Denoising Astronomical Images Using the Fourier Transform

Henry Filberto Shenelo - 13523108[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523108@std.stei.itb.ac.id, [2]henryfilbertoshenelo@gmail.com

***Abstract*— Astronomical images are invaluable for understanding celestial phenomena but are often corrupted by noise from atmospheric, sensor, and optical distortions, hindering effective analysis. To address this, denoising techniques using the Fourier Transform provide a robust solution. By transforming images into the frequency domain, high-frequency noise can be isolated and filtered, enhancing image clarity while preserving essential details. This study explores Fourier Transform-based denoising for astronomical images, utilizing the Fast Fourier Transform (FFT) for computational efficiency. Results demonstrate the effectiveness of noise removal, with varying frequency retention levels influencing the balance between noise reduction and image sharpness. The role of phase in image reconstruction is also emphasized, revealing its importance in maintaining structural integrity. These findings underscore the potential of Fourier Transform techniques in improving the quality of astronomical data for further scientific investigation.**

***Keywords*— Astronomical image, denoising, Fourier Transform, frequency domain.**

I. Introduction

Astronomical imaging plays a crucial role in understanding the universe, providing invaluable data for studying celestial bodies and phenomena. However, these images are often marred by noise due to various factors, including atmospheric turbulence, sensor limitations, and optical distortions. Such impairments can obscure critical details, complicating the extraction of meaningful insights from the data.

To address these challenges, denoising technique has become essential tools in astrophysics and related fields. Among the various methods available, the Fourier Transform offers a powerful mathematical framework for analysing and processing images in the frequency domain. By decomposing an image into its frequency components, the Fourier Transform enables targeted manipulation of noise, facilitating the recovery of high-quality images with enhanced clarity and detail.

This paper explores the application of Fourier Transform-based techniques for denoising astronomical images. We focus on leveraging its ability to isolate high-frequency noise, presenting a systematic approach to restore the fidelity of astronomical data. Additionally, we evaluate the effectiveness of these techniques through visual and quantitative comparisons, highlighting their potential to improve the accuracy of subsequent scientific analysis.

This paper is organized as follows: the introduction is followed by a theoretical basis, which explains the principles of complex number and the Fourier Transform. Next, the implementation section details the practical application of these principles, while the results section presents and analyzes the outcomes. Finally, the conclusion and recommendations section summarize the findings and provides suggestions for future improvements in astronomical image restoration.

II. Theoretical Basis

A. *Complex Number*

A complex number is composed of two parts, real and imaginary parts. It can be written as

$$z = x + iy$$

where x and y are real and i defined as

$$i^2 = -1.$$

Complex number can be written in the polar form as

$$z = r(\cos\theta + i\sin\theta)$$

where

$$r = \sqrt{x^2 + y^2} \text{ and } \theta = \tan^{-1}\frac{y}{x}.$$

Euler's formula also shows that

$$e^{i\theta} = \cos\theta + \sin\theta.$$

We can interpret Euler's formula geometrically where $e^{i\theta}$ is a rotor that rotates z to z' by an angle θ counterclockwise.

Why is this important to our discussion? Fourier transform that we are going to use decomposes wave into a sum of sinusoidal (sine and cosine) waves, so instead of calculating them separately, we can write it in the complex form of

$$Ae^{i(2\pi\nu t+\varphi)}.$$

Writing it in complex form has nice properties when dealing with summation of sinusoidal waves. Summation can be done without any trigonometry identities.

$$\varphi(t) = \varphi_1(t) + \varphi_2(t) = A_2e^{i(2\pi\nu t+\varphi_1)} + A_2e^{i(2\pi\nu t+\varphi_2)}.$$

Complex form also preserves the information for both the magnitude and phase of wave. This is very important in image reconstruction as taking only the magnitude (real part) will make the image unrecognizable.

B. *Fourier Transform*

The Fourier transform is a mathematical operation that converts data from the time or spatial domain into the frequency domain. It is defined as

$$Y(\omega) = \int_{-\infty}^{+\infty} y(t) e^{-i\omega t} dt$$

We can also write it as

$$Y(\omega) = \int_{-\infty}^{+\infty} y(t)\cos(\omega t)\, dt \; - \; i\int_{-\infty}^{+\infty} y(t)\sin(\omega t)\, dt$$

As can be seen, it is more concise and easier to compute when written as a complex form. Therefore, we will use the complex form.

When performing a Fourier Transform, a signal or image is represented in terms of its frequency components, which consist of two key elements: magnitude and phase. These components are derived from the complex representation of the transform and are essential for fully reconstructing the original data.

The magnitude of a Fourier Transform represents the amplitude or intensity of each frequency component in the signal. In terms of images, the magnitude describes how much a particular frequency contributes to the image. It is calculated as:

$$|F(u,v)| = \sqrt{Re(F(u,v))^2 + Im(F(u,v))^2}$$

The magnitude is what we usually visualize when creating a frequency spectrum, where low frequencies are near the center and higher frequencies radiate outward.

The phase of a Fourier Transform encodes the spatial arrangement of frequency components, essentially determining where the structures or patterns appear in the image. The phase is given by:

$$\varphi(u,v) = \tan^{-1}\left(\frac{Im(F(u,v))}{Re(F(u,v))}\right)$$

While the magnitude tells us "how much" of each frequency exists, the phase tells us "where" these frequencies are located in the image. This makes the phase critical for reconstructing the structural details of the image. Without the phase, the reconstructed image would lose all recognizable features, appearing as an abstract blur.

C. *Discrete Fourier Transform (DFT)*

For applications in image processing, the Discrete Fourier Transform (DFT) is commonly used because digital images are represented as discrete pixel values. The DFT transforms a discrete spatial representation of an image into its frequency domain, making it suitable for computational manipulation. The DFT and its inverse are defined as

$$Y_j = \sum_{k=0}^{n-1} y_k e^{-2\pi ijk/n}$$

$$y_k = \frac{1}{n}\sum_{k=0}^{n-1} y_k e^{+2\pi ijk/n}$$

However, doing this calculation directly is expensive since it requires n^2 calculations. To simplify the computation, the Fast Fourier Transform (FFT) is often used. FFT is an efficient algorithm to compute the DFT. It is widely used in image processing (and in everything) due to its computational efficiency, requiring only $n \log_2 n$. One of the algorithms to perform FFT is Cooley-Tukey algorithm. In short, the idea is to split DFT into its even and odd terms. We can write this as

$$Y_j^{k\ even} + W^k Y_j^{k\ odd}$$

where W = $e^{-2\pi i/n}$.
We can do this recursively until we get a single point.

Since we are dealing with 2D Images, the DFT can be extended into the 2D plane. The formula is given by

$$F[k,l] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

and the inverse of it is

$$f[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n] e^{+j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

The Cooley-Tukey FFT can be applied as well by doing it separately for each dimension (rows and columns).

D. *Noise*

Noise in image processing refers to unwanted variations in pixel values that distort the true signal in an image. Noise can degrade the quality of an image, making it harder to extract meaningful information. Some types of noise commonly found are:

1. Gaussian noise, characterized by random pixel intensity variations following a normal distribution.
2. Salt-and-Pepper noise, appears as random white (salt) or black (pepper) pixels.
3. Poisson noise, arises from the quantum nature of light and affects low-light or high-sensitivity imaging systems.

Noise can originate form various sources in astronomical images such as

1. Sensor noise
2. Photon shot noise
3. Background light
4. Atmospheric turbulence.

E. *Low-Pass Filtering*

A low-pass filter allows low-frequency components of a signal to pass through while removing higher-frequency components. The cutoff frequency determines the boundary between the low and high frequencies. Frequencies lower than the cutoff are preserved, while those higher are suppressed.

For this implementation, we are using Ideal Low-Pass Filter. This type of filter is characterized by a sharp cutoff, where all frequencies higher than a certain threshold are completely removed, and all frequencies lower than this threshold are retained. We can write this as

$$F'(u,v) = \begin{cases} F(u,v), if\ u,v \in low-frequency\ range \\ 0, otherwise. \end{cases}$$

III. IMPLEMENTATION

A. *Image Denoising*

The following codes are modified from https://scipy-lectures.org/intro/scipy/auto_examples/solutions/plot_fft_image_denoise.html#read-and-plot-the-image

To denoise the image, we first converted it to grayscale.

```
im = plt.imread('moonlanding.jpg')
im = np.dot(im[..., :3], [0.2989, 0.5870, 0.1140])
plt.figure()
plt.imshow(im, plt.cm.gray)
plt.title('Original Grayscale Image')
```

Next, we perform a Fourier Transform on the grayscale image to convert it to the frequency domain. Figure 1 displays the Fourier Transform spectrum.

```
im_fft = fftpack.fft2(im)

def plot_spectrum(im_fft):
    from matplotlib.colors import LogNorm
    # A logarithmic colormap
    plt.imshow(np.abs(im_fft),
norm=LogNorm(vmin=5))
    plt.colorbar()

plt.figure()
plot_spectrum(im_fft)
plt.title('Fourier transform')
```

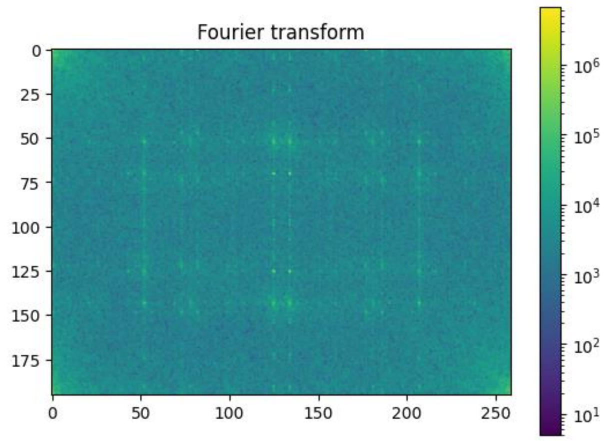

Figure 1. Fourier Transform Spectrum Before Denoising

Then, we filter the noise by zeroing out the high-frequency components. We keep only *k* of the lowest frequencies in both the rows and columns. Figure 2 displays the filtered spectrum.

```
keep_fraction = 0.1

im_fft2 = im_fft.copy()
r, c = im_fft2.shape

# Set to zero all rows with indices between
#r*keep_fraction and r*(1-keep_fraction):

im_fft2[int(r*keep_fraction):int(r*(1-
keep_fraction))] = 0

# Similarly with the columns:
im_fft2[:, int(c*keep_fraction):int(c*(1-
keep_fraction))] = 0

plt.figure()
plot_spectrum(im_fft2)
plt.title('Filtered Spectrum')
```

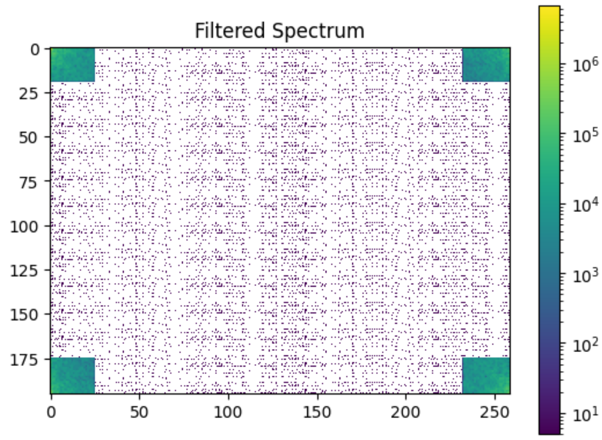

Figure 2. Fourier Transform Spectrum After Denoising

Finally, we reconstruct the image using Inverse Fast Fourier Transform, only displaying the real part.

```
im_new = fftpack.ifft2(im_fft2).real
plt.imsave('moonlanding_denoise.jpg', im_new,
cmap='gray')
```

To quantitatively compare the original image and the denoised image, we can evaluate the noise level in each image with the following code.

Source: J. Immerkær, “Fast Noise Variance Estimation”, Computer Vision and Image Understanding, Vol. 64, No. 2, pp. 300-302, Sep. 1996

```
def estimate_noise(I):
    H, W = I.shape
    M = [[1, -2, 1],
         [-2, 4, -2],
         [1, -2, 1]]
    sigma = np.sum(np.sum(np.absolute(convolve2d(I,
M))))
    sigma = sigma * math.sqrt(0.5 * math.pi) / (6 *
(W-2) * (H-2))
    return sigma
```

By isolating and reconstructing the images using only their phase or magnitude components from the Fourier Transform, we can also demonstrate the critical role of phase in preserving structural details during image restoration.

```
magnitude = np.abs(im_fft)
im_magnitude = fftpack.ifft2(magnitude).real
plt.figure()
plt.imshow(im_magnitude, plt.cm.gray)
plt.title('Image using only Magnitude')
phase = np.angle(im_fft)
average_magnitude = np.mean(magnitude)
im_phase = fftpack.ifft2(average_magnitude *
np.exp(1j * phase)).real
plt.figure()
plt.imshow(im_phase, plt.cm.gray)
plt.title('Image using only Phase with Average
Magnitude')
plt.show()
```

IV. RESULT

Based on the implementation, we use the image of the moon landing that contains salt-and-pepper noise, characterized by random black and white pixels distributed throughout the image. This type of noise is possibly caused by transmission errors or corruption during storage or retrieval. To denoise the image, low-pass filtering is applied with varying values of k, which represents the cutoff frequency. For each value of k, the noise level is estimated to evaluate its effect on the filtering process. The results display a comparison between the original noisy image and the denoised images. The original image exhibits significant visible noise, which hinders analysis by obscuring key details. In contrast, the denoised images show improvements in clarity, as high-frequency noise is effectively filtered out. However, as the parameter k decreases, the noise is further reduced, enhancing the overall smoothness of the image. Unfortunately, this comes at the cost of losing essential high-frequency details, causing the image to appear increasingly blurred. The trade-off between noise reduction and image sharpness emphasizes the importance of selecting an optimal value of k to balance minimizing noise while preserving important image details.

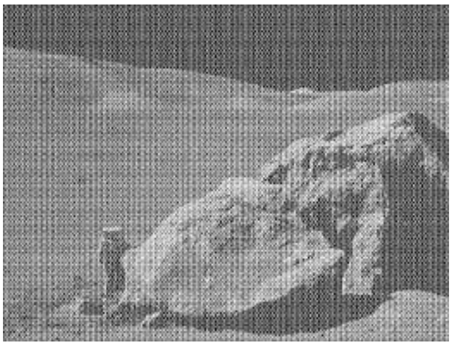

Figure 3. Image Before Denoising

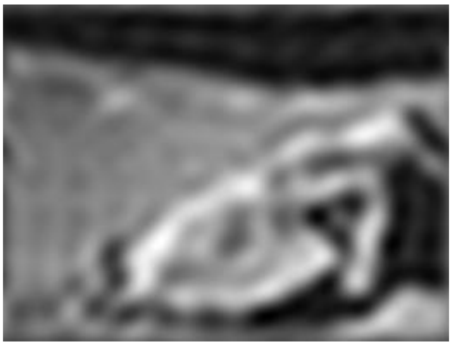

Figure 4. Image After Denoising (k = 0.05)

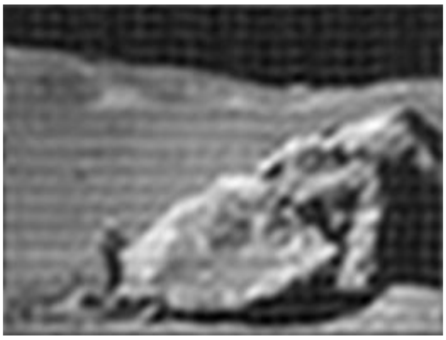

Figure 5. Image After Denoising (k = 0.1)

Figure 6. Image After Denoising (*k* = 0.2)

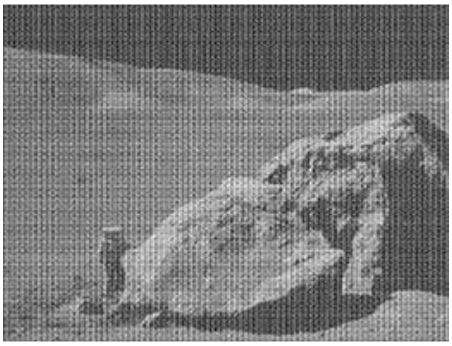

Figure 7. Image After Denoising (*k* = 0.4)

The value of the estimated noise for each image is shown in Table 1.

Table 1. Estimated Noise for different *k* value

k	Estimated noise
Original Image	53.47096
0.4	11.79659
0.2	0.41757
0.1	0.03547
0.05	0.01048

Another important result that can be demonstrated is the critical role of phase in image restoration and reconstruction. By reconstructing the image using only the phase or only the magnitude components, we can clearly observe their differing contributions to the final image. When the reconstruction is performed using only the phase information, the resulting image retains most of the structural details and spatial features, showing that the phase carries the majority of the information required to reconstruct the image's geometry and outline. This highlights the phase's pivotal role in preserving the overall structure and details of the image, even after denoising. In contrast, reconstructing the image using only the magnitude component yields a vastly different result. The magnitude contains information primarily about the intensity of the frequency components, and without the corresponding phase information, the resulting image often appears very dark, featureless, or entirely black. This underscores the limited role of magnitude in retaining structural details. Therefore, in the context of image restoration and denoising, preserving phase information is critical to ensure the retention of essential details and the accuracy of the reconstructed image.

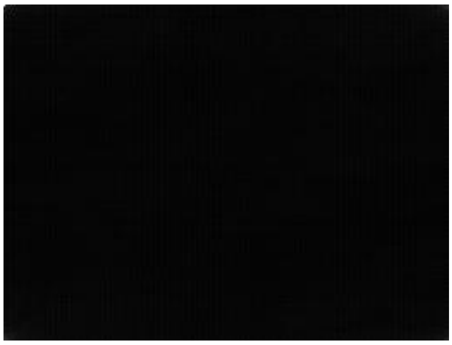

Figure 8. Image Using Only Magnitude

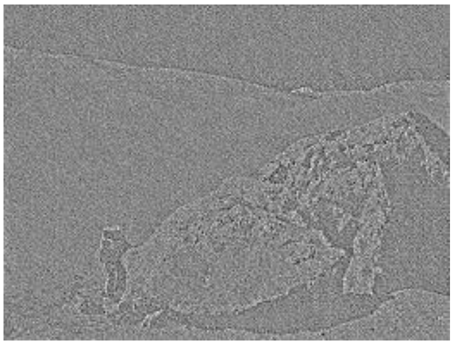

Figure 9. Image Using Only Phase with Average Magnitude

V. CONCLUSION

In this paper, we explored the use of the Fourier Transform for denoising astronomical images, demonstrating its effectiveness in reducing noise and enhancing image clarity. By filtering out high-frequency components in the frequency domain, we successfully reduced noise levels while preserving essential image details. However, our findings highlight the trade-off between noise reduction and image sharpness, emphasizing the need to carefully select the frequency retention parameter, *k*, to achieve an optimal balance.

Additionally, through the reconstruction of images using only phase or magnitude components, we showed that the phase plays a critical role in preserving structural information. The magnitude primarily contributes to the intensity, while the phase retains essential spatial details necessary for accurate image reconstruction.

Future work can explore adaptive filtering techniques and more advanced algorithms to address the limitations of high-frequency detail loss while improving noise reduction efficiency. The Fourier Transform remains a powerful tool for astronomical image processing, offering insights into noise removal and the fundamental importance of frequency domain analysis.

VI. APPENDIX

To provide a clearer understanding of the concepts and methods discussed in this paper, a supplementary video has been prepared. Link: https://youtu.be/FnG_xncsA24

VII. ACKNOWLEDGMENT

The Author would like to express gratitude to Mr. Rinaldi Munir, the lecturer for Linear Algebra and Geometry at Bandung Institute of Technology, for his guidance and comprehensive teaching of the subject, which provided the foundation for understanding the concepts applied in this paper.

The Author also acknowledges the various sources of information that greatly contributed to the completion of this paper. These include academic journals, articles, online resources, and publicly available code repositories, all of which offered invaluable insights and practical tools necessary for the development of this work.

.

REFERENCES

[1] M. Ullsperger and S. Debener, *Signal Processing for Neuroscientists: An Introduction to the Analysis of Physiological Signals*, Chapter 6, Academic Press, 2010.

[2] P. Harrison, "Fourier Transform." [Online]. Available: https://faraday.physics.utoronto.ca/PVB/Harrison/FourierTransform.pdf.

[3] D. H. Bailey and P. N. Swarztrauber, "A fast method for the numerical evaluation of continuous Fourier and Laplace transforms," *SIAM J. on Scientific Computing*, vol. 15, no. 5, pp. 1105–1110, Sept. 1994.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson, 2008.

[5] "Fourier Transform and Applications." [Online]. Available: https://www.corsi.univr.it/documenti/OccorrenzaIns/matdid/matdid027832.pdf.

[6] "Image Filtering in Frequency Domain | Image Processing II," YouTube. [Online]. Available: https://youtu.be/OOu5KP3Gvx0?si=fuIxNOb1EQQegGuh.

[7] "FFT Image Denoising," Scipy Lectures. [Online]. Available: https://scipy-lectures.org/intro/scipy/auto_examples/solutions/plot_fft_image_denoise.html#read-and-plot-the-image.

[8] J. Immerkær, "Fast noise variance estimation," *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 300–302, Sep. 1996.

[9] R. N. Bracewell, *The Fourier Transform and Its Applications*. McGraw-Hill, 2000.

STATEMENT

Hereby, I declare that this paper I have written is my own work, not a reproduction or translation of someone else's paper, and not plagiarized.

Bandung, 01 January 2025

Henry Filberto Shenelo 13523108