**Code structure:**

**main.m:** the main code runs the 4 different tasks based on 4 images with various parameters settings, and output the grayscale images and 3D plot with m(x,y), m’(x,y) and final segmented results.

**support functions:**

**g.m: the Gaussian part of GEF**

This function is used to compute the circularly-symmetric Gaussian of one pixel. The inputs are the value of sigma and the x/y coordinate of the pixel. The input includes only x or only y since we compute convolution in x and y separately in main function. The output is the circularly-symmetric Gaussian value of that pixel. This function does not call any other functions.

**hx.m: GEF of h(x)**

Since we compute GEF in x and y separately in main function, is separated into and . This function is used to compute the GEF in x, i.e. . The inputs are F, theta, sigma, and x coordinate of the pixel. The output is the computation result of x coordinate. This function does not call any other functions.

**hy.m: GEF of h(y)**

This function is the same as hx.m. The only difference between them is that the inputs of this function are F, theta, sigma, and y coordinate of the pixel, not x. This function is used to compute the GEF in y, i.e. and does not call any other functions.

**segment.m:** This function is used to do segmentation with discriminative threshold of each texture for classification. The inputs are results after Gabor filter ( or ), the original image, sigma, and threshold. The output is the visualized segmentation result. This function does not call any other functions.