National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Dr. Wajahat Hussain **Dated:** 14-3-2107

Course/Section: BEE6-B **Semester:** 6th Semester

EE-330 Digital Signal Processing

Lab5: Digital Images A/D and D/A

Name	Reg. no.	Report Marks / 10	Lab Quiz-Viva Marks / 5	Total / 15
Saad Iqbal	111394			
Usman Iqbal	111393			
Abdullah Bin Asif	111596			

Objectives:

The objective in this lab is to introduce digital images as a second useful type of signal. We will show how the A-to-D sampling and the D-to-A reconstruction processes are carried out for digital images. In particular, we will show a commonly used method of image zooming (reconstruction) that gives “poor” results.

- ✓ Familiarization with digital images.
- ✓ Working with images in Matlab.
- ✓ Sampling of images
- ✓ Familiarization with reconstruction of images

Introduction:

In this lab, we will learn and explore the working of down-sampling and reconstruction of images in MATLAB. We will also learn the concepts of pixels and spatial frequency in the grey-scale pattern of any image.

LAB TASK 1:

Part (a)

Generate a simple test image in which all of the columns are identical by using the following outer product:

```
xpix = ones(256,1)*cos(2*pi*(0:255)/16);
```

Display the image and explain the gray-scale pattern that you see. How wide are the bands in number of pixels? How can you predict that width from the formula for xpix?

Answer:

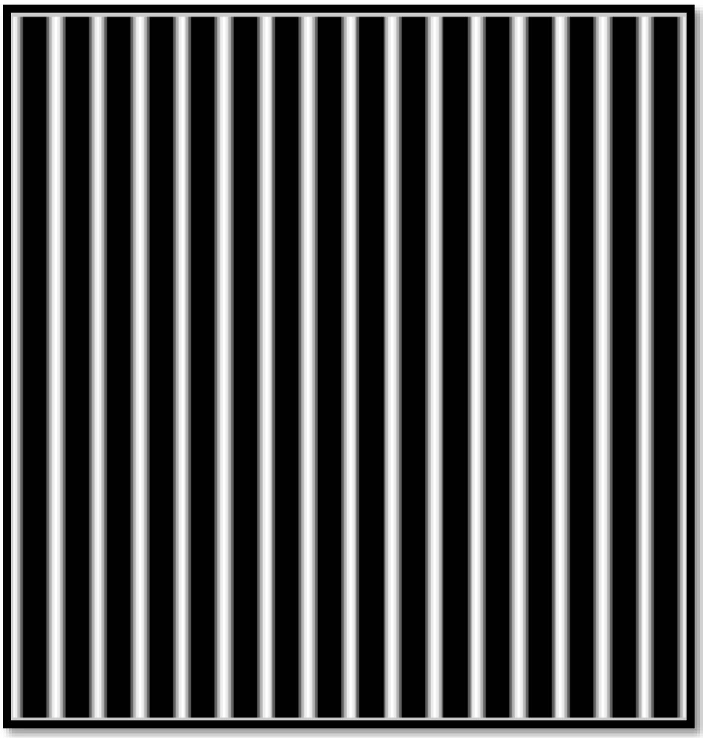

Figure No.1

It changes its grey-scale pattern from white to black with cosine function as the function also predicts so. There are 16 white-black bands and each consists of 16 pixels. It can also be told by looking at the function $\frac{256}{16} = 16\, pixels.$

Part (b)

In the previous part, which data value in xpix is represented by white? Which one by black?

Answer:

The values which are greater than 0.5 are shown by white and as it gets decreased, color become grey and anything less than and equal to zero are shown by black color.

Part (c)

Explain how you would produce an image with bands that are horizontal. Give the formula that would create a 400*400 image with 5 horizontal black bands separated by white bands. Write the MATLAB code to make this image and display it.

MATLAB Code:

```
myimg = ones(400,1)*cos(2*pi*(0:399)/80);
myimg_final = myimg.';

imshow(myimg_final)
```

MATLAB graph:

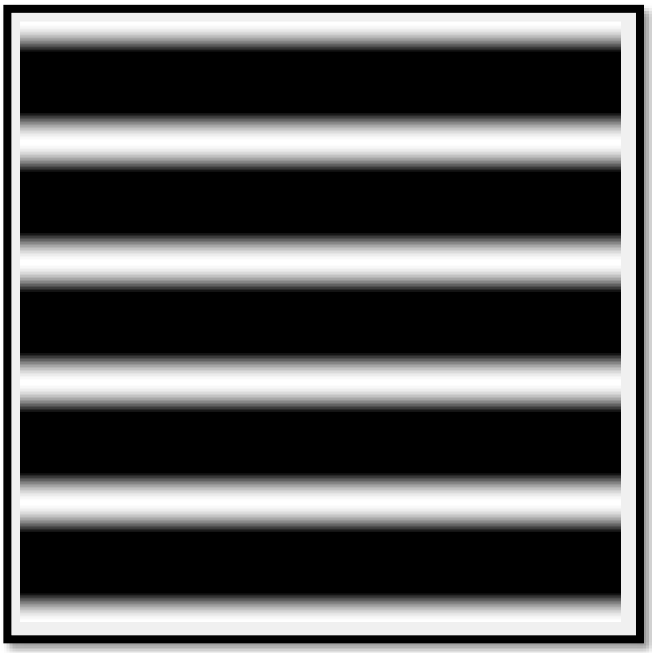

Figure No.2

LAB TASK 2:

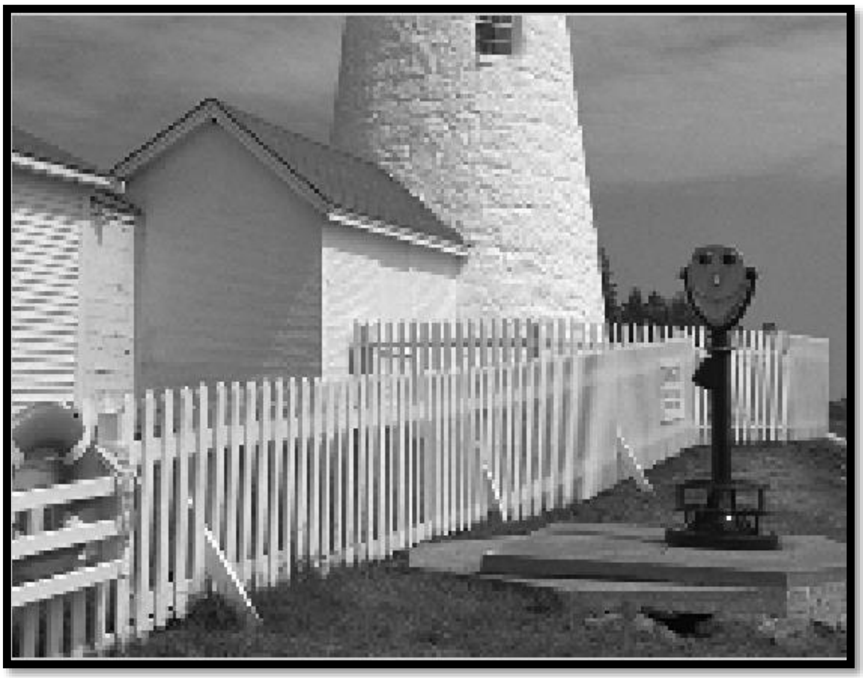

Figure No.3

Down-sampled by the factor of two

Part (a)

Describe how the aliasing appears visually. Compare the original to the down sampled image. Which parts of the image show the aliasing effects most dramatically?

Answer:

Aliasing effect is pretty obvious in this picture when compared with the original picture. Fence in the picture shows the aliasing effect more dramatically.

Part (b)

This part is challenging: explain why the aliasing happens in the lighthouse image by using a “frequency domain” explanation. (Note that the fence provides a sort of “spatial chirp” where the spatial frequency increases from left to right.)

Answer:

The part where the fence is present in this picture has its frequency changing very abruptly going from left to right which is the main reason for more aliasing in this factor.

LAB TASK 3

Reconstruction of Images

When an image has been sampled, we can fill in the missing samples by doing interpolation. For images, this would be analogous to the sine-wave interpolation which is part of the reconstruction process in a D-to-A converter. We could use a "square pulse" or a "triangular pulse" or other pulse shapes for the reconstruction.

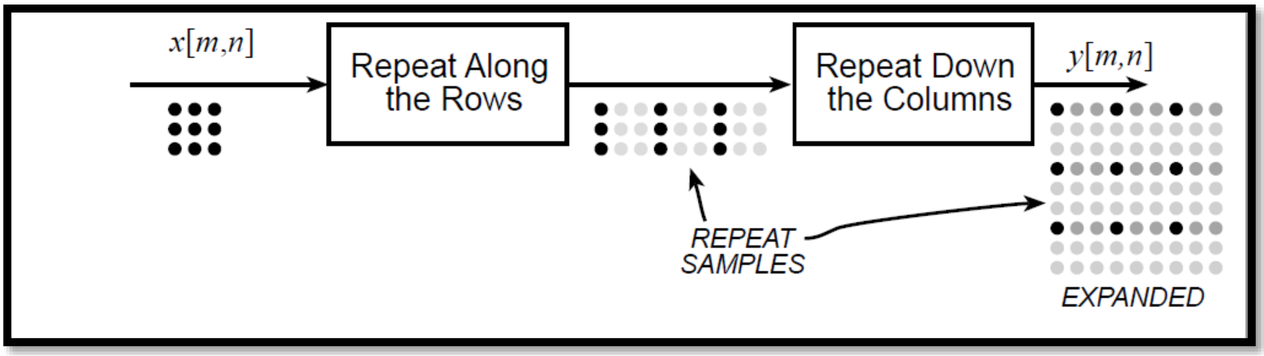

Figure No.4

Figure 4: 2-D Interpolation broken down into row and column operations: the gray dots indicate repeated data values created by a zero-order hold; or, in the case of linear interpolation, they are the interpolated values.

For these reconstruction experiments, use the lighthouse image, down-sampled by a factor of 3 (similar to what you did in Section 2.3). You will have to generate this by loading in the image from lighthouse.mat to get the image which is in the array called xx. A down-sampled lighthouse image should be created and stored in the variable xx3. The objective will be to reconstruct an approximation to the original lighthouse image, which is 256x256, from the smaller down-sampled image.

Part (a):

The simplest interpolation would be reconstruction with a square pulse which produces a "zero-order hold." Here is a method that works for a one-dimensional signal (i.e., one row or one column of the image), assuming that we start with a row vector xr1, and the result is the row vector xr1hold.

```
xr1 = (-2).^(0:6);
L = length(xr1);
nn = ceil((0.999:1:4*L)/4); %<--Round up to the integer part
xr1hold = xr1(nn);
```

Plot the vector xr1hold to verify that it is a zero-order hold version derived from xr1. Explain what values are contained in the indexing vector nn. If xr1holdis treated as an interpolated version of xr1, then what is the *interpolation factor*? Your lab report should include an explanation for this part, but plots are optional—use them if they simplify the explanation.

Solution:

MATLAB Code:

```
xr1 = (-2).^(0:6);
L = length(xr1);
nn = ceil((0.999:1:4*L)/4);
xr1hold = xr1(nn);
plot(xr1hold)
figure()
plot(xr1)
```

Interpolation factor=4

We have observe from our code in matrix representation that each value was repeated four times. Therefore, the interpolation factor is four.

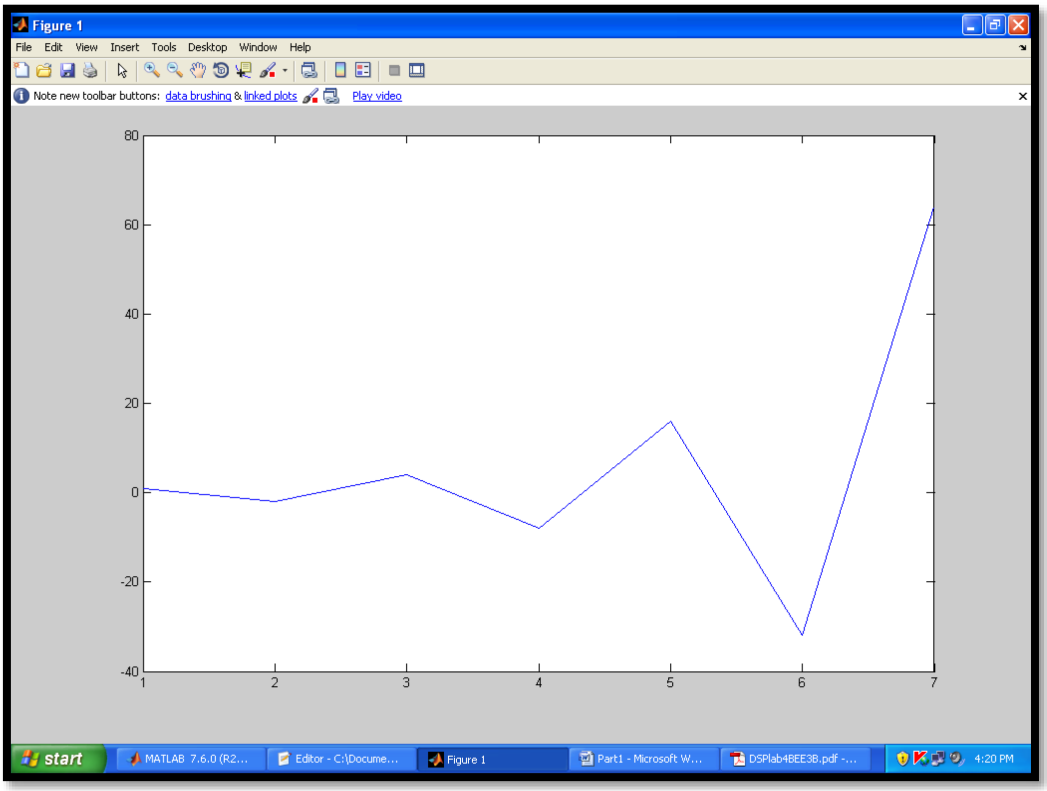

Figure No.5a (Plot of xrl)

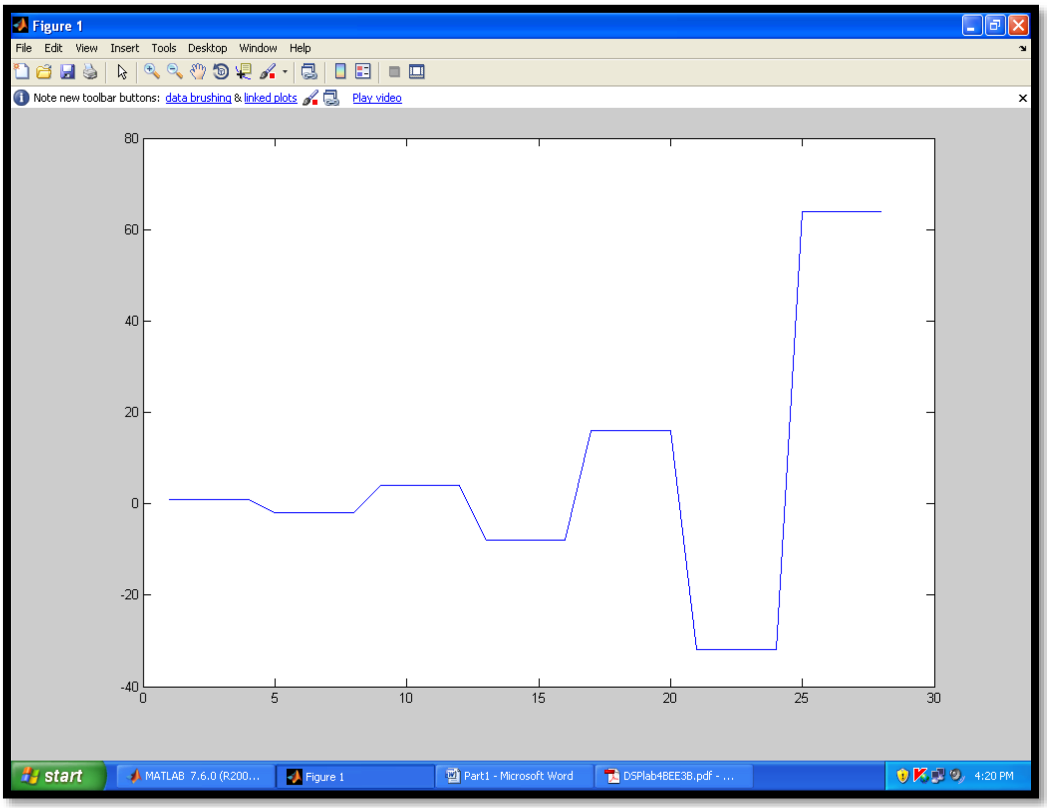

Figure No.5b (Plot of xrl hold)

Part (b)

Now return to the down-sampled lighthouse image, and process all the rows of xx3 to fill in the missing points. Use the zero-order hold idea from part (a), but do it for an interpolation factor of 3. Call the result xholdrows. Display xholdrows as an image, and compare it to the down sampled image xx3; compare the size of the images as well as their content.

Solution

MATLAB Code:

```
clear all
clc
load lighthouse
xx=lighthouse
xx3 = xx(1:3:end,1:3:end); %downsampled by 3
L=size(xx3);
nn = ceil((0.999:1:3*L(1))/3);%sample and hold rows
xholdrows = xx3(nn,:);
imshow(xholdrows)
```

Comparison:

xx3	109x142
xholdrows	327x142

The rows of the image are sharper but the column content is still blurred i.e. missing content due to earlier down sampling.

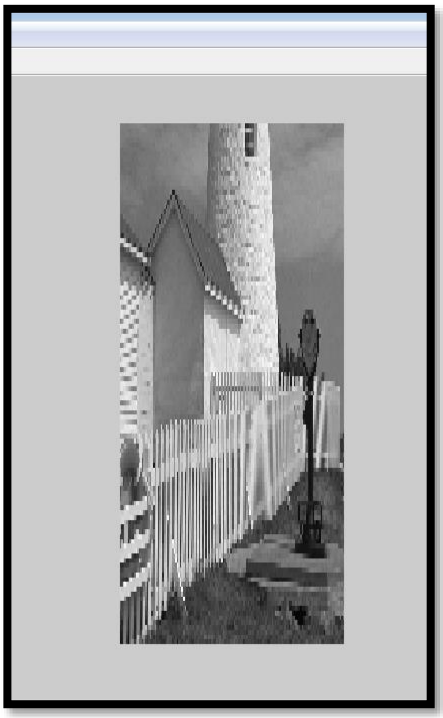

Figure No.6 (Reconstructed image from xholdrows)

Part (c)

Now process all the columns of xholdrows to fill in the missing points in each column and call the result xhold. Compare the result (xhold) to the original image lighthouse. Include your code for parts (b) and (c) in the lab report.

Solution:

MATLAB Code:

```
clear all
clc
load lighthouse
xx=lighthouse
xx3 = xx(1:3:end,1:3:end); %down sampled by 3
```

```
L=size(xx3);
nn = ceil((0.999:1:3*L(1))/3);  %sample and hold rows
mm = ceil((0.999:1:3*L(2))/3);  %sample and hold columns
xhold = xx3(nn,mm);
imshow(xhold)
```

MATLAB Graph:

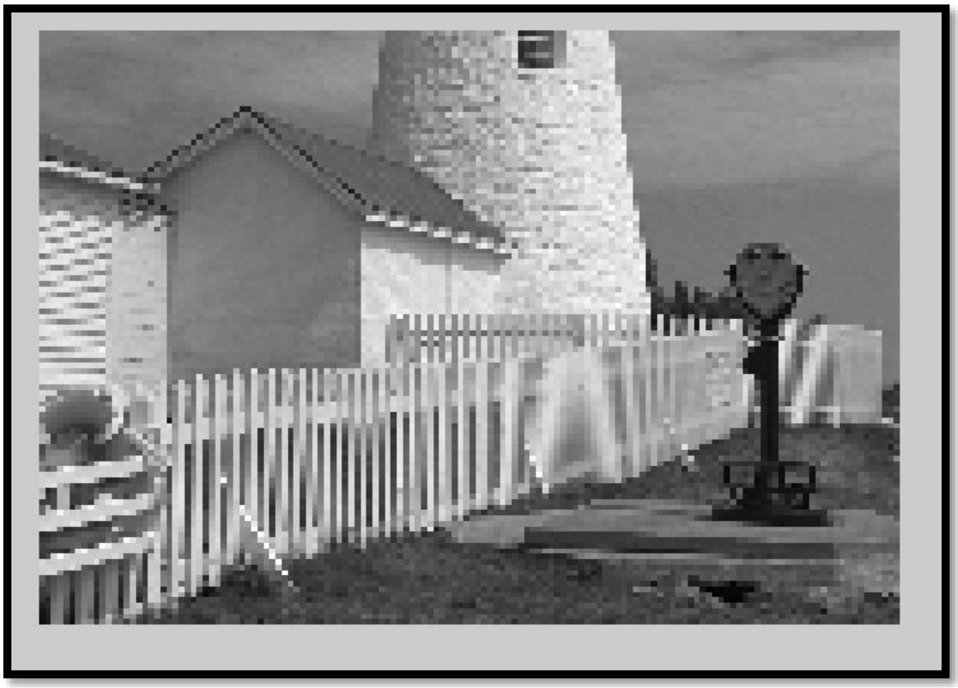

Figure No.7 (result of xhold, interpolated image)

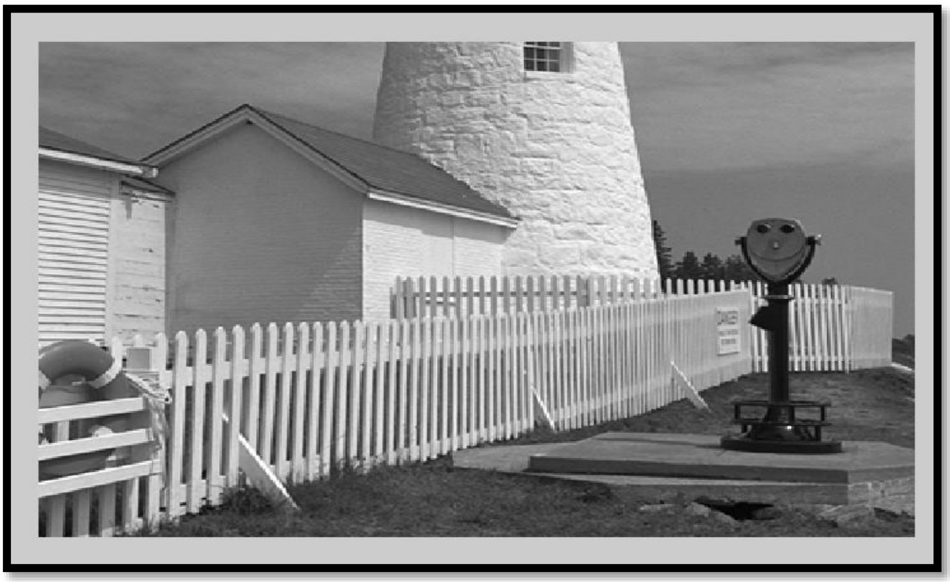

Figure No.8 (Original Sample)

comparison

xx	326x426
xhold	327x426

When the image was reconstructed, the content was not recovered and we see the image is distorted and blurred as compared to the original image. By interpolation we were no able to achieve original image quality.

Part (d)

Linear interpolation can be done in MATLAB using the interp1function (that’s “interp-one”).When unsure about a command, use help. Its default mode is linear interpolation, which is equivalent to using the ’*linear’ option, but interp1can also do other types of polynomial interpolation. Here is an example on a 1-D signal:

```
n1 = 0:6;
xr1 = (-2).ˆn1;
tti = 0:0.1:6; %--locations between the n1 indicesxr1
linear = interp1(n1,xr1,tti); %--function is INTERP-ONE
stem(tti,xr1linear)
```

For the example above, what is the interpolation factor when converting xr1to xr1linear?

Answer:

The interpolation factor is 10, since there are ten values between two points.

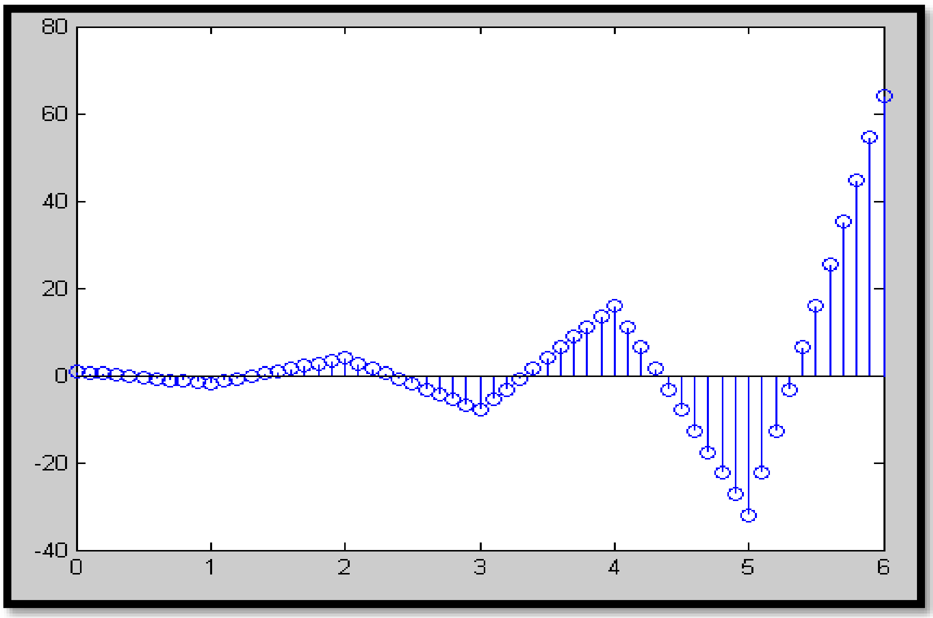

Figure No.9

Part (e)

In the case of the lighthouse image, you need to carry out a linear interpolation operation on both the rows and columns of the down-sampled image xx3. This requires two calls to the interp1 function, because one call will only process all the columns of a matrix.[10] Name the interpolated output image xxlinear. Include your code for this part in the lab report.

Solution:

MATLAB Code:

```
load lighthouse
xx=ww;
xx3 = xx(1:3:end,1:3:end);
uint8(xx3);
L=size(xx3);
n1r = 1:L(1);
n1c = 1:L(2);
ttr = 1:0.333:L(1);
ttc = 1:0.333:L(2);
rowint = interp1 (n1r,xx3,ttr);
transrowint = rowint';
columninttrans = interp1 (n1c, transrowint,ttc);
xxlinear = columninttrans';
imshow(xxlinear);
```

MATLAB Graph:

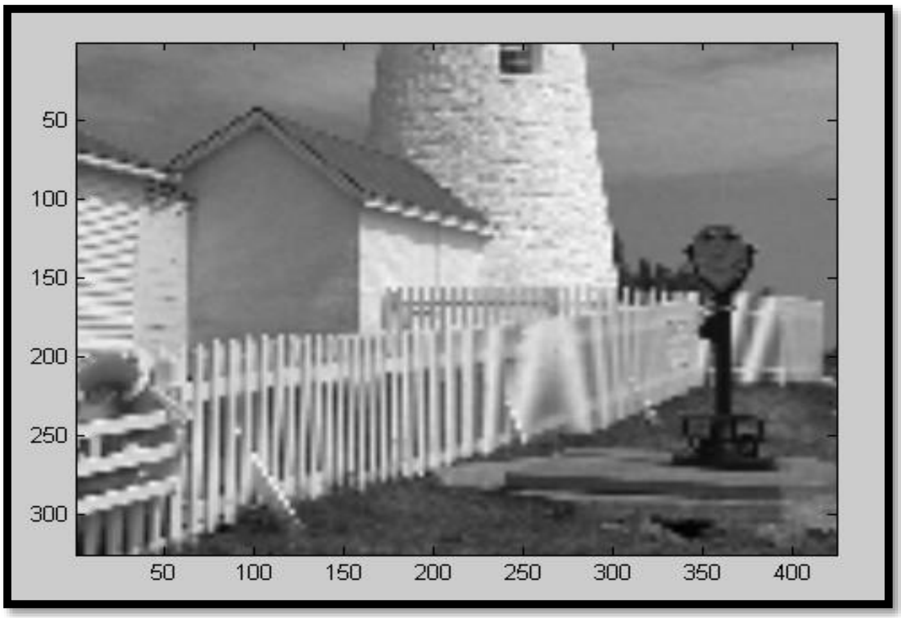

Figure No.10

Part (f)

Compare xxlinear to the original image lighthouse. Comment on the visual appearance of the "reconstructed" image versus the original; point out differences and similarities. Can the reconstruction (i.e., zooming) process remove the aliasing effects from the down-sampled lighthouse image?

Answer:

We have use two different techniques to obtain the original images i. e down sampling and interpolation. Interpolated image is better than the down sampled image, as we obtain the same size in it as that of original. But if we compare interpolated with the original image, we clearly observed that the image is still blurry and edges are not sharp. We cannot obtain the original image unless we fill in the same values of samples that we removed during down sampling.

Part (f)

Compare the quality of the linear interpolation result to the zero-order hold result. Point out regions where they differ and try to justify this difference by estimating the local frequency content. In other words, look for regions of "low-frequency" content and "high-frequency" content and see how the interpolation quality is dependent on this factor. A couple of questions to think about: Are edges low frequency or high frequency features? Are the fence posts low frequency or high frequency features? Is the background a low frequency or high frequency feature?
Comment: You might use MATLAB's zooming feature to show details in small patches of the output image. However, be careful because zooming does its own interpolation, probably a zero-order hold.

Answer:

The two different types of interpolation techniques studied above are linear interpolation and zero hold interpolation. Zero order hold interpolation holds a specific value until next sample arrives whereas linear interpolation adds value with linear difference between them. For the two figures we can clearly distinguish the difference between two techniques but actually linear interpolation has a better chance of filling up the exact samples in the signal.

The interpolation quality is greatly dependent on all high frequency content as when down sampling this high frequency content is more vulnerable to data loss , so when we reconstruct/interpolate the down sampled image using any of the two techniques above, we still are not able to make up for the loss of content that happened during down sampling. These, high frequency areas are mostly the fence areas, the edges of objects etc, the low frequency contents e.g. the background can be easily recovered as seen by in the interpolation figures in above questions.

 Figure No.11	 **Figure No.12**

Conclusion:

This lab was really helpful as we observe the practical usage of our DSP knowledge in image processing. We studied the effect of down sampling on the image quality and then reconstructed the image using different and linear interpolation techniques. We also studied the effect of aliasing and see its effect on signal. In short, this lab was really helpful in reconstruction of original signal.
