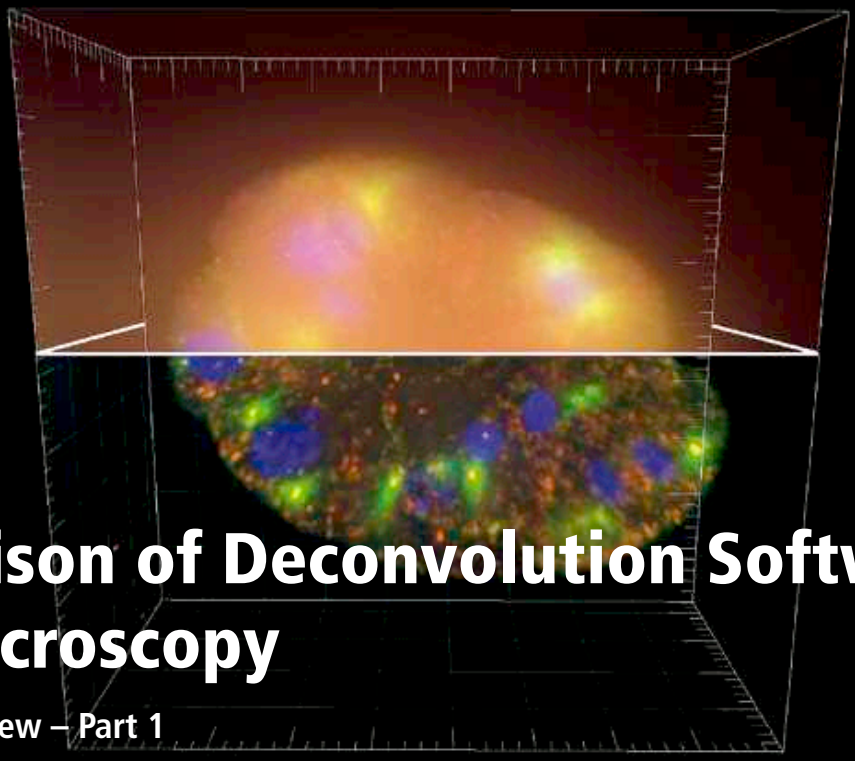

Comparison of Deconvolution Software in 3D Microscopy

A User Point of View – Part 1

Deconvolution is an image restoration technique which improves image contrast, resolution and signal to noise ratio. In modern optical microscopy and biological research deconvolution is becoming a fundamental processing step which allows for better image analysis. Deconvolution remains however a challenging task as the result depends strongly on the algorithm chosen, the parameters settings and the kinds of structures in the processed dataset. As a core facility of bio-imaging and microscopy, we aim with this study to compare the performances of different deconvolution software. In this first part of our survey we present deconvolution related problems, we introduce software we took into account, and we provide the complete dataset we produced for software testing and a PSF generator. A second part will follow the present one. In the second part we will highlight advantages and weak points of tested software by the statement of the performed tests.

Introduction

Deconvolution is an image processing technique that restores the effective object representation. Deconvolution algorithms have applications in astronomy, physics, material science, medicine and biology. As a microscopy and optics core facility for biological research, we focus on the restoration of microscopy images. In modern biological research, deconvolution is becoming not only a fundamental, but almost a standard image processing step when analyzing small relevant details. For example, deconvolution can reveal hidden pertinent structures and can improve segmentation with the amelioration of contrast [1]. It is also recommended when doing colocalization analysis [2].

The acquired images differ from the true object since they are unavoidably affected by noise and convolution effects due to the optical system. The optical blur is basically linked to the diffraction-limited nature of the acquisition system, and the resulting distortion of a point source can be specified by the point spread function (PSF). The noise introduced in the image derives both from the digital sensor imprecision and from the inherent statistical behavior of light. The latter is predominant and can be modeled through Poisson statistics. Conceptually, a deconvolution algorithm de-blurs the image by eliminating the out-of-focus light contributions to each voxel intensity value, therefore achieving the most significant resolution and contrast improvement in the third dimension [5, 6], provided the image has been correctly acquired [3, 4].

Various commercial and open-source software packages are available for deconvolution. They are computationally extensive, requiring high-end processors and huge memory capacities as deconvolution is mostly applied to large multi-dimensional datasets. Despite the effort to provide user-friendly solutions, deconvolution remains a challenging task in choosing the good software, the right algorithm and the correct settings.

In this context we aim to work out a performance evaluation and comparison of different tools through a solid working guideline in terms of deconvolution parameters settings and result evaluation.

This study does not consider fast filtering solutions and concentrates on iterative algorithms only, as one can expect better results from iterative techniques. Blind deconvolution is also excluded because it is conceptually unrelated to the other methods as it does not use an *a priori* defined PSF.

We performed several deconvolution tests on different kinds of datasets. Methodology is reported in the following. Results will be exposed in the second part of this survey.

Materials and Methods:

Software Presentation

We took into account two classes of packages, the commercial ones and the open source ones.

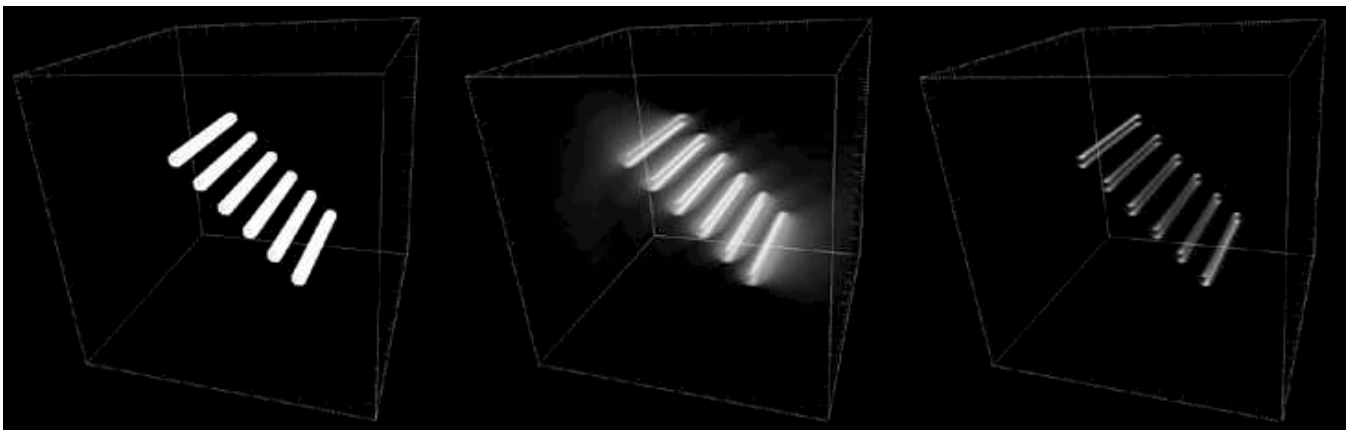

Fig. 1: synthetic image, six parallel hollow bars. Dimensions 256 X 256 X 256 voxels, 16 MB, 16 bit dynamic range. In sequence: original image, volume convolved with theoretical PSF and corrupted by Gaussian noise (σ=15) and Poisson noise (SNR = 30), and deconvolution result (*DeconvolutionLab*, Gold algorithm, 40 iterations).

For the commercial deconvolution software, we chose *SVI HuygensPro* (www.svi.nl/) and *MediaCybernetics AutoDeblur* (www.mediacy.com/).

HuygensPro and *AutoDeblur* implement different iterative algorithms, among which the most popular are the ones based on the maximum likelihood estimation. Useful pre-processing modules are available in both packages, such as bleaching and spherical aberration correction and background suppression. The user interfaces are intuitive and the parameters settings for the deconvolution are quite similar. The basic parameters to be set are the number of iterations and the variable linked to the amount of noise in the image, that is, to the degree of regularization of the result. It is also necessary to provide the right PSF; alternatively *HuygensPro* can compute a theoretical PSF while *AutoDeblur* incorporates a blind deconvolution method.

Among the open source solutions, we found various plugins for ImageJ (http://rsb.info.nih.gov/ij/), a public domain, Java-based image processing and analysis software developed at the National Institutes of Health. Typically these plugins do not implement a variety of pre-deconvolution processing steps and because they do not compute theoretical PSFs, the user must provide an external one. As a general comment, the approach of these tools is more technical and specifically addressed to an expert user in comparison to commercial software. *Parallel Iterative Deconvolution* by P. Wendykier (http://sites.google.com/site/piotr.wendykier/software/deconvolution/paralleliterativedeconvolution/) and *DeconvolutionLab* (http://bigwww.epfl.ch/deconvolution/) implement different least square algorithms, while *Iterative Deconvolve 3D* by B. Dougherty (www.optinav.com/Iterative-Deconvolve-3D.htm/) adopts an iterative implementation of the Wiener filter.

Concerning software portability, *HuygensPro* runs on Windows, Linux and Mac OS X platforms. Moreover, IRIX and Itanium versions of HuygensPro are available upon request. *AutoDeblur* can only run on Windows. ImageJ plugins run on Windows, Linux and Mac OS X platforms.

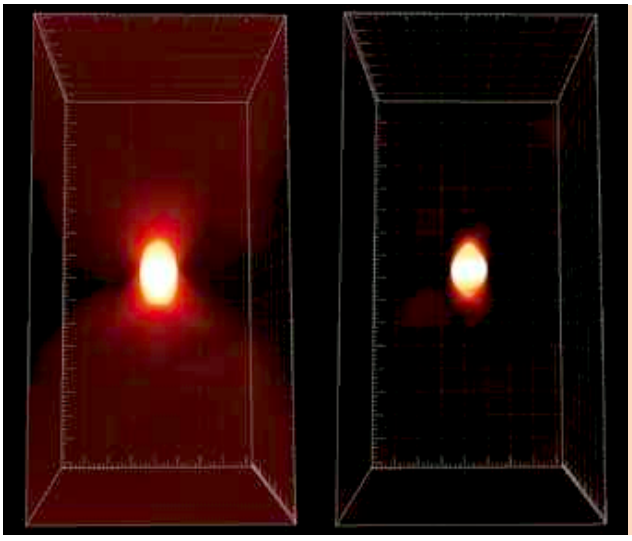

Fig. 2: InSpeck green fluorescent bead, diameter 2.5 µm. Widefield image, Olympus Cell R, 63X 1.4NA oil objective. Voxel size 64.5 X 64.5 X 160 nm; dimensions 256 X 256 X 256 voxels, 32 MB, 16 bit dynamic range. Aquired image (on the left) and deconvolution result (on the right; *Auto Deblur,* 40 iterations).

General Guidelines

A fair comparison of different deconvolution software requires a well-defined protocol taking into account the large variety of algorithms and the high dependency of results on the optimization of parameters, on the type of object structures and on the level of noise.

The studied packages implement several deconvolution algorithms differently, meaning that the user must set different parameters for each package. Moreover, the various tools allow different levels of control on the algorithms variables. Therefore, to get comparable deconvolution results, we simplified the tests as much as possible.

Deconvolutions on different kinds of images were run in the same conditions, by limiting the number of parameters to be optimized. For each test image, the same PSF was used with all the considered software. Pre-processing of the datasets, such as smoothing or background correction, was avoided even if directly integrated in the software as normal pre-deconvolution steps. Our goal was to compare deconvolution results and not the pre-processing steps, even if in some cases the result could be better with appropriate pre-processing. As a general rule we also discarded speeding-up options offered by some tools which are done at the expense of restoration quality.

In general the two crucial parameters to be tuned are the number of iterations and the coefficient linked to the degree of regularization to be achieved, which is inherent to the noisiness of the image. Concerning the number of iterations, we always compared results obtained at the 40th iteration. The regularization param-

eter needs instead to be optimized for each image as it is linked to the level of noise, and for each tool, as it is not always presented in the same form. For example *HuygensPro* asks for a Signal-to-Noise Ratio value while *DeconvolutionLab* asks for a '*lambda*' regularization value (Richardson-Lucy with TV regularization algorithm). Specifically, for each tool and for each processed image, preliminary tests were done to identify a coarse range of regularization parameter variation which gives back consistent results. Then, a finer optimization of the parameter was performed to get the best result. For the parameters optimization, the same amount of effort was spent on the different software. The results which are explained here are the product of a thorough parameters optimization process.

Test Images

The choice of three-dimensional datasets for the testing of deconvolution tools is a critical part of the comparison protocol. We considered widefield images only. Particularly, we individuated three types of images: synthetic images, acquired images of a fluorescent bead and acquired images of a biological sample.

First, we generated a synthetic image of six parallel hollow bars (fig. 1). The original volume was convolved with a theoretical PSF, and corrupted by Gaussian noise and Poisson noise with different resulting Signal-to-Noise Ratios (SNR = 30, 15). This kind of data allows us to have a ground-truth to assess the deconvolution results; it allows for a quantitative evaluation of the algorithm performance via deviation indicators. Moreover, as we corrupted the images with different degrees of Poisson noise (to simulate the shot noise for different microscope configurations) we could evaluate the behavior of the algorithms depending on image noisiness.

The second test volume was a fluorescent bead with a known diameter of 2.5 µm (fig. 2). This dataset had the advantage of offering a simple object on which it was easy to evaluate, in a quantitative way, shape and dimension recovery after deconvolution.

Finally, we performed analysis on a biological sample image, a *Caenorhabditis elegans* embryo (*C. elegans*) containing DAPI, FITC and CY3 stainings, acquired in real working conditions (fig. 3) where deconvolution effects can be evaluated on different kinds of structures: extended objects (the chromosomes in the nuclei), filaments (the microtubules) and point-wise spots (a protein detected with CY3).

One contribution of this paper is to make a set of images and a PSF generator freely available for further deconvolution tests. More details about our experimental protocol and the complete dataset (generated at the *Ecole Polytechnique Fédérale de Lausanne*.) are available at http://bigwww.epfl.ch/deconvoluiton/.

Instruments

The images have been acquired by an *Olympus CellR* widefield system.

All the deconvolutions have been run on the same machine, a 2 *Intel Xeon Dual-Core CPUs* 2.66 GHz, 10 GB of RAM, and under the same conditions. The computation time and the memory consumption peaks have been evaluated through the *Windows Vista Resource Monitor (Perfmon)*. The generation of the theoretical PSFs was based on the Rayleigh-Sommerfeld diffraction theory and was performed with a plugin for ImageJ that can be found at our dataset page (http://bigwww.epfl.ch/deconvoluiton/).

For data analysis and visualization we used *Matlab, Molecular Devices MetaMorph* and *Bitplane Imaris*.

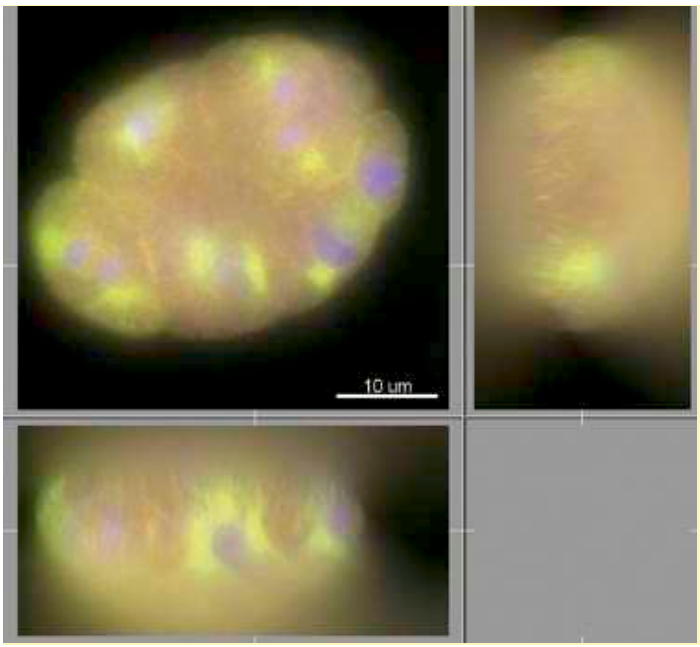

Fig. 3: ***C. elegans*** **embryo: DAPI, FITC and CY3 staining. Widefield image, Olympus Cell R, 100X 1.4NA oil objective. Voxel size 64.5 X 64.5 X 200 nm; dimensions 673 X 714 X 111 voxels, 303 MB, 16 bit dynamic range.**

First Conclusions

In this first part of the study we presented a brief overview on deconvolution technique. Then we introduced the deconvolution software we selected for our comparison, pointing out their differences and their salient points. Finally we made available different datasets suited for deconvolution testing and a PSF generator.

In the second part of the study we will extensively expose the criteria of comparison we adopted and our results.

Acknowledgements

We thank our colleagues José Artacho, Thierry Laroche, J.-C. Floyd Sarria, Jane Kraehenbuehl and Virginie Früh (BIOp group) for their contribution and suggestions. We thank Debora Keller for *C. elegans* samples. We thank *SVI and MediaCybernetics* for their kind availability and for the demos of the software.

References

[1] Chomik A. *et al.*: Journal of Optics 28, 225–233 (1997)
[2] Landmann L.: Journal of Microscopy 208, 134–147 (2002)
[3] Sibarita J.-B.: Advances In Biochemical Engineering/Biotechnology, Springer Berlin, 201–243 (2005)
[4] Wallace W. *et al.*: BioTechniques 31, 1076–1097 (2001)
[5] Biggs D.S.C.: Biophotonics International 11, 32–36 (2004)
[6] Sarder P. *et al.*: IEEE Signal Processing Magazine, 23, 32–45 (2006W)

Authors:
Alessandra Griffa, Nathalie Garin*; Bioimaging and Optics platform (BIOp), Ecole Polytechnique Fédérale de Lausanne (EPFL)
Daniel Sage; Biomedical Imaging Group (BIG), Ecole Polytechnique Fédérale de Lausanne (EPFL)

***Contact:**
Leica Microsystems
Heerbrugg, Switzerland
Nathalie Garin
nathalie.garin@leica-microsystems.com