

# Real Time Deconvolution of Adaptive Optics Ground Based Telescope Imagery

**Toby Sanders, Robert Hedges**

*Lickenbrock Technologies*

**Timothy Schulz**

*Michigan Technological University*

**Melena Abijaoude, John Peters**

*Lickenbrock Technologies*

**Michael Steinbock**

*AFRL*

**Timothy Holmes**

*Lickenbrock Technologies*

## ABSTRACT

A system is under research and development that will produce and display deconvolved images of low earth orbiting objects during fly over. The goal of the first product delivery is 2 frames per second when the image frames are 512x512 and the telescope operates with adaptive optics. The target system is the 3.5 meter telescope of the Starfire Optical Range. The efforts have focused on issues that include, (1), speed of computation, (2), noise suppression, and (3), dynamic range degradation due to glint. The solution includes an improved variation of the multiframe blind deconvolution algorithm designed by one of the research team. The new variation has a speed advantage in that it requires less computation compared to its predecessor version. The foundation of the algorithm is the maximum likelihood estimation for the Poisson noise model. Fixed point iterations solve the objective function optimization. Part of the speed solution is the incorporation of Graphical Processing Units (GPUs) operating in parallel that serve as massively parallel processors. Another part of the speed solution is the design of optimization strategies that accelerate the convergence of the solution. The first prototype of the system was delivered to the Starfire Optical Range. Tests show that the frame rate of 2 frames per second is achievable with the current version of the system and future versions will be faster. A prototype variation of the algorithm for subpixel resolution is presented along with compelling examples in simulations and real case use. Finally, a new post-processing visual enhancement technique is proposed with several examples, which in part helps deal with the dynamic range degradation due to glint. Later versions of the system will target other ground-based telescope systems, including those without adaptive optics. The system may be extended, as well, for astronomical imaging and space telescopes.

## 1. INTRODUCTION

Ground-based telescopes are used for space observation in surveillance and defense. The imagery acquired by the telescopes are typically blurred and distorted by atmospheric turbulence. Adaptive optics reduces these effects by sensing the wavefront phase disturbances and automatically adjusting deformable mirrors to correct for this [8]. Even still, residual atmospheric turbulence remain. Moreover, additional blur remains due to the diffraction limit of the telescope determined by its limited aperture and optical aberrations inherent in the optical train of the telescope. For these reasons software and algorithms are needed to improve the imagery via deconvolution.

---

DISTRIBUTION A. Approved for public release: distribution unlimited.

A prototyped blind deconvolution algorithm for deconvolving such imagery was demonstrated in [18], in which the authors propose an algorithm yielding high-quality imagery with the future capability to operate in real time. The deconvolution uses a multiframe approach [19, 20], where a single output restored frame is resolved from multiple consecutive input frames. The multiple frame approach notably reduces the negative impact of noise. The term *blind* implies that the optical system's blur function, called the point spread function (PSF), must be determined within the deconvolution algorithm. The foundation of the algorithm is a maximum likelihood estimation (MLE) [9, 15], combined with a constrained model for the PSFs. Some details are provided in Section 2.

This work realizes the real time capability of this algorithm allowed by two key factors: (1), high-performance computing architecture using GPU and CUDA processing, (2), new numerical optimization techniques that reduce the number of iterations needed to achieve a certain level of convergence by a factor of 5-10. In addition to this, a super-resolution variant of the algorithm is derived and implemented, which is useful often in the case that the telescope camera has low resolution.

Finally a post-processing technique is developed for enhanced visual quality of the deconvolved images. It works by first improving the contrast through an adaptive histogram equalization [14], and second using a modified image sharpening technique that enhances edges while suppressing noise [7]. The histogram equalization helps deal with the large dynamic range of values inherent in the imagery, often referred to as glint. The sharpening technique does not improve the resolution, but significantly improves the visual quality to the human viewer.

## 2. DECONVOLUTION MODEL

We denote the image when want to restore by the variable  $u$ , for unknown. The  $k$ th frame measurement of  $u$  in the multiframe data is modeled as

$$d_k(x) = \text{Pois}(h_k * u(x) + b_k(x)), \quad (1)$$

where  $h_k$  are unknown point spread function (PSFs),  $b_k$  is a background image that is *nearly* constant, and  $x$  is the pixel index. The asterisk denotes convolution, which in the discretized image domain over  $N^2$  pixels is defined by

$$f * g(x) = \sum_{j=1}^{N^2} f(y_j)g(x - y_j). \quad (2)$$

Since  $h_k$  is unknown, the deconvolution is considered *blind*. The notation "Pois" indicates that the data at each pixel  $x$  has Poisson noise inherent in the measurements, which is a very common model used for photon image capture [2]. The goal of the deconvolution is to restore  $u$  from a series of measurements  $d_k$ ,  $k = 1, 2, \dots, K$ , where  $K = 5$  for example.

The foundation of our algorithm is the maximum-likelihood estimation (MLE) combined with a constrained model for the PSFs [19, 18]. Given the Poisson noise model, the MLE maximizes the probability of the data given the image and PSFs written as

$$\max_{u, \{h_k\}_k} p(\{d_k\}_k | u, \{h_k\}_k).$$

Taking the logarithm leads to the objective function to maximize:

$$\max_{u, \{h_k\}_k} F(u, \{h_k\}_k) = \max_{u, \{h_k\}_k} \sum_{k=1}^K \sum_{x=1}^{N^2} -i_k(x) + d_k(x) \ln(i_k(x)), \quad (3)$$

where  $i_k(x) = h_k * u(x) + b_k(x)$ . For single frame non-blind deconvolution ( $K = 1$  and  $h_k$  known), this estimator has been classically solved using the Richardson-Lucy (RL) algorithm [15, 11, 6].

To solve for the object  $u$  and the PSFs simultaneously, an alternating minimization is used, where for a fixed set of estimated PSFs, a new estimate of  $u$  is iterated, and visa-versa. The updates are given by fixed point iterations. This alternating minimization is repeated until some desired convergence, which in [18] is set at 250 total rounds of alternating minimizations. The update on  $u$  is similar to the RL algorithm. It is derived by differentiating the objective function with respect to  $u(y)$  for each pixel  $y$ , setting to zero, and rearranging the expression giving rise to the following fixed point iteration [18]:

$$u^{n+1}(x) = u^n(x) \left( \sum_{k=1}^K \left( \frac{d_k}{i_k^n} \right) * \tilde{h}_k(x) \right) / K. \quad (4)$$

The superscript  $n$  denotes the iteration count. The denominator  $K$  arises from our normalization constraint,  $\sum_x h_k(x) = 1$  for each  $K$ . The tilde notation denotes  $\tilde{h}_k(x) = h_k(-x)$ .

The problem of also determining the PSFs makes the deconvolution problem far more challenging. For blind deconvolution in general, additional assumptions for the PSFs are paramount. The approach here uses a constrained model for the PSFs based on the optics and aperture of the telescope. The aperture provides a framework for the PSFs, with a free parameter to be the optically and turbulence-induced phase aberration. The updates for the phase (and hence the PSFs) implemented here remain unchanged from the previous work (see [18] for details). Finally, the algorithm just described has been modified to include total variation regularizations and higher order variants to deal with noisy data sets [16], which is typically referred to as the maximum a posteriori estimation. The parameters for the regularization are automated by implementing a modification to the fast deconvolution parameter optimization techniques developed in [17].

### 3. ACCELERATED CONVERGENCE

Several ideas were tested to improve the convergence rate of the algorithm in order to reduce the number of iterations, and likewise computations, needed to achieve a certain level of convergence. Namely, line-search methods and Nesterov-like acceleration (NA) were investigated. They were tested for both accelerating the object and phase. They proved to be most useful for reducing the number of object iterations needed. The details and results of this work are given below.

#### 3.1 Line-search methods

Line-search methods are commonly used in gradient descent algorithms [22]. Consider minimizing an objective function  $f(x)$  over  $x$ . Then the  $n$ th iteration of a gradient descent algorithm takes the form

$$x^{n+1} = x^n - \tau^n \nabla f(x^n), \quad (5)$$

where  $\tau^n$  is the gradient step length. A line-search involves searching over the parameter  $\tau^n$  at each iteration to optimize the descent length<sup>1</sup>. The point of searching over the step length, as opposed to simply fixing a suitable  $\tau$  and moving to the next descent direction, is that re-evaluating the gradient can be much more costly than performing a search over  $\tau$ , making it computationally advantageous to perform a search. The search of  $\tau$  is found by roughly finding the following minimization through a search:

$$\tau^n = \arg \min_{\tau} f(x^n - \tau \nabla f(x^n)).$$

To what extent ones searches for this exact minimizer depends on the payoffs between the cost of the search and the cost of evaluating the next gradient descent direction.

A similar line-search method may also be implemented to accelerate the fixed point iteration in (4) (see e.g. [10]). The rational for doing so is similar to the gradient descent. Re-evaluating the next fixed point iteration is costly involving  $2K$  convolutions (hence up to  $6K$  FFTs). Alternatively, for the line-search we must evaluate the objective function involving  $K$  convolutions. However, these are the same convolutions needed for the fixed point iterations, so that they should be saved for the evaluation of the next iteration and not wasted.

For our problem we write the line-search vector as  $d^n = u^{n+1} - u^n$ , where  $u^n$  and  $u^{n+1}$  are defined as in (4). The intermediate solution returned from the line search is written as  $v^{n+1} = u^n + \tau^n(u^{n+1} - u^n)$ , and after the search is completed  $u^{n+1}$  is replaced by  $v^{n+1}$ . Without a line-search,  $\tau^n = 1$ . To accelerate the algorithm, we test larger values of  $\tau^n$ , evaluate the objective function  $F$  in (3), and continue to increase  $\tau^n$  so long as  $F$  increases. For simplicity, we increase  $\tau^n$  by factors of 2 for testing. Though this approach will reduce the number of iterations, evaluating the objective function for each new test value comes at some cost.

#### 3.2 Nesterov-like acceleration (NA)

Considering the gradient descent written in (5), Nesterov's accelerated gradient descent [13, 12] takes the form

$$\begin{aligned} x^{n+1} &= y^n - \tau \nabla f(y^n) \\ y^{n+1} &= x^{n+1} + \alpha^n(x^{n+1} - x^n). \end{aligned} \quad (6)$$

---

<sup>1</sup>In some instances, the optimal step length can be determined with an analytical expression.

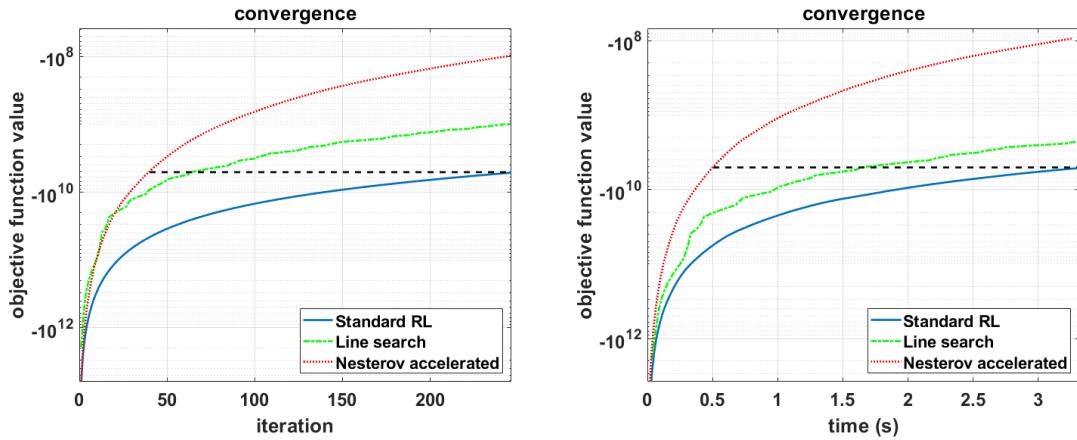


Fig. 1: Comparison of the convergence of the standard algorithm with the acceleration methods on a simulated example. The Nesterov acceleration provides significant speed up in terms of both number of iterations and time. The line search method provides some speed up while also requiring more time for the line search in each iteration.

The intermediate estimates  $y^n$  are sometimes referred to as scaled heavy-ball acceleration. An analogous acceleration method for RL deconvolution algorithms was proposed much later by Biggs *et al* [3], with what appears to be no knowledge of Nesterov's original work. Convergence in the case of the RL algorithm was recently proven [23].

We modify our fixed point iteration in (4) to take the following form

$$\begin{aligned} u^{n+1} &= v^n(x) \left( \sum_{k=1}^K \left( \frac{d_k}{\ell_k^n} \right) * \tilde{h}_k(x) \right) / K, \\ v^{n+1} &= u^{n+1} + \alpha^n (u^{n+1} - u^n) \end{aligned} \quad (7)$$

where  $\ell_k^n = v^n * h_k + b_k$ . This is simply the analog to (6) for our fixed point iteration. It is similar to that in [3] with two exceptions. First, our acceleration is modified for the multiframe deconvolution. Second, we use an explicit expression for  $\alpha^n$ , whereas Biggs *et al* requires a computational procedure for computing  $\alpha^n$ . Our empirical results have indicated that setting  $\alpha^n = (n-1)/(n+2)$  works very well, as suggested in [21]. This approach comes at no notable computational cost within each iteration, while significantly improving the rate of convergence, shown below.

### 3.3 Numerical Results

The NA and line-search acceleration were tested on real and simulated data sets. Figure 1 shows the convergence plots of these methods in a simulated example on a 512x512 image and compares with the standard algorithm without acceleration. The left shows the value of the objective function at each iteration. The plot on the right shows the value of the objective function as a function of the time elapsed<sup>2</sup>. In both cases, the NA shows significant improvements, accelerating the convergence rate from the standard algorithm by a factor of 5-10 (indicated by the horizontal black-dashed line). Figure 2 shows the deconvolved images resulting from this simulation. The deconvolutions from the standard algorithm and the NA algorithm at 50 iterations are compared. They agree with the plots in Figure 1 in the sense that the larger value of the objective function seen in the accelerated case leads to a sharper image.

Figure 3 shows the results of the NA on a real data set. The convergence plots (bottom left) indicate a similar reduction in computational load as in the simulated examples. The results of the accelerated and standard deconvolution images at both 25 and 150 iterations shown here indicate significant improvement in image quality with the accelerated method.

<sup>2</sup>The examples were tested in prototype MATLAB code on standard CPU hardware, therefore they do not show the real time speed.

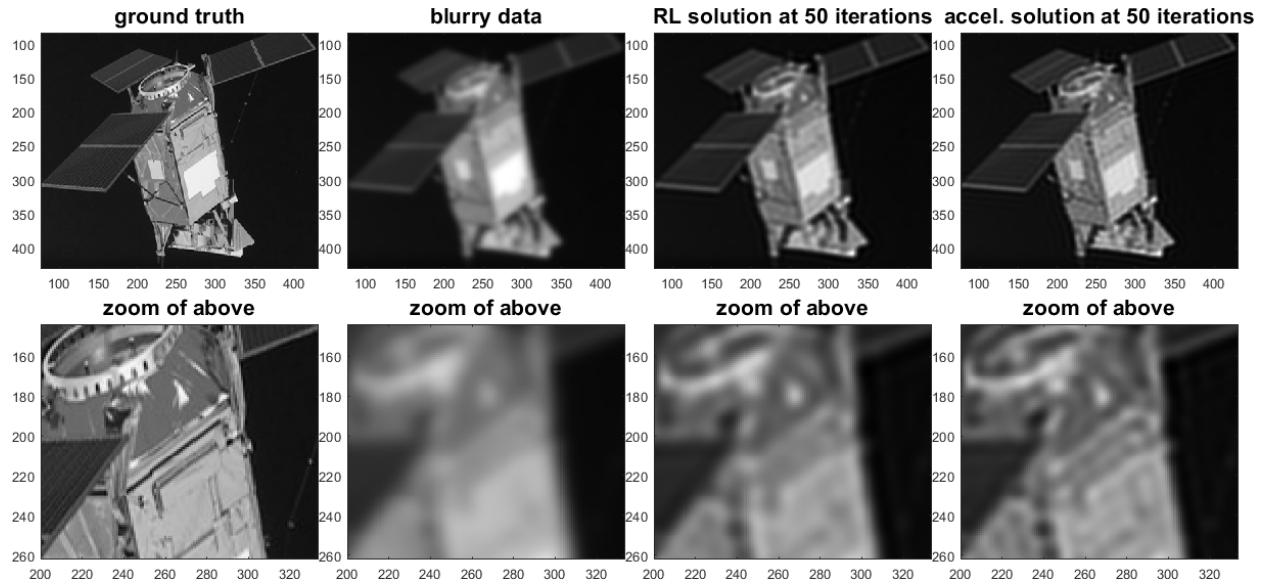


Fig. 2: Comparison of the deblurred image data after 50 iterations with the standard algorithm and the Nesterov accelerated algorithm.

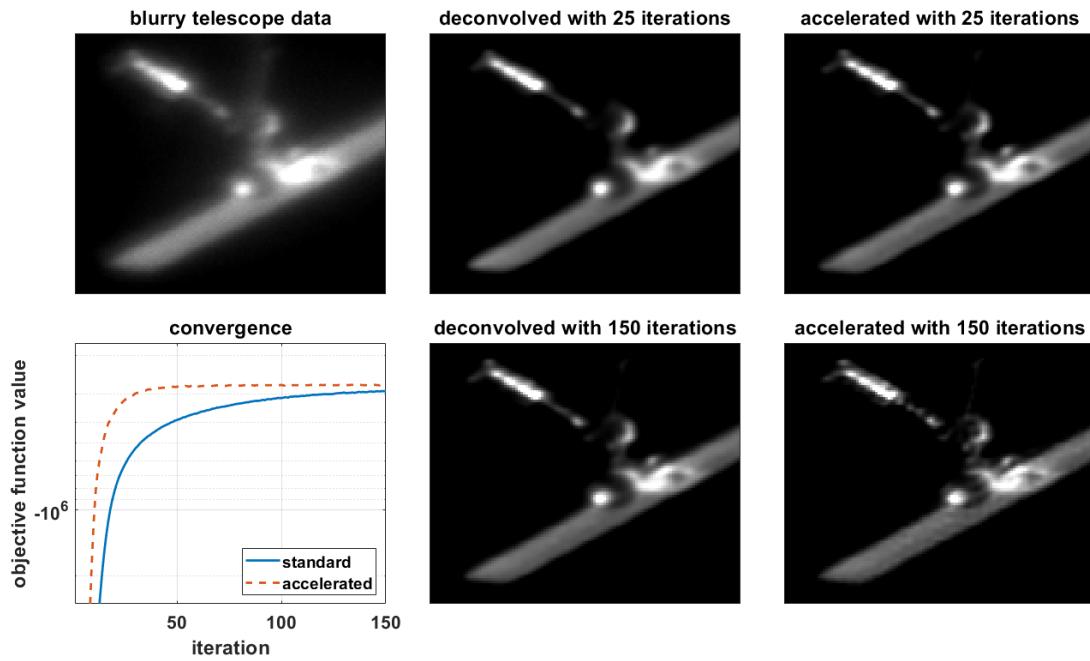


Fig. 3: Comparison of the deblurred image data with the standard algorithm and the Nesterov accelerated algorithm. Our empirical results indicate that 25-50 iterations of the accelerated algorithm yields results similar to 250 iterations of the standard algorithm.

## 4. CUDA IMPLEMENTATION

In order to meet the real-time requirements the prototype MATLAB implementation was migrated to C++/CUDA. Once ported to CUDA, several optimizations were made.

### 4.1 CUDA Best-Practices

The first approach was to follow CUDA best practices by minimizing data transfers between the CPU and GPU, parallelizing algorithms through the removal of loops, and utilizing concurrent calculations. To minimize data transfers, the input image is copied to the GPU, then all iteration operations are performed on the GPU. Only after the iteration is complete is the deblurred image copied back to the CPU for display.

Algorithms were parallelized by making use of the Fourier Convolution Theorem, which allows us to evaluate the convolutions with 3 FFTs and a Hadamard product. The CUDA library contains a highly optimized FFT routine, and the Hadamard-product is perfectly suited to a parallel (kernel-based) implementation. Further optimizations were implemented by using the real-to-complex FFTs which exploit its conjugate symmetry; cutting memory usage in half, and also cutting the number of Hadamard-products in half as well as reducing the computations for the 2-D FFTs.

### 4.2 Block Processing of FFTs

To further optimize the convolution calculations batch processing of the FFT was performed, thus allowing to process entire entire block of input data frames in one call. Instead of sequentially computing the FFT.

$$\mathcal{F}(d), \mathcal{F}(h_1), \mathcal{F}(h_2), \dots, \mathcal{F}(h_K) \quad (8)$$

We can compute

$$\mathcal{F}([d, h_1, h_2, \dots, h_K]) \text{ denoted as } \mathcal{F}([d, \vec{h}]) \quad (9)$$

which has  $(k + 1)$  2-D FFTs per call. This provided significant speed-up. It also reduced the proportionality between execution time and number of Frames-per-block; in a serial implementation each additional frame as an additional convolution to compute, in parallel it only increases the size of the batch, and may not increase the execution time.

### 4.3 Parallel Phase and Object Processing

Further optimization made use of the fact that the phase and object iterations are independent in the sense that the phase iteration does not make use of the updated object. Since both iterations have the same initial conditions they can be computed concurrently which cuts the per-iteration time by nearly 10 percent.

### 4.4 Benchmarks

We measured an increase in frame-rate as each of the steps above were implemented, (see Table 1). The effect of the Nesterov acceleration, (see section 3) was key in surpassing the required frame-rate of 2 frames/sec.

Method	Maximum Frame-rate (frames/s)
Serial phase after object implementation	0.25
Block FFTs and parallel phase and object	0.44
Block FFTs and parallel phase and object + Nesterov	2.14*

Table 1: Various implementation stages vs Maximum Achievable frame-rate for 512x512 images and 5 frames/block.  
*\*Note: the Nesterov frame-rate improvement stems from reducing the number of iterations from 250 to 50 due to the higher rate of convergence provided by the Nesterov step*

## 5. SUBPIXEL RESOLUTION

A variant of the algorithm was developed to increase the resolution of the original input blurry image. This is useful whenever the telescope detector grid is coarse resulting in an observable blocky-pixelated image. To increase the resolution, we model each cell on the detector grid to be split into a new 3x3 mesh for 9 new pixels (see Figure 4).

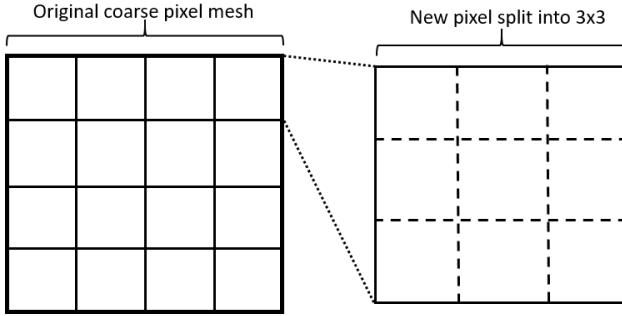


Fig. 4: For subpixel resolution, from the original coarse mesh each pixel is split into a new  $3 \times 3$  mesh increasing the resolution by a factor of 9.

We assume that the value measured at the original cell contains the sum of the values of the 9 new pixels. Then with respect to the new image mesh, the acquired data values are now modeled as

$$d_k(x) = \text{Pois}[g * (h_k * u)(x) + b_k(x)], \quad (10)$$

where  $h_k$ ,  $u$ , and  $b_k$  are as before. The function  $g(x)$  is a  $3 \times 3$  averaging kernel with each value taking  $1/9$ , hence convolving with  $g$  is the operation that averages each  $3 \times 3$  block in the fine mesh to get to the coarse mesh. The values of  $x$  in this equation on the left hand side are only measured at every 9th pixel for those on the right hand side. Let's say the total set of pixels in the measured image is  $N^2$  and those in the refined image mesh is  $9N^2$ . Then if we write  $g_k(x) = g * h_k(x)$ , the new objective function to replace (3) is given by

$$\max_{u, \{h_k\}_k} F(u, \{h_k\}_k) = \max_{u, \{h_k\}_k} \sum_{k=1}^K \sum_{x \in S} -i_k(x) + d_k(x) \ln(i_k(x)), \quad (11)$$

where  $S$  is the subset of  $N^2$  measured pixels, and  $i_k$  is given by

$$i_k(x) = g_k * u(x) + b_k(x).$$

To determine the new fixed point iteration for  $u$ , we differentiate (11) with respect to  $u(y)$  and set equal to zero to obtain

$$0 = \frac{\partial F}{\partial u(y)} = \sum_{k=1}^K \sum_{x \in S} -g_k(x-y) + \frac{d_k(x)}{i_k(x)} g_k(x-y). \quad (12)$$

Since the original PSFs  $h_k$  sum to 1, observe the first sum to be

$$\sum_{x \in S} -g_k(x-y) = -1/9.$$

Define a mask over the measured pixels by

$$m(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S. \end{cases}$$

Then (12) may be re-written as <sup>3</sup>

$$\begin{aligned} 0 &= -K/9 + \sum_{k=1}^K \sum_{x=1}^{9N^2} \frac{d_k(x)m(x)}{i_k(x)} g_k(x-y) \\ &= -K/9 + \sum_{k=1}^K \left( \frac{d_k \cdot m}{i_k} \right) * \tilde{g}_k(y). \end{aligned} \quad (13)$$

<sup>3</sup>Observe here that  $d_k(x)$  is listed over pixels in which it is not measured, but in these instances it is multiplied by  $m(x) = 0$ , so it can be considered to take any arbitrary values here.

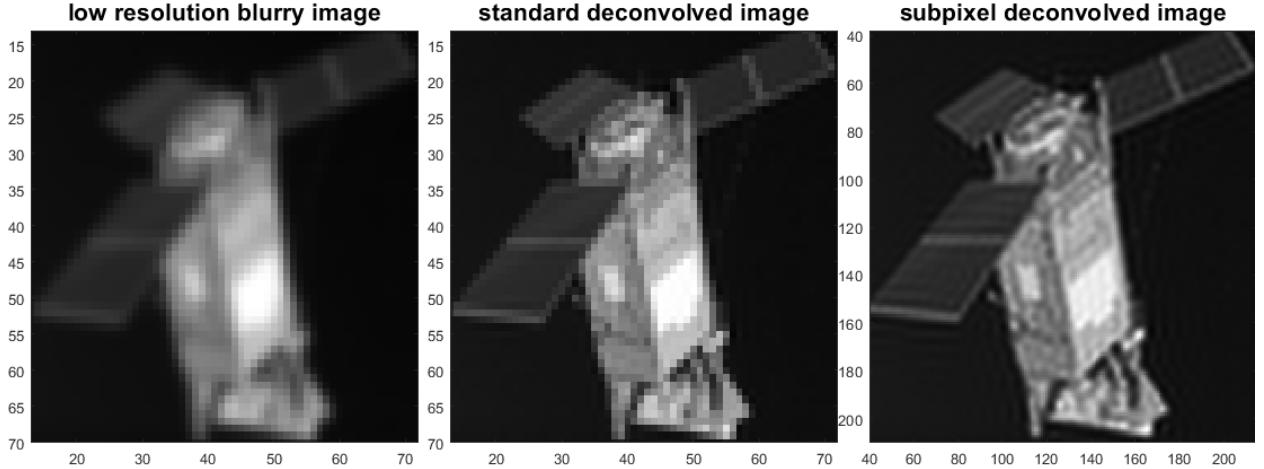


Fig. 5: Simulated example demonstrating the effectiveness of the subpixel algorithm

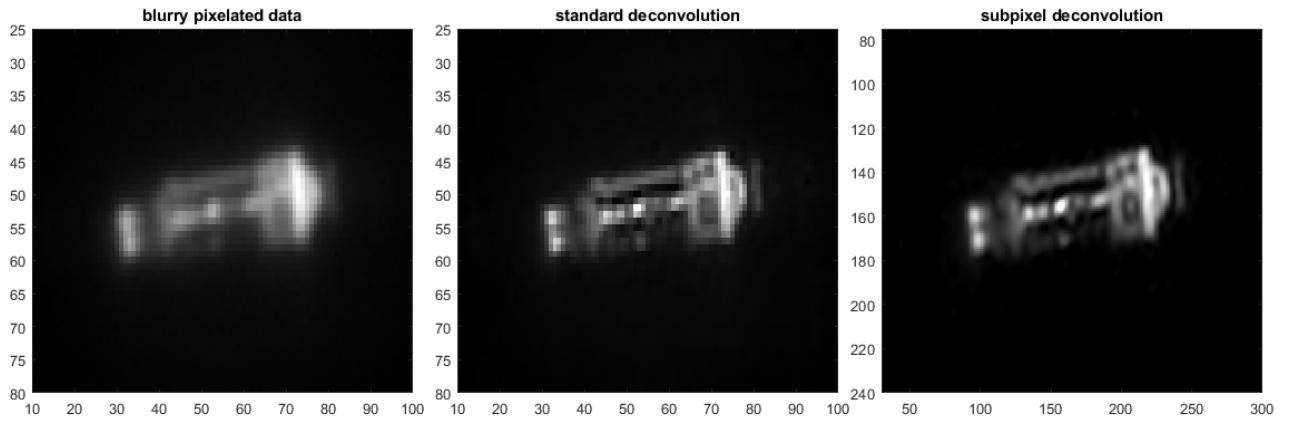


Fig. 6: Real case use of the subpixel algorithm.

Rearranging terms and multiplying through by  $u$  leads to the fixed point equation for the subpixel algorithm:

$$u^{n+1}(x) = u^n(x) \left( \sum_{k=1}^K \left( \frac{d_k \cdot m}{i_k^n} \right) * \tilde{g}_k(x) \right) / (K/9). \quad (14)$$

This algorithm is accelerated in the same way as in (6).

Results on the subpixel resolution are shown in Figures 5 and 6. In Figure 5 is a simulated example, where the original image (shown in Figure 2) is down-sampled by a factor of 6 in each dimension, and then blurred with a Gaussian PSF. A total of 5 images were used and one of these images is shown in the left of the figure. The standard resolution algorithm was used to deconvolve this image (middle), and while significantly sharper, it shows notable pixelation. The subpixel deconvolved image (right) shows significantly improved visual image quality.

Results for a real data set are shown in Figure 6. The dimension of the original blurry image is 128x128, and the object in view only covers a small portion of the image, making the visual very pixelated. A total of 5 frames were again used for the processing. The standard deconvolved image (middle) is improved but still pixelated. The subpixel resolution (right) again shows significantly improved visual quality.

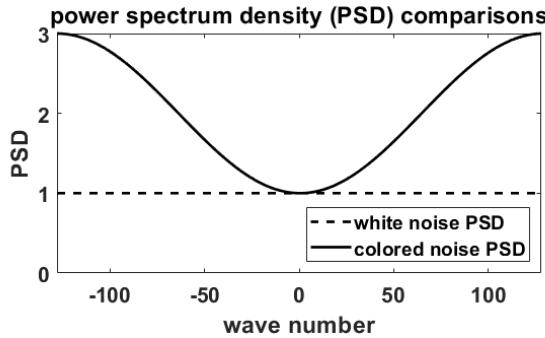


Fig. 7: Comparison of PSDs of the noise before (white) and after (colored) sharpening.

## 6. VISUAL ENHANCEMENT BY POST-PROCESSING

In this section we introduce some new tools for post-processing to enhance the visual quality of the output images. These methods are not intended reduce the blur or improve the resolution, but instead modify the image in order to improve the visual quality to the human inspector. The main issue that needs to be address for human perception is the image contrast. Because of the large dynamic range of values in the images (particularly after deblurring/sharpening), a basic linear grayscale colormap is usually insufficient for display. For example, referring back to figures 3 and 6, a simple contrast adjustment was used to visualize these. However, there still remain low light panels on the right side that cannot be seen with this color map. This issue is often referred to as glint correction.

A glint correction method for satellite imagery was recently proposed in [1], and we demonstrate notable improvements to this approach. First, an adaptive histogram equalization (HEQ) is implemented, which is a well-known technique to enhance detail for such images with poor contrast [14]. We found this approach provides comparable results to the current glint correction.

Next, we consider a visual enhancement by edge sharpening with a Laplacian filter [7]. Denoting the image obtained from the histogram equalization by  $u$ , then the sharpened image is given by

$$v = u + \alpha(g * u),$$

where  $g$  is the Laplacian Kernel

$$g = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix},$$

and  $\alpha > 0$  is a constant that controls the level of sharpening. Typical values for  $\alpha$  are in the range  $[0, 3]$  (note  $\alpha = 0$  returns the original image). Larger values of  $\alpha$  results in greater enhancement of edges. The idea of this approach is that convolution with  $g$  acts as a high pass filter, hence the procedure captures the edge information of the image and adds it back to the original to enhance them. However, noise located at these high frequencies will also be captured and enhanced. For large values of  $\alpha$  (e.g.  $\alpha = 3$ ), noise becomes very apparent even when the noise is not perceived by the human eye in the original image. Therefore typically  $\alpha$  is chosen to strike a balance between sharpening and keeping the noise suppressed.

We propose a fix for this that allows us to choose an aggressive value for  $\alpha$  without adding noise to the modified image. We combined the sharpening method with the 3D block-matching (BM3D) colored denoising approach [4, 5]. In its basic form, BM3D is one of the most powerful off the shelf algorithms for denoising images with white noise [4]. Furthermore, this denoising algorithm was modified to deal with colored noise [5]. If we assume the power spectrum density (PSD) of the deconvolved image is *somewhat* white, then the power spectrum of the sharpened image is colored (see Figure 7). The colored denoising version of BM3D adapts the wavelet thresholding values dependent on the location in frequency of the wavelets and the input PSD of the noisy image.

With this approach, a very aggressive value of  $\alpha$  may be chosen without introducing much noise. The summary of this full process is, that given the output image  $u$  from the maximum-likelihood deconvolution, the visually enhanced

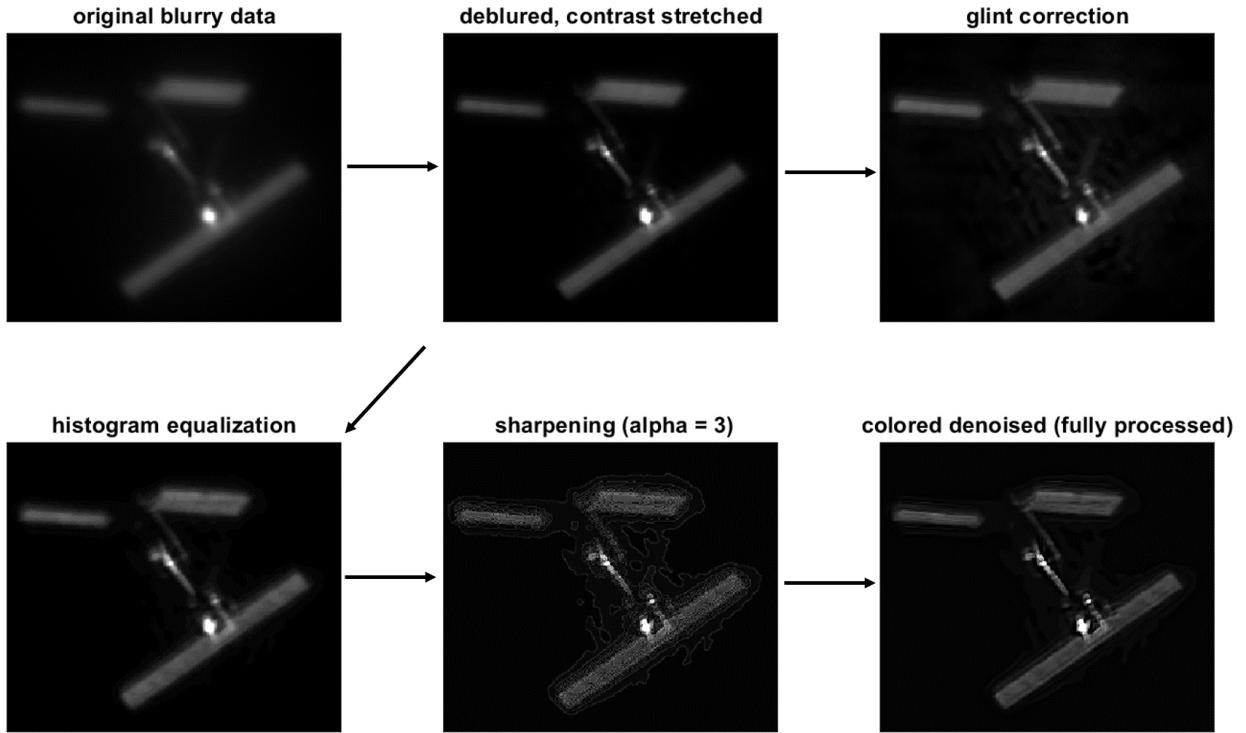


Fig. 8: Full work flow from the original blurry image, to the deblurred and visually enhanced sharpened image. Top right image is the glint correction proposed in [1] and bottom right is the current methodology.

image is given by the sequence of steps:

$$\begin{aligned}
 u_1 &= \text{HEQ}(u) && \text{(contrast enhancement by HEQ)} \\
 u_2 &= u_1 + \alpha(g * u_1) && \text{(edge enhancement by Laplacian filtering)} \\
 u_3 &= \text{BM3D}(u_2; \alpha, g) && \text{(colored noise reduction by BM3D).}
 \end{aligned}$$

An example showing the full process, starting with the original blurry image and finishing with the deblurred and post-processed image is shown in Figure 8. The visually enhanced image from [1] is shown for comparison in the top right. Observe that the sharpened image (bottom, middle) significantly brings out noise that is not visible in the deconvolved image, but the colored denoising with BM3D effectively eliminates this noise resulting in a visually pleasing final result. Additional examples of the sharpening approach are shown in Figure 9.

## 7. SUMMARY

We have presented a software that is capable of deconvolving telescope imagery with the multiframe blind deconvolution approach at a rate 2 frames per second. This capability was achieved by first carefully porting the algorithm onto massively parallel GPUs with CUDA best practices, and second designing and testing several optimization techniques to improve the rate of convergence. Our results showed that a novel Nesterov-like acceleration approach achieved the best results, allowing us to reduce the number of iterations needed by a factor of 5-10. A subpixel variation of the algorithm was derived and prototyped in MATLAB, and our numerical results showed this is very useful in simulation and practice whenever the object in view is low resolution. Finally, some new visual enhancement techniques were proposed to improve the contrast and viewing experience to the human user. These results showed good promise and this work will be ongoing. Later versions of our software will target a broader set of data sets, including those without adaptive optics. In addition, a prototype algorithm to deal with bad and *sensitive* pixels on the camera detector is currently under investigation.

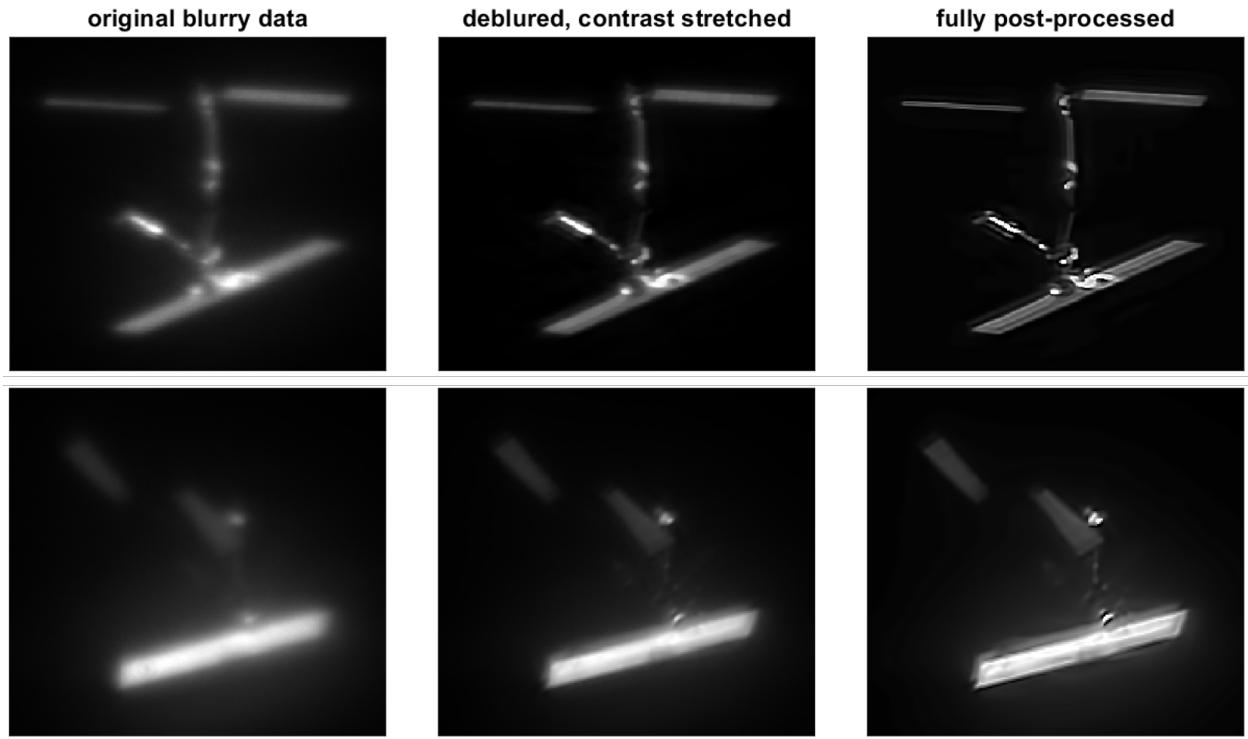


Fig. 9: Examples showing original data, the deblurred image, and the fully post-processed image.

#### ACKNOWLEDGEMENTS

The authors would like to thank Captain Anastacio Arreloa for all of his help on the project and software testing. Funding for this work is provided by the Air Force Research Laboratory, contract number FA9451-19-C-0579.

- [1] Melena Abijaoude, Timothy Schulz, David Gray, Mark Ostroot, Miguel Pineros, and Timothy Holmes. Practical aspects of multiframe blind deconvolution for ground based telescopes. In *Unconventional and Indirect Imaging, Image Reconstruction, and Wavefront Sensing 2018*, volume 10772, page 1077212. International Society for Optics and Photonics, 2018.
- [2] Mario Bertero, Patrizia Boccacci, Gabriele Desiderà, and Giuseppe Vicidomini. Image deblurring with Poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009.
- [3] David SC Biggs and Mark Andrews. Acceleration of iterative image restoration algorithms. *Applied optics*, 36(8):1766–1775, 1997.
- [4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
- [5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207. International Society for Optics and Photonics, 2008.
- [6] DA Fish, AM Brinicombe, ER Pike, and JG Walker. Blind deconvolution by means of the Richardson–Lucy algorithm. *JOSA A*, 12(1):58–65, 1995.
- [7] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.
- [8] John W Hardy. *Adaptive optics for astronomical telescopes*, volume 16. Oxford University Press on Demand, 1998.
- [9] Timothy J Holmes. Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach. *JOSA A*, 9(7):1052–1061, 1992.
- [10] Timothy J Holmes and Yi-Hwa Liu. Acceleration of maximum-likelihood image restoration for fluorescence

- microscopy and other noncoherent imagery. *JOSA A*, 8(6):893–907, 1991.
- [11] Leon B Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.
- [12] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [13] Yurii Nesterov. A method of solving a convex programming problem with convergence rate  $O(1/k^2)$ . In *Soviet Mathematics Doklady*, 27:372–376, 2013.
- [14] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.
- [15] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59, 1972.
- [16] Toby Sanders and Rodrigo B Platte. Multiscale higher-order TV operators for L1 regularization. *Advanced structural and chemical imaging*, 4(1):12, 2018.
- [17] Toby Sanders, Rodrigo B Platte, and Robert D Skeel. Effective new methods for automated parameter selection in regularized inverse problems. *Applied Numerical Mathematics*, 152:29–48, 2020.
- [18] Timothy Schulz, Melena Abijaoude, David Gray, Mark Ostroot, Miguel Pineros, and Timothy Holmes. Advances in multiframe blind deconvolution for ground based telescopes. In *Unconventional and Indirect Imaging, Image Reconstruction, and Wavefront Sensing 2018*, volume 10772, page 107720F. International Society for Optics and Photonics, 2018.
- [19] Timothy J Schulz. Multiframe blind deconvolution of astronomical images. *JOSA A*, 10(5):1064–1073, 1993.
- [20] Timothy J Schulz, Bruce E Stribling, and Jason J Miller. Multiframe blind deconvolution with real data: imagery of the hubble space telescope. *Optics express*, 1(11):355–362, 1997.
- [21] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pages 2510–2518, 2014.
- [22] Andreas Wächter and Lorenz T Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2005.
- [23] Hongbin Wang and Paul C Miller. Scaled heavy-ball acceleration of the Richardson-Lucy algorithm for 3D microscopy image restoration. *IEEE Transactions on Image Processing*, 23(2):848–854, 2013.