Deblurring and Enhancing Astrophotographs Using PSF Based Digital Image Processing

Group Number: 19

Weekly Report 3

Group Members:

- Hetanshu Parmar
- Purvish Parekh
- Tarjani Patel
- Vrushti Patel
- Ansh Rathva

1 Introduction

In order to enhance astrophotography and minimize the effects the blurring, we applied deconvolution and Wiener filter to obtain images with partially reversed effects of blurring. However, the problem of deblurring still persists and affects the visibility of the image. Also, we tried estimating different PSFs to find the best fit PSF and ultimately decided to apply Gaussian PSF. Our main focus is to retain the fainter features of the image, making it easier to distinguish them. So, we aim to solve this problem while also keeping the deblurring techniques in hindsight.

2 Methodology

During the third week, we focused on utilizing more raw data to achieve clearer and more precise output images. We also found out, according to the guidance provided by the faculty, about where the Wiener Filter can be applied and where it can't be applied. We found that the Wiener Filter can only be applied to stationary or linearly motion-blurred images. And, it doesn't work for non-static and non-linearly motion-blurred images. In our case, we know that the noise we aim to remove is static Gaussian PSF noise. Therefore, we can definitely use the Wiener Filter to our advantage for removing high-frequency noises.

Working with more raw images, we found out that the processed image is much darker, and fainter details are missing in the output image. Here, in the given images, we can see the difference in input and output images by directly processing the raw images.

Therefore, to enhance the output image, we applied contrast stretching to the image, highlighting the fainter details. Simply put, we discarded the darkest 2% and brightest 2% pixels, and mapped the remaining pixels with full intensity

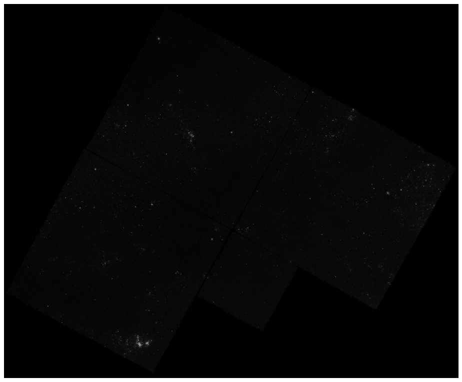

Figure 1: Raw Input Image

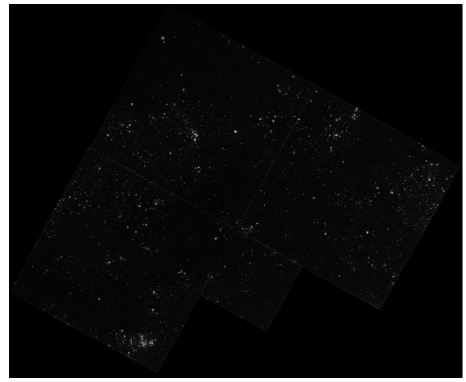

Figure 2: Directly Processed Output Image

range. Below is the code for the entire pipeline, which includes loading the image, applying contrast-stretching, the Wiener filter, and Richardson-Lucy Deconvolution.

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
from skimage import img_as_float, restoration, exposure
from numpy.fft import fft2, ifft2
from astropy.io import fits

# --- Load Image ---
with fits.open("/content/Hst3-6.fits") as hdul:
    img_data = hdul[0].data

astro = img_as_float(img_data)

# --- Define PSF (example: Gaussian 7x7) ---
def gaussian_psf(size=7, sigma=1.2):
    ax = np.linspace(-(size-1)/2, (size-1)/2, size)
    xx, yy = np.meshgrid(ax, ax)
    psf = np.exp(-(xx*2 + yy2) / (2*sigma*2))
    psf /= psf.sum()
    return psf

psf = gaussian_psf(size=7, sigma=1.2)

# --- Wiener filter ---
def wiener_filter(img, kernel, K):
    kernel /= np.sum(kernel)
    dummy = fft2(img)
```

```
    kernel_fft = fft2(kernel, s=img.shape)
    wiener = np.conj(kernel_fft) / (np.abs(kernel_fft) ** 2 + K)
    result = np.abs(ifft2(dummy * wiener))
    return result

astro_wiener = wiener_filter(astro, psf, K=0.0001)

# --- Contrast stretching to enhance faint regions ---
p2, p98 = np.percentile(astro_wiener, (2, 98))  # discard outliers
astro_stretched = exposure.rescale_intensity(astro_wiener, in_range=(p2, p98))

# --- Richardson-Lucy Deconvolution ---
deconv_rl = restoration.richardson_lucy(astro_stretched, psf, num_iter=30, clip=True)

# --- Optional: contrast stretch the final output again ---
p2, p98 = np.percentile(deconv_rl, (2, 98))
deconv_rl_stretched = exposure.rescale_intensity(deconv_rl, in_range=(p2, p98))

# --- Display Results ---
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.imshow(astro, cmap='gray')
plt.title("Original HST Image")
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(astro_stretched, cmap='gray')
plt.title("After Wiener + Contrast Stretch")
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(deconv_rl_stretched, cmap='gray')
plt.title("After RL Deconvolution + Stretch")
plt.axis('off')

plt.tight_layout()
plt.show()

# --- Save outputs ---
cv2.imwrite("Original.png", (astro * 255).astype(np.uint8))
cv2.imwrite("Wiener_Contrast.png", (astro_stretched * 255).astype(np.uint8))
cv2.imwrite("Wiener_RL_Contrast.png", (deconv_rl_stretched * 255).astype(np.uint8))
```

3 Results

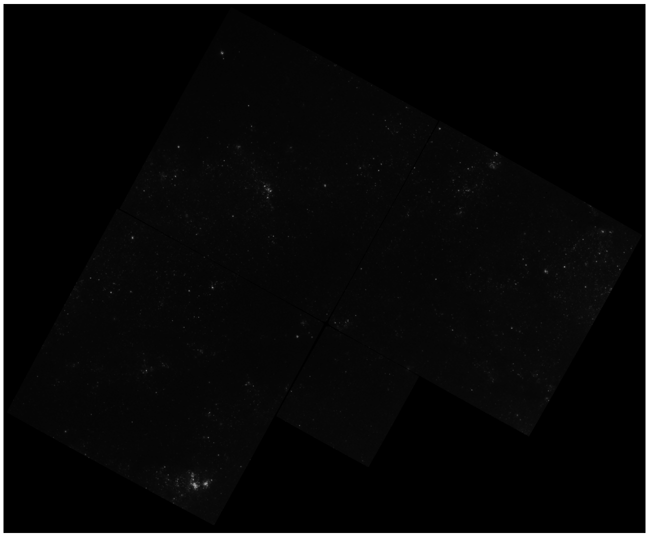

Figure 3: Raw Input Image

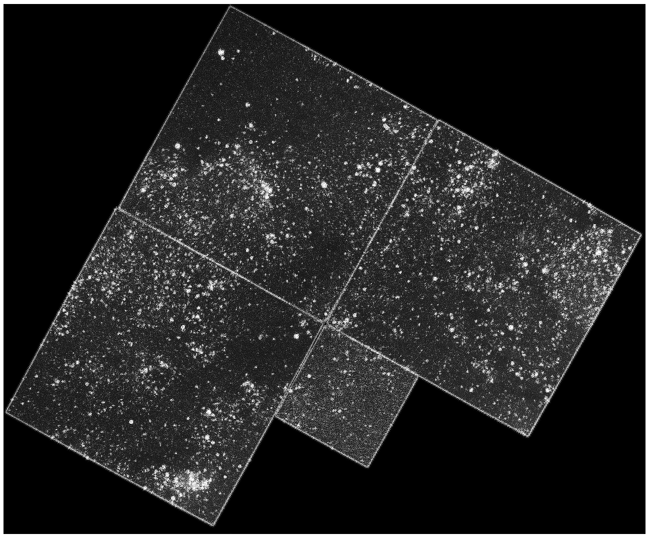

Figure 4: Output Image

4 Problems Faced:

- The problem of partial blurring persists.
- Since the output obtained was too dark to distinguish the features of the image, we had to apply contrast stretching to enhance the faint regions. We process the output image again by contrast stretching for clarity. But it is still difficult to clearly interpret all the features of the image.
- Also, we have not used any color mapping techniques to extend this solution to the colored images. The same method is less likely to work, since color mapping would also require aligning and stacking of the images, and modelling PSFs to get clarity.
- Since the size of the true FITS image captured by HST is too large, it imposes a challenge on the processing of a larger dataset.

5 Next Set of Work:

In the upcoming week, we plan to:

- Apply stacking since it reduced noise, and models a near true PSF.
- Stacking also gives a better signal to noise ratio
- Astrophotographs are often captured with a certain filter that corresponds to a certain wavelength. So, we will align and stack the R,G and B images to obtain a visible colored image that retains/holds the complete information captured by the system.
- Color mapping will help to solve the challenge of lesser visibility of fainter features.
- Apply this techniques to a larger dataset.

6 Conclusion

For this week, we aimed at removing the Gaussian PSF noise by using the Wiener filters. We also tried and tested different processing techniques to increase the quality of the obtained image. We applied contrast stretching to solve the problem of lesser visibility of the fainter features. Next, we will attempt to implement stacking of the images to model a true PSF which will provide a better quality of the deblurred image.

References

[1] E. J. Groth, “A pattern recognition program for cosmic ray rejection on CCD images,” *The Astronomical Journal*, vol. 118, no. 4, pp. 1764–1776, 2006. [Online]. Available: https://iopscience.iop.org/article/10.1086/510117/pdf)

[2] Mikulski Archive for Space Telescopes (MAST), “MAST Portal.” [Online]. Available: https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html