Assignment A2: Spatial Filtering

CS 6640
Fall 2018

Assigned: 5 September 2018

Due: 19 September 2018

For this problem, handin a report (PDF file) as required below as well as the Matlab .m files for the functions described by the headers below, and any help functions you write.

None of the functions should write to the interpreter, draw, etc. unless explicitly required by the header.

Some notes:

- Indent headers correctly (5 spaces indented lines)
- Do not exceed 72 characters per source line
- Handle borders of the image by using the 'same' option where possible, or using whatever elements in the image overlap with the filter.
- CS4640_Hessian: definitely use the gradient function

1. Study texture analysis by developing the function CS6640_Laws, and then applying it to our videos using the Matlab kmeans function. Discuss whether the texture recognition algorithm given here works well:

```
 0. Suppose the 1x7 filter vectors are:
    L7 = [1 6 15 20 15 6 1]
    E7 = [-1 -4 -5 0 5 4 1]
    S7 = [-1 -2 1 4 1 -2 -1]
```

```
W7 = [-1 0 3 0 -3 0 1]
R7 = [1 -2 -1 4 -1 -2 1]
O7 = [-1 6 -15 20 -15 6 -1]
```

1. Obtain 9 Laws 7x7 texture filters from products of the 1x7 filters as described below, as well as a mean and variance filter (based on 7x7 neighborhood).

   Filters (formed as outer product of two 1x7 vectors):
   - i. L7,L7
   - ii. L7,E7
   - iii. L7,S7
   - iv. L7,W7
   - v. L7,R7
   - vi. L7,O7
   - vii. E7,E7
   - viii. W7,R7
   - ix. W7, O7
   - x. mean

2. Apply (convolve) each filter, f, to the image, im (f*im; where * denotes convolution).

3. For each filter response, f*im, compute
   S = 7x7 mean filter applied to absolute value of f*im
5. Create a set of texture vectors (size is M*Nx10) from the S vectors.
6. Use kmeans to find clusters and see if they represent anything meaningful.

```
function T = CS6640_Laws(im)
% CS6640_Laws - compute texture parameters
% On input:
%     im (MxN array): input image
% On output:
%     T (M*Nx10 array): texture parameters
%        each texture parameter is a column vector in T
% Call:
%     T = CS6640_Laws(im);
% Author:
%     <Your name>
%     UU
```

```
%     Fall 2018
%
```

2. Study image registration using the quadratic approach and rigid transform approaches, as well as the Harris operator to find corresponding points in two images. Figure out whether image registration is necessary in our videos.

```
function H_im = CS6640_Harris(im)
% CS6640_Harris - compute Harris operator at each pixel
% On input:
%     im (MxN array): graylevel image
% On output:
%     H_im (MxN array): Harris value (normalized)
% Call:
%     H = CS6640_Harris(im);
% Author:
%     <Your name>
%     UU
%     Fall 2018
%
```

```
function [imr,q,A] = CS6640_register(im,s,cpts)
% CS6640_register - produce registered image and
%    transform
% On input:
%     im (MxN array): input image
%     s (int): transform switch: if 1, then quadratic,
%        else affine
%     cpts (2kx2 array): k corresponding points (evens are one set;
%        odds the other)
% On output:
%     imr (MxN array): registered image
%     q (1x12 vector): quadratic coefficients
%     A (3x3 array): affinetransform
% Call:
%     [imr,q,A] = CS6640_register(im,1);
% Author:
%     <Your name>
```

```
%     UU
%     Fall 2018
%
```

```
function [xp,yp] = CS6640_transform2D(x,y,q)
% CS6640_transform2D - produce transformed points
%   from given points and quadratic coeffients
% On input:
%     x (float): x value
%     y (float): y value
%     q (1x12 vector): quadratic coefficients
% On output:
%     xp (float): x value of transformed point
%     yp (float): y value of transformed point
% Call:
%     [xp,yp] = CS6640_transform2D(3,3,q);
% Author:
%     <Your name>
%     UU
%     Fall 2018
%
```

3. Develop a temporal differential operator that allows motion tracking. Try it out on our videos and discuss the results.

```
function [M,tracks] = CS6640_track(video)
% CS6640_track - track motion in a video
% On input:
%     video (video structure): input video
% On output:
%     M (movie structure): movie of detected differences
%         in video
%     tracks (kx2 array): row,col center of mass of largest
%         moving object in sequential video frames
% Call:
%     [M,tr] = CS6640_track(video);
% Author:
%     <Your name>
```

```
%      UU
%      Fall 2018
%
```