# CS 663 : Digital Image Processing : Assignment 3

Instructor : Suyash P. Awate

Due Date : 11 Sep 2015, Sunday, 11:55 pm

Instructions for submission are at `www.cse.iitb.ac.in/~suyash/cs663/submissionStyle.pdf`

The folder structure is at
`www.cse.iitb.ac.in/~suyash/cs663/assignment3_Segmentation.zip`

Note: The input data / image(s) for a question is / are present in the corresponding data/ subfolder.

**5 points** are reserved for submission in the described format.

1. (30 points) Harris Corner Detection.

   Input image: `2/data/boat.mat`.

   Assume the pixel dimensions to be equal along both axes, i.e., assume an aspect ratio of 1:1 for the axes.

   Shift and rescale the intensities in the image to lie within the range $[0, 1]$.

   Implement the Harris corner detector algorithm. The parameters underlying this algorithm are: two Gaussian smoothing levels involved in computing the structure tensor, the constant $k$ in the corner-ness measure. Tune these three parameters to get the best results.

   • Write a function `myHarrisCornerDetector.m` to implement this.

   • Display the derivative images, corresponding to the derivatives along the $X$ and $Y$ axes.

   • Display the images (along with a colormap) of the two eigenvalues of the structure tensor, evaluated at each pixel.

   • Display the image (along with a colormap) of the Harris corner-ness measure. Positive values in this image must correspond to a corner structure in the image.

   • Report all three parameter values used.

2. (40 points) Image Segmentation using mean shift.

   Input image: `3/data/baboonColor.png`.

   Take this $512 \times 512$ pixel image, smooth it using Gaussian convolution with standard deviation $1$ pixel width, and subsample the smoothed image by a factor of $2$ in each spatial dimension to produce a $256 \times 256$ image. Use this smaller-sized image for the following experiment. If this image is still too large for your computer's memory, then you may resize further.

Implement the algorithm for mean-shift image segmentation using both color (RGB) and spatial-coordinate (XY) features. Tune parameters suitably to get a segmented image with at least 5 segments and no more than 50 segments. To improve code efficiency, you may use Matlab functions like knnsearch(), bsxfun(), etc. For this image, about 20 iterations should be sufficient for reaching close to convergence. You may select a random subset of nearest neighbors, in feature space, for the mean-shift updates to reduce running time. Each iteration can run in about 10-20 seconds on a typical personal computer.

• Write a function `myMeanShiftSegmentation.m` to implement this.

• Display the (i) original image along with (ii) the segmented image that shows color-coded pixels (and, thus, segments) using the color component of the converged feature vectors.

• Report the following parameter values: Gaussian kernel bandwidth for the color feature, Gaussian kernel bandwidth for the spatial feature, number of iterations.