Project 6: Radon Transform

Contents

Course No: ECE 5256

Due Date: 3/17/2021

Q1.) Acquire an image and perform the Radon transform on it, and reconstruct the image using the commands radon and iradon respectively, and display the results. Be sure to use enough angle resolution to give reasonable results. Use two different combinations of filters and interpolation in the inverse command and indicate which one gives the best result determined by the lowest mean squared error (MSE) between the original image and the reconstructed image. The MSE is a scalar value that is the sum of ((original image – reconstructed image).^2)/(number of pixels in an image).

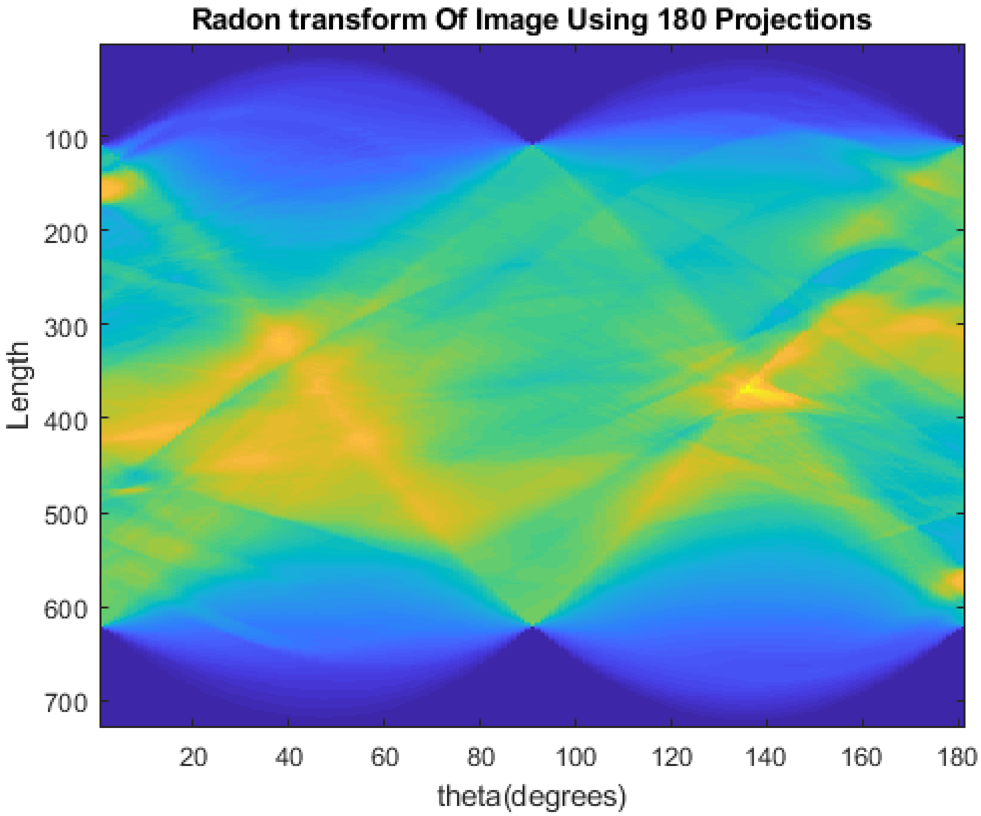
Acquire an image

```
I = imread("Lena.png");
J=rgb2gray(I);
imshow(J);
title("Original Image");
```

Original Image

Apply Radon transform on image

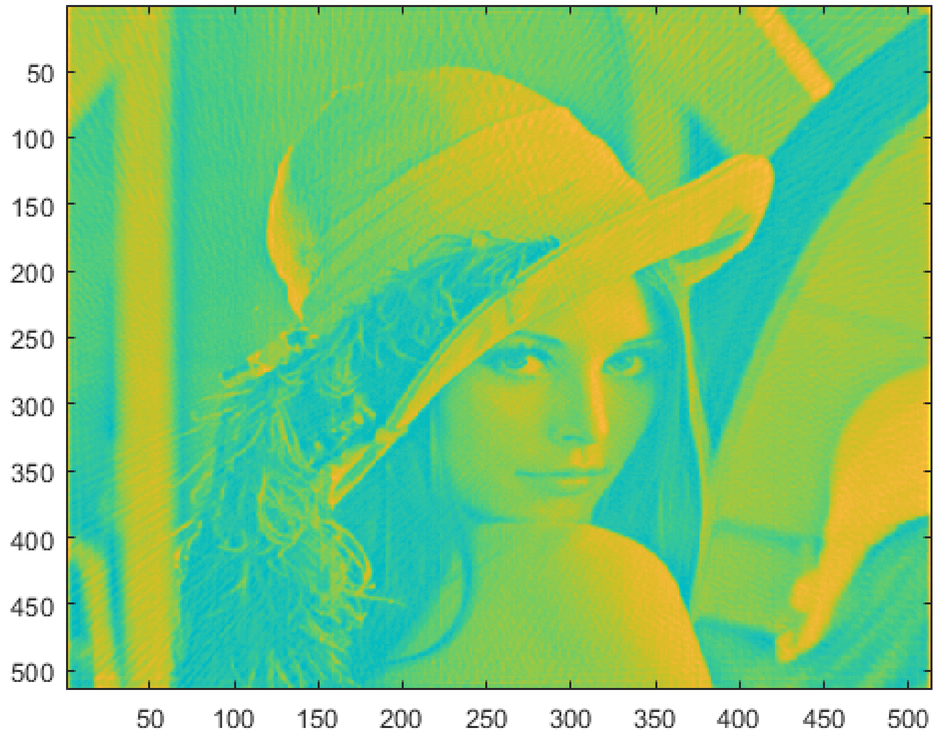
```
theta = 0:180;
R = radon(J,theta);
figure;imagesc(R);
title("Radon transform Of Image Using 180 Projections");
xlabel("theta(degrees)");
ylabel("Length");
```



Apply inverse Radon Transform

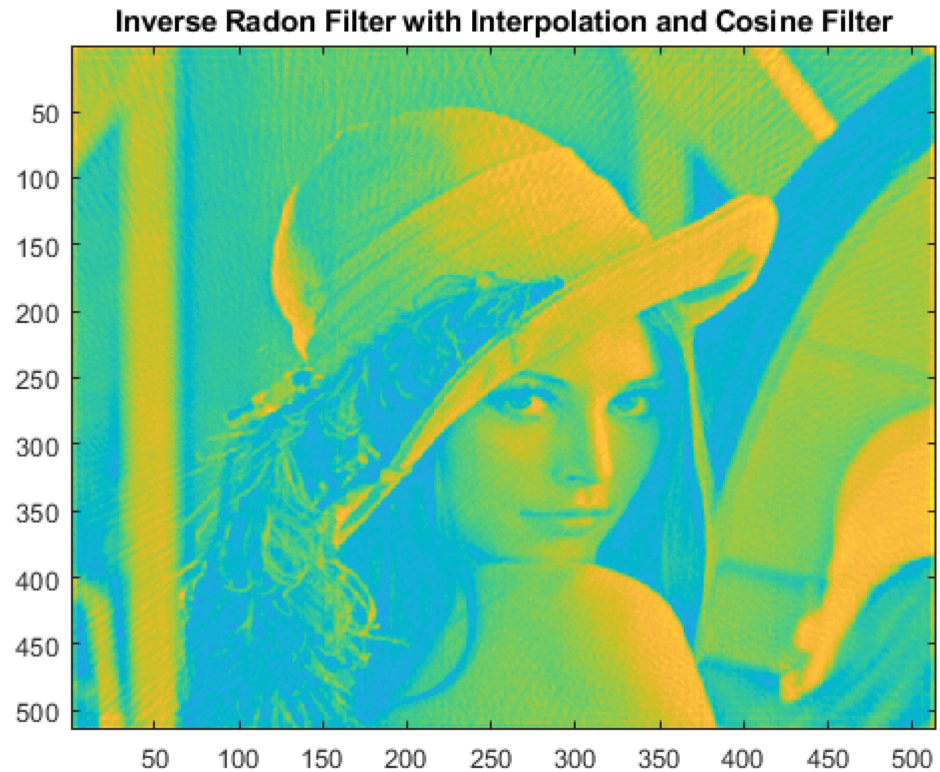
Inverse Radon without Interpolation

```
I1 = iradon(R,theta); %I = iradon(R,THETA,INTERPOLATION,FILTER,FREQUENCY_SCALING,OUTPUT_SIZE)
figure;title("Inverse Radon filter without Interpolation");
imagesc(I1);
```



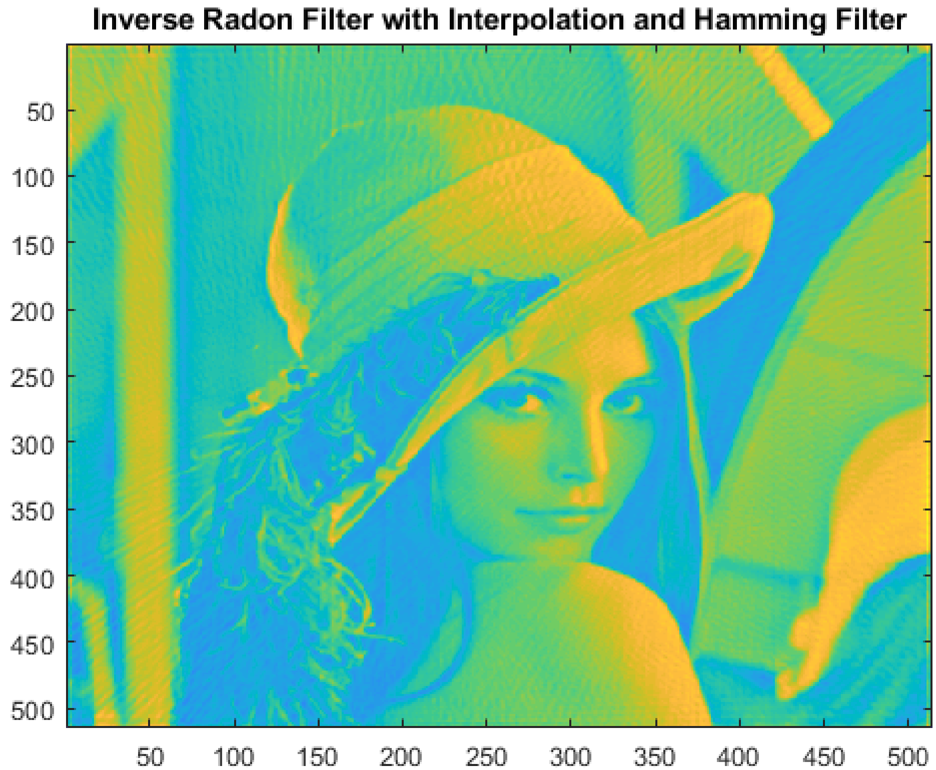
Inverse Radon with Interpolation = Nearest and Filter = Cosine

```
I2 = iradon(R,theta,'nearest','Cosine'); %Interpolation and Filter appllied
imagesc(I2);
title("Inverse Radon Filter with Interpolation and Cosine Filter");
```



Inverse Radon with Interpolation = spline and Filter = Hamming

```
I3 = iradon(R,theta,'spline','Hamming'); %Interpolation and Filter appllied
imagesc(I3);
title("Inverse Radon Filter with Interpolation and Hamming Filter");
```



lowest mean squared error (MSE) between the original image and the reconstructed image

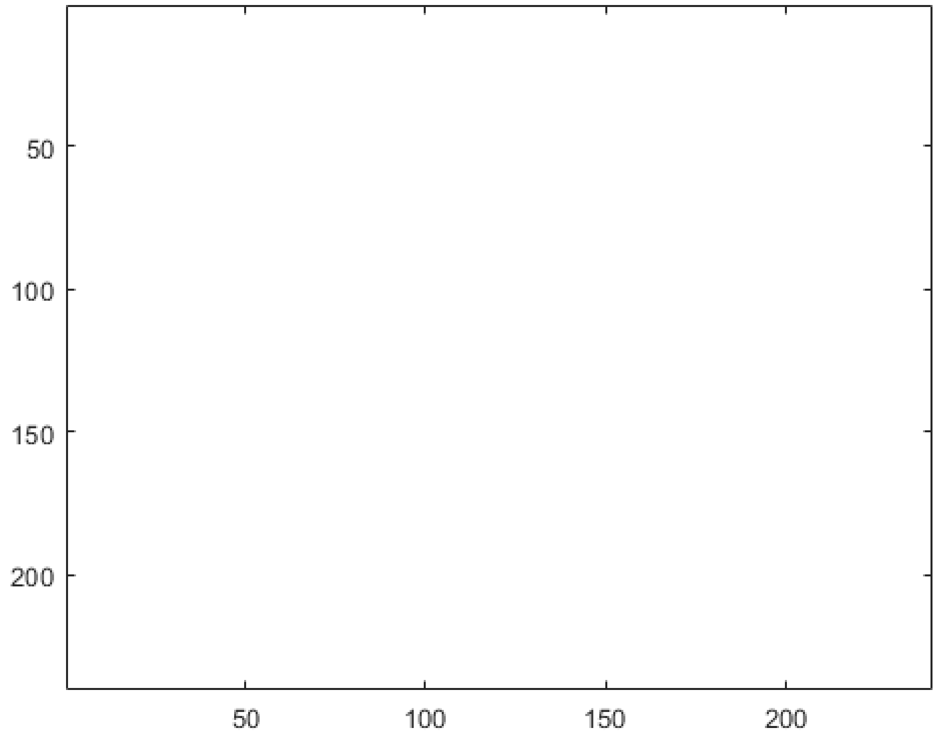
```
J=double(J);
I2=double(I2);
I=double(I);
[M, N] = size(J);
%MSE Of Original Image - reconstructed Image
MSE=sum(J-I2(1,512))+(J-I(1,512))+(J-I3(1,512)).^2/(M*N);
N=min(MSE(1:512));
fprintf('The MSE Value obtained is  1.441981596707358e+03');

% The MSE value obtained is 1.441981596707358e+03
```

```
The MSE Value obtained is  1.441981596707358e+03
```

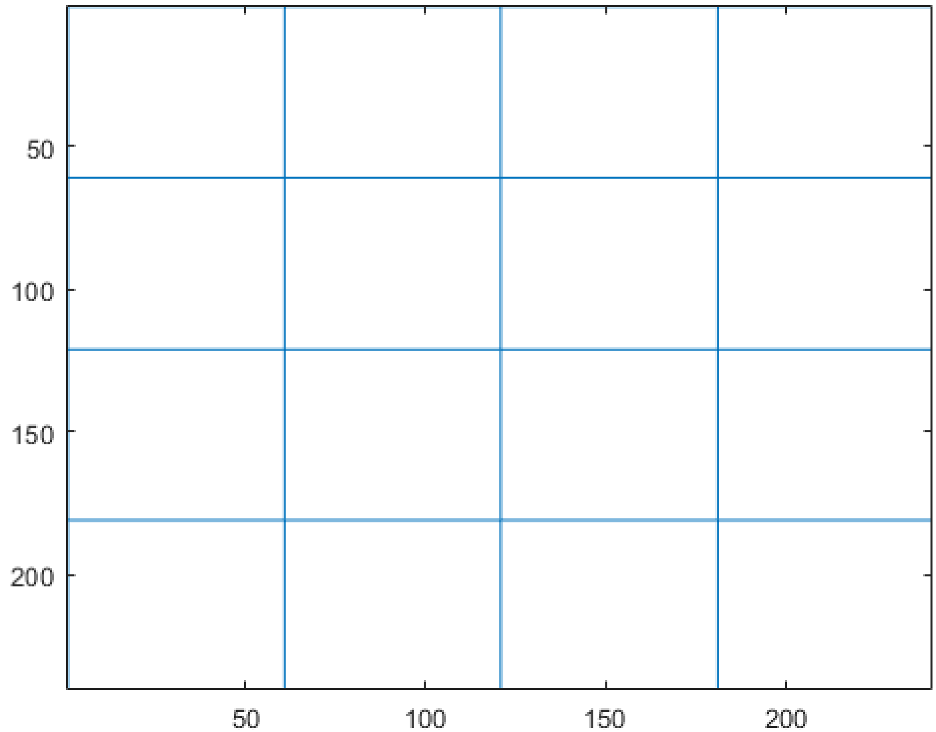
Q2.) Create an image of a grid with three line in the horizonal direction, and three in the vertical (white on background). Add noise to the image where the standard deviation of the noise is about 20% of the value of the lines. Then, take the Radon transform of the noisy image. Try to reconstruct only the grid by only allowing the dominate pixels (or small areas) in the inverse Radon transform.

```
%Create a image with white background.
whiteImage = ones(240, 240);
colormap(white);
imagesc(whiteImage);
```



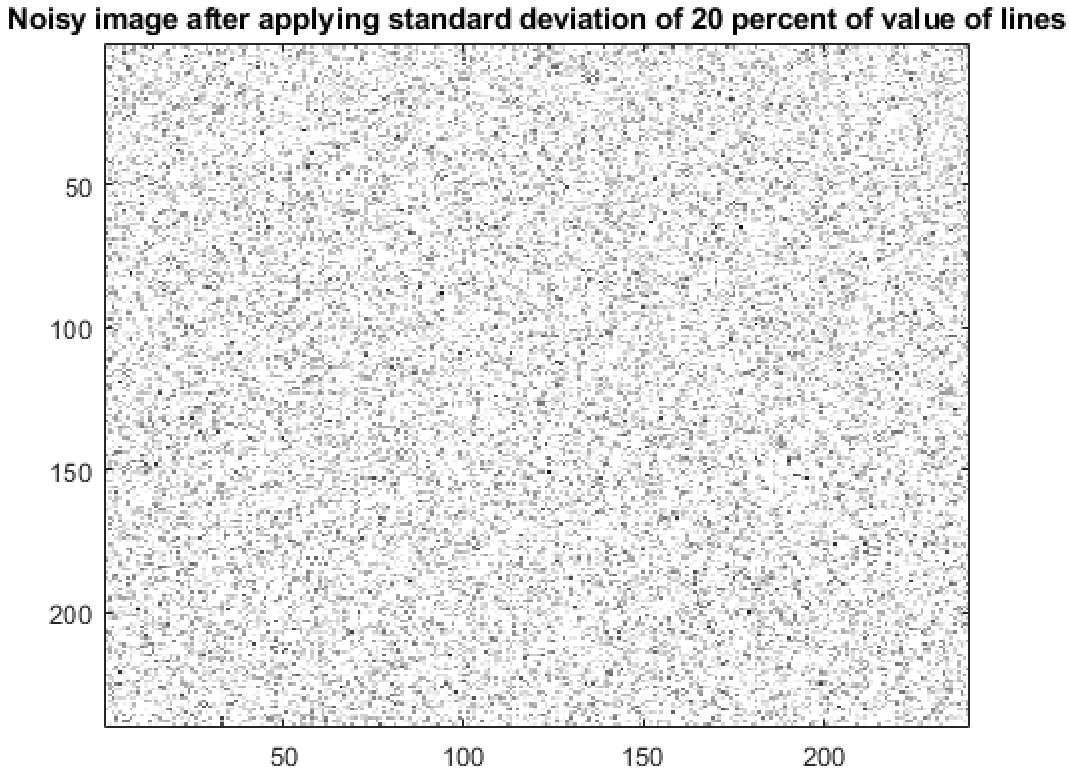
Adding vertical and Horizontal Lines on the image.

```
[i,j]=size(whiteImage);
spacing = 60;
for row=1:spacing:240
    Z=line([1,j],[row,row]);
end
for column=1:spacing:240
    Z=line([column,column],[1,j]);
end
```



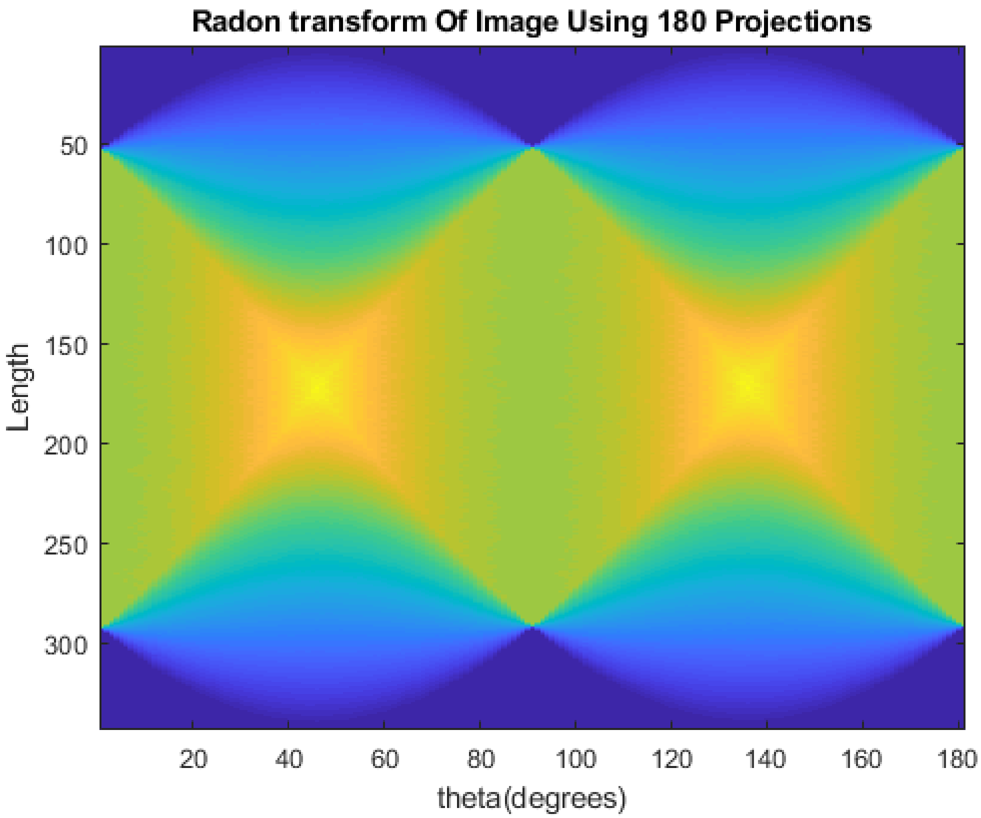
Add noise to the image where the standard deviation of the noise is about 20% of the value of the lines

```
A = imnoise(whiteImage,'gaussian');
standard_dev = 0.20.*(A);
imagesc(standard_dev);
colormap(gray);
title("Noisy image after applying standard deviation of 20 percent of value of lines");
```



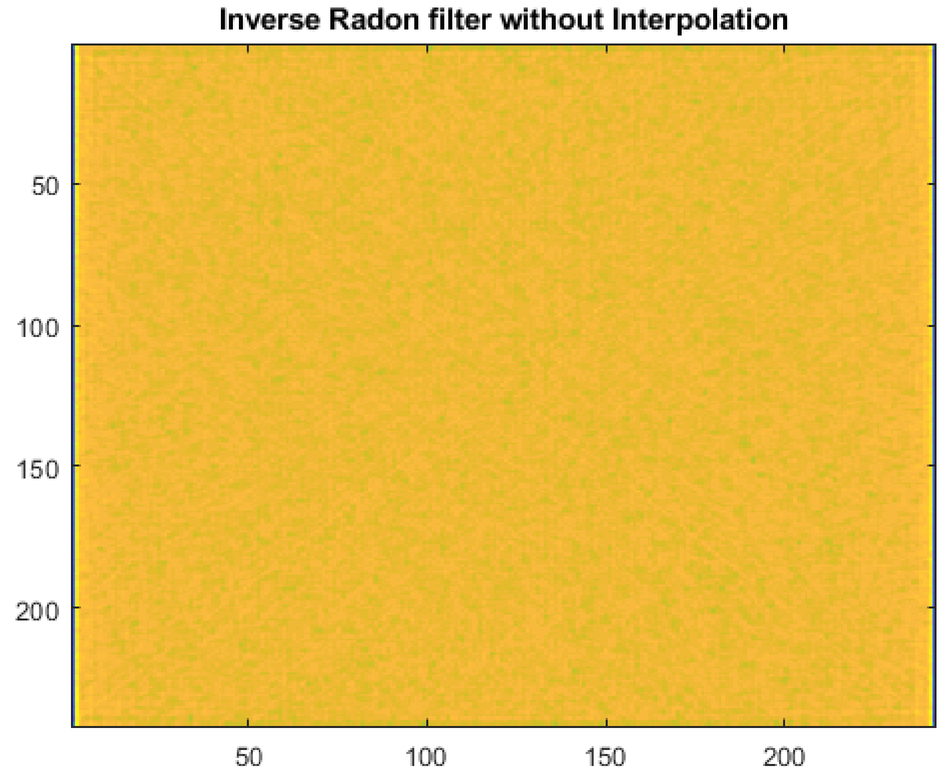
Take the Radon transform of the noisy image

```
thetas = 0:180;
K_noise = radon(standard_dev,thetas);
figure;imagesc(K_noise);
title("Radon transform Of Image Using 180 Projections");
xlabel("theta(degrees)");
ylabel("Length");
```



Try to reconstruct only the grid by only allowing the dominate pixels (or small areas) in the inverse Radon transform

```
I1_noise = iradon(K_noise,thetas); %I = iradon(R,THETA,INTERPOLATION,FILTER,FREQUENCY_SCALING,OUTPUT_SIZE)
imagesc(I1_noise);
title("Inverse Radon filter without Interpolation");
```



Published with MATLAB® R2019a