

EE301 Lab8

Javier Macossay-Hernandez, USC ID 9907093913, Lab Section 30813

```
clc
clear all
close all

load('lab8_data.mat')

%%Question 1

%part a
hh = 1/7*ones(7,1); %the 7 point averaging filter

yy1 = filter(hh,1,simple);
yy2 = filter2(hh,simple,'same');
figure

subplot(3,1,1)
plot(simple)
title('simple')

subplot(3,1,2)
plot(yy1);
title('filter')

subplot(3,1,3)
plot(yy2);
title('filter 2')

MS1=sum(abs(yy1-simple).^2)/length(simple)
MS2=sum(abs(yy2-simple).^2)/length(simple)

%part b

%here we measure the noise power by comparing the MS between the clean and
%noise signal
MS3 = sum(abs(simple_noise-simple).^2)/length(simple)
figure
subplot(3,1,1)
plot(simple)
subplot(3,1,2)
plot(simple_noise)

yy3 = filter2(hh,simple_noise,'same');
%here we compare the output of the noisy signal after filter with the clean
%signal
MS4 = sum(abs(yy3-simple).^2)/length(simple)
subplot(3,1,3)
```

```
plot(yy3)

%the noise power after the filter, is the difference between the
%(signal+noise) error, and the output distortion just distorted signal
Noise_before_filter = MS3
Noise_after_filter = abs(MS4-MS2)

%the noise power before the filter was -19, and after the filter is 4.6
%Thus, the noise power went down with penalty of distortion.

%the total error is
total_error = MS4

%the error from the noise is
Noise_after_filter = abs(MS4-MS2)

%the error from the distortion
error_from_distortion = MS2

%the filter distortion is the most dominant error source

%part c
figure
subplot(4,2,[1 2])
plot(simple)

%3 point filter
hh3 = ones(3,1)/3
yyhh3 = filter2(hh3, simple_noise,'same');
subplot(4,2,3);
plot(yyhh3)
subplot(4,2,4);
stem(hh3); % the filter coefficients

%5 point filter
hh5 = ones(5,1)/5
yyhh5 = filter2(hh5, simple_noise,'same');
subplot(4,2,5);
plot(yyhh5)
subplot(4,2,6);
stem(hh5); % the filter coefficients

%9 point filter
hh9 = ones(9,1)/9
yyhh9 = filter2(hh9, simple_noise,'same');
subplot(4,2,7);
plot(yyhh9)
subplot(4,2,8);
stem(hh9); % the filter coefficients

MSE_forN3 = sum(abs(yyhh3-simple).^2)/length(simple)
MSE_forN5 = sum(abs(yyhh5-simple).^2)/length(simple)
MSE_forN7 = MS4 %from the previous part
```

```
MSE_forN9 = sum(abs(yyhh9-simple).^2)/length(simple)

%the best filter is N3 which has the lowest mean error. Going through the
%filter was worse than not going through the filter.

%part d
figure
subplot(4,2,[1 2])
plot(simple)

%smooth with 0.5
gh05=g_smooth(0.5)
yygh05 = filter2(gh05,simple_noise,'same');
subplot(4,2,3)
plot(yygh05)
subplot(4,2,4)
stem(gh05) %the filter coefficient

%smooth with 0.75
gh075 = g_smooth(0.75)
yygh075 = filter2(gh075,simple_noise,'same');
subplot(4,2,5)
plot(yygh075)
subplot(4,2,6)
stem(gh075) % the filter coefficient

%smooth with 1
gh1=g_smooth(1)
yygh1=filter2(gh1,simple_noise,'same');
subplot(4,2,7)
plot(yygh1)
subplot(4,2,8)
stem(gh1) % the filter coefficient

MSE_forgh05 = sum(abs(yygh05-simple).^2)/length(simple)
MSE_forgh075 = sum(abs(yygh075-simple).^2)/length(simple)
MSE_forgh1 = sum(abs(yygh1-simple).^2)/length(simple)

%the best filter is gh05 because the results yield that it has the lowest
%mean error.

%%Question 2
%part a
pep = imread('peppers.tif');
pep_n1 = imread('peppers_noise1.tif');
figure
imagesc(pep);
figure
imagesc(pep_n1);

%Filter for the 2D image, this is the first filter
hhpic1=g_smooth2(5);
figure
```

```

subplot(1,2,1)
imagesc(hhpic1);
subplot(1,2,2)
surf(hhpic1);

%part b
figure
subplot(1,2,1)
imagesc(pep);
subplot(1,2,2)
imagesc(pep_n1);

%This line is to find the size of the picture (rows and columns)
[a1,a2]=size(pep);
MSpic1=sum(sum((pep-pep_n1).^2))/(a1*a2)

%part c
l=0.1:0.2:10.1;
pep_n1=im2double(pep_n1);
pep=im2double(pep);

for vvv=1:length(l)
    hhloop=g_smooth(l(vvv));
    yyloop=(filter2(hhloop,pep_n1,'same'));
    MSloop(vvv)=sum(sum((yyloop-pep).^2))/(a1*a2);
end

figure
plot(l,MSloop)

%taking into consideration all the filters and by comparing them, I
%conclude that the optimum filter is 1.1

hhloop=g_smooth(1.1);
yyloop=(filter2(hhloop,pep_n1,'same'));
figure
subplot(1,2,1)
imagesc(pep_n1)
subplot(1,2,2)
imagesc(yyloop)

%%Question 3

%part a
pep_n2=imread('peppers_noise2.tif');
figure
imagesc(pep_n2)
pep_n2d = im2double(pep_n2);

double_pep_n2 = im2double(pep_n2);
MSpic2 = sum(sum((pep-double_pep_n2).^2))/(a1*a2);
disp('The MSE between the peppers and the peppers with salt and pepper noise is the following
value:')

```

```

disp(MSpic2)

%part b
hhpic2=g_smooth2(1.3);
yypic2=filter2(hhpic2,pep_n2d,'same');
figure
subplot(1,2,1)
imagesc(pep_n2d)
subplot(1,2,2)
imagesc(yypic2)

MSpic2_lowpass = sum(sum((pep-yypic2).^2))/(a1*a2);
disp('The MSE between the smoothing filters image and the peppers is the following value:')
disp(MSpic2_lowpass)

%part c
N = 3;
yy3 = medfilt1(medfilt1(pep_n2d,N),'N');
N = 5;
yy5 = medfilt1(medfilt1(pep_n2d,N),'N');

figure
subplot(1,2,1)
imagesc(yy3)
subplot(1,2,2)
imagesc(yy5)

MSpic2_med3 = sum(sum((pep-yy3).^2))/(a1*a2);
MSpic2_med5 = sum(sum((pep-yy5).^2))/(a1*a2);
disp('The MSE between the median filter (N=3) image and the peppers is the following value:')
disp(MSpic2_med3)
disp('The MSE between the median filter (N=5) image and the peppers is the following value:')
disp(MSpic2_med5)

%The filter of order 5 looks almost as the original. In contrast, filter of
%order 3 leaves some salt and pepper noise on the image.

%In the case of the median filters, they remove the salt and pepper noise.
%In the case of the smoothing filters, the black and white dots where the
%pepper and salt were located at ended up having distortion.

```

```

MS1 =

    160.3250

```

```

MS2 =

    43.6464

```

```

MS3 =

```

19.7631

MS4 =

48.3135

Noise_before_filter =

19.7631

Noise_after_filter =

4.6671

total_error =

48.3135

Noise_after_filter =

4.6671

error_from_distortion =

43.6464

hh3 =

0.3333

0.3333

0.3333

hh5 =

0.2000

0.2000

0.2000

0.2000

0.2000

hh9 =

0.1111

```
0.1111
0.1111
0.1111
0.1111
0.1111
0.1111
0.1111
0.1111
```

```
MSE_forN3 =
```

```
24.5223
```

```
MSE_forN5 =
```

```
36.2779
```

```
MSE_forN7 =
```

```
48.3135
```

```
MSE_forN9 =
```

```
60.9562
```

```
gh05 =
```

```
0.1065
0.7870
0.1065
```

```
gh075 =
```

```
0.0152
0.2188
0.5321
0.2188
0.0152
```

```
gh1 =
```

```
0.0044
0.0540
0.2420
0.3991
0.2420
```

0.0540
0.0044

MSE_forgh05 =

14.9318

MSE_forgh075 =

16.9504

MSE_forgh1 =

21.4572

MSpic1 =

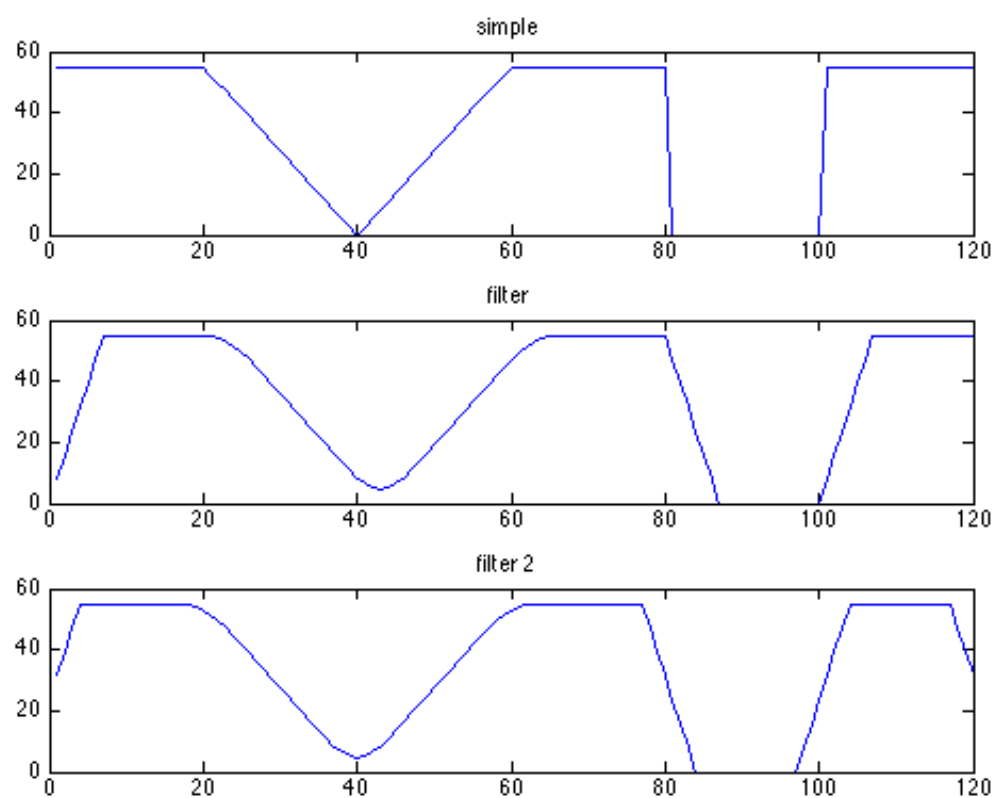
78.5430

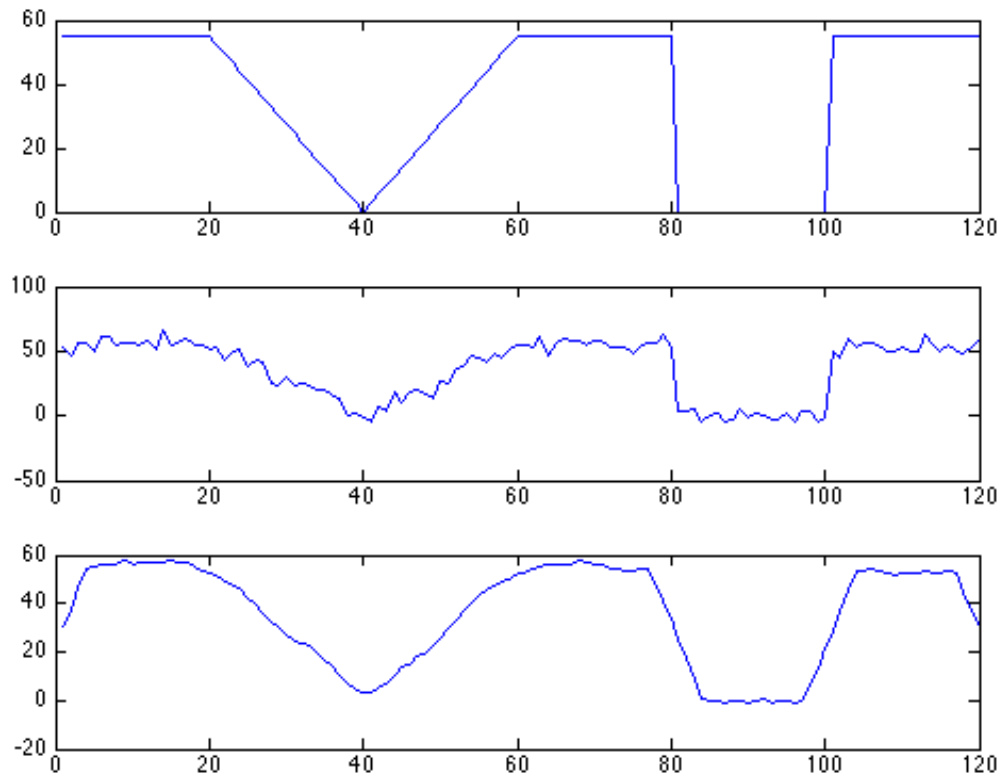
The MSE between the peppers and the peppers with salt and pepper noise is the following value:
0.0676

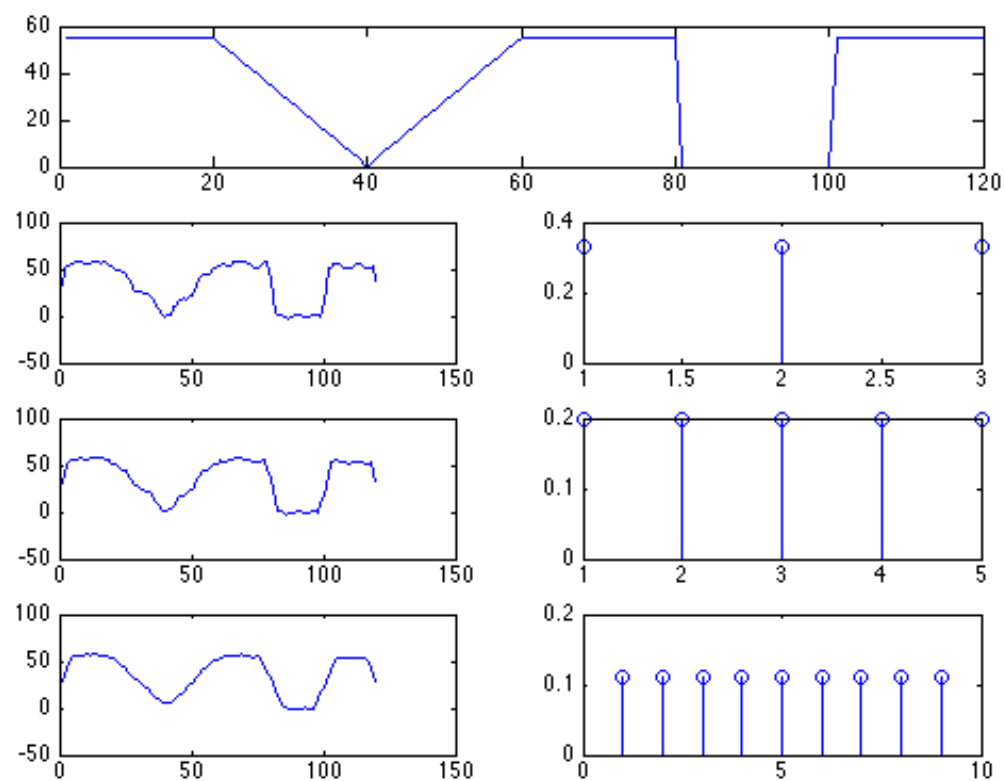
The MSE between the smoothing filters image and the peppers is the following value:
0.0099

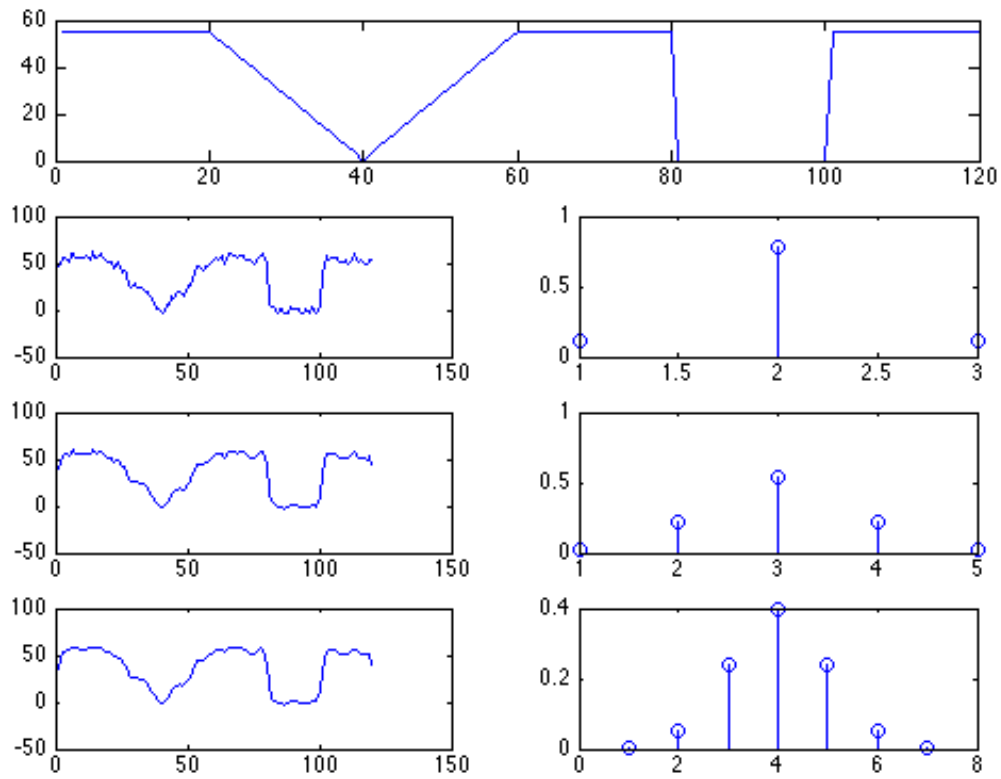
The MSE between the median filter (N=3) image and the peppers is the following value:
0.0031

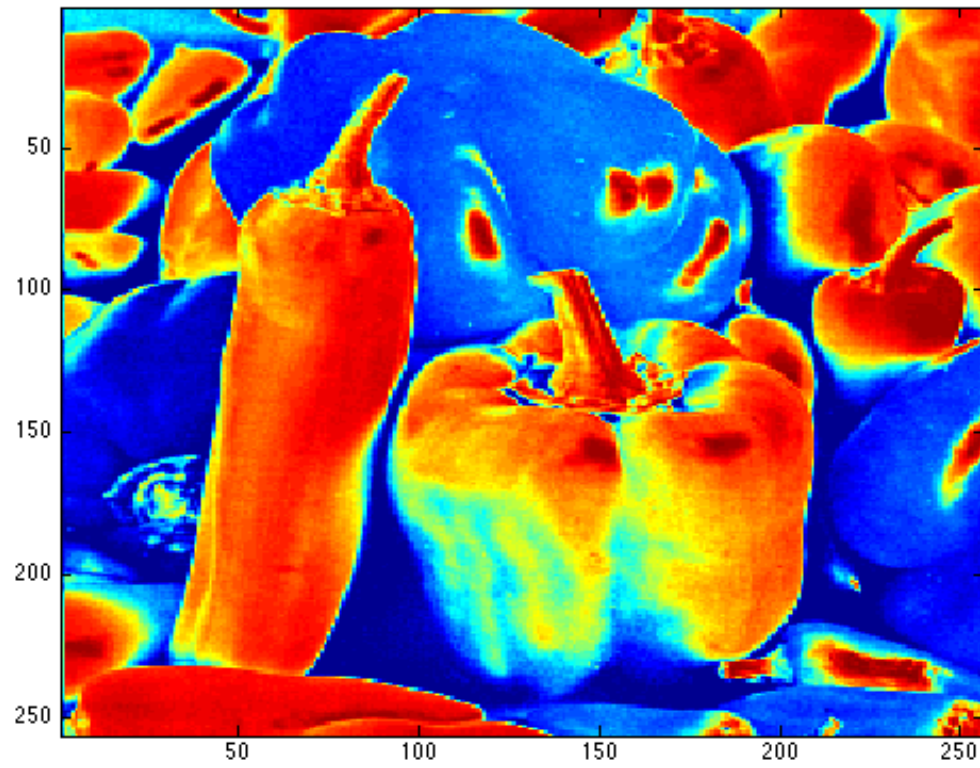
The MSE between the median filter (N=5) image and the peppers is the following value:
0.0027

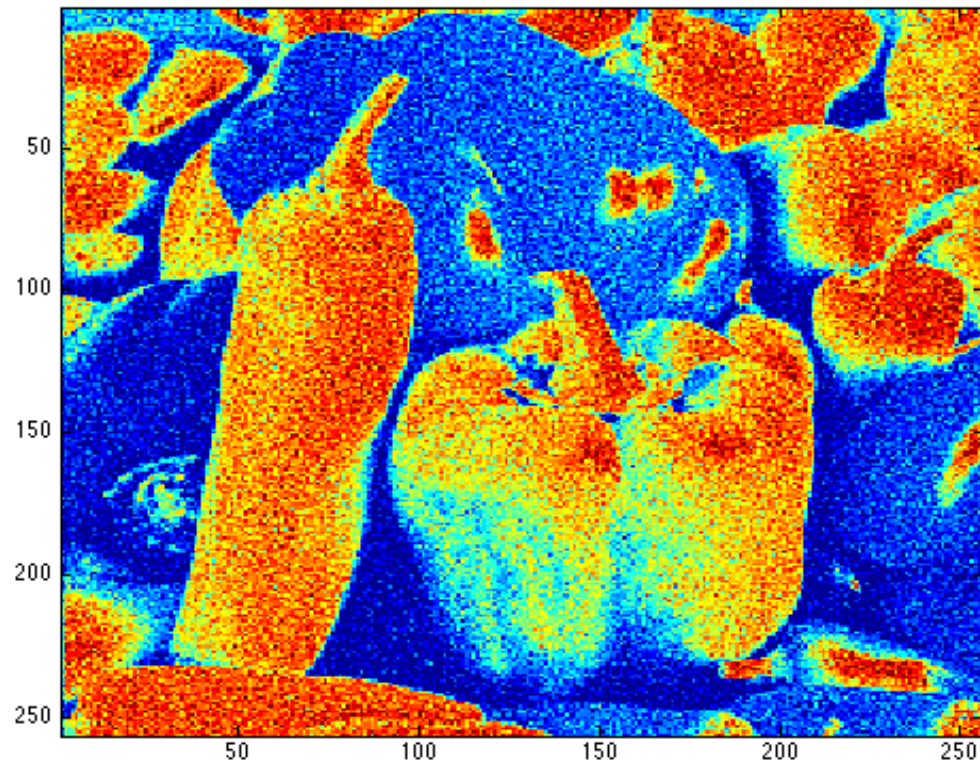


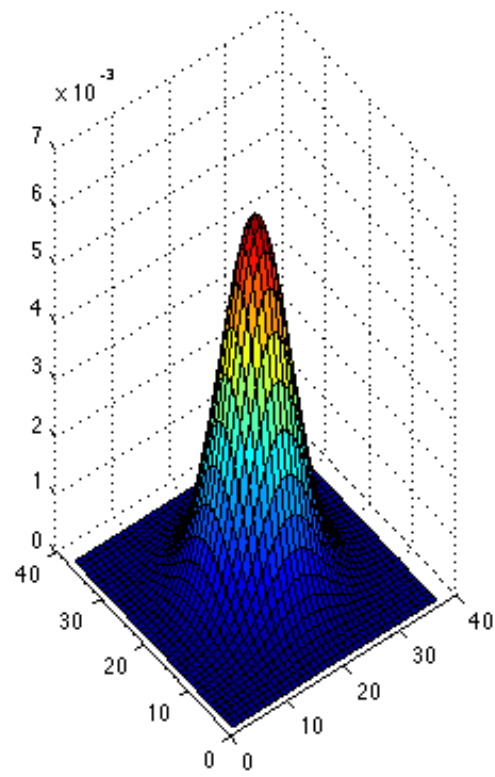
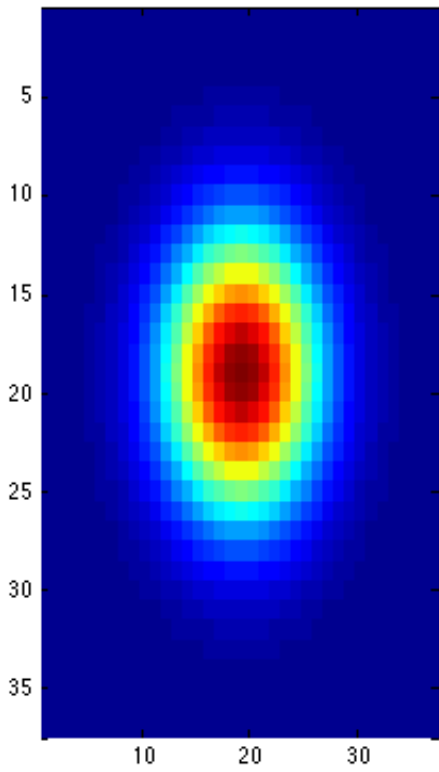


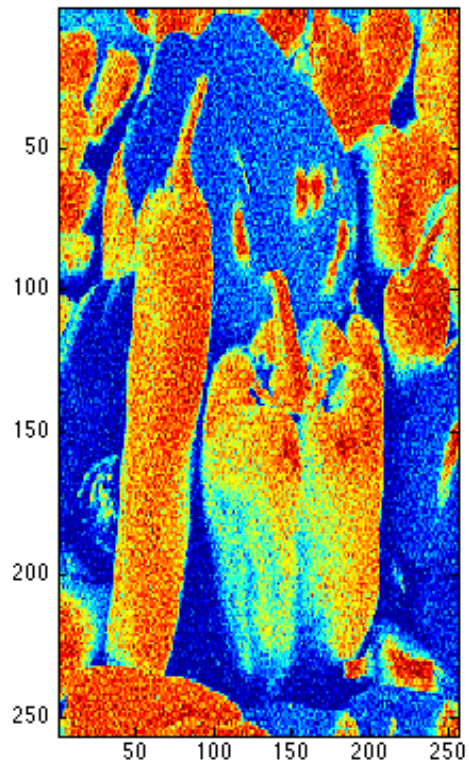
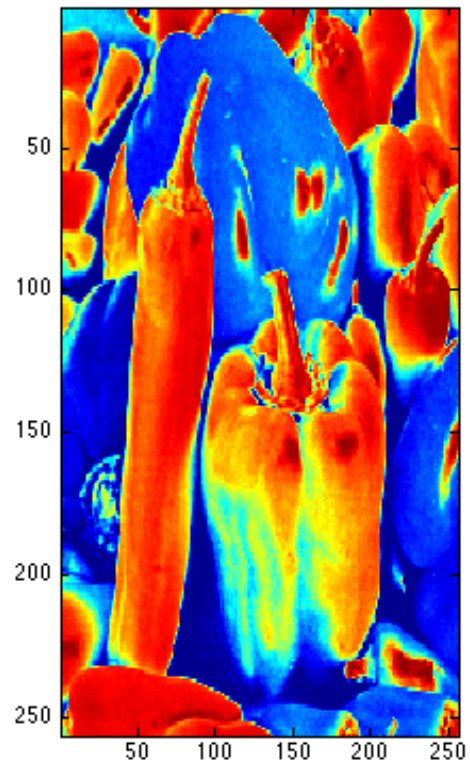


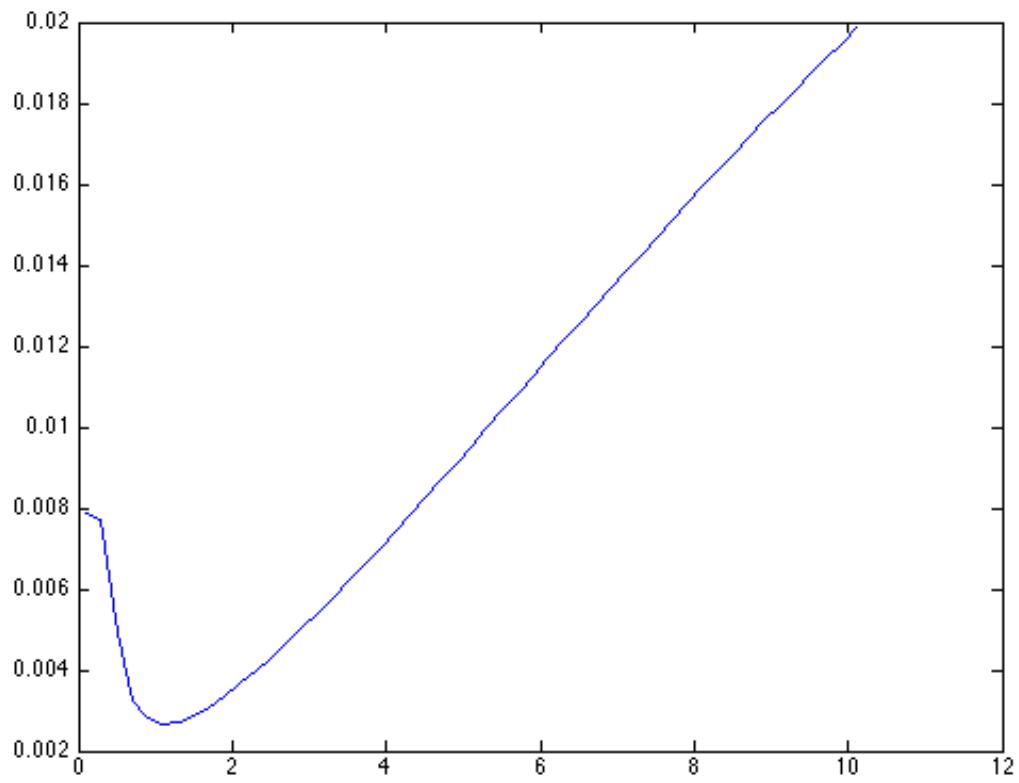


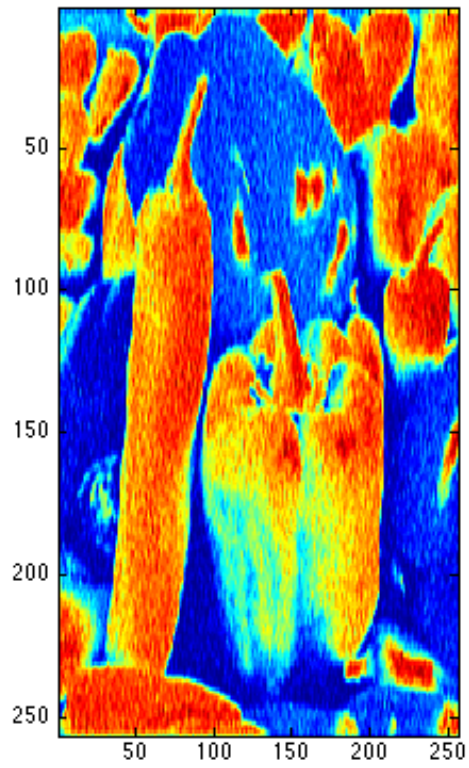
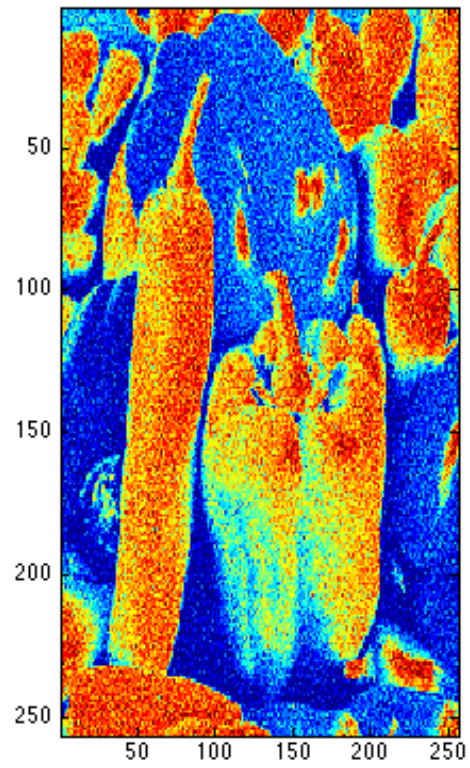


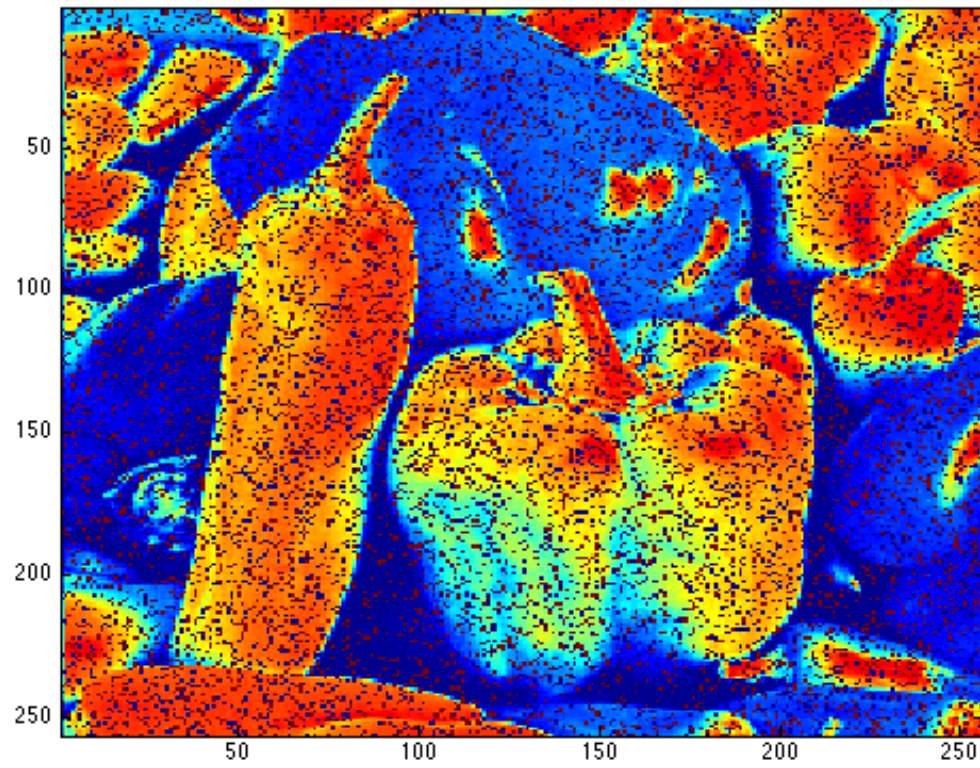


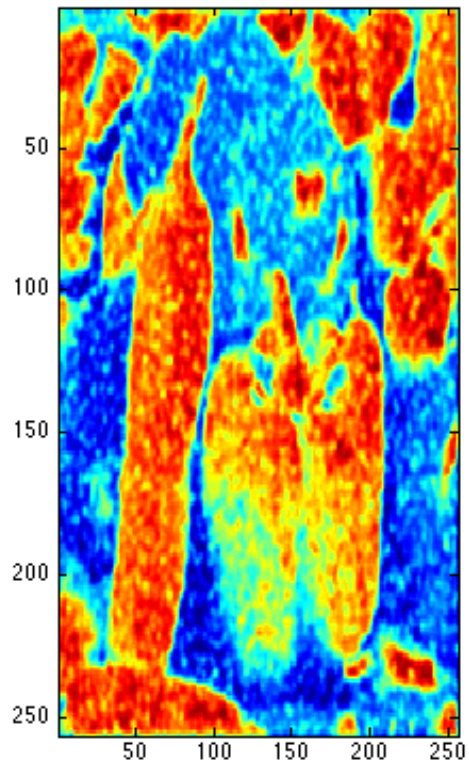
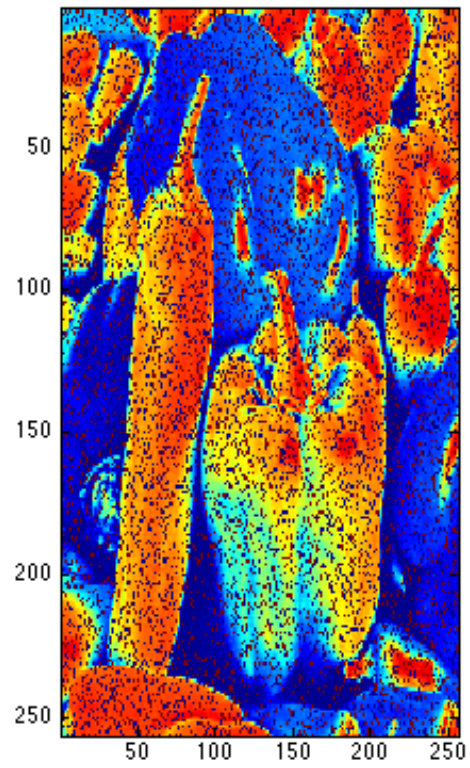


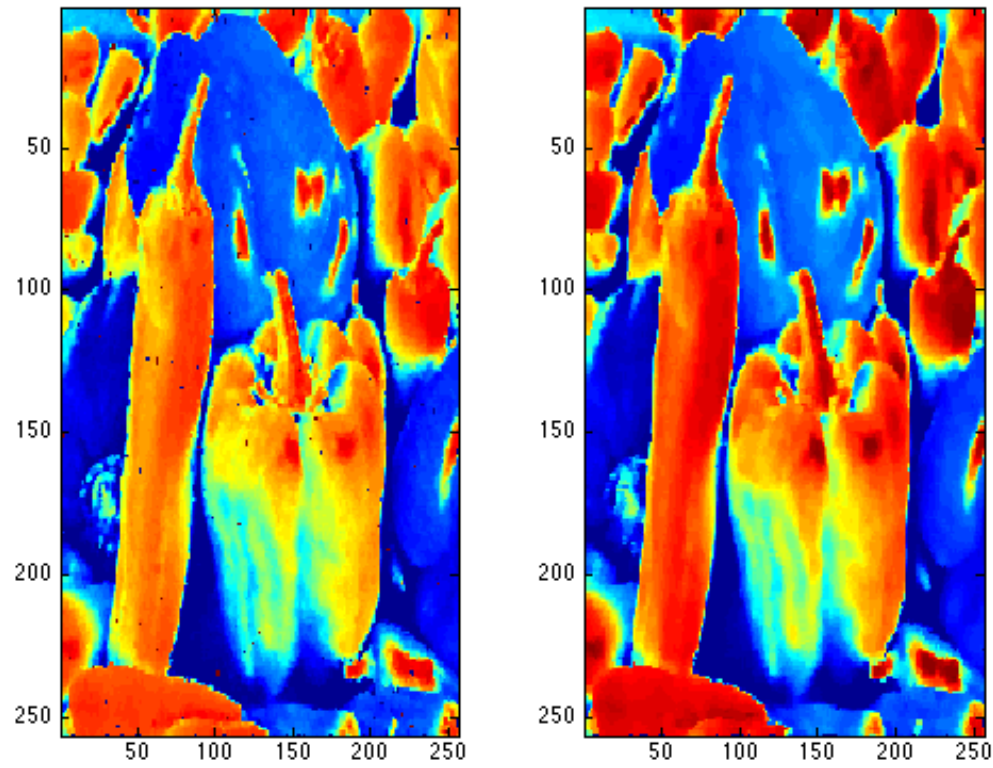












Published with MATLAB® R2013b