CS663: ASSIGNMENT-2

16D070005,16D070014,16D070027

August 20, 2018

0.1 Image Sharpening.

```
function myUnsharpMasking(A,hsize,sigma,amount)
%A = "..\data\superMoonCrop.mat";
B = load(A);
X=struct2array(B);

%-------------blurring-----------------
G = fspecial('gaussian',hsize,sigma);
blur = imfilter(X,G);
%figure(2),imshow(blur,[]);
%unsharpMask = original_img +K*(original-blur)
%k(weighing factor)
%-----------------UNSHARP MASKING-----------
Y = X-blur;
M = X + amount*Y;
%------------------------------stretching part-----------
orgImg = contrastStretching(X);
figure(1),imshow(orgImg,[])
title('Linearly Stretched Original Image');
colorbar('southoutside');

finImage = contrastStretching(M);
figure(2),imshow(finImage,[]);
title('Linearly Stretched Sharpen Image');
colorbar('southoutside');
```

```
end

%------------stretching----------------------------------------------------
function C = contrastStretching(X)
[rows,colm] = size(X);
low = min(X);
lowest = min(low);  %minimum intensity
high = max(X);
highest = max(high);   %maximum intensity
Z = 255/double(highest-lowest);
C  = zeros(rows,colm);
C(1:rows,1:colm) = double(X(1:rows,1:colm) - lowest)*(Z);
end
```

0.2 Edge-preserving Smoothing using Bilateral Filtering

```
function myBilateralFiltering(input_img,sigma_intensity,sigma_space,hws)
%input_img = load('../data/barbara.mat');
%input_img =  load('../data/grassNoisy.mat');
%input_img = load('../data/honeyCombReal_Noisy.mat');

c = cell2mat(struct2cell(input_img));
Intensity_diff = (max(c(:)) - min(c(:)))*0.05;
a = c + Intensity_diff*randn(size(c));

window_size = 2*(hws) + 1;

pad_a = padarray(a,[hws,hws],0,'both');

space_gauss = fspecial('gaussian',window_size,sigma_space);

m = size(a,1);
n = size(a,2);
l = size(a,3);
```

```
for r = 1:l
temp(:,:) = pad_a(:,:,r);
for i = hws+1:hws+m
    for j = hws+1:hws+n

        intensity_mask = temp((i-hws):(i+hws),(j-hws):(j+hws));
        intensity_gauss = normpdf(intensity_mask,temp(i,j),sigma_intensity);

        x = (intensity_gauss).*(space_gauss);
        y = x.*(intensity_mask);

        p = sum(x(:));
        q = sum(y(:));

        temp(i,j) = q/p;
    end
end
filtered_img(:,:,r) = temp((hws+1):(hws+m),(hws+1):(hws+n));
end

sd = ((filtered_img - c).^2);
rmsd = sqrt(mean(sd(:)));

figure(1),imshow(c,[]);
title('Original');
colorbar('southoutside');
figure(2),imshow(a,[]);
title('Noisy Input');
colorbar('southoutside');
figure(3),imshow(filtered_img,[]);
title('Bilateral Filtering');
colorbar('southoutside');
```

0.3 Edge-preserving Smoothing using Patch-Based Filtering.

```
function bigimg =  myBilinearInterpolation(b,k)
% load('../data/honeyCombReal_Noisy.mat')
% b = imgCorrupt;
% k = 2;

m = size(b,1);
n = size(b,2);

i = zeros(k*m-k+1,k*n-k+1,'double');
img = im2double(i);

for r = 0:(k*m-k+1)
    for c = 0:(k*n-k+1)
        fr = floor(r/k);
        fc = floor(c/k);
        cr = ceil(r/k);
        cc = ceil(c/k);
        if cc == n
            cc = n-1;
        end
        if cr == m
            cr = m-1;
        end

        d2r = (k*cr-r);
        d2c = (k*cc-c);
        d1r = (r - k*fr);
        d1c = (c - k*fc);

        if d1r == 0
            d1r = k;
        end
        if d1c == 0
            d1c = k;
        end
```

```
        x = (d2r)*(d2c)*double(b(fr+1,fc+1)) + (d1r)*(d2c)*double(b(cr+1,fc+1)) +
        i(r+1,c+1) = double(x/k*k);
        img(r+1,c+1) = double(x/k*k);
    end
end

bigimg = img;
% %
% figure(3),imshow(b,[]);
% title('Original Image');
% colorbar('southoutside');
% figure(4),imshow(img, []);
% title('Bilinear Interpolation');
% colorbar('southoutside');
end
----------------------------------------------------------------------
function output = myShrinkImageByFactorD(a,d)
% A = '../data/circles_concentric';
% b = imread(A,'png');

%d = 3;
filt = fspecial('gaussian',5,0.66);
b = imfilter(a,filt);

r = floor(size(b,1)/d);
c = floor(size(b,2)/d);

m = zeros(r,c,'double');
m(1:r,1:c) = b(d:d:d*r,d:d:d*c);

output = m;

% figure(1),imshow(b);
% title('Original Image');
% colorbar('southoutside');
% figure(2),imshow(m);
% title('Shrunken Image');
```

```
% colorbar('southoutside');
end
-----------------------------------------------------------------
function rmsd = myPatchBasedFiltering(img,imgOrig,sigma)
% load('../data/barbara.mat')
% load('../data/grassNoisy.mat')
%
% load('../data/honeyCombReal_Noisy.mat')

% img = imgCorrupt;
% imgOrig = imread('../data/grass.png');
% sigma = 0.35;
% img = myShrinkImageByFactorD(img1,2);

m = size(img,1);
n = size(img,2);
o = zeros(m,n,'double');
o = double(imgOrig);
patch_size = 9;
p = (patch_size-1)/2;
mask = fspecial('gaussian',patch_size,3);
mask = 81*mask;

window_size = 25;
w = (window_size-1)/2;

m=m+p+p;
n=n+p+p;
paddedimg = zeros(m,n);
paddedimg(1+p:m-p,1+p:n-p) = img(:,:);
paddedimg(1:p,:) = paddedimg(1+p:p+p,:);
paddedimg(m-p+1:m,:) = paddedimg(m-p-p+1:m-p,:);
paddedimg(:,1:p) = paddedimg(:,1+p:p+p);
paddedimg(:,n-p+1:n) = paddedimg(:,n-p-p+1:n-p);

%%

newimg = img;
```

```
for r = 1+p:m-p
    for c = 1+p:n-p
        mypatch = paddedimg(r-p:r+p,c-p:c+p);
        mypatch = mypatch.*mask;
        wrmin = max(1+p,r-w);
        wrmax = min(m-p,r+w);
        wcmin = max(1+p,c-w);
        wcmax = min(n-p,c+w);

        distmat = zeros(wrmax-wrmin+1,wcmin-wcmax+1);
        for i = wrmin:wrmax
        for j = wcmin:wcmax
            newpatch = paddedimg(i-p:i+p,j-p:j+p);
            newpatch = newpatch.*mask;
            distmat(i-wrmin+1,j-wcmin+1) = sqrt(sumsqr(mypatch-newpatch));
        end
        end
        weights = gaussmf(distmat,[sigma,0]);
        sum_weights = sum(weights(:));
        weights = weights/sum_weights;

        newimg(r-p,c-p) = sum(sum((paddedimg(wrmin:wrmax,wcmin:wcmax).*weights(:,:
    end
end
%%
newimg = newimg./max(newimg(:));
o = o./max(o(:));
sd = ((newimg-o).^2);
rmsd = sqrt(mean(sd(:)));
%%
figure(1),imshow(o,[]);
figure(2),imshow(img,[]);
% big_newimg = myBilinearInterpolation(newimg,2);
figure(3),imshow(newimg,[]);

end
```