Computer Vision

Exercises of Lab 7

Exercise 7.1: K-Means clustering

The goal of this exercise is to get some experience with K-means clustering and think about the choice of features that we use for clustering.

Open Exercise7_1.m and run the code. In line 26 all pixels are assigned to cluster id 0 by default.

Construct a feature map with RGB color features for each pixel/observation.

Use MATLAB's kmeans-function for clustering all observations into numClust clusters. Save the cluster indices of all observations in the variable cinds. See the documentation of kmeans in MATLAB to make sure you pass the input variables correctly.

Modify the feature representation to also include the spatial positions (variables X and Y). Does it work the way you intended? If not, figure out why, and try to fix the problem.

Try playing around with the number of clusters and see the impact.

Exercise 7.2: Mean shift clustering

The goal of this exercise is to experiment with another clustering algorithm, called Mean Shift clustering.

However, as opposed to traditional mean shift where each pixel/point is handled separately, here, we use an approximation. We only calculate mean shift for a limited number of points (called seed points), and thereby get the modes of these points. Additionally, we store the path of each seed point that led it from its original position in feature space to the mode. We then assume that other points (non seed-points) that are located closely along this path will get the same mode, and we therefore assign them the same cluster.

Open Exercise7_2.m and look through the code. Here, we use 100 seed points and track their positions as we iteratively apply the mean shift to each. In each iteration, the mean shift of each seed point is given by the average position of all pixels, whose feature vector is within a radius of the seed point's feature vector. For each seed point, we keep track of all those pixels that at some point during the loop were within the radius. This results in 100 clusters (one for each seed point). Finally, we iteratively merge the clusters using bottom-up clustering. In each iteration, the two clusters lying closest to each other are merged. This proceeds until no two clusters are closer than the radius.

Task 1: The variable Zref_vec is the feature vector of the current seed point (the features are RGB + spatial position). In line 78, change the code such that the variable dist(i,j) is the Euclidian distance between Zref_vec and the feature vector at spatial position (i,j). This feature vector is Z(i,j,:).

Task 2: The mean shift of each seed point is given by the average of the pixel positions of all those pixels that lie within the radius. The indices of these pixels are stored in the variable add_to_list. Make sure you understand how add_to_list is calculated. Modify the code in lines 92-93 to calculate the mean shift (i.e., do the appropriate calculations of variables new_seedRow and new_seedCol; new row and column coordinate of the seed point, respectively).

Are all pixels assigned a cluster label with this method?