

DeconvolutionLab2: An open-source software for deconvolution microscopy

Daniel Sage, Lauréne Donati, Ferréol Soulez, Denis Fortun, Guillaume Schmit, Arne Seitz, Romain Guiet, Cédric Vonesch, Michael Unser

▶ **To cite this version:**

Daniel Sage, Lauréne Donati, Ferréol Soulez, Denis Fortun, Guillaume Schmit, et al.. DeconvolutionLab2: An open-source software for deconvolution microscopy. Methods, 2017, 115, pp.28-41. 10.1016/j.ymeth.2016.12.015 . insu-01632774

HAL Id: insu-01632774
https://insu.hal.science/insu-01632774

Submitted on 13 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Methods 115 (2017) 28–41

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

DeconvolutionLab2: An open-source software for deconvolution microscopy

Daniel Sage [a,*], Lauréne Donati [a], Ferréol Soulez [a], Denis Fortun [b], Guillaume Schmit [a], Arne Seitz [c], Romain Guiet [c], Cédric Vonesch [b], Michael Unser [a]

[a] Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[b] Center for Biomedical Imaging-Signal Processing Core (CIBM-SP), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[c] BioImaging and Optics Platform, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 7 September 2016
Received in revised form 21 December 2016
Accepted 30 December 2016
Available online 3 January 2017

Keywords:
Deconvolution microscopy
Open-source software
Standard algorithms
Textbook approach
Reference datasets

ABSTRACT

Images in fluorescence microscopy are inherently blurred due to the limit of diffraction of light. The purpose of deconvolution microscopy is to compensate numerically for this degradation. Deconvolution is widely used to restore fine details of 3D biological samples. Unfortunately, dealing with deconvolution tools is not straightforward. Among others, end users have to select the appropriate algorithm, calibration and parametrization, while potentially facing demanding computational tasks. To make deconvolution more accessible, we have developed a practical platform for deconvolution microscopy called `DeconvolutionLab`. Freely distributed, `DeconvolutionLab` hosts standard algorithms for 3D microscopy deconvolution and drives them through a user-oriented interface. In this paper, we take advantage of the release of `DeconvolutionLab2` to provide a complete description of the software package and its built-in deconvolution algorithms. We examine several standard algorithms used in deconvolution microscopy, notably: Regularized inverse filter, Tikhonov regularization, Landweber, Tikhonov–Miller, Richardson–Lucy, and fast iterative shrinkage-thresholding. We evaluate these methods over large 3D microscopy images using simulated datasets and real experimental images. We distinguish the algorithms in terms of image quality, performance, usability and computational requirements. Our presentation is completed with a discussion of recent trends in deconvolution, inspired by the results of the Grand Challenge on deconvolution microscopy that was recently organized.

© 2017 Elsevier Inc. All rights reserved.

Contents

1. Introduction . . . 29
2. `DeconvolutionLab2`: A Java open-source software package . . . 30
   2.1. `DeconvolutionLab`: The original `ImageJ` deconvolution tool . . . 30
   2.2. `DeconvolutionLab2`: The remasterized `Java` deconvolution tool . . . 30
      2.2.1. Practical details . . . 31
3. Deconvolution algorithms . . . 31
   3.1. Image-formation model . . . 31
   3.2. Naive inverse filtering . . . 32
   3.3. Tihkonov regularization . . . 32
   3.4. Regularized inverse filtering . . . 32
   3.5. Landweber . . . 32
   3.6. Tikhonov–Miller . . . 32
   3.7. Fast iterative soft-thresholing . . . 32
   3.8. Richardson–Lucy . . . 33
   3.9. Richardson–Lucy with total-variation regularization . . . 33

* Corresponding author.
E-mail addresses: daniel.sage@epfl.ch (D. Sage), laurene.donati@epfl.ch (L. Donati), ferreol.soulez@epfl.ch (F. Soulez), denis.fortun@epfl.ch (D. Fortun).

http://dx.doi.org/10.1016/j.ymeth.2016.12.015
1046-2023/© 2017 Elsevier Inc. All rights reserved.

4. Deconvolution in practice . . . 33
4.1. Image acquisition . . . 33
4.2. Point-spread function . . . 33
4.3. Setting of parameters . . . 33
4.3.1. Ghosts and ringing . . . 33
5. Experimental illustrations . . . 34
5.1. Synthetic data . . . 34
5.2. Isolated bead . . . 35
5.3. Widefield data . . . 35
6. Discussion: trends in deconvolution . . . 36
6.1. Blind deconvolution . . . 39
6.2. Space-varying deconvolution . . . 39
Acknowledgement . . . 39
Appendix A. Implementation aspects . . . 39
A.1. FFT Libraries . . . 39
A.2. Dissection of an algorithm . . . 39
A.2.1. Implementation of the Landweber algorithm . . . 39
A.2.2. Java snippet of Landweber . . . 40
References . . . 40

1. Introduction

The widespread development of fluorescent-labeling techniques has rendered fluorescent microscopy one of the most popular imaging modalities in biology. An epifluorescence (a.k.a. widefield) microscope is indeed the simplest modality for observing cellular structures: After labelling with a fluorescent dye, the biological specimen is illuminated at the excitation wavelength. The fluorescence emission is used to form the image. A 3D acquisition of the cell is built as a z-stack of 2D images, by moving the focal plane through the sample.

Unfortunately, the resolution of 3D micrographs is intrinsically limited by the diffraction of light; structures closer than the Rayleigh criterion cannot be distinguished. For a popular fluorophore (DAPI, emission wavelength $\lambda = 470$ nm) and for the standard numerical aperture NA $= 1.4$ and diffraction index $n_i = 1.51$ nm, the Rayleigh criterion predicts that it is impossible to observe details smaller than about $0.61\frac{\lambda}{\mathrm{NA}} \approx 200$ nm in the lateral sections and $2\frac{n_i\lambda^2}{\mathrm{NA}} \approx 700$ nm along the optical axis [1]. Thus, the resolution is anisotropic, *i.e.*, the resolution along the depth axis is lower than the resolution in the lateral dimensions. Moreover, this resolution is usually insufficient to satisfy the current demands of biological research for the visualisation of intracellular organelles. The impact of diffraction is perceived as a blur, where fine details are obscured by the haze produced by out-of-focus light. The acquired blurred image can be mathematically modeled as the result of convolving the observed objects with a 3D point-spread function (PSF). This PSF is the diffraction pattern of the light that would be emitted from an infinitesimal point-like object and collected by the microscope. In other words, the PSF sums up the effects of the imaging setup on the observations.

There are two approaches to improve the resolution: (i) changing the microscope design to improve the shape of the PSF (e.g. confocal, multiphoton and most super-resolution microscopy modalities), (ii) numerically inverting the blurring process to enhance micrographs: the deconvolution. The ultimate goal of deconvolution is to restore the original signal that was degraded by the acquisition system (see Fig. 1). It relies on methods that have to be carefully optimized to preserve biological information. We present these methods in Section 3.

Deconvolution is a versatile restoration technique that has been found useful in various contexts such as biomedical signal processing, electro-encephalography, seismic signal (1D), astronomy (2D), or biology (3D). It performs well in 1D or 2D, but its results are the most impressive for 3D volumetric data, especially when the PSF is large axially. In this case, 3D deconvolution has the capability to combine lateral and axial information when restoring the original signal.

Deconvolution has multiple advantages. It is applicable to even the simplest optical setup, reducing financial costs and streamlining the acquisition pipeline. In addition to the resolution improvement, indirect benefits of deconvolution are contrast enhancement and noise reduction. As it mitigates the effect of noise, it can be used in low-light condition. The dim excitation light lowers bleaching probability of fluorophores and is therefore beneficial in terms of photo-toxicity in living cells. Not surprisingly, deconvolution is used routinely by microscopists and has become a popular pre-processing tool to further image-analysis steps such as segmentation and tracking. Unfortunately, without a proper tuning of the algorithms parameters, the deconvolved volume can be corrupted by artifacts that might prevent sound biological interpretation. Among such possible degradations, the most notable ones are noise amplification, ringing (known as Gibbs or Runge phenomenon) and aliasing (both spatial and spectral).

The deconvolution of micrographs was first investigated by Agard and Sedat [2]. Many variations and improvements have been proposed since then [3–7]. Some of these "deconvolution microscopy" methods led to various commercial and open-source software implementations [8,9]. The typical cost of a commercial package varies between USD 5000 and USD 10,000. At the time of writing this paper, the most popular ones are: `Huygens` (Scientific Volume Imaging); `DeltaVision Deconvolution` (Applied Precision, GE Healthcare Life Science); and `AutoQuant` (MediaCybernetics). Some of these commercial solutions (e.g., `Huygens`) specialize in the processing of large data and are capable of running unattended deconvolution in batch mode [10].

Meanwhile, several open-source deconvolution solutions exist too, often taking the form of an `ImageJ`[1] plugin. One of the first such platform that was made available is the popular `DeconvolutionLab` software developed at the Biomedical Imaging Group (EPFL) and detailed in the present paper. Freely distributed, `DeconvolutionLab` hosts various algorithms for 3D microscopy deconvolution and drives them through a user-oriented interface. Other open-source softwares also exist, including Nick Linnenbrügger's `DeconvolutionJ`, Bob Dougerthy's `Iterative Deconvolve 3D`[2] which implements a deconvolution approach for the mapping of

[1] http://imagej.nih.gov/ij/.
[2] http://www.optinav.info/Iterative-Deconvolve-3D.htm.

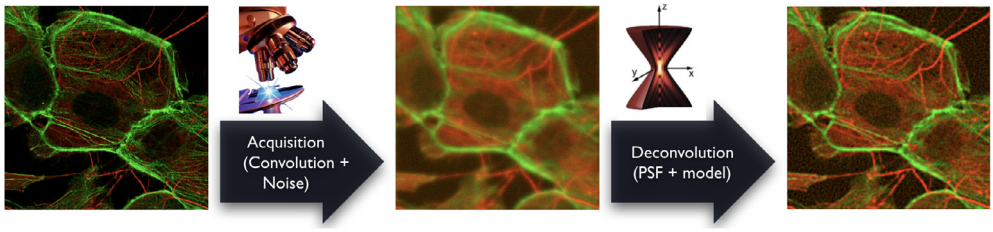


Fig. 1. Principle of the deconvolution of a *z*-stackof images, presented here as the maximum-intensity projection of the volumetric data.

acoustic sources, Piotr Wendykier's `Parallel Iterative Deconvolution`[3] which proposes four iterative algorithms, and the `MiTiV`[4] project that proposes blind deconvolution software.

The deconvolution of three-dimensional data is a computationally heavy process. Fortunately, the last decades have seen a strong increase in the general accessibility to computing power. Without special equipment, it has now become possible to deconvolve data of practical size $(512 \times 512 \times 64)$ on a 8 GB consumer-grade computer in less than a couple minutes. Thus, the number of users having gained access to deconvolution has grown markedly through the years, which stresses the need for *accessible* and *user-friendly* software packages for deconvolution microscopy. This need is heightened by the fact that many potential users are biologists or life-science students, who may lack in computer and algorithmic literacy, so that they would have to be educated about the different available algorithms. Among others, the required skills address the selection of parameters, the control of computational and memory costs, and the recognition of restoration artifacts.

In this paper, we take advantage of the release of `DeconvolutionLab2`, the revamped sequel of `DeconvolutionLab`, to provide a complete description of the software and its built-in deconvolution algorithms. In regards to the aforementioned pedagogical aspects, the present paper equally intends as a step toward the education of inexperienced users.

2. `DeconvolutionLab2`: A Java open-source software package

Although microscope manufacturers may sometimes propose well-integrated software packages, their solutions are often mere black boxes. This situation prevents users to make an informed choice on which commercial deconvolution software is the most appropriate for their task at hand. Conversely, many deconvolution methods have been described in the scientific literature over the past twenty years, sometimes accompanied by open-source implementations. But even then, end users who do not master the underlying principles of deconvolution might face difficulties in selecting the method best suited to their needs. Moreover, academic packages meant to investigate some aspects of an algorithm are usually poorly designed in terms of user interface and applicable only to a specific class of signals.

At the Biomedical Imaging Group (EPFL), we have taken upon ourselves to develop the freely available software package `DeconvolutionLab`[5] to experiment with 3D deconvolution microscopy. `DeconvolutionLab` is a software platform that hosts various algorithms and drives them through a unified and user-friendly interface. After ten years of experience with this package, we have revamped it and renamed it `DeconvolutionLab2`. This second version keeps the same key ingredients that made the success of the first version: Java source code, efficient FFT (fast Fourier transform), pluggable algorithms and an accommodating user interface.

2.1. `DeconvolutionLab`: The original `ImageJ` deconvolution tool

`DeconvolutionLab` was initially developed for educational purposes at EPFL. For over a decade it has been allowing students to conduct deconvolution experiments with the most representative classical algorithms, as well as with some more recent ones such as fast iterative soft-thresholding [11], Richardson–Lucy total variation [12], and thresholded Landweber [13]. Nowadays, we still train students with the help of `DeconvolutionLab`.

We have made `DeconvolutionLab` freely available since its release as an `ImageJ` plugin. As `ImageJ` is the *de facto* standard software tool of biological imaging, most biologists know how to install `DeconvolutionLab` on their own and can rapidly experiment with it. The package permits the deconvolution of large biological images at least as efficiently as commercial software packages [9]. With the passing years, our contribution has also gained popularity in several microscopic core facilities, where one of its favorite uses is for internal training. Moreover, from an academic perspective, `DeconvolutionLab` was deployed in more than seventy-five publications for various modalities (widefield, confocal [14], 2-photons [15], STED [16], light-sheet [14]). These works cover a wide range of applications, including neuroscience [15,17], osteology [16], microbiology [18], plant science [14] and material science [19].

2.2. `DeconvolutionLab2`: The remasterized `Java` deconvolution tool

The present paper focuses on the complete description of `DeconvolutionLab2`, the sequel to `DeconvolutionLab`. It is a freely accessible and open-source software package running on Windows, Linux, and Mac OS operating systems. The package can be linked to well-known imaging software platforms. The backbone of the software architecture is a library that contains the number-crunching elements of the deconvolution task. The current list of built-in algorithms includes:

Naive inverse filtering (NIF, Section 3.2);
Tikhonov regularization (TR, Section 3.3);
Regularized inverse filtering (RIF, Section 3.4);
Landweber (LW, Section 3.5);
Tikhonov–Miller (TM, Section 3.6);
Fast iterative soft-thresholding (FISTA, Section 3.7);
Richardson–Lucy (RL, Section 3.8);
Richardson–Lucy with total-variation regularization (RL-TV, Section 3.9).

New algorithms are easily pluggable into the framework of `DeconvolutionLab2`. The source code is written in Java 1.6, as closely as possible to the text-book definition of the algorithms.

[3] https://sites.google.com/site/piotrwendykier/software/deconvolution/paralleliterativedeconvolution.
[4] https://mitiv.univ-lyon1.fr/.
[5] http://bigwww.epfl.ch/deconvolution/.

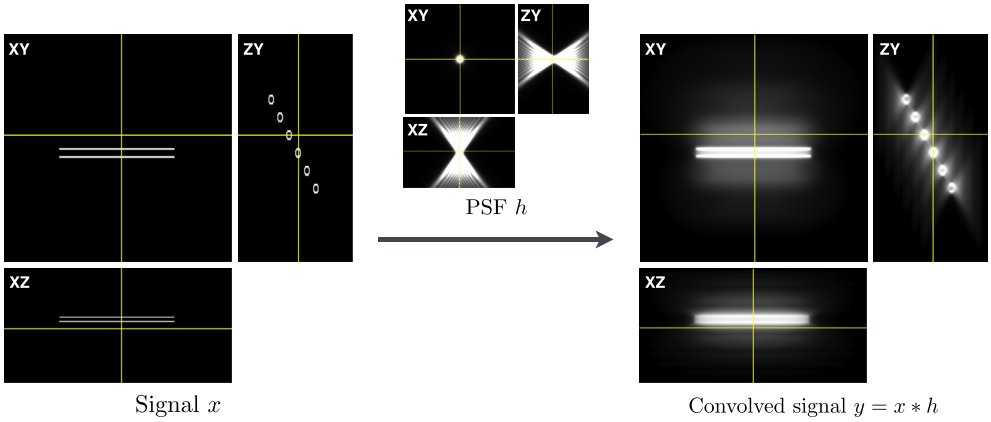


Fig. 2. Visualization of the convolution of simulated tubes with a PSF defined by the Born & Wolf model.

Inquisitive minds inclined to peruse the code will find it fosters the understanding of deconvolution.

Our goal with `DeconvolutionLab2` is to make deconvolution broadly accessible to the community of all those who show interest in this technique. To achieve such a goal, we provide a user-friendly front-end interface that also accommodates non-experts. Our software package is able to process large volumes on a mid-range desktop computer, or even on a laptop computer.

To experiment with the software, we share test data on the `DeconvolutionLab2` website. These data include synthetic and real cases to help benchmarking algorithms. `DeconvolutionLab2` can act not only as a didactic tool equipped with a simulator (convolution and noise generator), but also as a validation module that gives access to the signal-to-noise ratio between a ground-truth image and the output of every algorithm.

Like `DeconvolutionLab`, `DeconvolutionLab2` is able to process data relevant to real biological applications. However, and contrarily to the commercial software packages, our tools are restricted to deconvolution alone. We intentionally apply neither pre-processing nor postprocessing. Compared to `DeconvolutionLab`, `DeconvolutionLab2` includes new fast Fourier transform (FFT) libraries (see Appendix A.1), a recordable macro for ImageJ, new apodization functions, new padding schemes, and new switchable constraints in the space domain.

2.2.1. Practical details

`DeconvolutionLab2` is delivered as a plugin for ImageJ [20], for Fiji [21], and for the new bio-imaging platform Icy [22]. Since it is a Java class, it is also callable from the MATLAB command line and runnable as a standalone application through a Java Virtual Machine. For batch processing, we recommend calling `DeconvolutionLab2` from an ImageJ macro. This key feature enables one to handle time-lapse images or multiple channels, which need to be processed individually, in sequence.

Deconvolution is a heavy computational task in terms of running time and memory usage. In `DeconvolutionLab2`, we tried to find the best tradeoff between computational efficiency and code readability. The deconvolution is implemented in the discrete Fourier domain, so that the most time-consuming task is the FFT. Some iterative algorithms may require several FFT at every iteration, which can consume more than nine tenth of the runtime. Therefore, it is of utmost importance to rely on efficient FFT libraries.

3. Deconvolution algorithms

In this section, we recall the basic principles of image formation in fluorescence microscopy and give a brief technical description of the algorithms implemented in `DeconvolutionLab2`. We focus on the impact of the underlying models and the influence of the parameters. For an in-depth understanding and a more complete overview of the deconvolution field, we refer to the reviews [3,5,6,23] that cover most of the methods described here.

3.1. Image-formation model

Fluorescence microscopes are often assumed to be shift-invariant, which means that the response of the system does not depend on the position in the image. Therefore, they can be characterized by a PSF which approximates the distortions of the signal in the optical system. More elaborated approximations (e.g. spatially varying PSF) are described in Section 6.2). From a signal-processing point of view, the acquisition of images is modeled as the convolution of the fluorophore distribution x in the observed volume with the PSF h, followed by a degradation by noise. The convolution operation is defined at a given 3D location $p \in \mathbb{R}^3$ by

$$(x * h)(\boldsymbol{p}) = \int_{\mathbb{R}^3} x(\boldsymbol{r}) h(\boldsymbol{p} - \boldsymbol{r})\, \mathrm{d}\boldsymbol{r}. \tag{1}$$

In epifluorescence microscopy, the shape of the PSF in the image domain, shown in Fig. 2 with the Born and Wolf model [24,25], is typically such that it produces an anisotropic blurring of the signal. The resolution of the convolved signal is usually three times lower in the axial direction than in the lateral plane.

From now on we consider a discretized model. We denote by $\boldsymbol{y} \in \mathbb{R}^N$ the observed volume in vector form, $\boldsymbol{x} \in \mathbb{R}^K$ the underlying fluorescence signal, and $\mathbf{H} \in \mathbb{R}^{N \times K}$ the PSF matrix defined such that the discretization of the convolution defined in Eq. 1 writes as the matrix multiplication $\mathbf{H}\boldsymbol{x}$. Possibly, we may want to perform the reconstruction at an output resolution that differs from the input resolution, or to handle carefully border effects by estimating an image $\boldsymbol{x}$ with larger size, whereby $K \neq N$.

For a circulant and shift invariant discrete PSF with $K = N$, the matrix–vector multiplication $\mathbf{H}\boldsymbol{x}$ becomes an element-wise multiplication in the Fourier domain: $\hat{\boldsymbol{y}} = \hat{\boldsymbol{h}} \times \hat{\boldsymbol{x}}$ where $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{x}}$ are the discrete Fourier transform coefficients of $\boldsymbol{y}$ and $\boldsymbol{x}$, and $\hat{\boldsymbol{h}}$ are the coefficients of the discrete Fourier transform of the PSF. This permits efficient computation of $\mathbf{H}\boldsymbol{x}$, both in terms of speed and memory requirements through the use of the fast Fourier transform (FFT) algorithms. Every deconvolution algorithm we present in this paper relies on this technique.

The discrete image acquisition model is then

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{n}, \tag{2}$$

with $\boldsymbol{n} \in \mathbb{R}^N$ an additive noise component. The acquired images are affected by several sources of noise, which are often modeled by

two components. The first component is signal-dependent and models the fluctuation of the number of photons arriving at a given pixel. This so-called shot noise follows a Poisson distribution whose mean depends on the intensity of the incoming light. The second component accounts for various other distortions such as a background signal, read-out noise, or quantization noise, which are usually modeled as additive Gaussian noise. Note that in the case of Poisson noise, the variable $\boldsymbol{n}$ depends on the data $\boldsymbol{y}$ in Eq. (2). We decided to drop this dependency for the sake of clarity of the notations.

The aim of deconvolution algorithms is to invert the noisy convolution process defined in Eq. 2, thereby producing an estimated image $\tilde{\boldsymbol{x}}$ from the knowledge of $\boldsymbol{y}$ and $\mathbf{H}$, and the assumptions about the noise $\boldsymbol{n}$.

3.2. Naive inverse filtering

The simplest approach to deconvolution consists in minimizing a least-squares cost function $\mathscr{C}(\boldsymbol{x})$ that measures the similarity between the observation $\boldsymbol{y}$ and the current estimate $\mathbf{H}\boldsymbol{x}$, so that

$$\tilde{x} = \underset{x}{\operatorname{argmin}}\ \mathscr{C}(x) \tag{3}$$

$$\text{with } \mathscr{C}(x) = ||y - Hx||^2. \tag{4}$$

We call it naive inverse filtering. It corresponds to maximum-likelihood estimation in the presence of Gaussian noise. The solution can be computed efficiently in the Fourier domain and amounts to the coefficient-wise division

$$\hat{\tilde{\boldsymbol{x}}} = \frac{\hat{\boldsymbol{y}}}{\max\left(\hat{\boldsymbol{h}}, \epsilon\right)}, \tag{5}$$

where max denotes the element-wise maximum and ϵ is a small constant to avoid divisions by zero. The final solution is then obtained by taking the inverse Fourier transform of $\hat{\tilde{x}}$.

The method is parameter-free and the direct inversion in the Fourier domain leads to fast computations. Unfortunately, the NIF tends to amplify measurement noise, resulting in spurious high-frequency oscillations.

3.3. Tihkonov regularization

A way to avoid the noise amplification of NIF is to add to the cost function (4) the regularization term $\|\boldsymbol{x}\|_2^2$ to penalize high values of the solution [26]. This leads to

$$\mathscr{C}(\boldsymbol{x}) = \|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_2^2, \tag{6}$$

where λ is a regularization parameter that balances the contribution of the two terms. The explicit minimizer of (6) is

$$\boldsymbol{x} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\boldsymbol{y}, \tag{7}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{H}^T$ denotes the adjoint of $\mathbf{H}$. As for NIF, the solution (7) can be computed directly in the Fourier domain. This formulation can also be interpreted as a *maximum a posteriori* model. There, the regularization introduces prior information about the signal to guide the estimation.

3.4. Regularized inverse filtering

Other types of regularizations than TR can be used. A popular approach that performs well is to impose smoothness on $\boldsymbol{x}$ by penalizing the energy of its derivative. The resulting cost function is

$$\mathscr{C}(\boldsymbol{x}) = \|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|^2 + \lambda\|\mathbf{L}\boldsymbol{x}\|_2^2, \tag{8}$$

where $\mathbf{L}$ is a matrix that corresponds to the discretization of a differential operator. In `deconvolutionLab2`, we use the Laplacian operator. The explicit minimizer of (8) is given by

$$\boldsymbol{x} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{L}^T\mathbf{L}\right)^{-1}\mathbf{H}^T\boldsymbol{y}. \tag{9}$$

When the filtering by $\mathbf{L}^T\mathbf{L}$ has a whitening effect on $\boldsymbol{x}$ and λ is defined as the inverse of the noise variance, RIF amounts to Wiener filtering [27].

3.5. Landweber

The LW algorithm minimizes the same least-squares cost function than NIF. But, instead of expressing the solution through direct inversion, it resorts to an iterative gradient-descent approach [28]. In `DeconvolutionLab2`, we take advantage of the iterative nature of LW to impose a nonnegativity constraint at each iteration. Each update indexed by k can be written as

$$\boldsymbol{x}^{(k+1)} = \mathscr{P}_{(\mathbb{R}^+)^K}\left\{\boldsymbol{x}^{(k)} + \gamma\mathbf{H}^T\left(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^{(k)}\right)\right\}, \tag{10}$$

where γ is a step size parameter and $\mathscr{P}_{(\mathbb{R}^+)^K}\{\boldsymbol{x}\} = \max(\boldsymbol{x}, 0)$ is the component-wise projection of $\boldsymbol{x}$ onto the set $(\mathbb{R}^+)^K$.

Minimizing the energy (4) only enforces data fidelity of the result. The consequence is that the solution at convergence of iterations (10) tends to produce an over-fitting of the noise in the input data.However, one may obtain a satisfactory tradeoff between deconvolution and noise amplification if the algorithm is stopped early. In fact, the number of iterations is made to act as a pseudo regularization parameter. This phenomenon occurs for all maximum-likelihood based algorithms.

3.6. Tikhonov–Miller

Similarly with the LW method, the *TM* algorithm uses iterative gradient descent to minimize the regularized inverse filter cost (8). The iterative procedure is

$$\boldsymbol{x}^{(k+1)} = \mathscr{P}_{(\mathbb{R}^+)^K}\left\{\boldsymbol{x}^{(k)} + \gamma\left(\mathbf{H}^T\boldsymbol{y} - \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{L}^T\mathbf{L}\right)\boldsymbol{x}^{(k)}\right)\right\}. \tag{11}$$

When iterative projections onto the set $(\mathbb{R}^+)^K$ are performed, the method is sometimes referred to as iteratively constrained Tikhonov–Miller (ICTM).

3.7. Fast iterative soft-thresholing

Alternative regularization terms to the one in (8) can be considered. In particular, sparsity constraints in the wavelet domain have proven to yield better preservation of image details and discontinuities. The associated cost function is

$$\mathscr{C}(\boldsymbol{x}) = \|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|^2 + \lambda\|\mathbf{W}\boldsymbol{x}\|_1, \tag{12}$$

where $\mathbf{W}$ represents a wavelet transform. Due to the non-smoothness of the ℓ_1 norm, gradient-descent algorithms cannot be used. However, the problem can be solved efficiently by fast iterative soft-thresholding [11] with the following iterations:

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{s}^{(k)} - \gamma\mathbf{H}^T(\mathbf{H}\boldsymbol{s}^{(k)} - \boldsymbol{y}) \tag{13}$$

$$\boldsymbol{x}^{(k+1)} = \mathbf{W}^T\mathscr{T}(\mathbf{W}\boldsymbol{z}^{(k+1)}, \gamma\lambda), \tag{14}$$

$$p^{(k+1)} = \frac{1}{2}\left(1 + \sqrt{1 + 4p^{(k)^2}}\right) \tag{15}$$

$$\boldsymbol{s}^{(k+1)} = \boldsymbol{x}^{(k+1)} + \frac{p^{(k)} - 1}{p_{(k+1)}}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}). \tag{16}$$

There, γ is a step size that can be determined explicitly to ensure convergence [11], and $\mathscr{T}(\cdot,\tau)$ is a soft-thresholding operator with threshold τ.

3.8. Richardson–Lucy

The *RL* method [29,30] is a maximum-likelihood approach, like NIF. The difference is that RL assumes that the noise follows a Poisson distribution, which leads to

$$\mathscr{C}(\boldsymbol{x}) = \mathbf{1}^T\mathbf{H}\boldsymbol{x} - \boldsymbol{y}^T\log(\mathbf{H}\boldsymbol{x}), \tag{17}$$

where the log operation is applied component-wise and $\mathbf{1} = (1,\dots,\mathbf{1}) \in \mathbb{N}^N$. The iterative minimization of (17) can be understood as a multiplicative gradient descent and writes

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} \times \mathbf{H}^T\left(\frac{\boldsymbol{y}}{\mathbf{H}\boldsymbol{x}^{(k)}}\right), \tag{18}$$

where the multiplication $\times$ and the division $\boldsymbol{y}/(\mathbf{H}\boldsymbol{x}^{(k)})$ are understood to be component-wise.

Since the updates of $\boldsymbol{x}$ are multiplicative, nonnegativity is naturally ensured by the algorithm for any nonnegative starting point. As a maximum-likelihood method, the solution of RL is subject to the same noise-amplification problem as NIF and LW. Thus, the optimal number of iterations should be heuristically set to stop the algorithm before convergence.

3.9. Richardson–Lucy with total-variation regularization

To counterbalance the noise amplification effect of RL, a regularization term can be added to (17) [12]. The total-variation (TV) regularizer penalizes the ℓ_1 norm of the gradient of the signal, with

$$\mathscr{C}(\boldsymbol{x}) = \mathbf{1}^T\mathbf{H}\boldsymbol{x} - \boldsymbol{y}^T\log(\mathbf{H}\boldsymbol{x}) + \lambda\|\mathbf{D}\boldsymbol{x}\|_1. \tag{19}$$

There, $\mathbf{D}$ is the finite-difference matrix for first-order derivatives. In [12], a differentiable approximation of the ℓ_1 norm is used and the multiplicative iterations are expressed as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} \times \mathbf{H}^T\left[\frac{\boldsymbol{y}}{\mathbf{H}\boldsymbol{x}^{(k)}}\right] \times \frac{1}{1+\lambda\boldsymbol{g}^{(k)}}, \tag{20}$$

where $\boldsymbol{g}^{(k)}$ is the derivative of a regularized version of the l_1 norm of $\mathbf{D}\boldsymbol{x}^{(k)}$.

Compared to the ℓ_2 penalization used in (8), the ℓ_1 norm yields piecewise-constant results that better preserve image discontinuities.

4. Deconvolution in practice

4.1. Image acquisition

The preparation of samples and the design of the imaging system are of paramount importance to a successful deconvolution. In particular, it is critical to take into account elements of the imaging system such as calibration, sampling, and noise level. These practical issues have been well considered in the literature [4,31]. Specifically for deconvolution, it is also recommended to validate the acquisition and the further processing of *known* samples to avoid false interpretations, especially in the context of quantitative imaging assays [32,33].

4.2. Point-spread function

The quality of the deconvolution relies on the accuracy of the 3D PSF, which is the optical signature of an (ideally infinitesimally small) point. It is affected mostly by the objective, the medium, and the coverslip. A PSF can be obtained either experimentally or theoretically.

An experimental PSF can be deduced from the acquisition of the *z*-stack of a sparse field of spherical beads of very small diameter (e.g., (100 nm). Regions of interest are cut in the data centered around several well-contrasted beads and averaged. Microscopists generally agree that the experimental PSF captures well the aberrations of the microscope, but that the resolution of an experimental PSF is tied to the resolution of the acquisition. Unless super-resolution methods are deployed, this enforces $N = K$ (see Section 3.1).

By contrast, a theoretical PSF can be computed from a mathematical model. In addition to being able to lift the restriction on resolution, microscopists generally appreciate the convenience of software packages like `PSFGenerator`[6] that allow them to tune freely microscope parameters such as numerical aperture (NA), wavelength, and pixelsize [25].

Table 1
Important deconvolution parameters per method.

Method	Parameters	Section
NIF	–	3.2
TR	λ	3.3
RIF	λ	3.4
L	M_{iter}, γ	3.5
TM	$M_{\text{iter}}, \gamma, \lambda$	3.6
FISTA	$M_{\text{iter}}, \lambda, \gamma$	3.7
RL	M_{iter}	3.8
RL-TV	M_{iter}, λ	3.9

4.3. Setting of parameters

A few important parameters are shared by groups of deconvolution methods. In this section, we give practical hints about the meaning and the impact of the main parameters for each group. We provide the parameters-per-method associations in Table 1.

Regularization parameter λ. When the cost function contains a regularization term weighted by λ, the value of λ balances the contribution of data fidelity and regularization. For algorithms with Tikhonov-type regularization, higher values of λ result in smoother images. Finally, by setting $\lambda = 0$, TR and RIF become equivalent to NIF, and RL-TV becomes equivalent to RL.
Number M_{iter} **of iterations**. For all iterative methods, M_{iter} puts a cap on the number of iterations. How to set M_{iter} follows one of two rules: either the deconvolution method is known to reach the desired solution at convergence, in which case M_{iter} has to be chosen large enough; or noise amplification happens during convergence, in which case M_{iter} has to be chosen small enough so that the deconvolution procedure stops before noise dominates. In the latter case, the choice of the appropriate M_{iter} has a crucial impact on the result.
Stepsize. Methods based on gradient descent rely on a stepsize $\gamma \in (0, 1]$ which determines the speed of convergence. Small values of γ encourage safe but slow convergence.

4.3.1. Ghosts and ringing

Every deconvolution algorithm presented in Section 3 relies partly on circular convolutions computed through FFT. Compared to space-based approaches, Fourier-based approaches reduce the computational cost of handling a PSF that would have a wide spatial support. The downside is the appearance of Fourier-related artifacts such as ghosts and ringing.

[6] http://bigwww.epfl.ch/algorithms/psfgenerator/.

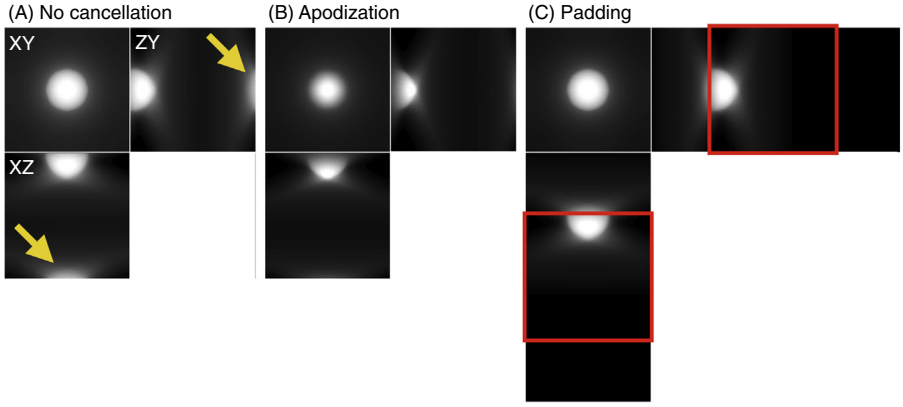


Fig. 3. Illustration of border artifacts after a deconvolution operation on a bead placed on the top of the volume of size 128 × 128 × 128 pixels. The illustrations are presented as orthogonal sections. (A) Deconvolution without artifact-cancellation processing was applied on the signal; the arrow shows the impact of ghosting. (B) Deconvolution with Hann apodization along the axial direction. (C) Deconvolution with a zero-padding extension to (128 × 128 × 256) pixels (only the red surrounding of the signal will be kept).

Data subjected to a FFT must necessarily be assumed to be periodic. This implies that borders at opposite sides of the image are implicitly abutting once periodization is taken into account. Consequently, structures near the border of an image, once processed, will spill over the opposite border, letting ghosts appear.

Data subjected to a FFT must necessarily be assumed to be bandlimited. This implies that the sharp transitions of intensities found in an image (i.e., the edges), once processed, will incur local overshoots and undershoots of intensity. This mechanism is called ringing. Nonnegativity constraints may help cancel this artifact, but only with regard to undershoots, and only for those undershoots that would otherwise result in negative values. Nonnegativity, commonly positivity, therefore makes a lot of sense in fluorescence microscopy.

Inconveniently, Fourier-related artifacts frequently appear, particularly in the axial direction since this direction is often sampled to a lesser extent than the lateral ones. For instance, if a biological cell physically extends outside of the bottom of the acquired volume and is thus virtually cropped at acquisition time, then a reverse ghost of the cell will appear on the top part of the volume after deconvolution. At the same time, ringing artifacts will reveal themselves as waves in the background and as Gibbs phenomena in the high-contrast areas.

To attenuate these artifacts, we have implemented two countermeasures in `DeconvolutionLab2`: apodization and zero-padding. Apodization consists in multiplying the input data by a window function that gradually sets the signal to zero near the borders of the image. Depending on the window specifics, the central part of the data may or may not remain pristine. In `DeconvolutionLab2`, we have made available the five classical apodization functions referred to as Cosine, Hamming, Hann, Tukey, and Welch. They can be applied independently along the axial and the lateral directions. As shown in Fig. 3(B), apodization succeeds in cancelling the ghost object, but also reduces the intensity of the data.

While it modifies the data, apodization proceeds without a change in the image dimensions. Conversely, zero-padding maintains the data intact but modifies the dimensions of the image by extending its border with zero values. For practical reasons related to the computational efficiency of the FFT, the width of the extension is generally chosen such that the size of the extended image is highly decomposable as a product of small prime numbers. To facilitate adherence to this constraint, `DeconvolutionLab2` automatically proposes extensions to the next even number, to the next multiple of 2 and 3, to the next multiple of 2, 3, and 5, and to the next power of 2, independently in the axial and the lateral directions. As shown in Fig. 3(C), zero-padding succeeds in cancelling the ghost object, but does so at an increased computational cost compared to apodization.

5. Experimental illustrations

We now illustrate the performance of `DeconvolutionLab2` and its built-in algorithms by restoring various types of degraded 3D images (i.e., synthetic volumes, beads, and real volumes). Visualizations of the deconvolution results are provided and quantitative measurements are reported when available. The data, as well as the corresponding model of the theoretical PSF, are available online[7].

5.1. Synthetic data

We applied all `DeconvolutionLab2` algorithms on a synthetically degraded volume. The ground-truth data consisted of a stack of 128 axial views of size 512 × 256 pixels depicting cellular microtubules. To mimic the acquisition artifacts of classical wide-field microscopes, blurring and noise were generated on the ground-truth volume through the *Convolution* tool of `DeconvolutionLab2`. More precisely, the 3D data was convolved with a theoretical PSF and a mixture of Gaussian and Poisson noise was added to the volume.

The effect of the deconvolution algorithms is illustrated in Figs. 4 and 5, while the quantitative measurements after deconvolution are reported in Table 2. The visual and quantitative outputs lead to similar observations.

Firstly and most obviously, the severe artifacts introduced by the NIF algorithm lead to non-exploitable results. The introduction of regularization (TR, RIF) enables decent deconvolution results, but the presence of undesirable ringing artifacts still hinder correct visualization of the imaged structure. As supported by Table 2, the beneficial effect of deconvolution increases when classical iterative algorithms (LW, RL, TM) are applied. However, the cost of doing so translates into an augmentation of the required computational resources.

Finally, the more advanced methods (FISTA, RL-TV) were also applied to the data. Interestingly, although RL-TV is theoretically more sophisticated than RL, the algorithm yields similar deconvolution results when applied to the present data. This can be explained by the fact that the structure of the considered object

[7] http://bigwww.epfl.ch/deconvolution/.

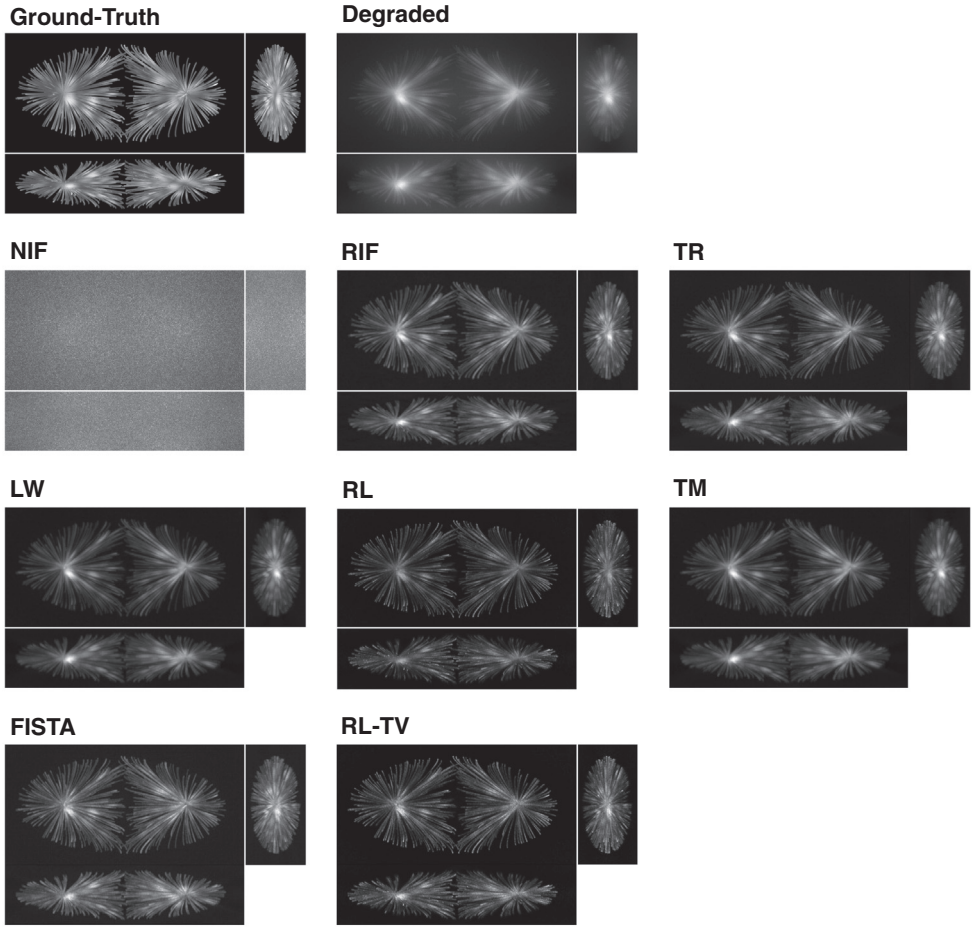


Fig. 4. Orthogonal sections of the maximum intensity projection (MIP) of a degraded 3D synthetic volume after its deconvolution by `DeconvolutionLab2` algorithms. From top left to bottom right: Ground-truth volume, Degraded volume (*i.e.*, simulated acquisition), Naive Inverse Filter, Tikhonov regularization (low regularization), Regularized Inverse Filter (low regularization), Landweber ($\tau = 1.0$, 2000 iterations), Richardson–Lucy (150 iterations), Tikhonov–Miller (low regularization, $\tau = 1.5$, 150 iterations), FISTA (low regularization, $\tau = 1.5$, 50 iterations), Richardson–Lucy with TV (low regularization, 150 iterations). The data, as well as the corresponding PSF, are available online. A non-negativity constraint was used for all algorithms. The setting of the optimal parameters for each deconvolution algorithm was performed through visual assessment.

imposes a negligible level of regularization during deconvolution. Indeed, the synthetic sample harbors thin filament-like structures which are difficult to recover through a TV regularizer, since TV tends to promote piece-wise constant surfaces. This illustrates the fact that the efficiency of a certain deconvolution algorithm may vary with the type of the data being processed. Thus, one cannot straightforwardly use the results presented above as a direct indicator of the individual performance of each deconvolution algorithm. Moreover, depending on the data size, time constraints and the available computational resources, some less advanced methods may be better suited for the deconvolution task at hand.

5.2. Isolated bead

We apply several algorithms of `DeconvolutionLab2` on a z-stack called "Bead" [9]. The volume displays a single fluorescent bead, which corresponds to a sphere with known diameter of 2.5 µm. The z-stack was acquired on a standard widefield microscope ($\lambda = 530$ nm,NA $= 1.4$); the lateral pixelsize is 64.5 nm and the stepsize in the axial direction is 160 nm.

The effect of the deconvolution algorithms is illustrated in Fig. 6, while the measurements of the full width at half maximum (FWHM) of the bead in the lateral and axial directions after deconvolution are reported in Table 3. We first observe that the NIF algorithm is not able to recover the bead. For the RIF algorithm, the effect of regularization on the deconvolution process becomes evident. Blurred images and overestimated dimensions are observed when the RIF regularization factor is overly increased, while setting it too low generates ringings. For the LW algorithm, the best results are obtained with 64 iterations. When the number of iterations is insufficiency, the effect of deconvolution is imperceptible. By contrast, using a too large number of iterations leads to high frequency artifacts appearing near the contour of the bead. This simple dataset thus illustrates the importance of the selection of the parameters for a given deconvolution method.

5.3. Widefield data

Finally, we briefly illustrate how `DeconvolutionLab2` may be used in a practical application to efficiently deconvolve real biomicroscopy data. We work with a 3D visualization of a *C. elegans* embryo which was acquired on a standard wide-field microscope ($\lambda = 542$ nm,NA $= 1.4$); the lateral pixelsize is 64.5 nm and the stepsize in the axial direction is 160 nm. As shown in Fig. 7, our 3D measurement displays some non-desirable visual features, such as a relatively poor contrast or an indistinguishability of certain neighboring centrosomes.

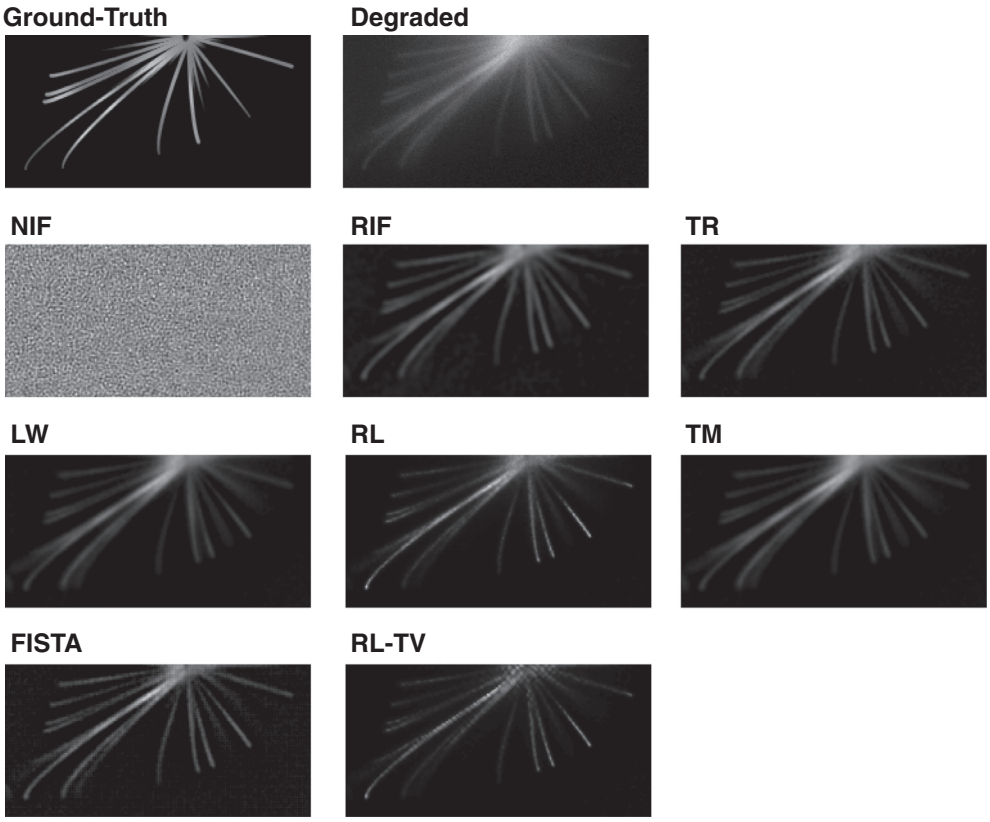


Fig. 5. Zooms on XY-views of a degraded synthetic volume after its deconvolution by `DeconvolutionLab2` algorithms. From top left to bottom right: Ground-truth volume, Degraded volume (*i.e.*, simulated acquisition), Naive Inverse Filter, Regularized Inverse Filter (low regularization), Tikhonov regularization (low regularization), Landweber ($\tau = 1.0$, 2000 iterations), Richardson–Lucy (150 iterations), Tikhonov–Miller (low regularization, $\tau = 1.5$, 150 iterations), FISTA (low regularization, $\tau = 1.5$, 50 iterations), Richardson–Lucy with TV (low regularization, 150 iterations). The data, as well as the corresponding PSF, are available online. The zoom corresponds to a cropping with positions (244, 128, 238, 119) on the 64th z-slice.A non-negativity constraint was used for all algorithms. The setting of the optimal parameters for each deconvolution algorithm was performed through visual assessment.

Table 2
Quality and computational efficiency of the `DeconvolutionLab2` algorithms for the deconvolution of degraded 3D synthetic data. For comparison, the results of a widely-used commercial software (Huygens) and L2D-A3D (the "Learn 2D, Apply 3D" method [34] that won "3D Deconvolution Microscopy" challenge) were also added into this table. To assess the deconvolution performance, the signal-to-noise ratio (SNR), the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) were computed after an initial normalization of the volumes in ImageJ. Indications of the computation time and the memory ratio values are reported to allow for comparison of the computational complexity of the available algorithms. The "Required RAM" is the peak of allocated memory to run the algorithm on an input dataset of 16,000,000 voxels. The "Memory Ratio" corresponds to the ratio between the required memory and the number of voxels of the input dataset. The deconvolution tasks were performed on a Mac OS X 2 × 3.06 GHz 6-Core Intel Xeon for all algorithms except for the Huygens software that was run on a 48-core server on Linux Red hat Entreprise

Algorithm	SNR [dB]	PSNR [dB]	SSIM [–]	Time [s]	Required RAM [Mb]	Memory Ratio
NIF	−75.45	−49.79	6.29e−9	7.6	258	×16.1
RIF	3.47	29.13	3.41e−2	7.0	322	×20.1
TR	2.78	28.45	2.48e−2	6.4	258	×16.1
LW	2.57	28.23	2.06e−2	2107	888	×55.5
FISTA	3.37	29.04	3.87e−2	1400	599	×37.4
TM	2.56	28.22	2.05e−2	2128	1016	×63.5
RL	3.66	29.33	3.30e−2	1661	258	×16.1
RL-TV	3.36	29.03	3.34e−2	2759	621	×38.8
Huygens (CMLE)	2.47	28.13	1.84e−2	180	n/a	n/a
L2D-A3D	7.27	32.94	6.73e−2	7200	n/a	n/a

To enhance the visual condition of this measurement, we apply three distinct deconvolution algorithms (TR, LW, RL) to it. The results after deconvolution are shown in Fig. 7. Globally, we observe similar effects than with the previous data sets. For all algorithms, the deconvolution permits a notable increase of the sharpness of the imaged structures and reduces out-of-focus blurring. Moreover, the iterative algorithms (LW, RL) yield better results than basic methods (RIF) at the cost of a more expensive computational need.

6. Discussion: trends in deconvolution

Similarly to many inverse problems, deconvolution requires one to express and minimize a cost function. As exemplified in Eqs. (6), (8), (12) and (19), the common form taken by this cost function is composed of a data-fidelity term that measures how well the model $\mathbf{H}\boldsymbol{x}$ represents the data $\boldsymbol{y}$, and a regularization function that enforces some priors. Deconvolution methods are thus characterized by three ingredients: (i) data-fidelity measure; (ii)

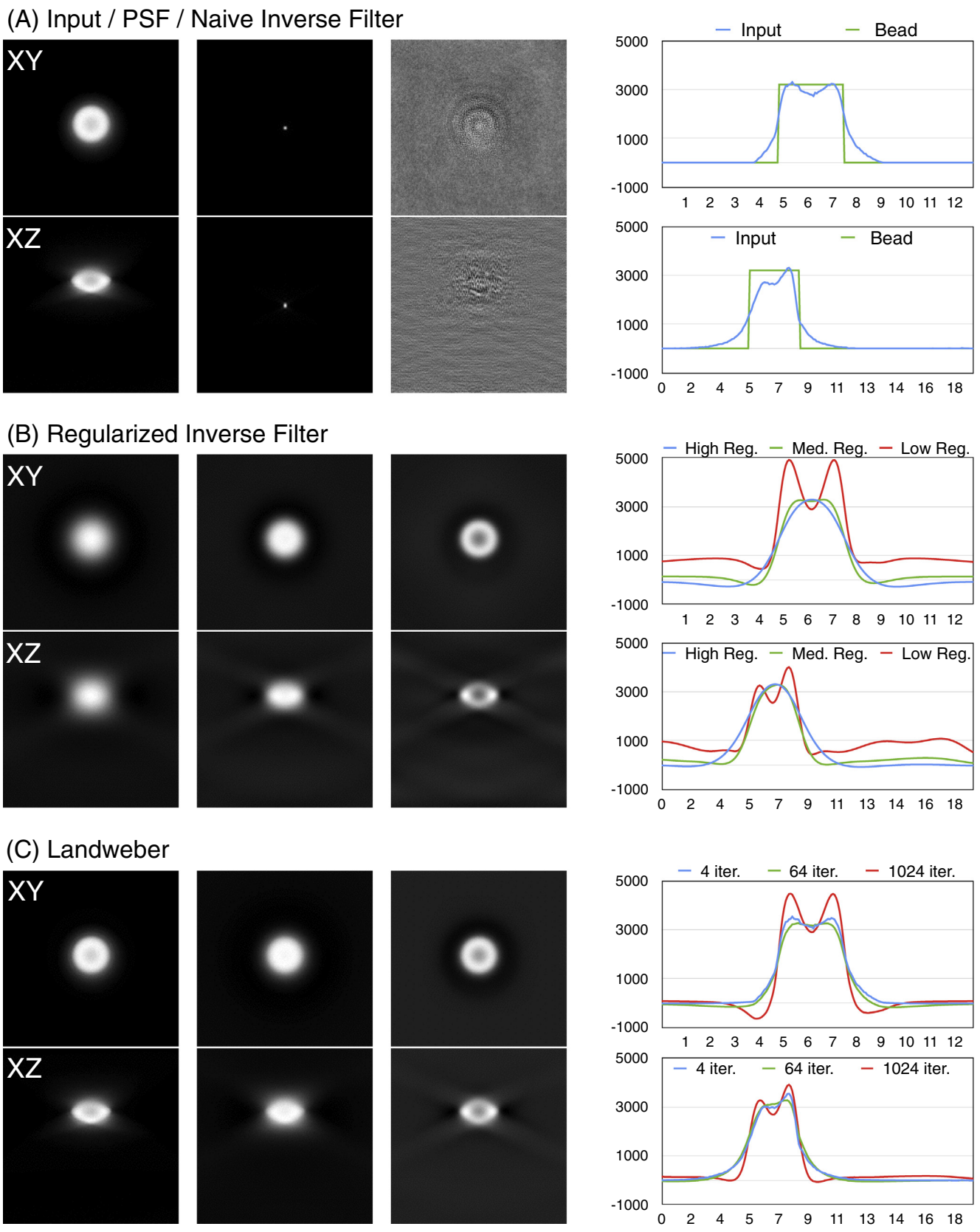


Fig. 6. Two orthogonal sections (XY and XZ) of the volumetric data before and after deconvolution. The plots show intensity profiles, the upper plot of a panel is the lateral profile trough the bead; the lower plot is the axial profile. The unit is μm. (A) From left to right: input image, PSF, and the result of the NIF algorithm. Plots show the intensity profile of the input (blue line) and theoretical shape of the bead (green line). (B) Results of the RIF algorithm with various settings. From left to right: low level of regularization (Low Reg.), medium level of regularization (Med Reg.), and high level of regularization (High reg.). (C) Results of the Landweber algorithm with various numbers of iterations. From left to right: 4 iterations, 64 iterations, and 1024 iterations.

Table 3
Lateral FWHM and axial FWHM of the bead measure on line profiles for the input image (upper row) and for various algorithms and settings.

Algorithm	Settings	Lateral FWHM [nm]	Axial FWHM [nm]
Acquisition		*2695.33*	*3979.46*
RIF	Reg: Low	2630	5909
RIF	Reg: Medium	2616	4881
RIF	Reg: High	2716	4900
Landweber	4 iterations	2714	4624
Landweber	64 iterations	2711	4777
Landweber	1024 iterations	2605	4449

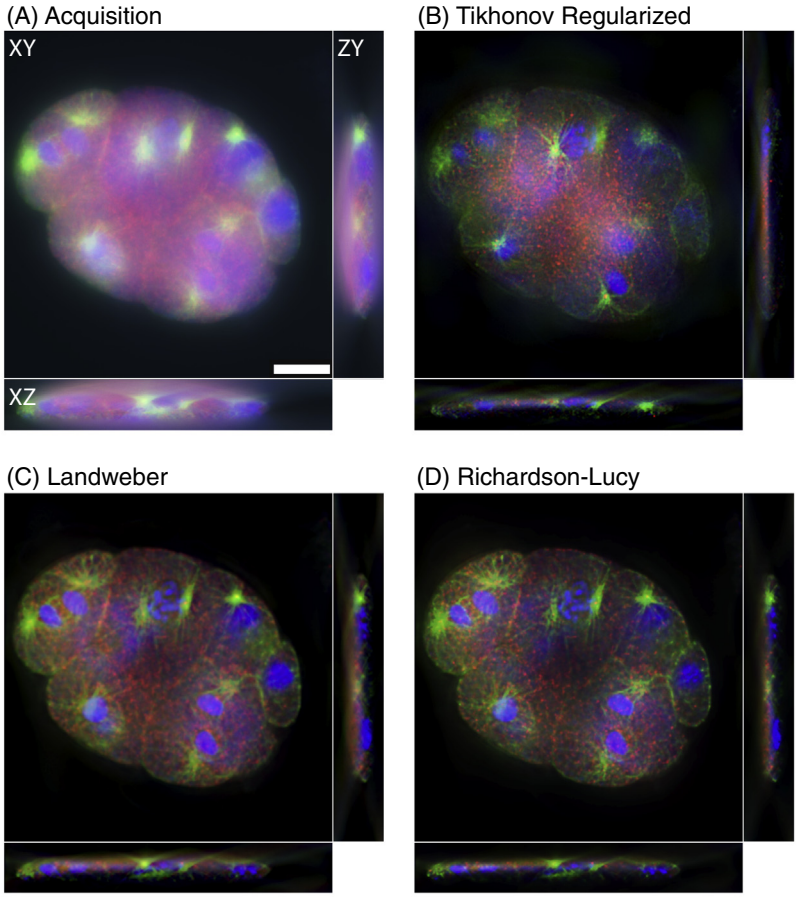


Fig. 7. Orthogonal sections of the *C. Elegens* volume (size: $672 \times 712 \times 104$ voxels). For better visualization, a Gamma correction have been applied to the images. Scale bar is $10\,\mu$m. The data are available online. (A) Acquisition. (B) Tikhonov Regularized. (C) Landweber deconvolution (200 iterations). (D) Richardson Lucy deconvolution (200 iterations). Advanced iterative algorithms permit better distinction between two neighboring centrosomes.

regularization prior; and (iii) minimization algorithm. The impact of each block is quite independent, so that improvements can be devised separately. Typically, one can:

upgrade the data-fidelity term by devising a more precise image-formation model and by gaining and taking advantage of a deeper knowledge of the statistics of the measurement noise;
use prior-promoting regularizers that fit the object better;
deploy robuster and faster optimization schemes.

These three topics are shared by all inverse problems. Deconvolution microscopy can benefit from every improvements in this currently very active field of research.

The priors introduced by the regularizer must be chosen carefully to retain usefulness while avoiding the pitfall of overfitting. During the last decade, the compressive-sensing and the sparsity theories gave theoretical grounds to the observation that the ℓ_1-based regularizers of (8) and (19) in Sections 3.7 and 3.9, respectively, always perform better than the ℓ_2-based regularizers of Eqs. (6) and (8) in Sections 3.3 and 3.4, respectively.

Out of a dozen of competing methods, the methods that ranked first [34] and third [35] in the "2014 International Challenges on 3D Deconvolution Microscopy" took advantage of regularizers that were based on recent advances in signal processing[8]. The authors of

[8] http://bigwww.epfl.ch/deconvolution/challenge/.

[35] bring to fruition a second-order total-variation regularizer called a Schatten norm, while the method "Learn 2D Apply 3D" of [34] did exploit the fact that the resolution is much better within the lateral sections than along the axial direction. Assuming that the structures of interest are isotropic, it learned from the lateral sections of the acquired volume a dictionary of 2D high-resolution features that are used as priors to enhance the resolution along the axial direction. Approaches where the priors are learned appear to be very efficient; we surmise that the recent successes of deep neural networks in machine learning will soon lead to improved deconvolution algorithms [36] in microscopy. Finally, many modern deconvolution methods rely on state-of-the-art optimization schemes that can deal with non-differentiable ℓ_1 functions, for instance on proximal algorithms such as the alternating-direction method of multipliers [37].

Up to now in this paper, we have assumed that the PSF was known, either through ancillary measurements or through modeling. Moreover, we have assumed spatial shift-invariance of the systems. We now present approaches that have been recently developed to handle imaging situations in which these assumptions are not met.

6.1. Blind deconvolution

Blind deconvolution attempts to jointly estimate the object x and the PSF h from the data alone, without relying on ancillary measurements. It is a challenging, strongly ill-posed, and nonlinear problem. As an example, among other degeneracy issues [38], it must address that of scale, characterized by $(\alpha h) * (\frac{1}{\alpha} x) = h * x$ for any non-vanishing α. As it turns out, setting a meaningful value to $||h||$ is highly nontrivial. Some proposed methods are explicitly designed to overcome degeneracies (scale and others) using an optically motivated parameterization of the PSF [39–41] or estimating the PSF from a dictionary [42]. Currently, the trend followed by all blind-deconvolution algorithms for fluorescence microscopy is to resort to iteratively alternating between the deconvolution and the estimation of the PSF [39–45].

6.2. Space-varying deconvolution

The deconvolution of large micrographs faces an important issue: in practice, the PSF varies across the field of view. In particular, a depth-varying PSF is often induced by a refractive index mismatch between the immersion medium and the specimen. In this case, the PSF suffers of spherical aberrations that get stronger as the focal plan is deeper inside the sample. This effect can be clearly seen on the Fig. 7(A) where the image is sharper at the bottom where the objective is closer to the sample.

Space-varying deconvolution raises two important problems. First, the assumption that the PSF varies across the field of view implies that the blurring process can no longer be modeled as a *convolution*. Hence, space-varying deconvolution is an oxymoron. As a consequence, FFT-based algorithm can no longer be used. The computational cost of space-varying deconvolution tends to rise as the square of the number of voxels. However using some approximations, several fast methods to model space-varying convolution have been proposed (see [46] for a review). In the refractive index mismatch case, as the size of the data along the depth axis is usually much smaller than along lateral axes, a depth only varying deblurring algorithm is much more tractable and several methods have been proposed in that case [47–50].

The second issue raised by the space varying deconvolution is how to estimate the PSF variation across the 3D object. With the exception of the case of refractive index mismatch where the PSF depth variation can be analytically known, one has to infer the PSFs from the data in a space varying blind deconvolution algorithm. Up to now, only one attempt [41] has been done in that direction.

Acknowledgement

The authors thank warmly Dr. Philippe Thévenaz for his useful feedback on the paper. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 692726 "GlobalBioIm: Global integrative framework for computational bio-imaging").

Appendix A. Implementation aspects

A.1. FFT Libraries

The algorithms that we have proposed made an extensive usage of the fast Fourier transform (FFT). For instance, one iteration of the Richardson–Lucy algorithm is composed of two multiplications in the Fourier domain (74 ms), a division in the space domain (51 ms), an application of the non-negativity constraint (6 ms), and two FFT/FFT^{-1} (2'520 ms). The FFT/FFT^{-1} are representing 95% of the computational time of this algorithm see (Table 4). `DeconvolutionLab2` has a Java wrapper for three FFT libraries.

- AcademicFFT[9]. This is pure Java library running on any platform. The source code of AcademicFFT is bundled with `DeconvolutionLab2`. It handles arbitrary data lengths, memory permitting. It is standalone; no external library is required.
- JTransforms[10]. This is the first, open-source, fast multithreaded FFT library written in pure Java. It is bundled with Fiji and Icy, but JTransforms is not part of the classical distribution of ImageJ.
- FFTW 2.0[11] [51]. FFTW is a C routine library for computing the fast Fourier transform in several dimensions, of arbitrary input size, and of both real and complex data. FFTW is one of the fastest FFT library. `DeconvolutionLab2` includes a wrapper for FFTW version 2.0 and it includes pre-compiled binaries for Mac OS X and Windows 32-bits and 64-bits.

A.2. Dissection of an algorithm

We present a complete iterative algorithm from its mathematical formula to its Java snippet. Here, we choose to detail the Landweber algorithm Section 3.5.

We first reformulate the iteration to reduce the number of operations in the discrete Fourier domain and to limit the memory consumption.

A.2.1. Implementation of the Landweber algorithm

The original formulation is reduced to one multiplication and one addition in the discrete Fourier domain for every iterations.

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \gamma \mathbf{H}^T \left(\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^k\right) \tag{21}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \gamma \mathbf{H}^T \mathbf{H}\boldsymbol{x}^k + \gamma \mathbf{H}^T \boldsymbol{y} \tag{22}$$

$$\boldsymbol{x}^{k+1} = \left(\mathbf{I} - \gamma \mathbf{H}^T \mathbf{H}\right)\boldsymbol{x}^k + \gamma \mathbf{H}^T \boldsymbol{y} \tag{23}$$

$$\boldsymbol{x}^{k+1} = \mathbf{A}\boldsymbol{x}^k + \boldsymbol{g} \tag{24}$$

Using this expression, the variables $\mathbf{A}$ and $\boldsymbol{g}$ can be precomputed.

[9] http://bigwww.epfl.ch/thevenaz/academicfft/.
[10] https://sites.google.com/site/piotrwendykier/software/jtransforms.
[11] http://www.fftw.org/.

Table 4
Computation time for a FFT and FFT^{-1} for a volume of size $N \times N \times N$. This experiment was performed on a Mac OS X 2.5 GHz Intel Core i7.

N (size)	FFTW2 [ms]	JTransforms [ms]	AcademicFFT [ms]
$32 \times 32 \times 32$	1.5	9.8	11.5
$37 \times 37 \times 37$	13.3	30.3	17.2
$56 \times 56 \times 56$	9.6	12.8	34.8
$64 \times 64 \times 64$	17.1	23.5	38.6
$74 \times 74 \times 74$	101.2	61.8	111.0
$111 \times 111 \times 111$	353.9	189.1	324.0
$128 \times 128 \times 128$	247.4	151.9	577.9
$147 \times 147 \times 147$	347.4	243.3	620.8
$223 \times 223 \times 223$	7200.0	1615.4	4910.0
$256 \times 256 \times 256$	2937.7	1743.9	7860.0
$294 \times 294 \times 294$	3090.0	2197.7	11,200.0
$446 \times 446 \times 446$	62,200.0	46,700.0	61,100.0
$512 \times 512 \times 512$	35,000.0	25,900.0	141,000.0

$$\mathbf{A} = \left(\mathbf{I} - \gamma \mathbf{H}^T \mathbf{H}\right) \tag{25}$$

$$\boldsymbol{g} = \gamma \mathbf{H}^T \boldsymbol{y} \tag{26}$$

A.2.2. Java snippet of Landweber

We choose the Java code of the Landweber algorithm. The iteration mechanism is handled by the object `controller` which is an instance of the class `Controller`. The instance of the Java FFT wrapper class is `fft` that contains two methods `transform()` and `inverse()`. The Java classes `ComplexSignal` and `RealSignal` are two classes of `DeconvolutionLab2` to store complex 3D signals and real 3D signals, respectively. The input variables are the two `RealSignal` objects, `input` and `psf` and the scalar parameter `gamma` which is the step parameter of the Landweber algorithm.

Landweber algorithm

```
// RealSignal y: this is the input volume to deconvolve
// RealSignal h: this is the PSF volume
// RealSignal x: this is the output deconvolved volume
// Operations.delta() is a high-level method to compute
   (I- gamma Ht H) public RealSignal call() {ComplexSignal Y
   = fft.transform(y);
   ComplexSignal H  = fft.transform(h);
   ComplexSignal A  = Operations.delta(gamma, H);
   ComplexSignal G  = Operations.multiplyConjugate(gamma,
   H, Y);
   ComplexSignal X  = G.duplicate();
while(!controller.ends(X)) {X.times(A);
     X.plus(G);
     constraint(X);}
   RealSignal x  = fft.inverse(X);
   return x;}
```

References

[1] S. Inoué, Foundations of confocal scanned imaging in light microscopy, Handbook of biological confocal microscopy, Springer, 2006, pp. 1–19.
[2] D.A. Agard, J.W. Sedat, Three-dimensional architecture of a polytene nucleus, Nature 302 (5910) (1983) 676–681.
[3] J.G. McNally, T. Karpova, J. Cooper, J.A. Conchello, Three-dimensional imaging by deconvolution microscopy, Methods 19 (3) (1999) 373–385.
[4] W. Wallace, L.H. Schaefer, J.R. Swedlow, A workingperson's guide to deconvolution in light microscopy, Biotechniques 31 (5) (2001) 1076–1097.
[5] J. Sibarita, Deconvolution microscopy, Microsc. Tech. (2005) 1288–1291.
[6] P. Sarder, A. Nehorai, Deconvolution methods for 3D fluorescence microscopy images, IEEE Signal Processing Mag. 23 (3) (2006) 32–45.
[7] M. Bertero, P. Boccacci, G. Desiderà, G. Vicidomini, Image deblurring with poisson data: from cells to galaxies, Inverse Prob. 25 (12) (2009) 123006.
[8] A. Griffa, N. Garin, D. Sage, Comparison of deconvolution software in 3D microscopy. a user point of view part 1, G.I.T., Imag. Microsc. 1 (2010) 43–45.
[9] A. Griffa, N. Garin, D. Sage, Comparison of deconvolution software in 3D microscopy. a user point of view part 2, G.I.T., Imag. Microsc. 1 (2010) 41–43.
[10] A. Ponti, P. Schwarb, A. Gulati, V. Bäcker, Huygens remote manager, Imag. Microsc. 9 (2) (2007) 57–58.
[11] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (1) (2009) 183–202.
[12] N. Dey, L. Blanc-Féraud, C. Zimmer, Z. Kam, P. Roux, J. Olivo-Marin, J. Zerubia, Richardson–Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution, Microsc. Res. Tech. 69 (2006) 260–266.
[13] C. Vonesch, M. Unser, A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution, IEEE Trans. Image Process. 17 (4) (2008) 539–549.
[14] V. Smékalová, I. Luptovčiak, G. Komis, O. Šamajová, M. Ovečka, A. Doskočilová, T. Takáč, P. Vadovič, O. Novák, T. Pechan, et al., Involvement of YODA and mitogen activated protein kinase 6 in arabidopsis post-embryogenic root development through auxin up-regulation and cell division plane orientation, New Phytologist 203 (4) (2014) 1175–1193.
[15] R. Chen, J. Zhang, Y. Wu, D. Wang, G. Feng, Y.-P. Tang, Z. Teng, C. Chen, Monoacylglycerol lipase is a therapeutic target for alzheimer's disease, Cell Rep. 2 (5) (2012) 1329–1339.
[16] T. Deguchi, M.H. Alanne, E. Fazeli, K.M. Fagerlund, P. Pennanen, P. Lehenkari, P. E. Hänninen, J. Peltonen, T. Näreoja, In vitro model of bone to facilitate measurement of adhesion forces and super-resolution imaging of osteoclasts, Sci. Rep. 6 (2016) 22585.
[17] M.M. Dobbin, R. Madabhushi, L. Pan, Y. Chen, D. Kim, J. Gao, B. Ahanonu, P.-C. Pao, Y. Qiu, Y. Zhao, et al., SIRT1 collaborates with ATM and HDAC1 to maintain genomic stability in neurons, Nat. Neurosci. 16 (8) (2013) 1008–1015.
[18] M. Kicia, N. Janeczko, J. Lewicka, A.B. Hendrich, Comparison of the effects of subinhibitory concentrations of ciprofloxacin and colistin on the morphology of cardiolipin domains in escherichia coli membranes, J. Med. Microbiol. 61 (4) (2012) 520–524.
[19] M. Padilla, B. Michl, B. Thaidigsmann, W. Warta, M. Schubert, Short-circuit current density mapping for solar cells, Solar Energy Mater. Solar Cells 120 (2014) 282–288.
[20] C.A. Schneider, W.S. Rasband, K.W. Eliceiri, NIH Image to ImageJ: 25 years of image analysis, Nat. Methods 9 (7) (2012) 671–675.
[21] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al., Fiji: an open-source platform for biological-image analysis, Nat. Methods 9 (7) (2012) 676–682.
[22] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, et al., Icy: an open bioimage informatics platform for extended reproducible research, Nat. Methods 9 (7) (2012) 690–696.
[23] J. Starck, E. Pantin, F. Murtagh, Deconvolution in astronomy: A review, Publ. Astron. Soc. Pac. 114 (800) (2002) 1051.
[24] M. Born, E. Wolf, Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light, Pergamon Press, 1956.
[25] H. Kirshner, F. Aguet, D. Sage, M. Unser, 3-D PSF fitting for fluorescence microscopy: implementation and localization application, J. Microsc. 249 (1) (2013) 13–25.
[26] A. Tikhonov, Solution of incorrectly formulated problems and the regularization method, Soviet Mathematics Dokl., vol. 5, 1963, pp. 1035–1038.
[27] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series, vol. 2, MIT press Cambridge, 1949.
[28] L. Landweber, An iteration formula for fredholm integral equations of the first kind, Am. J. Math. 73 (3) (1951) 615–624.
[29] W.H. Richardson, Bayesian-based iterative method of image restoration, J. Optical Soc. Am. 62 (1972) 55–59.
[30] L.B. Lucy, An iterative technique for the rectification of observed distributions, Astrophys. J. 79 (6) (1974) 745–754.
[31] J.R. Swedlow, M. Platani, Live cell imaging using wide-field microscopy and deconvolution, Cell Struct. Funct. 27 (5) (2002) 335–341.
[32] J.R. Swedlow, Quantitative fluorescence microscopy and image deconvolution, Methods Cell Biol. 81 (2007) 447–465.
[33] J.-S. Lee, T. Wee, C.M. Brown, Calibration of wide-field deconvolution microscopy for quantitative fluorescence imaging, J. Biomol. Technol. 25 (2014) 31–40.
[34] F. Soulez, A learn 2D, apply 3D method for 3D deconvolution microscopy, IEEE 11th International Symposium on Biomedical Imaging (ISBI), IEEE, 2014, pp. 1075–1078.
[35] S. Lefkimmiatis, M. Unser, Poisson image reconstruction with hessian Schatten-norm regularization, IEEE Trans. Image Process. 22 (11) (2013) 4314–4327.
[36] L. Xu, J.S. Ren, C. Liu, J. Jia, Deep convolutional neural network for image deconvolution, Advances in Neural Information Processing Systems, 2014, pp. 1790–1798.
[37] N. Parikh, S.P. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239.
[38] F. Soulez, M. Unser, Superresolution with optically motivated blind deconvolution, in: Computational Optical Sensing and Imaging, Optical Soc. Am., 2016. pp. JT3A–38.
[39] J. Markham, J.-A. Conchello, Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur, J. Opt. Soc. Am. A 16 (10) (1999) 2377–2391.

[40] F. Soulez, L. Denis, Y. Tourneur, E. Thiébaut, Blind deconvolution of 3D data in wide field fluorescence microscopy, 9th IEEE International Symposium on Biomedical Imaging (ISBI), IEEE, 2012, pp. 1735–1738.
[41] B. Kim, T. Naemura, Blind depth-variant deconvolution of 3D data in wide-field fluorescence microscopy, Sci. Rep. 5 (2015). 9894 EP.
[42] T. Kenig, Z. Kam, A. Feuer, Blind image deconvolution using machine learning for three-dimensional microscopy, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2191–2204.
[43] L.M. Mugnier, T. Fusco, J.-M. Conan, MISTRAL: a myopic edge-preserving image restoration method, with application to astronomical adaptive-optics-corrected long-exposure images, J. Opt. Soc. Am. A 21 (10) (2004) 1841–1854.
[44] E.F.Y. Hom, F. Marchis, T.K. Lee, S. Haase, D.A. Agard, J.W. Sedat, AIDA: an adaptive image deconvolution algorithm with application to multi-frame and three-dimensional data, J. Opt. Soc. Am. A 24 (6) (2007) 1580–1600.
[45] M. Keuper, T. Schmidt, M. Temerinac-Ott, J. Padeken, P. Heun, O. Ronneberger, T. Brox, Blind deconvolution of widefield fluorescence microscopic data by regularization of the optical transfer function (otf), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2179–2186.
[46] L. Denis, E. Thiébaut, F. Soulez, J.-M. Becker, R. Mourya, Fast approximations of shift-variant blur, Int. J. Comput. Vision 115 (3) (2015) 253–278.
[47] M. Arigovindan, J. Shaevitz, J. McGowan, J.W. Sedat, D.A. Agard, A parallel product convolution approach for representing the depth varying point spread functions in 3D widefield microscopy based on principal component analysis, Opt. Express 18 (7) (2010) 6461–6476.
[48] N. Patwary, C. Preza, Image restoration for three-dimensional fluorescence microscopy using an orthonormal basis for efficient representation of depth-variant point-spread functions, Biomed. Opt. Express 6 (10) (2015) 3826–3841.
[49] E. Maalouf, B. Colicchio, A. Dieterlen, Fluorescence microscopy three-dimensional depth variant point spread function interpolation using zernike moments, J. Opt. Soc. Am. A 28 (9) (2011) 1864–1870.
[50] S. Ben Hadj, L. Blanc-Féraud, G. Aubert, Space variant blind image restoration, SIAM J. Imag. Sci. 7 (4) (2014) 2196–2225.
[51] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2) (2005) 216–231. special issue on Program Generation, Optimization, and Platform Adaptation.