Project 3: Image Restoration

Table of Contents

Name: Socheath Sok ID: 014470701 Class: EE 483, T/Th 2:00 - 3:15 pm

Ch 8: Problem 6ab

Choose any grayscale image of your choice and add 5% salt and pepper noise. Remove the noise with average filtering and median filtering. Which method gives the best results?

```
clear; clc; close all;
cman = imread('cameraman.png');
n5 = imnoise(cman,'salt & pepper');  % default: 5% noise
c3 = [ 0 1 0 ; 1 1 1 ; 0 1 0 ];
      % 3x3 cross mask
c5 = [ 0 0 1 0 0 ; 0 0 1 0 0 ; 1 1 1 1 1 ; 0 0 1 0 0 ; 0 0 1 0 0 ];
      % 5x5 cross mask
x3 = [ 1 0 1 ; 0 1 0 ; 1 0 1];
      % 3x3 X mask
x5 = [ 1 0 0 0 1 ; 0 1 0 1 0 ; 0 0 1 0 0 ; 0 1 0 1 0 ; 1 0 0 0 1];
      % 5x5 X mask

figure(1);
subplot(1,2,1);imshow(cman); title('Original');
subplot(1,2,2); imshow(n5); title('5% Salt & Pepper noise')

figure(2); sgtitle('Average Filters');
subplot(1,2,1); imshow(imfilter(n5,fspecial('average')));
 title('3x3');
subplot(1,2,2); imshow(imfilter(n5,fspecial('average',5)));
 title('5x5');

figure(3); sgtitle('Median Filters')
subplot(2,3,1); imshow(medfilt2(n5)); title('3x3 Full Mask');
subplot(2,3,4); imshow(medfilt2(n5,[5,5])); title('5x5 Full Mask');
subplot(2,3,2); imshow(ordfilt2(n5,3,c3)); title('3x3 Cross Mask');
subplot(2,3,5); imshow(ordfilt2(n5,5,c5)); title('5x5 Cross Mask');
subplot(2,3,3); imshow(ordfilt2(n5,3,x3)); title('3x3 X Mask');
subplot(2,3,6); imshow(ordfilt2(n5,5,x5)); title('5x5 X Mask');
```

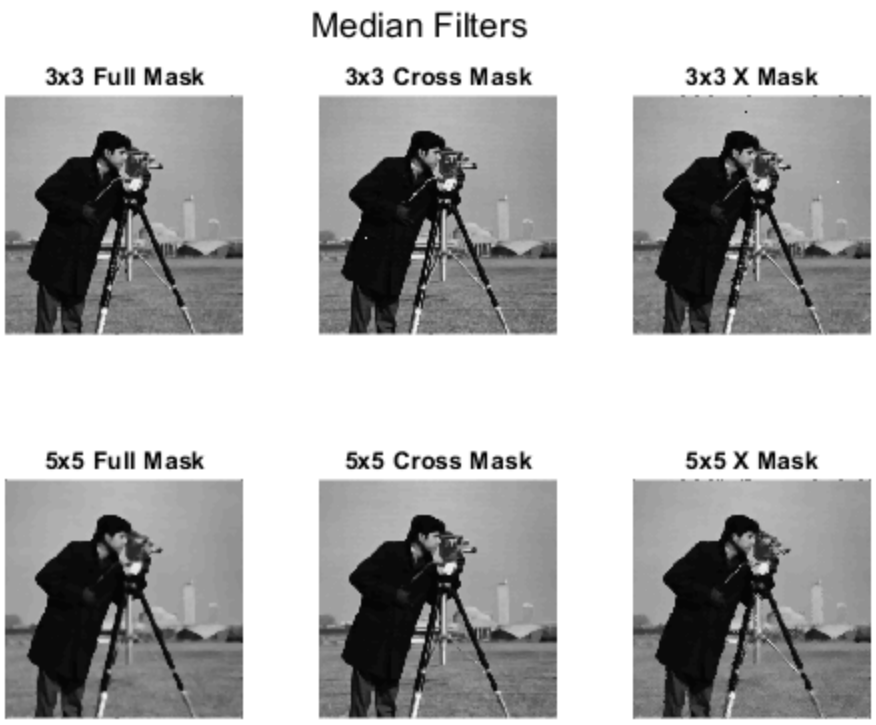
Original

5% Salt & Pepper noise

Average Filters

3x3

5x5

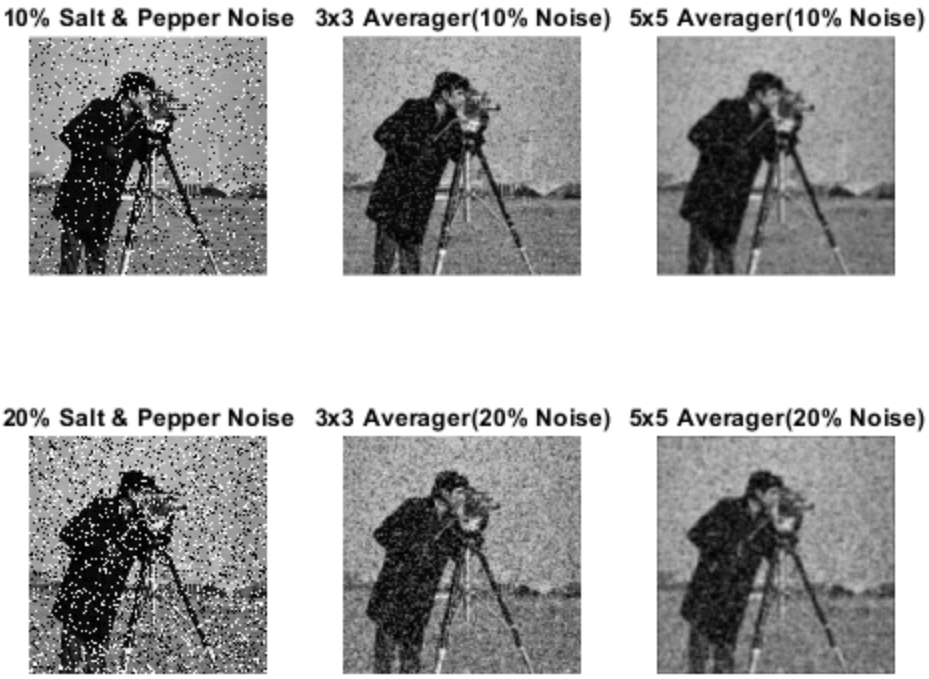


Ch 8: Problem 7ab

Repeat problem 6 with 10% and 20% noise.

```
n10 = imnoise(cman,'salt & pepper',0.1);        % 10% noise
n20 = imnoise(cman,'salt & pepper',0.2);        % 20% noise
figure(6);
subplot(2,3,1); imshow(n10); title('10% Salt & Pepper Noise');
subplot(2,3,2); imshow(imfilter(n10,fspecial('average',3)));
 title('3x3 Averager(10% Noise)');
subplot(2,3,3); imshow(imfilter(n10,fspecial('average',5)));
 title('5x5 Averager(10% Noise)');
subplot(2,3,4); imshow(n20); title('20% Salt & Pepper Noise');
subplot(2,3,5); imshow(imfilter(n20,fspecial('average',3)));
 title('3x3 Averager(20% Noise)');
subplot(2,3,6); imshow(imfilter(n20,fspecial('average',5)));
 title('5x5 Averager(20% Noise)');

figure(7); sgtitle('Median Filters on 10% Noise');
subplot(2,3,1); imshow(medfilt2(n10)); title('3x3 Full Mask');
subplot(2,3,4); imshow(medfilt2(n10,[5,5])); title('5x5 Full Mask');
subplot(2,3,2); imshow(ordfilt2(n10,3,c3)); title('3x3 Cross Mask');
subplot(2,3,5); imshow(ordfilt2(n10,5,c5)); title('5x5 Cross Mask');
subplot(2,3,3); imshow(ordfilt2(n10,3,x3)); title('3x3 X Mask');
subplot(2,3,6); imshow(ordfilt2(n10,5,x5)); title('5x5 X Mask');
```

```
figure(8); sgtitle('Median Filters on 20% Noise');
subplot(2,3,1); imshow(medfilt2(n20)); title('3x3 Full Mask');
subplot(2,3,4); imshow(medfilt2(n20,[5,5])); title('5x5 Full Mask');
subplot(2,3,2); imshow(ordfilt2(n20,3,c3)); title('3x3 Cross Mask');
subplot(2,3,5); imshow(ordfilt2(n20,5,c5)); title('5x5 Cross Mask');
subplot(2,3,3); imshow(ordfilt2(n20,3,x3)); title('3x3 X Mask');
subplot(2,3,6); imshow(ordfilt2(n20,5,x5)); title('5x5 X Mask');
```



Median Filters on 10% Noise

3x3 Full Mask

3x3 Cross Mask

3x3 X Mask

5x5 Full Mask

5x5 Cross Mask

5x5 X Mask

Median Filters on 20% Noise

3x3 Full Mask

3x3 Cross Mask

3x3 X Mask

5x5 Full Mask

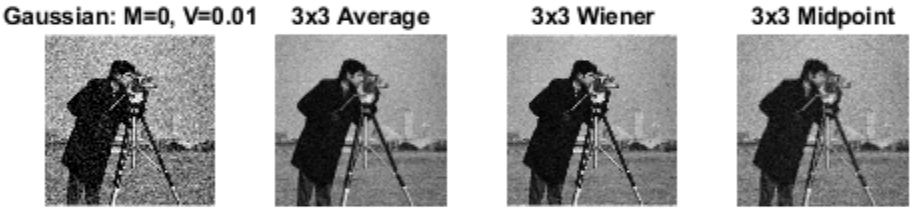
5x5 Cross Mask

5x5 X Mask

Ch 8: Problem 9 & 10

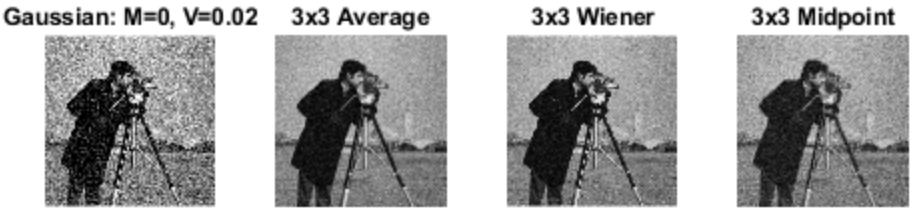
Add Gaussian noise to an image with specific parameters. Remove noise using average, Wiener, and midtpoint filters. Can filters produce satisfactory results with the last two noisy images? How do the results compare with spatial Wiener filtering or using a blurring filter?

```
clear; clc; close all;
e = imread('cameraman.png');
g1 = imnoise(e,'gaussian');              % M=0, V=0.01
g2 = imnoise(e,'gaussian',0,0.02);       % M=0, V=0.02
g5 = imnoise(e,'gaussian',0,0.05);       % M=0, V=0.05
g10 = imnoise(e,'gaussian',0,0.1);       % M=0, V=0.1

figure(1);
subplot(1,4,1); imshow(g1); title('Gaussian: M=0, V=0.01');
subplot(1,4,2); imshow(imfilter(g1,fspecial('average'))); title('3x3
 Average');
subplot(1,4,3); imshow(wiener2(g1)); title('3x3 Wiener');
min = ordfilt2(g1,1,ones(3,3));
max = ordfilt2(g1,9,ones(3,3));
subplot(1,4,4); imshow(imlincomb(0.5,min,0.5,max)); title('3x3
 Midpoint');
```

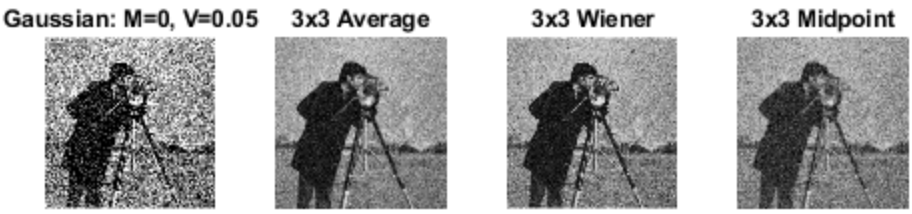



```
figure(2);
subplot(1,4,1); imshow(g2); title('Gaussian: M=0, V=0.02');
```

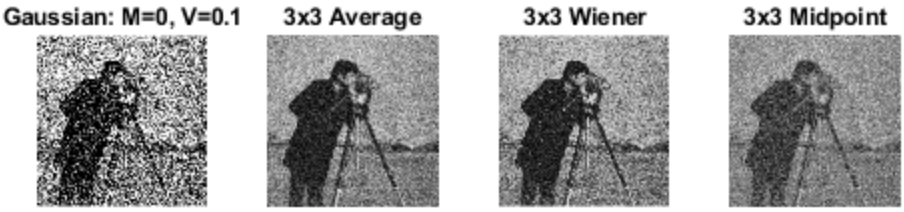
```
subplot(1,4,2); imshow(imfilter(g2,fspecial('average'))); title('3x3
 Average');
subplot(1,4,3); imshow(wiener2(g2)); title('3x3 Wiener');
min = ordfilt2(g2,1,ones(3,3));
max = ordfilt2(g2,9,ones(3,3));
subplot(1,4,4); imshow(imlincomb(0.5,min,0.5,max)); title('3x3
 Midpoint');
```




```
figure(3);
subplot(1,4,1); imshow(g5); title('Gaussian: M=0, V=0.05');
subplot(1,4,2); imshow(imfilter(g5,fspecial('average'))); title('3x3
 Average');
subplot(1,4,3); imshow(wiener2(g5)); title('3x3 Wiener');
min = ordfilt2(g5,1,ones(3,3));
max = ordfilt2(g5,9,ones(3,3));
subplot(1,4,4); imshow(imlincomb(0.5,min,0.5,max)); title('3x3
 Midpoint');
```




```
figure(4);
subplot(1,4,1); imshow(g10); title('Gaussian: M=0, V=0.1');
subplot(1,4,2); imshow(imfilter(g10,fspecial('average'))); title('3x3
 Average');
subplot(1,4,3); imshow(wiener2(g10)); title('3x3 Wiener');
min = ordfilt2(g10,1,ones(3,3));
max = ordfilt2(g10,9,ones(3,3));
subplot(1,4,4); imshow(imlincomb(0.5,min,0.5,max)); title('3x3
 Midpoint');
```



Ch 8: Problem 12

Add sine waves (s = 1+sin(x+y/1.5) to an image. Remove the noise uisng band-reject and criss-cross filterings. Which one give the best result?

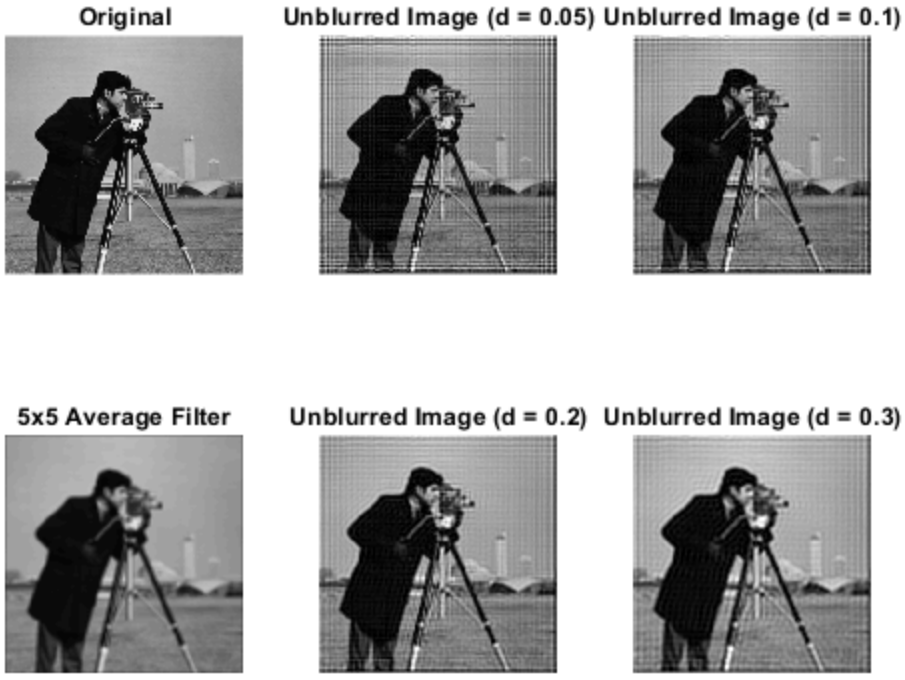
```
clear; clc; close all;
cman = imread('cameraman.png');

% Adding Periodic noise
[r, c] = size(cman);
[y,x] = meshgrid(1:c,1:r);
s = 1+sin(x+y/1.5);
pn = (2*im2double(cman) + s/2)/3;

% 2D FFT
gf = fftshift(fft2(pn));
gdisp = mat2gray(log(1+abs(gf)));

% Process the maximum locations with DC removed
xc = ceil((1+r)/2);
yc = ceil((1+c)/2);
gf2 = im2uint8(mat2gray(abs(gf)));
gf2(xc,yc) = 0;
[xm,ym] = find(gf2==max(gf2(:)));
dsq = (xm-xc).^2 + (ym-yc).^2  ;
gf2c = gf;  gf2c(xc,yc) = 0;
```

```
d = sqrt(dsq(1));

% Band-reject filter + Inverse FFT
z = sqrt((x-xc).^2 + (y-yc).^2);
k = 1;
br = (z < floor(d-k) | z > ceil(d+k));
Gfr5 = gf.*br;
Gdisp5 = mat2gray(log(1+abs(Gfr5)));
gfr5 = mat2gray(abs(ifft2(Gfr5)));

% Criss-cross filter
Gfcc = gf;
Gfcc(xm,:) = 0;
Gfcc(:,ym) = 0;
Gdisp = mat2gray(log(1+abs(Gfcc)));
gfcc = mat2gray(abs(ifft2(Gfcc)));

subplot(1,3,1); imshow(pn); title('Image with Periodic Noise');
subplot(1,3,2); imshow(gfr5); title('Band-Reject, k =1');
subplot(1,3,3); imshow(gfcc); title('Criss-Cross');
```

Ch 8: Problem 14,and 15

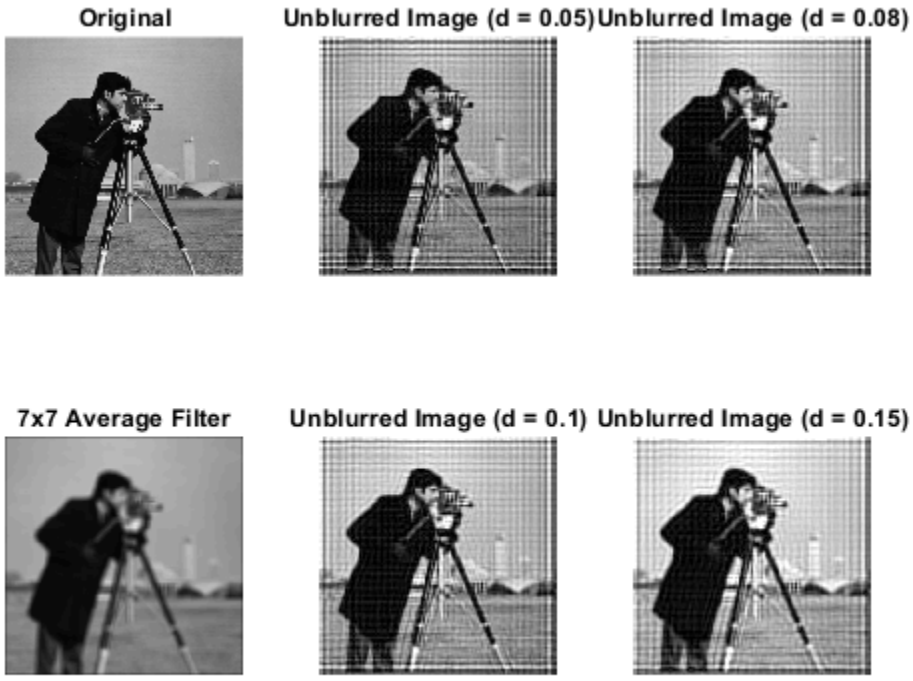
Apply a 5x5 and 7x7 blurring filters to a grayscale image. Deblur the result using inverse filtering with constrained division. Which threshold gives the best results?

```
clear; clc; close all;
en = imread('cameraman.png');
f = fspecial('average',5);
blur = imfilter(en,f);

z = zeros(size(en));
z(1:5,1:5) = f;
ft = fft2(z);

d = 0.05;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb01 = 1.2*(mat2gray(ba));

d = 0.1;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb005 = 1.2*(mat2gray(ba));

d = 0.2;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb002 = 1.2*(mat2gray(ba));;

d = 0.3;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb001 = 1.2*(mat2gray(ba));

subplot(2,3,1); imshow(en); title('Original');
subplot(2,3,4); imshow(blur); title('5x5 Average Filter');
subplot(2,3,2); imshow(unb01); title('Unblurred Image (d = 0.05)');
subplot(2,3,3); imshow(unb005); title('Unblurred Image (d = 0.1)');
subplot(2,3,5); imshow(unb002); title('Unblurred Image (d = 0.2)');
subplot(2,3,6); imshow(unb001); title('Unblurred Image (d = 0.3)');
```




```
clear; clc; close all;
en = imread('cameraman.png');
f = fspecial('average',7);
blur = imfilter(en,f);

z = zeros(size(en));
z(1:7,1:7) = f;
ft = fft2(z);

d = 0.05;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb01 = 1.5*(mat2gray(ba));

d = 0.08;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb005 = 1.5*(mat2gray(ba));

d = 0.1;
ft(find(abs(ft)<d)) = 1;
ba = abs(ifft2(fft2(blur)./ft));
unb002 = 1.5*(mat2gray(ba));;

d = 0.15;
ft(find(abs(ft)<d)) = 1;
```

```
ba = abs(ifft2(fft2(blur)./ft));
unb001 = 1.5*(mat2gray(ba));

subplot(2,3,1); imshow(en); title('Original');
subplot(2,3,4); imshow(blur); title('7x7 Average Filter');
subplot(2,3,2); imshow(unb01); title('Unblurred Image (d = 0.05)');
subplot(2,3,3); imshow(unb005); title('Unblurred Image (d = 0.08)');
subplot(2,3,5); imshow(unb002); title('Unblurred Image (d = 0.1)');
subplot(2,3,6); imshow(unb001); title('Unblurred Image (d = 0.15)');
```

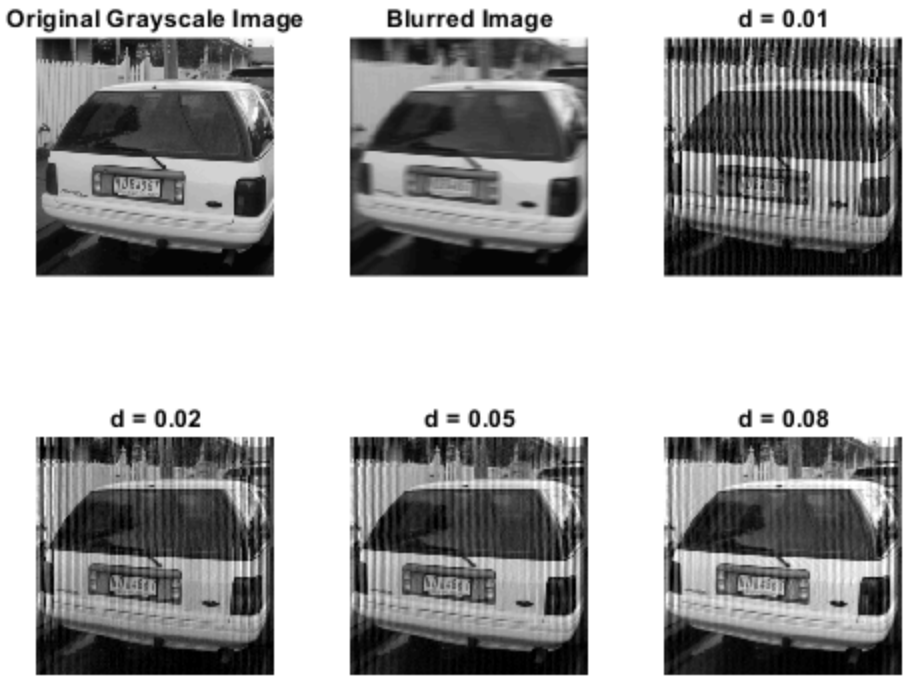


Ch 8: Problem 16

Experiment with different values of the threshold for motion blurring example. What gives the best results?

```
car = imread('car.png');     % 256 x 256 grayscale car
m = fspecial('motion',9,0); % Motion filter
cm = imfilter(car,m);        % Apply motion blur to car

m2 = zeros(size(car));       % create zero array
m2(1,1:9) = m;               % Put motion filter into zero array
mf = fft2(m2);               % Transform of motion filter

d = 0.01;                        % threshold
mf(find(abs(mf)<d)) = 1;         % find constraint values
bmi = abs(ifft2(fft2(cm)./mf)); % Apply inverse motion filter in
 frequency domain
```

```
t1 = 2*mat2gray(bmi);           % Convert to grayscale and double
 (contrast enhancement)

d = 0.02;
mf(find(abs(mf)<d)) = 1;
bmi = abs(ifft2(fft2(cm)./mf));
t2 = 2*mat2gray(bmi);
d = 0.05;
mf(find(abs(mf)<d)) = 1;
bmi = abs(ifft2(fft2(cm)./mf));
t3 = 2*mat2gray(bmi);
d = 0.08;
mf(find(abs(mf)<d)) = 1;
bmi = abs(ifft2(fft2(cm)./mf));
t4 = 2*mat2gray(bmi);

figure;
subplot(2,3,1); imshow(car); title('Original Grayscale Image');
subplot(2,3,2); imshow(cm); title('Blurred Image');
subplot(2,3,3); imshow(t1); title('d = 0.01');
subplot(2,3,4); imshow(t2); title('d = 0.02');
subplot(2,3,5); imshow(t3); title('d = 0.05');
subplot(2,3,6); imshow(t4); title('d = 0.08');
```



Ch 13: Problem 10

Add Gaussian noise to an RGB color image x. View image and remove the noise using average and Wiener filtering on each RGB component.

```
clear; clc; close all;
x = imread('iris.png');

figure(1);
subplot(2,2,1); imshow(x);
title('Original Color Image');
subplot(2,2,2); imshow(x(:,:,1));
title('Red Component');
subplot(2,2,3); imshow(x(:,:,2));
title('Green Component');
subplot(2,2,4); imshow(x(:,:,3));
title('Blue Component');

xn = imnoise(x,'gaussian');

figure(2); sgtitle('Images with Gaussian Noise Added');
subplot(2,2,1); imshow(xn);
title('Original Color Image');
subplot(2,2,2); imshow(xn(:,:,1));
title('Red Component');
subplot(2,2,3); imshow(xn(:,:,2));
title('Green Component');
subplot(2,2,4); imshow(xn(:,:,3));
title('Blue Component');

% Applying average filter to RGB
n = zeros(size(xn));
t = {'Red','Green','Blue'};
for k = 1:3;
    n(:,:,k) = mat2gray(imfilter(xn(:,:,k),fspecial('average',3)));
    figure(3); sgtitle('Average Filters');
  %  subplot(2,2,1); imshow(n); title('Original');
    subplot(2,2,k); imshow(n(:,:,k))
    title((t(k)));
end;
subplot(2,2,4); imshow(n); title('Filtered Color Image');

% Applying Wiener filter to RGB
n = zeros(size(xn));
t = {'Red','Green','Blue'};
for k = 1:3;
    n(:,:,k) = mat2gray(wiener2(xn(:,:,k)));
    figure(4); sgtitle('Wiener Filters');
   % subplot(2,2,1); imshow(wiener2(gg)); title('Original Grayscale');
    subplot(2,2,k); imshow(n(:,:,k))
    title((t(k)));
end;
subplot(2,2,4); imshow(n); title('Filtered Color Image');
```

Original Color Image

Red Component

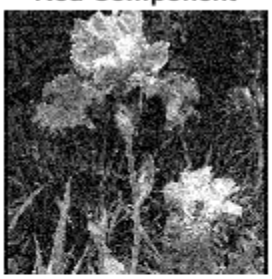
Green Component

Blue Component

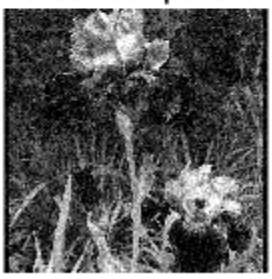
Images with Gaussian Noise Added

Original Color Image

Red Component



Green Component



Blue Component

Average Filters

Red

Green

Blue

Filtered Color Image

Wiener Filters

Red

Green

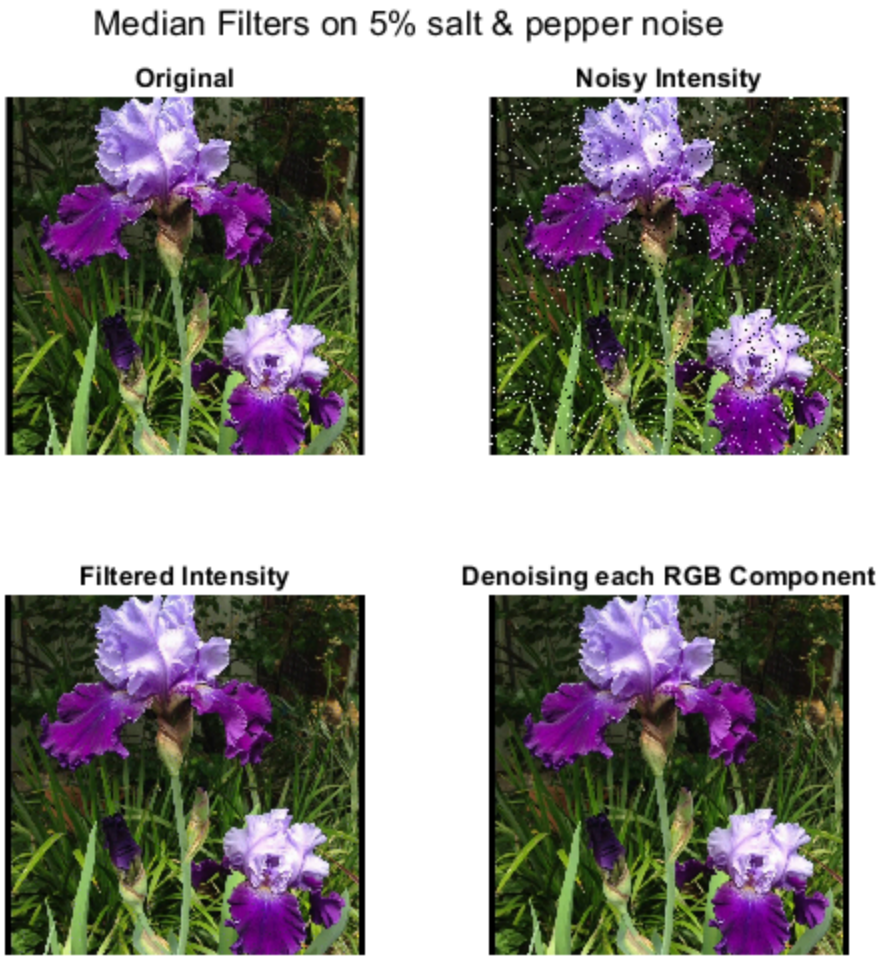
Blue

Filtered Color Image

Ch 13: Problem 11

Add salt and pepper noise to the intensity component of a color image. Convert back to RGB for display and compare to Figure 13.21. Denoise using median filter for each RGB components. Which one gives the best results? Experiment with larger amounts of noise and with Gaussian noise.

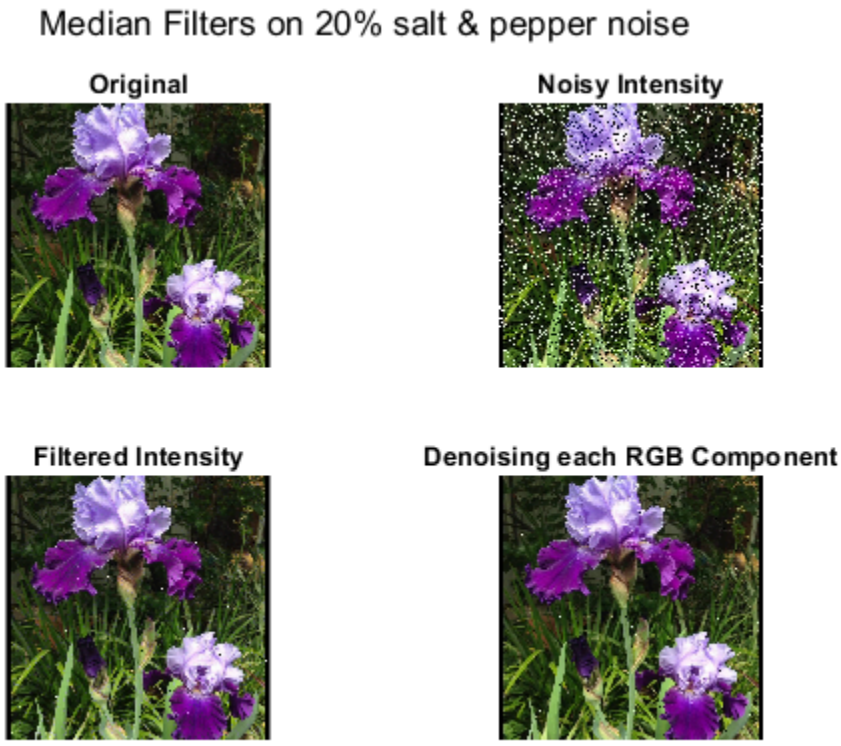
```
clear; clc; close all;
tw = imread('iris.png');
ty = rgb2ntsc(tw);
tn = imnoise(ty(:,:,1),'salt & pepper');
ty(:,:,1) = tn;
rgb = ntsc2rgb(ty);
subplot(2,2,1);imshow(rgb); title('Noisy Intensity');
subplot(2,2,2);imshow(rgb(:,:,1)); title('Red');
subplot(2,2,3);imshow(rgb(:,:,2)); title('Green');
subplot(2,2,4);imshow(rgb(:,:,3)); title('Blue');

% Filtering the Intensity
close all;
c = rgb2ntsc(rgb);
c(:,:,1)= medfilt2(c(:,:,1),[3,3]);
r = ntsc2rgb(c);
imshow(r); title('Filtered Intensity');

figure(1);
subplot(2,2,1); imshow(tw); title('Original');
subplot(2,2,2); imshow(rgb); title('Noisy Intensity');
subplot(2,2,3); imshow(r); title('Filtered Intensity');

% Filtering RGB Components
n = zeros(size(rgb));
for k = 1:3;    % 3x3 Median filter applied to each primary color
    n(:,:,k) = mat2gray(medfilt2(rgb(:,:,k),[3,3]));
    figure(1); sgtitle('Median Filters on 5% salt & pepper noise');
    subplot(2,2,4); imshow(n)
    title('Denoising each RGB Component');
end;
```



20% Salt & Pepper Noise Density

```
clear; clc; close all;
tw = imread('iris.png');
ty = rgb2ntsc(tw);
tn = imnoise(ty(:,:,1),'salt & pepper',0.2);
ty(:,:,1) = tn;
rgb = ntsc2rgb(ty);

% Filtering the Intensity
c = rgb2ntsc(rgb);
c(:,:,1)= medfilt2(c(:,:,1),[3,3]);
r = ntsc2rgb(c);

figure(2);
subplot(2,2,1); imshow(tw); title('Original');
subplot(2,2,2); imshow(rgb); title('Noisy Intensity');
subplot(2,2,3); imshow(r); title('Filtered Intensity');
```

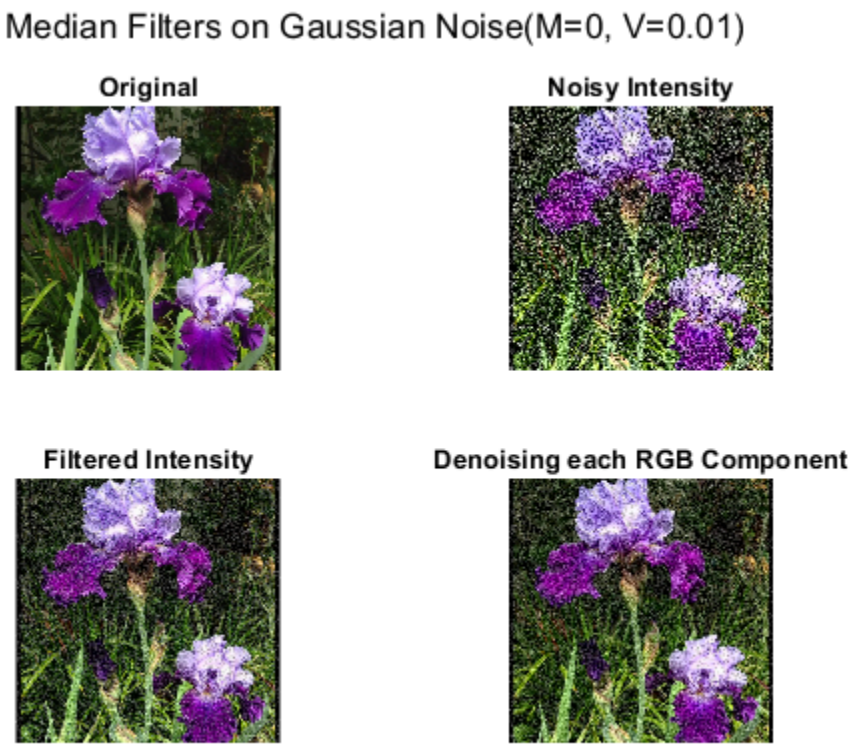
```
% Filtering RGB components
n = zeros(size(rgb));
for k = 1:3;    % 3x3 Median filter applied to each primary color
    n(:,:,k) = mat2gray(medfilt2(rgb(:,:,k),[3,3]));
    figure(2); sgtitle('Median Filters on 20% salt & pepper noise');
    subplot(2,2,4); imshow(n)
    title('Denoising each RGB Component');
end;
```



Gaussian Noise

```
clear; clc; close all;
tw = imread('iris.png');
ty = rgb2ntsc(tw);
tn = imnoise(ty(:,:,1),'gaussian',0,0.1);
ty(:,:,1) = tn;
rgb = ntsc2rgb(ty);

% Filtering the Intensity
c = rgb2ntsc(rgb);
c(:,:,1)= medfilt2(c(:,:,1),[3,3]);
r = ntsc2rgb(c);

figure(1);
subplot(2,2,1); imshow(tw); title('Original');
subplot(2,2,2); imshow(rgb); title('Noisy Intensity');
```

```
subplot(2,2,3); imshow(r); title('Filtered Intensity');

% Filtering RGB components
n = zeros(size(rgb));
t = {'Red','Green','Blue'};
for k = 1:3;    % 3x3 Median filter applied to each primary color
    n(:,:,k) = mat2gray(medfilt2(rgb(:,:,k),[3,3]));
    figure(1); sgtitle('Median Filters on Gaussian Noise(M=0,
 V=0.01)');
    subplot(2,2,4); imshow(n)
    title('Denoising each RGB Component');
end;
```




```
close all;
```

Published with MATLAB® R2019b