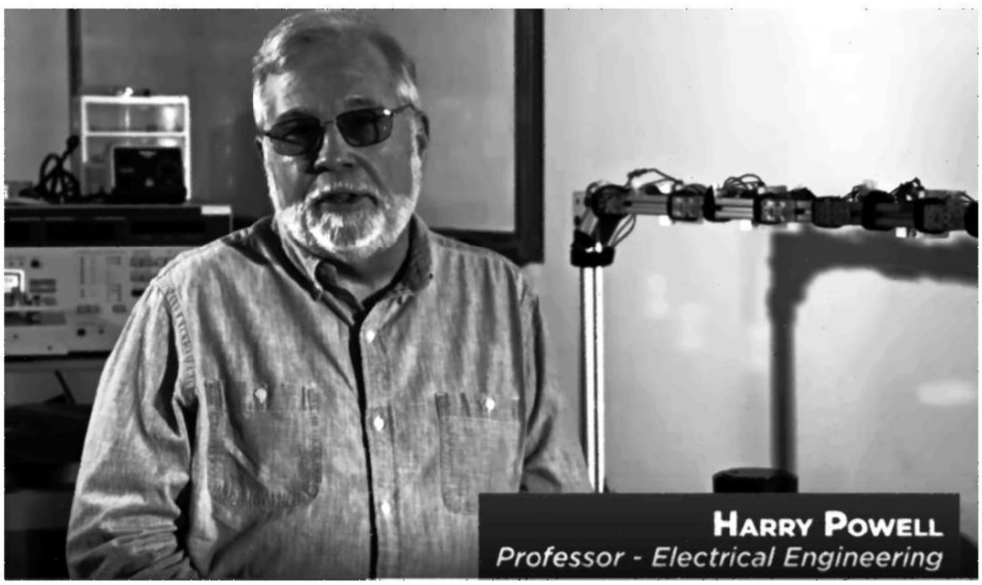
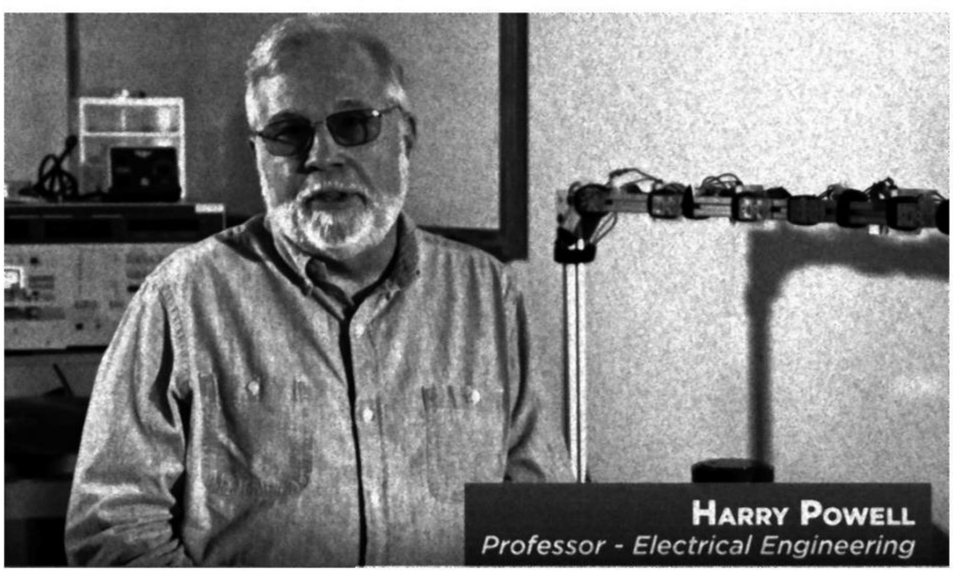
Question 1 15 points
Design a "Trimmed Mean Filter"
Apply your filter to the images with suitable parameters (P, Q) attached to this homework.
Report P, Q
(check the module), and the final image(s).

I choose P=4, Q=2 for 3*3 window
Because the P is usually equal to 2Q, for 9 elements, it can be written as 9=2+5+2

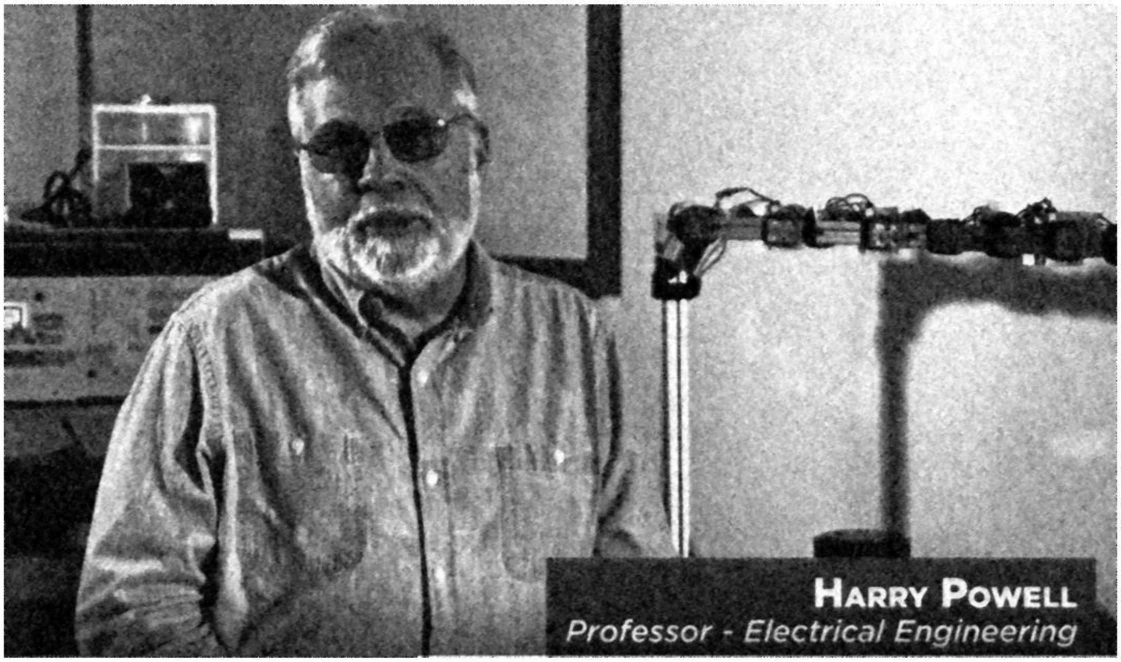
Here is the results and code:
Tri_HarrySP.jpg



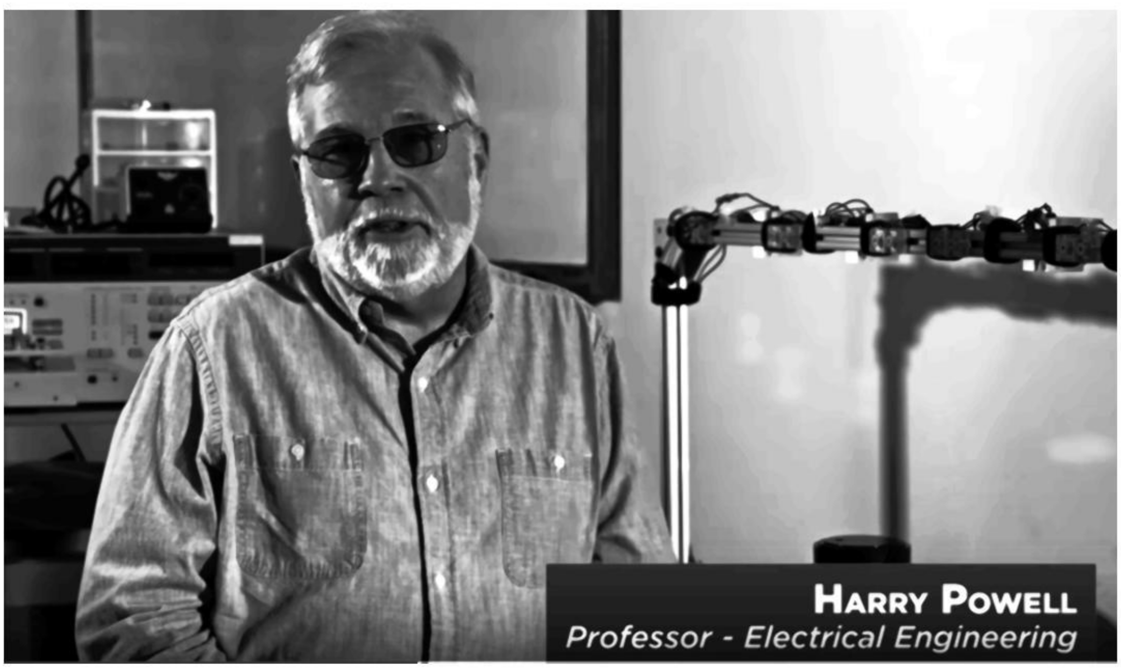
Tri_HarrySpeckle.jpg



Tri_HarryGauss.jpg



Tri_HarryBW.jpg



Code:

```
Img=imread('HarryBW.png');
Img=double(Img);
%imshow(I);

window_size=3;
stack=zeros(1,window_size^2);
LC=size(Img);
row=LC(1);
col=LC(2);

 for i=2:(row-1)
     for j=2:(col-1)
         window=[Img(i-1,j-1),Img(i,j-1),Img(i+1,j-1),...
                 Img(i-1,j),Img(i,j),Img(i+1,j),...
                 Img(i-1,j+1),Img(i,j+1),Img(i+1,j+1)];
         %Top=median(window); %test the median filter
         I=sort(window);
         Img(i,j)=(I(3)+I(4)+I(5)+I(6)+I(7))*1/5; %9=2+5+2
     end
 end

imshow(Img,[]);
```

Question 2 25 points

Design a "Bilateral Filter" (Module 5).

Apply your filter to the images with suitable parameters (K_G, σ_G, K_H, σ_H) attached to this homework.

Report K_G, σ_G, K_H, σ_H (check the module), and the final image(s).

Based on class slides and this material:
https://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf

1. For K_G, K_H

I think these two coefficients are liner coefficients. Change them will cause the whole picture become brighter or darker. So I choose K_G=1, K_H=1 for convenience. Though I know these coefficients are designed by $\frac{1}{\sqrt{2\pi * \sigma}}$. But there is little difference by doing this.

2. For σ_G, σ_H

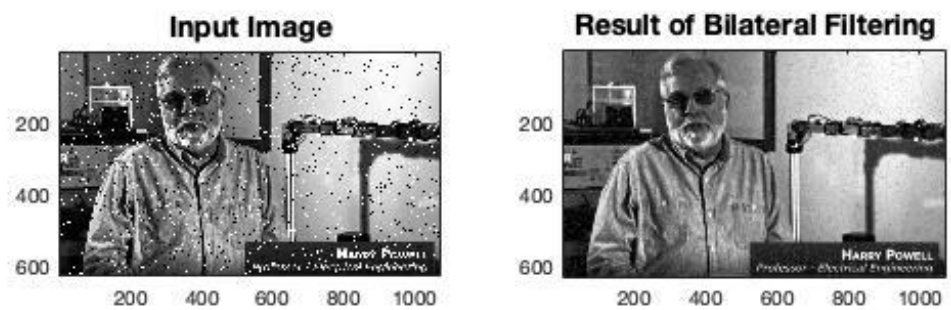
These two coefficients are important.

I. For spatial domain (Euclidean distance.),
The kernel size is usually 6 * sigma +1 of sigma. Because the points outside the center point 3 * sigma' effect are only very small(Probability 3-sigma theory).
The kernel size is 9, so I choose σ_G= 1.3

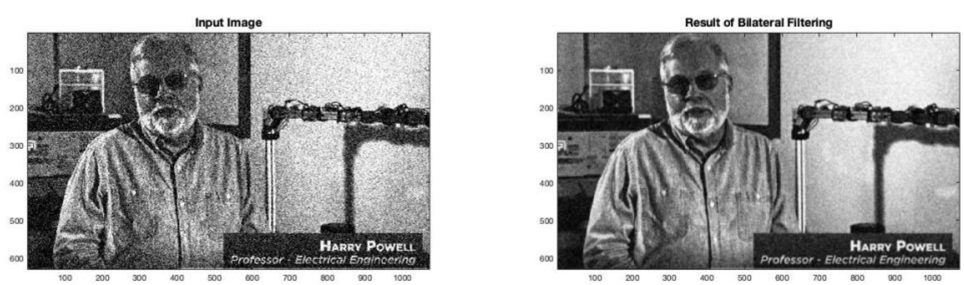
II. For luminance similarity(Range filter)

σ_H = abs（gray(i,j)- gray(mean_i, mean_j) By get_var.m, the mean of image is 1797= σ_H.

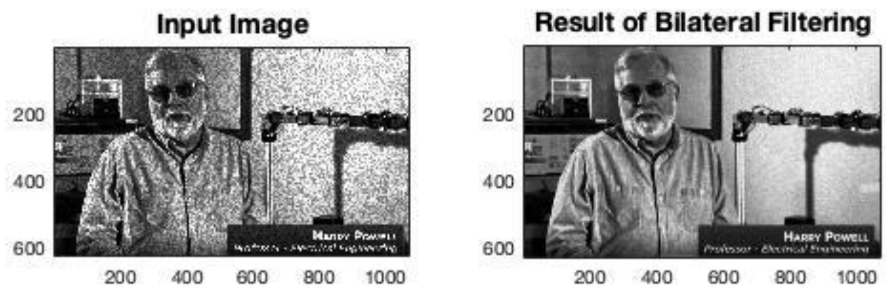
Results: I checked different values, the model works good.
SP.jpg



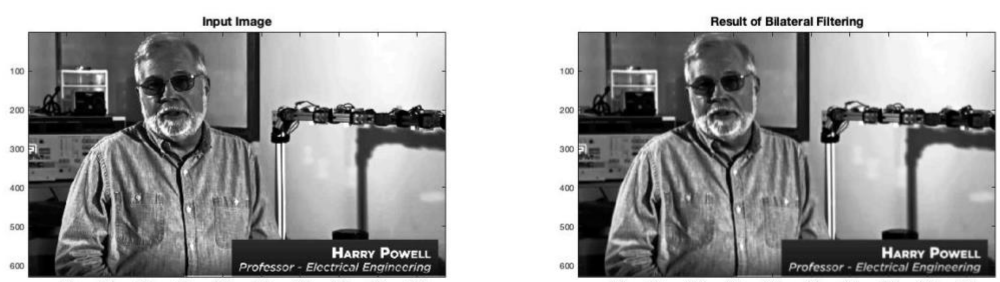
Gauss.jpg



Speckle.jpg



BW.jpg



Code:
Some codes are from https://github.com/GKalliatakis/Bilateral-Filtering/blob/master/bilateralGrayscale.m

```
clear;
clc;
tic;
Img=imread('HarryBW.png');
[row col dim]=size(Img);
sigma_r=1797;
sigma_s=1.3;

if dim>2
Img=rgb2gray(Img);
end

window_size=9;

%transforms the domain specified by vectors from -window_size to window_size
into arrays X and Y
[x,y]=meshgrid(-window_size:window_size,-window_size:window_size);

%% Domain filter
%The weights depend on the spatial distance (to the center pixel x) only;
therefore, it is calculated once and saved.
domain_filter=exp(-(x.^2+y.^2)/(2*sigma_s^2));

%% Repeat for all pixels
[r,c]=size(Img);
output=zeros(size(Img));

% Create waitbar.
h = waitbar(0,'Wait...');
set(h,'Name','Bilateral Fiter Processing');

for i=1:r
    for j=1:c

        %Adjusting the window size
        imin=max(i-window_size,1);
        imax=min(i+window_size,r);
        jmin=max(j-window_size,1);
        jmax=min(j+window_size,c);
        I=Img(imin:imax,jmin:jmax);

        range_filter=exp(-double(I-Img(i,j)).^2/(2*sigma_r^2)); % Range
filter

        %Taking the product of the range and domain filter.The combination is
refered to as Bilater Filter
        BilateralFilter=range_filter.*domain_filter((imin:imax)-
i+window_size+1,(jmin:jmax)-j+window_size+1);

        Fnorm=sum(BilateralFilter(:));
```

```
        output(i,j)=sum(sum(BilateralFilter.*double(I)))/Fnorm; %normalize
the output
    end
    waitbar(i/(r-sigma_s));
end
close(h)
elapsed_time=toc;
clc;
fprintf('Total elapsed time is: %f seconds \n\n', elapsed_time);
imwrite(output,'outputImg.jpg');


%% Display the results
figure(1); clf;
set(gcf,'Name','Grayscale Bilateral Filtering Results');
subplot(1,2,1); imagesc(Img);
axis image; colormap gray;
title('Input Image','fontweight','bold','fontsize',14);
subplot(1,2,2); imagesc(output);
axis image; colormap gray;
title('Result of Bilateral Filtering','fontweight','bold','fontsize',14);
```