THE UNIVERSITY OF TEXAS AT ARLINGTON, ELECTRICAL ENGINEERING DEPARTMENT

TERM PROJECT

Approximate Information Vectorized Recovery, UP-SAMPLING, DOWN-SAMPLING and FILTERING IMAGES

EE-5350

Professor: Soontorn Oraintara

Minhtuan D. Vo

Fall 2000

1 **Write Matlab function scaling_filter.m to design the impulse response of the scaling linear-phase filter H(z) that should be used. The function should have the following input-output format:**

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % %
% Impulse Response of the linear-phase filter H(z).   %
% Input and Output have the following format:         %
%     L: Upsampling factor                            %
%     M: Downsampling factor                          %
%     N: Filter order (filter length = N+1)           %
%     h: Impulse response of the filter H(z)          %
%     x[n]-->[^L]--->[H(z)]--->[vM]--->y[n]           %
% % % % % % % % % % % % % % % % % % % % % % % % % % % %

function [h] = scaling_filter(L,M,N)

% If M>L, then pi/M is the dominant cutoff frequency
% and if M<L, then pi/N is the dominant cutoff frequency.

if M>L
   f_cutoff=1/M;
else
   f_cutoff=1/L;
end
x = remez(N, [0 0.95*f_cutoff f_cutoff 1], [1 1 0 0]);
[h]=x;
```

2. **Write a Matlab function scaler.m to scale the size of an image (2-D signal/matrix) by a given ratio. The function should have the following input-output format:**

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% Function scaler.m can be used to scale the size of an     %
% image (2-D signal/matrix) by a given ratio.  The          %
% function should have the following input/output format    %
%     function [image_out] = scaler(image_in,L,M,h)
%           L: Upsampling factor                            %
%           M: Downsampling factor                          %
%           h: impulse response of the filter H(z)          %
%           image_in: Original input image                  %
%           image_out: Scaled output image                  %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %


function [image_out] = scaler(image_in,L,M,h)

% Up-scale an image

[r,c]=size(image_in);            % get the size of an image
x1(L*r,L*c)=[0];                 % creat an overall image
x1(1:L:L*r,1:L:L*c)=image_in;    % add zeros in the middle
up_scale=x1;                     % final image after upscale

% filtering the image with filter H(z)

x=conv2(x1,h);                   % Convolution with the row
y=conv2(x',h);                   % Convolution with column
y1=y';                           % The result
```

```
% Down scale an image

[ry1,cy1]=size(y1);
x2=y1(1:M:ry1,1:M:cy1);
[image_out]=x2;
```

3. **The file tv.mat which can be downloaded from the class website contain a 300x400 test image. Downsampling the tv image by a factor of 3 in both horizontal and vertical direction. Do you see any aliasing/distortion? Now using your function scaling-filter.m to design a filter h[n] which will be used to scale the image by a factor of 1/3, i.e. L=1 and M=3. Plot the impulse, magnitude and phase response of the filter. Use your function scaler.m to scale the tv image. Display and compare the scaled image with the one without filtering.**

- Obtaining TV Image from the Class Website:

```
» load tv.mat
» imagesc(tv);
» colormap gray;
» title('Original TV Image');
```

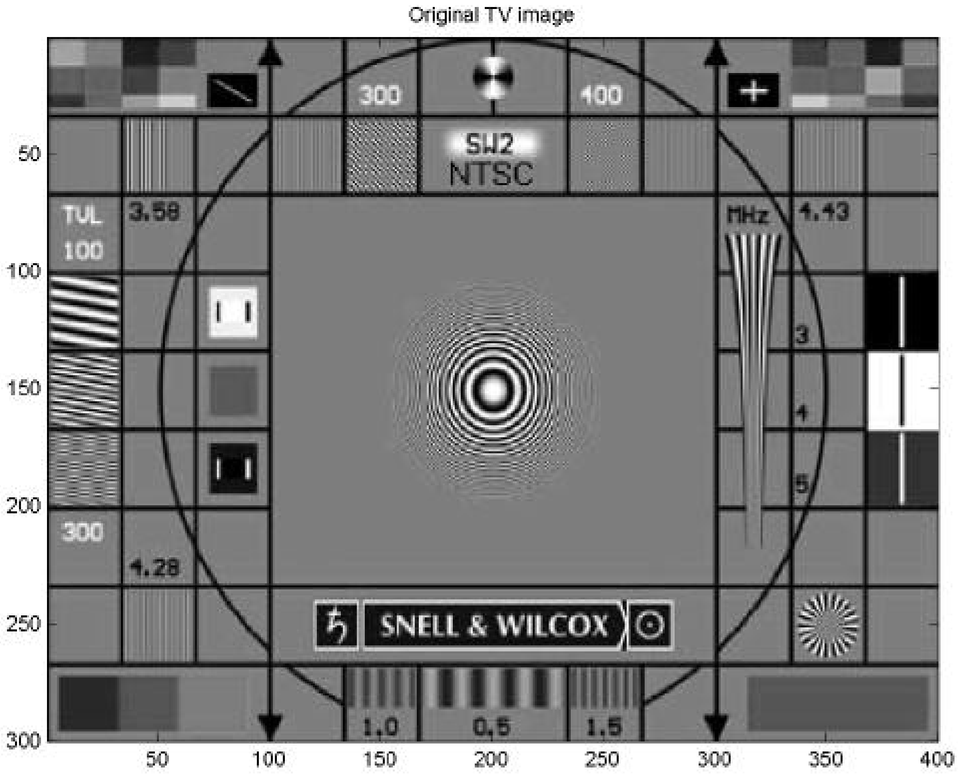


Figure 1: *Original TV Image Obtained from the Class Web Site*

- Downsampling tv image by a factor of 3 (row & column)

There are two ways to do the 3-time downsampling. The first way is to downsampling is to take sample off of the original image depending on the rate of the downsampling, and the second way is to downsampling with filtering

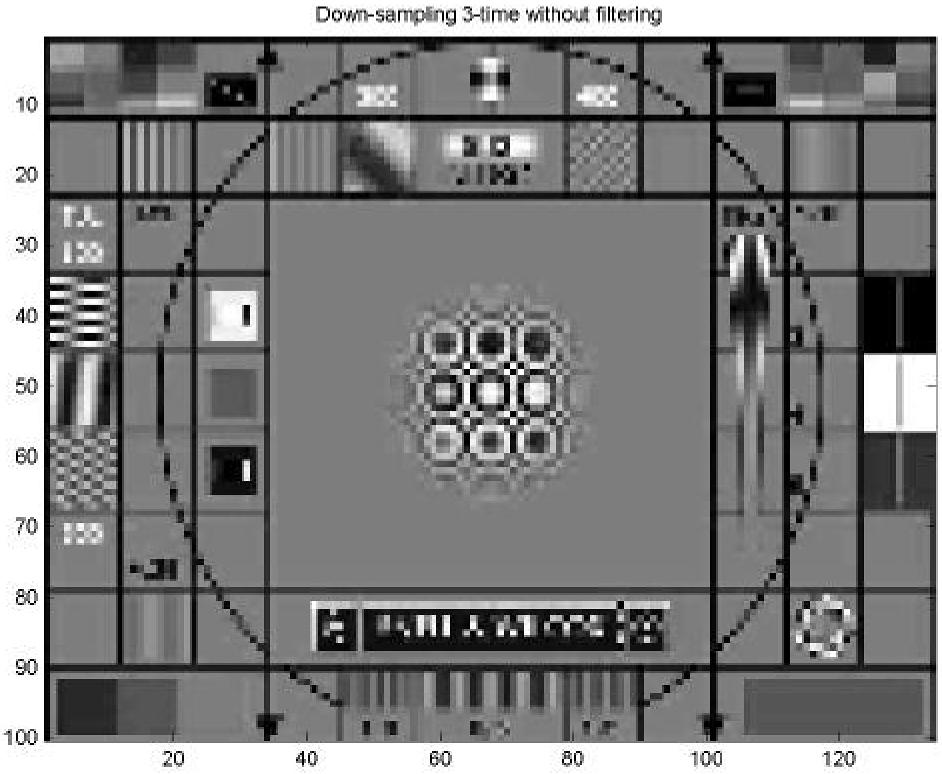


Figure 2: *3-time Downsampling By Taking Off Samples From the Original Image*

- <u>Any Distortion/Aliasing?</u>

  Figure 2 shows the first way to downsampling an image without filtering, and it clearly shows the aliasing existing in this method. Also, the distortion can be seen because of losing data when down sampling.

- <u>Using filter</u>

  The second way can be used to downsampling an image without (or the least) aliasing is to use filtering. Two types of using filters are: using the built-in filter of the decimate function, or applying an appropriate filter for the previous downsampling method. The built-in filter of the decimate function and its 3-time downsampling can be seen in figure 3. In this case the aliasing has been reduced dramatically to compare with figure 2. However, distortion can not be avoided since data has been cut down to 3 times, so the sharpness of the image (figure 3) can not be compared with the original image as in figure 1. Figure 4 is an image of the first method of downsampling, but this time an H3(z) low-pass filter is used to reduce aliasing. The aliasing in figure 4 is much lower than the aliasing in figure 2, but the clarity of the image has been distorted since samples has been reduced. Different type of filters yield different results, and this can be seen in comparison between figure 3 and figure 4. The built-in filter in the decimate function is an eight-order Chebyshev type, so it produces a lesser aliasing as in figure 3. The normal low-pass filter in figure 4 is not sharped enough to avoid as much asilasing, and that produces a poorer image as in figure 4. Both images in figure 3 and 4 are still having distortion since samples have been reduced.

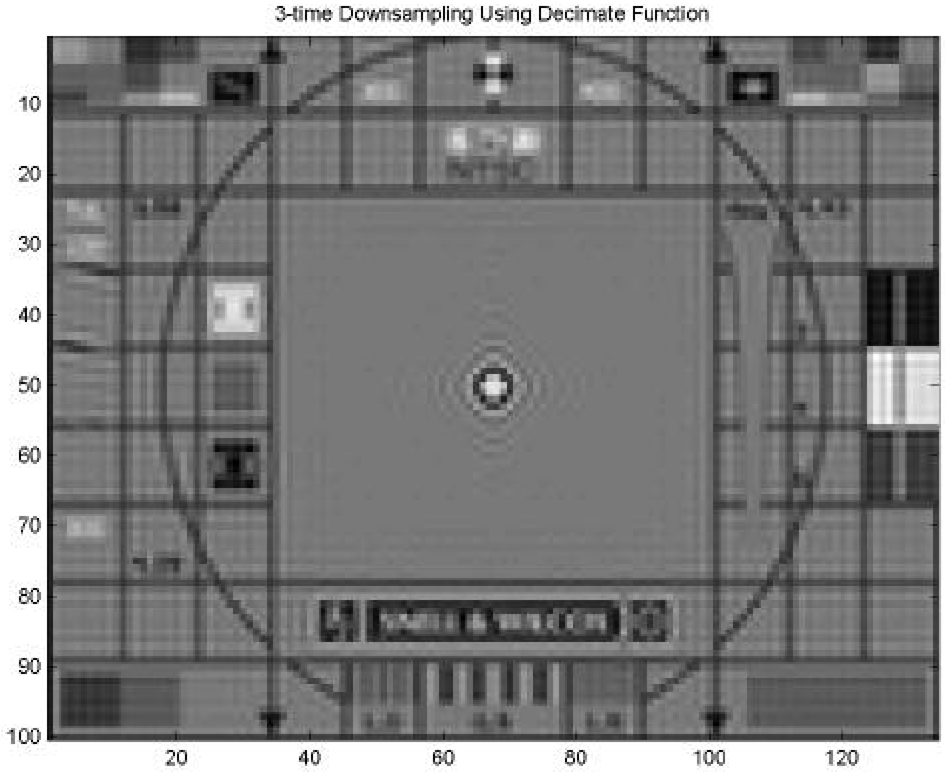


Figure 3: *3-times Downsamping Using Decimate Function and Built-in Filtering*

```
% % % % % % % % % % % % % % % % % % % % % % % % % % %
% This function can be used to downsampling an image%
% using decimate func.                              %
% image_in:  image input having mxn matrix          %
% R        :  the rate to be decimate               %
% image_out: the output image having 1/R of the rate%
% % % % % % % % % % % % % % % % % % % % % % % % % % %

function [image_out] = mtdvdecimate(image_in,R)

[m,n]=size(image_in);
for i=1:m
   x(i,:)=decimate(image_in(i,:),R);
end
x1=x';
[o,p]=size(x1);
for i=1:o
   y(i,:)=decimate(x1(i,:),R);
end
image_out=y';
```

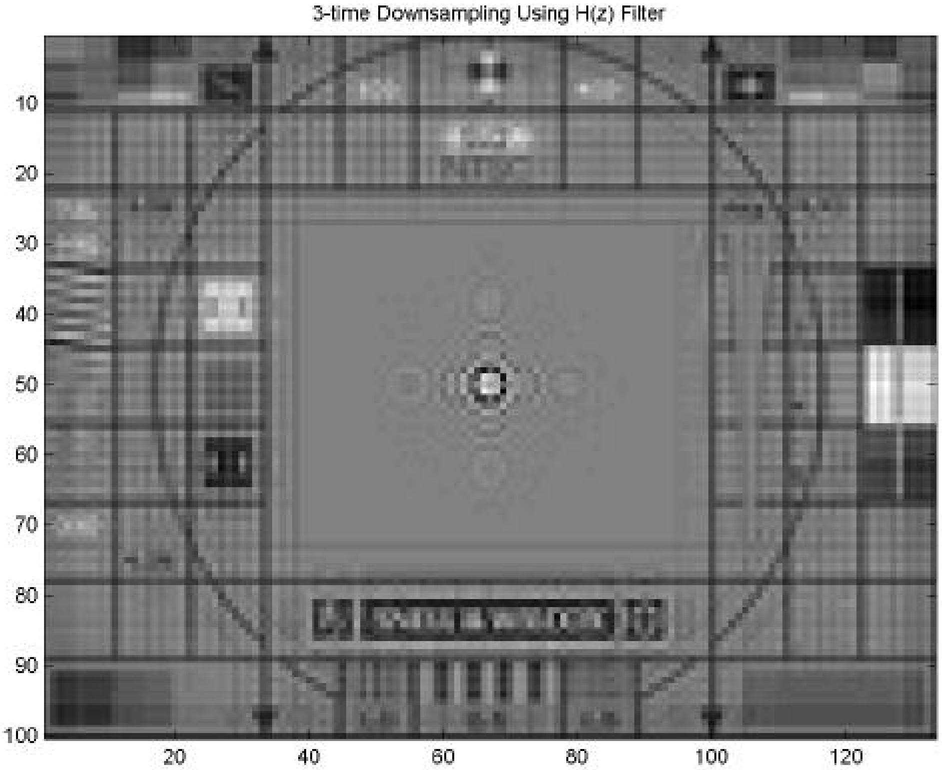


Figure 4: *3-time Downsampling of the First Method and Apply H(z) Filtering*

- Plot Magnitude and Phase of the Filter using H3(z) scaling_filter

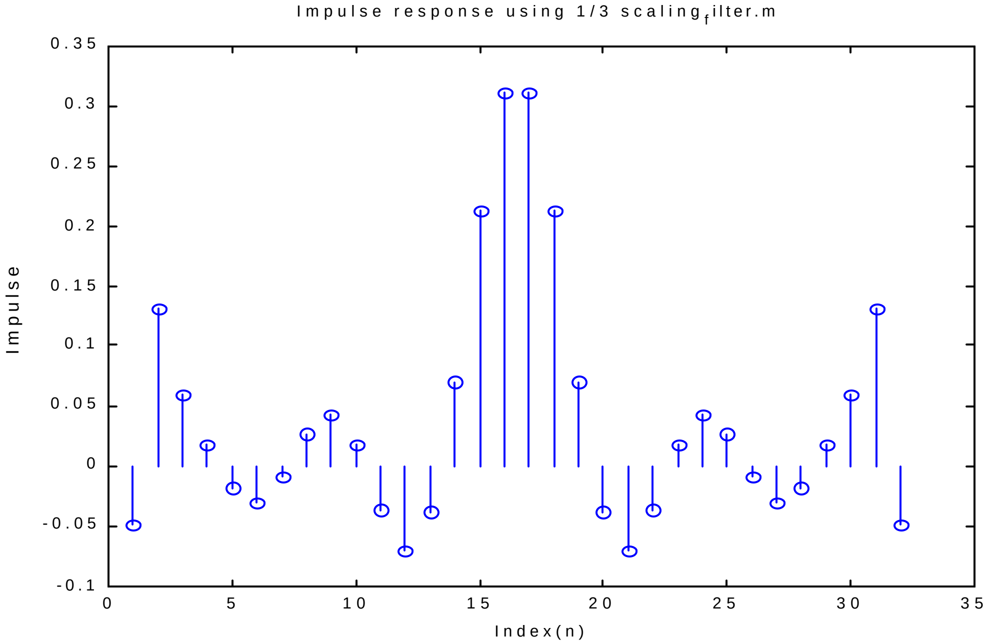
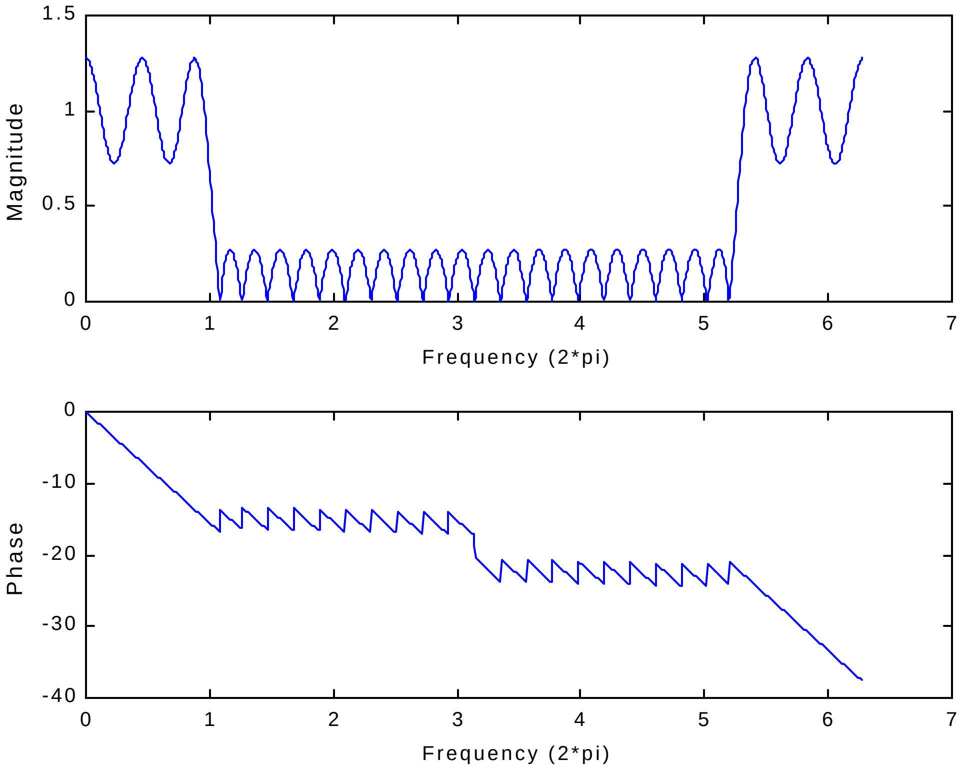


Figure 5: *Impulse Response of the H3(z) Filter Using in Figure 4*



<u>Figure 6:</u> *Magnitude and Phase Response of H3(z) filter Using in figure 4*

4. **Upsample the TV image by a factor of 2(L=2 and M=1) without any filtering. What is the effect of imaging distortion appearing in the upsampled image. Apply a scaling filter obtained by using the scaling_filter.m to the upsampled image. Display and compare the results.**

    - <u>Upsample by a factor of 2 without any filtering:</u>

      Figure 7 shows a 2-Time Upsampling image without applying filter.

    - <u>The Effect of Imaging Distortion:</u>

      Theoretically, upsampling will add zero(s) in the middle of the original samples, and figure 7 shows the distortion of these zeros affect to the original image. Zeros are added in both horizontal and vertical coordinates, so they create a net-like-zeros over the original image and spread the original image out depending on the upsampling scale. This phenomenon is called aliasing. The zeros added effect can be avoided by using the interpolation between zeros and original samples, and figure 8 shows the interpolation with high-order filtering.

    - <u>2-time Downsampling Using scaling-filer H4(z):</u>

      Figure 9 shows an image of 2-time upsampling with applying H4(z) filter.

    - <u>Un-filtering and Filtering images Comparison:</u>

      Increasing sampling rate involves operation analogous to D/C conversion (Oppenheim, p.172), so the additional rate will appear on the unfiltered images as lines in figure 7.

Therefore, the upsampling image needs to be LP filtered with the cutoff frequency pi/L (Oppenheim, p.172). As the case of the D/C converter, it is possible to obtain an interpolation formula that uses the LP impulse response of

$$h[n]=[\sin(\pi.n/L)]/(\pi.n/L)$$

to obtain the upsampling sequence

$$x(n)=\sum_{k=-\infty}^{\infty} x(k)\cdot\frac{\sin\left[\pi\cdot\frac{(n-k\cdot L)}{L}\right]}{\pi\cdot\frac{(n-k\cdot L)}{L}}$$

In practice, an ideal LPF cannot be implemented exactly, therefore, the different between each LPF algorithm would yield different results. Figure 8 uses the built-in LP filter that has high-order filter, so the image looks smoother. Figure 10 uses low-order LPF to filter the aliasing, so the image still shows a very small quantity of aliasing.

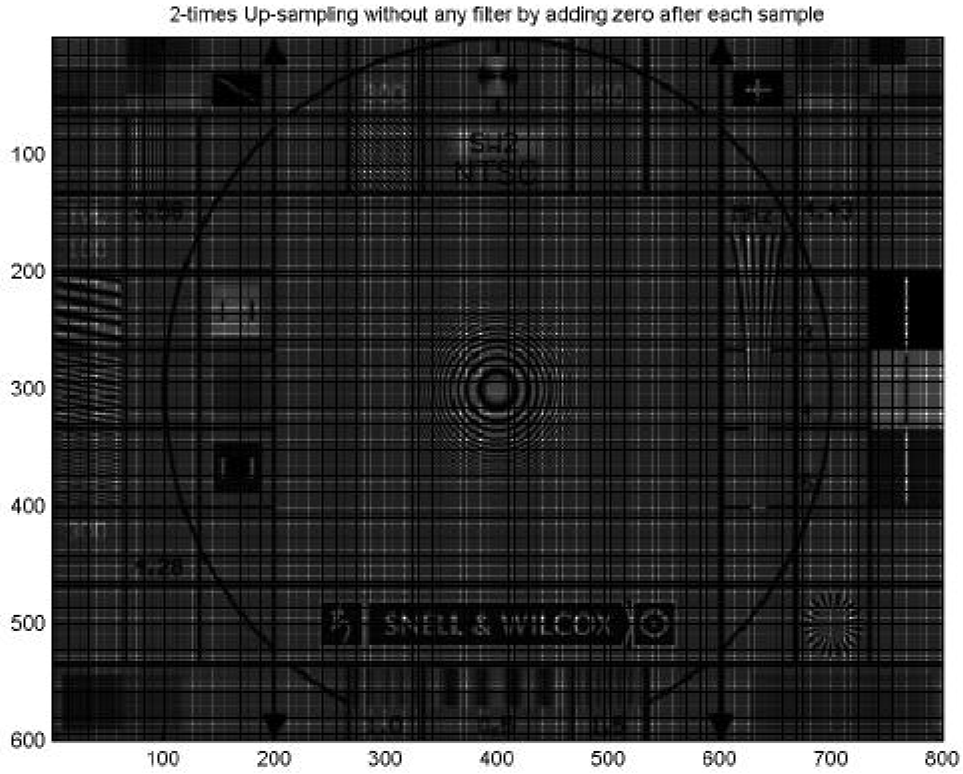


Figure 7: *2-time Upsampling Without Applying Any Filtering*

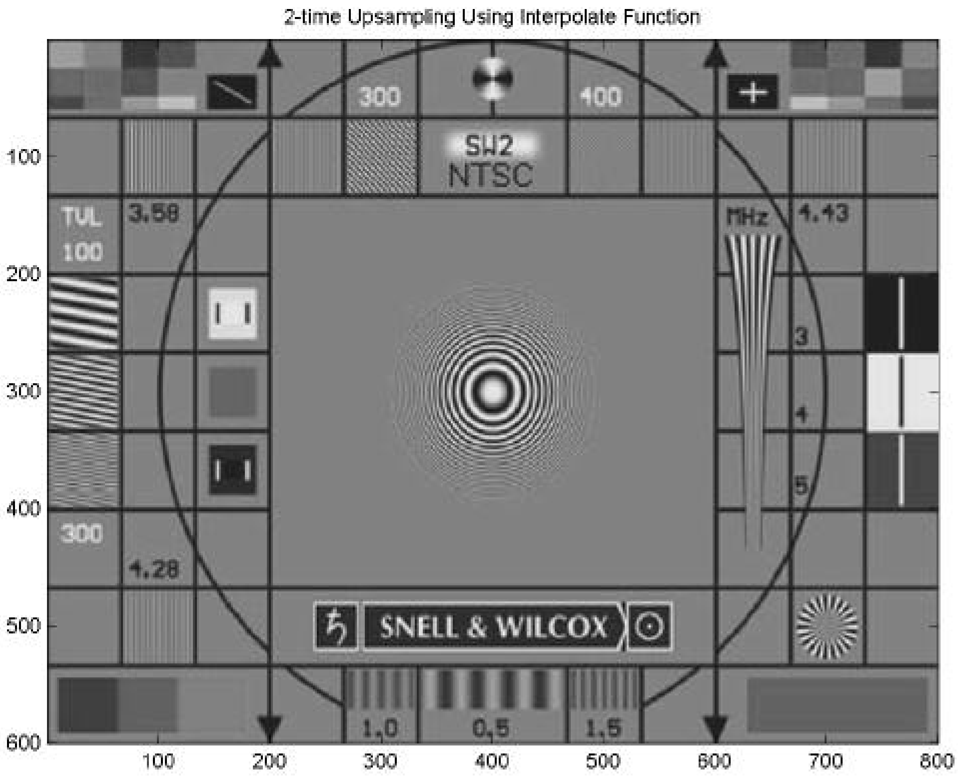

2-time Upsampling Using Interpolate Function

Figure 8: *2-time Upsampling Reducing Reduces Zeros Adding Effect by Interpolation.*

Image is perfectly produced by using higher order filter even scale up 2 times as in figure 8. However, the scope of this project is not explained the filter used in figure 8, so for more information contact matlab technical support engineer at **www.matwork.com**. The code can be displayed below:

```
% This function can be used to upsampling an image
% using interpolation func.
% image_in:  image input having mxn matrix
% R      :  the rate to be interpolated
% image_out: the output image having R times of the
%original rate %
% % % % % % % % % % % % % % % % % % % % % % % % % % % %

function [image_out] = mtdvinterp(image_in,R)

[m,n]=size(image_in);
for i=1:m
   x(i,:)=interp(image_in(i,:),R);
end
x1=x';
[o,p]=size(x1);
for i=1:o
   y(i,:)=interp(x1(i,:),R);
end
image_out=y';
```

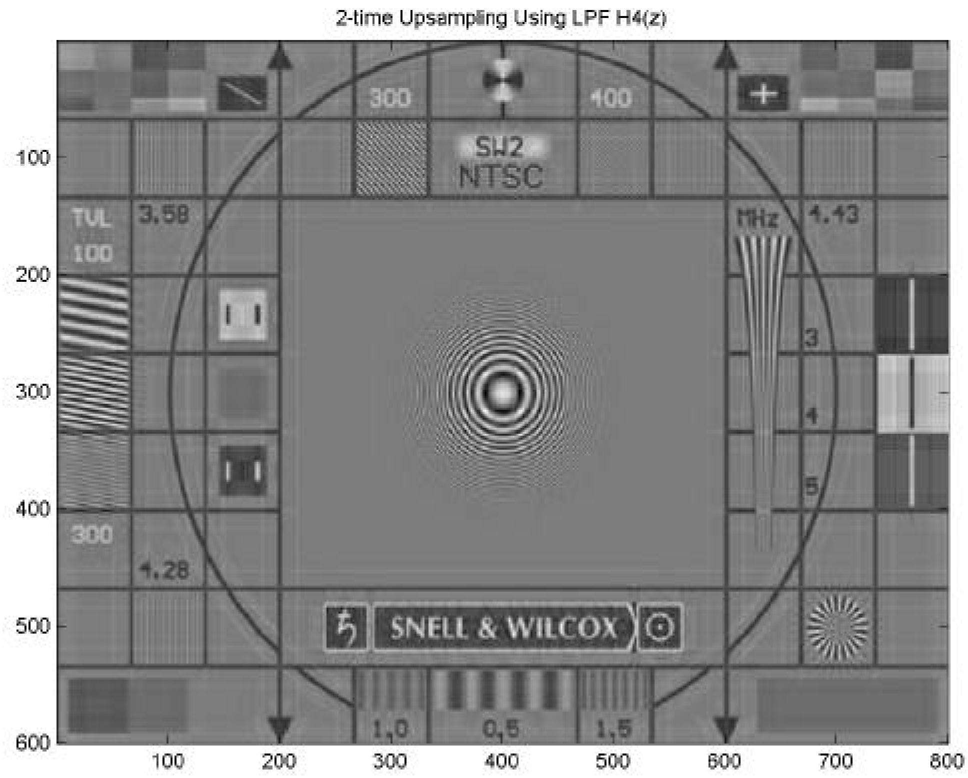


Figure 9: *2-time Upsampling Image with Filter H4(z)*

5. **Design a length-31 FIR filter for scaling the image by a factor of 5/4 (L=5 and M=4). Plot the impulse, magnitude and phase response of the filter. Use the filter to scale the TV image by a factor of 5/4. Is there any major distortion? Explain.**

   - <u>Plot the impulse, magnitude and phase response:</u>

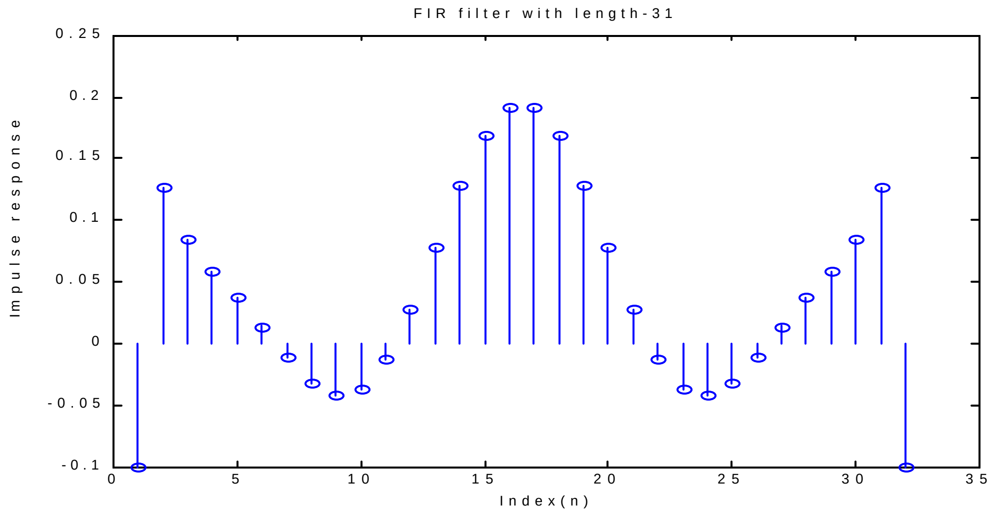
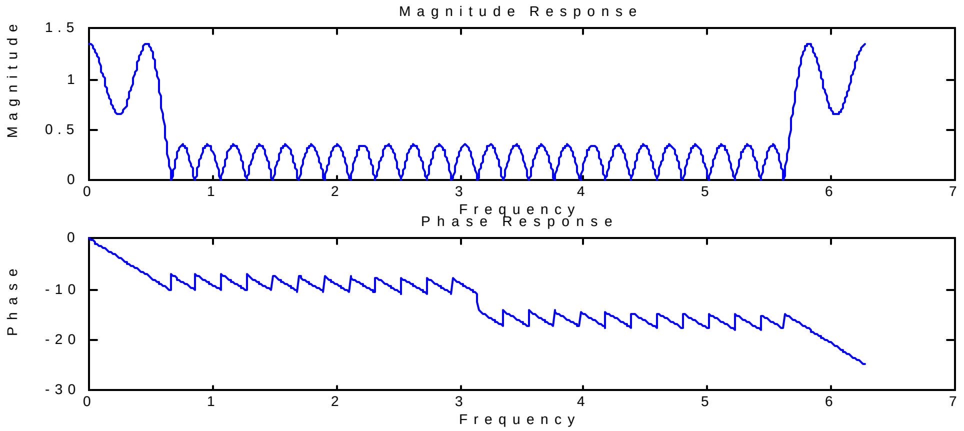
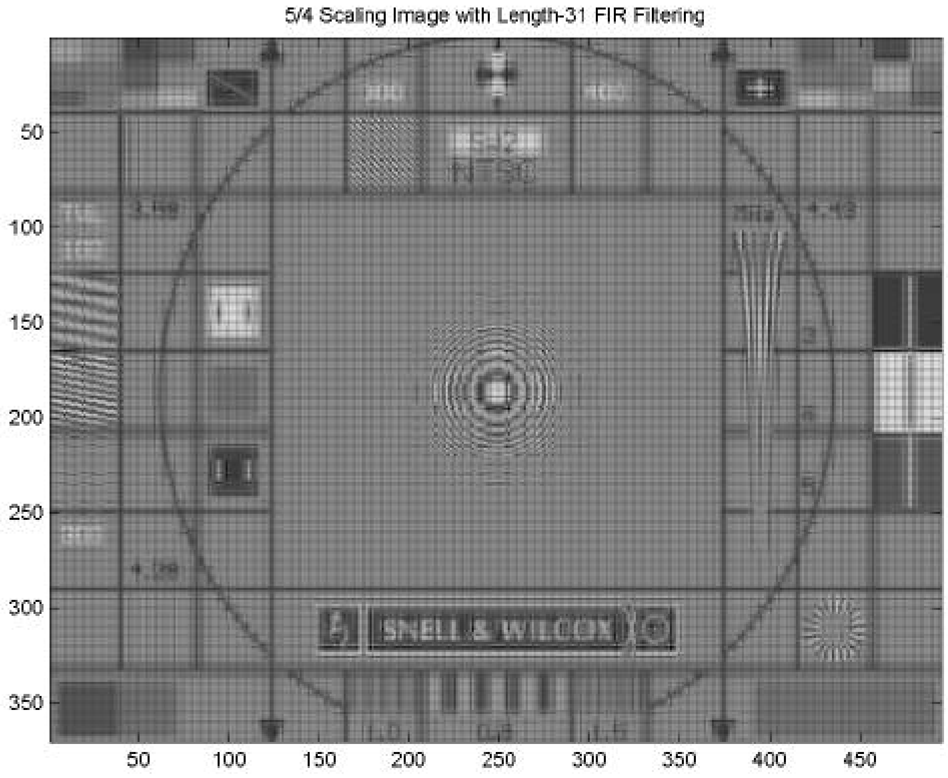


Figure 10: *Impulse response of a Length-31 FIR for the 5/4 Scaling Image*



<u>Figure 11:</u> *Magnitude and Phase response of a Length-31 FIR for the 5/4 Scaling Image*

- <u>Scale 5/4 TV image:</u>



<u>Figure 12:</u> *5/4 Scaling Image Using Length-31 FIR Filter*

- <u>Distortion and Explanation:</u>

  There is a major distortion in the 5/4 scaling image using scaler.m function. The dark-line nest over the tv image can be seen in figure 12. The simplest explanation is the 5/4 upsampling/downsampling would leave a residue of the scaling, and this residue appears as a dark-line nest over the image. This can be reduced by a ideal LPF. However, the length-31 FIR filter is not as perfect, so the image still shows some distortion.

6. **Subtract the tv image by its mean value and call the new image tv0. Use the filter obtained in 5. to scale this image tv0 by a factor of 5/4. At the end, add the mean value of the original tv image back to the scaled output of the tv0 image. Comapre the results obtained from 5. and 6. How would you explain the results?**

```
» load tv.mat
» tv6_mean=tv-mean(mean(tv));
» h6=scaling_filter(5,4,31);
» tv6_mean_wh6f=scaler(tv6_mean,5,4,h6);
» imagesc(tv6_mean_wh6f);
» colormap gray;
» tv6_meanadd_wh6f=tv6_mean_wh6f+mean(mean(tv));
» imagesc(tv6_meanadd_wh6f);
» title('5/4 Being Scaled Image By Subtracting and Adding Mean Value');
```

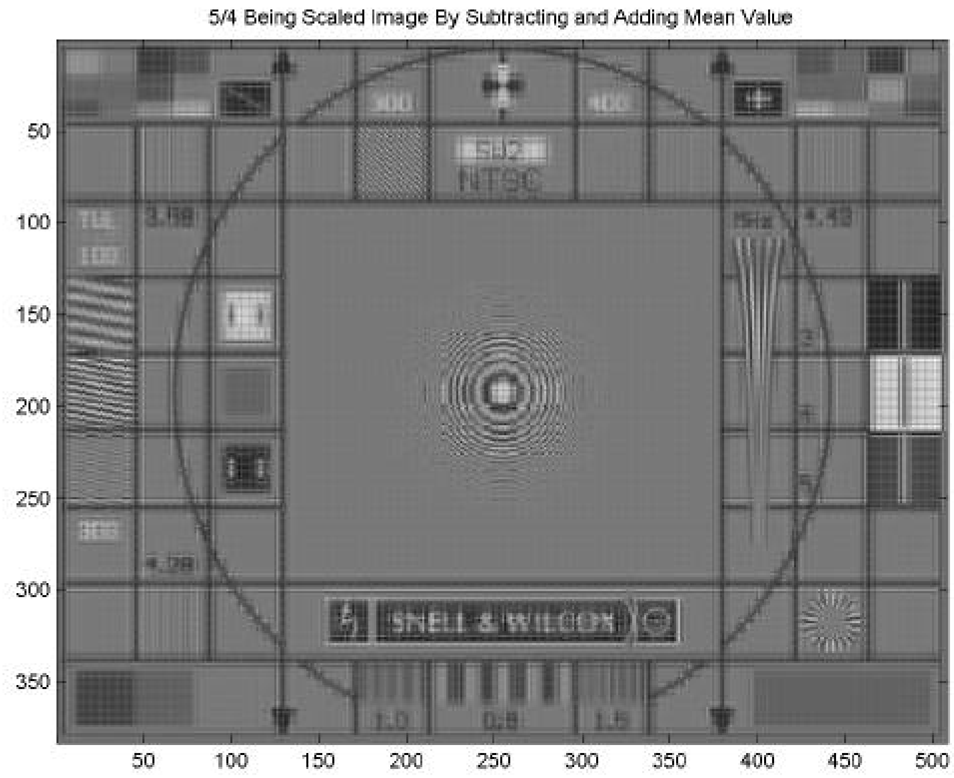


Figure 13: *Subtracting Mean, 5/4 Scaled Image, Adding mean*

<u>Comparison:</u>

- The same filter is used for both images in figure 12 and 13, but figure 13 is smoother than figure 12. However, both images have some major distortions due to the non-ideal LPF to filter out the aliasings.

<u>Explanation:</u>

- The image being scale by 5/4 has very high contrast between pixel, and that results in high frequency. These high frequencies are being filtered out by the scaling_filter(5,4,31), and they can be seen as lines in figure 12.
- Figure 13 shows the result of the same 5/4-scaled image, but before scaling the image has been subtracted by its mean value. Hence, the contrast between pixel has been normalized to yield a smother image as in figure 13.

7. Download the filter h1.mat from the class website. This file contains the impulse response of a length-31 FIR filter. Plot the impulse, magnitude and phase response of the filter. Compare the time and frequency characteristics of this filter to the one obtain from 5. Use this new filter to scale the tv image by a factor of 5/4 and display it. Is there any major distortion in this image? Compare it with the results obtained in 5. and 6.

```
» load h1.mat
» stem(h1);
» title('Impulse Response of the length-31 FIR h1 filter');
» ylabel('Impulse');
» xlabel('Index')
» H1=fft(h1,1240);
» w=pi*[0:2/1024:2-2/1024];
» plot(w,abs(H1));
» title('Magnitude Response');
» ylabel('Magnitude');
» subplot(212);
» plot(w,unwrap(angle(H1)));
» title('Phase Response');
» ylabel('Phase');
» xlabel('Frequency');
```

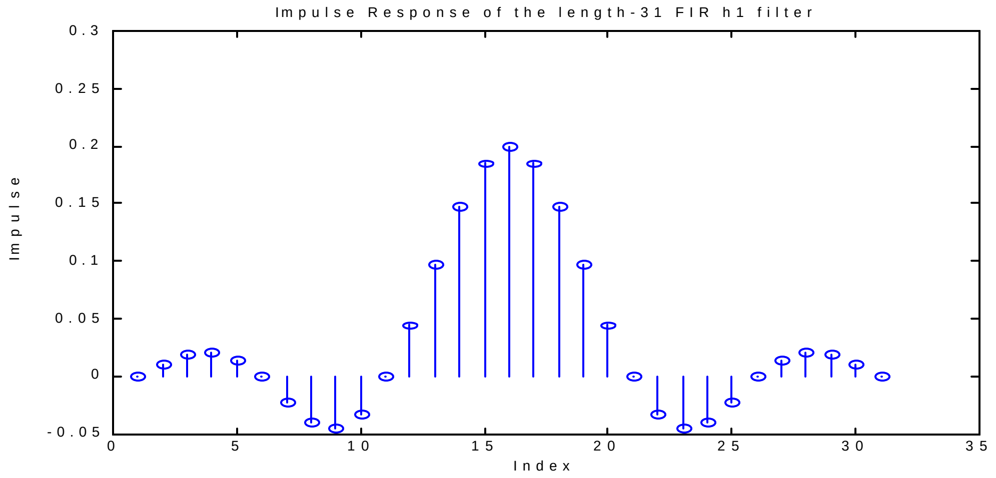


Figure 14: *Impulse Response of the h1.mat filter.*

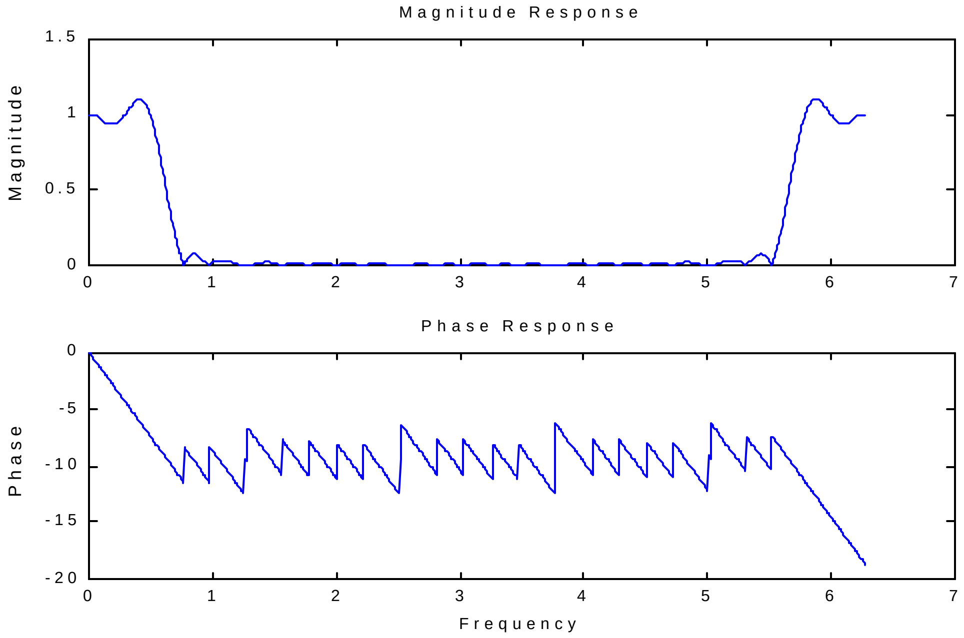


Figure 15: *Magnitude and Frequency Response of The h1.mat Filter.*

Comparison between the filters of question 5 and h1 filter:

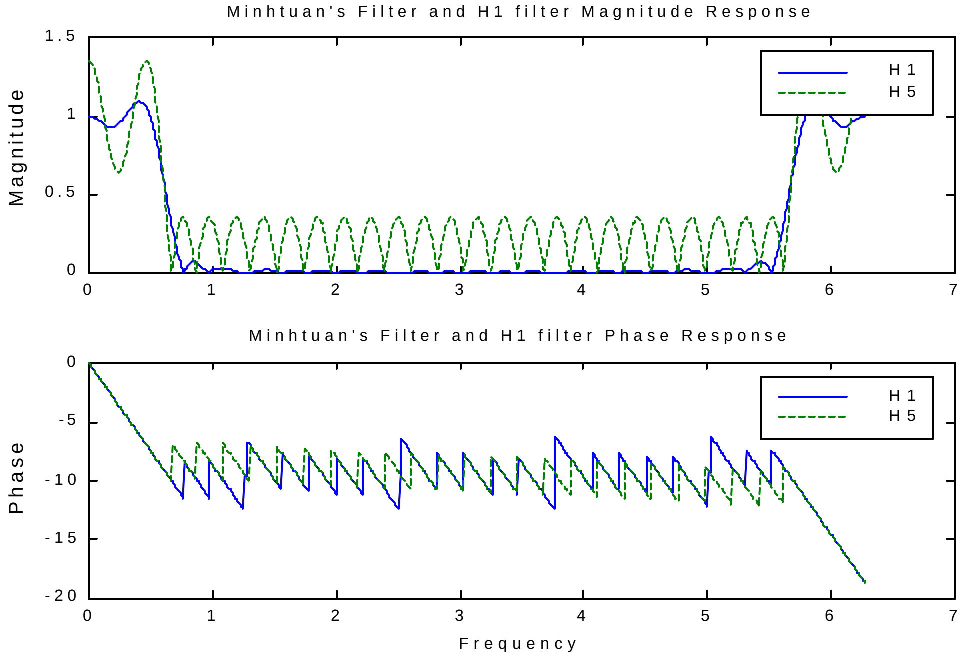


Figure 16: *Magnitude and Phase Comparison Between H1(z) and H5(z)*

- H1(z) and H5(z) have the same cutoff frequency, pi/5 rad.
- H1(z) has lower ripple factor than H5(z) in both pass-band and stop-band.
- H1(z) has a better impulse response than H5(z).
- H1(z) is continuous and stable with time. H5(z) is discontinuous, and unstable with time.

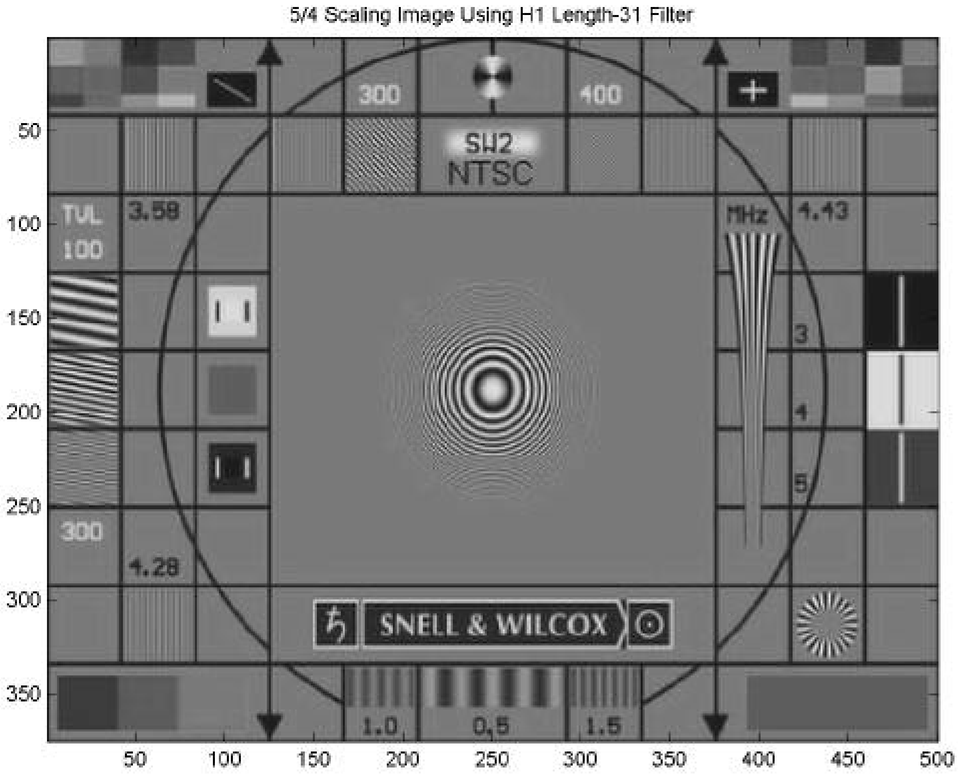


Figure 16: *Tv Image Being Scaled by 5/4 Using H1(z) Filter*

Comparison:

- This image is a perfect 5/4-scaled image to compare with the original tv image. There is no major distortion here.
- Images in figure 12 and 13 used non ideal filter H5(z), and the aliasings are not filteref out completely. More than that the high ripple factors in both pass-band and stop-band yield higher distortion in figure 12 and 13 to compare with a smooth figure 16.

8. **Download the filter h2.mat from the class website. This file contains the impulse response of a length-200 nonlinear-phase FIR filter. Plot the impulse, magnitude and phase responses of the filter. Compare the time and frequency characteristics of this filter to the one obtained from 5. Use this new filter to scale the tv image by a factor of 5/4 and display it. Is there any major distortion in this image? Compare it with the results obtained in 5., 6., and 7.**

```
» H2=fft(h2,1024);
» w=pi*[0:2/1024:2-2/1024];
» subplot(211);
» plot(w,abs(H2));
» title('Magnitude Response');
» ylabel('Magnitude');
» subplot(212);
» plot(w,unwrap(angle(H2)));
```

```
» title('Phase Response');
» ylabel('Phase');
» xlabel('Frequency');
```

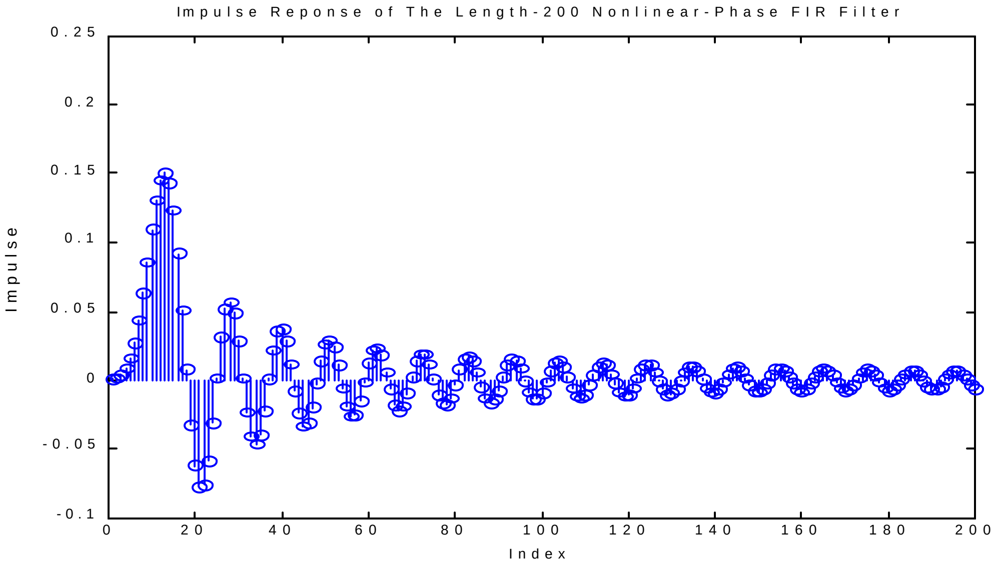


***Figure 17:** Impulse response of the Length-200 Nonlinear Phase FIR Filter*

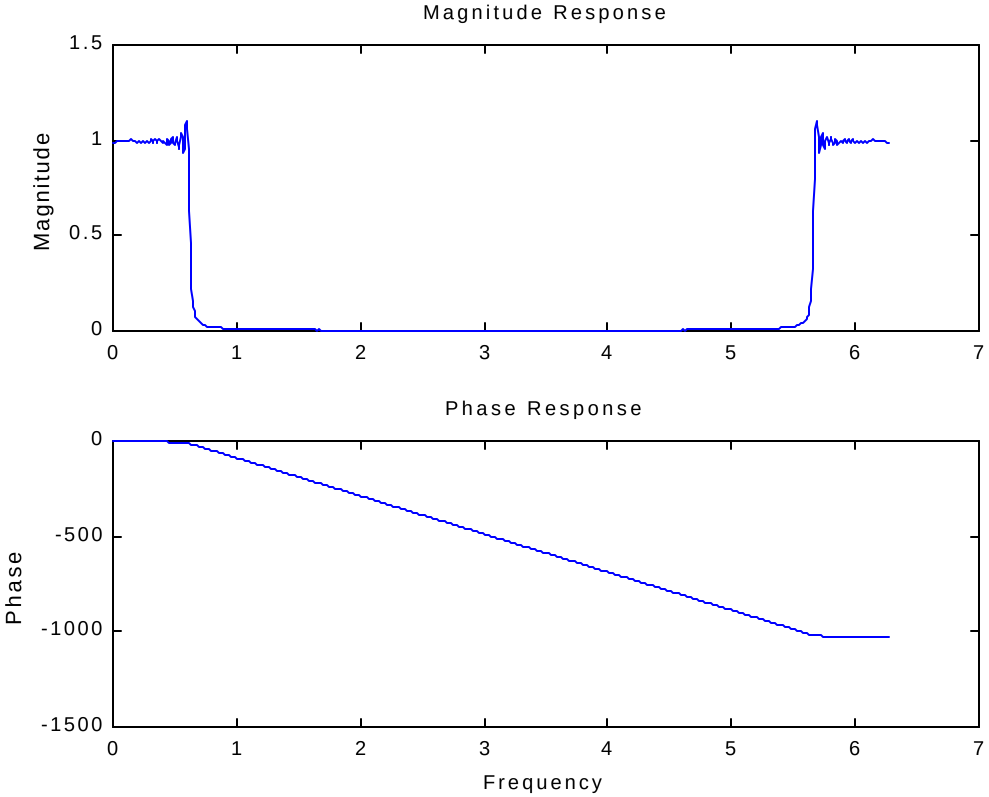


***Figure 18:** Magnitude and Phase Response of the Length-200 Nonlinear Phase Filter.*

```
» H5=fft(h5,1024);
» H2=fft(h2,1024);
» w=pi*[0:2/1024:2-2/1024];
» subplot(211);
» plot(w,abs(H2),'-',w,abs(H5),'--');
» legend('H2','H5');
» title('Magnitude Response of H2 and H5 filter');
» ylabel('Magnitude');
» subplot(212);
» plot(w,unwrap(angle(H2)),'-',w,unwrap(angle(H5)),'--');
» title('Phase Response of H2 and H5 filter');
» ylabel('Phase');
» xlabel('Frequency');
» legend('H2','H5');
```

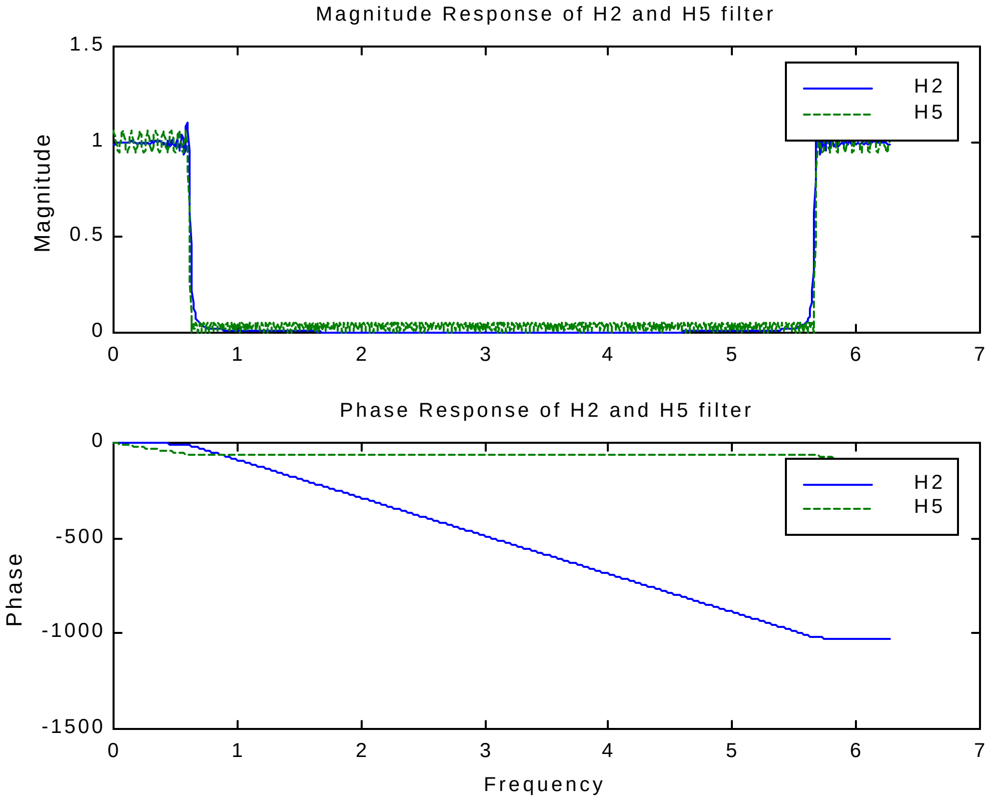


Figure 19: *Magnitude and Phase Comparison Between H2(z) and H5(z).*

Comparison:

- H2(z) and H5(z) have the same cutoff frequency, pi/5 rad.
- H2(z) has an extreme low ripple factor in the stopband to compare with much higher ripple than H5(z) in the stop-band. However, the ripple in the pass band in H2(z) is non-linear that would cause phase shifting in the final image, figure 20.
- H1(z) has a better impulse response than H5(z).
- H1(z) is stable with time. H5(z) is discontinuous, and unstable with time.

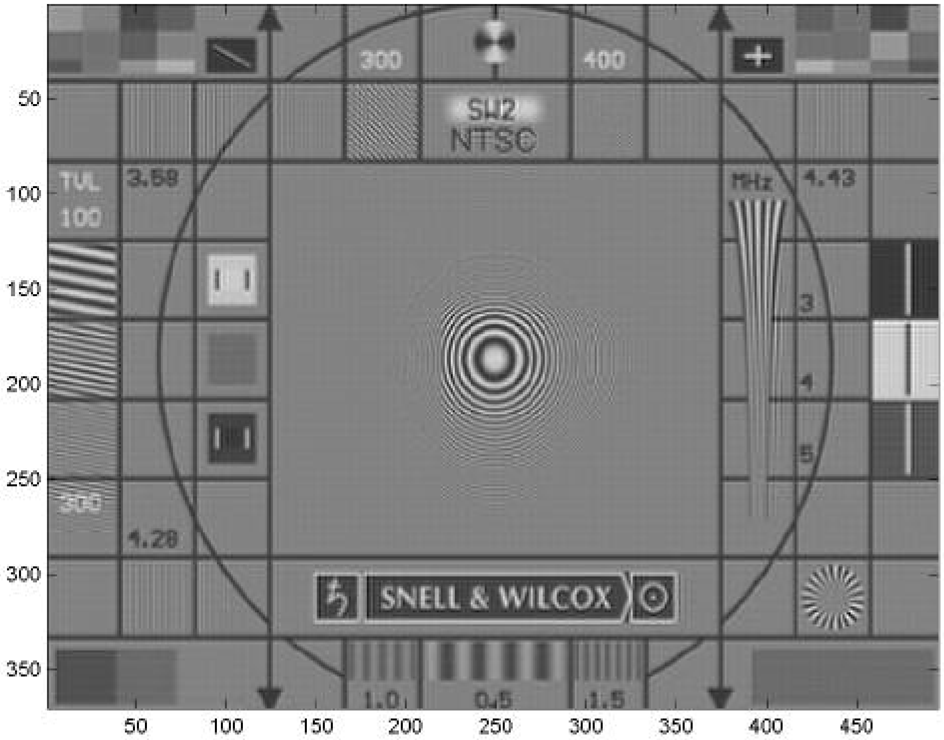


Figure 20: *Tv Image Being Scaled 5/4 Using H2(z) Filter.*

Comparison:

- The aliasings in the image of figure 12 in question 5 were not completely filtered out dues to the non-ideal LPF, so the result image has major distortion.
- Reducing the contrast in the image eliminated the aliasings in the image of figure 13 in question 6. However, the non-ideal filter H5(z) could not completely filtered out the aliasings, so the image still has some distortion.
- Figure 16 shows a near perfect image after scaling 5/4 from the original tv image. The low ripple and linearity of the near ideal filter has produced a better image to compare with the one using H5(z) filter as in figure 12 and 13.
- Figure 20 is a production of the tv scale 5/4 Using non-linear phase filter. In this case, the filter has a perfect ripple factor in the stop-band, but the pass-band is non-linear. Therefore, the image displays some minor shifting of the phase where high probability of aliasings exists. However, no major distortion is generated during the filter process.

Bibliography

- Haddad, Richard A., Parson, Thomas W. *Digital Signal Processing Theory, Applications, and Hardware.* Computer Science Press. New York, NY. 1991.
- Oppenheim, Alan V., Schafer, Ronald W., with Buck, John R. *Discrete-Time Signal Processing.* Prentice Hall. Upper Saddle River, NJ. 1999.
- Oraintara, Soontorn. *EE-5350 Classnote, and hints.* Fall 2000.
- Sayood, Khalid. *Introduction to Data Compression.* Morgan Kaufmann Publishers, Inc. San Francisco, CA. 1996.