Exploration into Image Deblurring
Tim Ho
Introduction
Analogous to the one-dimensional signals analyzed in class, images are multidimensional signals that could be altered and manipulated using processing techniques and methodologies. Particularly, the filters and convolutions in the spatial domain could be used to introduce and reduce blurring. However, the task of deblurring often requires information about the distortion, blurring and noise, to be known. For example, introducing blur by convolving an image with an arbitrary PSF (PSF), or impulse response of a system, can be easily reversed by dividing by the fourier transform of the PSF in the frequency domain. Since information about distortion in a practical setting is usually unknown, understanding and achieving "blind" deblurring, or deblurring without prior information, are the research focus of this project.
When it comes to images, fourier transforms to the frequency domain are not as intuitive as the transform of an audio signal to its frequency component. Although the mathematics is the same, the Fourier transform measures spatial frequencies. Images with smooth color transitions will have many low frequencies, whereas those with abrupt changes in color will have many high frequencies. This characteristic will serve as a foundation for methods adopted and proposed in this project.
Three ways of blur reduction will be tested in this project: processing through fourier transform, high pass filtering, and processing through built-in functions of the Matlab Image Processing Toolbox. Initially, blur will be introduced to an arbitrary image, in this case a picture of a marketplace on Duke East Campus. Noise will also be introduced at varying degrees to test the tolerance of the reduction technique. Blur reduction techniques will be applied to this initial image for characterization. Then, the same techniques will be applied to an arbitrarily chosen blurred image with no prior information of distortion.

Methods
Blur Introduction
A circular averaging filter is chosen to be the PSF using the *fspecial* function and specifying 'disk', which creates a predefined 2-D filter. Three radii of sizes 5, 10, and 20 were tested. Blur is introduced by convolving the image with the circular average filter through the use of the *conv2* command in Matlab.

Noise Introduction
Noise is introduced to an image through the use of the *imnoise* function from the Image Processing Toolbox. Gaussian noise was chosen in this case, however, the choice was arbitrary. Three noise of variance $10^{-5},\ 10^{-10},\ and\ 10^{-15}$are tested.

Blur Reduction Through Fourier Transform
The fourier transform of the blurred image was taken using the *fft2* command. Using the formula $y = h * x$ in the spatial domain corresponding to $Y = HX$in the frequency domain, arbitrary H's, or frequency responses, could be created to deblur the image by the means of $X = Y/H$. In this case, the blurred image created by the circular averaging filter of 20 was chosen as it introduced the most blurr. In order to ensure that the dimensions matches, the following code was written:

```
%%pad zeros to correct dimension, imgb = blurred image, psf = arbritary
%%point state function
newh = zeros(size(imgb));
psfsize = size(psf);
newh(1: psfsize(1), 1:psfsize(2))=psf;
H = fft2(newh);
```

Figure 1: Code Creating Padded Frequency Response

Through the use of the frequency response, image was deblurred by conducting the operation $Y./H$and transformed back to the spatial domain using an inverse Fourier transform through the *ifft2* command.

Viewing Frequency Spectra

To view the frequency spectra, the *imshow* function from the Image Processing Toolbox was used. In addition, the fftshift command was used to center the zero of the spectra to the center. Absolute value and log were taken as the frequencies span several orders of magnitude.

Blur Reduction Through High Pass Filter

An ideal high pass filter was created by zeroing frequencies below a predetermined threshold. As a result, a square of zero is created at the center of the frequency spectra. Three thresholds of 500, 1000, and 1500 were tested. The threshold was determined arbitrarily from the frequency spectra. The following code was written to create the ideal high pass filter.

```
%%highpass filter; k = threshold; n = horizontal variable; m = verticle
%%variable; Ysize determined dimension of the fourier transform of image;
%%Y1n is the zero-centered magnitude spectra
k = 1500;
n=Ysize(1);
m=Ysize(2);
array2add = zeros(k);
if n > k && m > k
   Y1n((n/2 - k/2 + 1):(n/2 - k/2 + k), (m/2 - k/2 + 1):(m/2 - k/2 + k)) =
   array2add;
end
```

Figure 2: Code Creating Ideal High Pass Filter

Blur Reduction Through Built in Functions of the Image Processing Toolbox

The *blindeconv* function is used for reduction. Arbitrary circular averaging filter was again chosen for the operation to serve as the initial estimate of the PSF. Three numbers of iteration were chosen of 5, 10, and 20.

Blur Reduction of Total Blind Images

All three methods of blur reduction were tested on a “blind” image. Since both the Fourier transform method and *blindeconv* require an input of an arbitrary PSF to serve as an initial guess, three sizes of circular averaging filter was tested of radius 5, 10, and 20.

Noise Reduction

Noise reduction was carried out through two methods: averaging filter and median filter. The averaging filter was created using the *filter2* function and passing-in an averaging filter using *fspecial*(‘average’). The median filtering was carried out using *medfilt2*.

Results and Discussion

Blur and Noise Introduction

As the radius of the circular averaging filter increased, the amount of blur increased (Figure 3). This makes intuitive sense as a larger filter would average across a larger area, which smoothes images. The smoothing effect induces blurring.

Figure 3: Raw Image (left) Compared with Blurred Image with Circular Averaging Filter of Size 5 (middle left), 10 (middle right), and 20 (right)

However in order to test the noise tolerance of the blur reduction methods tested in this project, noise was also induced. Despite the wide range of the variances, the noise seems to be visually congruent (Figure 4). Theoretically, a higher variance should mean more noise induced, as standard deviance measures expected amplitude and variance measures expected power.

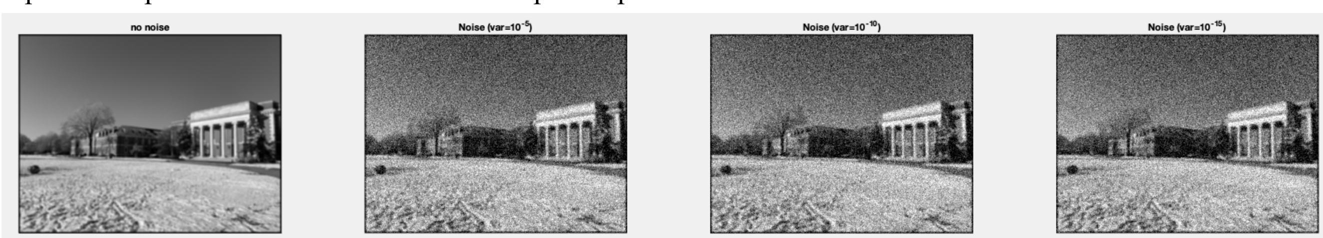


Figure 4: Blurred Image (left) Compared with Blurred Image with Noise of variance 10^-5 (middle left), 10^-10 (middle right), and 10^-15 (right)

Blur Reduction Through Fourier Transform

The results of the reduction through Fourier transform is shown in Figure 5. As expected, when the same PSF transformed into the frequency domain and used as frequency response for deblurring, the deblurring was successful. However, when another arbitrary PSF was used for response, the deblurring was not successful. Multiple combinations of variables were tested and the deblurring only seemed to work when the original PSF was used as frequency response for the X=Y/H operation This shows that the Fourier transform method requires prior information of the distortion. This makes intuitive sense as the operation X=Y/H is an inherently a function with one input matched to an output. Changing the frequency response changes the behavior of the system.

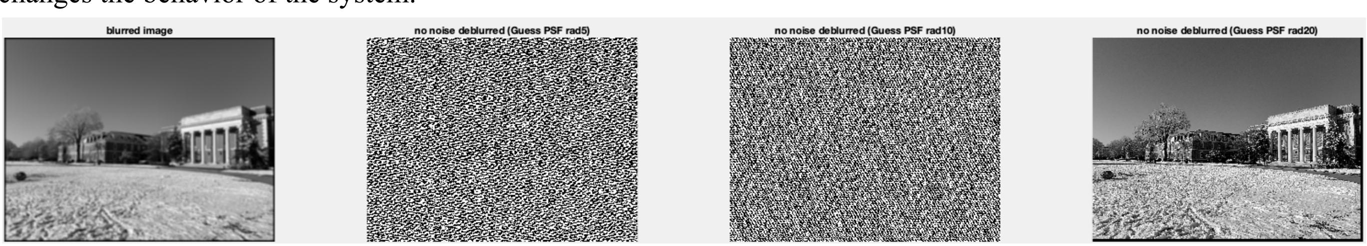


Figure 5: Raw Image (left) Compared with Deblurred Image (no noise) using Frequency Response of Circular Averaging Filter of Radius 5 (middle left), 10 (middle right), and 20 (right)

However, when noise was added, the Fourier transform blur reduction method does not seem to work anymore (Figure 6). Again, multiple combinations of variables were tested, yielding no successful result. The noise seemed to be enhanced by the operations, which was not expected as no operators or responses used seem to inherently magnify noise. Because of this, another method of noise reduction was implemented to try to bypass this problem of noise magnification.

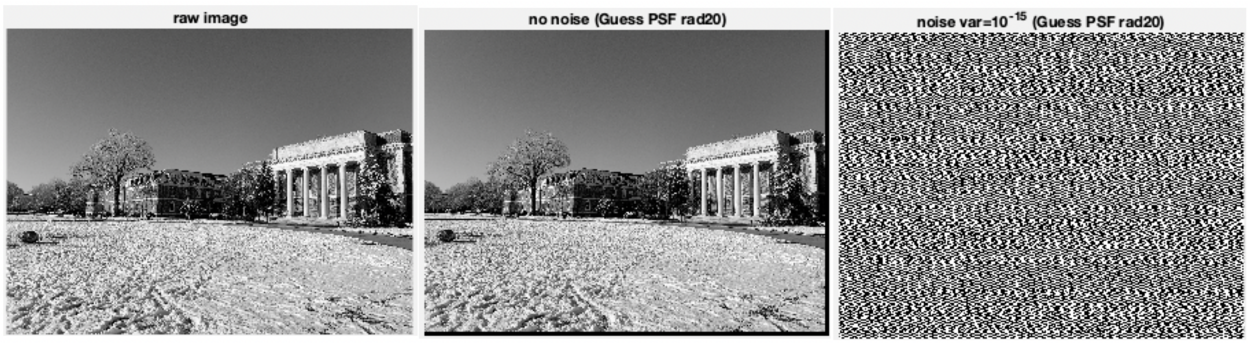


Figure 6: Raw Image (left) Compared with Deblurred Image using Frequency Response of Circular Averaging Filter of Radius 20 without noise (middle) and with noise (right)

Of the two noise reduction methods, averaging filter and median filter, the median filter (Figure 7) seems to work better. Median filter preserves edges because instead of taking averages, it replaces entries with the median of neighboring values. On the other hand, the averaging filter introduced dark hue, most likely due to the image being darker in general. The median filter seems to be able to remove noise well.

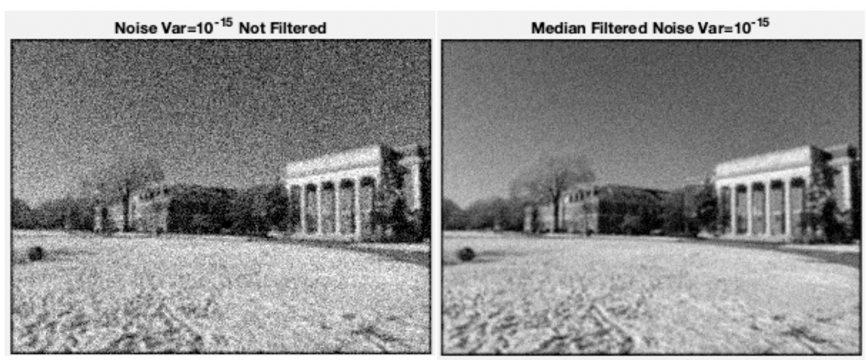


Figure 7: Raw Noisy Image (left) vs Median Filtered Image (right)

After removing noise, the noise-filtered image was again passed through the Fourier blur reduction method, but the deblurring was still not successful with noise magnification. It is conjectured that although noise reduction was able to filter out noise visibly, it does not filter out major players of noise magnification. Therefore, noise is still magnified, resulting in unsuccessful blur reduction.

Blur Reduction Through High Pass Filter

It was conjectured here that blur tends to be lower frequency signals. This assumption was made as blur tends to have smooth color shades. This smooth coloring would translate to a lower frequency. Therefore, if a high pass filter was carried out to remove low frequency signals, image deblurring may be accomplished. The high pass filter created in this project is visualized in Figure 8. The results of the high pass filter when applied does not seem to deblur the picture very well. Instead, the results of deblurring cannot be determined as the filter removed a significant amount of signal that it darkened the photo. Multiple sizes of high pass filters were tested, and the image seemed to get dark significantly at all sizes. This issue may have arisen as the low pass filter removed the DC-value, image mean, component of the frequency spectra. This image may have contributed significantly to the image brightness. Therefore, the high pass filter does not seem to be ideal. For the future, testing a bandstop filter may yield a better result.

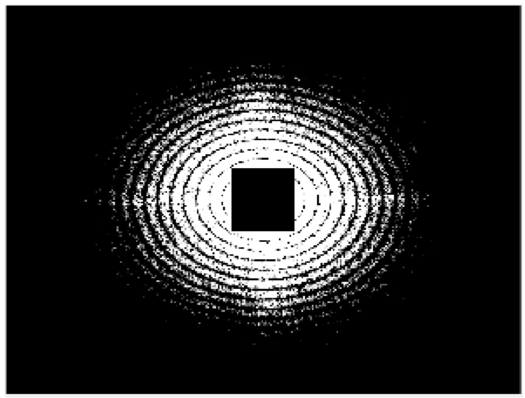


Figure 8: High Pass Filtered Frequency Spectra

Figure 9: Raw Image (left), Blurred Image (middle), and Deblurred Image Using High-Pass Filter (right)

Blur Reduction Through blindeconv of Image Processing Toolbox

Matlab has a function that is made specifically for blind deblurring. The function takes an input PSF as an initial guess and iterates to deblur the image. Not only is it able to deblur blurred images (Figure 10), it is also able to deblur blurred images with noise, which both methods above cannot accomplish (Figure 11). In addition, the deblurring effect seems to improve as iterations increase. This is the case as the function is able to provide a better estimate of the PSF and deblurred image after each iteration. To deblur a truly blind image, utilizing the blindeconv seems to be the best method. In addition, the success of the *blindeconv* function seems to suggest that the solution to the blind deblurring problem may have to involve some sort of optimization algorithm. This algorithm may have memory or other characteristics. Nevertheless, more involved algorithms and systems will be needed to truly deblur a blind image.

Figure 10: Blurred Image (left), and Deblurred Image Using Matlab Built-in Filter in 5 Iterations (middle), and 20 Iterations (right)

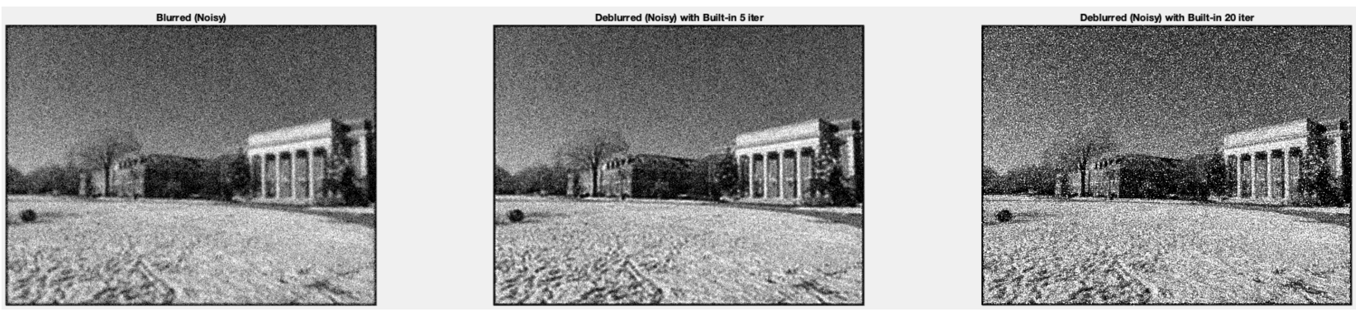
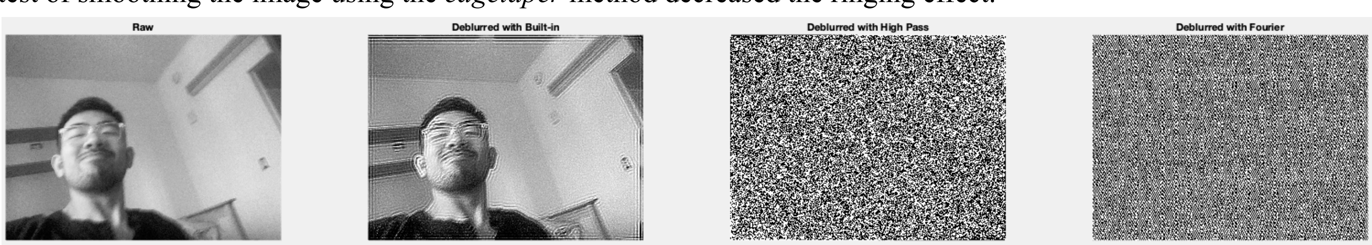


Figure 11: Blurred Noisy Image (left), and Deblurred Noisy Image Using Matlab Built-in Filter in 5 Iterations (middle), and 20 Iterations (right)

Blur Reduction of Total Blind Images

The final test of the project applies the three blind deblurring systems proposed on a true blind picture. As expected, only the built-in matlab function worked well in deblurring the system. The Fourier transform method likely did not work as the guess frequency response was not the actual frequency response. Despite multiple changes to the threshold, the high pass filter method also did not seem to yield a response, which was not expected. After graphing the frequency spectra, it was evident that frequency components were spread throughout the graph, which would mean any high pass filter would not significantly change the blurring of the image. Another interesting finding was the appearance of edges. This effect is known as ringing and is caused by high frequency drop-off at edges of an image. A quick test of smoothing the image using the *edgetaper* method decreased the ringing effect.



Conclusion

The goal of the project was to determine if blind deblurring was possible. Of the three methods tested, only the built-in method of the Image Processing Toolbox was successful, while the other two proposed methods showed limited success. The results of this project show that the blind deblurring problem may require an additional optimization algorithm to truly deblur a blind image. Additional methodologies such as utilizing generative adversarial networks (GANs) may also be indispensable.