Research in Astronomy and Astrophysics, 25:065005 (13pp), 2025 June

https://doi.org/10.1088/1674-4527/adcb8c
CSTR: 32081.14.RAA.adcb8c

An Iteration Method for Astronomical Image Deconvolution and Reconstruction

F. Sine[1], M. Lu[1], and L. Zhang[1,2,3]
[1] College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China; sinefu@hotmail.com, lizhang.science@gmail.com
[2] State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
Received 2025 January 23; revised 2025 March 17; accepted 2025 April 7; published 2025 May 28

Abstract

Imaging is an important method for astronomy research. In practice, original images acquired by a telescope are often convolved and blurred by the point-spread function (PSF), which is a very unfavorable situation for many scientific studies including astronomy. This paper introduced a single equation iterative method for solving complex linear equations, and this method can deconvolute dirty images, eliminate the effects of the PSF well. With different PSFs, this algorithm shows very good results in deconvolution. Also, with a giant PSF of aperture synthesis imaging, this algorithm improves the peak signal-to-noise ratio and structural similarity of the dirty images by 41.0% and 33.9% on average. In addition, this paper proves that the algorithm can deconvolute the dirty image by making full use of the information of each pixel in the image, even if the dirty image has salt and pepper noise or even lost areas; by its excellent properties of flexible operation to a single pixel, all these bad situations can be dealt with and the image can be restored.

Key words: methods: numerical – techniques: image processing – techniques: interferometric

1. Introduction

Image deconvolution has always been an important research direction in the field of astronomical image processing, because in many cases, an image obtained from a telescope is inevitably blurred due to many factors, such as limited resolution, bandwidth limitation and the point-spread function (PSF). The major factor is the PSF, convoluting and blurring the image, which makes us hard to see the details of astronomical objects. The PSF, literally making the point light source spread around, is generally the impulse response of an imaging system. The impact of PSF is transferring the true light distribution into a worse linear space. A bigger main lobe or sidelobe will let the basis less perpendicular to each other to make the possible true image solution spread in many locations, which is hard to solve.

A group of radio telescopes can use Very Long Baseline Interferometry (VLBI) for imaging, but the antenna sampling on the virtual objective lens cannot completely cover the entire observation surface, the PSF of the virtual telescope will have a non-negligible sidelobe, which will make the image very blur (Faulkner & de Vaate 2015). In this case, a good deconvolution algorithm is needed, trying to solve the true image and make it clear enough to see details.

From the traditional signal processing view, the function of a linear time-invariant system in the frequency domain is to multiply the spectrum of the original signal by a system function, which is reflected in the time domain as a convolution process, and in frequency domain the deconvolution is eliminating the system function of convolution. For images, it can also be deconvoluted by eliminating the system function of convolution in the frequency domain (Rafael & Gonzalez 2017), which needs to extend to two-dimensional discrete Fourier transform. In real cases, it is difficult to do this directly, some preprocessing is generally needed to the dirty images (Dong et al. 2017; Rafael & Gonzalez 2017).

There is much research on image restoration and reconstruction by image feature analysis, including the methods of establishing dictionaries through sparse representation, which match features in dirty images to restore them. Based on image feature analysis theories, Yang et al. (2010), Hu et al. (2020) studied image reconstruction methods and other issues. The light distribution of original images should be inferred from the specific features in dirty images. The problem lies in how to establish the relationship between these features and restoration. Studies by Lu et al. (2014), Dai et al. (2012), Zhang et al. (2017) have discussed how to establish sparse representation dictionaries and optimize methods for image reconstruction. Although the results are great somehow, unfortunately, these methods, which are all data-based, have their common defects. They cannot analyze the convolution process quantitatively and accurately like traditional methods, and are hard to deal with different situations in the convolution process and so difficult to ensure the accuracy and effectiveness of the deconvolution results. So, the application of these methods in astronomical imaging is not so ideal.

[3] Corresponding author.

In the field of astronomical image processing, Högbom (1974) proposed the CLEAN algorithm (Högbom CLEAN) in 1974, which is to find the intensity and position of peaks one by one in dirty images, make the convolution kernel multiply by the intensity, and then subtract it from the dirty image, and finally restore the image with Gaussian function as the convolution kernel. After years of research and improvement, Högbom CLEAN has improved quite a lot, among which the most representative is the adaptive scale pixel decomposition algorithm (Asp-CLEAN algorithm) proposed by Bhatnagar & Cornwell (2004), which minimizes the objective function to find the optimal model components. To speed up this method, Zhang et al. (2016) proposed a faster algorithm, namely the Asp-Clean2016 algorithm. Although the Asp-CLEAN already includes a zero-scale approximation of compressed radiation, their computational methods are not efficient. Later, (Zhang et al. 2018, Zhang et al. 2019) combined the Högbom-CLEAN algorithm and the Asp-Clean2016 algorithm to propose the fused-clean algorithm. The combination of these two algorithms not only ensures the effective separation of emission and noise, but also speeds up the deconvolution. Besides, there are other scale adaptive algorithm variants, such as the later improved algorithm by Zhang et al. (2020).

The CLEAN algorithm has made great achievements in the field of astronomical image deconvolution, but this kind of algorithm did not fundamentally solve the deconvolution problem, it is just formally eliminating the complex sidelobe of convolution kernel in image, then making the image look cleaner. It perhaps does not restore much new effective information. The Richardson-Lucy algorithm (Richardson 1972; Lucy 1974), another method in space domain, can also deconvolute images. This algorithm iterates images based on the Newton iterative method to restore images. This algorithm has a good effect on many simple deconvolution problems, but it also has some defects in applicability. First, this algorithm does not guarantee its convergence; second, the algorithm is sensitive to the defects in dirty images and difficult to deal with. Fish et al. (1995), Laasmaa et al. (2011) have studied features, applications, and improvements of this algorithm.

The method used in this paper is also a traditional deconvolution method in spatial domain, not a popular data-based method, the explanation is relatively similar to the Richardson-Lucy algorithm, but if choosing the way solving linear equations to do deconvolution, it is unnecessary to use the Newton iterative method. In history, there is a better way to solve massive linear equations, which is the Kaczmarz algorithm, proposed by Kaczmarz (1937). But this kind of algorithm was not mentioned in many linear algebra and numerical analysis textbooks, we did not know the history of this method until the experiments finished long time. This method got several improvements after its invention, in 2009, Strohmer & Vershynin (2009) introduced randomization to this method, Needell (2010) studied in its random sampling, these improvements speed up the convergence of this method and rise the quality of solution. It is highly consistent with our improvements in the iteration sequence in the program to make the results better. This method was introduced to Computed Tomography (CT) image reconstruction by Gordon et al. (1970) named algebraic reconstruction technique (ART) in the medical field.

Introducing this method to astronomical image deconvolution and reconstruction will gain many advantages. First, it does not have to calculate the whole image or a block area at the same time, it can do iterative operations at the pixel level, which gives it great flexibility to deal with various situations and complex problems in the image. Also, this algorithm can ensure its convergence for deconvolution, thus it can do more iterations, and the result will be closer to the original image.

2. Imaging Process and Deconvolution Method

2.1. The Mathematical Description of Imaging

Traditionally, in the spatial domain of the image, convolution dirty image $I_b(x,\ y)$ is the result of each light source $I_o(x_0, y_0)$ in the original image $I_o(x, y)$ passing through a linear system with the response $h(x,\ y)$ into a weighted diffused distribution:

$$I_b(x,\ y) = \iint I_o(\tau, \varepsilon) h(x - \tau,\ y - \varepsilon) d\tau d\varepsilon \tag{1}$$

In the discrete case, the expression should be:

$$I_b(x,\ y) = \sum_{\tau}\sum_{\varepsilon} I_o(\tau, \varepsilon) h(x - \tau,\ y - \varepsilon) \tag{2}$$

On the contrary, when there is a convolution dirty image, the value of each point $I_b(x_0, y_0)$in the dirty image is determined by the brightness of the light source distribution centered around $(x_0,\ y_0)$ in the original image according to a linear response $h_T(x, y)$, that is:

$$I_b(x_0,\ y_0) = \iint I_o(x - x_0,\ y - y_0) \quad h_T(x,\ y) dxdy \tag{3}$$

In the discrete case:

$$I_b(x_0,\ y_0) = \sum_{x}\sum_{y} I_o(x - x_0,\ y - y_0) \quad h_T(x,\ y) \tag{4}$$

where $h_T(x,\ y)$ is the central symmetry of the convolution system response $h(x,\ y)$. This is obviously a linear equation, which, given $I_b(x_0,\ y_0)$, can be represented by a hyperplane in n-dimensional or even infinite dimensional space. There is a different thinking, in the narrative of linear equations, a brightness value $I_b(x_0,\ y_0)$ in the dirty image will approximately constrain the light source distribution centered around the (x_0, y_0) in the original image $I_o(x, y)$ in a hyperplane. That is only light distributions in this hyperplane, yielding $I_b(x_0,\ y_0)$ in the dirty image after convolution.

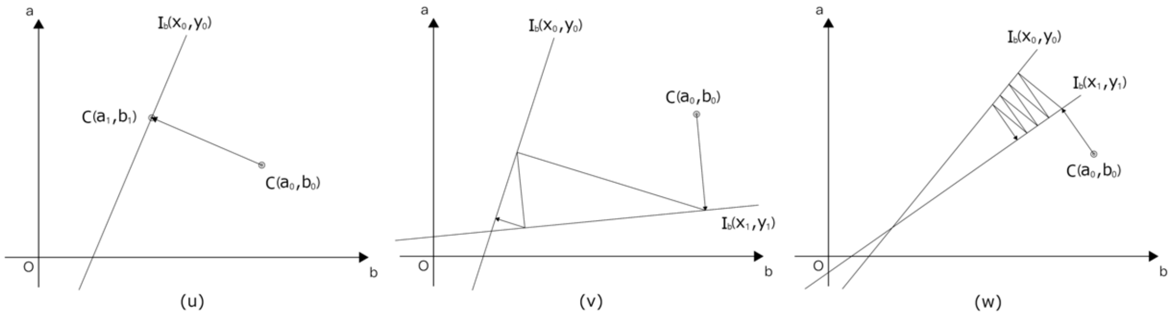

Figure 1. (u) To get a light distribution in a single pixel hyperplane, the most direct way is moving along with the normal line and get into the hyperplane; (v) The case that the angle between the linear equations is greater; (w) The case that the angle between the linear equations is smaller.

Convolution dirty image $I_b(x, y)$ has many values of light, which will produce a series of hyperplanes to constrain the image distribution $I_o(x, y)$ in n-dimensional space. To restore the image, it should find an image distribution as far as possible that can meet all these constraints. Even if these constraint solutions are not unique or can never be found exactly, it should try to find one of them as close as possible.

This can also be considered as solving a matrix equation, which is obviously not easy, if converting it into a matrix problem, the construction of this matrix equation will be extremely difficult to get, we must first transform $h(x, y)$ into a large matrix, and then cut and splice this picture into a vector. If $h(x, y)$ is slightly larger, this vector will be very long. This matrix equation generally has infinitely many solutions or even is impossible to solve.

Particularly, for radio interferometric imaging, it is not needed to deconvolute the image after it was convoluted. In this point of view, these equations can be set directly by phase and amplitude data. After several selection and adjustments, solving them together by some methods is also good ways to get the image.

2.2. Linear Equation Solving Strategy

As mentioned above, point values $I_b(x_i,\ y_i)$ in dirty image $I_b(x, y)$ should constrain the light source distribution into many hyperplanes around the points $(x_i,\ y_i)$ in the original image $I_o(x, y)$. Suppose that an initial light source distribution $I_c(x, y)$ is not in these hyperplanes, to get a light distribution in the hyperplane, obviously a reasonable method is entering the hyperplane directly along the normal direction of the hyperplane, at least to ensure that the new result must be in the hyperplane, and it is easy to calculate. This method first proposed by Kaczmarz (1937), and it is already applied in CT scan image reconstruction by Gordon et al. (1970). Introducing it to astronomical image deconvolution, discussed only in the discrete case, the $h_T(x,\ y)$ can be seen as a vector and the normal direction of the hyperplane, the distance from the initial distribution $I_c(x, y)$ to the hyperplane can be calculated by:

$$d = \frac{\sum_x\sum_y h_T(x,\ y) I_c(x - x_0,\ y - y_0) - I_b(x_0,\ y_0)}{\sqrt{\sum_x\sum_y h_T(x,\ \ y)^2}} \tag{5}$$

Notice that there is no absolute value function above the fraction because the distance needs to include the direction relative to the normal vector. The original distribution $I_c(x, y)$ moves distance d alone with the normal vector $h_T(x,\ y)$ to get a new distribution $I_c^1(x,\ y)$:

$$\begin{aligned} & I_c^1(x - x_0,\ y - y_0) \\ & \quad = I_c(x - x_0,\ y - y_0) \\ & \quad + \frac{d}{\sqrt{\sum_x\sum_y h_T(x,\ y)^2}} h_T(x,\ y) \\ & = I_c(x - x_0,\ y - \ y_0) \\ & \quad + \frac{\sum_x\sum_y h_T(x,\ y) I_o(x - x_0,\ y - y_0) - I_b(x_0,\ y_0)}{\sum_x\sum_y h_T(x,\ y)^2} h_T(x,\ y) \end{aligned} \tag{6}$$

In general, the scope of convolution kernel $h(x, y)$ is not very large, most of values outside a small area of the center are very small. If the influence area is small, they can directly be set to zero, then the calculation of the above formula will be very simple, updating $I_o(x, y)$ only needs to be in a small area around $(x_0,\ y_0)$. In this way, for the whole image, multiple locations with pixels can be updated at the same time, so it is regarded as a complete iteration when all pixels in the entire dirty image are calculated.

Figure 1(v) and Figure 1(w) show the process of the algorithm in detail. The $I_c(a_i, b_i)$ is $I_c(x, y)$, using (a, b) is to separate from $I_b(x_i,\ y_i)$, which indicates I_c is a distribution. If it is only shown in a two-dimensional space, there are two pixels $I_b(x_0,\ y_0)$ and $I_b(x_1,\ y_1)$, generating two constraint hyperplanes, and the iteration process only runs in these two pixels. In Figure 1(v), the angle between the two hyperplanes is relatively larger, when the algorithm is running, making the light distribution quickly close to their common intersection and it is solving these two linear equations. But in Figure 1(w), the angle between the two hyperplanes is relatively smaller, after

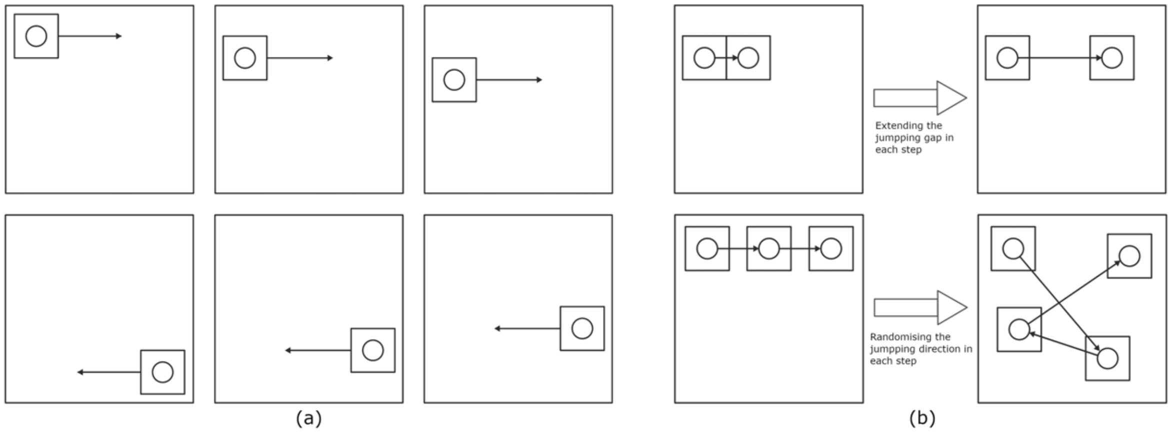

Figure 2. (a) The combination of single direction sequential scanning; (b) The random direction leapfrogging scanning.

many iterations, the result is still far from their common intersection. Generally, the convergency of this method is obvious, despite the no solution cases, the result will definitely converge to a particular solution, even in a no solution case, the result will be bouncing between hyperplanes rather than diverge.

When this method is used in high-dimensional space, it is easy to have such a small angle between the constraint hyperplanes, resulting in slow convergence, and there are always many accumulated errors that are hard to eliminate. This is still one of the inherent defects of this method. Thanks to the flexibility of this method, iterating with a single pixel, this problem can be eased by the iteration scanning design.

Strohmer & Vershynin (2009) randomized the Kaczmarz method, proved that randomized the iteration process can highly speed up this method and raised the effects. Applying it in image deconvolution, because an image is a two-dimensional matrix, there are many ways to traverse their elements. This paper optimized the iteration scanning strategy at the program level, raised the performance greatly. Ordinarily, it is easy to think about a scanning order of left-to-right, top-to-bottom one by one, but the parallelism of these convolution kernels in the adjacent positions is often relatively high, if using this scanning mode will cause errors accumulated in some specific cases, which are difficult to eliminate. Changing the order or direction of scanning and combining different strategies can make the results much better.

First, for the left-to-right, top-to-bottom order, it can be offset by a complementary strategy, setting the right to left, bottom to top order. The scanning strategy can change in this way without increasing the computation consumption of one iteration. It can design various other scanning strategies, such as adding the scanning order from right to left, top to bottom or from left to right, bottom to top.

As shown in Figure 2(a), it can combine scanning in four directions, and it is the first scanning and iteration method adopted in the experiment of this paper. In fact, there are many more complex ways to traverse the entire image.

To further improve the program, getting the second scanning and iteration method is shown in Figure 2(b). The traversal process can be skipped interspersed in the image, to avoid the regions with high parallelism in the traversal and speed up convergence. First, we introduce the leapfrogging scanning to the algorithm and wider the gap properly. Then, we introduce randomness to jump directions, each jump will not be in one direction, but in different directions, until all pixels have been traversed. Finally, the scan order is changed once after each traversal.

The experiment of this study finally draws a conclusion that increasing the randomness of the traversal process can greatly improve the deconvolution effect of the algorithm, the more randomness the better effect.

2.3. Image Defects

If a deconvolution method needs to do something more, there are some cases that are difficult to deal with by other traditional deconvolution methods, that is, there is pepper and salt noise in the image, or some areas that are damaged cannot be used for some reason. The case is shown in Figure 3.

In the face of this situation, the noise in the image is usually smoothed and filtered or segmented the image for deconvolution. However, segmenting the image will lose information of pixels near the incision and fail to make full use of the information.

If using the single equation iteration solving method, they can all be dealt with in the simplest way, that is, when the scanning program detects pixels in the image are invalid, the program can be set to skip these pixels. For the damaged area in the image, it can also directly set the pixels of the damaged area to be invalid. It can handle these problems without any big change of the program.

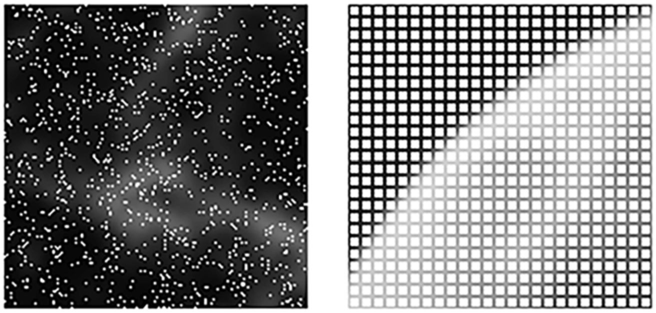

Figure 3. The blur images would sometimes contain salt and pepper noise or missing parts.

In practice, it can set a preprocessing program scanning all the pixels of the image, when the pixel meets some conditions, the pixel is set invalid, and set the pixel to a specific value, such as 1000:

$$\text{if(condition): } I_b(x,\ y) = 1000 \tag{7}$$

Add an operation to the iteration solving program, when it detects a pixel value that is particularly set, skip this pixel of solving:

$$\text{if}(I_b(x,\ y) = =1000)\text{: pass} \tag{8}$$

If a bad pixel value has passed, the error would not affect the result, but that is not enough to reconstruct the image. A good property of convolution itself is that the information of brightness in a position will save to surrounding pixels after convolution. Considering how this method deconvolutes the image, when it calculates a linear equation, a pixel value will change the image distribution around, not a point, that gives a chance to recover the information somewhere lost. It may be full of questions, because in linear algebra, it is a case that these equations are insolvable, but in practice, it is generally enough to recover the image in many cases.

2.4. The Interaction Between Noise and Convolution Kernel

In a general case, the errors or noise sensitivity to convolution kernels can be interpreted as when the point value in the dirty image shifted, the intersection of the hyperplanes would shift more or less. Obviously, it depends on the interrelationship between the pixel equations: In the face of this situation, the noise in the image is usually smoothed and filtered or segmented the image for deconvolution. However, segmenting the image will lose information of pixels near the incision and fail to make full use of the information.

Comparing Figure 4(u) and Figure 4(v), it could be found that when the angle between the hyperplanes is very small, the values of pixels change a little, their intersection will produce a large shift, that means the ability of convolution kernel to stabilize the image is low. In n-dimensional space, the parallelism of hyperplanes depends on the angle between their normal vectors. Suppose there are two hyperplanes in the n-dimensional space, and their normal vectors are $[x_1,\ x_2,\cdots x_n]$ and $[y_1,\ y_2,\ \cdots\ y_n]$.

As shown in Figure 4(w), the angle θ between two hyperplanes is also the angle between their normal vectors, the cosecant of this angle can well reflect the relationship of equations and the interaction between noise and convolution. So, first calculate the distance d from a normal vector to another hyperplane, then get the sine value of the angle θ.

$$\sin\theta = \frac{\sqrt{\sum_n [x_1,\ x_2,\cdots x_n]^2 - d^2}}{\sqrt{\sum_n [x_1,\ x_2,\cdots x_n]^2}}$$
$$\sin\theta = \sqrt{1 - \left(\frac{[x_1,\ x_2,\cdots x_n]\cdot[y_1,\ y_2,\cdots y_n]}{\sum_n [y_1,\ y_2,\cdots y_n]^2}\right)^2} \tag{9}$$

The cscθ is reciprocal of sinθ, it can define a parameter, the shift parallelism, calculated by a convolution kernel $h(x, y)$ with its own shift $h(x - \Delta x,\ y - \Delta y)$: substitute $[x_1, x_2,\ \cdots\ x_n,]$ and $[y_1,\ y_2,\ \cdots\ y_n,]$ as the convolution kernel and its shift, then get the shift parallelism function $\partial_h(\Delta x,\ \Delta y)$ of the convolution kernel:

$$\partial_h(\Delta x,\ \Delta y) = \frac{\sum_x\sum_y (h_T(x,\ y))^2}{\sqrt{\left(\sum_x\sum_y (h_T(x,\ y))^2\right)^2 - \left(\sum_x\sum_y h_T(x,\ y) h_T(x - \Delta x,\ y - \Delta y)\right)^2}} \tag{10}$$

When $\partial_h(\Delta x,\ \Delta y)$ equals 1, the two hyperplanes are perpendicular to each other, and when a pixel value in the dirty image is offset by e, the intersection of these hyperplanes will also be offset by e. When $\partial_h(\Delta x,\ \Delta y)$ value is not 1, when a pixel value in the dirty image is offset by e, the intersection will be offset by s:

$$s = e * \partial_h(\Delta x,\ \Delta y) \tag{11}$$

Considering the shape of function cscθ, the $\partial_h(\Delta x,\ \Delta y)$ is always greater than 1, thus the convolution kernel will always magnify the noise in the dirty image. When $\partial_h(\Delta x,\ \Delta y)$ is very big, a small shift in pixel values can cause a very large shift of the intersection s, quickly destroying the result of deconvolution.

The shift parallelism of a convolution kernel is highly related to its size, shape, shift direction and distance. The shift parallelism can study the ability of these convolution kernels to resist noise in different directions and distances, explain why the image resolution cannot be increased indefinitely, how to adjust the single pixel deconvolution method in process to get better results.

The converging speed of this algorithm also highly depends on it. It could estimate the converging speed at different location and direction in deconvolution process. If somewhere there is a big sidelobe with a very low shift parallelism, it is easily appearing a repeated major pattern like shadow here,

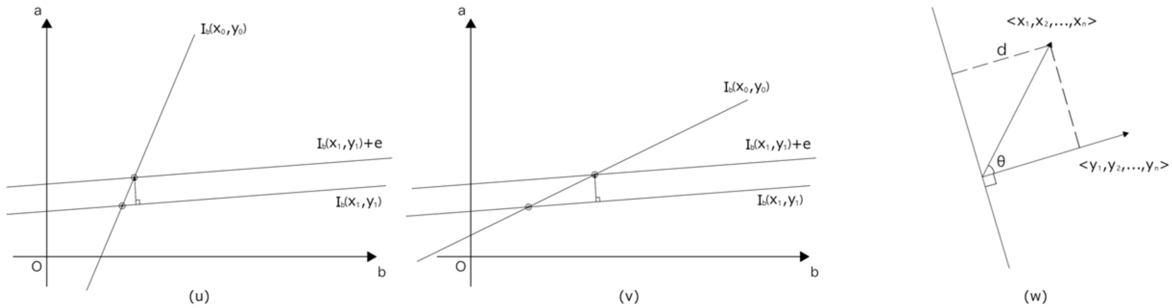

Figure 4. (u)The result shifted when hyperplanes are more perpendicular; (v) The result shifted when hyperplanes are more parallel; (w) The way to calculate the sinθ between constraint hyperplanes.

because the shadow solution in this case is very close to the true solution and hard to remove it.

An important conclusion is that, when the convolution is inevitable but controllable, the convolution kernel with low shift parallelism in specific direction and distance should be selected for saving information.

There is something hidden here, that is the resolution can be sacrificed to deal with noise, for different convolution kernels, as long as the scanning interval is large enough in the image and the shift parallelism is low enough between scanning pixels, the image can be restored as much as possible by reducing the resolution. To some extent, it shows some relationship between noise, convolution and image resolution in the spatial domain.

This noise analysis is applied to normal optical imaging. In radio interferometric imaging, the virtual generated image would not contain pixel value shift, but contain phase and amplitude measuring error, they will impact the results in some different ways but will less impact than thermal noise. The analysis of this kind of noise is basically the same in this point of view, it may be viewed as noise on the PSF or something else, which will be a huge topic out of the major in this paper.

2.5. The Optical Basics of This Method Compared with CLEAN Algorithm

The classical deconvolution algorithm Högbom CLEAN is the basis for most deconvolution algorithms used in astronomical image deconvolution. The CLEAN algorithm is the assumption that the true light distribution of the sky can be decomposed into a finite number of point light sources, and that most of the surrounding region is devoid of light sources. These light sources can be gradually found and removed from dirty images by some calculations.

Compared with CLEAN, this algorithm is also an iterative method, but it is quite different in some basic assumptions. First, this algorithm assumes that the sky image is made up of many point sources, but it assumes that the point sources can be everywhere. For an imaging system, the smaller point sources in an adjacent tiny region can be approximated as a point coherent source due to Fraunhofer (far field) diffraction (Eugene 2021), which means that the whole imaging region can be pre-divided into a finite number of small grid regions. Assuming there is only a coherent source within each region, the number of small regions is the pre-assumed image resolution. The intensity of each region in the image can be known as some linear combination of the coherent source intensities in the surrounding small, divided regions (many details of the optical are omitted here, but the result is the same), in this assumption the deconvolution problem becomes solving linear equations and restoring the true coherent source intensities in the small regions of image. Different from the CLEAN algorithm, it does not search the locations of all point sources, avoids the interference problems encountered in the CLEAN algorithm when searching for the near point sources, because they are all solved together.

Although both this algorithm and CLEAN are traditional non-data-based algorithms, this algorithm is significantly different from CLEAN in some basic assumptions and actual operation. But it can also complete the deconvolution task. This algorithm will have a specific application scenario compared with CLEAN. For example, it will have better flexibility and adaptability in the case of complex surface light sources.

3. Experiments

3.1. Experiment Settings

In the experiment part of this paper, four astronomical images (A, B, C, D) shown in Figure 5(a) are used to test this algorithm and all images have unified format of 512*512 resolution and monochrome among which image A is the nebula and image B is the Sun, which are mainly composed of complex distributed spontaneous light sources to test the algorithm's deconvolution ability under complex point sources. Image C and image D are two lunar surface images, which are mainly composed of large area reflection sources, to test the deconvolution ability of the algorithm in the face of such a large area of low contrast.

Two convolution kernels shown in Figure 5(b) are used to convolute four images, which are all-one matrix and mixed

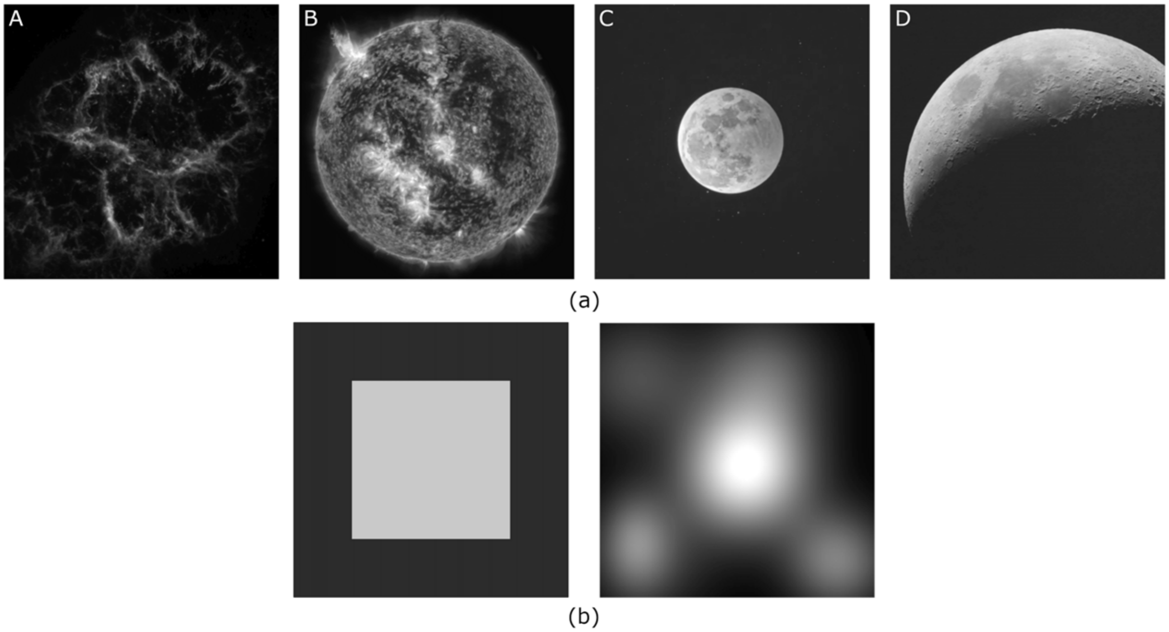

Figure 5. (a) Four astronomical images (A, B, C, D) were used to test the deconvolution performance of the algorithm in the face of high contrast point sources and low contrast reflection sources; (b) The two convolution kernels tested by the experiment.

Gaussian function, and the mixed Gaussian function is a combination of three lobes. These two convolution cores are set for a simple general test of the effectiveness.

The iteration number was set by several pre-testing, showing that at the first simple testing cases, the results reached desired sharpness generally after 200 $\sim$ 300 iterations. The converging speed highly depends on the scanning mode, also size and shape of PSF. So, there is no explanation specifically. But the iteration number could be set by the generating difference, which means generating an image by convoluting the reconstruction image with PSF, calculating the difference between this image and the true dirty image, if reaching a threshold.

In the experiment part, two indexes are used to measure the deconvolution effects, they are peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). PSNR is an index that represents the ratio of the maximum possible power of the signal and the power of the destructive noise that affects its representation accuracy (Welstead 1999). There are two monochromatic images I and K, if one is the noise approximation to the other, their mean square error is defined as:

$$\mathrm{MSE} = \frac{1}{mn} xy[Ix,\, y - K(x,\, y)]2 \tag{12}$$

The definition of Peak Signal-to-Noise Ratio (PSNR) is:

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\max_I}{\mathrm{MSE}}\right) \tag{13}$$

where $\max_I$ is the maximum value of the image point. SSIM is an index to measure the similarity of two images (Wang et al. 2004). For the two images calculated in SSIM, one is an uncompressed non-distortion image I, and the other is a distorted image K. The SSIM of two images is:

$$\mathrm{SSIM}(I,\, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)} \tag{14}$$

where μ is the mean of the image, σ is the variance and covariance of the image, and c is the constant.

3.2. Deconvolution Verification of Different Convolution Kernels

The four images are convoluted with two convolution kernels, and then deconvolved by this algorithm in first muti-direction traversal. Finally, it can be seen from the experiments that a good result can be basically achieved after 200 $\sim$ 300 iterations. Figures 6(a) and (b) show the results of the algorithm. The PSNR and SSIM of these images are shown in Table 1.

Visually, for all dirty images, the algorithm has achieved a good restoration effect, the PSNR has improved by 43.2% on average, and the SSIM has improved by 22.1% on average. It is not difficult to see the performance of the algorithm. What is counterintuitive is that the deconvolution effect is better for more complex mixed Gaussian function convolution kernel. The reason is actually very simple, such complex convolution kernels will have lower shift parallelism in the same location, the information of the original image is better saved in the pixels of the dirty image to facilitate restoration. In general, the

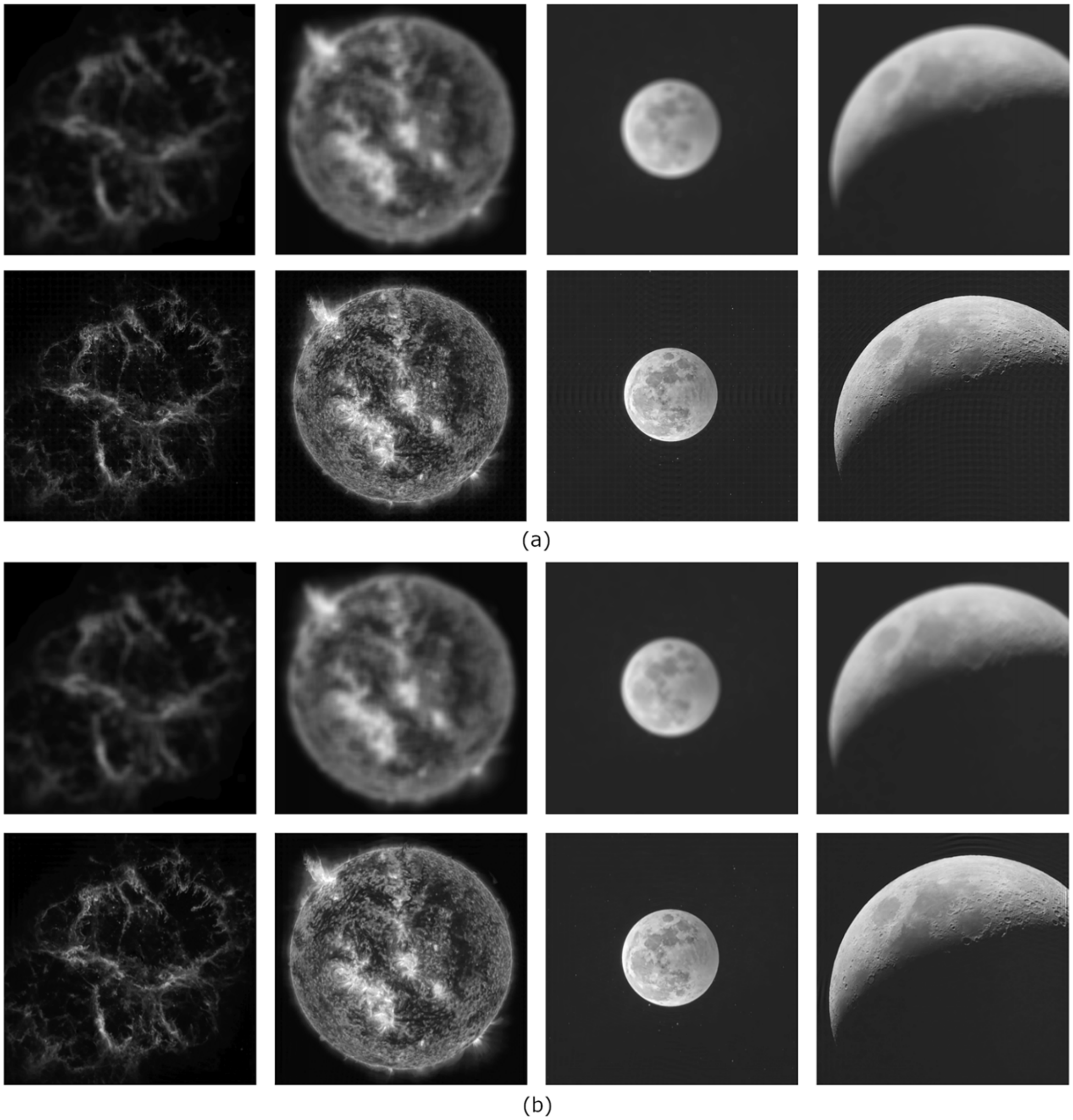

Figure 6. (a) The dirty image generated by all-one matrix convolution kernel and the deconvolution results of this algorithm; (b) The dirty image generated by mixed Gaussian function convolution kernel and the deconvolution results of this algorithm.

program has done the deconvolution restoration for all these dirty images, and the effect is remarkable.

3.3. Deconvolution Verification of Radio Interferometry Imaging

To test the deconvolution ability of the algorithm for radio interferometry imaging, using radio astronomy simulation, calibration, and imaging library (RASCIL)[3] to generate a convolution kernel of radio interferometry imaging. The properties of radio interferometry imaging depend on the number and distribution of antenna arrays, frequency and observation mode settings. Setting the antenna arrays as a standard configuration LOWBD2 from SKA-Low in Square Kilometer Array (SKA) project, rmax = 750, observation mode as snapshot, center frequency at 100 MHz, bandwidth at 1 MHz, R.A. $+15°$, decl. $-45°$, and precession reference date as J2000, that is, 2000 January 1.5. The coordinates use the international celestial reference frame. Then we get a u, v

[3] https://gitlab.com/ska-telescope/external/rascil-main

Table 1
Deconvolution Results of the Single Direction Scanning through Different Convolution Kernels

Using all-one matrix				
Image	PSNR		SSIM	
	Blur	Deconvolution	Blur	Deconvolution
A	28.43	38.27	0.75	0.91
B	24.11	35.32	0.57	0.92
C	29.79	40.35	0.93	0.95
D	22.94	37.85	0.75	0.93
Using mixed Gaussian function matrix				
Image	PSNR		SSIM	
	Blur	Deconvolution	Blur	Deconvolution
A	29.66	42.41	0.79	0.97
B	25.23	38.34	0.64	0.96
C	31.91	45.02	0.94	0.99
D	27.7	37.1	0.85	0.95

coverage of the antenna arrays in the setting observation process. We use Bezier function to fit the *u*, *v* coverage and obtain an approximate distribution of light-through holes on the virtual objective. According to the diffraction law (Eugene 2021), we calculate the two-dimensional Fourier transform of the distribution of light-through holes, to get the approximate convolution kernel of the interferometry imaging in the settings above, which is specifically a distribution with a size of 256*256, as shown in Figure 7.

Analyzing the properties of the convolution kernel, it can be found that its main lobe width is about 25, and there are many sidelobes with larger width and energy, the overall energy distribution is relatively dispersed, and it will significantly damage the original image, which also means that it is very difficult to restore. The experiment still uses the previous four astronomical images to verify the deconvolution ability of the algorithm in the face of astronomical images. The four images are uniformly monochrome with a resolution of 512*512, and the 256*256 radio interferometry imaging convolution kernel is used to convolute and blur these images.

As shown in Figure 8, it can be seen that the damage caused by the radio interferometry convolution kernel is not at the same level as that caused by the simple test convolution kernel used in the previous experiment. By comparison, it can be found that at least some features of the moon surface can still be distinguished under the simple convolution kernel in the previous experiment, but these features cannot be distinguished at this convolution kernel. Such "surface" diffusion source and low contrast situation are almost hopeless for the traditional CLEAN algorithm.

In this part of experiment, the program sets to the leapfrogging and random direction scanning traversal. The

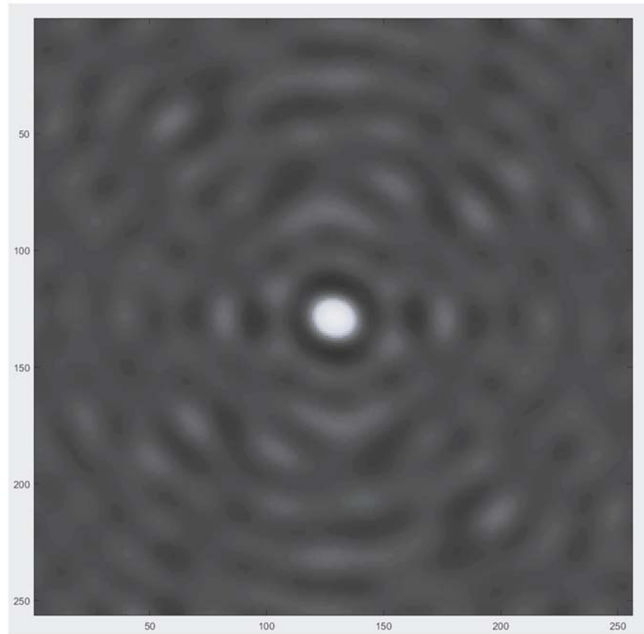

Figure 7. A convolution kernel of radio interferometry imaging from SKA.

result is also shown in Figure 8, and it finally proved that in facing such a giant convolution kernel, this algorithm still has quite a good effect.

Although there are still many flaws in the image, overall, they are already at a usable level for astronomical observations. Restoring such a highly damaged image to a usable level is already a difficult task. The PSNR and SSIM of these images are shown in Table 2:

After iteration, the algorithm increases the PSNR of the four dirty images by 41% on average, increases the SSIM by 33.9% on average. As a non-data-based traditional method, this is quite a good result for such a large complex convolution kernel.

To be honest, the solving process is time consuming when facing a large 256*256 convolution kernel, compared with smaller convolution kernel finished within 30 minutes, many hours were needed. But considering the effectiveness, it is worth to trade. We have not parallelized this algorithm, it should run way more faster on multicores or GPUs to save time. There are also many iteration tricks to speed up convergence. A good news is that the memory consumption is no more than 1G in facing this super large PSF. So, it is basically suitable for every modern desktop computer.

We also deconvolute the dirty image C with the Högbom CLEAN algorithm for comparison. The PSF setting also using 256*256 SKA PSF above, deconvolutes dirty image C with several adjustments. The best result is shown in Figure 9.

Figure 9(a) shows that all point sources have been found in dirty image C by Högbom CLEAN. Figure 9(b) shows all point sources have been found convoluted with the clean beam,

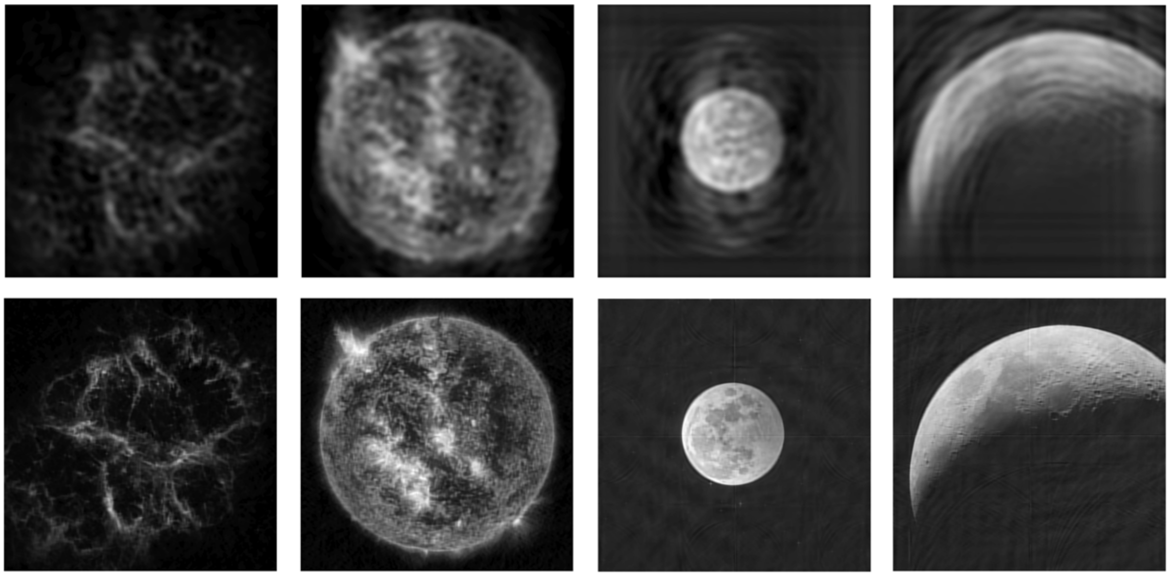

Figure 8. Convoluting these 4 images with this large radio interferometry convolution kernel and their deconvolution results.

Table 2
Deconvolution Performance of this Method for Radio Interferometry Imaging

Image	PSNR		SSIM	
	Blur	Deconvolution	Blur	Deconvolution
A	21.26	30.25	0.36	0.82
B	26.57	35.07	0.65	0.89
C	24.02	35.75	0.85	0.95
D	21.94	31.31	0.78	0.88

Table 3
Performance of this Method with 5% Pepper Noise

Image	PSNR		SSIM	
	Without noise	With noise	Without noise	With noise
A	42.41	42.26	0.97	0.98
B	38.34	38.18	0.97	0.97
C	45.02	45.34	0.99	0.99
D	37.1	37.23	0.95	0.96

which is a simple Gaussian function as the final result. To be honest, the CLEAN algorithm did make the dirty image look cleaner, there were less fuzzy waves surrounding the major moon pattern. But also, the generated image with clean beam does not very look like the original image compared with the algorithm that this paper introduced, the only thing correct may be the basic shape. It was indicated that in such a low contrast case, the Högbom CLEAN could not find the true sources correctly, because in many regions, the surrounding brightness is too similar to confuse the CLEAN algorithm. Thus this makes the final result lose almost all true details.

3.4. Reconstruction of Image Defects

The first experiment in this part was set up to add 5% salt-and-pepper noise to the previous four dirty images and use the four-direction combining scanning, then deconvolute these four dirty images. The results are shown in Figure 10. The PSNR and SSIM of these images are shown in Table 3.

It can be seen that this little salt and pepper noise has little effect on this deconvolution method based on single pixel iterations, and the peak signal-to-noise ratio and the structural similarity have almost not decreased. A closer look at the image shows a slight decrease in sharpness and resolution compared to the deconvolution results without salt and pepper noise, but it is not a major problem anyway.

If the area of the lost block in the image is relatively large and the total number of lost pixels is large, the algorithm will have different performances in different cases. For the convenience of the experiment, set the dirty image D losing the small blocks of pixels in the same size at an equal interval. There are four settings: a (3^*3 interval 6), b (10^*10 interval 20), c (6^*6 interval 8), and d (16^*16 interval 40). The pre-processing algorithm sets the pixel value of these missing parts to 1000. The deconvolution algorithm processed these blocked and dirty images, and the effect is shown in Figure 11. The PSNR and SSIM of these images are shown in Table 4, the rightmost of the table is the reference value restored without defects.

It can be seen that this missing block has a significant impact on the image deconvolution of this algorithm, after all, it loses too much information of brightness. But in any case, this is not

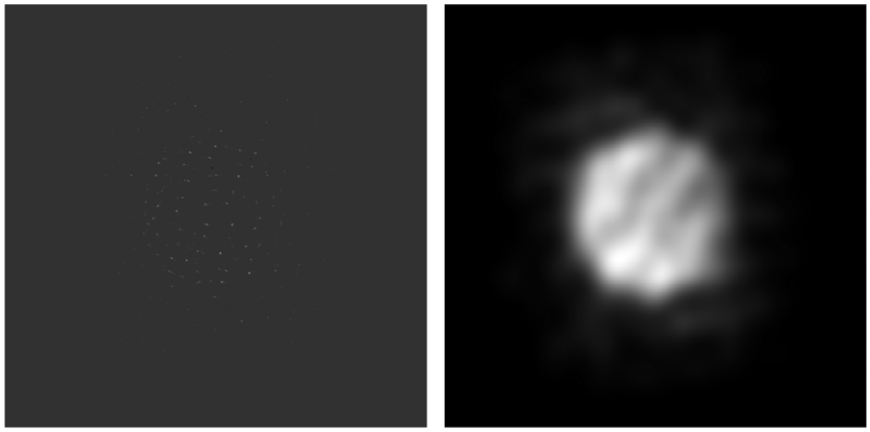

Figure 9. The deconvolution results of dirty image C with the Högbom CLEAN algorithm.

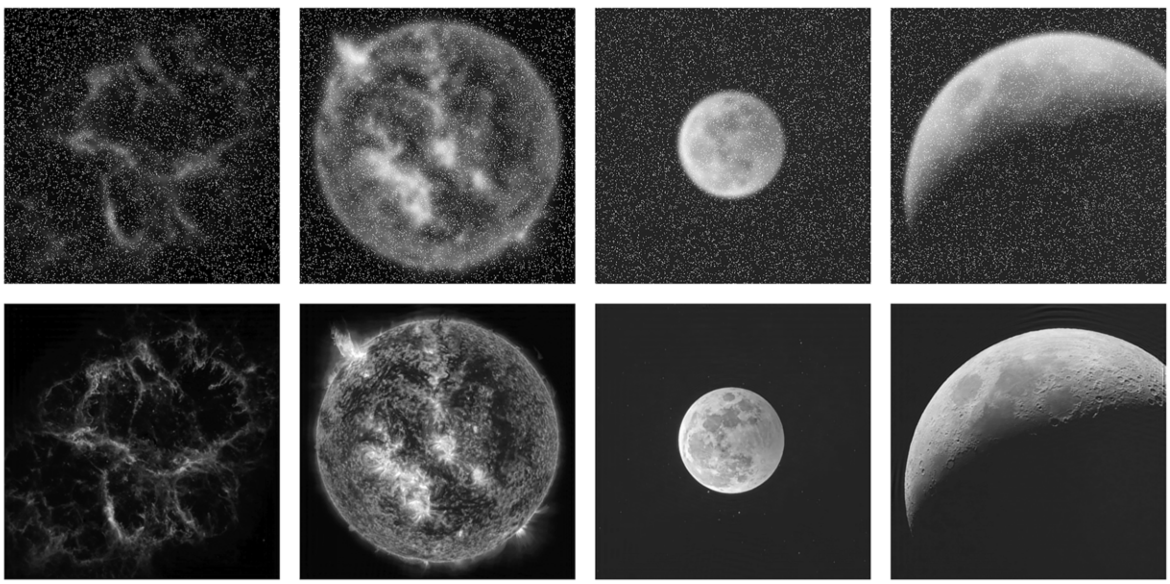

Figure 10. The situation that dirty images contain salt and pepper noise and their deconvolution results.

Table 4
Performance of this Method with Missing Parts in the Images

Image	a	b	c	d	n/a
PSNR	34.28	30.09	27.89	26.3	37.02
SSIM	0.9	0.8	0.74	0.79	0.95

enough to break the algorithm. Figure 12(a) shows, zooming into the image, it can be found that the algorithm successfully completes and restores the image as much as possible through the information retained in the slender around the lost parts, and finally presents a complete image with good details.

The program sets the convolution kernel of a certain size, and naturally, a convolution kernel generally has an outer area with very low values and irrelevance. If the lost area is too large and beyond the convolution kernel can retain its information, the restoration will fail. Figure 12(b) shows, if there are only sporadic losses in some small areas, the algorithm can completely solve and restore information within the surrounding pixels, and it seems that the lost areas basically have no impact, just like the case that these lost areas do not seem to exist, or this lost information is redundant.

The experiment proves that the algorithm can basically make full use of the information to restore the image and proves its

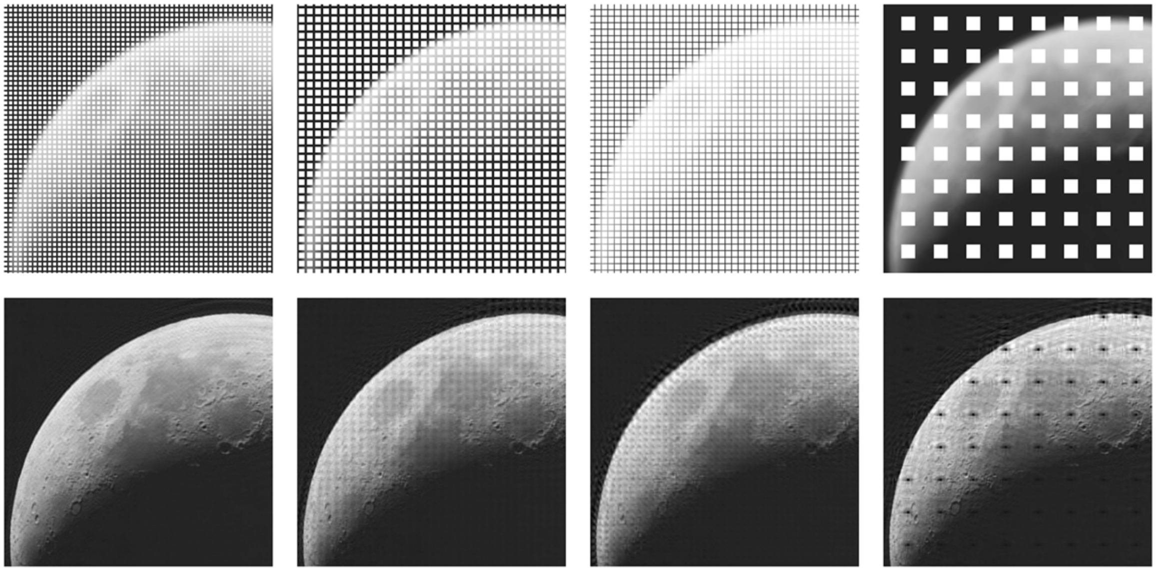

Figure 11. The situation that dirty images contain missing parts and their deconvolution results.

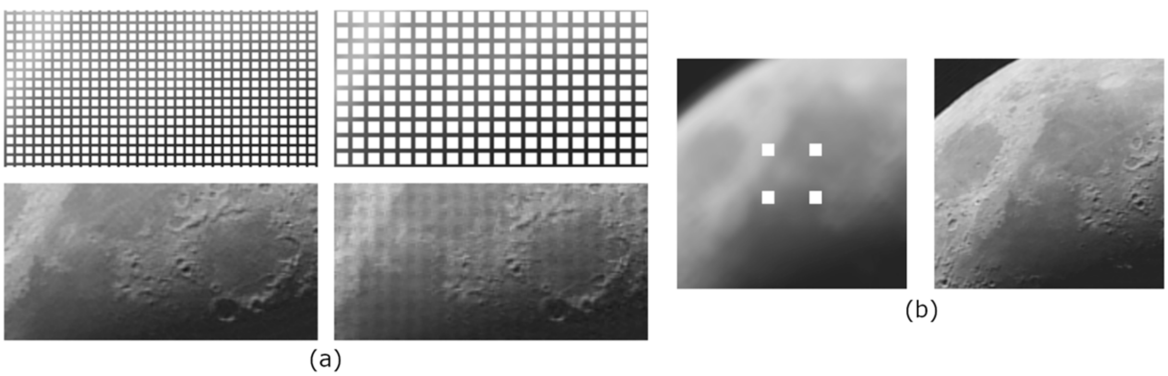

Figure 12. (a) Locally comparing the dirty image with the defective image and the restored image; (b) If information surrounding the missing parts is enough, after deconvolution these missing parts can be filled completely.

high efficiency in utilizing the information in the dirty image. Meanwhile, as a traditional image deconvolution method, it does not introduce false parts caused by data set selection like other data-based methods.

At the same time, this also brings an idea. When the deconvolution algorithm is mature enough, the convolution process can even be used in the reverse way. In other words, sometimes the convolution process can be used to disperse the light information and save it to many other locations around in the image, to make full use of the capacity of image sensors and get more redundancy.

4. Conclusion

To study the complex deconvolution problem of interferometry imaging in radio astronomy, this paper reanalyzed the process of general image deconvolution from the perspective of a dirty image, proposing that the point brightness in the dirty image will constrain the light distribution around to a hyperplane and point out that one way to image deconvolution is to solve the brightness linear equations.

This paper introduced the Kaczmarz method for solving linear equations, which is a type of iterative algorithms that step by step enter the hyperplanes according to their normal direction. It is especially suitable for these uniform equations such as image deconvolution. By adding different scan directions for iteration, the deconvolution results could be very good, the peak signal-to-noise ratio of the dirty images is improved by 43.2% on average, and the structural similarity is improved by 22.1% on average.

The scan iteration method of the algorithm is further improved, and the leapfrog scanning and random direction scanning are introduced, and the algorithm is applied to the baseline interferometry imaging deconvolution in radio astronomy. The real interferometry imaging will produce large complex convolution kernel, which causes strong damage to various structural details in astronomical images. After deconvolution by this improved algorithm, the image is restored to a good and recognizable degree, the peak signal-to-noise ratio is improved by 41.0% on average, and the structural similarity is improved by 33.9% on average.

At the same time, the experiments in this paper also verify that the deconvolution algorithm can make full use of the information in the dirty image to restore the image without destroying the image structure by setting conditions to skip the invalid pixels when there are defects in the image. Because the algorithm achieves pixel-level operation, it shows strong adaptability in dealing with practical problems.

Acknowledgments

The work was partially supported by the National Key R&D Program of China (No. 2022YFE0133700), the National Natural Science Foundation of China (NSFC, No. 12273007), the Guizhou Provincial Excellent Young Science and Technology Talent Program (No. YQK[2023]006), the National SKA Program of China (No. 2020SKA0110300), the NSFC (No. 11963003), the Guizhou Provincial Basic Research Program (Natural Science) (No. ZK[2022]143), the Cultivation project of Guizhou University (No. [2020]76).

References

Bhatnagar, S., & Cornwell, T. J. 2004, A&A, 426, 747
Dong, Z., Long, X., Chen, L., et al. 2017, in Int. Symp. Intelligent Sig. Proc. Commun. Syst. (ISPACS) (Xiamen, China: ISPACS)
Dai, W., Xu, T., & Wang, W. 2012, ITSP, 60, 6340
Eugene, H. 2021, Optics (5th Edn.) (London: Pearson)
Fish, D. A., Brinicombe, A. M., Pike, E. R., et al. 1995, JOSAA, 12, 58
Faulkner, A. J., & de Vaate, J. G. B. 2015, SKA low frequency aperture array in Int. Symp. Antennas and Propagation & USNC/URSI National Radio Science Meeting (Piscataway, NJ: IEEE), 1368
Gordon, R., Bender, R., & Herman, G. T. 1970, JThBi, 29, 471
Högbom, J.A. 1974, A&AS, 15, 417
Hu, Q., Hu, S., & Zhang, F. 2020, Signal Processing Image Communication, 83, 115758
Kaczmarz, S. 1937, Bulletin International de l'Academie Polonaise des Sciences et des Lettres A, 35, 355357
Lucy, L. B. 1974, AJ, 79, 745
Lu, C., Shi, J., & Jia, J. 2014, ITIP, 23, 837
Laasmaa, M., Vendelin, M., Peterson, P., et al. 2011, JMic, 243, 124
Needell, D. 2010, Numerical Mathematics, 50, 395403
Rafael, C., & Gonzalez, R. E. 2017, Digital Image Processing (4th Edn.) (London: Pearson)
Richardson, W. H. 1972, JOSA, 62, 55
Strohmer, T., & Vershynin, R. 2009, JFAA, 15, 262278
Welstead, S. T. 1999, Fractal and Wavelet Image Compression Techniques (Washington, DC: SPIE), 155
Wang, Z., Bovik, A. C., Sheikh, H. R., et al. 2004, ITIP, 13, 600
Yang, J., Wright, J., Huang, T. S., et al. 2010, ITIP, 19, 2861
Zhang, F., Cen, Y., Zhao, R., et al. 2017, Neurocomputing, 236, 32
Zhang, L., Wang, C. B., & Ye, L. 2018, A&A, 618, A165
Zhang, L., Bhatnagar, S., Rau, U., & Zhang, M. 2016, A&A, 592, 1
Zhang, L., Xu, L., & Zhang, M. 2020, PASP, 132, 041001
Zhang, L., Xu, L., Zhang, M., et al. 2019, RAA, 19, 79