide all functions you wrote with your zipped file. Functions you do not submit may cause you lose a portion of your grade. Please make sure that you comment your code. You must also write a report that contains input and output images. Input images are given in the images folder and if you want, you can apply your code into other images. You should also explain the usage of your program. You should also include report.pdf file with your zipped file