Assignment A7: Texture

CS 5320/6320
Spring 2016

Assigned: 28 Fubruary 2016

Due: 16 March 2016

For this problem, handin a lab report A7.pdf (include name, date, assignment and class number in pdf) which develops and studies texture analysis. Note that you may need to break up the image into smaller parts to get OK results (e.g., 128x128 sum-windows that have 4 textures).

Implement Algorithm 6.3 from the text and test it for texture classification on the mixed texture image (im_tex which can be found in the A7_tex.mat file in data/A7).

Develop the Matlab functions implementing (1) CS5320_spot1, (2) CS5320_spot2, (3) CS5320_bar, (4) CS5320_texture_parameters, and (5) CS5320_k_means_texture. I have provided the CS5320_oriented_Gaussian function that is used. You should handin the report

A7.pdf as well as the source code developed in the study. The code should conform to the style requested in the class materials.

In addition, please turn in a hardcopy of the report in class before the start of class on March 16, 2016.

Write a lab report in the format (please do not deviate from this format!) described in the course materials.

```
function S1 = CS5320_spot1
% CS5320_spot1 - texture feature: spot 1
% On input:
%     N/A
```

```
% On output:
%     S1 (11x11 array): spot filter
% Call:
%     S1 = CS5320_spot1
% Method:
%   uses combination of 3 symmetric Gaussian filters:
%   S1 = G1 - 2*G2 + G3
%   where
%     G1 is Gaussian with sigma = 0.62
%     G2 is Gaussian with sigma = 0.1
%     G3 is Gaussian with sigma = 1.6
% Author:
%     <Your Name>
%     UU
%     Spring 2016
%

function S2 = CS5320_spot2
% CS5320_spot2 - texture feature: spot 2
% On input:
%     N/A
% On output:
%     S2 (11x11 array): spot filter
% Call:
%     S2 = CS5320_spot2
% Method:
%   uses combination of 2 symmetric Gaussian filters:
%   S1 = G1 - G2
%   where
%     G1 is Gaussian with sigma = 0.71
%     G2 is Gaussian with sigma = 1.14
% Author:
%     <Your Name>
%     UU
%     Spring 2016
%

function B = CS5320_bar(a,b,c,d)
% CS5320_bar - texture feature: bar
% On input:
%     a (float): see below
```

```
%     b (float): see below
%     c (float): see below
%     d (float): see below
%  These parameters appear in the equation:
%  G_{s1,s2}(x,y) = exp(-(ax+by)^2/(2s1^2)-(cx+dy)^2/(2s2^2)
%    with s1 and s2 the sigmas in x and y, respectively
% On output:
%     B (11x11 array): spot filter
% Call:
%     B = CS5320_bar(1,0,0,-1);
% Method:
%   uses combination of 3 oriented Gaussian filters:
%   B = -G1 + 2G2 - G3
%   where
%     G1: oriented Gaussian from:
%     a,b,c,d,2,1,0,1,xmin,xmax,ymin,ymax,0.1
%     G2: oriented Gaussian from:
%     a,b,c,d,2,1,0,0,xmin,xmax,ymin,ymax,0.1
%     G3: oriented Gaussian from:
%     a,b,c,d,2,1,0,-1,xmin,xmax,ymin,ymax,0.1
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function params = CS5320_texture_params(im)
% CS5320_texture_params - compute texture parameters
% On input:
%     im (mxnx3 array): input image
% On output:
%     params (mxnx16 array): texture parameter image
%      channel  1: spot1 summarized positive
%      channel  2: spot1 summarized negative
%      channel  3: spot2 summarized positive
%      channel  4: spot2 summarized negative
%      channel  5: bar (0) summarized positive
%      channel  6: bar (0) summarized negative
%      channel  7: bar (45) summarized positive
%      channel  8: bar (45) summarized negative
%      channel  9: bar (90) summarized positive
```

```
%      channel 10: bar (90) summarized negative
%      channel 11: bar (135) summarized positive
%      channel 12: bar (135) summarized negative
%      channel 13: mean summarized positive
%      channel 14: mean summarized negative
%      channel 15: variance summarized positive
%      channel 16: variance summarized negative
% Call:
%     p_im = CS5320_texture_params(im_tex);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function [clusters,elements] = CS5320_k_means_texture(params,k)
% CS5320_k_means_texture - classifies textures into k clusters
% On input:
%     im (mxnxp array): texture parameter images (p mxn images)
%     k (int): number of clusters desired
% On output:
%     clusters (kxp array): k cluster center vectors
%     elements (mxn array): cluster index image
%        elements(r,c) is cluster number
% Call:
%     [cl,el] = CS5320_k_means_texture(p_im,20);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%
```