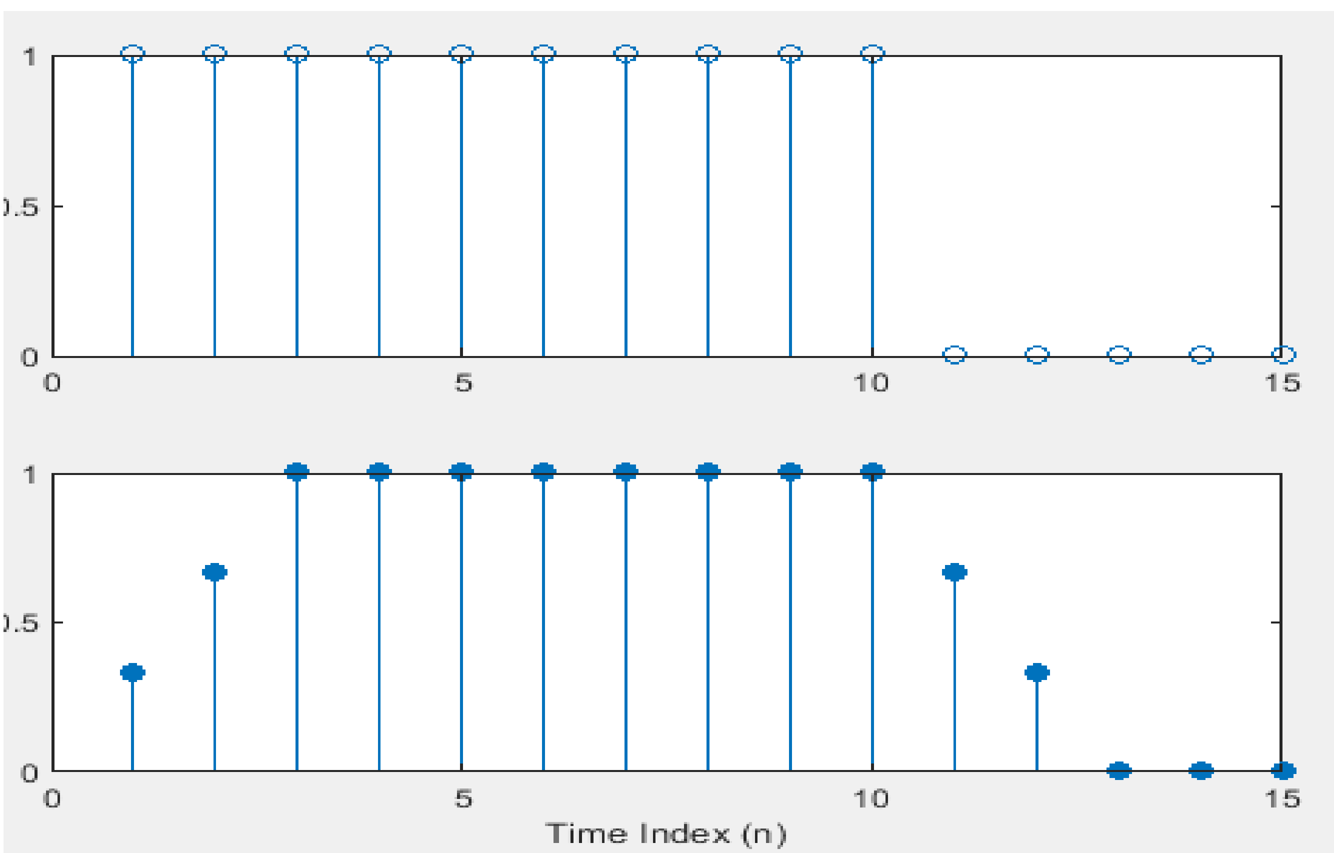
```
diary on
format compact
%Johnny Li
%EEL3135 Fall 2018
%Lab 4 Part 1

%1.1

type averager

%averager Implement the 3-point Averager on an input signal and use a
%stem plot to display the result.
%Script for 1.1.

%Following Code is given.
%Illustrate the filtering action of the 3-point averager.
xx =[ones(1,10),zeros(1,5)]; %<--Input signal
nn = 1:length(xx); %<--Time indices
bk = [1/3 1/3 1/3]; %<--Filter coefficients
yy = firfilt(bk, xx); %<--Compute the output
yy = conv(bk, xx); %<--Equivalent method to compute output

%Make a stem plot of the input signal and output signals in the same figure.
figure(3);
clf
subplot(2,1,1);
stem(nn, xx(nn))
subplot(2,1,2);
stem(nn, yy(nn), 'filled')
xlabel('Time Index (n)')

averager
```



```
%Question: What characteristics of the input signal are most affected by
%the averager?
%The characteristic of the input signal that is most affected by the
%averager is the length of the signal. The input length is 14 whiles, after
%going through the averager, the output signal length is 17 which is an
%increase in length.

%Question: What characteristics of the input signal are least affected by
%the averager?
%The characteristic of the input signal that is least affected by the
%averager is the amplitude of the signal. The resulting output signal's
%amplitude is more influenced by the filter coefficients (bk), rather than
%on the input signal.

%Question: What is the relationship between the lengths of input, output,
%and coefficient vector?
%The relationship between the lengths of input, output, and coefficient
%vector is that the length of the non-zero output is the sum of the length
%of the non-zero input signal plus the length of the non-zero coefficient
%vector. The length of the input vector (xx) is 9 and the length of the
%coefficient vector is 3, therefore the output length is 9+3=12. The stem
%plots only show the first 15 values as set in the parameter, including
%zeros.

%1.2.1

load('labdat.mat')

%Question: What values of r and P and will give an echo with strength 85% of
the
%original, with time delay 0.22s? (fs = 8000)
%The value of P is equal to the time*fs, therefore P=8000*.22= 1760.
%The value of r is equal to the echo with strength, therefore r=0.85.

type echofilter

%echofilter Implement the echo filter and use it on the signal in vector x2
%obtained from labdat.mat.
%Script for 1.2.1.

%The value of r=0.85 (echo strength) and P=1760 (time*fs) from calculation
%of previous question.
%Frequency is given as 8000 Hz.
r=0.85;
%Time delay)
P=1760;
fs=8000;

%Load values
load('labdat.mat')

%Storage Vector
signal=zeros(1,P);
%Edit the starting and ending location for each echo.
signal(1)=1;
```

```
signal(P)=r;

%Echo function: y[n]=x1[n]+rx1[n-P]
xx=firfilt(signal,x2);

%For clipping
xx=xx/(max(abs(xx)));

%Create audio file
audiowrite('echofilter.wav',xx,fs);

echofilter


%1.2.2

type multiechos

%multiechos Filter the x2 signal with this multi-echo filter. Calls
%multiecho function.
%Script for 1.2.2.

%The value of r=0.85 (echo strength) and P=1760 (time*fs) from calculation
%of previous question.
%Frequency is given as 8000 Hz.
r=0.85;
%Time delay)
P=1760;
fs=8000;

%Load values
load('labdat.mat')

%Storage Vector
signal=zeros(1,P);
%Edit the starting and ending location for each echo.
signal(1)=1;
signal(P)=r;

%Echo1
xx=firfilt(signal,x2);

%Four “single echo” systems.
xa=firfilt(signal,xx);
xb=firfilt(signal,xa);
xc=firfilt(signal,xb);
xd=firfilt(signal,xc);

%For clipping
xd=xd/(max(abs(xd)));

%Create autofile
audiowrite('multiecho.wav',xd,fs);
```

%Derive (by hand) the impulse response of a reverb system produced by
%cascading five “single echo” systems.

Johnny Li
EECE 2135
P.3

Lab 4.1

1.2.2) $y[n] = x_1[n] + r x_1[n-P] \quad r = 0.85 \quad P = 1760$

Cascade 5 "single echo" system

$y[n] = h[n] * h[n] * h[n] * h[n] * h[n]$

Doing two at a time

$$h[n] * h[n] = (\delta[n] + r\delta[n-P])(\delta[n] + r\delta[n-P])$$
$$= (\delta[n] + 0.85\delta[n-1760])(\delta[n] + 0.85\delta[n-1760])$$
$$h^2[n] = \delta[n] + 1.7\delta[n-1760] + 0.723\delta[n-3520]$$

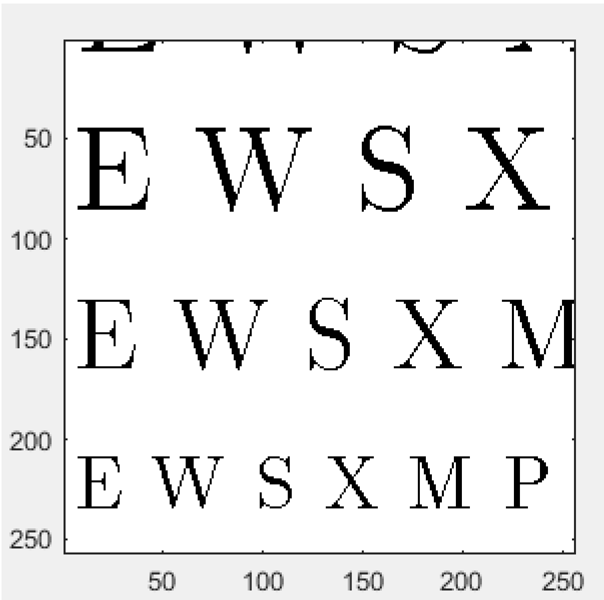
$$h^2[n] * h[n] = (\delta[n] + 1.7\delta[n-1760] + 0.723\delta[n-3520]) * (\delta[n] + 0.85\delta[n-1760])$$
$$= \delta[n] + 1.7\delta[n-1760] + 0.723\delta[n-3520] + 0.85\delta[n-1760] + 1.445\delta[n-3520] + 0.615\delta[n-5280],$$
$$h^3[n] = \delta[n] + 2.55\delta[n-1760] + 2.168\delta[n-3520] + 0.615\delta[n-5280]$$

$$h^3[n] * h[n] = (\delta[n] + 2.55\delta[n-1760] + 2.168\delta[n-3520] + 0.615\delta[n-5280]) * (\delta[n] + 0.85\delta[n-1760])$$
$$= \delta[n] + 2.55\delta[n-1760] + 2.168\delta[n-3520] + 0.615\delta[n-5280] + 0.85\delta[n-1760] + 3.40\delta[n-3520] + 3.018\delta[n-5280] + 1.465\delta[n-7040]$$
$$h^4[n] = \delta[n] + 3.4\delta[n-1760] + 5.568\delta[n-3520] + 3.633\delta[n-5280] + 1.465\delta[n-7040]$$

$$h^4[n] * h[n] = (\delta[n] + 3.4\delta[n-1760] + 5.568\delta[n-3520] + 3.633\delta[n-5280] + 1.465\delta[n-7040]) * (\delta[n] + 0.85\delta[n-1760])$$
$$= \delta[n] + 3.4\delta[n-1760] + 5.568\delta[n-3520] + 3.633\delta[n-5280] + 1.465\delta[n-7040] + 0.85\delta[n-1760] + 4.25\delta[n-3520] + 6.418[n-5280] + 4.483\delta[n-7040] + 2.315\delta[n-8800]$$

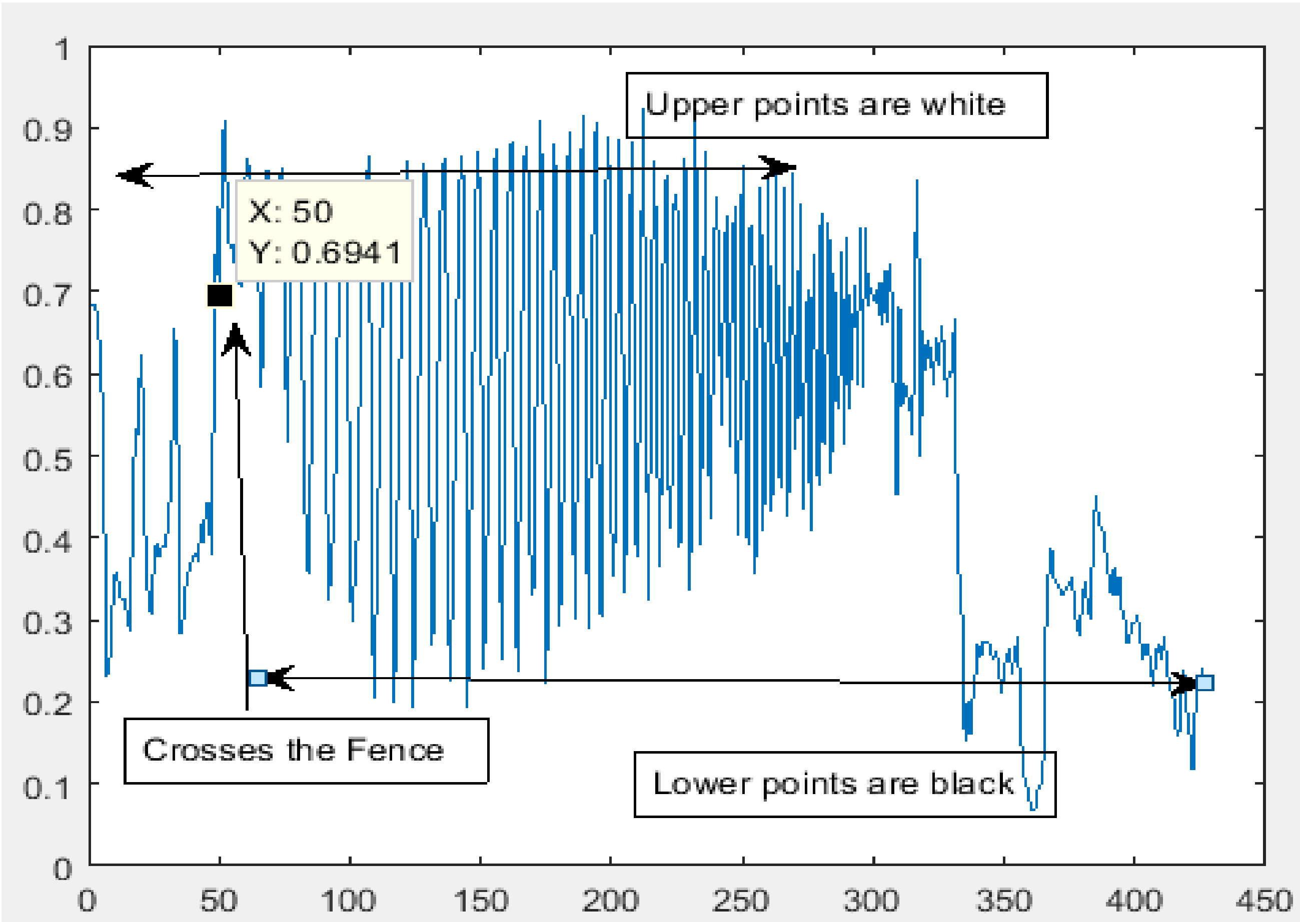
$$\boxed{h^5[n] = \delta[n] + 4.25\delta[n-1760] + 9.818\delta[n-3520] + 10.051\delta[n-5280] + 5.948\delta[n-7040] + 2.315\delta[n-8800]}$$

```
%1.3.1

%Use show_img to display the image in echart.mat
load('echart.mat')
%Auto scale and in grayscale.
show_img(echart,0,1,'gray');
```



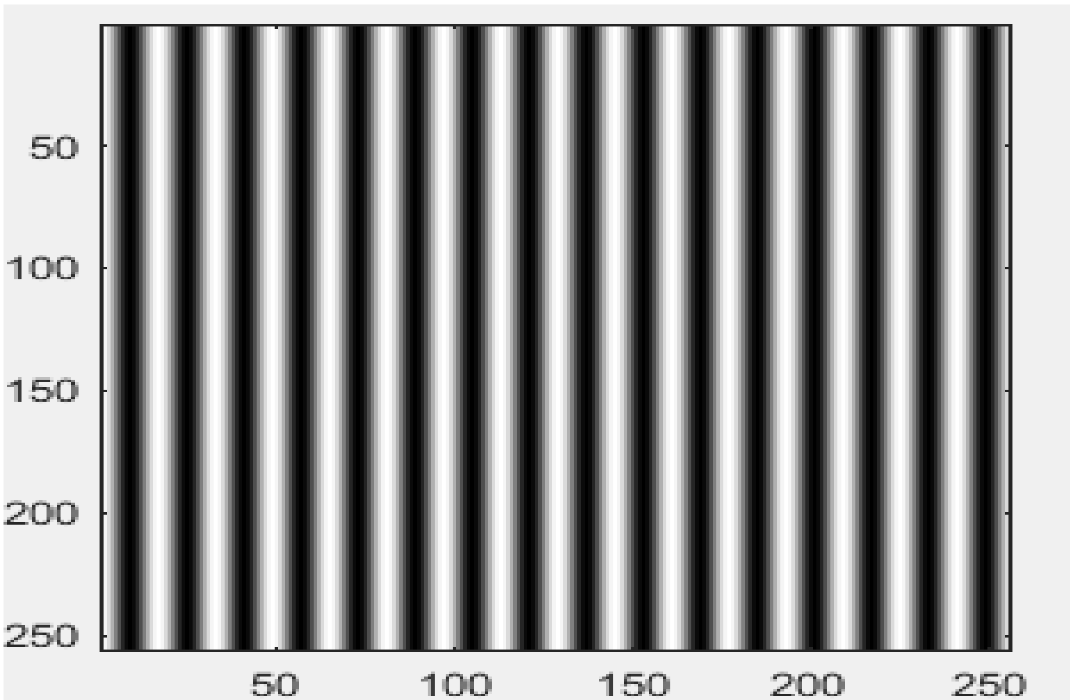

```
%1.3.2

%Load and display the 326 x 426 lighthouse image from the lighthouse.mat in
%grayscale.
load('lighthouse.mat')
show_img(xx,0,1,'gray(256)')
```

```
%Use the colon operator to extract the 225th row of the "lighthouse" image,
%and plot it as a discrete-time one-dimensional signal.
plot(xx(225,:))
```

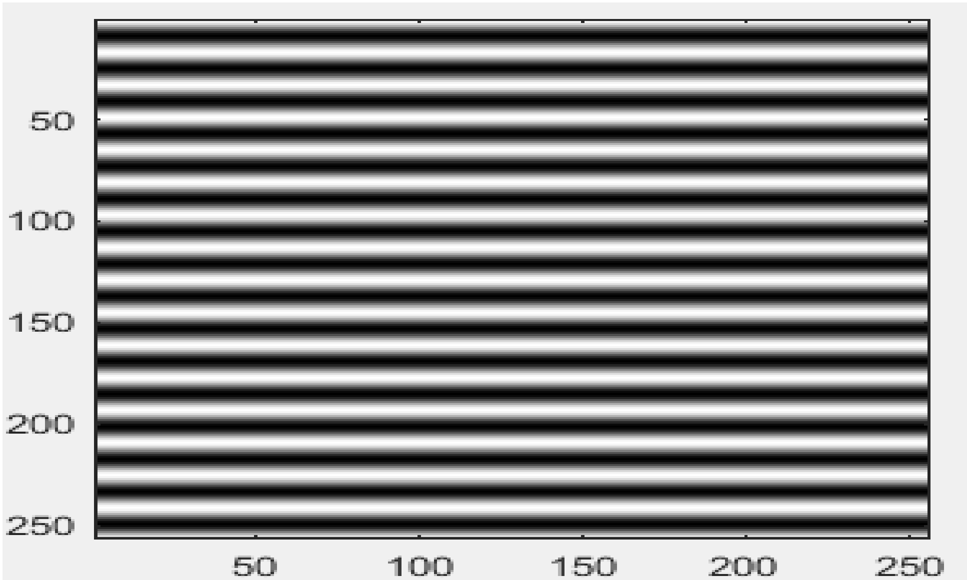



```
%Question: Which values represent white?
%The values that represent white are the upper points of the plots.

%Question: Which values represent black?
%The values that represent black are the lower points of the plots.

%Question: Where does the 225th row cross the fence?
%The 225th row crosses the fence at the leftmost x-value of the plotted
%lighthouse signal, thus its 50 pixels from the left.

%Question: What features of the image correlate with the periodic-like
%portion of the plot?
%The features of the image that correlate with the periodic-like portion of
%the plot are the spaces between stakes on the fence which correlates with
%the oscillation of the plotted lighthouse signal. The respected height of
%the fences, which get smaller the as it gets farer from the left, also
%seems to correlate with the amplitude of the oscillating signal.
```

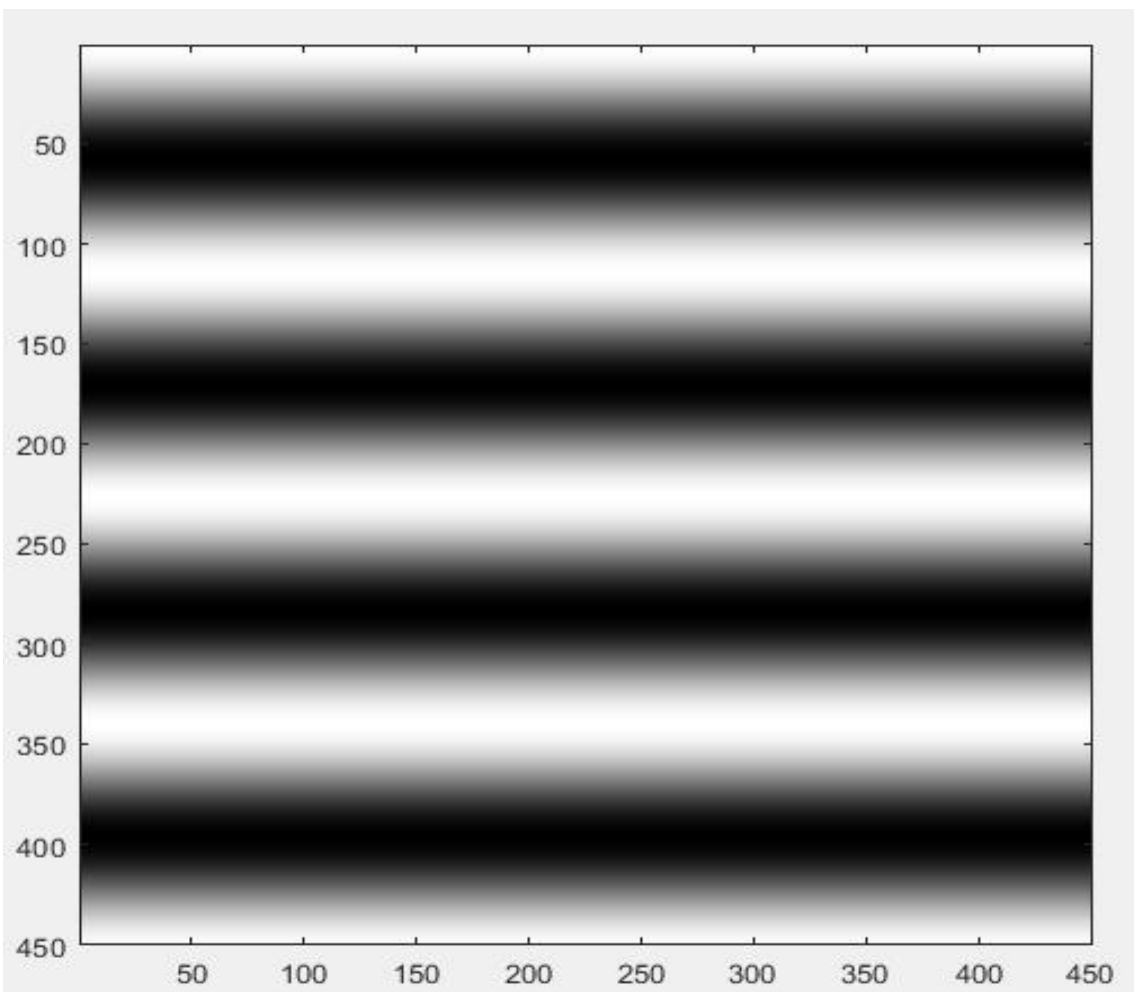
```
%1.3.3

%Display this synthetic image in which all the columns are identical.
xpix = ones(256,1)*cos(2*pi*(0:255)/16);
show_img(xpix,0,1,'gray(256)')
```




```
%Question: How wide are the bands in number of pixels?
%The bands are 16 pixels wide, obtained in the dividing value inside the
%cosine function.

%Question: How is this width related to the formula for xpix?
%This width related to the formula for xpix as the number of bands obtained
%involves dividing the image length (256) by the bands width (16), thus
%obtaining the number of bands. For instance, cos(2pi(256/16))=
%cos(2pi(16))=cos(32pi).

%Question: How would you produce an image with horizontal bands?
%To produce the image with horizontal bands, transpose the image which
%would effectively rotate it by 90 degrees.
xpixtrans=transpose(xpix);
show_img(xpixtrans,0,1,'gray(256)')
```

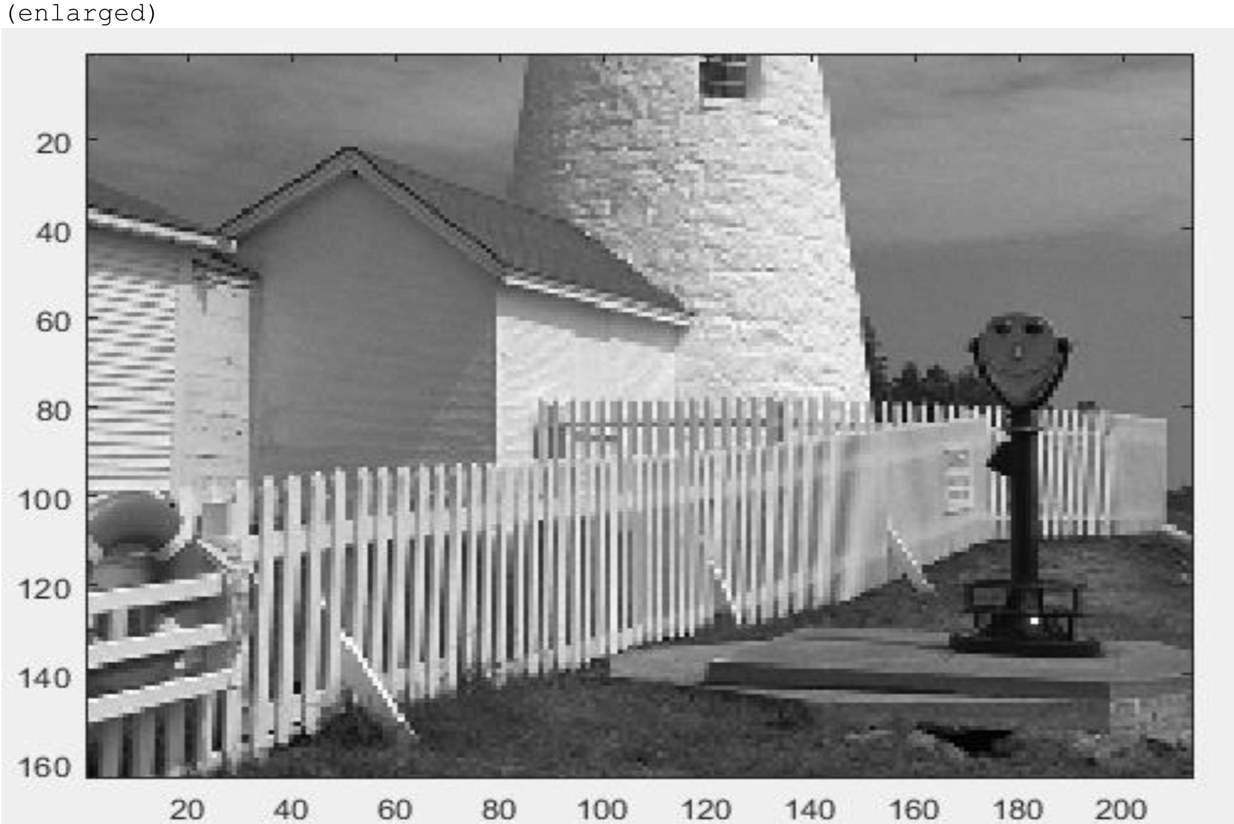



```
%Create (and display/submit) a 450x450 image with 4 horizontal black bands
%separated by white bands.
xpix4 = ones(450,1)*cos(2*pi*(0:449)/113);
xpix4trans=transpose(xpix4);
show_img(xpix4trans,0,1,'gray(256)')
```



```
%1.3.4

load('lighthouse.mat')

%Down-sample the lighthouse image in both dimensions by a factor of 2.
wp = xx(1:2:end,1:2:end);
show_img(wp,0,1,'gray(256)')
```

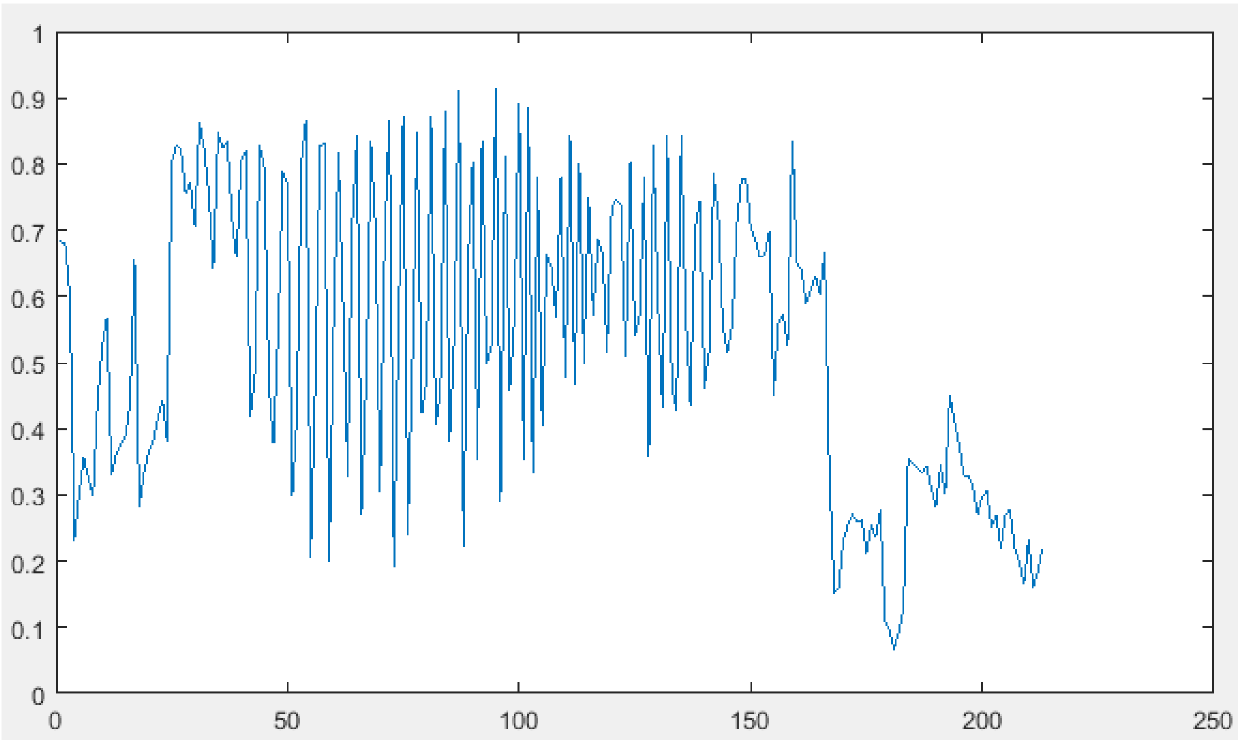
(enlarged)




```
%Question: What is the size of the down-sampled image?
%The size of the down-sampled image is 163x213 pixels.

%Question: Describe how the aliasing appears visually.
%The image appears in half the size but seems similar to the original,
%being proportionally accurate. When the image is enlarged it become blurry
%since there are less pixels thus decreasing the quality of the image.

%Question: Which parts of the image most dramatically show the effects of
the aliasing?
%The parts of the image that dramatically show the effects of the aliasing
%was the fence where the intense details were lost as the image became
%burry and the very fine area began to merge together.

%Question: Why does the aliasing manifest itself in these places?
%Aliasing manifest itself in these places because those areas contain the
%most complex details where the lack of sampling and pixels causes some
%distortions in the image. For instance, the spaces between the fences can
%become quite narrow and numerous but the lack of sampling results in less
```

%defined separations while the lack of pixels results in blurriness.

%Estimate the frequency of the aliased features in cycles per pixel.



%Estimate the frequency of the aliased features in cycles per pixel.
%The estimated frequency is 0.33 cycles per pixel.

%Question: How does your estimation of aliased features fit into the
%Sampling Theorem?
%The estimated frequency of the aliased fit into the Sampling Theorem as
%to have an image without errors the frequency must be within the
%Nyquist frequency but since the estimated frequency is less than that
%aliasing occurred.

diary off