https://doi.org/10.3311/PPme.41216 | 25
Creative Commons Attribution

**Periodica Polytechnica Mechanical Engineering**, 70(1), pp. 25–38, 2026

# Convolutional Methods in Image Enhancement

Imre Forgács[1*], Gábor Kovács[1]

[1] Department of Mechatronics, Optics and Mechanical Engineering Informatics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: forgacsimre@edu.bme.hu

Received: 09 June 2025, Accepted: 08 January 2026, Published online: 11 February 2026

**Abstract**
Convolution and deconvolution are essential in image processing for enhancement, analysis, and feature extraction. Convolution is widely used for filtering and edge detection, while deconvolution restores blurred images by recovering hidden details. These techniques are particularly important in space research, where high spatial resolution is coupled with 12–16-bit ADC, and optical quality is often degraded by stray light, making data interpretation challenging. Additionally, high-resolution images with large spatial dimensions and a high number of pixels, commonly encountered in space research, require significant computational resources, leading to slow processing times.

Our research focuses on optimizing convolution and deconvolution techniques using CUDA technology to accelerate processing. We developed a custom CUDA-based stray light removal method, achieving performance comparable to a previous C++ implementation while significantly reducing processing time through parallelization, resulting in an approximately 83% reduction in execution time and a runtime below half a second.

For deconvolution, we implemented multiple algorithms in MATLAB and CUDA environments, including Wiener filtering, Richardson–Lucy deconvolution, relevant regularization methods, and blind deconvolution. The Richardson–Lucy method, due to its iterative nature, is computationally intensive, which motivated its CUDA implementation. Leveraging GPU parallelization, we achieved substantial speed improvements – specifically, more than a 52% reduction in execution time – while maintaining result quality.

This paper proposes multiple deconvolution solutions for various image processing tasks and demonstrates the effectiveness and applicability of parallel programming in image enhancement algorithms. These contributions are particularly valuable for large-scale images and real-time applications.

**Keywords**
image enhancement, convolution, stray light correction, deconvolution, CUDA

## 1 Introduction

In today's increasingly digitized world, image processing has become one of the most dynamically evolving fields, finding applications in areas ranging from medical diagnostics to satellite image analysis. Each year, the focus on developing various procedures and methods in this domain intensifies, particularly due to the growing importance of space exploration. Enhancing image quality, extracting information, and reducing noise are fundamental tasks in image processing. Convolution and deconvolution play a central role in these processes.

Convolution is a fundamental mathematical operation that combines two functions to produce a third one, expressing how the shape of one function is modified by the other. In image processing, convolution is mostly applied as a discrete operation between an image and a filter or a Point Spread Function (PSF), enabling spatial filtering tasks such as smoothing, sharpening, edge detection, and feature extraction (Eq. (1)). Thus, convolution is widely used operation for filtering, edge detection, and feature enhancement [1].

In contrast, deconvolution is primarily used to sharpen blurred or distorted images, efficiently restoring the original information [2]. Deconvolution can be interpreted as the inverse operation of convolution, aiming to recover the original signal or image from a blurred observation. However, it is typically an ill-posed problem, where small perturbations such as noise can lead to significant reconstruction errors.

This is particularly critical in fields where image quality is impacted by numerous interfering factors, such as in space exploration or microscopic imaging. Nonetheless,

**Cite this article as:** Forgács, I., Kovács, G. "Convolutional Methods in Image Enhancement", Periodica Polytechnica Mechanical Engineering, 70(1), pp. 25–38, 2026. https://doi.org/10.3311/PPme.41216

implementing these methods presents several challenges. A major issue is the computational demand: processing high-resolution images through convolution and deconvolution requires significant resources and results in long processing times [3]. This is especially true for iterative deconvolution algorithms, such as the Richardson-Lucy (RL) method [4].

Another challenge involves handling image borders, where convolution and inverse convolution operations can introduce distortions, such as boundary effects, which refer to distortions or anomalies occurring at the edges of an image, like fringing effects – anomalies along sharp edges manifesting as color or intensity anomalies, stripes, or ghosting. Ringing artifacts, which appear as oscillating patterns, particularly around sharp edges, should also be noted [5, 6]. Additionally, removing stray light effectively is a complex task that requires specialized techniques [7]. Stray light refers to unwanted light reaching the imaging sensor through scattering, reflections, or diffraction within the optical system (e.g., Fig. 1 [8]), and its suppression is particularly critical in space missions, where extreme illumination conditions and high-contrast scenes can significantly degrade image quality and compromise scientific measurements (Fig. 2 [9]).

Therefore, our research presents potential solutions to these challenges, primarily approached from a software perspective. We considered the most widely used methods and algorithms during the writing of this paper, as they are the most relevant to the field, though not exhaustively. The methods chosen for presentation were selected to effectively demonstrate the characteristics of algorithms based on similar principles, providing a useful overview and addressing key challenges in related fields.

Our research focuses on the application and optimization of convolution and deconvolution techniques, with an emphasis on presenting these methods, improving their

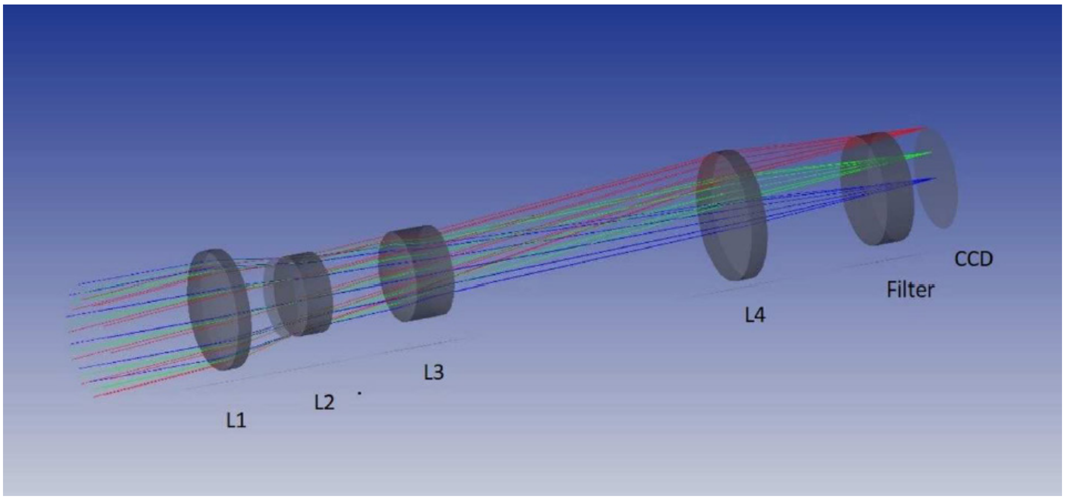


**Fig. 1** The Dawn FC optical system consists of four lenses (L1–L4) and a selectable bandpass filter [8]

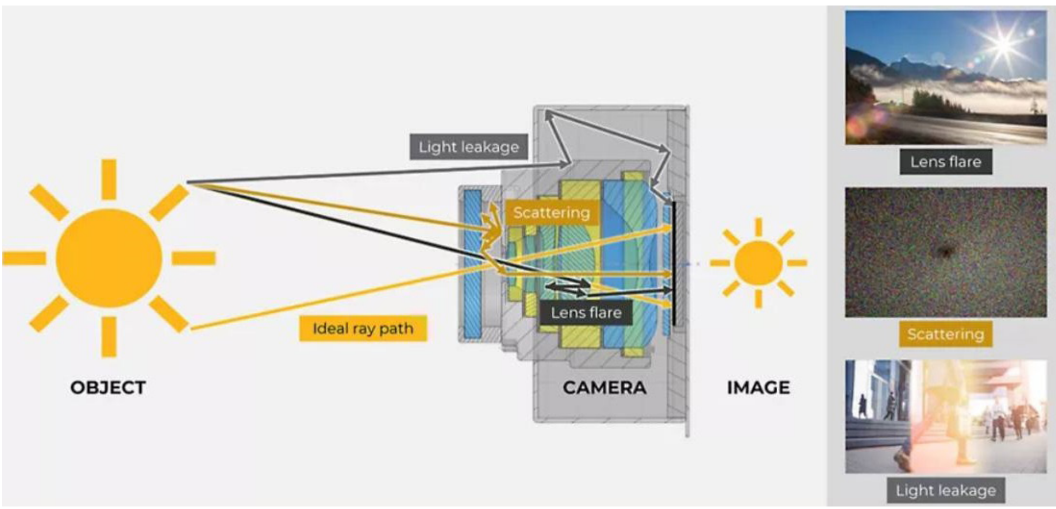


**Fig. 2** Sources and forms of stray light visually explained [9]

efficiency, comparing them, reducing processing times, and enhancing image quality – particularly for large images and time-critical applications, which is crucial across all areas of image enhancement and image processing in a smaller or a broader sense.

## 2 Stray light correction

Stray light introduces significant errors across all areas of image processing, but it poses particularly severe distortions in space research, where image sharpness is critical for achieving accurate results. Stray light manifests in multiple forms, each requiring different approaches for mitigation and correction; in this paper we present correctio for the in-field stray-light, where the unwanted additional image artifacts are generated by light sources inside the field-of-view. Our method is implemented in a Compute Unified Device Architecture (CUDA) environment, ensuring near-optimal processing time [10].

### 2.1 NASA's Dawn mission

The Dawn spacecraft was the ninth mission of NASA's Discovery program, launched on September 27, 2007, from the Kennedy Space Center [11]. The mission's primary objective was to study in detail the two largest objects in the asteroid belt: the protoplanet Vesta and the dwarf planet Ceres [11–13].

The first major milestone occurred on July 16, 2011, when Dawn entered orbit around Vesta and conducted a year-long close-range examination of the asteroid. The spacecraft determined that Vesta, like Earth, is a differentiated celestial body with an internal core, mantle, and crust. Additionally, it successfully mapped the asteroid's surface, characterizing surface features such as the Rheasilvia crater, a massive impact structure located at Vesta's south pole [14].

Following this phase, Dawn departed for its next target, Ceres, on August 5, 2012, reaching orbit around the dwarf planet in February 2015. The spacecraft identified bright spots on Ceres' surface, particularly within the Occator crater, which were later confirmed to be sodium carbonate deposits – indicating potential cryovolcanic (ice-volcanic) activity [11]. Furthermore, during the study of the smallest known dwarf planet in the Solar System, evidence was found for the presence of water ice and the possibility of subsurface oceans.

From the perspective of our research, the most crucial instrument on the spacecraft is undoubtedly its camera system, the Framing Camera (FC). Additionally, it is important to note that the optical calibration of the cameras was carried out by the Department of Mechatronics, Optics, and Mechanical Engineering Informatics at the Budapest University of Technology and Economics. Further details on this calibration can be found in [8, 15].

The FC system consists of two identical high-spatial-resolution optical imaging devices: FC1 and FC2 (Fig. 1). However, a total of four FC units were built for the Dawn mission: two flight units (FC1, FC2) and two backup units (FM1, FS2). Detailed information about the calibration of these cameras can be found in [8]. The two backup units were utilized during ground-based tests to better understand and characterize the stray light phenomenon known as "in-field stray light" within the camera's field of view. The camera optics were designed as a telecentric lens system, as illustrated in Fig. 1.

This design ensures that the principal ray exiting the final lens is perpendicular to both the filters and the Charge-Coupled Device (CCD), making the transmission bands of the filters' dichroic layers independent of the field-of-view position (Fig. 1). The optics are made of radiation-resistant glass and have been corrected for near-diffraction-limited performance. Further details about the camera system can be found in [16].

### 2.2 Stray light

Stray light is unwanted light that originates within the optical system through various mechanisms, degrading image quality. Similarly, in-field stray light refers to a specific case of this phenomenon, where light originates from objects within the optical system's field of view and reaches the detector due to scattering from optical surfaces or surface imperfections [8]. This issue has multiple sources, some of which are illustrated in Fig. 2. These sources include material defects and inhomogeneities within optical materials, diffraction effects, or surface scattering caused by microscopic contaminants or roughness on optical surfaces. Kovacs et al. [15] and van den Berg et al. [17] highlight internal reflections and ghost images as primary causes of stray light. Multiple reflections can result in ghost images, further degrading image quality. In the case of the Dawn spacecraft, the primary sources of in-field stray light are components located near the image plane of the cameras: the CCD detector, the bandpass filters, and the final lens that is denoted as L4 in Fig. 1.

Stray light increases the background brightness of the image, thereby reducing contrast (Fig. 2). Additionally, it introduces unwanted signals on the detector and complicates the precise calibration of the optical system, ultimately decreasing the reliability of measurements [9].

However, accurate calibration using stars is inherently challenging, as quasi-point light sources (the point spread functions, PSFs, of stars) do not always fully illuminate the photosensitive area of a pixel, leading to significant variations in the integrated flux response [8].

The Dawn mission carried two Framing Cameras onboard. Therefore, the images were acquired using two identical but physically separate camera units, each featuring similar optics and electronics. The CCD detector consists of 1024 × 1024 pixels, with each pixel measuring 14 × 14 μm. Since the detector surface lacks an optical coating, reflections (such as ghost images) occur within the camera due to high reflectivity. The most significant reflection arises between the CCD detector and the bandpass filters. When combined with the color filters, this results in substantial stray light contamination [15]. Local pixel correction methods proved insufficient for mitigating stray light effectively, as mineralogical analyses required spectral data [18].

It is important to note that several methods are commonly employed to manage and reduce stray light. For instance, the application of various anti-reflective and matte black coatings on appropriate optical surfaces minimizes unwanted reflections. Additionally, specialized structures such as baffles and light traps can be integrated into an optical system to block undesired light paths. Moreover, advanced software techniques exist for modeling and correcting stray light effects. These image processing approaches include convolutional and deconvolutional methods, typically applied during post-processing.

In the case of the Dawn spacecraft, an early correction method, developed after its approach to Vesta, initially addressed only the Narcissus ghost image caused by reflections from filter surfaces (Fig. 3). However, a later method was capable of handling additional sources of stray light [8]. This approach models the total stray light effect as a convolutional process, like the image formation itself, as ghost reflections are generated uniformly across all image regions.

## 3 Convolution-based correction method

Due to its numerous advantageous properties, convolution is one of the most widely used operations in image processing. Accordingly, the developed stray light correction algorithm is also based on convolution (Eq. (1)), similar to the method applied in the DAWN space probe. However, our approach is CUDA-based, ensuring optimal execution time and memory usage. This means that while the original algorithm runs in seconds, our solution provides at least equally accurate results in a fraction of a second.

Since execution time depends not only on the algorithmic design and software-level optimizations but also strongly on the underlying hardware, it is important to briefly describe the computational platform used in this work. The reported convolution and deconvolution results were obtained using hardware that can be considered average or even below average by current space research standards.

Specifically, all measurements and experiments were performed on a Lenovo Legion 5 laptop equipped with an AMD Ryzen 7 4800H CPU with Radeon Graphics, 32 GB of RAM, and an NVIDIA GeForce GTX 1660 Ti GPU with 6 GB of dedicated memory, running Windows 10. Regarding the software environment, MATLAB R2023a [19] and CUDA 11.8 [10] were used for the implementations. It should be noted that the applied methods and functions are also available in earlier and later versions of these software platforms.

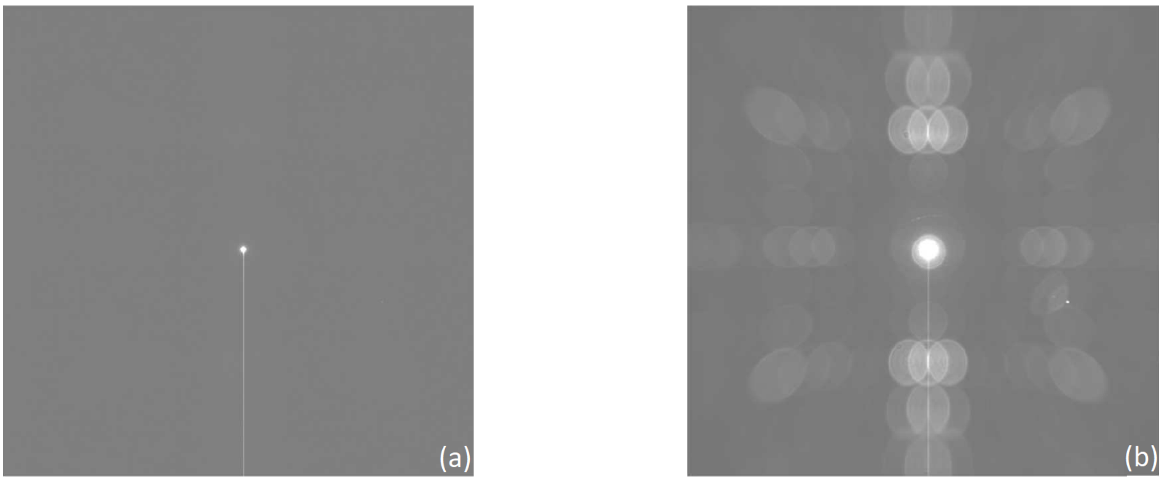


**Fig. 3** The Narcissus ghost image of a bright point light source with an exposure time of (a) 10 ms and (b) 1000 ms [8]

### 3.1 Convolutional method

Convolution is a fundamental mathematical operation that plays a crucial role in signal and image processing. In essence, convolution combines two functions – a given signal or image input and a filter (also referred to as a kernel) – to generate a third function. This operation enables various transformations of signals and images, such as filtering, sharpening, blurring, and edge detection [20, 21]. Convolution is a unique and powerful technique because it considers spatial relationships between pixels, which cannot be achieved through simple pointwise operations.

Mathematically, convolution is an operation that merges two functions to produce a third function. It is commutative, associative, and distributive over addition and can be defined for both functions and distributions [22, 23]. In general, convolution can be expressed as an integral over the interval $(-\infty, \infty)$, where the product of the functions $f$ and $g$ (both defined and integrable over $(-\infty, \infty)$) is evaluated. However, in modern practice, particularly in digital signal and image processing, most functions are discrete. Therefore, in such cases, the discrete version of convolution must be used (Eq. (1)), where summation replaces integration due to the discrete nature of the values:

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m) \cdot g(n-m), \quad (1)$$

where $f(m)$ is the discrete input, $g(n - m)$ the shifted discrete kernel, * is the convolution operator and $(f * g)(n)$ is the output at index $n$. The discrete convolution operation (Eq. (1)) depends on four key parameters:

1. Kernel size: the dimensions of the kernel, typically square (same width and height).
2. Stride: the step size by which the kernel moves from one computation to the next.
3. Dilation: the spacing between values within the kernel, affecting its receptive field.
4. Padding: the method used to fill empty values around the edges of the input (various techniques exist for this [24]).

In image processing, it is important to note that discrete convolution can be efficiently computed using the Fast Fourier Transform (FFT), which is implemented as a built-in function in several programming platforms (e.g., MATLAB [19], CUDA [10]).

Overall, convolution in image processing is the process of applying a kernel to each pixel in the input image, weighing its immediate neighbors accordingly. It is also crucial to emphasize that convolution is not traditional matrix multiplication and should not be confused with it. In the case of images, the two-dimensional version of discrete convolution is commonly used.

### 3.2 CUDA implementation and results

The original C++ algorithm performed computations using the Central Processing Unit (CPU) without leveraging the Graphics Processing Unit (GPU), resulting in execution times of several seconds. Therefore, the primary objective was to minimize the multi-second runtime while maintaining effective image enhancement.

For this work, we used 16-bit image files, and the algorithm was successfully tested with the images shown in Fig. 4. The first Flexible Image Transport System (FITS) file (Fig. 4 (a)) is an image captured by the DAWN spacecraft, depicting the asteroid 4 Vesta, with a resolution of 1024 × 1024 pixels. The second 16-bit FITS file (Fig. 4 (b)) is the corresponding kernel for the previous image – the point spread function (PSF) – which models

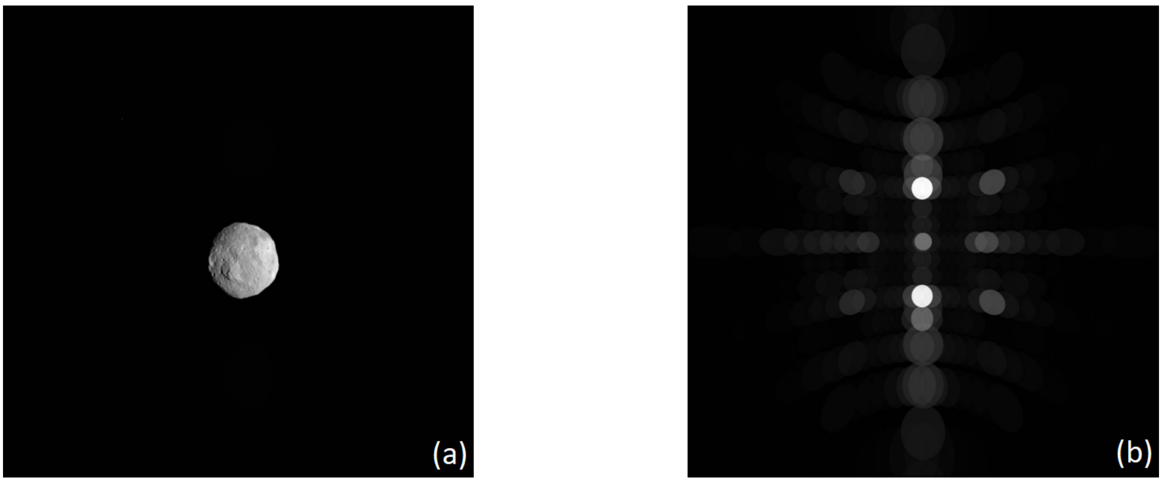


**Fig. 4** The original DAWN image of (a) Vesta, affected by stray light and (b) the kernel modeling the stray light

the distortions affecting the image, specifically the stray light. This kernel image is twice the size of the first one, measuring 2048 × 2048 pixels.

We performed the stray light removal by our custom developed CUDA-based C++ program, which implements an iterative convolution-based image enhancement procedure. This code is based on the single thread method presented in [8], which discusses the DAWN probe's camera system, and calibration. While the theoretical background of the two approaches is essentially identical and both rely on the same convolution-based principles, the primary differences arise from the implementation strategy and the computational environment. The original C++ algorithm was implemented as a single-threaded CPU-based solution without GPU acceleration, whereas the CUDA implementation for example, exploits massive parallelism on the GPU, leading to significantly improved computational efficiency. For further details regarding the implementation of the environment, the libraries used, and specifics of the code, feel free to contact us personally or via email.

We also developed a MATLAB [19] version of the algorithm, to enable a direct comparison between the two programs and to better evaluate our CUDA [10] implementation. This MATLAB [19] program employs the same method but without leveraging CUDA-specific optimizations. Additionally, it is important to highlight that while the classical MATLAB [19] code is likely slower due to the programming language and environment, it still provides a reasonable representation of the runtime differences. This is because the MATLAB [19] implementation, like the original algorithm, does not utilize CUDA [10], GPU acceleration, or parallel computing capabilities. Furthermore, both the CUDA [10] and MATLAB [19] implementations achieve similar effectiveness in removing stray light, as demonstrated in the comparison between Figs. 5 and 6.

In this case as well – as for all image-based and runtime results presented throughout the paper – we report and illustrate the best achieved outcomes. Accordingly, Fig. 5 (b) presents the best result – final processed image – obtained using the CUDA-based implementation, while Fig. 6 (b) shows the best result achieved with the MATLAB [19] implementation. In both cases, the optimal result was reached after the first iteration, which was expected given the characteristics of the applied method and the properties of the input data [8].

The scales in Figs. 5 and 6 represent the unitless pixel intensity values, where 0 corresponds to the darkest areas and 1 to the brightest regions. In Figs. 5 and 6, the CLAHE filter was applied, enhancing the visualization and making it clearer that identical image quality results can be achieved using both platforms.

In terms of image quality, the best results were achieved after the first iteration. The computational performance was also significantly improved alongside image enhancement. While the original C++ code had a runtime of several seconds, the CUDA [10] implementation successfully reduced the execution time of the stray light removal algorithm to below half a second (Fig. 7).

More precisely, with just a single iteration, the original C++ code required *several seconds* to complete, whereas the CUDA [10] implementation achieved a runtime of

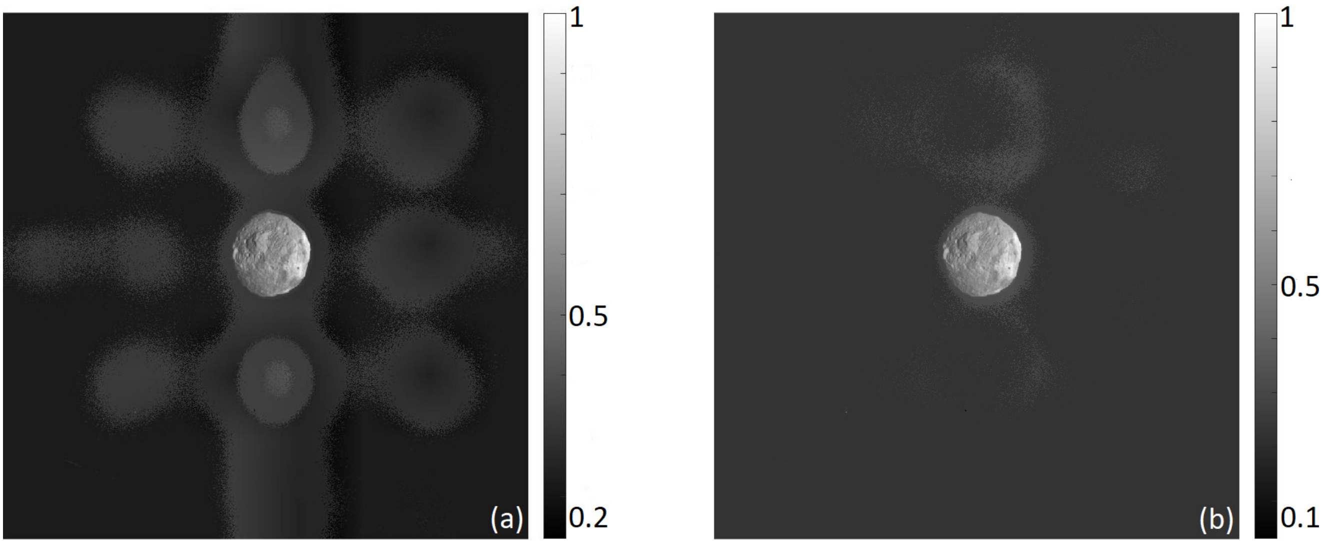


**Fig. 5** (a) The original image and (b) the result after the first iteration with CUDA [10]

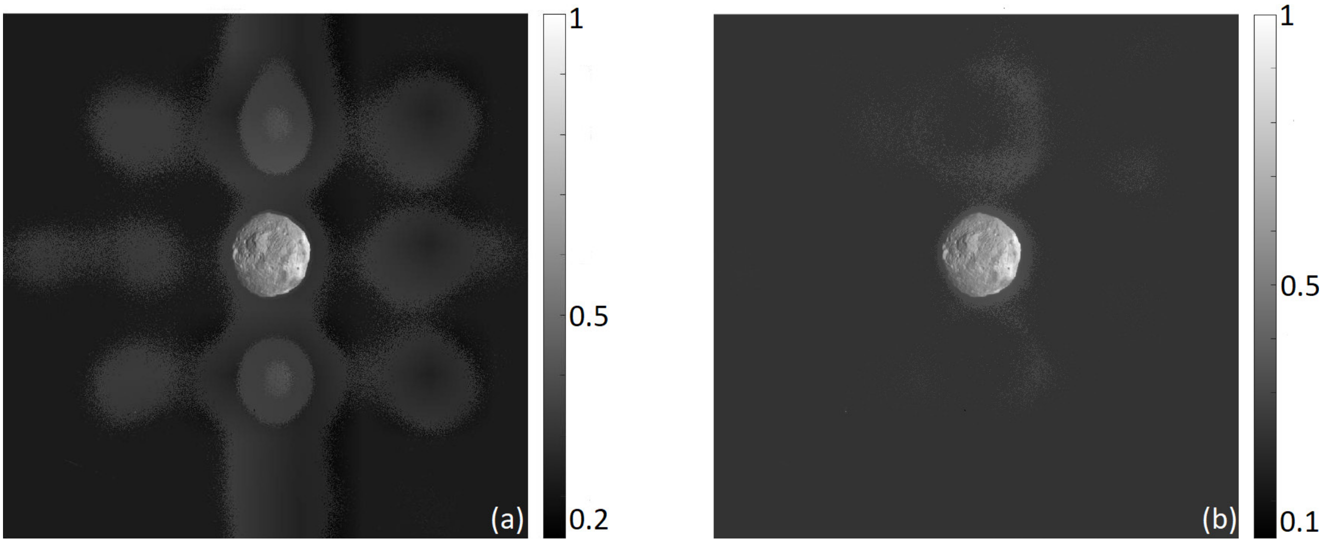


**Fig. 6** (a) The original image and (b) the result achieved with MATLAB [19] after one iteration

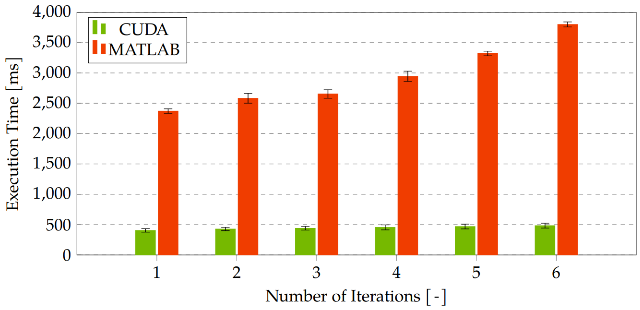


**Fig. 7** Execution time of the CUDA [10] and MATLAB [19] code in each iteration, with minimum and maximum values

404.791 ms. In comparison, the MATLAB [19] implementation required 2371.757 ms (= 2.37 s) for one iteration.

Fig. 7 illustrates the timing results, showing the execution times of the MATLAB [19] and CUDA [10] codes over the first six iterations. For each iteration, 20 measurements were taken, and the average execution time was plotted. The error bars on the columns represent the extreme values recorded in each iteration.

Each additional iteration increased the runtime by approximately 10–20 ms for the CUDA [10] implementation, whereas for the MATLAB [19] implementation, this increase ranged from 150 to 300 ms per iteration.

## 4 Deconvolution

Imaging is a crucial field in both science and industry, as various imaging techniques provide access to information that would otherwise be unattainable. However, acquired image data not only contains the desired information but also a combination of distortions introduced by the optical system's imaging and detection properties. Additionally, these two-dimensional images capture details from both the in-focus plane and out-of-focus regions [25]. This can result in a low signal-to-background ratio due to blurred contributions from defocused areas.

The Signal-to-Noise Ratio (SNR) quantifies the strength of the desired signal relative to background noise (undesired signal). Various approaches aim to minimize the impact of out-of-focus regions in the final image, yet they cannot fully compensate for distortions introduced by optical systems, such as aberrations. Based on this concept, an image can be considered as a combination of ideal information from the sample and distortions inherent in the optical system. If these distortions can be mathematically described, they can also be mitigated using appropriate methods. Deconvolution-based image processing is designed to address this challenge [26–28].

### 4.1 Instruments and images used for deconvolution

The images and instruments used for the convolution-based methods were introduced previously in Sections 2.1 and 3.2. In Section 4.1, the imaging data and instruments applied for the deconvolution experiments are briefly described. Although convolution and deconvolution techniques differ in both purpose and implementation, all methods presented in this study were implemented on the same hardware platform and under similar computational conditions and environments.

For the motion blur experiments, an older consumer-grade digital camera, the Panasonic Lumix DMC-F2, was used. This camera was selected intentionally, as older imaging devices typically lack advanced built-in image stabilization and post-processing algorithms, making them more suitable for capturing motion-blurred images.

The Panasonic Lumix DMC-F2 features a 10-megapixel CCD sensor (1/2.5″) with 4× optical zoom. The motion-blurred test image shown in Fig. 8 (a) was acquired using this camera and stored as a 3648 × 2736 pixels JPEG file. Additional technical specifications of the camera can be found in the manufacturer's user manual [29].

To demonstrate defocus blur and to present the corresponding deconvolution results, an image acquired by the Asteroid Framing Camera onboard ESA's Hera mission (AFC/Hera) was used. The image is a 2048 × 2048 pixels FITS file depicting a chessboard-patterned human figure (Fig. 9). The AFC instrument, developed by Jena-Optronik (Germany), has a mass of approximately 1.3 kg and dimensions comparable to a household vase [30]. Since defocus blur is a frequent challenge in space imaging, the results obtained in this work may also be beneficial for other cameras of different missions or other instruments of the Hera mission, such as ASPECT and HyperScout-H, which are complementary hyperspectral imaging systems.

### 4.2 Different types of deconvolution methods

The general deconvolution problem (Eq. (2)) can be expressed as the convolution product of two functions, where the first factor, $f$, represents the initial "clean" signal we aim to recover. The second factor, $g$, is the signal or kernel (PSF in imaging) that causes the distortion of $f$. The result, $h$, is the fixed, already distorted signal. In other words, $f$ is the signal that has been convolved with $g$ before detection, and $h$ is its distorted version, which is difficult to recognize in its original form. All deconvolution techniques – discussed in image processing – aim to provide a solution to Eq. (2) with different approaches:

$$f * g = h. \tag{2}$$

The simplest solution is to reverse the process – applying the inverse of the operation that caused the distortion, i.e., the general inverse deconvolution, since mathematically, deconvolution is the inverse of convolution [31]. Both operations are widely used in signal and image processing.

From a mathematical perspective, convolution is a well-defined, stable operation, while its inverse is, like most inverse problems, ill-posed [32]. This refers to expressions that do not satisfy the three Hadamard criteria for well-posed problems:

1. existence of a solution;
2. uniqueness of the solution;
3. continuity of the solution with respect to parameters or input data.

In practice, this means that most inverse methods, due to their mathematical instability, are highly sensitive to noise and small errors in the input. However, the procedure can be refined by adding a new factor: general noise. This approach will yield more accurate results, as assuming a noisy image is noise-free leads to erroneous estimates of $f$ and $g$. The lower the signal-to-noise ratio, the worse the result. This is why simple inverse convolution often fails to provide satisfactory results [28, 31, 32]. As a result, various deconvolution techniques exist, each approaching the task differently [5, 33].

### 4.3 Regularization methods and Wiener filtering

To perform deconvolution more efficiently, it is necessary to account for noise and disturbances affecting the system. This requires modifying the general inverse convolution approach, as a numerically more stable solution is needed than simply taking the ratio of Fourier transforms. Therefore, some form of regularization must be applied to yield a more natural-looking solution [32–34].


**Fig. 8** Motion blur in a real-world environment: (a) The original image and (b) the corrected image

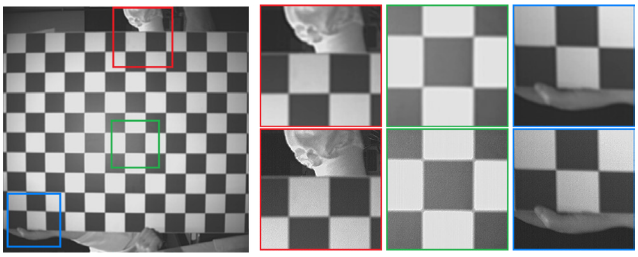

**Fig. 9** On the left is the original defocused image, and on the right, the top row shows cropped scenes from the original. Below these blurred sections are details from the Wiener deconvolution results, color-coded accordingly

The primary issue is that when the denominator, i.e., the $g$ kernel and its Fourier transform, take very small values, inaccurate or erroneous results may arise. In some cases, no solution is obtained at all dues to values close to zero in the kernel. A potential solution is to add a small constant to the denominator to avoid division by zero. However, since we are dealing with complex numbers that cannot be easily ordered, determining the appropriate value to ensure it is greater than zero is not straightforward. The regularization constant addresses this challenge, but it introduces both strengths and weaknesses. The choice of this value significantly influences the result's quality. A value that is too large may overly smooth the solution, losing important details, while a value that is too small can amplify the noise.

### 4.4 Iterative methods

Iterative deconvolution methods encompass a variety of algorithms aimed at image sharpening or restoring original information, providing an iterative solution to the inverse convolution problem. These methods are characterized by using iterations to refine the solution step by step, gradually approaching the correct result [33, 34]. Algorithms can be categorized in various ways, such as by what is refined at each step. Some use the Point Spread Function (PSF), while most minimize an error or cost function over multiple iterations. Possible categorization criteria include:

- applied regularization;
- mathematical approach;
- PSF knowledge and type;
- noise knowledge and type.

These methods are particularly useful for nonlinear problems or when the PSF is only partially or not at all known. However, like all deconvolution methods, iterative algorithms have limitations, which should be considered before choosing the appropriate method for a given task. Key considerations include:

- Convergence: these algorithms do not always guarantee finding the global minimum. They may not converge stably and could settle on local minima, a common issue with many search algorithms.
- Noise: excessive iterations or improper regularization may amplify undesirable noise or distortions, such as fringing.
- Computational demand: due to the iterative nature, these methods are more computationally intensive than others, requiring more steps to reach the final solution. The more iterations, the more time-consuming the process, especially for high-resolution images.
- Initial estimate: a proper starting estimate is crucial as it greatly influences the entire process. For example, using a smooth Gaussian kernel as an initial PSF for a motion-blurred image will lead to poorer results than using a motion blur kernel.

Despite these challenges, iterative deconvolution methods are widely used in various industrial and scientific fields due to their effectiveness for suitable tasks.

### 4.5 Blind deconvolution

In engineering and applied mathematics, blind deconvolution refers to a technique where the impulse response function used in convolution is not explicitly known. This task is typically approached by making assumptions about the input and estimating the impulse response function through analysis of the resulting data. Most blind deconvolution algorithms assume that both the input and impulse response function reside within known, suitable domains. Despite such assumptions, blind deconvolution remains a challenging multivariable, non-convex optimization problem [33, 35].

In image processing terms, blind deconvolution is a technique that enables the restoration of a distorted image, given the blurred (distorted) image itself. A key distinction is that while linear and nonlinear deconvolution techniques work with known PSFs, blind deconvolution estimates PSF using one or more blurred images.

This method has many variants and applications. However, all blind deconvolution approaches aim to provide the best solution to the inverse convolution problem, where the kernel ($g$) is unknown. In this case, only the observed blurred image ($h$) is available, while both the kernel ($g$) and noise are unknown. The goal is to restore the original sharp image ($f$) and determine the PSF ($g$), using only the recorded blurred image ($h$).

It is noteworthy that blind deconvolution is one of the most researched areas of deconvolution methods, as in engineering practice, the blurred image is often the only available data, with reliable information only about it.

## 5 Results and CUDA implementation

The deconvolution algorithms were implemented in both MATLAB [19] and CUDA [10] environments, like the convolution algorithm for stray light removal. This provided a solid reference point for evaluation, enabling the development of more efficient solutions to the deconvolution problem. Among the algorithms implemented were Wiener deconvolution, general and Tikhonov regularization, Richardson-Lucy deconvolution, and blind deconvolution, to name a few.

We addressed a variety of distortions and aberrations, with a primary focus on motion blur and defocus, as these are the most prevalent image artifacts in many application areas, particularly in space research.

The CUDA [10] and MATLAB [19] algorithms demonstrated significant improvements in both image enhancement and runtime efficiency. These algorithms can, of course, be fine-tuned for specific tasks to optimize their performance. For example, spatially dependent distortions can be handled through local deconvolution with minimal adjustments.

As the research focused on real-world images with actual distortions, we evaluated image quality improvements using objective metrics like PSNR (Peak Signal-to-Noise Ratio) and MSE (Mean Squared Error). However, subjective assessments were ultimately prioritized, as the end users of these images are people viewing the results, with no further processing or use in additional algorithms. Thus, subjective evaluation methods were deemed most appropriate for this context.

### 5.1 Motion blur

Motion blur is a phenomenon where images become blurred due to the movement of the camera or the observed object during the exposure time (Fig. 8). This distortion occurs when the captured image changes during a single exposure, caused by fast motion or a long exposure time [36]. The blur appears as streaks [37], and it is particularly noticeable for point-like objects or light sources, such as stars or lamps. On a blurred image, these objects manifest as streaks in the direction of motion, which can be used to approximate the motion blur point spread function (PSF) affecting the entire image.

The linear model is most used to model motion blur, assuming uniform linear motion and corresponding uniform blur. The kernel can be easily constructed once the length and angle of the motion are known. However, in practice, these values, like the PSF, can only be estimated. The tested and refined algorithms were also evaluated in real-world conditions. Fig. 8 (a) was captured with my very old digital camera (Panasonic Lumix DMC-F2) by increasing the exposure time and moving the camera evenly from left to right during the shot. Various kernels and parameter settings were tested for each method. The best result was achieved using general regularization deconvolution (Fig. 8 (b)), shown alongside the original image in Fig. 8. The second-best result was obtained with Wiener deconvolution.

### 5.2 Defocus blur

Defocus is an optical aberration occurring when an image is simply out of focus. It is one of the most common optical errors encountered by users of cameras, video cameras, microscopes, or telescopes. Consequently, most imaging devices include focusing mechanisms to minimize defocus and maximize image quality. Optically, defocus refers to the shift of focus along the optical axis away from the detection surface [38, 39]. In the human eye, defocus blur appears as monocular visual cues, with objects outside the depth of field appearing blurred on the retina [40]. Generally, defocus reduces sharpness and contrast, turning initially sharp, high-contrast edges into gradual transitions, with fine image details becoming barely visible or invisible. Several factors can cause defocus, including lens aberrations, limited depth of field, or varying object distances. This optical error is common in microscopy, astronomy, and medical imaging, and must be considered across all imaging fields. The mathematics of defocus can, like motion blur, be precisely described using convolution operations.

After testing and refinement, the focus was placed on real images with actual imaging errors and distortions. We used images captured by the HERA spacecraft's AFC camera, which are affected by defocus. One such image, taken with the AFC camera of the HERA spacecraft, shows a blurred chessboard pattern in a laboratory (Fig. 9). The defocus in this image arises from the camera calibration and the environment, as the spacecraft cameras were not calibrated for Earth's lab environment and distances.

For the AFC camera image (Fig. 9), the best results were achieved using appropriately parameterized Wiener deconvolution. However, similar image enhancement was achieved with regularization and Richardson-Lucy deconvolution, although the latter two methods resulted in images with slightly more noticeable noise. These minor differences are likely due to imperfect parameterization rather than the methods themselves. Fig. 9 highlights the results for the AFC camera defocus. On the left is the original image captured with the AFC camera. The top row shows cropped details from the original image, while the bottom row presents cropped scenes from the Wiener deconvolution results, color-coded accordingly.

As shown in Fig. 9, a simple but well-parameterized deconvolution technique can achieve remarkably strong image enhancement results, even in the case of space-borne camera systems. The improvement is most clearly observable in the regions highlighted by the red frames, particularly around the mask, while a significant enhancement is also visible within the blue-framed area when focusing on the thumb. Furthermore, all cropped scenes demonstrate a successful sharpening of the checkerboard pattern, indicating improved edge transitions between black and white regions. In practice, this can lead to more accurate edge detection, which is beneficial in engineering applications as well as in applied optics and image processing. For instance, improved edge definition supports more reliable calibration procedures and therefore enables the acquisition of higher-quality, sharper images across a wide range of applications.

Overall, defocus blur is a common issue affecting image sharpness across various industrial and scientific fields involving optical systems and cameras. However, modeling defocus with a circular kernel *via* convolution enables the use of deconvolution techniques, which effectively improve any blur and optical aberration modeled by convolution operations.

### 5.3 Time and CUDA implementation

The techniques presented and described here, when applied correctly, yield good image results. However, it is important to highlight that there can be significant variations in runtime between different deconvolution methods, particularly for iterative procedures. In case of our implemented algorithms, the Richardson-Lucy (RL) and the blind deconvolution methods are based on iterations, and in general performed less efficiently considering the runtime. In modern image enhancement, the speed at which an algorithm returns results is becoming increasingly critical, especially for real-time applications.

Similarly to the convolution experiments, all deconvolution measurements and runtime evaluations were performed using the same hardware and software environment described in Section 3. The deconvolution algorithms were implemented using the same MATLAB [19] and CUDA [10] framework, following an approach comparable to that applied for convolution, without relying on CUDA-specific parallel optimization features within MATLAB [19].

Unlike the convolution case, multiple deconvolution algorithms were implemented across the examined platforms, and their detailed descriptions are therefore provided in the corresponding Sections 4.3–4.5 with appropriate references. In general, the implementation workflow of the deconvolution methods followed a common structure: loading the input image, optional preprocessing depending on the applied algorithm, definition or estimation of the kernel (the PSF that causes the distortion of the input image), execution of the deconvolution procedure, and finally the generation of the output image followed by result evaluation.

Corresponding to the represented convolutional method, we measured and compiled statistics for the runtime. The image quality results are not discussed here, as similar improvements can be achieved using the same solutions, as shown in the convolution section. Therefore, the focus of this article is the representation of the improvement of runtime performance.

It is important to note the results in the graph (Fig. 10), where both algorithms provided the best results after the 16th iteration for the chessboard image (Fig. 9). MATLAB [19] took 13.55 s, while the CUDA [10] code completed under 6.41 s. Additionally, the results in Fig. 10 show nearly linear growth for both cases, with CUDA [10] time values increasing at a slower rate than MATLAB [19],

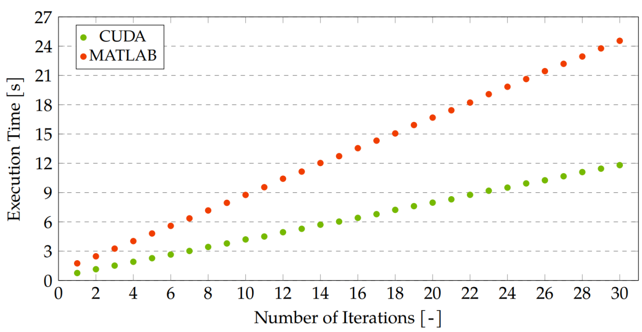


**Fig. 10** Execution time of CUDA [10] and MATLAB [19] codes per iteration for the image captured with the AFC camera

as expected. Thus, the graph in Fig. 10 illustrates the measured runtime for both CUDA [10] and MATLAB [19] in processing the real defocused image from the AFC camera.

## 6 Summary

In today's increasingly digital world, image processing is one of the most rapidly evolving fields, with numerous industrial and scientific applications, including robotics, medicine, manufacturing, automotive, and space exploration. Modern image processing aims not only to extract and restore information effectively but also to enhance images efficiently and quickly.

Convolution is widely used for filtering, edge detection, and feature extraction, allowing image sharpening, blurring, and the application of specialized filters to highlight important details or remove noise. Conversely, deconvolution is primarily used to restore sharpness in blurred or distorted images, recovering the original information as accurately as possible. However, both methods face challenges such as reducing computational time, managing noise sensitivity and edge artifacts, and improving image restoration quality. Therefore, optimizing parameters and the point spread function (PSF) is crucial to enhance images while minimizing noise without significant information loss. Fast processing is particularly critical in modern applications, including space probes, autonomous vehicles, satellites, and robotic systems, where high-resolution images and complex computations are required.

In this paper, we reviewed and compared widely used convolution- and deconvolution-based image enhancement techniques in terms of runtime and restoration quality. Our optimizations significantly reduced processing times, improving efficiency. Notably, CUDA [10] acceleration effectively sped up the Richardson-Lucy deconvolution algorithm while maintaining image quality. CUDA's [10] parallel computing capabilities substantially reduce runtimes, optimizing resource utilization. More specifically, the runtime of the presented convolution-based method was reduced by approximately 83%, while for the implemented deconvolution-based methods a reduction of more than 52% in execution time was achieved. This enables faster execution of complex algorithms (e.g., deconvolution), efficient processing of large datasets, and improved performance in real-time applications. Moreover, considering the achieved image quality improvements – particularly the results obtained using images acquired by the AFC/Hera instrument – the presented methods can serve as a solid basis for further space research applications, including image processing pipelines, camera calibration procedures, and image enhancement tasks in future missions.

## References

[1] Nowinski, W. "The speedup of parallel image reconstruction", Computers and Artificial Intelligence, 12(4), pp. 337–344, 1993.

[2] Makarkin, M., Bratashov, D. "State-of-the-Art Approaches for Image Deconvolution Problems, including Modern Deep Learning Architectures", Micromachines, 12(12), 1558, 2021. https://doi.org/10.3390/mi12121558

[3] Demoment, G. "Image reconstruction and restoration: Overview of common estimation structures and problems", IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(12), pp. 2024–2036, 1989. https://doi.org/10.1109/29.45551

[4] Prato, M., Cavicchioli, R., Zanni, L., Boccacci, P., Bertero, M. "Efficient deconvolution methods for astronomical imaging: algorithms and IDL-GPU codes", Astronomy & Astrophysics, 539, A133, 2012. https://doi.org/10.1051/0004-6361/201118681

[5] Lee, J.-H., Ho, Y.-S. "High-quality non-blind image deconvolution with adaptive regularization", Journal of Visual Communication and Image Representation, 22(7), pp. 653–663, 2011. https://doi.org/10.1016/j.jvcir.2011.07.010

[6] La Camera, A., Schreiber, L., Diolaiti, E., Boccacci, P., Bertero, M., Bellazzini, M., Ciliegi, P. "A method for space-variant deblurring with application to adaptive optics imaging in astronomy", Astronomy & Astrophysics, 579, A1, 2015. https://doi.org/10.1051/0004-6361/201525610

[7] Saranathan, S., van Noort, M., Solanki, S. K. "Correction of atmospheric stray light in restored slit spectra", Astronomy & Astrophysics, 653, A17, 2021. https://doi.org/10.1051/0004-6361/201937100

[8] Kovács, G., Nathues, A., Sierks, H., Gutiérrez Marqués, P., Hoffmann, M., Thangjam, G. S. "The Scientific Calibration of the Dawn Framing Camera", Space Science Reviews, 220(1), 4, 2024. https://doi.org/10.1007/s11214-023-01039-w

[9] Ansys "What is Stray Light?", [online] Available at: https://www.ansys.com/simulation-topics/what-is-stray-light [Accessed: 23 February 2025]

[10] NVIDIA Corporation "CUDA Toolkit, (11.8)", [computer program] Available at: https://developer.nvidia.com/cuda-11-8-0-download-archive [Accessed: 24 February 2025]

[11] Russell, C. T., Raymond, C. A. "The Dawn Mission to Vesta and Ceres", In: The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres, Springer, 2012, pp. 3–23. ISBN 978-1-4614-4903-4 https://doi.org/10.1007/978-1-4614-4903-4_2

[12] Russell, C. T., Coradini, A., Christensen, U., De Sanctis, M. C., Feldman, W. C., …, Zuber, M. T. "Dawn: A journey in space and time", Planetary and Space Science, 52(5–6), pp. 465–489, 2004. https://doi.org/10.1016/j.pss.2003.06.013

[13] Russell, C. T., Capaccioni, F., Coradini, A., De Sanctis, M. C., Feldman, W. C., …, Zuber, M. T. "Dawn Mission to Vesta and Ceres: Symbiosis between Terrestrial Observations and Robotic Exploration", Earth, Moon, and Planets, 101(1), pp. 65–91, 2007. https://doi.org/10.1007/s11038-007-9151-9

[14] Russell, C. T., Raymond, C. A, Jaumann, R., McSween, H. Y., De Sanctis, M. C., …, Joy, S. P. "Dawn completes its mission at 4 Vesta", Meteoritics & Planetary Science, 48(11), pp. 2076–2089, 2013. https://doi.org/10.1111/maps.12091

[15] Kovacs, G., Sierks, H., Nathues, A., Richards, M., Gutiérrez-Marques, P. "Stray light calibration of the Dawn Framing Camera", In: Proceedings of Sensors, Systems, and Next-Generation Satellites XVII, Dresden, Germany, 2013, 888912. ISBN 9780819497581 https://doi.org/10.1117/12.2030584

[16] Sierks, H., Keller, H. U., Jaumann, R., Michalik, H., Behnke, T., …, Tschentscher M. "The Dawn Framing Camera", Space Science Reviews, 163(1), pp. 263–327, 2011. https://doi.org/10.1007/s11214-011-9745-4

[17] van den Berg, T. J. T. P., Franssen, L., Kruijt, B., Coppens, J. E. "History of ocular straylight measurement: a review", Zeitschrift für Medizinische Physik, 23(1), pp. 6–20, 2013. https://doi.org/10.1016/j.zemedi.2012.10.009

[18] Schröder, S. E., Maue, T., Gutiérrez Marqués, P., Mottola, S., Aye, K. M., Sierks, H., Keller, H. U., Nathues, A. "In-flight calibration of the Dawn Framing Camera", Icarus, 226(2), pp. 1304–1317, 2013. https://doi.org/10.1016/j.icarus.2013.07.036

[19] The MathWorks, Inc. "MATLAB, (R2023a)", [computer program] Available at: https://www.mathworks.com/products/new_products/release2023a.html [Accessed: 24 February 2025]

[20] Jain, A. K. "Fundamentals of Digital Image Processing", Prentice-Hall, Inc., 1989. ISBN 0-13-336165-9 [online] Available at: https://www.academia.edu/39878253/Anil_K_Jain_Fundamentals_of_Digital_Image_Processing [Accessed: 23 February 2025]

[21] Rosenfeld, A., Kak, A. C. "Digital Picture Processing", Academic Press, 1976. ISBN 0-12-597360-8 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=ANiNDAAAQBAJ&oi=fnd&pg=PP1&dq=Rosenfeld,+A.+%22Digital+picture+processing%22,+Academic+Press,+1976.+ISBN+978-0120437014&ots=vi-AqkjqxQb&sig=Pc6HwKPjfTIs_pGJOqZwycBiURg&redir_esc=y#v=onepage&q&f=false [Accessed: 23 February 2025]

[22] Dimovski, I. H. "Convolutional Calculus", Kluwer Academic Publishers, 1990. ISBN 978-94-010-6723-2 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=G4LpCAAAQBAJ&oi=fnd&pg=PR9&dq=Dimovski,+I.+H.+%22Convolutional+calculus%22&ots=MMpZAQquAU&sig=LhcfcqQ63k7c_IGHROfSvfEdIyg&redir_esc=y#v=onepage&q=Dimovski%2C%20I.%20H.%20%22Convolutional%20calculus%22&f=false [Accessed: 23 February 2025]

[23] Hirschman, I. I., Widder, D. V. "The convolution transform", Courier Corporation, 2012. ISBN 9780486154565

[24] Rik Voorhaar "Blind Deconvolution #3: More about non-blind deconvolution", [online] Available at: https://www.rikvoorhaar.com/blog/deconvolution_part3 [Accessed: 23 February 2025]

[25] Teledyne Vision Solutions "Image Restoration Through Deconvolution", [online] Available at: https://www.teledynevisionsolutions.com/learn/learning-center/scientific-imaging/image-restoration-through-deconvolution/ [Accessed: 23 February 2025]

[26] Rik Voorhaar "Blind Deconvolution #1: Non-blind Deconvolution", [online] Available at: https://rikvoorhaar.com/blog/deconvolution_part1 [Accessed: 23 February 2025]

[27] Jansson, P. A. "Deconvolution of Images and Spectra", Dover Publications, Inc., 2012. ISBN 978-0-486-45325-5 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=-CP0AwAAQBAJ&oi=fnd&pg=PP1&dq=Jansson,+P.+A.+Deconvolution+of+images+and+spectra.+Courier+Corporation,+2014.+ISBN+978-0486779941&ots=V8YjeM2ABG&sig=5PTzbgeUagGX7K-pV_KlADDsK10&redir_esc=y#v=onepage&q&f=false [Accessed: 23 February 2025]

[28] Riad, S. M. "The deconvolution problem: An overview", Proceedings of the IEEE, 74(1), pp. 82–85, 1986. https://doi.org/10.1109/PROC.1986.13407

[29] Panasonic "Product Guide - DMC-F2", [online] Available at: https://help.na.panasonic.com/camera-camcorder/lumix-point-and-shoot/dmc-f2/ [Accessed: 23 February 2025]

[30] The European Space Agency "Eyes on Hera: Asteroid mission's cameras ready", [online] Available at: https://www.esa.int/Space_Safety/Hera/Eyes_on_Hera_Asteroid_mission_s_cameras_ready [Accessed: 23 February 2025]

[31] Groetsch, C. W. "Inverse Problems in the Mathematical Sciences", Vieweg+Teubner Verlag, 1993. ISBN 3-528-06545-1 https://doi.org/10.1007/978-3-322-99202-4

[32] Engl, H. W., Groetsch, C. W. "Inverse and ill-posed problems", Academic Press, Inc. 2014. ISBN 978-1483272658 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=6pHOBQAAQBAJ&oi=fnd&pg=PP1&dq=Inverse+and+ill-posed+problems&ots=IYn3hx3gMG&sig=b6Uf_E35nZXWbp5F5EYQK5Glai0&redir_esc=y#v=onepage&q=Inverse%20and%20ill-posed%20problems&f=false [Accessed: 23 February 2025]

[33] Dimri, V. "Deconvolution and Inverse Theory: Application to Geophysical Problems", Elsevier, 2013. ISBN 9781483291376 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=t8_-BAAAQBAJ&oi=fnd&pg=PP1&dq=Dimri,+V.+Deconvolution+and+inverse+theory:+application+to+geophysical+problems.+Elsevier,+2013.+ISBN+978-0080999871&ots=_nY73-8X3F&sig=Gf6RS_1QfuKx62znoNRCLD3CTtE&redir_esc=y#v=onepage&q&f=false [Accessed: 23 February 2025]

[34] Hansen, P. C. "Deconvolution and Regularization with Toeplitz Matrices", Numerical Algorithms, 29(4), pp. 323–378, 2002. https://doi.org/10.1023/A:1015222829062

[35] Chaudhuri, S., Velmurugan, R., Rameshan, R. "Blind image deconvolution", Springer, 2014. ISBN 978-3319104843 https://doi.org/10.1007/978-3-319-10485-0

[36] Navarro, F., Serón, F. J., Gutierrez, D. "Motion Blur Rendering: State of the Art", Computer Graphics Forum, 30(1), pp. 3–26, 2011. https://doi.org/10.1111/j.1467-8659.2010.01840.x

[37] Tiwari, S., Shukla, V. P., Singh, A. K., Biradar, S. R. "Review of Motion Blur Estimation Techniques", Journal of Image and Graphics, 1(4), pp. 176–184, 2013. https://doi.org/10.12720/joig.1.4.176-184

[38] Thibos, L. N., Bradley, A., Liu, T., López-Gil, N. "Spherical Aberration and the Sign of Defocus", Optometry and Vision Science, 90(11), pp. 1284–1291, 2013. https://doi.org/10.1097/OPX.0000000000000040

[39] Sada, M. M., Goyani, M. M. "Image Deblurring Techniques – A Detail Review", [pdf] International Journal of Scientific Research in Science, Engineering and Technology, 4(2), pp. 176–188, 2018. Available at: https://www.researchgate.net/profile/Goyani-Mahesh/publication/338208138_Image_Deblurring_Techniques_-A_Detail_Review/links/5e06ccfca6fdcc2837438982/Image-Deblurring-Techniques-A-Detail-Review.pdf [Accessed: 23 February 2025]

[40] Valenzise, G., Alain, M., Zerman, E., Ozcinar, C. "Immersive Video Technologies", Elsevier, Inc., 2023. ISBN 978-0-323-91755-1 [online] Available at: https://books.google.hu/books?hl=hu&lr=&id=-A9vEAAAQBAJ&oi=fnd&pg=PP1&dq=Valenzise,+G.,+Alain,+M.,+Zerman,+E.,+Ozcinar,+C.+%22Immersive+Video+Technologies%22,+Academic+Press,+2022.+ISBN&ots=Np5yYsIT1T&sig=to4PKJnvCRVF4QxaY2WK2dQkf-U&redir_esc=y#v=onepage&q=Valenzise%2C%20G.%2C%20Alain%2C%20M.%2C%20Zerman%2C%20E.%2C%20Ozcinar%2C%20C.%20%22Immersive%20Video%20Technologies%22%2C%20Academic%20Press%2C%202022.%20ISBN&f=false [Accessed: 23 February 2025]