Final Exam Report

This final exam is an interesting and truly open project that we would get in the real world. With only a system and the degraded image, we are tasked with restoring the degraded image to its true image.

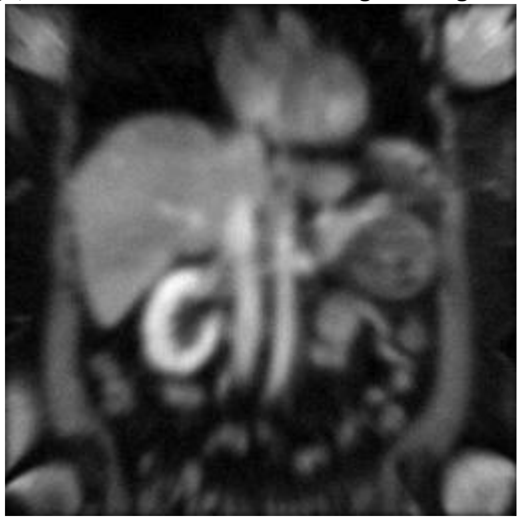

Figure 1: The given degraded image.

Starting in such a situation requires us to gather as much information about the system as possible before trying to restore the given image. To begin the analysis, I passed a grid of delta functions through the system.

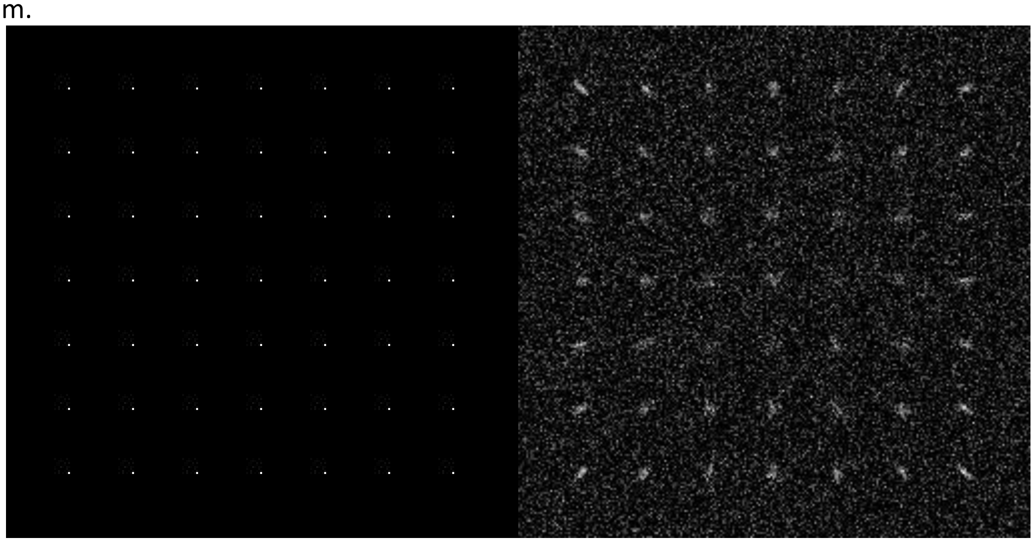

Figure 2: Left is input grid, right is output grid.

From this test, we can conclude that the system adds noise to the image. More importantly from this test we can conclude the system is shift variant. The delta functions in the corners are the least attenuated of the delta functions. The ones along the axes are more similar to the noise floor than the corner delta functions. Just from visual inspection, the delta functions in the corners are less attenuated and also more smeared along the diagonal than the middle delta functions. To double check this result, I increased the size of the delta functions to squares.

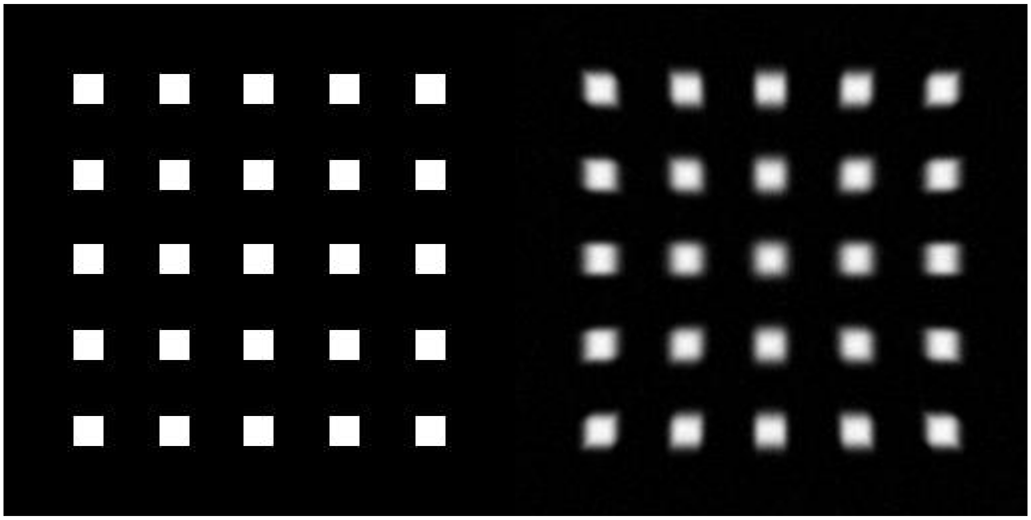

Figure 3: Left is input grid, right is output grid

The output of Figure 3 illustrates that the middle square appears to just be blurred while the top right square smears the square along the x=y line, whereas the right most square on the x-axis is smeared in only the x-direction. Thus, we can firmly conclude that the system of shift variant.

Next, we conducted the test for linearity based on the linearity principle.

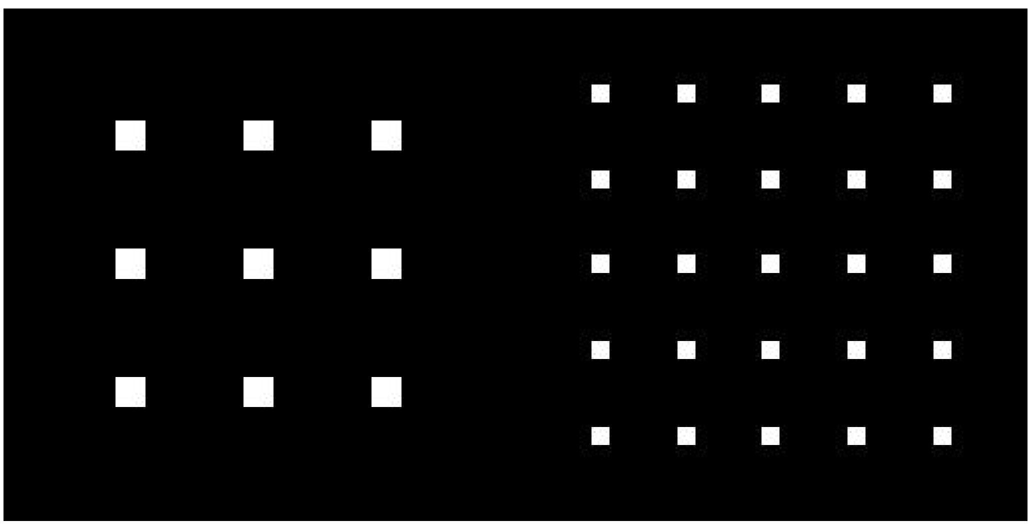

Figure 4: The left is the first input image, right is second input image.

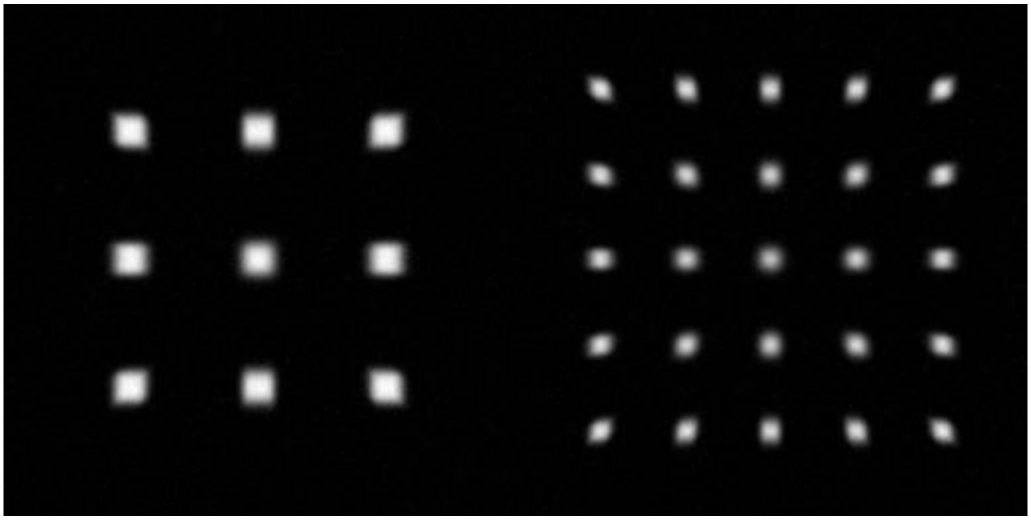

Figure 5: These are the corresponding outputs to Figure 4 images.

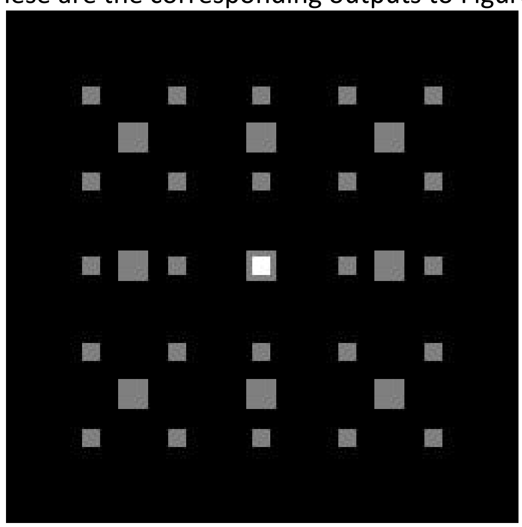

Figure 6: Input images combined.

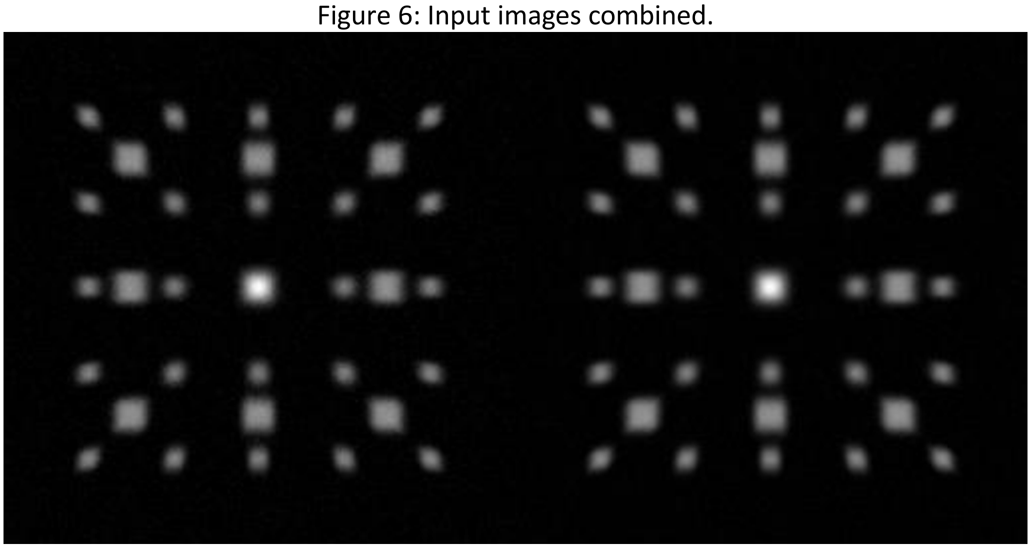

Figure 7: Left is the output of Figure 6 and the right is the combination of the images from Figure 5.

Visual inspection, it would seem that the system is linear since both images in Figure 7 appear the same. I also performed a quantitative analysis to confirm my hypothesis. The two images were not a perfect match, number for number, because there is always some error in calculations, of course. I took the difference in the images of Figure 7 and summed the image. That sum is 21146 in this case. From Programming Assignment 3, we found the highest error was 31 and that system was revealed to be linear. It would at first seem that this system in nonlinear, however the images from both assignments have different intensity scales. To normalize my measurement, I divided this total error by the total sum of maximum possible intensities of two image sets. The resulting error rates were on the order of 10^-4 meaning the error rates were the same. Since the PA3 system was linear, I can quantitatively conclude that this system is also linear.

To further characterize this system we could get the FWHM of different PSFs at different locations, but to do this we need to remove the noise from the image. To do this, we next must characterize the noise.

I passed a constant image of non-zero intensity through the system 10 times, then found the mean and variance images.

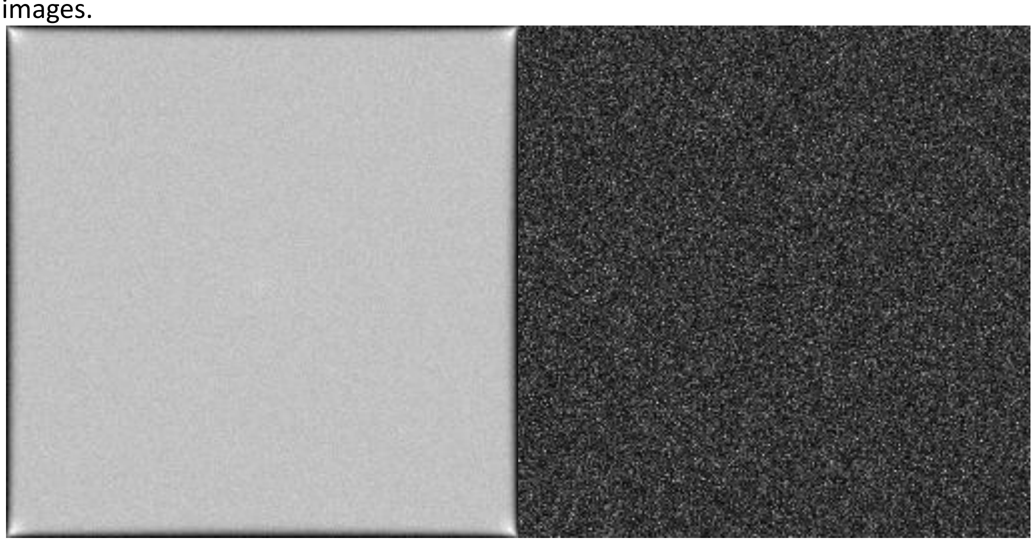

Figure 8: Left is the mean image and right is the variance image.

The mean and variance images maintain a constant structure throughout the images in Figure 8. Thus, by visual inspection, I can confidently conclude that this system is stationary. Also, I found the spatial mean to be about the same mean as the intensity of the mean image in Figure 8. This means that the noise is also ergodic.

The next step was then to find the distribution of the noise. I, again, passed a non-zero constant intensity image through the system to get Figure 9.

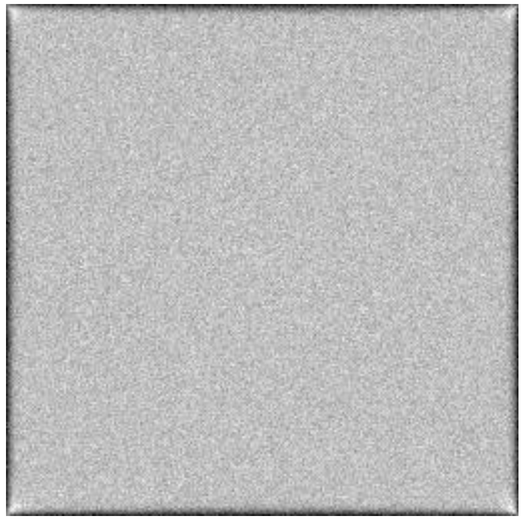

Figure 9: An output image of an input of constant intensity.

The constant intensity of the input image was subtracted from this image to get an image of just the noise. A histogram of the values was determined as shown in Figure 10.

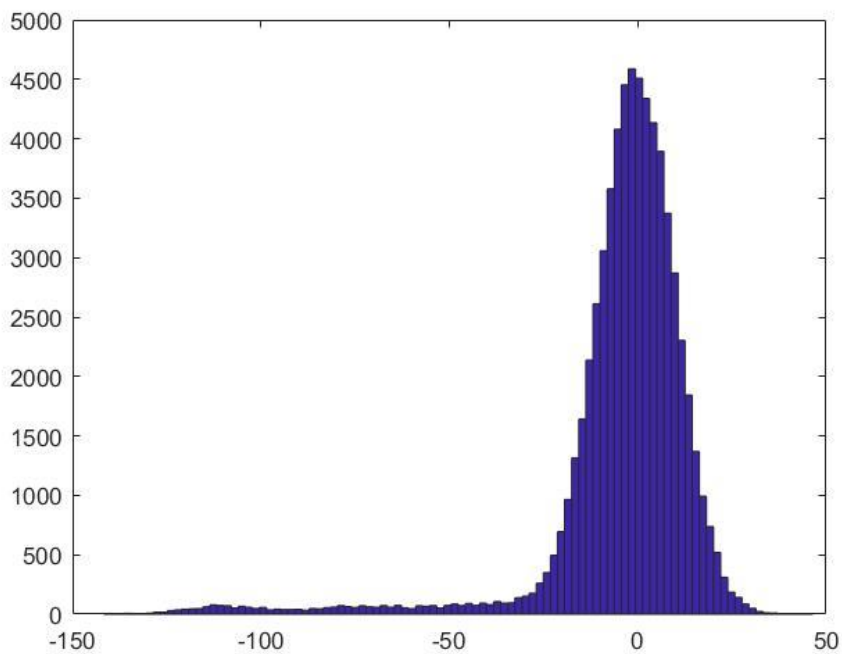

Figure 10: The PDF of the noise image.

In addition to the PDF in Figure 10, I also found the spatial mean and spatial variance to be -4 and 392.05, respectively. We can see that the distribution has a lower end tail that is not reflected on the upper end. As a result, we can conclude this noise is Poisson distributed.

We can further characterize the noise by calculating the autocorrelation and corresponding PSD.

Figure 11: Left is the autocorrelation and right is the corresponding PSD.

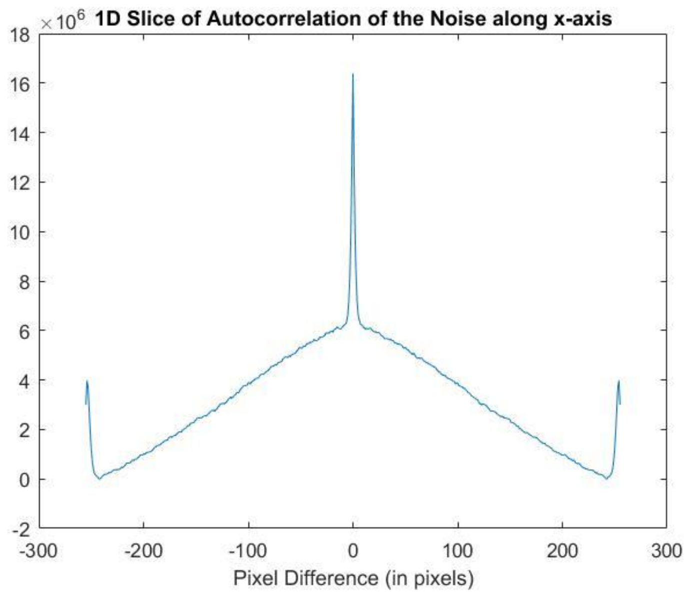

Figure 12: A 1D slice of Rnn along the x-axis

As we see in Figures 11 and 12, the autocorrelation function does not appear to be a delta function centered in the middle. We actually see that the image is most correlated in the x and y directions. So, Point A would be more correlated with Point B 100 pixels to the right than Point C which is 10 pixels in the diagonal direction. We can conclude that this noise is correlated, but it is correlated in a weird way.

Thus far, we've learned that system 4 is linear and shift variant with stationary, correlated, Poisson-distributed noise; this is not the easiest system to restore. Based on the description provided with the exam, it seems as though we will have to utilize image restoration tools from the end of the course to solve this problem. As seen in the slides, the first step of restoration is choosing a criterion.

First, there is MMSE. However, we would need Rff, H, and MMSE requires that n is zero-mean and uncorrelated. Second, there is LS. We would only need H and the estimate is unbiased. Third, there is ML. It is unbiased and has the smallest error the of unbiased estimators. We would only need H to get the estimate. Note, ML is WLS if n is normal, and WLS is LS if n is stationary and uncorrelated. Weights are important if n is correlated. Lastly, there is MAP which requires H, a mean image, and beta.

All of the estimators need H. Unfortunately, we cannot get H exactly due to the shift variance. However, we can approximate H by dividing the image into sub-images, then assume shift invariance inside the sub-image so that we can use the PSF to get the H for that sub-image. Another thing to note is that many of the estimators prefer zero-mean, Gaussian noise. We do not have exactly that, however, we can model the system to be just that seeing that the Poisson distribution in Figure 10 takes a similar shape to a Gaussian distribution. With these assumptions, we can effectively make it easier for us by treating this as a linear and shift-invariant (sub-images) system, with approximately zero-mean, Gaussian, correlated noise. Unfortunately, this still requires us to address the correlated noise issue.

Of the estimators listed above, I am confident MMSE is not likely because it requires Rff and uncorrelated noise. Next, we have the family of ML, WLS, and LS. Since we have correlated noise, it is reasonable to guess that the solution would not be LS. I believe WLS would be a possibility since it is stated that the weights are important if n is correlated. However, it seems I would have to change the Matlab code in least_squares_iter to accept correlated noise, which seems like a stretch in terms of solutions. This also goes for ML as a solution, since we could only use least_squares_iter on certain assumptions. The only way to use this family for a solution would be to remove the noise completely, then apply the iterative solution.

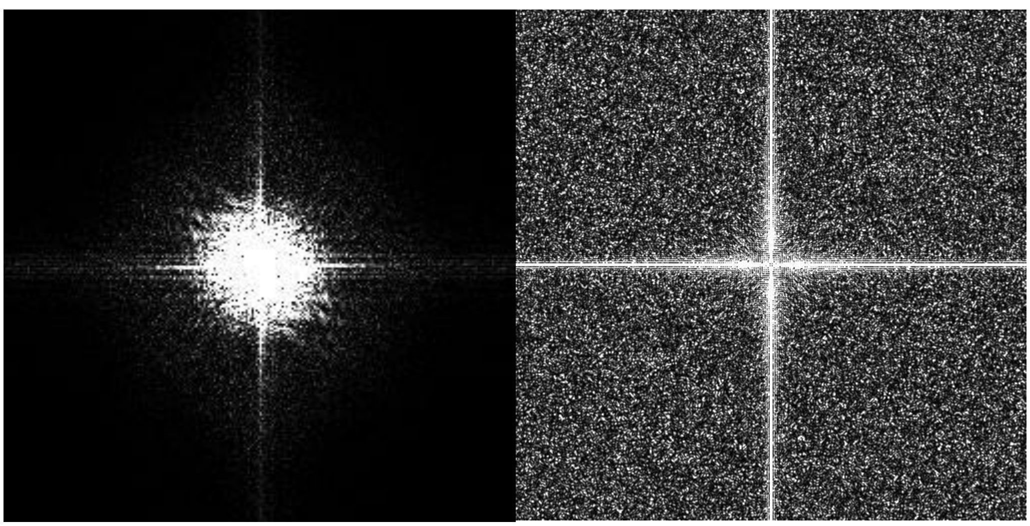

Figure 13: Left is the degrade image PSD and the right is the noise PSD.

As seen in Figure 13, it may be possible to reduce the noise somewhat. The + pattern on the right can be seen in the degraded image's PSD. We could try to filter out the noise by removing frequencies along the x and y axis that lie outside of the low frequency circle. This would reduce some, but not all, noise in the degraded image. Thus, our first possible solution would be to assume zero-mean, Gaussian noise, followed by tiled sub-imaging to assume shift invariance, followed by filtering of noise, followed by least_squares_iter of the approximately noise-free sub-image. This solution is sketchy, seeing that there still exists correlated noise in the image, but if we were to ever use ML/WLS/LS it would be most likely through this way.

I would like to note quickly that there is a slide in the notes that states we can use the KL transform to decorrelate noise, but I do not believe there is anything in the provided examples that goes through applying the KL transform in a situation such as this. I am sure it is possible, but for the scope of this final I doubt it is the intended path. It is just something to note, as I am trying to gauge which path would result in the best solution.

Lastly, there is the option of a MAP estimator. We saw in PA8 that the MAP can take any form between the ML estimator and the maximum of the prior, which in the case of map_norm_iter is a Gaussian prior. The MAP from PA8 did a good job of not blowing up noise in the image when compared with the ML estimator. Thus, based on all of the discussion presented here, I believe the best course would be to assume zero-mean, Gaussian noise, followed by tiled sub-imaging to assume shift invariance, followed by filtering of higher frequency noise, followed by map_norm_iter of the sub-image.

In this case, we must define the filtering approach and the parameters for map_norm_iter (start image, H, number of iterations, mean image, and beta). From PA8, we found the start image to be the same as the measured image, mean image to be a constant image valued at the mean of the measured image, and beta to be best at 0.1. These may not hold up for this problem, but it is a good place to start. Next, we will be able to pass a point source through the system to get H. Finally, we can settle on small iterations to begin so the solution doesn't take forever.

The filtering approach will be to extract the 1D x and y axes from the PSD of the degraded image, model and apply a butterworth filter, then replacing the old vectors with the new ones to get suppression. To get the cutoff, I looked at the PSD of the degraded image.

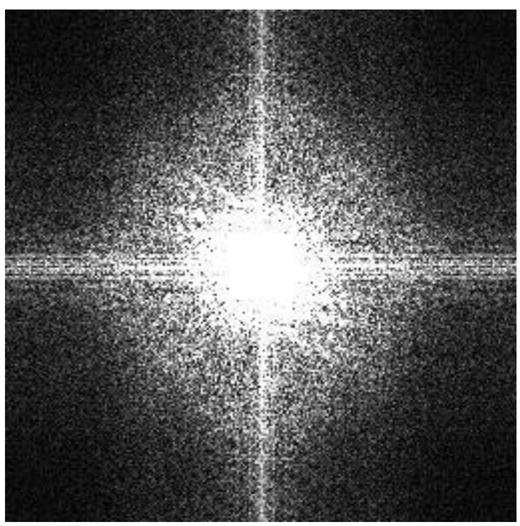

Figure 14: Degraded image PSD

I lowered the upper range of the PSD from the left of Figure 13 in order to see more of the PSD. It turns out that the center white circle does not get any larger, so it appears this is where most of the true image content lies. We can also see that the noise PSD begins to prevail past that circle. So I made the cutoff to be the radius of that circle, which is 37 cycles per frame from the DC. Also, the lines of the cross are about 10 cycles per frame wide, so I will need to extract more frequencies around the x and y-axis vectors.

It was discovered along the way that filtering the noise would be easiest on the full-scale image instead of the sub-images because the pattern shown in Figure 14 does not appear in even the largest sub-image's (64x64) PSD.

Figure 15: Specially designed filter to remove noisy frequencies.

To filter the image, I designed a custom notch filter that removed parts of the cross that were not in the circle. I also applied a gaussian filter to smooth the binary filter so there would be no ringing effects from the filter. The result of applying the filter in Figure 15 to the FFT of Figure 14 can be seen in the next figure.

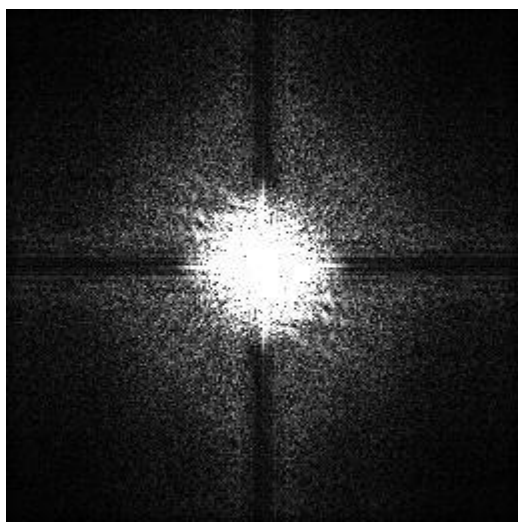

Figure 16: The filtered degraded image's FFT.

The inverse was taken to get the filtered degraded image, which looked extremely similar to the original image. Completing the algorithm, I partitioned the filtered image into sub-images and performed MAP restoration on each one. The measured image and the start image were the sub-image. The mean image was a constant sub-image with mean equal to the mean of the entire filtered image. I used a beta of 0.1 (from PA8) and 10 iterations. The H matrix was fabricated from the sub-image's H by making every column of the H matrix to be the sub-image's H. Of course, this is on the assumption that the sub-image is shift invariant. The result came to be Figure 17.

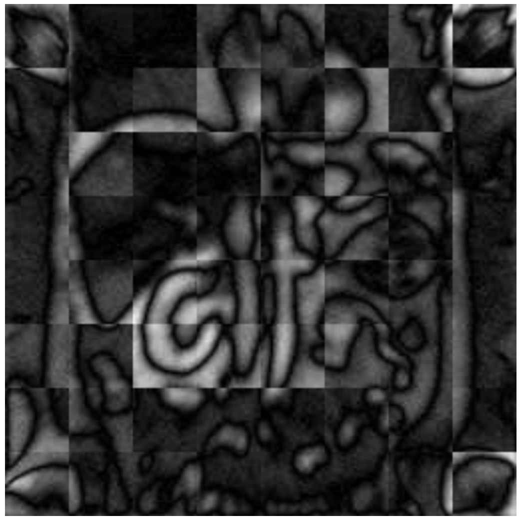

Figure 17: Final image output.

The resulting image is disappointing, however I believe the reasoning for each step getting here is solid. Unfortunately, I will have to leave this image as it is due to time constraints, however I would like to give some ideas of what went wrong and how this can be fixed.

The first issue that comes to mind is that maybe the assumption of shift-invariance in the sub-images is too much of a stretch. I could create the H matrix by finding every PSF in the image, however this would be time consuming. Another thought is maybe I could have used individual point source images instead of a grid since this would be a better characteristic of how a point source is spread out. One solution may be to stitch the images together and color correct based on some reasoning.

Whatever the fault may be, I am pleased with the unique solution I came up with to tackle a difficult real-world issue. This final is a testament to how hard image processing can be. Hopefully with time and experience algorithm will come more naturally to me so that I can more elegantly combine the tools of this field.