ECE438 - Laboratory 11a
Increasing Image Resolution

Trey E. Shenk

December 6, 2011

1 Introduction

When displaying or printing an image, it is often necessary to change its resolution. If a poster-size print is to be made, the most obvious method is to simply enlarge the pixels (sometimes called zero-hold or nearest neighbor). This simple approach gives a blocky output. In this lab, you will explore other methods of creating a more aesthetic image.

Figure 1: 'Lena' and an enlargement using nearest neighbor.

2 Increasing Image Resolution

2.1 The Case for One Dimension

You have already had a lab dealing with upsampling in the 1-D case. As a review, first the original signal was upsampled by factor L by inserting $L-1$ zeros in between each sample point. The signal was then lowpass filtered. These steps and the result is shown Figure 2.

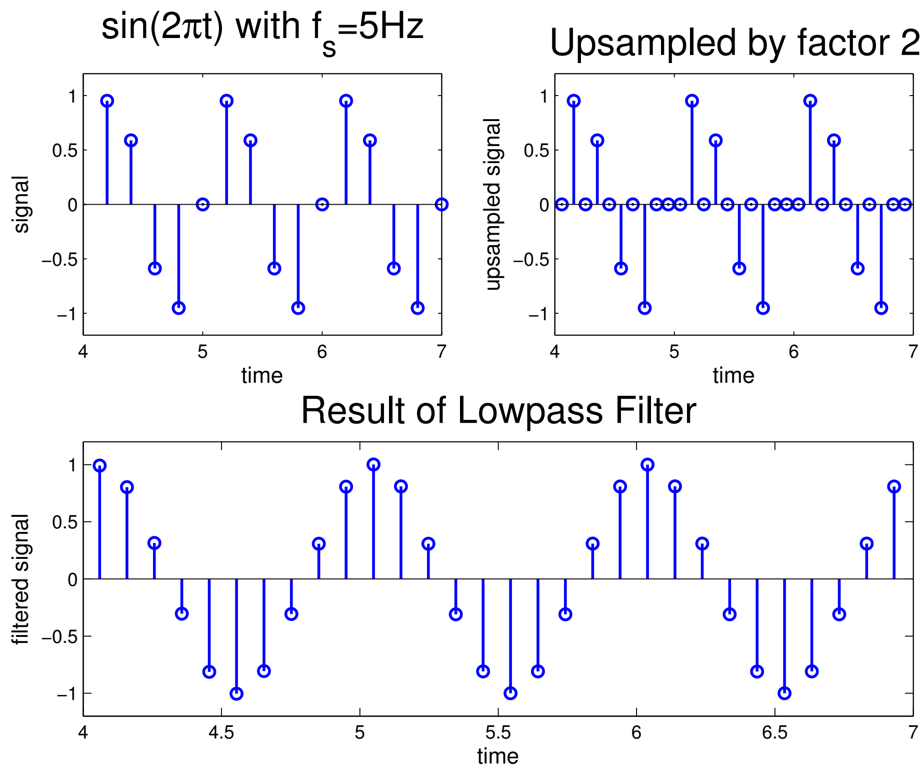

Figure 2: An example of up sampling in one dimension.

Most of what you have learned in 1-D can be easily extended to 2-D. The concepts of zero-padding, aliased images, and filtering can be applied to images with little modification.

2.2 Your Job

In this lab, your goal is to take a picture and increase its resolution. First, download the picture of lena. Now increase the resolution of the image. There are many ways of approaching this problem. You can use concepts that were developed in previous labs to accomplish this. The deliverables for this part are:

- A short description of your strategy. Include reasons for choosing your approach.
- What parameters, if any, did you have to tweak to get your output?
- The results of your procedure. Show images upsampled by factor 2, 4, and 8.
- Discuss the output. Are there artifacts? What might you do to remove the artifacts? What problems would you expect if you used a factor of 32?
- Include your code.

3 Hints

Remember what you have done in 1-D. You can use the same type of filter used in 1-D to design an 2-D separable filter.

A few MATLAB commands that would be useful for making your filter are: firpmord, firpm, and repmat. The syntax of firpmord is `[N, Fo, Ao, W] = firpmord(F, A, DEV, Fs)`. where 'F' are the cutoff frequencies (passband and stopband edges), 'A' are the amplitudes of the bands, 'DEV' are the max ripples in the bands, and 'Fs' is the sampling frequency. Fs and F are related and one can be chosen arbitrarily. The function `firpm` can then be used to design a FIR filter using `b = firpm(N, Fo, Ao, W)`, with `b` being the filter coefficients.

To easily form a 2-D separable filter, you can use the command `repmat`. An example of using `repmat` is:

```
x = [1, 2, 3, 4]';

y = [6, 4, 2, 0];

z = x * y;
```