PROCESSING OF IMAGES

ID: 20493425

DEGREE: MSC ELECTRONIC COMMUNICATIONS AND COMPUTER ENGNEERING

MODULE: DIGITAL SIGNAL PROCESSING EEEE4120

Table of Contents

INTRODUCTION

Noise in images can occur during the capture, processing, or transmission of an image. This happens because of factors such as electronic interference during said capture, and data compression after transmission. It has an impact on the quality of images by producing random variations in pixel intensity values, resulting in a grainy, distorted or dim appearance of the image. This makes it difficult to detect fine details in an image. In addition, noise can interfere with image processing algorithms, leading to inaccurate results. In this case, astronomical objects are often so far away that it is difficult to produce high resolution images of them. Pictures in low-light conditions and high ISO settings which are often produced resulting in images with more digital noise or grain. This project is to remove specific noise from each image out of three in order to improve the quality and clarity.

TYPES OF NOISE AFFECTING IMAGES:

Salt-and-pepper noise: This noise is characterized by random black and white pixels scattered throughout the image thereby “salting” the image.

Gaussian noise: This noise is distributed according to a Gaussian distribution. It can be caused by various factors, such as electronic noise in an image sensor or fluctuations in the intensity of light.

Shot noise: This is caused by light and mostly seen in low-light images. It is a random fluctuation in the intensity of the light and results in a "grainy" appearance and is difficult to remove without losing image detail.

Impulse noise: This type of noise consists of random pixels that are either much brighter or much darker than the surrounding pixels because of high-amplitude spikes in the intensity of the image. It is caused by electrical interference.

Periodic noise: This noise happens when there is a quantization error and it is characterized by visible stripes in the image. These visible stripes appear periodically through the image due to the periodic fluctuations in pixel intensity.

Chromatic aberration: This is a type of distortion that can occur in photographic lenses when all colours of light are not focused on the same point. Fringes of colour is seen around the edges of objects in the image giving a blurred effect. It can be avoided by using a good quality lens.

AIMS AND OUTCOMES:

The aim of this is to develop filters to able to process three images provided (raw1-image22.png, raw2-image22.png and raw3-image22.png) by removing noise and aberration in MATLAB. This will involve the development of algorithms and techniques for identifying and correcting these issues in these digital images.

The outcome of this coursework will be images with a much less noise and aberration given in the raw image files. This involves implementing a range of image processing techniques, such as smoothing, sharpening, and edge detection for each individual image in order to improve the overall quality. Another outcome is to have a high value of signal-to-noise ratio so the filtered image looks close enough to the original image.

REVIEW OF NOISE SOURCES:

NOISES AFFECTING 'raw1-image22.png':

Salt and pepper noise

Salt and pepper noise occurs randomly throughout the image. It is caused by errors in the data transmission process, faulty pixels in an image sensor or electrical interference. They impact images by making the image appear grainy and blurry, and can also cause loss of detail and contrast. The image will be less clear and more difficult to analyse. To mitigate salt and pepper noise, use median filtering, which replaces the intensity value of each pixel with the median intensity value of the surrounding pixels. This helps smooth out the noise and improve the overall quality of the image. The noise in mathematical form is represented as:

$$P(z)\begin{cases} Pa & for\ z = a \\ Pb & for\ z = b \\ 0 & otherwise \end{cases} \quad [1]$$

Low light intensity

Low light intensity in an image is when the level of light of the image being captured by the camera is low, resulting in a dim image that appears dark and may have a reduced level of detail and contrast. This can make the image less useful for certain applications. Its real-world source is poor lighting conditions, the time of day if too dark or the camera settings. To avoid this noise completely, it is advised to use a camera with a larger aperture, which allows more light to enter the lens and reach the sensor. To mitigate low light intensity in an image, increase the exposure using image enhancement MATLAB functions to improve the overall brightness and contrast of the image. Low light intensity in an image can be represented by a low level of pixel intensity values, which can be displayed using a probability distribution function. The pixel intensity values in the image follows a Gaussian distribution with a low mean and low standard deviation.

NOISES AFFECTING 'raw2-image22.png':

Shot noise or Poisson noise

Shot noise or Poisson noise leads to speckled images. Its real-world source is caused by the random fluctuations of photons in the imaging system, which can result in random variations in the pixel intensity values. To mitigate shot noise, one approach is to use the 2D FFT image filtering technique to detect and emphasize certain features in the image. The approach used in this paper is with the use of a median filter which reduces the appearance of a grainy look. This helps reduce the random fluctuations in the pixel intensity values and improve the overall quality of the image.

In mathematical terms, shot noise can be represented as:

$$P(N) = \exp(-(N))\frac{(N)^N}{N!} \quad [2]$$

This equation is called the Poisson distribution. N is the amount of detected photons on average. The standard deviation of this Poisson distribution is equal to the square root of the mean of the distribution. This equation describes the probability of a given number of photons reaching a given pixel in the image, which can then be used to calculate the expected pixel intensity values.

Periodic noise

Periodic noise reduces the contrast and clarity of the image and its real-world source is caused by electrical interference, issues with the imaging sensor or incorrect camera exposure settings. To mitigate periodic noise, one approach is to use image filtering techniques, such as median filtering or low-pass filters, which can help to smooth out the periodic fluctuations in the pixel intensity values and improve the overall quality of the image. In this report 2D FFT is filter is applied. Periodic noise can be displayed using a periodic function, such as a sine or cosine function. This function can be used to describe the periodic fluctuations in the pixel intensity values, and can be used to calculate the expected intensity values for a given pixel in the image.

NOISES AFFECTING 'raw3-image22.png':

Chromatic Aberration:

Chromatic aberration can make the image appear less sharp and can make it more difficult to accurately detect or identify objects in the scene. Chromatic aberration is often caused by the optics of the imaging system, such as the lenses, which can refract light differently based on its colour. It can be avoided by using a more pronounced wide-angle or zoom lens. To mitigate chromatic aberration, an approach is to use an image processing technique: deconvolution, which helps correct the distortion.

METHODS AND RESULTS:

Processing solutions

Salt and Pepper noise in 'raw1-image22.png'
Implementation of Median Filter

A median filter is used to remove the salt and pepper noise in an image. To implement median filter to remove salt and pepper noise from an image, the input parameters for the filter were defined, including the size of the neighbourhood which replaces each pixel with the median value of its neighbouring pixels. This can help to reduce the amount of noise in the image, by replacing noisy pixels with more consistent values. In this code, the medfilt3 function is applied to the input image cl, and the result is stored in the output image I_filtered.

Justification of Median Filter

The median filter is good for removing this type of noise because it is not sensitive to outlier values. The input parameters for the filter '[5,5,3]' specify the size of the filter window, with a 5x5 window in the first two dimensions and 3 in the third dimension for pixel. This filter size is large enough to capture a significant number of neighbouring pixels, which can help to accurately estimate the median value and reduce the amount of noise in the image. It is also small enough because the larger the filter size, the more aggressive noise reduction needs to be which could lead to more blurriness and loss of detail in the image. An example is showed in Figure 2 below, when the third dimension neighbourhood was increased to 5 from 3, the image turned grey and lost some detail.

The results obtained from processing

Figure 1 showing the removal of salt and pepper noise after using a median filter

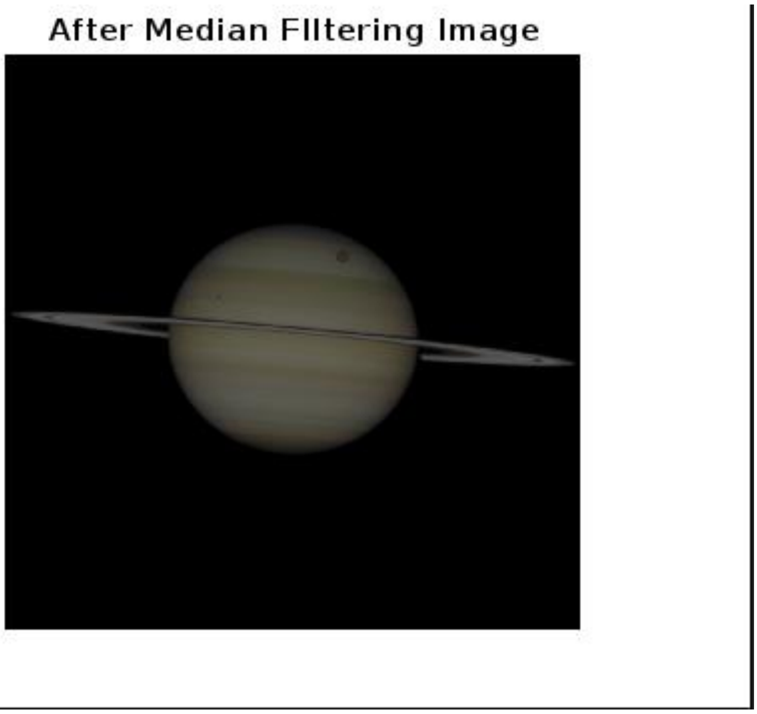
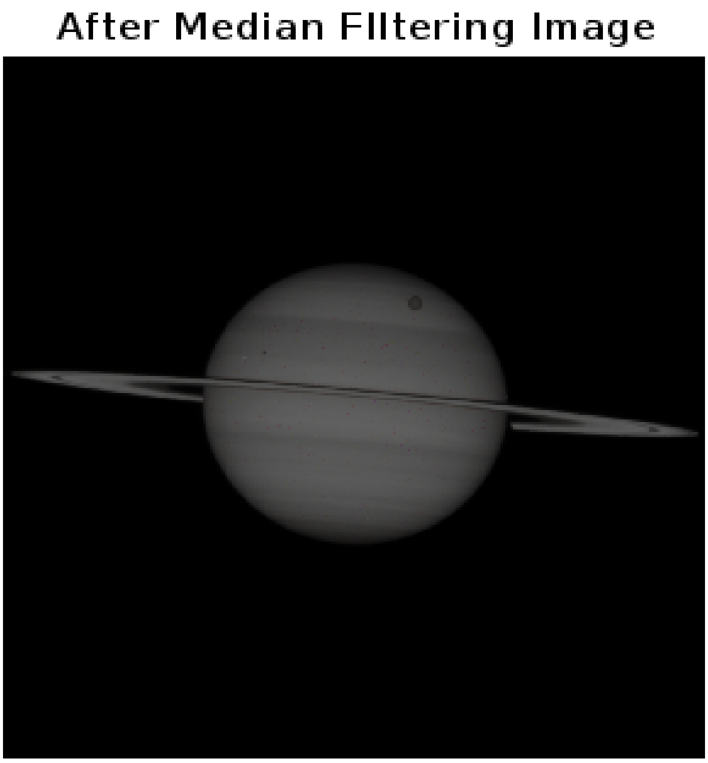


Figure 2 showing the removal of salt and pepper noise after using a median filter but with the wrong parameters



Low Light Intensity in 'raw1-image22.png'

Implementation of Method

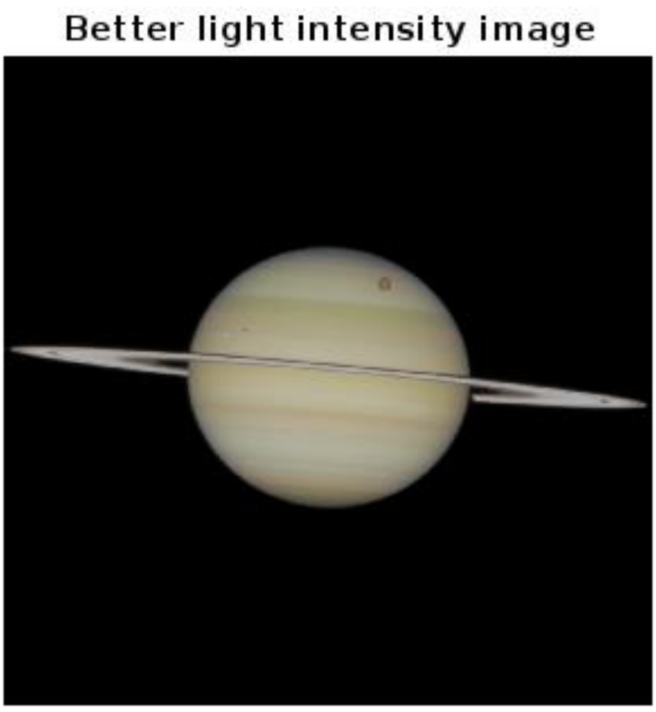
The function 'imlocalbrighten' is known for brightening low-light intensity areas in RGB images. 'imlocalbrighten' uses a contrast enhancement method to brighten the dark areas of the input image, while preserving the details and colours in the brighter areas. This function is applied to the already filtered image from figure 1 above. 'imlocalbrighten' identifies the dark areas and applies a nonlinear scaling function to brighten the said dark areas. An input parameter of 0.4 is used and specifies the amount of brightening the image needs. This is then all stored in the B variable.

Justification of Method

In using 'imlocalbrighten', only the brightness of small regions is affected, when other methods were tried, the brightness of the whole image was affected which was not necessary. This helps preserve contrast of the image and the look of areas too dark. 'imlocalbrighten' uses a nonlinear scaling function hence adjusting the brightness per pixel. Most methods use linear scaling which is more prone to loss of detail. The input parameter '0.4' is used because any higher, the image appears too bright and any lower, it is quite dim.

The results obtained from processing

Figure 3 showing better light intensity image in 'raw1-image22.png'



Analysis of the obtained results:

The obtained results is as close to the clean image as possible, as with filtering it would not exactly be copy and paste results. The median filtering did the part by removing the salt and pepper noise in the image as not one grain of it is left behind in the processed images as seen in the above pictures. In the case of brightening the image, a brighter image would've been preferred and closer to the clean image but if the input parameter got any higher than 0.4, most of the detail will be lost.

Shot (Poisson) noise in 'raw2-image22.png'

Implementation of Median Filter

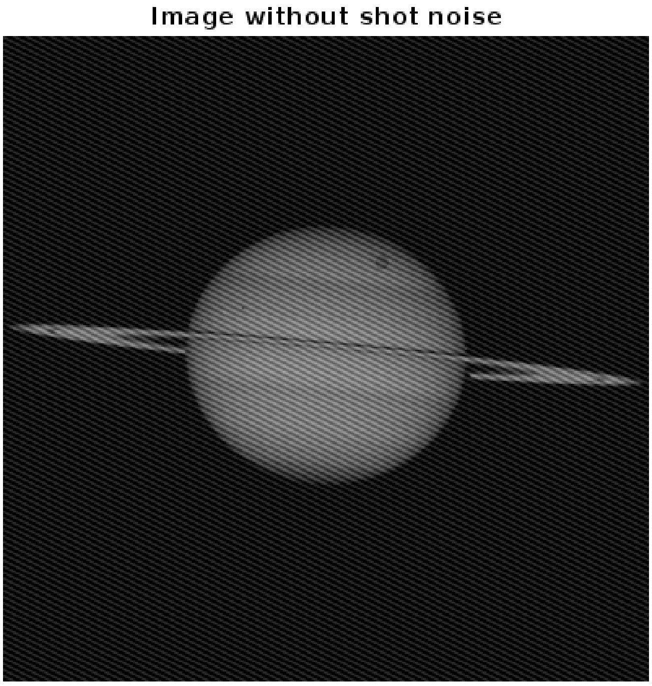
A median filter is used to remove the shot noise in this image. The implementation of a median filter works by replacing each pixel in the image with the median value of close by pixels thereby smoothing out the raw2-image using the medfilt2() function.

Justification of Median Filter

This removes any isolated pixels that may be caused by shot noise by using the median value. As stated earlier on, median filtering is a non-linear filtering technique unlike the gaussian filter which means the edges are preserved and not blurred out and the gaussian filter is not as effective in removing isolated pixels which is the cause of grainy images caused by shot noise.

The result obtained from processing

Figure 4 showing the removal of shot noise after using a median filter



Periodic noise in 'raw2-image22.png'

Implementation of 2D Fast Fourier Transform (FFT)

2D FFT and a low-pass filter is used to remove periodic noise from the input image after the shot noise has been removed. The code read the 'raw2-image22' and converted it to grayscale. The code then processes the FFT of the noisy image and shifts it so that the DC component is in the centre. This is done so the low-pass filter can be applied. The FFT in figure 5 is then displayed to show the presence of high-frequencies, which gives rise to the periodic noise in the image. The code applies a low-pass filter to the shifted FFT and it is implemented as a circular region centred at the middle of the FFT, with a radius of input parameter '100'. This filter allows low-frequency components of the FFT and removes high-frequency noise components. After applying the low-pass filter, the code shifted the FFT back to its original position and processes the inverse FFT to get the denoised image.

Justification of 2D Fast Fourier Transform (FFT)

The FFT and low-pass filter method is the go to method for the removal of noise in this image because it removes periodic noise easily. By applying a low-pass filter to the FFT, the high-frequency components in the noise is removed whilst preserving the low-frequency components therefore keeping the detail and originality of the image. The specific choices of input parameters (e.g. the radius of the low-pass filter) were determined by experimenting with different values and observing the resulting denoised images. In this case, a radius of 100 was chosen because it removed the periodic noise while still preserving the details of the underlying image.

The values used in the code for the low-pass filter (i.e. i - center(1))^2 + (j - center(2))^2 > radius^2) are chosen to create a circular region centred at the middle of the FFT. This circular region acts as the low-pass filter, letting through only the low-frequency components of the FFT.
The outer loop, for 'i = 1:size(filter, 1)' and the inner loop 'for j = 1:size(filter, 2)' has to do with dimensions of the FFT.

The if statement, 'if (i - center(1))^2 + (j - center(2))^2 > radius^2' checks whether the current location in the filter is outside of the circular region. This is done by computing the distance from the current location to the centre. If the distance is greater than the radius of the filter, then the current location is outside of the circular region and the value of the filter at that location is set to zero filter(i, j) = 0; removing the high-frequency components of the FFT at that location.

The results obtained from processing

Figure 5 showing the shifted FFT of the input image

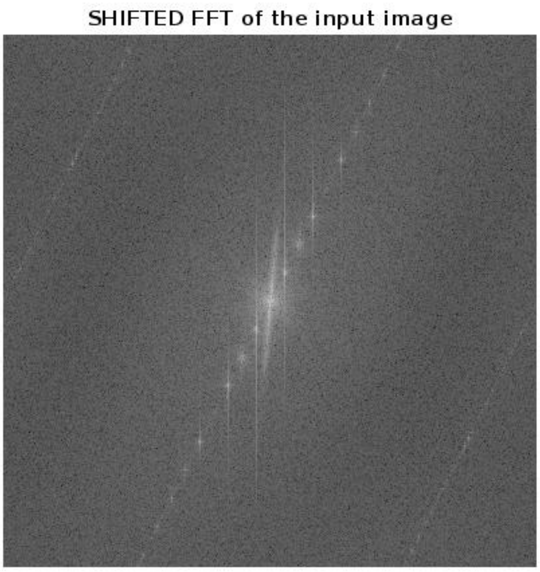
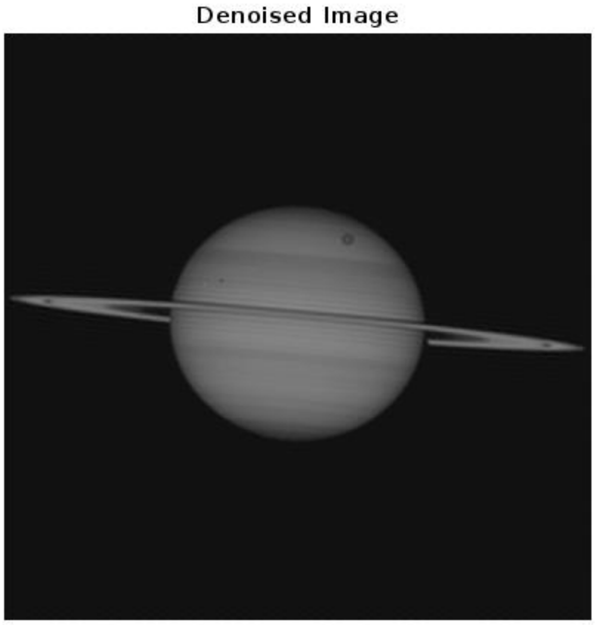


Figure 6 showing the denoised image (image without shot and periodic noise)



Analysis of the obtained results:

The obtained results is as close to the clean image as possible. The image has been turned to grey for processing purposes and the shot noise and periodic noise are not visible in the images above. In the case of the shot noise, the processed image appeared to be less grainy than it appeared. This can be noticed with very close observation as shot noise is a difficult noise to remove and detect as it happens in low light images whereas one has to strain. Median filtering is not perfect and might have possibly not gotten it all but as said earlier, the clean image will be the only perfect image and these filtered ones are to be as close as possible. In the case of the periodic noise, the stripes were removed with the use of 2D FFT and low pass filters described above. These filters did the part as one would not know this noise affected the image when looking at the processed image. It was easier to detect than the shot noise.

Chromatic Aberration in 'raw3-image22.png'

Implementation of Method

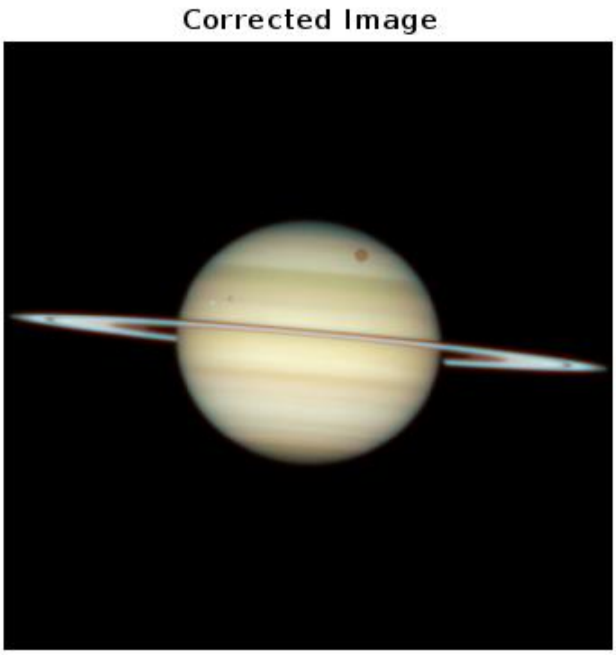
This code first converts the image to the Lab which separates the image into its luminance L and chromatic (a and b) components. Then the a and b channels are shifted to correct the chromatic aberration and after the correction, these components are combined to the corrected channels into one image and converts the image back to the RGB.

Justification of Method

This is a good choice for correcting chromatic aberration because it divides the luminance and chromatic components of the image, allowing for a better correction of the chromatic aberration. The input parameters for the 'imtranslate' function is the amount needed to shift the a and b channels. Chromatic aberration is a type of distortion that can occur in images due to the different wavelengths of light being focused at different points and this function shifts the a and b channels to correct the distortion and blur.

The result obtained from processing

Figure 7 showing the removal of chromic aberration



Analysis of the obtained results:

Chromatic Aberration is a hard noise to detect and remove as sometimes as in this case you have to strain to notice the difference between an image without it and one with. The corrected image looks slightly less blurred than the raw3-image22 provided indicating the chromatic aberration has been removed.

Conclusion

The effectiveness of using image processing techniques in MATLAB has been demonstrated to remove noise and aberration from three images, two of these images having two different kinds of noise. These noises can be caused by factors such as electronic interference, data compression, and fluctuations in light intensity, impacting the quality. The goal is to produce high-quality images with a high signal-to-noise ratio.

A median filter was used to remove salt and pepper noise, which resulted in a cleaner and more accurate representation of the original image. The 'imlocalbrighten' function was then used to remove low light intensity noise, which improved the overall appearance of the image by increasing its apparent brightness while preserving its contrast and detail. The median filter worked well for the salt and pepper noise as each pixel is replaced by the median value of neighbouring pixels instead of the average. The imlocalbrighten function did not work as well as I would have liked but in comparison to other options seemed to be the best option.

Shot noise is often seen in low-light images. It is caused by fluctuations in the intensity of light and results in a grainy appearance that can be difficult to remove without losing image detail. Median filtering was able to reduce the shot noise for the same reason as mentioned above. The quality of the image in terms of shot noise was improved by applying this but I am not confident the complete removal of all of the shot noise was achieved. The stripes caused by periodic noise because of the pixel intensity was mitigated completely with the use of a 2D FFT and low-pass filter. This combination was perfect for the expected results as the 2D FFT identifies and removes periodic noise. It converts a time-domain signal (in this case, an image) into a frequency-domain representation. This representation shows the amplitude and phase of the different frequency components that make up the signal. The combination of the two selectively removes the periodic noise and produces a better result rather than using the low pass filter alone, which would only attenuate all high frequencies in the image, including the details.

Chromatic aberration happens because of the different wavelengths of light focused at different points. Separating the image to L, a and b allows for control over correction process as image is split into individual components and adjustments to each is made. With this method, the colour output of the image is preserved.

References

[1] “Digital Image Processing Chapter 5: Image Restoration.” 23-Jun-2006.

[2] “Quick facts #3: Poisson noise.” radio.astro.gla.ac.uk.