160010031_160010011_160010058_assignmentSegmentation

by Krishna Wadhwani

Submission date: 03-Sep-2018 12:02AM (UTC+0800)

Submission ID: 995548757

File name: code.txt (3.88K)

Word count: 428

Character count: 3303

```
function myHarrisCornerDetector(input)

M = size(input,1);

N = size(input,2);

G = fspecial('gaussian', 5 , 1);

input = imfilter(input, G);

image = myLinearContrastStretching(input/255);

dx = [-1 0 1; -1 0 1; -1 0 1];

dy = dx';

Ix = imfilter(image, dx);

Iy = imfilter(image, dy);

Image_eigen_1 = zeros(M,N);

Image_eigen_2 = zeros(M,N);

G = fspecial('gaussian', 8 , 1.5);
```

```
Ix2 = imfilter(Ix.*Ix, G);

Iy2 = imfilter(Iy.*Iy, G);

Ixy = imfilter(Ix.*Iy, G);

determinant_A = Ix2.*Iy2 - Ixy.*Ixy;

trace_A = Ix2 + Iy2;

k = 0.06;

threshold = 0.04;

cornerness_matrix = determinant_A - k*(trace_A.*trace_A);

cornerness = (cornerness_matrix>threshold)*255;

for i = 1:M

    for j = 1:N

        a11 = Ix2(i,j);

        a22 = Iy2(i,j);
    a21 = Ixy(i,j);
    A = [a11 a21; a21 a22];
    eigen_values = eig(A);
    Image_eigen_1(i,j) = max(eigen_values);
    Image_eigen_2(i,j) = min(eigen_values);
  end
end
myNumOfColors = 200;
myColorScale = [ [0:1/(myNumOfColors-1):1]' , [0:1/(myNumOfColors-1):1]' ,
[0:1/(myNumOfColors-1):1]' ];
figure('name', 'Original Image after gaussian smoothing and intensity rescaling')
imagesc(image);
colormap (myColorScale);
colormap gray;

daspect ([1 1 1]);

axis tight;

colorbar

title('Original Image after gaussian smoothing and intensity rescaling')

figure('name', 'derivative images')

subplot(1,2,1)

imshow(Ix);

colormap (myColorScale);

colormap gray;

daspect ([1 1 1]);

axis on;

colorbar
```

```
title('Derivative along X')
subplot(1,2,2)
imshow(Iy);
colormap (myColorScale);
colormap gray;
daspect ([1 1 1]);
axis on;
colorbar
title('Derivative along Y')
figure('name', 'Eigen Value Images')
subplot(1,2,1)
imagesc(Image_eigen_1);
colormap (myColorScale);
colormap gray;
daspect ([1 1 1]);
axis tight;
colorbar
title('Larger Eigen Values')
subplot(1,2,2)
imagesc(Image_eigen_2);
colormap (myColorScale);
colormap gray;
daspect ([1 1 1]);
axis tight;
colorbar
title('Smaller Eigen Values')
```

```
figure('name', 'cornerness measure')

imagesc(cornerness);

colormap (myColorScale);

colormap gray;

daspect ([1 1 1]);

axis tight;

colorbar

title('Cornerness Measure')

end

function newImage = myMeanShiftSegmentation(d, hs, hr, k, numberOfIterations)

I = imread('../data/baboonColor.png');

smoothImage = imgaussfilt(I,1);

image = smoothImage(1:d:end,1:d:end,:);
image = im2double(image);

[rows,cols,~] = size(image);

%constructing the 5D feature-space for points on the image

featureSpace = [];

for i=1:rows

    for j=1:cols

        pixel = image(i,j,:);

        featureSpace = [featureSpace; [i/rows, j/cols, pixel(1), pixel(2), pixel(3)]];

    end

end

featureSpace(:,1:2) = featureSpace(:,1:2) / (hs);

featureSpace(:,3:5) = featureSpace(:,3:5) / (hr);

newImage = ones(size(image));
```

```
for i=1:rows
    for j=1:cols
        pixel = image(i,j,:);
        featureSpaceVector = [i/rows, j/cols, pixel(1), pixel(2), pixel(3)];
        convergedFeatureVector = meanShiftProcedure(k, featureSpace,
featureSpaceVector, numberOfIterations);
        newImage(i,j,:) = convergedFeatureVector(3:5);
%         disp(strcat(string(i) ,',', string(j)));
    end
end
myNumOfColors = 200;
myColorScale = [ [0:1/(myNumOfColors-1):1]' , [0:1/(myNumOfColors-1):1]' ,
[0:1/(myNumOfColors-1):1]' ];
```

```
figure('name', 'input image')
subplot(2,1,1)
imagesc(I);
colormap (myColorScale);
colormap jet;
daspect ([1 1 1]);
axis tight;
colorbar
title('Input Image')
subplot(2,1,2)
imagesc(newImage);
colormap (myColorScale);
colormap jet;
```

```
daspect ([1 1 1]);

axis tight;

colorbar

title('Segmented Image')

end

function convergedFeatureVector = meanShiftProcedure(k, featureSpace,

featureSpaceVector, numberOfIterations)

    for i=1:numberOfIterations

        [Xid, D] = knnsearch(featureSpace, featureSpaceVector, 'k', k);

        weights = exp(-times(D,D))';

        estimate = sum(times(featureSpace(Xid,:), weights)) / sum(weights);

        featureSpaceVector = estimate;

    end
    convergedFeatureVector = estimate;


end
```

160010031_160010011_160010058_assignmentSegmentation

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12