function [BW,maskedImage] = segmentImage(X)

gaborX = createGaborFeatures(X);

% Threshold image - adaptive threshold

BW = imbinarize(X, 'adaptive', 'Sensitivity', 0.520000, 'ForegroundPolarity', 'dark');

% Active contour with texture

iterations = 8;

BW = activecontour(gaborX, BW, iterations, 'Chan-Vese');

% Create masked image.

maskedImage = X;

maskedImage(~BW) = 0;

end

function gaborFeatures = createGaborFeatures(im)

if size(im,3) == 3

im = prepLab(im);

end

im = im2single(im);

imageSize = size(im);

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(im(:,:,1),g);

for i = 1:length(g)

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

K = 3;

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

end

% Increases liklihood that neighboring pixels/subregions are segmented together

X = 1:numCols;

Y = 1:numRows;

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

featureSet = cat(3,gabormag,X);

featureSet = cat(3,featureSet,Y);

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

% Normalize feature set

featureSet = featureSet - mean(featureSet);

featureSet = featureSet ./ std(featureSet);

gaborFeatures = reshape(featureSet,[numRows,numCols,size(featureSet,2)]);

% Add color/intensity into feature set

gaborFeatures = cat(3,gaborFeatures,im);

end

function out = prepLab(in)

% Convert L\*a\*b\* image to range [0,1]

out = in;

out(:,:,1) = in(:,:,1) / 100; % L range is [0 100].

out(:,:,2) = (in(:,:,2) + 86.1827) / 184.4170; % a\* range is [-86.1827,98.2343].

out(:,:,3) = (in(:,:,3) + 107.8602) / 202.3382; % b\* range is [-107.8602,94.4780].

end