160050030_160050031_160050033_Assignment3

by Vikrant Garg

Submission date: 02-Sep-2018 09:41PM (UTC+0800)
Submission ID: 995937417
File name: report.txt (7.44K)
Word count: 988
Character count: 6054

```
function imd = sobelOp(im,dir)

%UNTITLED3 Summary of this function goes here

%   Detailed explanation goes here

%Sobel matrix

if dir == 'x'; sobelMat = [-1 0 1;-2 0 2;-1 0 1]; end

if dir == 'y'; sobelMat = [1 2 1; 0 0 0; -1 -2 -1]; end

%padding image with the same values at boundary

imt = cat(2,im(:,1),im);

imt = cat(2,imt(:,end),imt);

imt = cat(1,imt(1,:),imt);

imt = cat(1,imt(end,:),imt);

[row,col] = size(im);

imd = zeros(row,col);
```

```
for i = 1:row

    for j = 1:col

        m1 = imt(i:i+2,j:j+2).*sobelMat;

        imd(i,j) = abs(sum(sum(m1)));

    end

end

end

function imo = gaussianSmooth(im,ss)

%UNTITLED2 Summary of this function goes here

%return gaussian smoothed image

%   Detailed explanation goes here

%size

[row,col] = size(im);
```

```
imo = zeros(row,col);

for i = 1:row

  for j = 1:col

    l = floor(j - 3*ss);

    if (l < 1);  l = 1; end

    r = floor(j + 3*ss);

    if (r > col);  r = col; end

    t = floor(i - 3*ss);

    if (t < 1);  t = 1; end

    b = floor(i + 3*ss);

    if (b > row);  b = row; end

    X = im(t:b,l:r);

    sp_r = 1:b-t+1 ;
```

```
        sp_r = sp_r';
        sp_r = repmat(sp_r,1,r-l+1);
        sp_c = 1:r-l+1;
        sp_c = repmat(sp_c,b-t+1,1);
        sp_r = sp_r - (i - t + 1);
        sp_c = sp_c - (j - l + 1);
        sp_r = sp_r.*sp_r;
        sp_c = sp_c.*sp_c;
        sp = sp_r + sp_c;
        sp = exp((-0.5/ss^2)*sp);
        imo(i,j) = sum(sum(sp.*X))/sum(sum(sp));
    end
end
```

```
end

%% MyMainScript

tic;

%% Your code here

load('../data/boat.mat')

imageOrig = imageOrig/256;

myHarrisCornerDetector(imageOrig,2,2,0.05);

toc;

function [] = myHarrisCornerDetector(im,s1,s2,k)

%UNTITLED Summary of this function goes here

%   Detailed explanation goes here

%%

%%%%%%%%%%%%%%Testing Block%%%%%%%%%%%%%%%%%%%
```

```
% load('../data/boat.mat')

% im = imageOrig/256;

% s1 = 2;

% s2 = 2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%size

[row,col] = size(im);

img1 = gaussianSmooth(im,s1); %smoothing 1

%X derivative

imgx = sobelOp(img1,'x');

% Y derivative

imgy = sobelOp(img1,'y');

imgx2 = imgx.*imgx;
```

```
imgy2 = imgy.*imgy;

imgxy = imgx.*imgy;

A11 = zeros(row,col);

A12 = zeros(row,col);

A22 = zeros(row,col);

for i = 1:row

    for j = 1:col

        l = floor(j - 3*s2);

        if (l < 1);  l = 1; end

        r = floor(j + 3*s2);

        if (r > col);  r = col; end

        t = floor(i - 3*s2);

        if (t < 1);  t = 1; end
```

```
b = floor(i + 3*s2);

if (b > row);  b = row; end

A11t = imgx2(t:b,l:r);

A12t = imgxy(t:b,l:r);

A22t = imgy2(t:b,l:r);

sp_r = 1:b-t+1 ;

sp_r = sp_r';

sp_r = repmat(sp_r,1,r-l+1);

sp_c = 1:r-l+1;

sp_c = repmat(sp_c,b-t+1,1);

sp_r = sp_r - (i - t + 1);

sp_c = sp_c - (j - l + 1);

sp_r = sp_r.*sp_r;
```

```
        sp_c = sp_c.*sp_c;

        sp = sp_r + sp_c;

        sp = exp((-0.5/s2^2)*sp);

        sp_sum = sum(sum(sp));

        A11(i,j) = sum(sum(A11t.*sp))/sp_sum;

        A12(i,j) = sum(sum(A12t.*sp))/sp_sum;

        A22(i,j) = sum(sum(A22t.*sp))/sp_sum;

    end

end

%% Harris corner-ness measure

% k = 0.05;

trace = A11+A22;

C = A11.*A22 - A12.^2 - k*(trace.^2);
```

```
C = C/max(max(C));

%% eigenvalues

eigen1 = (A11+A22 + sqrt((A11-A22).^2 + 4*A12.^2))/2;

eigen2 = (A11+A22 - sqrt((A11-A22).^2 + 4*A12.^2))/2;

%figures

myNumOfColors=200;

myColorScale = [(0:1/(myNumOfColors-1):1)',(0:1/(myNumOfColors-
1):1)',(0:1/(myNumOfColors-1):1)'];

figure(1)

imshow(mat2gray(imgx))

colormap(myColorScale);

colormap gray;

figure(2)
```

```
imshow(mat2gray(imgy))

colormap(myColorScale);

colormap gray;

figure(3)

imshow(mat2gray(eigen1))

colormap(myColorScale);

colormap gray;

colorbar

figure(4)

imshow(mat2gray(eigen2))

colormap(myColorScale);

colormap gray;

colorbar
```

```
%%

figure(5)

imshow(C)

colormap(myColorScale);

colormap gray;

colorbar

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [] = myMeanShiftSegmentation(filename,hr,hs,nitr,nnbr)

%UNTITLED4 Summary of this function goes here

%   Detailed explanation goes here

%%%%%%%%Test block%%%%%%%%%%%%%
```

```
% filename = '../data/baboonColor.png';

%  hr = 250;

%  hs = 100;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% read image

im= imread(filename);

% im=mat2gray(im);

% filter image

im = imgaussfilt(im,1);

% resize image

im=imresize(im,.5);

[row,col,~] = size(im);

% upImg = zeros(row,col,3);
```

```
% selecting n nearest neighbours
% n = 75;
% gradient steps
% tau = 3;
wait = waitbar(0,'Wait');
%%%
feature=zeros(5,row*col);
for i = 1:row
    for j=1:col
        feature(:,i*col - col +j)=[i;j;im(i,j,1); im(i,j, 2); im(i,j,3);];
    end
end
X=feature';
```

```
X = cast(X,'double');

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

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

for z = 1:nitr

%     X(:, 3:5) = X(:, 3:5)/(hr);

%     X(:, 1:2) = X(:, 1:2)/(hs);

    Mdl = KDTreeSearcher(X);

    [idx,D] = knnsearch(Mdl,X,'K',nnbr); % finding indexes of k nearest nbrs to (i,j) pixel by
color

    res = exp(-(D.^2));

    suma=sum(res,2);

%     X(:, 3:5) = X(:, 3:5)*(hr);

%     X(:, 1:2) = X(:, 1:2)*(hs);
```

```
    X(:,3) = sum(res.*reshape(X(idx(:,:),3),[],nnbr),2)./suma;

    X(:,4) = sum(res.*reshape(X(idx(:,:),4),[],nnbr),2)./suma;

    X(:,5) = sum(res.*reshape(X(idx(:,:),5),[],nnbr),2)./suma;

    waitbar(z/nitr,wait,'Chill bro, we are still computing..');

end

close(wait)

final_im = zeros(row,col,3);

for i = 1:row

    for j = 1:col

        final_im(i,j,:) = X(i*col-col+j,3:5);

    end

end

%%
```

```
%figures

myNumOfColors=200;

myColorScale = [(0:1/(myNumOfColors-1):1)',(0:1/(myNumOfColors-
1):1)',(0:1/(myNumOfColors-1):1)'];

subplot(1,2,1);

imshow(imread(filename));

colormap(myColorScale);

subplot(1,2,2);

imshow(final_im);

title(['hr=' num2str(hr) 'hs=' num2str(hs)] )

colormap(myColorScale);

end

%% MyMainScript
```

```
tic;

%% Your code here

% myMeanShiftSegmentation(filename, hr, hs, number_of_iteration,
number_of_neighbour);

figure(1)

myMeanShiftSegmentation('../data/baboonColor.png', 250, 100, 20, 100);

% figure(2)

% myMeanShiftSegmentation('../data/baboonColor.png', 500, 100, 20, 100);

% figure(3)

% myMeanShiftSegmentation('../data/baboonColor.png', 250, 200, 20, 100);

% figure(4)

% myMeanShiftSegmentation('../data/baboonColor.png', 250, 100, 40, 100);

% figure(5)
```

% myMeanShiftSegmentation('../data/baboonColor.png', 250, 100, 20, 500);

%%

toc;

Assignment 3 Question 2

Files Included--

myMainScript.m

myMeanShiftSegmentation.m

Resized image to 256*256

Parameters- hs=100; hr=250; number of iteration =20; number of neighbor =100

Output:

Justification for optimum values:

hs: On decreasing its value we saw that segments were not formed clearly i.e. image

was just

smoothed and no clear cut segments were visible and on increasing its value we saw

that too

many segments were formed

hr: On increasing its value we saw that colors were mixing hence merging unnecessary

amounts of segments. And on decreasing its value we saw that colors were not properly

mixed

i.e. there were many local maximas(converging points were increased) hence many

segments

were shown.

ASSIGNMENT 3 QUESTION 1

Files included --

gaussianSmooth.m

myMainScript.m

myHarrisCornerDetector.m

sobelOp.m

Parameters - gaussian parameter 1 (s1) = 2 gaussian parameter 2 (s2) = 2

Cornerness measure (k) = 0.05

OUTPUT -

For boat.mat

Figure 1 - X derivativeFigure2 Y derivativeFigure 3 --- Eigenvalue 1Figure 4 -- Eigenvalue 2Figure 5 -- Cornerness - Measure

Description

White spots in figure 5 are corresponding to corners in original images. K = 0.05 is optimal

value. For k = 0.1 and 0.2 corners are missing out. And for K around 0.01 white spots

corresponding to edges also appear.

160050030_160050031_160050033_Assignment3

ORIGINALITY REPORT

25%	14%	16%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	amaliepews.blogg.no Internet Source	4%
2	Submitted to Indian Institute of Technology, Bombay Student Paper	4%
3	Submitted to University of Southern California Student Paper	4%
4	ebalda.blogspot.com Internet Source	3%
5	"Applications of Quantum Mechanics", Quantum Physics, 2007 Publication	2%
6	Submitted to Associatie K.U.Leuven Student Paper	2%
7	Tom Lyche, Jean-Louis Merrien. "Exercises in Computational Mathematics with MATLAB", Springer Nature America, Inc, 2014 Publication	2%

de.mathworks.com

8	Internet Source	2%
9	Berndt, Jürgen. "Totally Geodesic Submanifolds of Riemannian Symmetric Spaces", Springer Proceedings in Mathematics & Statistics, 2014. Publication	1%
10	Nourhan El Beheiry, Mohamed El Sharkawy, Mona Lotfy, Said Elnoubi. "Modifications to fast and efficient spatial error concealment technique for block-based video coding systems", 2009 52nd IEEE International Midwest Symposium on Circuits and Systems, 2009 Publication	<1%
11	Shengrong Gong, Chunping Liu, Yi Ji, Baojiang Zhong, Yonggang Li, Husheng Dong. "Advanced Image and Video Processing Using MATLAB", Springer Nature America, Inc, 2019 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

160050030_160050031_160050033_Assignment3

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21