Comparison of Deconvolution Software in 3D Microscopy

A User Point of View - Part 1 + Part 2

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.
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) 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.

**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 parameter 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/deconvolution/.

**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/deconvolution/).

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

**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.

**Comparison of Deconvolution Software**

A User Point of View - Part 2
Deconvolution is an image processing technique that restores the effective object representation [3][4], allowing to improve images analysis step such as segmentation [1] or colocalization study [2]. We performed several deconvolution tests on different kinds of datasets. The methodology has been reported in Part 1. Evaluation criteria and results are reported here.

Materials and Methods:

Evaluation Parameters

The aim of this work is to compare the performance of different deconvolution tools, quantitatively and qualitatively. Therefore we defined a series of evaluation parameters which allows this comparison. The synthetic dataset is the only one for which we have a ground-truth. As an index of deconvolution efficacy, namely a quantitative indicator of deblurring and noise reduction, we computed the normalized Root Mean Square Error (RMSE) between the deconvolved synthetic volumes and the original one, and between the convolved and noise-corrupted volumes and the original one. We normalized the dataset such as to minimize the RMSE between each couple of images, an important step due to the different dynamic ranges of the deconvolved image between different software results.

Beads images were evaluated by Full Width at Half Maximum (FWHM) on axial and radial intensity profiles passing through the center of the object (Figure 6). This was intended to quantify the resolution improvement indicated by a FWHM closer to the real bead diameter after processing, as expected. Moreover we evaluated the relative contrast between the border and the center of the bead, which is a direct measurement of contrast enhancement. We specifically computed a relative contrast on intensity profiles extracted from the images by normalizing the absolute contrast with respect to the maximum intensity value of the profiles.

Biological images were by computing the relative contrast on intensity profiles extracted from the FITC channel to quantify contrast enhancement. We evaluated the contrast between separate microtubules (Figure 8). Moreover, for a qualitative evaluation of deconvolution benefits on the biological images, six people (five experts and one naïve user) were asked to judge the results in terms of well defined parameters: sharpness and objects discrimination. C. elegans images were also evaluated for the number of protein aggregates detected with CY3 and detection done automatically or by eye were compared. The automatic spots detection technique was a combination of Gaussian filtering and local maxima detection, without threshold (Figure 7). Finally, to evaluate software performance, we
considered the elapsed computation time and peak memory consumption during deconvolution.

**Results and Discussion**

On the basis of the described protocol [7], we performed various deconvolutions on three test datasets, with the aim of comparing the performance of the different tools, quantitatively and qualitatively and also in terms of usability, time and memory usage. Even if it is certainly not possible to draw an absolute ranking of the various packages, we will still highlight their advantages and weak points by the statement of the performed tests. All the evaluation parameters values and results are reported in Table 1.

**Quality of the Restoration, Quantitative Considerations**

Image deconvolution yields a deblurring of the volume, an indirect noise reduction, a resolution improvement and a contrast enhancement. To quantify and compare these benefits, we computed some quantitative parameters specific to each dataset. For the synthetic images, for which we have a ground-truth, the normalized Root Mean Squared Error (RMSE) is an indicator of deblurring and noise reduction after deconvolution. As we expected, the reduction of the RMSE respect to the undeconvolved stack was generally significant, and higher in case of less noise-corrupted images. However comparing the RMSEs, we could notice that AutoDeblur, HuygensPro and DeconvolutionLab showed a superior robustness to noise compared to Parallel Iterative Deconvolution and Iterative Deconvolve 3D. We compared the RMSEs of the synthetic image corrupted by Poisson noise with SNR equal to 30 and equal to 15. We observed that the increase of the RMSEs with the SNR decrease was much more significant for Parallel Iterative Deconvolution and Iterative Deconvolve 3D. Therefore these two tools appear less robust to noise. Concerning the bead acquisitions, we report the Full Width at Half Maximums (FWHMs) evaluated on intensity profiles extracted from the images before and after deconvolution. We always observed a reduction of the FWHMs in the results: after deconvolution the bead dimensions were closer to the original ones. This result shows how deconvolution improves resolution. The reduction of bead diameter in the deconvolved image was particularly consistent for the HuygensPro and the DeconvolutionLab results, in the axial direction. To quantify the contrast enhancement, we computed the local relative contrast between the border and the center of the bead. In all the deconvolution results we pointed out contrast amelioration. This amelioration was particularly significant in AutoDeblur results. From the C. elegans images, FITC channel, we evaluated the contrast variation before and after deconvolution between adjacent microtubules (Figure 8). The
contrast enhancement was particularly significant in AutoDeblur results.

**Quality of the Restoration, Qualitative Considerations**

To evaluate the deconvolution results, we always started from a visual inspection of the data. Figure 7 depicts de-blurring and de-noise effects of deconvolution. We can observe that *Parallel Iterative Deconvolution* and *Iterative Deconvolve 3D* give less good results compared to the other tools in case of highly noisy images. For a qualitative evaluation of the biological volumes restoration, six persons visually judged the deconvolution results. It emerged that the pushed contrast enhancement obtained with *AutoDeblur* can generate visually less realistic images, with more probable appearance of false structures. Strip-like artefacts have been observed on *HuygensPro* results (see also Figure 5). The outcomes from *AutoDeblur* and *HuygensPro* were nevertheless considered of high level and qualitatively equivalent. The results from DeconvolutionLab were generally judged fuzzier. Finally we compared the results of automatic and visual detection of spots on the *C. elegans* image, CY3 channel. On the non-deconvolved image the result of the automatic detection was not reliable, as too many local maxima were detected. The numbers of spots detected on *HuygensPro* and *DeconvolutionLab* results were similar to the counting by a user (Figure 7).

**Time and Memory Performance**

A final important consideration to be done concerns the performance of the tools in terms of runtime and memory consumption, as the deconvolution computational effort is in general particularly heavy. *Parallel Iterative Deconvolution* and *Iterative Deconvolve 3D* open-source plugins are inferior in this sense. This is a considerable disadvantage, as the deconvolution can easily fail because of heap memory exception, even for images of common size. On our 10 GB machine we were not able to deconvolve the bead image, 32 MB, and the *C. elegans* image, 101MB per channel, with these two plugins. As all the tests were performed on the same machine and in the same conditions, the processing time and the memory consumption of the different tools can be compared, numbers of iterations being equal.

**Conclusions**

The results of our tests are comparable as we followed a well defined working guideline [7] and the same adequate effort was put into the optimization of deconvolution parameters for all the algorithms. All the tools that we considered showed good level performance in terms of results quality. With *Parallel Iterative Deconvolution* and Iterative Deconvolve 3D we were
not able to perform all the tests because of out-of-memory exceptions. AutoDeblur showed a particularly high increment of contrast and produced much sharper results. This can facilitate the segmentation but makes the appearance of false structures more probable. HuygensPro produced results that appear more realistic to the expert eye, even if background artefacts were observed in the z direction. DeconvolutionLab showed a particularly good restoration of spatial resolution, but the results were generally considered fuzzier. DeconvolutionLab and HuygensPro showed better performances in terms of time and memory consumption. Both HuygensPro and AutoDeblur offer different possibilities for image pre-processing, such as background subtraction and spherical aberration corrections, which can certainly further improve the quality of the results but that were not applied in our tests to allow the comparison of the results. Moreover they are particularly user-friendly, especially concerning the setting of the deconvolution parameters. Parallel Iterative Deconvolution and Iterative Deconvolve 3D are addressed to the more expert user as the parameter setting is less immediate. They do not implement pre-processing steps and are open-source software.

**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:**


**Authors**

Nathalie Garin; Leica Microsystems, Switzerland (corresponding author), Alessandra Griffa; Bioimaging and Optics platform (BIOp), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Daniel Sage; Biomedical Imaging Group (BIG), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Contact

*Leica Microsystems*
Aloys-Fauquez 125
1018 Heerbrugg
Switzerland