Background.m

I1 = mat2gray(table2array(tdjgrbp51um1hr3rd020717(1:512,1:512)));

bw1 = imopen(I1,strel('rectangle',[10,512]));

I1\_flat = I1-bw1;

m0 = min(min(I1\_flat));

m1 = max(max(I1\_flat));

I1\_flat\_rescaled = mat2gray(I1\_flat,[m0 m1]);

E1 = edge(I1\_flat\_rescaled,'Canny');

figure

imshow(I1\_flat\_rescaled);

figure

imshow(E1);

figure, imshow(I1);

segmentation.m

imageSize = size(I1\_flat\_rescaled);

numRows = imageSize(1);

numCols = imageSize(2);

wavelengthMin = 4/sqrt(2);

wavelengthMax = hypot(numRows,numCols);

n = floor(log2(wavelengthMax/wavelengthMin));

wavelength = 2.^(0:(n-2)) \* wavelengthMin;

deltaTheta = 45;

orientation = 0:deltaTheta:(180-deltaTheta);

g = gabor(wavelength,orientation);

gabormag = imgaborfilt(I1\_flat\_rescaled,g);

for i = 1:length(g)

sigma = 0.5\*g(i).Wavelength;

K = 1.5;

gabormag(:,:,i) = imgaussfilt(gabormag(:,:,i),K\*sigma);

end

X = 1:numCols;

Y = 1:numRows;

[X,Y] = meshgrid(X,Y);

featureSet = cat(3,gabormag,X);

featureSet = cat(3,featureSet,Y);

numPoints = numRows\*numCols;

X = reshape(featureSet,numRows\*numCols,[]);

X = bsxfun(@minus, X, mean(X));

X = bsxfun(@rdivide,X,std(X));

coeff = pca(X);

feature2DImage = reshape(X\*coeff(:,1),numRows,numCols);

figure

imshow(feature2DImage,[])

kmeans.m

imshowpair(I1\_flat\_rescaled,feature2DImage,'montage');

L1 = kmeans(X,2,'Replicates',5);

L1 = reshape(L1,[numRows,numCols]);

figure

imshow(label2rgb(L1));

I1seg1 = zeros(size(I1\_flat\_rescaled),'like',I1\_flat\_rescaled);

I1seg2 = zeros(size(I1\_flat\_rescaled),'like',I1\_flat\_rescaled);

BW = L1 == 1;

BW = repmat(BW,[1 1 1]);

I1seg1(BW) = I1\_flat\_rescaled(BW);

I1seg2(~BW) = I1\_flat\_rescaled(~BW);

figure

imshowpair(I1seg2,I1seg1,'montage');

parameters.m

e2 = edge(I1seg2,'Canny');

figure

imshow(e2);

i1bw = imbinarize(I1seg2);

figure

imshow(i1bw);

e3 = edge(I1seg1,'Canny');

figure

imshow(e3);

i1bw1 = imbinarize(I1seg1);

figure

imshow(i1bw1);

n\_or = sum(sum(i1bw1));

n\_dis = sum(sum(i1bw));

order\_disorder\_ratio = n\_or/n\_dis;

pep = n\_or + n\_dis;

per\_coverage\_pep = 100\*pep/(512\*512);

per\_o\_cov = 100-100\*n\_dis/(512\*512);