Stevens Institute of Technology

Department of Computer Science

CS558: Computer Vision

Homework Assignment 2

Submitted by:
Hadia Hameed
CWID: 10440803

Submitted to:
Prof. Xinchao Wang

May 3, 2019

1 Problem 1 - k-means segmentation:

Apply k-means segmentation on $white - tower.png$ with k=10. The distance function should only consider the RGB color channels and ignore pixel coordinates. Randomly pick 10 RGB triplets from the existing pixels as initial seeds and run to convergence. After k-means has converged, represent each cluster with the average RGB value of its members, creating an image as in slide 18 of Week 6.

Solution:

Algorithm 1 k-means segmentation.

procedure myKmeans(I,k,thresh)
  initial_centroids $\leftarrow$ initialize k cluster centers
  $D = \infty \leftarrow$ initialize distance for each pixel i.
  $\mu = 1000 \leftarrow$ compare with threshold for convergence
  while μ > thresh **do**
    for *all pixels i in image I* **do**
      for *all clusters j in k* **do**
        $D(i, j) \leftarrow$ distance between ith pixel and jth cluster center.
      clst $\leftarrow$ assign cluster to ith pixel for which D(i,:) is minimum.
    new_centroids $\leftarrow$ mean center of each new cluster
    $\mu \leftarrow$ distance between old and new centroids
  output $\leftarrow$ each cluster represented with the average RGB value of its members.
return *output*

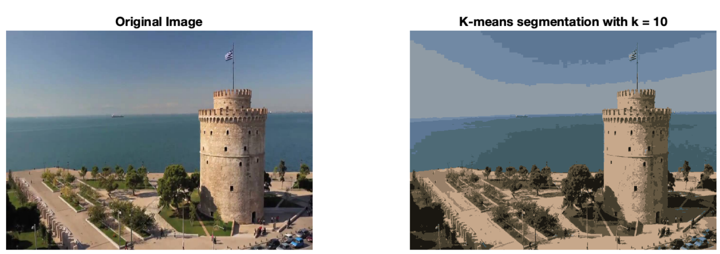

Figure 1: k-means segmentation

2 Problem 2 - SLIC:

Apply a variant of the SLIC algorithm to *wt_slic.png*, by implementing the following steps:

1. Divide the image in blocks of 50x50 pixels and initialize a centroid at the center of each block.
2. Compute the magnitude of the gradient in each of the RGB channels and use the square root of the sum of squares of the three magnitudes as the combined gradient magnitude. Move the centroids to the position with the smallest gradient magnitude in $3x3$ windows centered on the initial centroids.
3. Apply k-means in the 5D space of x; y;R; G;B. Use the Euclidean distance in this space, but divide x and y by 2.
4. After convergence, display the output image as in slide 41 of week 6: color pixels that touch two different clusters black and the remaining pixels by the average RGB value of their cluster.

Solution:

Algorithm 2 SLIC [1]

procedure MYSLIC(I,S,THRESH)
  $N \leftarrow$ Total number of pixels in input image I
  $K = N/S^2 \leftarrow$ Total number of superpixels
  $C_k \leftarrow$ initialize K cluster centers obtained by sampling pixels at regular grid steps S
  Move centroids to the lowest gradient position in a 3 x 3 neighborhood.
  $D \leftarrow \infty$ initialize distance for each pixel i.
  $image_{5D} \leftarrow$ 5D features of each pixel in image I
  $image_{5D}(:, end) = -1 \leftarrow$ assign initial cluster -1 to each pixel
  $\mu \leftarrow 1000$ compare with threshold for convergence
  while μ > thresh **do**
    for *each cluster k in* C_k **do**
      for *each pixel i in 2S x 2S neighborhood of* C_k **do**
        $D \leftarrow$ distance between ith pixel and kth C_k.
        if $D < d(i)$ **then**
          $d(i) \leftarrow D$
          $image_{5D} \leftarrow$ assign kth label to the corresponding pixel
    $C_k \leftarrow$ mean center of each new cluster
    $\mu \leftarrow$ distance between old and new C_k
  output $\leftarrow$ image with pixels that touch two different clusters black and the remaining pixels by the average RGB value of their cluster.
return *output*

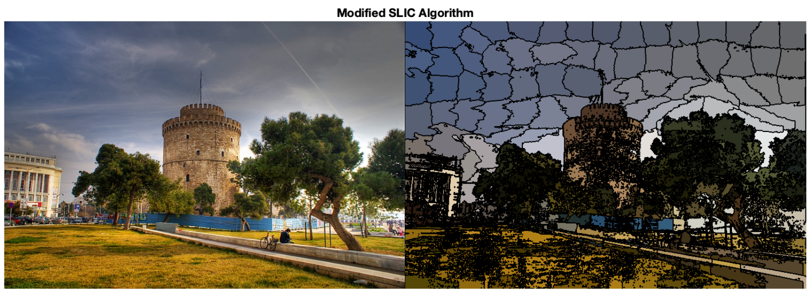

Figure 2: Simple Linear Iterative Clustering (SLIC) Algorithm [2]

References

[1] "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods." http://www.kev-smith.com/papers/SMITH_TPAMI12.pdf. Accessed: 2019-05-03.

[2] "SLIC Superpixels." http://www.kev-smith.com/papers/SLIC_Superpixels.pdf. Accessed: 2019-04-29.

