CMPE 565- Assignment #3

Due: 29.12.2019, 23:59

Note: Homework is to be done individually. You need to submit you reports and codes to Moodle. Please do not insert your codes to your reports. Codes should come separately. In your report just put the input images and output results and explicitly stated parameters being used (like sigma parameter for Gaussian).
The images are provided in "data" folder.

1) **[40]** Stitch the images given in **data** folder using the corresponding homography matrix (also given in the assignment folder). Write image stitching function:

   [stitchedImage]= ***Stitch*** (*image1*, *image2*, *hom*, *homInv*)

   image1, image2: the input images,
   hom: the homography and *homInv* its inverse,
   stitchedImage: result.

   Follow these steps:

   i. Compute the size of "stitchedImage. " To do this project the four corners of "image2" onto "image1" using Project and "homInv". Allocate the image.
   ii. Copy "image1" onto the "stitchedImage" at the right location.
   iii. For each pixel in "stitchedImage", project the point onto "image2". If it lies within image2's boundaries, add or blend the pixel's value to "stitchedImage. " When finding the value of image2's pixel use BilinearInterpolation (you can use interp2 function in Matlab).

   Required: For all image pairs in data folder run the stitching algorithm. Show results on your reports.

2) **[60]** You are going to implement Mean-Shift segmentation. As discussed in the class, the mean shift algorithm is a generic procedure for clustering a *d*-dimensional data set by associating each point in the data set to a peak (mode) of the data set's probability density. For each point, mean shift computes its associated peak by first defining a window of a given size (***bandwidth***) at the data point and computing the mean of the points that lie within the window. The algorithm then shifts the window to the mean and repeats until convergence, i.e., until the shift amount is less than a threshold (***stopT***). At each iteration the window will shift to a more densely populated portion of the data set until a peak is reached.
   Your code should start with the following function:

   [labels, peaks]=segmentMS(image, bandwidth, stopT)

where *image* is the input RGB image, *bandwidth* and *stopT* represent the window size and the stopping threshold respectively, and labels is the label matrix that represents the segmentation result, and peaks is an $m \times d$ matrix where each row corresponds to one of the m resulting peaks and each peak is represented by its d dimensional feature vector. This function should call the actual mean shift procedure that is defined as following:

peaks=MeanShift (data, ind, bandwidth, stopT)

where data is the $n \times d$ data matrix as input where each row corresponds to one of the n data points and each data point is defined as a d-dimensional feature vector. The parameters of the procedure include *bandwidth*, the size of the search window, and *stopT*, the threshold for the stopping criterion, *ind* is the index of the data point for which we wish to compute its associated density peak and *peak* is the resulting d-dimensional peak vector associated with *ind*. The labels matrix stores the id of the peak that the corresponding pixel's feature point converted to. Note that, the peaks should be compared after each call to MeanShift and similar peaks should be merged. In your implementation, you can consider two peaks to be similar if the distance between them is
$\leq bandwidth/2$. Also, if the peak associated with a data point is found to already exist in the peaks matrix, then its computed peak should be discarded and it should be assigned the label of the already existing peak in peaks.

Required: Run segmentation algorithm for your own data. Show results on your reports.