BLG 453E Homework - 5

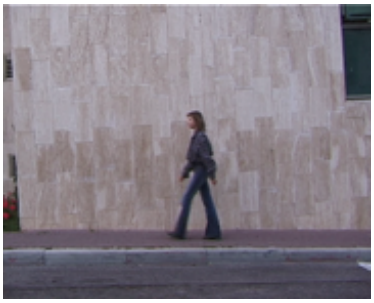
Due 05.01.2018 23:59

Policy: Please do your homework on your own (Do not copy paste your solutions from the internet or your friends). The code and the report you submitted must be your own work. All code must be implemented using **Python** programming language and **OpenCV Python wrapper**.

For your questions: albay@itu.edu.tr

1. **Background Subtraction**

   In this problem, you are going to implement a median filter to model background image in a sequence. Then you are going to use this background model to detect moving objects in the scene. The image sequence you are going to use in this problem consists of a walking person (in "daria_walk" folder). A sample image is given below:

   To detect the moving object, you are going to create a background model (array) and obtain a binary image by subtracting the frames from the background model. Background model is going to created using median filter.

   Iteratively do the following steps:

   (a) Read i^{th} image in the sequence, convert it to grayscale

   (b) For each coordinate in the image
      i. Construct a 1D signal using the intensity values of that coordinate in previous frames.
      ii. Find the median of this 1D signal
      iii. Update the background model by assigning the new median value.

   (c) After updating the background model for each coordinate, subtract the current frame from the background model and obtain binary image that represents the moving objects.

   (d) Record the binary image.

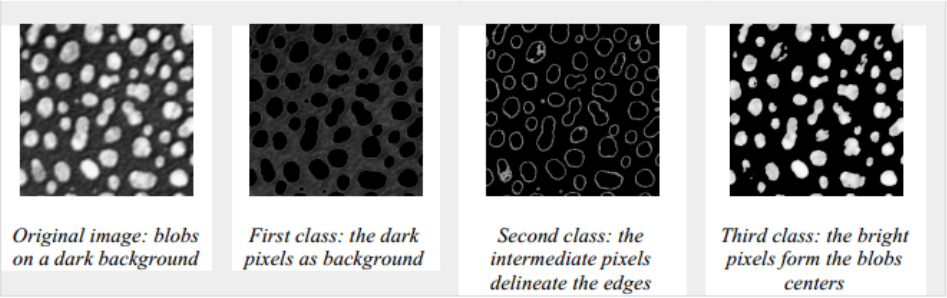
2. **Three-Class Segmentation using Otsu Tresholding**

   Check page 4-9 of the following slides: EI2011_12_16_segmentation_double.pdf

   Otsu Thresholding algorithm typically classifies pixels in two classes (or two set of objects): the one that have their intensity lower than a certain threshold (generally, the background), and the other (the interesting features).

   In this problem, you will implement and adapt the Otsu Tresholding algorithm to generate multiple thresholds and multiple classes from one single image. This will allow us to get background pixels, bright pixels and intermediate pixels. This might be of interest for images where there is such a pixel populations.

   In the example depicted below, based on the blob image, one could get the background, the blobs center and the blob edges out of it.



Original image: blobs on a dark background | *First class: the dark pixels as background* | *Second class: the intermediate pixels delineate the edges* | *Third class: the bright pixels form the blobs centers*

3. **Image Alignment by PCA**

   Here, you are going to use PCA to align images so that the shapes in the images overlap. You are given a set of binary guitar images in different orientation and scale. By applying PCA on the x, y points in silhouette images, you can obtain the directions of the principal axes in the image. Do the following for each guitar image:

   (a) Load the guitar image "guitar-X.gif" from the "Guitar_images" folder.

   (b) Use the 2D points in the silhouette (non-zero pixels) to construct a single $2 \times N$ matrix T.

   (c) Subtract the mean. Calculate the average point x, y and then subtract it from each original point in T.

   (d) Calculate the eigenvectors and eigenvalues of the covariance matrix S using built-in Singular Value Decomposition (SVD) function. There will be two eigenvectors that correspond to the directions of the principal axes of the guitar. The eigenvalues you obtained, represent the magnitudes of the direction vectors.

(e) The eigenvector with the highest eigenvalue corresponds to the major axis of the guitar. Using this eigenvector, you can calculate angle between the guitar and vertical coordinate axis. To make guitars in upright position, rotate the image by the angle you find. Show the results.

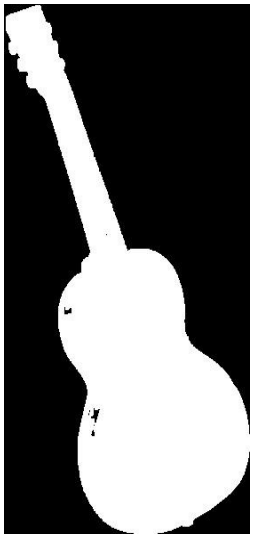


Figure 1: Original image

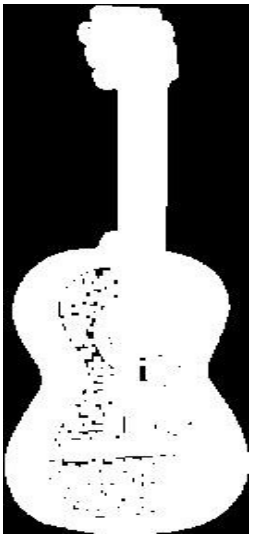


Figure 2: Aligned image