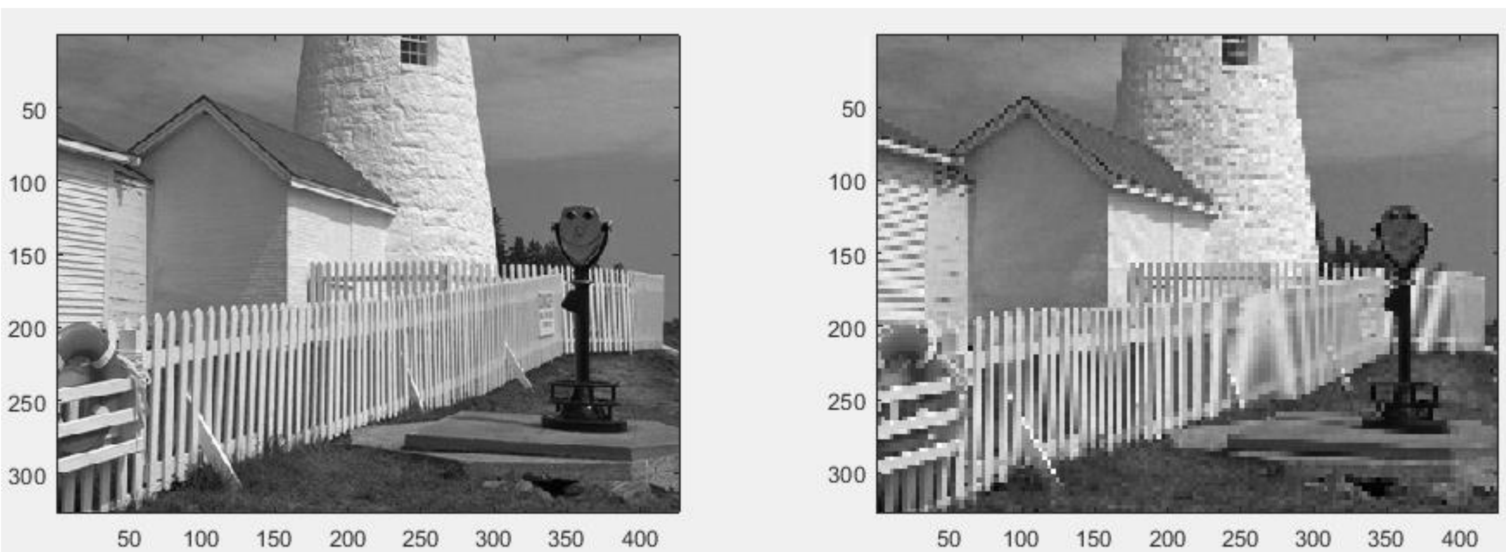
```
diary on
format compact
%Johnny Li
%EEL3135 Fall 2018
%Lab 4 Part 2

%2.1

%Plot the vectors hold1 to verify that it is indeed a zero-order hold
%derived from row.

%Code Given
row = (-2).^(0:6);
L = length(row);
nn= ceil((0.999:1:4*L)/4); %<---Round up to the integer
hold1 = row(nn);

%Question: What values are in the indexing vector nn, and why are they what
%they are?
%The values in the indexing vector nn is the length of the row, from 1 to 7.
%The length values index is called four times per a row value, by doing
%so, a consistence reconstruction of the vector is created in each section.
%Every point in the decimated array is duplicated in the row and column
%directions without change thus creating a square pulse.

type Interpolate

%Interpolate Interpolate a Lighthouse
%Script for 2.1.

%2.1.2.1

%Load the lighthouse image data from lighthouse.mat
load('lighthouse.mat')

%2.1.2.2

%Down-sample the lighthouse image by a factor of 3, call the new array x3.
x3 = xx(1:3:end,1:3:end);

%2.1.2.3

%Perform a zero-order hold on x3 to fill in the missing points:
%For an interpolation factor of 3, process all rows of x3 to fill in
%missing points in that direction. Call the result xrows.

row=size(x3,1);
col=size(x3,2);
nn= ceil((0.999:1:3*row)/3);
xrow=x3(nn,1:col);

%Dimension of xrow.
size(xrow)
```

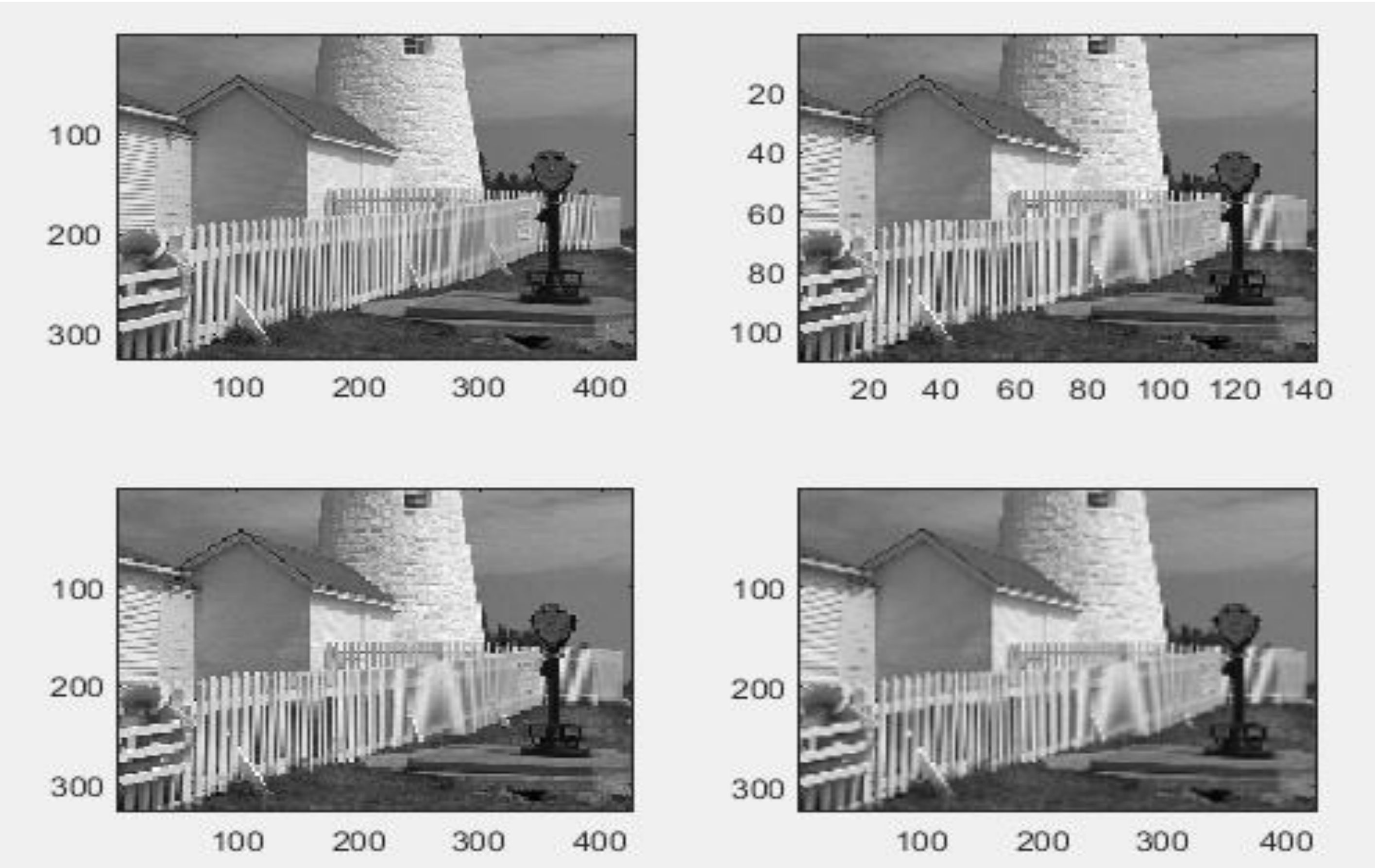
```
ans =

   327   142

%Questions: What are the dimensions of the xrows two-dimensional array?
%The dimensions of the xrows two-dimensional array is 327x142.

%2.1.2.4

%Process all the columns of xrows likewise (interpolation factor 3) to fill
% in the missing points in each column. Call this result xhold.
row=size(x3,1);
col=size(x3,2);
mm= ceil((0.999:1:3*col)/3.);
xhold=xrow(1:row*3,mm);

%Show the xhold (right) and original lighthouse (left) images on the same
%plot.
```




```
%Compare them and explain any differences that you can see.
%The orignial image (left) is very clear and detailed while the
%reconstructed image, xhold (right), is blurry and pixelated due to the
%repetitions.

%2.1.2.5

%Carry out linear interpolation operations on both the rows and the columns
%of the down-sampled lighthouse image x3.
%Previous values/code.
load('lighthouse.mat')
x3 = xx(1:3:end,1:3:end);
row=size(x3,1);
col=size(x3,2);

lrow=1:row;
```

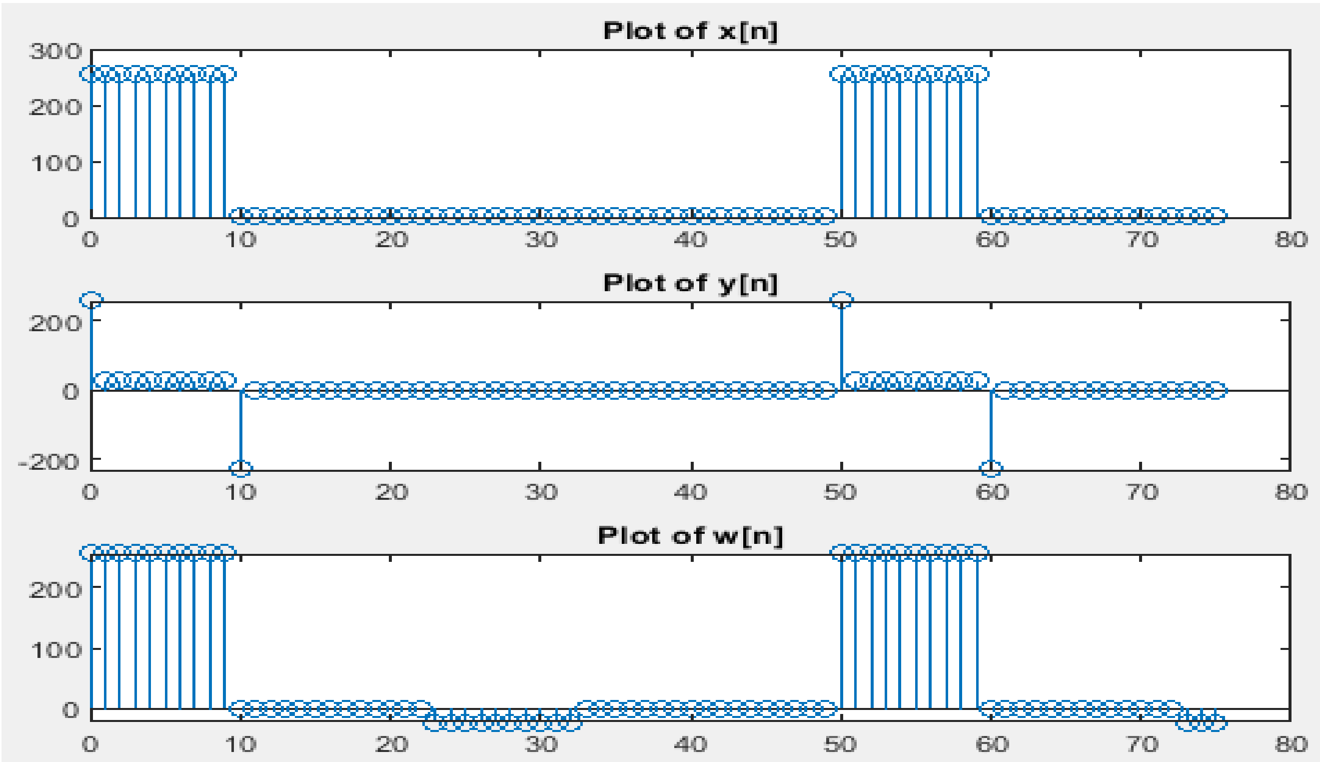
```
lcol=1:col;
%Ex:tt = 0:0.1:6; % <---locations between the n indices
ttr=1:1/3:length(lrow);
ttc=1:1/3:length(lcol);
%Apply linear interpolation operations.
rowlinear=interp1(lrow,x3,ttr);
%Transpose to fit dimension.
rowlineart=transpose(rowlinear);
%Apply linear interpolation operations.
holdlinear=interp1(lcol,rowlineart,ttc);
%Re-transpose to original dimension.
holdlineart=transpose(holdlinear);
show_img(holdlineart,0,1,'gray(255)');
```


```
%2.1.2.6

Show the original image, the down-sampled image, the zero-order-hold
%reconstructed image, and the linearly-interpolated reconstructed image in
%that order.
```

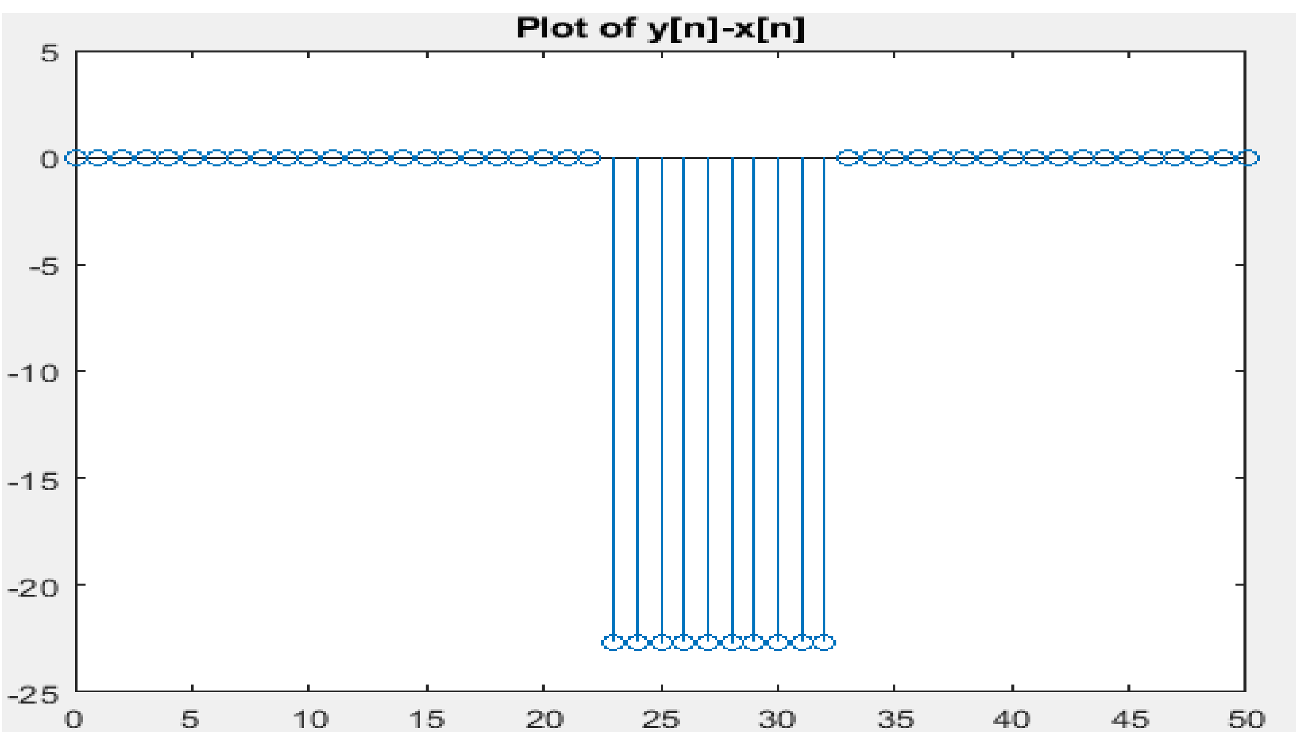



```
%Point out regions where the linear and zero-order reconstruction result
%images differ and try to justify this difference.
%In low frequency areas, such as the background and lighthouse, the linear
%interpolation reconstructed image looks better than the zero-order hold
%reconstructed image as it is more smoother and less pixelated. In high
%frequency areas, such as the fence due its sinusoidal repetition pattern,
%the linear interpolation reconstructed image is blurrier while the zero
%-order hold reconstructed image is more pixelated.

%Zero-order hold reconstruction, repeats certain parts in the image to fill
%in the missing portions resulting in a sharp transition between one set of
%values to another that forms defined borders, creating a pixelized image.
%In low frequency areas this pixelization is more noticeable as there is
%more variations in image while it is less noticeable in high frequency
%areas as there is less variations, long line of similar fences.

%Linear interpolation reconstruction, averages out the differences in the
%gap from the missing portions to fill it in, creating a smoother change.
%In low frequency areas there is a smooth transition between different
%pieces of the image and the color shift has improved from the zero-order
%hold reconstruction. In the high frequency areas, the image become blurrier
%because of the numerous gaps which has only slight deviations meaning the
%average is less definite.
```

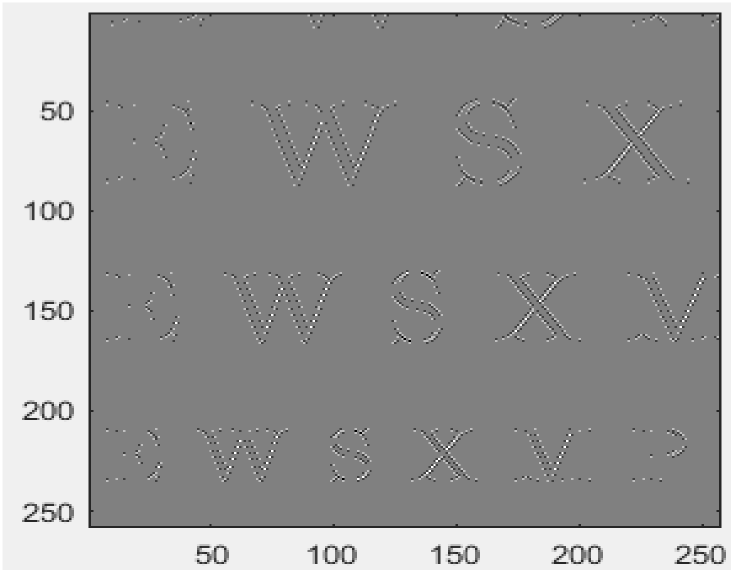
```
%Question: Are edges low-frequency or high-frequency features?
%The edges are low-frequency features.

%Question: Is the series of fence posts a low-frequency or high-frequency
%feature?
%The series of fence posts are a high-frequency feature.

%Question: Is the background a low-frequency or high-frequency feature?
%The background is a low-frequency feature.

type Restoration

%Restoration Restoration Filter for 1-D Data
%Script for 2.2

%2.2.1

%Implement an FIR filter on the following input signal:
%The function w[n] = x[n] - 0.9x[n-1] is an FIR filter.
%The convolution range is 1 to -0.9.
bb=[1,-0.9];
xx = 256*(rem(0:100,50)<10);
yy=firfilt(bb,xx);

%2.2.2

%Process the filtered signal w[n] from the previous section with this
%restoration filter. Use r=0.9 and M=22.
r=0.9;
M=22;
%Value from r and M
rm=0.9.^(0:M);
%yy taken from part 2.2.1.
ww=firfilt(rm,yy);

%2.2.3

%Plot x[n],w[n] ,y[n] and on the same figure.
%Make the discrete-time signal plots with stem plots but restrict the
%horizontal axis to the range 0<=n<=75.
%xx plot
subplot(3,1,1);
stem(0:75,xx(1:76));
title('Plot of x[n]');
%yy plot
subplot(3,1,2);
stem(0:75,yy(1:76));
title('Plot of y[n]');
%ww plot
subplot(3,1,3);
stem(0:75,ww(1:76));
title('Plot of w[n]');
```



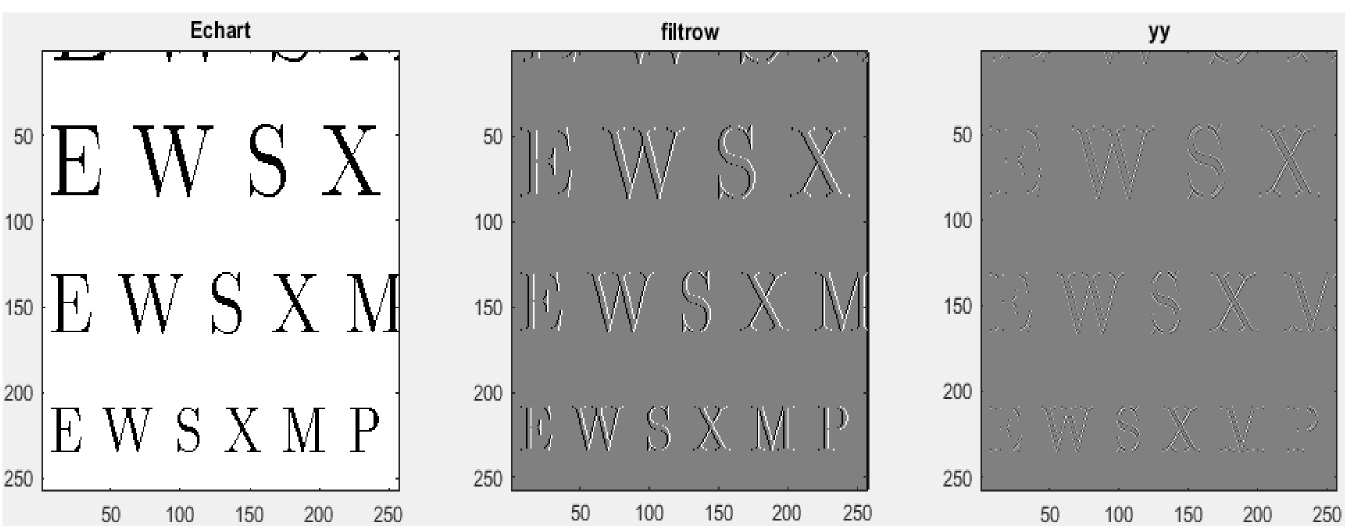

```
%2.2.4

%Make a plot of the error (difference) between y[n] and x[n] over the range
%0<=n<50.
diff=yy(1:51)-xx(1:51);
stem(0:50,diff);
title('Plot of y[n]-x[n]');
```



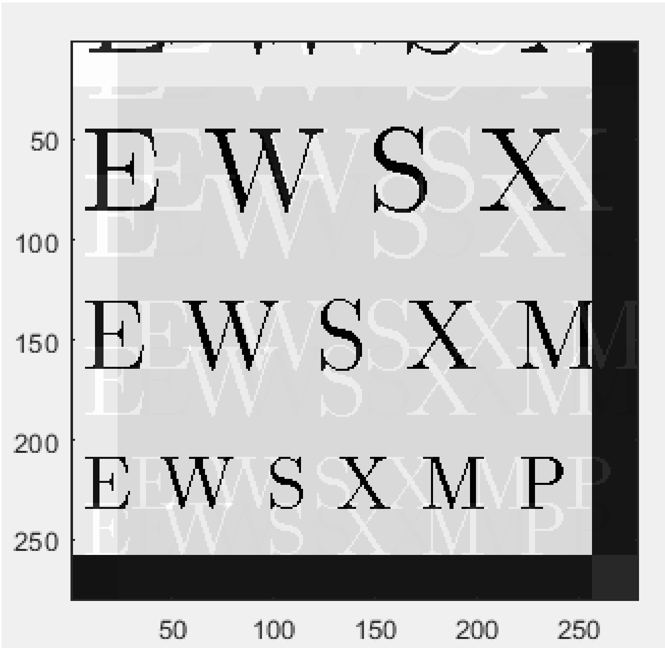
```
%2.2.5

%The maximum difference between the above y[n] and x[n] over the range
%0<=n<50.
%This is the worst-case error.
max(abs(diff))

ans =

   22.6891
%Max value is rounded to 22.7.

%2.2.6

%What does the error plot and worst-case error tell you about the quality
%of the restoration of x[n]?
%Since the error plots of x[n] and w[n] are very similar, the restoration
%of x[n] is quite accurate but the worst-case error indicates that there
%are slight deviations in the values from the original.

type Filtering

%Filtering Filtering Images with One-dimensional FIR filters.
%Script for 2.3.

%2.3.1.1

%Load in the echart image from the echart.m file.
load('echart.mat')

%Filter all the rows of the image with the conv2() function.
%To filter the image in the horizontal direction using a first-difference
%filter.
%Code given.
b = [1, -1];
filtrow=conv2(echart,b);

%2.3.1.2

%Filter the image in the vertical direction with this first-difference
%filter to produce the image y. This can be done by transposing the image
%then filtering the rows.
bt=transpose(b);
yy=conv2(filtrow,bt);
show_img(yy,0,1,'gray(256)');
```




```
%2.3.1.3

%Display the input image echart, the intermediate image row-filtered, and
%the output image y.
%Plot echart.
subplot(1,3,1);
show_img(echart,0,1,'gray(256)');
title('Echart');
%Plot Filtrow.
subplot(1,3,2);
show_img(filtrow,0,1,'gray(256)');
title('filtrow');
%Plot yy.
subplot(1,3,3);
show_img(yy,0,1,'gray(256)');
title('yy');
```



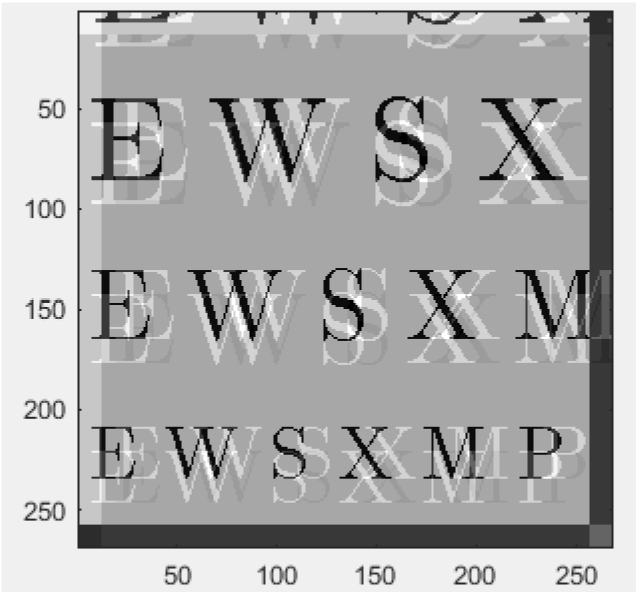
```
%Compare the three images and give a qualitative description of what you
%see.
%The original image, echart, is a very clear eye chart image with a strong
%contrast with the white background and black text. The image with the rows
%filter, filtrow, became grayed out image of the original which makes it
%difficult to read the text due to the lack of contrast with the text and
%background. The image with both the rows and columns filter, yy, is a
%very dark image from the original, which makes it super difficult to read
%the text as there is barely any contrast with the text and background to
%make out the text.

%2.3.2.1
%Load the echart image from the echart.m module. Pick q=0.9 for the first
%filter (FIR 1). Call the result echo90.
load('echart.mat')
q=0.9;
%Vector form.
qv=[1,-q];

%Using this filter, filter the echart image along the horizontal direction.
echo90a=conv2(echart,qv);
%Transpose the vector.
qvt=transpose(qv);
%Filter the resulting image vertically.
echo90=conv2(echo90a,qvt);

%2.3.2.2

%Convolve echo90 with FIR 2, choosing M=22 and r=0.9.
M=22;
r=0.9;
%Vector form.
mv=0:1:M;
%Produce r^(l)
rm=r.^mv;
%Convolve back to original.
deconv1=conv2(echo90,rm);
%Transpose the vector.
rmt=transpose(rm);
deconv2=conv2(deconv1,rmt);
%Display image.
show_img(deconv2,0,1,'gray(256)');
```



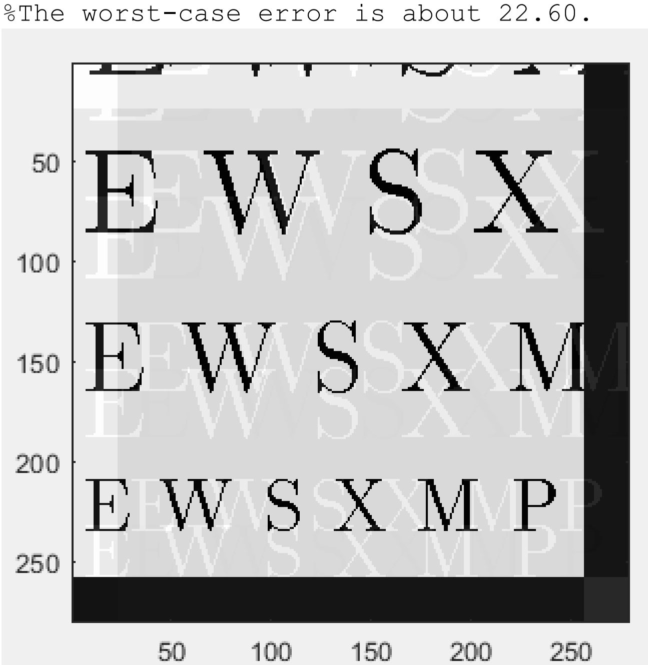

```
%Describe the visual appearance of the output qualitatively, showing the
%image, and explain its features by invoking mathematical understanding
%of the cascade filtering process and why you see “ghosts” in the output
%image.
%The visual output image has relativity clear and same size letter as the
%original image but also has visual 'echoes'. The visual echoes cause
%a repetition of the letters in different position known as ghosts as they
%are more faded and less visible. The interpolation of the image determines
%the position and color of the ghost where the first interpolation led to
%the shift in the horizontal axis, moving it to the right, while the second
%interpolation led to the shift in the vertical axis, moving it down. The
%cascading of interpolation results in overlaying shades with the respected
%shift of the image as since the ghost images heads downward right, the
%upper left area is left alone since there is no overlapping filter but the
%bottom right the two-overlapping filter therefore it gets increasingly
%darker.

%Determine the size and location of the “ghosts” relative to the original
%image.
%The size of the ghosts is the same as the original image, thus size of
%280x279 pixels.
%The location of the first ghosts, horizontal shift, is shifted 23 pixels
%to the right from the original. The location of the second ghosts,
%vertical shift, is shifted 23 pixels down from the original.

%2.3.2.3

%Evaluate the worst-case error.
err=echart(1:255)-deconv2(1:255);
max(abs(err))

ans =
```

```
  22.6005

%Worst-case error=22.6005.

%2.3.3.1

%Now try to deconvolve echo90 with several different FIR filters for FIR 2.
%Set r=0.9 and try several values for M, at the least M=11,22,33.
rr=0.9;
MM=11; %MM=22;  %MM=33;
%Vector form.
mmv=0:1:MM;
%Produce r^(l)
rmm=r.^mmv;
%Convolve back to original.
deconv1=conv2(echo90,rmm);
%Transpose the vector.
rmmt=transpose(rmm);
deconv2=conv2(deconv1,rmmt);
%Display image.
show_img(deconv2,0,1,'gray(256)');

%Pick the best result and explain why it is the best. Describe the visual
%appearance of the output qualitatively, and explain its features by
%invoking mathematical understanding of the cascade filtering process.
%The best result came from M=33 as its reconstruction resulted in the most
%clear and detailed image, the effect of the echoes was lessened and there
%was a more definite contrast with the background and letters.
%This is because the reconstruction goes through more values as M is higher
%which enables a more accurate result. The cascading of images results in
%the ghosts in the image but since the ghosts are barely visible, the
%shift from the cascade is almost a full phase from the original which
%would result in producing an original image. However, increasing M also
%increases the size of the darken portion of the image.

%2.3.3.2

%Question: How large is the worst-case error (from the previous part) in
%terms of number of gray levels?
%Evaluate the worst-case error.
err=echart(1:255)-deconv2(1:255);
max(abs(err))

ans =

    7.0923

%The worst-case error is about 7.09.

%Question: Evaluate worst-case error for each of the three filters in the
%previous section.
%M=11;
err=echart(1:255)-deconv2(1:255);
max(abs(err))
ans =
```

```
    72.0195

%The worst-case error is about 72.02.
```

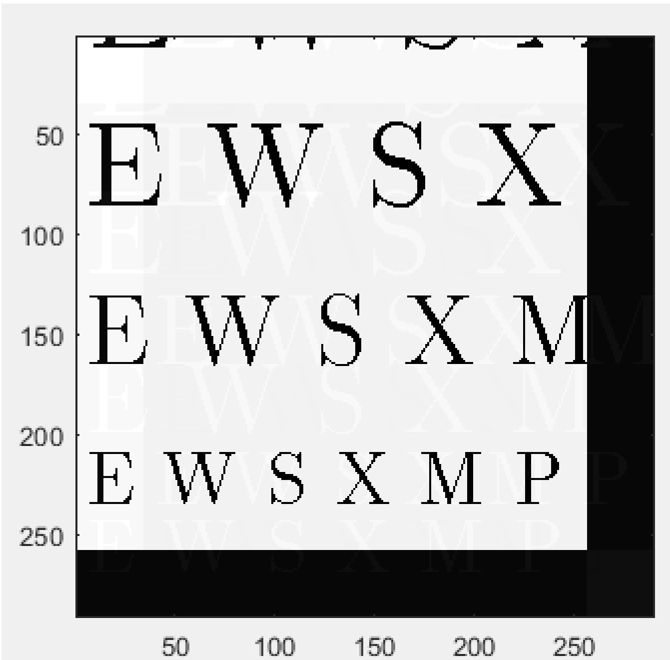



```
%M=22;
err=echart(1:255)-deconv2(1:255);
max(abs(err))
ans =

    22.6005

%The worst-case error is about 22.60.
```

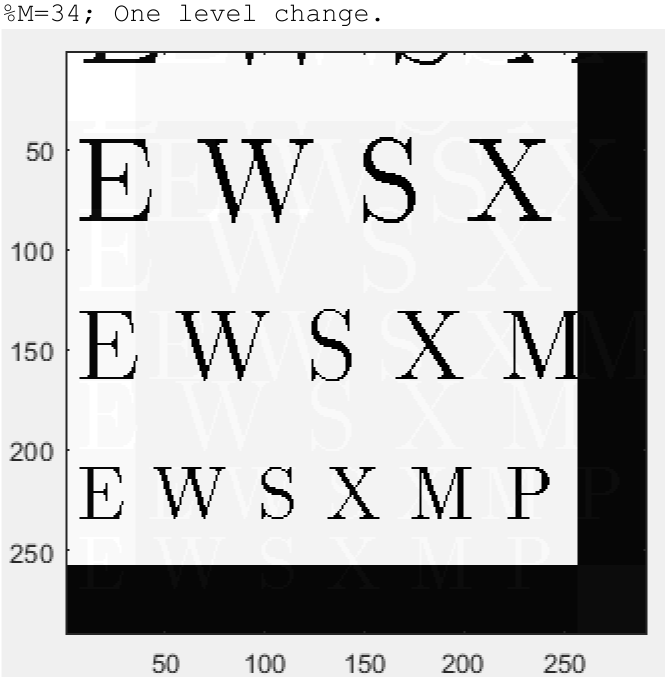



```
%M=33;
err=echart(1:255)-deconv2(1:255);
max(abs(err))
```

```
ans =

   7.0923

%The worst-case error is about 7.09.
```




```
%Question: Can your eyes perceive a gray scale change of one level, one
%part in 256?
%M=34; One level change.
```




```
%My eyes cannot perceive a gray scale change of one level as the plots
%looks identical.
diary off
```