check1

by Gurparkash Singh

Submission date: 21-Aug-2018 02:26AM (UTC+0800)
Submission ID: 991587310
File name: turnitin.txt (8.19K)
Word count: 814
Character count: 6887

```
function img_out = myLinearContrastStretching(img_mat)

    [~,~,d] = size(img_mat);

    for k = 1:d

        max_n = max(max(img_mat(:,:,k)));

        min_n = min(min(img_mat(:,:,k)));

        img_out(:,:,k) = (255/(max_n - min_n))*(img_mat(:,:,k) - min_n);

    end

end

function [imgOutput] = myUnsharpMasking(img_in, scale, radius)

    img = load(img_in);

    imgIn = img.imageOrig;

    imgInput = mat2gray(myLinearContrastStretching(mat2gray(imgIn)));

    gaussian = fspecial('gaussian', [5,5], radius);
```

```
    img_blurred = imfilter(imgIn, gaussian, 'conv');

    output_mat = imgIn + scale*(imgIn - img_blurred);

    imgOutput = mat2gray(myLinearContrastStretching(mat2gray(output_mat)));

    figure;

    subplot(1,2,1);

    imshow(imgInput), colorbar;

    title('Original Image');

    subplot(1,2,2);

    imshow(imgOutput), colorbar;

    title('Sharpened Image');

end

%% MyMainScript

tic;
```

```
myUnsharpMasking('../data/lionCrop.mat', 2, 2);

myUnsharpMasking('../data/superMoonCrop.mat', 0.1, 1);

toc;function [img_out, rmsd] = myBilateralFiltering(img, img_noisy, sigma_space,

sigma_intensity)

%     parameters

    k_size = 11;

%     sigma_space = 5.0;

%     sigma_intensity = 10.0;

    [h, w] = size(img);

    space_filter = fspecial('gaussian', k_size, sigma_space);

    img_out = img_noisy;

    sz = ceil((k_size-1)/2);

    for i=1:h
```

```
for j=1:w

    i_min = max(1, i-sz);

    i_max = min(h, i+sz);

    j_min = max(1, j-sz);

    j_max = min(w, j+sz);

    il = i_min - i + sz + 1;

    jl = j_min - j + sz + 1;

    ir = 2*sz + 1 - ((i+sz)-(i_max));

    jr = 2*sz + 1 - ((j+sz)-(j_max));

    im_part = img_noisy(i_min:i_max,j_min:j_max);

    int_filter = normpdf(im_part,img_noisy(i,j),sigma_intensity);

    filter = int_filter .* (space_filter(il:ir, jl:jr));

    filter = filter ./ sum(sum(filter));
```

```
            img_out(i,j) = sum(sum(filter .* im_part));
        end
    end
%     figure, imshow(mat2gray(img_out));
%     figure, imshow(mat2gray(img_noisy));
%     rmsd calculation
%     corrected image
    rmsd = (img - img_out);
    rmsd = sqrt((sum(sum(rmsd.^2)))/(h*w));
end
%% MyMainScript
tic;
%% Your code here
```

```
space = [1.0, 0.9, 1.1, 1.0, 1.0];

inten = [1.0, 1.0, 0.9, 0.9, 1.1];

%  barbara

img_name = '../data/barbara.mat';

image_struct = load(img_name);

img = image_struct.imageOrig;

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

img_noisy = img+noise;

sigma_space = 1.45;

sigma_inten = 9.9;

% iptsetpref('ImshowAxesVisible','on');

% figure;
```

```
% imshow(mat2gray(img)), colorbar;

% title('Input Image');

% figure;

% imshow(mat2gray(img_noisy)), colorbar;

% title('Noisy Image');

disp('Barbara');

for i=1:5

    s_sd = space(1,i)*sigma_space;

    i_sd = inten(1,i)*sigma_inten;

    [img_out, rmsd] = myBilateralFiltering(img, img_noisy, s_sd, i_sd);

%     figure;

%     imshow(mat2gray(img_out)), colorbar;

%     title(strcat(['Filtered Image, s-sd*', num2str(space(1,i)), ' i-id*', num2str(inten(1,i))]));
```

```
    disp(strcat([num2str(space(1,i)), ' ', num2str(inten(1,i)), ' ' , num2str(rmsd)]));

    t = strcat(['barbara_', num2str(space(1,i)), '_', num2str(inten(1,i)),'.png']);

    imwrite(mat2gray(img_out),t);

end

% grass

img = im2double(imread('../data/grass.png'));

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

img_noisy = img+noise;

% iptsetpref('ImshowAxesVisible','on');

% figure;

% imshow(mat2gray(img)), colorbar;

% title('Input Image');
```

```
% figure;

% imshow(mat2gray(img_noisy)), colorbar;

% title('Noisy Image');

sigma_space = 0.8;

sigma_inten = 0.165;

disp('Grass');

for i=1:5

    s_sd = space(1,i)*sigma_space;

    i_sd = inten(1,i)*sigma_inten;

    [img_out, rmsd] = myBilateralFiltering(img, img_noisy, s_sd, i_sd);

%     figure;

%     imshow(mat2gray(img_out)), colorbar;

%     title(strcat(['Filtered Image, s-sd*', num2str(space(1,i)), ' i-id*', num2str(inten(1,i))]));
```

```
    disp(strcat([num2str(space(1,i)), ' ', num2str(inten(1,i)), ' ' , num2str(rmsd)]));

    t = strcat(['grass_', num2str(space(1,i)), '_', num2str(inten(1,i)),'.png']);

    imwrite(mat2gray(img_out),t);

end

% % honeycomb

img = im2double(imread('../data/honeyCombReal.png'));

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

img_noisy = img+noise;

% iptsetpref('ImshowAxesVisible','on');

% figure;

% imshow(mat2gray(img)), colorbar;

% title('Input Image');
```

```
% figure;

% imshow(mat2gray(img_noisy)), colorbar;

% title('Noisy Image');

sigma_space = 1.05;

sigma_inten = 0.14;

disp('HoneyComb');

for i=1:5

    s_sd = space(1,i)*sigma_space;

    i_sd = inten(1,i)*sigma_inten;

    [img_out, rmsd] = myBilateralFiltering(img, img_noisy, s_sd, i_sd);

%     figure;

%     imshow(mat2gray(img_out)), colorbar;

%     title(strcat(['Filtered Image, s-sd*', num2str(space(1,i)), ' i-id*', num2str(inten(1,i))]));
```

```
    disp(strcat([num2str(space(1,i)), ' ', num2str(inten(1,i)), ' ' , num2str(rmsd)]));

    t = strcat(['honeycomb_', num2str(space(1,i)), '_', num2str(inten(1,i)),'.png']);

    imwrite(mat2gray(img_out),t);

end

toc;

%% MyMainScript

tic;

%% Your code here

% Converting intensity range of both images to [0,1].

% img = im2double(imread('../data/grass.png'));

% imgStruct = load('../data/grassNoisy.mat');

% img_noisy = im2double(imgStruct.imgCorrupt);

% imgStruct = load('../data/barbara.mat');
```

```
% img = imgStruct.imageOrig;

% img = im2double(imread('../data/honeyCombReal.png'));

% [h,w] = size(img);

% noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

% img = im2double(img);

% img_noisy = im2double(img+noise);

g_filter = fspecial('gaussian', 9, 2);

colormap gray;

imagesc(g_filter);

mults = [1.0, 0.9, 1.1];

%  barbara

img_name = '../data/barbara.mat';

image_struct = load(img_name);
```

```
img = image_struct.imageOrig;

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

img = im2double(img);

img_noisy = im2double(img+noise);

figure;

colormap gray;

imagesc(img);

title('barbara normal');

figure;

colormap gray;

imagesc(img_noisy);

title('barbara noisy');
```

```
sigma = 87;

disp('Barbara');

for i=1:3

    sd = mults(1,i)*sigma;

    [img_out, rmsd] = myPatchBasedFiltering(img,img_noisy,sd);

    disp(strcat([num2str(mults(1,i)), ' ', num2str(rmsd)]));

    t = strcat(['barbara_', num2str(mults(1,i)),'.png']);

    imwrite(mat2gray(img_out),t);

end

% %  grass

img = im2double(imread('../data/grass.png'));

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;
```

```
img_noisy = img+noise;

figure;

colormap gray;

imagesc(img);

title('grass normal');

figure;

colormap gray;

imagesc(img_noisy);

title('grass noisy');

sigma = 0.84;

disp('Grass');

for i=1:1

    sd = mults(1,i)*sigma;
```

```
    [img_out, rmsd] = myPatchBasedFiltering(img,img_noisy,sd);

    disp(strcat([num2str(mults(1,i)), ' ', num2str(rmsd)]));

    t = strcat(['grass_', num2str(mults(1,i)),'.png']);

    imwrite(mat2gray(img_out),t);

end

%

%  honeycomb

img = im2double(imread('../data/honeyCombReal.png'));

[h, w] = size(img);

noise = randn(h,w)*(max(max(img))-min(min(img)))*0.05;

img_noisy = img+noise;

figure;

colormap gray;
```

```
imagesc(img);

title('honeyComb normal');

figure;

colormap gray;

imagesc(img_noisy);

title('honeyComb noisy');

sigma = 0.90;

disp('Honeycomb');

for i=1:3

    sd = mults(1,i)*sigma;

    [img_out, rmsd] = myPatchBasedFiltering(img,img_noisy,sd);

    disp(strcat([num2str(mults(1,i)), ' ', num2str(rmsd)]));

    t = strcat(['honeycomb_', num2str(mults(1,i)),'.png']);
```

```
    imwrite(mat2gray(img_out),t);

end

% 0.84 for grassNoisy

% 0.9 for honeyComb

% 87 for barbara

toc;function [img_out,rmsd] = myPatchBasedFiltering(img_real, img_noisy, par)

    [h, w] = size(img_noisy);

    f = @(X) myPatchHelper(X, par);

    img_out = nlfilter(img_noisy, [33 33], f);

    rmsd = sqrt((sum(sum((img_real-img_out).^2)))/(h*w));

end

function res = myPatchHelper(window, h)

    g_filter = fspecial('gaussian',9,2);
```

```
    A = im2col(window, [9 9], 'sliding');

    B = repmat(A(:,ceil(end

    /2)), [1 size(A,2)]);

    C = repmat(g_filter(:), [1 size(A,2)]);

    D = (A-B).*C;

%     Scaling so that parameter can have high value

    D = D*100;

    weights = exp(-sum(D.*D)/(h*h));

    center_intensities = A(ceil(end

    /2),:);

    res = sum(weights.*center_intensities)/sum(weights);

end
```

check1

ORIGINALITY REPORT

6%	3%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Kochi University of Technology Student Paper	3%
2	Submitted to University of Westminster Student Paper	1%
3	roesland-uwahyudi.blogspot.com Internet Source	1%
4	fr.mathworks.com Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

check1