GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2025 Spring 2006
Lab #7: Frequency Response: Filtering Images

Date: 1–7 Mar 2006

You should read the Pre-Lab section of the lab and do all the exercises in the Pre-Lab section before your assigned lab time. You **MUST** complete the online Pre-Post-Lab exercise on Web-CT at the beginning of your scheduled lab session. You can use MATLAB and also consult your lab report or any notes you might have, but you cannot discuss the exercises with any other students. You will have approximately 20 minutes at the beginning of your lab session to complete the online Pre-Post-Lab exercise. The Pre-Post-Lab exercise for this lab includes some questions about concepts from the previous Lab report as well as questions on the Pre-Lab section of this lab.

The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. After completing the warm-up section, turn in the verification sheet to your TA.

Forgeries and plagiarism are a violation of the honor code and will be referred to the Dean of Students for disciplinary action. You are allowed to discuss lab exercises with other students and you are allowed to consult old lab reports, but you cannot give or receive written material or electronic files. Your submitted work should be original and it should be your own work.

It is only necessary to turn in Section 4 as this week's lab report; the lab report format is **Informal**.
The report will be due the next time your lab meets.

---

1 Introduction

The goal of this lab is to study the response of FIR filters to inputs such as complex exponentials and sinusoids. In the experiments of this lab, you will use `firfilt()`, or `conv()`, to implement filters and `freqz()` to obtain the filter's frequency response.[1] As a result, you should learn how to characterize a filter by knowing how it reacts to different frequency components in the input.

2 Pre-Lab

This lab also introduces a practical filter, the nulling filter. Nulling filters can be used to remove sinusoidal interference, e.g., jamming signals in a radar or communication system.

2.1 Frequency Response of FIR Filters

The output or *response* of a filter for a complex sinusoid input, $e^{j\hat{\omega}n}$, depends on the frequency, $\hat{\omega}$. Often a filter is described solely by how it affects different input frequencies—this is called the *frequency response*.

For example, the frequency response of the two-point averaging filter

$$y[n] = \tfrac{1}{2}x[n] + \tfrac{1}{2}x[n-1]$$

[1] If you are working at home and do not have the function `freqz.m`, there is a substitute available called `freekz.m`. You can find it in the *SP-First Toolbox,* or get it from the ECE-2025 WebCT page.

can be found by using a general complex exponential as an input and observing the output or response.

$$x[n] = Ae^{j(\hat{\omega}n + \phi)} \tag{1}$$
$$y[n] = \tfrac{1}{2}Ae^{j(\hat{\omega}n + \phi)} + \tfrac{1}{2}Ae^{j(\hat{\omega}(n-1) + \phi)} \tag{2}$$
$$= Ae^{j(\hat{\omega}n + \phi)}\tfrac{1}{2}\left\{1 + e^{-j\hat{\omega}}\right\} \tag{3}$$

In (3) there are two terms, the original input, and a term that is a function of $\hat{\omega}$. This second term is the frequency response and it is commonly denoted[2] by $H(e^{j\hat{\omega}})$.

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \tfrac{1}{2}\left\{1 + e^{-j\hat{\omega}}\right\} \tag{4}$$

Once the frequency response, $H(e^{j\hat{\omega}})$, has been determined, the effect of the filter on any complex exponential may be determined by evaluating $H(e^{j\hat{\omega}})$ at the corresponding frequency. The output signal, $y[n]$, will be a complex exponential whose complex amplitude has a constant magnitude and phase. The phase of $H(e^{j\hat{\omega}})$ describes the phase change of the complex sinusoid and the magnitude of $H(e^{j\hat{\omega}})$ describes the gain applied to the complex sinusoid.

The frequency response of a general FIR linear time-invariant system with filter coefficients $\{b_k\}$ is

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \sum_{k=0}^{M} b_k e^{-j\hat{\omega}k} \tag{5}$$

2.1.1 MATLAB Function for Frequency Response

MATLAB has a built-in function for computing the frequency response of a discrete-time LTI system. The following MATLAB statements show how to use `freqz` to compute and plot both the magnitude (absolute value) and the phase of the frequency response of a two-point averaging system as a function of $\hat{\omega}$ in the range $-\pi \le \hat{\omega} \le \pi$:

```
bb = [0.5, 0.5];          %-- Filter Coefficients
ww = -pi:(pi/100):pi;     %-- omega hat
H = freqz(bb, 1, ww);     %<--freekz.m is an alternative
subplot(2,1,1);
plot(ww, abs(H))
subplot(2,1,2);
plot(ww, angle(H))
xlabel('Normalized Radian Frequency')
```

For FIR filters, the second argument of `freqz( _, 1, _ )` must always be equal to `1`. The frequency vector `ww` should cover an interval of length 2π for $\hat{\omega}$, and its spacing must be fine enough to give a smooth curve for $H(e^{j\hat{\omega}})$. Note: we will always use capital `H` for the frequency response.[3]

2.2 Periodicity of the Frequency Response

The frequency responses of discrete-time filters are *always* periodic with period equal to 2π. Explain why this is the case by stating a definition of the frequency response and then considering two input sinusoids whose frequencies are $\hat{\omega}$ and $\hat{\omega} + 2\pi$.

$$x_1[n] = e^{j\hat{\omega}n} \qquad \text{versus} \qquad x_2[n] = e^{j(\hat{\omega}+2\pi)n}$$

Consult Chapter 6 for a mathematical proof that the outputs from each of these signals will be identical (basically because $x_1[n]$ is equal to $x_2[n]$.) **The implication of periodicity is that a plot of $H(e^{j\hat{\omega}})$ only has to be made over the interval $-\pi \le \hat{\omega} \le \pi$.**

[2] The notation $H(e^{j\hat{\omega}})$ is used in place of $\mathcal{H}(\hat{\omega})$ for the frequency response because we will eventually connect this notation with the z-transform, $H(z)$, in Chapter 7.

[3] If the output of the `freqz` function is not assigned, then plots are generated automatically; however, the magnitude is given in decibels which is a logarithmic scale. For linear magnitude plots a separate call to `plot` is necessary.

2.3 Frequency Response of the Four-Point Averager

In Chapter 6 we examined filters that compute the average of input samples over an interval. These filters are called "running average" filters or "averagers" and they have the following form for the L-point averager:

$$y[n] = \frac{1}{L}\sum_{k=0}^{L-1} x[n-k] \tag{6}$$

(a) Use Euler's formula and complex number manipulations to show that the frequency response for the 4-point running average operator is given by:

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \left(\frac{2\cos(0.5\hat{\omega}) + 2\cos(1.5\hat{\omega})}{4}\right) e^{-j1.5\hat{\omega}} = C(\hat{\omega})e^{j\psi(\hat{\omega})} \tag{7}$$

(b) Implement (7) directly in MATLAB. Use a vector that includes 400 samples between $-\pi$ and π for $\hat{\omega}$. Make plots of $C(\hat{\omega})$ and $\psi(\hat{\omega})$ versus $\hat{\omega}$, but keep in mind that these are not necessarily plots of the magnitude and phase. You would have to use `abs()` and `angle()` to extract the magnitude and phase of the frequency response for plotting.

(c) In this part, use `freqz.m` or `freekz.m` in MATLAB to compute $H(e^{j\hat{\omega}})$ numerically (from the filter coefficients) and plot its magnitude and phase versus $\hat{\omega}$. Write the appropriate MATLAB code to plot both the magnitude and phase of $H(e^{j\hat{\omega}})$. Follow the example in Section 2.1.1. The filter coefficient vector for the 4-point averager is defined via:

```
bb = 1/4*ones(1,4);
```

Recall that the function `freqz(bb,1,ww)` evaluates the frequency response for all frequencies in the vector `ww`. It uses the summation in (5), not the formula in (7). The filter coefficients are defined in the assignment to vector `bb`. How do your results compare with part (b)?
Note: the plots should not be identical, but you should be able to explain why they are equivalent by converting the minus sign in the negative values of $C(\hat{\omega})$ to a phase.

2.4 The MATLAB FIND Function

Often signal processing functions are performed in order to extract information that can be used to make a decision. The decision process inevitably requires logical tests, which might be done with `if-then` constructs in MATLAB. However, MATLAB permits vectorization of such tests, and the `find` function is one way to determine which elements of a vector meet a certain logical criterion. In the following example, `find` extracts all the numbers that "round" to 3:

```
xx = 1.4:0.33:5, jkl = find(round(xx)==3), xx(jkl)
```

The argument of the `find` function can be any logical expression, and `find` returns a list of indices where that logical expression is true. See `help` on `relop` for information.
Now, suppose that you have a frequency response:

```
ww = -pi:(pi/500):pi; HH = freqz( 1/4*ones(1,4), 1, ww );
```

Use the `find` command to determine the indices where `HH` is zero, or very small. Then use those indices to display the list of frequencies where `HH` is zero. Since there might be round-off error in calculating `HH`, the logical test should be a test for those indices where the magnitude (absolute value in MATLAB) of `HH` is less than some rather small number, e.g., 1×10^{-6}. Compare your answer to the frequency response that you plotted for the four-point averager in Section 2.3.

3 Warm-up

The first objective of this warm-up is to use a MATLAB GUI to demonstrate the frequency response If you are working in the ECE lab it is **NOT** necessary to install the GUI; otherwise, you must download the *SP-First* ZIP file and install it. The frequency response demo, `dltidemo`, is part of the *SP-First Toolbox*.

3.1 LTI Frequency Response Demo

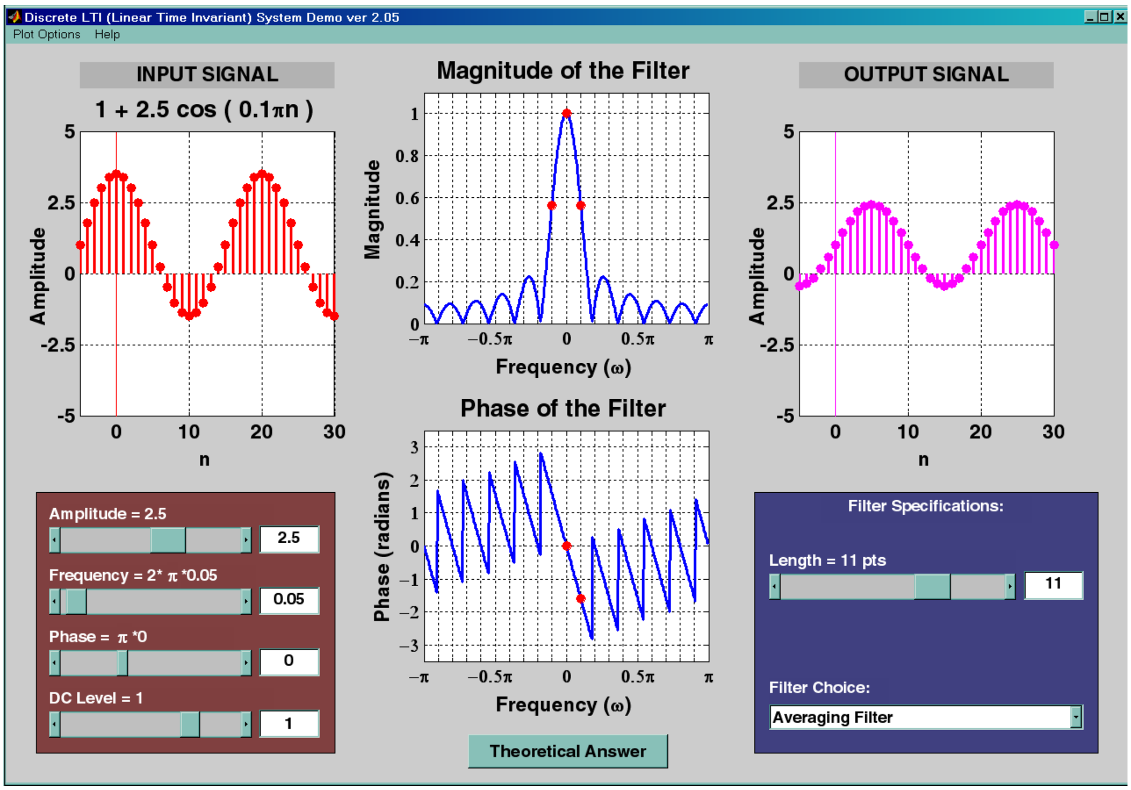

Figure 1: DLTI demo interface. The frequency label is ω because MATLAB won't display $\hat{\omega}$.

The `dltidemo` GUI illustrates the "sinusoid-IN gives sinusoid-OUT" property of LTI systems. In this demo, you can change the amplitude, phase and frequency of an input sinusoid, $x[n]$, and you can change the digital filter that processes the signal. Then the GUI will show the output signal, $y[n]$, which is also a sinusoid (at the same frequency). Figure 1 shows the interface for the `dltidemo` GUI. It is possible to see the formula for the output signal, if you click on the `Theoretical Answer` button located at the bottom-middle part of the window. The digital filter can be changed by choosing different options in the `Filter Specifications` box in the lower right-hand corner.

In the Warm-up, you should perform the following steps with the `dltidemo` GUI:

(a) Set the input to $x[n] = 1.5\cos(0.1\pi(n-4))$

(b) Set the digital filter to be a 9-point averager.

(c) Determine the formula for the output signal and write it in the form: $y[n] = A\cos(\hat{\omega}_0(n - n_d))$.

(d) Using n_d for $y[n]$ and the fact that the input signal had a peak at $n = 4$, determine the amount of delay through the filter. In other words, how much has the peak of the cosine wave shifted?

Instructor Verification (separate page)

(e) Now, determine the length of the averaging filter so that the output will be zero, i.e., $y[n] = 0$. Use the GUI to show that you have the correct filter to zero the output. If the filter length is more than 15, you will have to enter the "Filter Specifications" with the `user Input` option.

Instructor Verification (separate page)

(f) When the output is zero, the filter acts as a *Nulling Filter,* because it eliminates the input at $\hat{\omega} = 0.1\pi$. Which other frequencies $\hat{\omega}$ are also nulled? Find at least one.

3.2 Cascading Two Systems

More complicated systems are often made up from simple building blocks. In Fig. 2, two FIR filters are shown connected "in cascade."

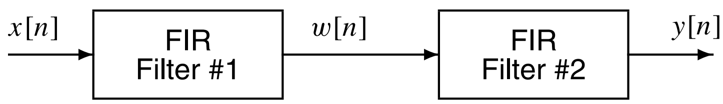

Figure 2: Cascade of two FIR filters.

Assume that the system in Fig. 2 is described by the two equations

$$w[n] = x[n] - x[n-1] \qquad \text{(FIR FILTER \#1)}$$
$$y[n] = w[n] + w[n-1] \qquad \text{(FIR FILTER \#2)}$$

(a) Use `freqz()` or `freekz()` in MATLAB to get the frequency responses. Plot the magnitude and phase of the frequency response for Filter #1, and also for Filter #2. Which one of these filters is a *highpass filter*?

(b) Filter #2 is a "nulling filter." Determine the frequency $\hat{\omega}$ of the sinusoid that is removed by Filter #2.

(c) Plot the magnitude and phase of the frequency response of the overall cascaded system.

(d) Explain how the individual frequency responses in part (a) are combined to get the overall frequency response in part (b). Comment on the magnitude combinations as well as the phase combinations.

Instructor Verification (separate page)

3.3 Debugging

In the following MATLAB function, `myfilt.m`:

```
function bb = myfilt(ncascades)
% create filter coefficients, a little bit at a time
%   cascade many first-difference filters
bb = 1;
for kk = 1:ncascades
  bb = conv(bb,[1,-1]);
end
```

show that you can use the MATLAB debugger to stop *during* the second iteration of the loop and plot the frequency response for the filter coefficients after the second iteration is complete. Suppose that the function is called from the command line via: `hh = myfilt(5).`

Instructor Verification (separate page)

4 Lab Exercises

For the exercises in this section use the *Tools* image available in `tools.mat` in the *SP-First* Toolbox.

4.1 Blurring an Image

If you want to blur an image with a 7-point averaging filter, the image would be processed first along the rows and then along the columns. The Row filter would do the filtering only in the horizontal direction:

$$y_1[m,n] = \frac{1}{7}\sum_{\ell=0}^{6} x[m, n-\ell] \qquad \text{for } 1 \le m \le M$$

Next, a Column filter would process the columns of the horizontally filtered image:

$$y_2[m,n] = \frac{1}{7}\sum_{k=0}^{6} y_1[m-k, n] \qquad \text{for } 1 \le n \le N$$

This would create yet another image that is filtered in both directions. Fortunately, MATLAB has a function `conv2( )` that will do these operations without `for` loops. It performs a more general filtering operation than row/column filtering, but it can do these simple 1-D operations also.

(a) To filter the image in the horizontal direction using a 7-point averager, use the following statement:

```
bh = ones(1,7)/7;
y1 = conv2(xx, bh);
```

In other words, the filter coefficients `bh` for the 7-point averager are stored in a *row* vector. This causes `conv2( )` to filter only in the *horizontal* direction. Display the input image `xx` and the output image `y1` on the screen at the same time. Compare the two images and describe what you see (qualitatively). Extract one row from the output image and compare it to the same row from the input image. Each row is a 1-D signal, so plot them together and use the plot to explain why the 7-point averaging filter causes blurring.

(b) Now filter the image `y1` in the vertical direction to produce the image `y2`. Plot all three of the images `xx`, `y1`, and `y2` on the screen at the same time.

(c) Repeat part (b) for a 21-point moving averager, and compare the outputs from the 21-point and 7-point averaging filters. Which one causes a more severe degradation of the original image? Include a discussion of why there are differences based on the frequency response of the 7-point and 21-point averaging filters.

4.2 Frequency Content of an Image

Filters can be used to investigate the frequency content of an image. From what we have seen so far, lowpass filtering of an image causes blurring. In this exercise we would like to prove that high-pass (or band-pass) filtering will "sharpen" an image. In the next section, we will examine a system for edge detection that also uses high-pass filters. For this exercise you should use either `tools.mat` as the test image.

(a) To make this demonstration we need some filters that give a smooth frequency response. For this purpose we will use "Gaussian" shaped functions for the FIR filter coefficients $\{b_k\}$. First, of all we need a low-pass filter:

$$b_k = \begin{cases} 1.1 & \text{for k=10} \\ e^{-0.06(k-10)^2} & \text{for } k = 0, 1, 2, \ldots, 8, 9, 11, 12, \ldots 20 \\ 0 & \text{elsewhere} \end{cases}$$

For this filter, plot the impulse response as a stem plot, and then plot the magnitude of the frequency response versus $\hat{\omega}$. Put them on the same page with a two-panel subplot.

(b) Produce a band-pass filter by the following definition:

$$\tilde{b}_k = \begin{cases} \cos(0.3\pi(k-10))e^{-0.12(k-10)^2} & \text{for } k = 0, 1, 2, \dots 20 \\ 0 & \text{elsewhere} \end{cases}$$

In this case the filter coefficients $\{\tilde{b}_k\}$ are a Gaussian multiplied by a cosine. Once again, plot the impulse response as a stem plot, and the magnitude of the frequency response versus $\hat{\omega}$. Put them on the same page with a two-panel subplot. Explain why this filter is called a bandpass filter (BPF).

(c) Load the test image from `tools.mat`. Filter this image along both its rows and columns with the LPF and save the result as $y[m,n]$ for the next part. Display the image to see the effect of the filter. What is the frequency content of $y[m,n]$? Is is mostly low frequency or high frequency?

(d) Filter the test image along both its rows and columns with the BPF $\{\tilde{b}_k\}$ and save the result as $v[m,n]$ for the next part. What is the frequency content of $v[m,n]$? Is is mostly low frequency or high frequency?
Note: if you try to display this image you will experience problems because is has negative values which must be rescaled to get a positive image for the CRT. The MATLAB function `show\_img` uses auto-scaling to take of that.

(e) The blurred image in $y[m,n]$ can be improved by adding in some of the band-pass filtered image $v[m,n]$. Use the following combination:

$$(1-\alpha)y[m,n] + \alpha v[m,n]$$

where α is a fraction, i.e., $0 \le \alpha \le 1$. In effect, α controls how much of the final result will come from the "band-pass filtered" image. Try to find the **best** value of α so that the combination looks as close as possible to the original. Use contrast stretching to made the backgrounds have the same "whiteness." Explain why the result looks sharper when the higher frequency information is included.

Note: When the contrast of the image has been compressed significantly so that the resulting display looks mostly gray, you can overcome the limited contrast by changing the *colormap* of the display in MATLAB. Normally a gray-scale image would be displayed using the `gray(256)` colormap which consists of a linear set of 256 values going from 0 to 1. The colormap is also a 256×3 matrix because it is normally used for the red, green and blue (RGB) image planes of a color image. If you want to "stretch" the contrast of an image, then the colormap has to have a set of zeros, followed by a linear ramp from 0 to 1, followed by ones. The length of the linear ramp would obviously be less than 256. Here is a snippet of MATLAB code that will produce such a colormap.

```
colormap(clip(linspace(-0.1,1.25,256)',0,1)*ones(1,3))
```

To see how it works made a plot of the `clip` part of the MATLAB statement (or list the values). Changing the two arguments -0.1 and 1.25 will control the lower and upper stretching of the contrast—the lower end being black and the upper end white.

4.3 The Method of Synthetic Highs (Optional)

"De-blurring" is a rather difficult process to carry out. However, the idea presented in the previous section can form the basis of a practical "de-blurring" system.[4]

[4] One difficulty with the exercise presented here is that it attempts to deblur the entire image by doing one operation on the entire image. In practice, the deblurring would have to be done locally with different operations on different parts of the image, because some regions of the image are background while other regions have details.

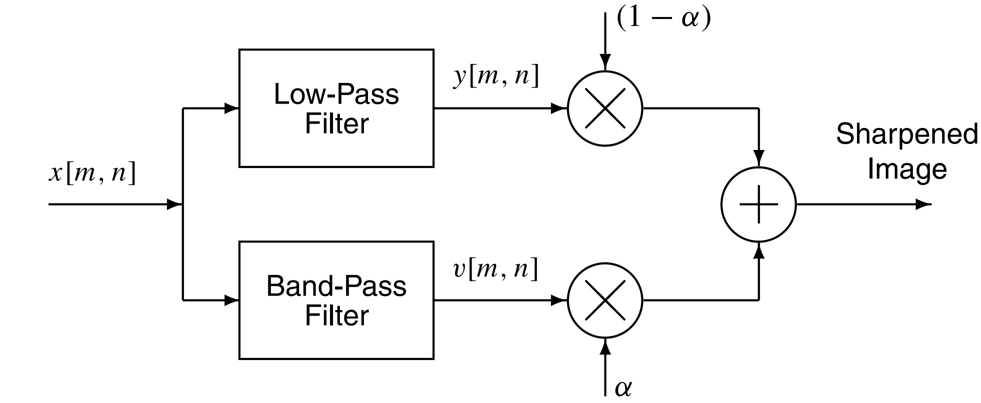

Figure 3: Test set-up for varying low and high frequency content of an image. The parameter α should be between 0 and 1.

We need to add in some high frequency information to sharpen an image.

We can't directly use the method of the previous section because the image $v[m,n]$ came from band-pass filtering the original. In a realistic "de-blurring' problem only the blurred image would be available as input to the de-blurring system. Somehow, we must construct the high-frequency content from the blurred image. One procedure for doing this is the method of "synthetic highs."

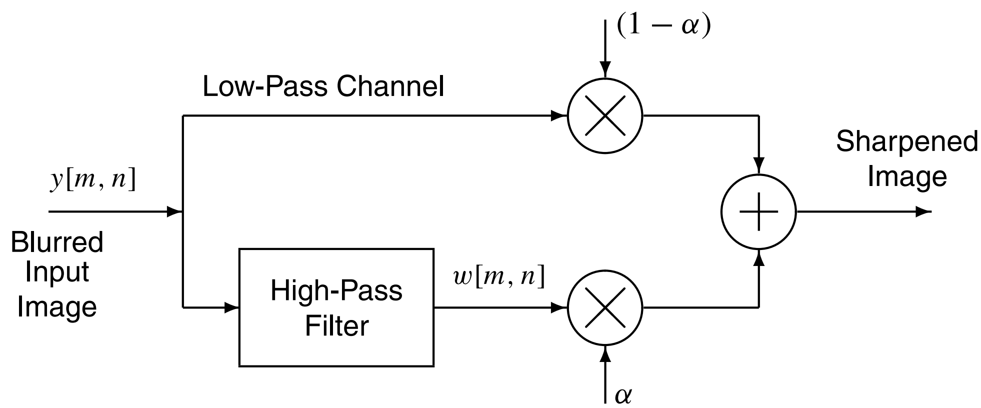

Figure 4: Method of Synthetic Highs. High frequency content of the image is generated by a second difference filter.

(a) Construct a high pass filter as the second difference filter:

$$y_1[n] = x[n] - x[n-1]$$
$$y[n] = y_1[n] - y_1[n-1]$$

which is the cascade of two first-difference filters. Plot the frequency response (magnitude) of this filter.

(b) Compute a synthetic-high image by passing the blurred image $y[m,n]$ through the second difference to obtain $w[m,n]$. Make sure you process all the rows and then all the columns.

(c) Explain the frequency content of $w[m,n]$ by considering that it was constructed by passing the original through the cascade of the blurring filter $\{b_k\}$ and then the second difference. Plot the frequency response of that cascade, and compare to the frequency response of the bandpass filter $\{\tilde{b}_k\}$ from the previous section.

(d) *Sharpening via Synthetic Highs:* Now use the same combination as before:

$$(1 - \alpha)y[m, n] + \alpha w[m, n]$$

where α is a fraction, i.e., $0 \leq \alpha \leq 1$. The value of α controls how much high-frequency content to add in for the purpose of sharpening.

(e) Try to find the best choice of α to get a sharpened result. Comment on how well your processing works, realizing that it is not a perfect process. Show an example of the enhanced versus blurred image. Use contrast stretching to improve the comparison of images.

Lab #7
ECE-2025
Spring-2006
INSTRUCTOR VERIFICATION PAGE

For each verification, be prepared to explain your answer and respond to other related questions that the lab TA's or professors might ask. Turn this page in at the end of your lab period.

Name: ______________________________ Date of Lab: __________

Part 3.1(d) Use the `dltidemo` GUI to illustrate the operation of a 9-point averaging filter. Determine the amount of delay through the filter, and write your answer in the space below. Also, determine the location of the zeros of the frequency response.

Verified:____________________ Date/Time:__________

Part 3.1(e) Use the `dltidemo` GUI to find a new digital FIR filter that will null the input signal. Determine the new filter length, and write your answer in the space below.

Verified:____________________ Date/Time:__________

Part 3.2 Plot the frequency response of the two filters in the cascade combination, and then explain how the magnitudes are combined and how the phases are combined to get the overall filter. Check the range of frequencies ($\hat{\omega}$) used for the plot.

Verified:____________________ Date/Time:__________

Part 3.3 Use the debugger to stop execution and plot the frequency response:

Verified:____________________ Date/Time:__________