Published in final edited form as:

J Microsc. 2009 December ; 236(3): 180–193. doi:10.1111/j.1365-2818.2009.03205.x.

An open-source deconvolution software package for 3-D quantitative fluorescence microscopy imaging

Y. SUN^{*}, P. DAVIS^{*}, E. A. KOSMACEK^{*}, F. IANZINI^{*,†}, and M. A. MACKEY^{*,†}

^{*} Department of Biomedical Engineering, University of Iowa, Iowa City, Iowa, U.S.A

[†] Department of Pathology, University of Iowa, Iowa City, Iowa, U.S.A

[‡] Department of Radiation Oncology, University of Iowa, Iowa City, Iowa, U.S.A

Summary

Deconvolution techniques have been widely used for restoring the 3-D quantitative information of an unknown specimen observed using a wide-field fluorescence microscope. Deconv, an open-source deconvolution software package, was developed for 3-D quantitative fluorescence microscopy imaging and was released under the GNU Public License. Deconv provides numerical routines for simulation of a 3-D point spread function and deconvolution routines implemented three constrained iterative deconvolution algorithms: one based on a Poisson noise model and two others based on a Gaussian noise model. These algorithms are presented and evaluated using synthetic images and experimentally obtained microscope images, and the use of the library is explained. Deconv allows users to assess the utility of these deconvolution algorithms and to determine which are suited for a particular imaging application. The design of Deconv makes it easy for deconvolution capabilities to be incorporated into existing imaging applications.

Keywords

Deconvolution; fluorescence microscopy; image restoration; iterative deconvolution; live cell imaging; 3-D quantitative fluorescence microscopy imaging

Introduction

Fluorescence microscopy is used in the biomedical and biological sciences to visualize specific features of biological specimens and study cellular mechanisms. However, a wide-field fluorescence microscopy imaging system is not able to optically section a 3-D object because sectioning images are blurred in the axial direction by contributions of out-of-focus light outside the plane of focus. Deconvolution, a computationally intensive image processing technique, has become an established procedure for restoring quantitative information from sectioning images acquired by fluorescence microscopy and also for improving image resolution and signal-to-noise ratio, since it was first introduced by Agard and Sedat (1983) to determine the 3-D topography of *Drosophila melanogaster* chromosomes. The primary function of deconvolution is to detect image blurring caused by out-of-focus light and reassign this light to its proper in-focus location. The image of a point source of light, known as the point spread function (PSF), provides an effective means to characterize this out-of-focus light (McNally *et al.*, 1999).

Many deconvolution algorithms have been developed, such as non-iterative regularized inverse-filtering deconvolution algorithms (Preza *et al.*, 1992a,b), constrained iterative deconvolution algorithms (Agard *et al.*, 1989; Holmes, 1989; Joshi & Miller, 1993; Conchello & McNally, 1996; Verveer & Jovin, 1997; Conchello, 1998; Verveer & Jovin 1998; Verveer *et al.*, 1999; Markham & Conchello, 2001; Schaefer *et al.*, 2001) and blind deconvolution algorithms (Holmes, 1992; Krishnamurthi *et al.*, 1995; Avinash, 1996; Markham & Conchello, 1999). It is not appropriate to judge which deconvolution method is the best for 3-D quantitative fluorescence microscopy imaging, as there is always a trade-off between computation cost and quality of deconvolved results.

To our knowledge, there are no open-source deconvolution software packages even though a few deconvolution software packages are available in the public domain: XCOSM (www.omrfcosm.omrf.org) is one available without cost to academic and non-profit institutions. It is written in C and FORTRAN, and includes regularized inverse-filtering deconvolution methods (Preza *et al.*, 1992a,b) and constrained iterative deconvolution algorithms developed based on a Poisson noise model (Conchello *et al.*, 1994; Conchello, 1995, 1998; Conchello & McNally, 1996; Markham & Conchello, 1997). Another software package is named as 3D_Deconvolution (<http://bigwww.epfl.ch/demo/deconvolution3D/>), which is written in JAVA as a plug-in to ImageJ (<http://rsb.info.nih.gov/ij/>). This package includes inverse-filtering deconvolution methods, an iterative Van Cittert's method and a ForWaRD-like method (Neelamani *et al.*, 2004). However, none of these packages are open source, thus making their implementation into other open-source software problematic, as the terms of most open-source software licenses require that all components of the software must also be open source. Open source does not just mean that the source code is freely available. Rather, open source promotes access to the design and production of quality software in research and commercial environments. Thus, the open source software model, originally proposed by the Free Software Foundation, was intended to provide a framework to support software development in an open environment. This distinct feature allows users to create software content through incremental individual effort or through collaboration.

Open source deconvolution software was needed to provide 3-D quantitative imaging capabilities for the Large Scale Digital Cell Analysis System (LSDCAS), a live cell imaging and analysis system that uses automated fluorescence and phase contrast microscopy. LSDCAS was designed to allow quantitative study of cells grown under conditions identical to those used in routine biochemical and molecular investigations, and has evolved into an extensible live cell imaging platform (Mackey & Ianzini 2002; Davis *et al.*, 2007). Thus, three constrained iterative deconvolution algorithms were developed: one based upon a Poisson noise model and two others based upon a Gaussian noise model, and then used to implement an open-source deconvolution software package. This software package, named as Deconv, provides a library that includes C++ classes designed for simulation of a 3-D PSF based on the optics used in a fluorescence microscopy imaging system and deconvolution of sectioning images acquired from a standard wide-field fluorescence microscope. Deconv includes iterative deconvolution algorithms based upon both Poisson and Gaussian noise models, from which most published deconvolution algorithms were developed, and thus has the capability of assessing the utilities of most widely used deconvolution routines for a particular imaging application. Deconv also includes image viewer utilities for visualizing, analysing and pre-processing 2-D or 3-D image data sets using OpenGL, as well as software applications for image deconvolution that use the Deconv library. Deconv provides an application programming interface for incorporation into existing imaging applications. Deconv is compatible with the GNU Public License and is freely downloadable at the LSDCAS website – <http://www.lsdcas.engineering.uiowa.edu>.

This report provides a discussion of the details of the implementation of the three iterative deconvolution algorithms and examines their utilities in processing simulated and experimentally derived microscope images.

Materials and methods

Cell Line and culture conditions

A human head and neck squamous carcinoma cell line, FaDu was used for this study. Cells were grown in Dulbecco's modified essential medium (GIBCO, Carlsbad, CA, U.S.A.) supplemented with 10% heat-inactivated foetal bovine serum (Hyclone, Logan, UT, U.S.A.) and antibiotics (100 units mL⁻¹ penicillin and 100 µg mL⁻¹ streptomycin). To visualize cells using fluorescence microscopy, culture were infected with an adenovirus construct (Ad5CMVeGFP) encoding green fluorescent proteins.

Microscope imaging system

Images of fluorescently labelled FaDu cells were obtained through a wide-field epi-fluorescence microscope using the LSDCAS software (Davis *et al.*, 2007). Fluorescence excitation was provided by a mercury vapour arc lamp using FITC excitation and emission filters (Davis *et al.*, 2007). Specimens were contained in a 25 cm² tissue culture flask (T-25). Eight-bit sectioning images were taken using an Olympus IX-71 inverted microscope (Olympus, Center Valley, PA, U.S.A.) equipped with a LiveCell stage incubator system (Pathology Devices, Inc., Westminster, MD, U.S.A.), which provides control of temperature, humidity and CO₂, and a Hamamatsu ORCA-285 CCD camera (Hamamatsu, Bridgewater, NJ, U.S.A.). A 20×/0.4 NA (numerical aperture) long working distance dry phase objective was used. Resolutions along X-, Y-, and Z-dimensions were 0.67 µm, 0.59 µm, and 1.0 µm per voxel, respectively (Davis *et al.*, 2007).

Computer software and platform

Deconv was developed for the GNU Linux operating system. Software development occurred using the Ubuntu Linux distribution and programs were compiled using the GNU C compiler system. For numerical analysis, the FFTW3 (www.FFTW3.org) library and the GNU scientific library (www.gnu.org/software/gsl) were used. The GLUT, GL and GLU libraries (www.mesa3d.org) are only required if the imaging viewer utilities are to be built. Complete information as to the build system is given by the 'README' file in the Deconv package.

Imaging model

The image formation of a standard fluorescence microscope is modelled by

$$\begin{aligned} g(X_i) &= \int h(X_i; X_o) f(X_o) dX_o + n(X_i) \\ &= \int h(X_i - X_o) f(X_o) dX_o + n(X_i) \end{aligned} \quad (1)$$

where g is the observed 3-D image, which is packaged from a set of 2-D sectioning images acquired from a microscope. The value f represents the ideal sectioning images predicted by the optics used in a microscope, which is also termed as the object, and is a function describing the 3-D distribution of labelled fluorescent dyes within a specimen. X_o is a coordinate of the object whose domain is X_{obj} . X_i is a coordinate of the observed image whose domain is X_{img} . The PSF $h(X_i; X_o)$ is the linear mapping between the object intensity $f(X_o)$ and the image intensity $g(X_i)$ mathematically described as $h(X_i - X_o)$ if it is assumed to be spatially invariant. The value n is additive noise, which is generally categorized to be either signal dependent for

a Poisson distribution or signal independent for a Gaussian distribution. Because most applications involve sampled imaging data, a discrete form of Eq. 1 is given as

$$g = Af + n \quad (2)$$

where g , f and n are now vectors representing the sampled image, object and noise, respectively. The convolution operator A is a matrix made up of the elements of the sampled PSF elements. Its inverse (A^{-1}) does not exist because the optical transfer function, which is the Fourier transform of the PSF, is band-limited. Thus, there is no unique solution for f , implying that deconvolution applied to fluorescence microscopy is an ill-posed inverse problem. Small perturbations such as noise or round-off errors can create large deviations in finding the solution (Bertero & Boccaci, 1998). Therefore, one needs to find an approximate solution for f , through the application of regularization methods and constraints derived from knowledge of the image formation process.

The PSF plays an important role in deconvolution and is often determined empirically through analysis of measurements of a small fluorescent bead. However, this procedure is time consuming (McNally *et al.*, 1999) and might not be feasible in all applications. An alternative is to numerically simulate a 3-D PSF. Such routines in Deconv were developed based on a computational PSF model derived by Gibson and Lanni (Gibson & Lanni 1989, 1991; Gibson, 1990; Kontoyannis & Lanni, 1996). This model is based on geometrical and physical optics and is used to predict the 3-D distribution of the imaging intensities formed by a point-source object under a variety of optical conditions. The analytical form of calculating a 3-D PSF in Deconv is given by the equation:

$$PSF(x, y, \Delta z) = \left| \int_0^{NA^2} J_0 \left\{ WN \cdot \sqrt{\eta} \cdot \sqrt{x^2 + y^2} \right\} \times \exp\{ \sqrt{-1} \cdot OPD(\eta, \Delta z) \} \cdot d\eta \right|^2 \quad (3)$$

where (x, y) is the coordinated of an imaging point at an optical section and Δz denotes the distance that this section is away from the focused imaging plane. The origin ($x = 0, y = 0, \Delta z = 0$) in the image space is conjugated to the position of the point-source object in the object space. The wave number, WN is equal to $2\pi/\lambda$ where λ is the wavelength of emitted light. η , derived from the normalized radius of the rear pupil of the objective lens, is the dummy integration variable and ranges from 0 to NA^2 , where NA is the numerical aperture of the objective lens. OPD is the optical path difference function determined by the optics used for imaging and the defocused distance (Δz). The OPD caused by using a different immersion medium or cover slip for the objective lens than the one suggested by the manufacturer, were discussed by Gibson and Lanni (1991), and these factors can be incorporated into the simulation of a 3-D PSF using Deconv.

Bayesian methodology

The conditional probability density relationship between the observed image and the object is constructed from Bayesian theory:

$$\begin{aligned} p(f|g) \cdot p(g) &= p(g|f) \cdot p(f) \\ \Rightarrow p(f|g) &= p(g|f) \cdot p(f)/p(g) \end{aligned} \quad (4)$$

where $p(g)$ and $p(f)$ are the probabilities of the observed image and the object, respectively, over all possible image realizations (Starck *et al.*, 2002). The Bayes' solution is found by

maximizing the right hand side of Eq. 4. The term $p(g)$ is constant and thus is not varied in the maximization process. The *maximum a posteriori* solution maximizes the product $p(g|f) \cdot p(f)$ over f and is equivalent to the maximum likelihood solution, which maximizes the density $p(g|f)$ over f , assuming a uniform probability density for $p(f)$. We use two different statistical models for $p(g|f)$: one is the Gaussian distribution model given in Eq. 5 and the maximum likelihood solution is solved by minimizing the function $\ell_g(f)$ over f in Eq. 6. The other is the Poisson distribution model described in Eq. 7 and maximum likelihood maximization is equivalent to maximizing the function $\ell_p(f)$ over f in Eq. 8.

$$p_g(g|f) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\int \left(g(X_i) - \int h(X_i - X_o) f(X_o) dX_o \right)^2 dX_i / (2\sigma^2) \right\} \quad (5)$$

$$\begin{aligned} \ell_g(f) &= \frac{1}{2} \cdot \int \left\{ g(X_i) - \int h(X_i - X_o) f(X_o) dX_o \right\}^2 dX_i \\ &= \frac{1}{2} \cdot \|g - Af\|^2 \end{aligned} \quad (6)$$

$$p_p(g|f) = \int \left\{ \left(\int h(X_i - X_o) f(X_o) dX_o \right)^{g(X_i)} \times \exp \left(-\int h(X_i - X_o) f(X_o) dX_o \right) / g(X_i) \right\} dX_i \quad (7)$$

$$\ell_p(f) = \int \left\{ g(X_i) \cdot \log \left(\int h(X_i - X_o) f(X_o) dX_o \right) - \int h(X_i - X_o) f(X_o) dX_o \right\} dX_i \quad (8)$$

The expectation maximization method is used to maximize the likelihood derived from the Poisson noise model (Conchello, 1998), and the corresponding deconvolution algorithm is referred to as the maximum likelihood–expectation maximization iterative deconvolution (ML-EMID). Two numerical methods are used to optimize the likelihood derived from the Gaussian noise model: one is the Landweber iteration method (Avinash, 1996) and the corresponding algorithm is referred to as the maximum likelihood–Landweber iterative deconvolution (ML-LWID). The other numerical method uses the conjugate gradient technique (Schaefer *et al.*, 2001) and the corresponding algorithm is referred to as the maximum likelihood–conjugate gradient iterative deconvolution (ML-CGID).

Maximum likelihood-Landweber iterative deconvolution

The ML-LWID algorithm is designed to minimize $\ell_g(f)$ using the Landweber iteration method and is an iteratively regularized inverse-filtering deconvolution method. The implementation of this algorithm is shown in Fig. 1 and is explained as follows

$$\begin{aligned} &g \xrightarrow{FFT} G; \quad h \xrightarrow{FFT} H; \quad f^{(0)} = g, \quad f^{(0)} \xrightarrow{FFT} F^{(0)}; \quad k=1 \\ &\text{pre-conditioning: } G(\omega) = \frac{G(\omega)}{\sqrt{|H(\omega)|^2 + \lambda_R}} \\ &\quad H(\omega) = \frac{H(\omega)}{\sqrt{|H(\omega)|^2 + \lambda_R}} \\ &\text{LW update: } F^{(k)}(\omega) = F^{(k-1)}(\omega) + H(\omega) \times \{G(\omega) - H(\omega)F^{(k-1)}(\omega)\}; \\ &\quad F^{(k)} \xrightarrow{\text{inverse FFT}} f^{(k)}, \quad f^{(k)} = P(f^{(k)}); \\ &\quad k = k + 1. \end{aligned} \quad (9)$$

where $f^{(0)}$ is the initialized estimated object and $f^{(k)}$ is the estimated object after k iterations; P is a projection operator which applies constraints, such as the non-negative constraint (described below), to an estimated object; λ_R is the regularization parameter used for pre-conditioning (a technique to improve the condition number of a matrix) the observed image and the PSF in the frequency domain. A chosen regularization parameter should satisfy the following two conditions: 1, $|H(\omega)|^2/(|H(\omega)|^2 + \lambda_R) \leq 1$ for any ω ; 2, $|H(\omega)|^2/(|H(\omega)|^2 + \lambda_R)$ is close to 1 for all ω such that $|H(\omega)|$ is greater than a suitable threshold value. The optimal value of λ_R can be found using a modified golden search method. In this method, λ_R is calculated at each searching step and used for pre-conditioning the observed image and the PSF as shown in Eq. 9; $\|G(\omega) - H(\omega) \cdot F(\omega)\|$ is then evaluated after running the *LW update* given in Eq. 9 for a number of iterations ($k = 1 \sim N_{PN}$); the optimal value of λ_R is finally obtained when the value of $\|G(\omega) - H(\omega) \cdot F(\omega)\|$ is minimized. Therefore, $G(\omega)$ and $H(\omega)$ taken into the main deconvolution loop are obtained through pre-conditioning the observed image and the PSF using the optimized regularization parameter (see Fig. 1).

Maximum likelihood-conjugate gradient iterative deconvolution

Landweber iteration is a stable minimization method but converges slowly. The conjugate gradient minimization method generally gives a better convergence than the steepest descent and Landweber iteration methods. The ML-CGID algorithm is designed to minimize $l_g(f)$ using the conjugate gradient method. The implementation of the ML-CGID algorithm is shown in Fig. 2 and is explained as follows

$$\begin{aligned}
 f^{(0)} &= g; \quad r^{(0)} = A^T(Af^{(0)} - g); \quad d^{(0)} = r^{(0)}; \\
 S_{i=j}(i, j) &= 1; \quad k=1 \\
 \text{pre-conditioning: } G(\omega) &= \frac{G(\omega)}{\sqrt{|H(\omega)|^2 + \lambda_R}} \\
 H(\omega) &= \frac{H(\omega)}{\sqrt{|H(\omega)|^2 + \lambda_R}} \\
 \text{CG update: } \alpha &= \frac{-[d^{(k-1)}]^T \cdot S \cdot r^{(k-1)}}{[Ad^{(k-1)}]^T \cdot S \cdot Ad^{(k-1)}}, \\
 S_{i=j}(i, j) &= \begin{cases} 1, & \text{if } f_{i(k)} = P(f_{i(k)}) \\ 0, & \text{otherwise} \end{cases} \\
 f^{(k)} &= f^{(k-1)} + \alpha \cdot d^{(k-1)}, \quad f^{(k)} = P(f^{(k)}); \\
 r^{(k)} &= A^T(Af^{(k)} - g), \quad \beta = \|r^{(k)}\|^2 / \|r^{(k-1)}\|^2, \\
 d^{(k)} &= r^{(k)} + \beta \cdot d^{(k-1)}; \\
 k &= k+1.
 \end{aligned} \tag{10}$$

where d is the conjugate gradient searching direction calculated from the steepest descent searching direction r . The value β is the step size for updating d . α is the step size used to update an estimated object and its optimal value can be calculated by $\{d \cdot r\} / \{[Ad]^T \cdot [Ad]\}$, however this procedure will not always produce correct values for α in the presence of constraints (Verveer & Jovin, 1997). Thus, a diagonal matrix S , whose dimensions are the same as A , is used to record which elements of an estimated object are changed due to the constraints enforced through the projection operator P . The same pre-conditioning technique described for ML-LWID is also used in ML-CGID and the same golden search method employing the *LW update* (see Eq. 9 in the previous section) is also applied to find the optimal value of λ_R used in ML-CGID.

The maximum likelihood estimation procedure has a tendency to amplify isolated bright spots in a deconvolution process. In order to suppress this, a regularization operator R is usually applied to an estimated object. The intensity regularization, which only penalizes overly bright spots without smoothing, is optional in ML-CGID. If the intensity regularization is applied to

ML-CGID, the penalized likelihood function for the Gaussian noise model ' $l_g(f) + (c/2) \cdot \|f\|^2$ ' will be minimized instead of $l_g(f)$, where c is the intensity penalty parameter and is found using a modified golden search method to minimize a generalized cross-validation function given in Eq. 11 (Schaefer *et al.*, 2001). Then, the steepest descent searching direction r and the step size α in the presence of the intensity regularization will be calculated using Eq. 12, where I is an identity matrix whose dimensions are the same as A .

$$GCV(c) = \int \frac{c^2 |G(\omega)|^2}{(|H(\omega)|^2 + c)^2} / \left| \int \frac{c}{|H(\omega)|^2 + c} \right|^2 \quad (11)$$

$$\begin{aligned} r^{(k)} &= (A^T A + cI) f^{(k)} - A^T g \\ &\quad - [d^{(k-1)}]^T \cdot S \cdot r^{(k-1)} \\ \alpha &= \frac{[Ad^{(k-1)}]^T \cdot S \cdot Ad^{(k-1)} + c \cdot [d^{(k-1)}]^T \cdot d^{(k-1)}}{[Ad^{(k-1)}]^T \cdot S \cdot Ad^{(k-1)} + c \cdot [d^{(k-1)}]^T \cdot d^{(k-1)}} \end{aligned} \quad (12)$$

Maximum likelihood-expectation maximization iterative deconvolution

The ML-EMID algorithm is designed to maximize $l_p(f)$ using the expectation maximization method. The implementation of the ML-EMID algorithm is shown in Fig. 3 and is explained as follows

$$\begin{aligned} f^{(0)} &= g; k=1 \\ EM \text{ update:} \\ f^{(k)}(X_o) &= f^{(k-1)}(X_o) \int \frac{h(X_o - X_i) g(X_i)}{\int h(X_i - X_o) f^{(k-1)}(X_o) dX_o} dX_i \\ f^{(k)} &= P(f^{(k)}) \\ k &= k+1. \end{aligned} \quad (13)$$

The *EM update* given in Eq. 13 can be transformed into a direction search approach, as shown in Eq. 14,

$$\begin{aligned} d(X_o) &= f^{(k-1)}(X_o) \times \left(\int h(X_o - X_i) \frac{g(X_i)}{\int h(X_i - X_o) f^{(k-1)}(X_o) dX_o} dX_i - 1 \right) \\ \beta(X_i) &= \int h(X_i - X_o) f^{(k-1)}(X_o) dX_o; \\ \gamma(X_i) &= \int h(X_i - X_o) d(X_o) dX_o; \\ \alpha &= 1; \alpha = \alpha + \frac{\int \left\{ g(X_i) \frac{\gamma(X_i)}{\beta(X_i) + \alpha \cdot \gamma(X_i)} - \gamma(X_i) \right\} dX_i}{\int \left\{ g(X_i) \frac{[\gamma(X_i)]^2}{[\beta(X_i) + \alpha \cdot \gamma(X_i)]^2} \right\} dX_i} \\ f^{(k)}(X_o) &= f^{(k-1)}(X_o) + \alpha \cdot d(X_o) \end{aligned} \quad (14)$$

where α is the acceleration parameter and d is the searching direction. Acceleration is achieved by choosing $\alpha > 1.0$, while keeping other variables constant to maximize $l_p(f^{(k-1)}, \alpha)$ subject to the constraint that $f^{(k)}$ is non-negative. The optimal value of α is solved using the iterative Newton method. There is a computation time trade-off using Newton acceleration, because the calculation of α takes more time per iteration even though the use of Newton acceleration completes the deconvolution process in less iterations. Thus, the use of Newton acceleration is optional in ML-EMID.

Problems associated with amplification of isolated bright spots are more serious in ML-EMID than in ML-CGID. Thus, the intensity regularization is always necessary in ML-EMID. The penalized likelihood function for the Poisson noise model is given as ' $l_p(f) - c \cdot \|f\|^2$ ' (Conchello and McNally, 1996), where the intensity penalty parameter c determines the tradeoff between fitting and smoothing the solution. One expects deconvolved images will be smoother with a higher value of c . An estimation for c is given by Conchello as the maximum value of the PSF divided by the maximum value of the observed image (Conchello & McNally, 1996). The intensity penalization applied to the estimated object is

$$f^{(k)}(X_o) = \frac{-1 + \sqrt{1 + 2 \cdot c \cdot f^{(k)}(X_o)}}{c} \quad (15)$$

Constraints

Three constraints are defined based on the projection onto convex sets theory (Jansson, 1997): the non-negative constraint, the spatial bound constraint and the frequency bound constraint. The non-negative constraint is based on the fact that the intensity of an element in an estimated object cannot be negative and is required for all deconvolution algorithms in Deconv. The estimated object projected to the non-negative convex set by this constraint is given in Eq. 16

$$P\{f^{(k)}(X_o)\} = \begin{cases} 0, & \text{if } f^{(k)}(X_o) < 0 \\ f^{(k)}(X_o), & \text{otherwise} \end{cases} \quad (16)$$

The spatial bound constraint is due to the fact that the estimated object vanishes outside the prescribed region C_S in the spatial domain. Its use is optional for all deconvolution algorithms in Deconv. The estimated object projected to the convex set C_S by this constraint is given in Eq. 17

$$P\{f^{(k)}(X_o)\} = \begin{cases} 0, & \text{if } X_o \notin C_S \\ f^{(k)}(X_o), & \text{if } X_o \in C_S \end{cases} \quad (17)$$

The frequency bound constraint is based on the fact that the optical transfer function vanishes outside the prescribed region C_F in the frequency domain. Its use is also optional for all deconvolution algorithms in Deconv. The optical transfer function is projected to the convex set C_F by this constraint is given in Eq. 18

$$P\{H(\omega)\} = \begin{cases} 0, & \text{if } \omega \notin C_F \\ H(\omega), & \text{if } \omega \in C_F \end{cases} \quad (18)$$

The Deconv library

As the core of Deconv, the Deconv library provides routines for simulating a 3-D PSF and deconvolution of optical sectioning images acquired from a standard wide-field fluorescence microscope. These numerical routines were implemented using the C++ programming language and their application can thus be made through calling the member functions of the C++ classes in the Deconv library. The manuscript 'Deconv Library User Manual' that

documents the Deconv library is included in the Deconv software package and example C++ programs utilizing the Deconv library along with the test data are also provided (see <http://lsdcas.engineering.uiowa.edu/repos/lsdcas/trunk/deconv/UserManual.pdf>).

The calculation of a 3-D PSF is based on the numerical integration at each of its voxels (see Eq. 3). The Deconv library provides two ways of simulating a 3-D PSF: (1) Perform an integration directly at each point of the 3-D grid and integrations are only required on one eighth of the total points due to that the PSF is cylindrically symmetric. (2) Combine x- and y-dimensions into a new dimension called radial (r) dimension such that r^2 is equal to the addition of x^2 and y^2 . The usage of r - and z -dimensions makes a new coordinate system – the cylindrical coordinate system, based on which a 2-D lookup table is built to store the PSF values at different radial and axial distances to the origin. The simulation of a 3-D PSF can thus be carried out by first generating a 2-D PSF lookup table and then calculating the 3-D PSF employing table lookups and spline interpolations.

The first way of calculating a 3-D PSF is simple and straightforward. However, the second way may be more preferable in some circumstances. For example, when the actual sampling rates of imaging along X- and Y-dimensions (pixel size) are larger than the Nyquist sampling rate (λ/NA 4.0) calculated from the NA of the objective lens and the emission light wavelength (λ), it is better to generate a 2-D PSF lookup table using a radial sampling rate smaller than the Nyquist sampling rate to avoid the under-sampling problem, and then to calculate the 3-D PSF based on the actual sampling rates of imaging.

The Deconv library provides routines used to store the information and data of a 2-D PSF lookup table into files as well as to read them from the saved files to create a 3-D PSF. Because the generation of a 3-D PSF using a 2-D PSF lookup table does not require numerical integrations which consume most of the time in the first way of creating a 3-D PSF, the capability of reproducing 3-D PSFs with different dimensions and different sampling rates from a saved 2-D PSF lookup table may be useful in some applications, such as deconvolving a large amount of image data sets, or deconvolution of a number of sub-image data sets with different dimensions segmented from an image data set with large dimensions.

Three deconvolution routines were implemented based on three constrained iterative deconvolution algorithms described above and each routine is associated with one C++ class in the Deconv library. All three classes were structured similarly so that the same procedures can be followed for using each method: (1) create a plan for a deconvolution process; (2) run the deconvolution process which will be regulated by number of parameters and control flags specified when the deconvolution plan was generated.

Some of the parameters and control flags are commonly used in each deconvolution routine, such as the criterion used to terminate a deconvolution process (explained below), the number of maximally allowed ‘deconvolving iterations (k in Eqs 9, 10 and 13)’ in the main deconvolution loop, the ‘*IsTrackLike*’ flag indicating whether to track the likelihood values (l_g in Eq. 6 and l_p in Eq. 8) iteratively or not, the ‘*CheckStatus*’ flag that is used to control whether to display the progress of a deconvolution process or not, and etc.

Other parameters and control flags are specifically applied in one or two of the three deconvolution routines. Because the input data need to be pre-conditioned in an ML-LWID or ML-CGID process, the regularization parameter (λ_R in Eqs. 9 and 10) used for the pre-conditioning process is required by using ML-LWID or ML-CGID. This parameter can be either automatically determined from the input data using a modified golden search method or manually specified by users to skip the searching process. The searching method is recommended for users who have little prior knowledge about the regularization parameter, although the manual input of this parameter will save the processing time spent in the searching

process. To use the searching method, one needs to specify the number of iterations (N_{PN}) for running *LW update* in each searching step. The usage of a larger value of N_{PN} may help find a more accurate regularization parameter and thus yield better pre-conditioning results. However, it will also result in a longer processing time. Running ML-LWID or ML-CGID on the image data sets acquired from our imaging system, we found the searched optimal values of the regularization parameter were barely changed when the value of N_{PN} is increased from 5. Thus, the default value of N_{PN} is 5 in the Deconv library. It is suggested that users should investigate the optimal value of N_{PN} for their imaging systems.

The option of applying the intensity regularization in an ML-CGID process is controlled by the '*IsApplyIR*' flag which is specified when an ML-CGID plan is created. The intensity penalty parameter used in an ML-CGID process is determined from the input data employing a modified golden search method to minimize Eq. 11. For an ML-EMID process, it is always recommended that the intensity regularization should be enforced every a number of iterations (N_{IR}) and this issue will be discussed below. The intensity penalty parameter used in an ML-EMID process can be either manually given or determined from the input data using Eq. 15, and the influence on the deconvolved results given by using a larger or smaller penalty was discussed above.

The convergence of ML-EMID can be improved using Newton acceleration; however due to the calculation of the acceleration parameter in each iteration, a computation time trade-off needs to be considered for using Newton acceleration. Thus, the usage of Newton acceleration is optional and controlled by the '*IsAccelerate*' flag which is specified when an ML-EMID plan is generated. It should be noticed that no FFT is involved in calculating the acceleration parameter (see Eq. 14). Thus, one may consider applying Newton acceleration for deconvolving an image data set with large dimensions, in which case the calculation of a FFT is very time consuming.

Once a deconvolution plan is generated, only one function call is needed to make the deconvolution going. The input arguments required by running each types of deconvolution are same except that each of them has a special work space due to different memory requirements. The required input arguments include a 3-D PSF, a 3-D image data set and an initialized estimated object, all with equivalent dimensions. The initialized object is usually the same as the observed image. The 3-D PSF can be either simulated using routines included in the Deconv library or obtained from an experimental measurement. Although a spatial or a frequency bound is not required, it can be applied by passing a 3-D binary mask defining the spatial support (C_S) or the frequency support (C_F), to a deconvolution routine. If the spatial support (C_S) is given, the spatial bound constraint will be iteratively enforced on the estimated object as shown in Eq. 17. If the frequency support (C_F) is used, the frequency bound constraint will only be applied on the optical transfer function at the beginning of a deconvolution process according to Eq. 18.

Results

Deconvolution of synthetic image data

In order to evaluate deconvolution algorithms in Deconv, a synthetic 3-D image was created from the convolution of a simulated object and a calculated PSF, and then deconvolved employing ML-LWID, ML-CGID and ML-EMID using the same PSF. A 3-D object composed of a cylinder, two crossed bars and eight spheres was created in a $256(X) \times 256(Y) \times 64(Z)$ volume. This object was then convolved with a small 3-D ($5 \times 5 \times 3$) Gaussian kernel to obtain the simulated object. A PSF that has the same dimensions of the simulated object was generated based on Eq. 3 and a 0.6 NA dry objective lens providing a resolution of $0.2 \mu\text{m}/\text{pixel}$ along X - and Y -dimensions. The sectioning interval of $1.0 \mu\text{m}$ along the optical axis was used. The

wavelength of emitted light was defined as 510 nm. It was also assumed that the *OPD* only depended on the defocused distance Δz . The synthetic 3-D image was finally obtained after adding noise, truncating and round-off errors to the convolved result of the simulated object and the simulated PSF. Both Gaussian noise (signal-independent distribution) and Poisson noise (signal-dependent distribution) were added. The signal to noise ratios for the Gaussian and Poisson noises were 22 db and 28 db, respectively. For better visualization and comparison, the simulated object (a), the synthetic image (b) and the deconvolved images using ML-LWID (c), ML-CGID (d) and ML-EMID (e) were all cropped to the size of $96 \times 96 \times 32$ and shown in Fig. 4. It was found that all deconvolution algorithms in Deconv effectively restored the information of the simulated object from the synthetic image.

All three deconvolution algorithms in Deconv were evaluated through analysis of the mean squared error between the normalized deconvolved image and the normalized simulated object defined in Eq. 19

$$e^{(k)} = \left\| f^{(k)} / \max f^{(k)} - f^T \right\|^2 / \text{size}\{f\} \quad (19)$$

where $f^{(k)}$ is a deconvolved image data set after k iterations and $\max\{f^{(k)}\}$ is its maximum value. f^T is the normalized simulated object, $\text{size}\{f\}$ is the product of the X -, Y - and Z -dimensions ($256 \times 256 \times 64$ for our simulated object). $e^{(k)}$ is the mean squared error between $f^{(k)}$ and f^T as a function of iteration number. Figure 5 shows the mean squared errors calculated in deconvolution of the synthetic image using ML-LWID, ML-CGID and ML-EMID for 1000 iterations. All three curves drop fast as the iteration number first increases, showing the restoration power of the deconvolution algorithms. At high iterations, all the curves reach a plateau indicating that an objective criterion for determining when to stop a deconvolution process is required.

The norm of the difference between the current and last estimated objects weighted by the norm of the current estimated object as a function of iteration number $\delta^{(k)}$ is defined in Eq. 20. The values of $\delta^{(k)}$ calculated in deconvolution of the synthetic image using ML-LWID, ML-CGID and ML-EMID are shown in Fig. 6, which demonstrates the convergence property of the deconvolution algorithms.

$$f^{(k)} = \left\| f^{(k)} - f^{(k-1)} \right\| / \left\| f^{(k)} \right\| \quad (20)$$

The average value of $\delta^{(k)}$ over the past ten iterations is used as a stopping criterion of all deconvolution algorithms in Deconv such that a deconvolution process will be terminated when this average value is less than a preset level. The criteria and performance characteristics of deconvolution of the synthetic image are given in Table 1. The criterion value ($1.0\text{E}-6$) used in ML-EMID was set larger than the criterion value ($1.0\text{E}-7$) used in ML-CGID and ML-LWID because Fig. 5 shows the $e^{(k)}$ curve for ML-EMID reaches the plateau after 200 iterations and the criterion value ($1.0\text{E}-6$) used in ML-EMID was reached after 218 iterations (see Table 1), meaning that a deconvolved result obtained using ML-EMID cannot be improved applying a smaller criterion value than $1.0\text{E}-6$. This fact indicates that one needs to choose different criterion values for different deconvolution algorithms. Figure 6 shows the $\delta^{(k)}$ curve for ML-CGID in Fig. 5 is below that for ML-LWID, demonstrating that the conjugate gradient method converges faster than the Landweber iteration method. The $\delta^{(k)}$ curve for ML-EMID dramatically raises and then drops every 40 iterations, which is how often the intensity regularization was applied in ML-EMID, indicating that the intensity regularization caused a

problem in solution convergence although it is quite useful for penalizing isolated bright spots in an estimated object. Thus, it is suggested that the intensity regularization is applied in ML-EMID every a number of iterations ($N_{IR} = 20\sim 80$) rather than iteratively. The synthetic image was deconvolved using ML-EMID with different values of N_{IR} and the difference was barely seen from deconvolved results when N_{IR} was altered by less than 10. However, one has to determine the trade-off such that more penalization but slower convergence is expected with a smaller value of N_{IR} .

Deconvolution of live cell image data

Uncorrected optical sectioning images collected from a fluorescently labelled FaDu cell were deconvolved using deconvolution algorithms in Deconv. The PSF used for deconvolution was simulated based on the optical characteristics of our imaging system (see 'Materials and methods'). It was observed from Fig. 7 that the blurring components of emitted light in the raw microscope images (a) were significantly removed after deconvolution using ML-LWID (b), ML-CGID (c) and ML-EMID (d). A thorough comparison of the deconvolved images shown in Fig. 7 indicated that ML-EMID gave a better restoration along the optical axis than ML-LWID and ML-CGID in this experiment. The criteria and performance characteristics of deconvolution of the fluorescently labelled cell images are shown in Table 2.

Discussion

The quality of deconvolved results can be assessed directly by comparing raw images with deconvolved images using our eyes based upon the morphology of graphics. However, it is almost impossible to make a quantitative comparison if observed specimens are unknown, which is true in most biological experiments. Therefore, the simulated object (Fig. 4a) and the synthetic image (Fig. 4b) were created for evaluating deconvolution algorithms in Deconv by measuring the mean squared errors between the deconvolved images and the simulated object. In terms of this error, the deconvolved result obtained using ML-LWID was the closest to the simulated object and also the deconvolved result of ML-LWID or ML-CGID was better than the one obtained by ML-EMID (Fig. 5). Although, thoroughly visual comparison of the small spheres in the restored images led us realize ML-EMID yielded a better axial restoration (see XZ-slices indexed at 37, 39 and 57 in Fig. 4). This conclusion was also supported by the results obtained in deconvolution of the FaDu cell images (see XZ-slices in Fig. 7). Visual comparison on the restored synthetic images or FaDu cell images also showed that the deconvolved results obtained using ML-LWID and ML-CGID were very close (Fig. 4c versus d and Fig. 7b versus c).

Besides the quality of deconvolved results, the speed of a deconvolution process is always a concern for the implementation of deconvolution software based on an iterative deconvolution algorithm. It was found from Fig. 6, Tables 1 and 2 that the ML-CGID algorithm converges faster than the ML-LWID and ML-EMID algorithms. However, ML-LWID takes less time per iteration (see Tables 1 and 2) because it requires less calculation. The deconvolution speed is not just dependent on the converging property of a deconvolution algorithm but also its complexity. This speed also varies for the implementation of an algorithm on different computer hardware, for example, the deconvolution speed can be improved using the parallel version of the FFTW3 library if Deconv were implemented on a multi-processors computer. In this case, the ML-CGID and ML-EMID algorithms will benefit more than the ML-LWID algorithm because more FFTs per iteration are calculated in them. The fastest deconvolution speed using Deconv is achieved by configuring ML-LWID such that only one iteration is performed after pre-conditioning. This use of ML-LWID is equivalent to a regularized non-iterative inverse-filtering deconvolution method.

Through the analyses based upon deconvolution of the synthetic images and fluorescently labelled cell images, we found that each of deconvolution algorithms in Deconv has its strength and weakness, and deconvolution is application dependent. It was also demonstrated that the ML-LWID and ML-CGID algorithms developed based on a Gaussian noise model have a similar behaviour in deconvolution but are fundamentally different from the ML-EMID algorithm which is based on a Poisson noise model. Using Deconv, one can assess the utilities of these deconvolution algorithms for a particular application. Deconv also provides options for users to explore trade-offs between deconvolution speed and the quality of the results obtained, within the context of a particular imaging application.

Acknowledgments

The project was supported by grant CA94801 from the U.S. National Institutes of Health.

References

- Agard DA, Sedat JW. Three-dimensional architecture of a polytene nucleus. *Nature* 1983;302:676–681. [PubMed: 6403872]
- Agard DA, Hiraoka Y, Shaw P, Sedat JW. Fluorescence microscopy in three dimensions. *Meth Cell Biol* 1989;30:353–377.
- Avinash GB. Data-driven, simultaneous blur and image restoration in 3-D fluorescence microscopy. *J Microsc* 1996;183:145–167.
- Bertero, M.; Boccaci, P. *Introduction to Inverse Problems in Imaging*. IOP Publishing; Bristol: 1998.
- Conchello J. Fluorescence photobleaching correction for expectation maximization algorithm. *SPIESer* 1995;2412:138–146.
- Conchello J. Superresolution and convergence properties of the expectation-maximization algorithm for maximum-likelihood deconvolution of incoherent images. *J Opt Soc Am A* 1998;15:2609–2619.
- Conchello J, McNally JG. Fast regularization technique for expectation maximization algorithm for optical sectioning microscopy. *SPIE Ser* 1996;2655:199–208.
- Conchello J, Kim JJ, Hansen EW. Enhanced three-dimensional reconstruction from confocal scanning microscope images. II Depth discrimination versus signal-to-noise ratio in partially confocal images. *Appl Opt* 1994;33:3740–3750.
- Davis PJ, Kosmacek EA, Sun Y, Ianzini F, Mackey MA. The Large Scale Digital Cell Analysis System: an open system for nonperturbing live cell imaging. *J Microsc* 2007;228:296–308. [PubMed: 18045324]
- Gibson, SF. PhD Dissertation. Carnegie Mellon University; Pittsburgh, PA: 1990. Modeling the 3-D imaging properties of the fluorescence light microscope.
- Gibson SF, Lanni F. Diffraction by a circular aperture as a model for three-dimensional optical microscopy. *J Opt Soc Am A* 1989;6:1357–1367. [PubMed: 2795290]
- Gibson SF, Lanni F. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *J Opt Soc Am A* 1991;8:1601–1613.
- Holmes TJ. Expectation-maximization restoration of band-limited, truncated point-process intensities with application in microscopy. *J Opt Soc Am A* 1989;6:1006–1014.
- Holmes TJ. Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach. *J Opt Soc Am A* 1992;9:1052–1061. [PubMed: 1634965]
- Jansson, PA. *Deconvolution of Images and Spectra*. 2. Academic Press; New York: 1997.
- Joshi S, Miller MI. Maximum a posteriori estimation with Good's roughness for three-dimensional optical-sectioning microscopy. *J Opt Soc Am A* 1993;10:1078–1085. [PubMed: 8496727]
- Kontoyannis NS, Lanni F. Measured and computed point spread function for indirect water immersion objective used in three-dimensional fluorescence microscopy. *SPIE ser* 1996;2655:34–42.
- Krishnamurthi V, Liu Y, Bhattacharyya S, Turner JN, Holmes TJ. Blind deconvolution of fluorescence micrographs by maximum-likelihood estimation. *Appl Opt* 1995;34:6633–6647.

- Mackey MA, Ianzini F. Development of the Large-Scale Digital Cell Analysis System. *Radiat Prot Dosim* 2002;99:81–94.
- Markham J, Conchello J. Tradeoffs in regularized maximum-likelihood image restoration. *SPIE ser* 1997;2984:136–145.
- Markham J, Conchello J. Parametric blind deconvolution of microscopic images: a robust method for the simultaneous estimation of image and blur. *J Opt Soc Am A* 1999;16:2377–2391.
- Markham J, Conchello J. Fast maximum-likelihood image-restoration algorithms for three-dimensional fluorescence microscopy. *J Opt Soc Am A* 2001;18:1062–1071.
- McNally JG, Karpova T, Cooper J, Conchello J. Three-dimensional imaging by deconvolution microscopy. *Methods* 1999;19:373–385. [PubMed: 10579932]
- Neelamani R, Choi H, Baraniuk R. ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Trans Signal Process* 2004;52:418–433.
- Preza C, Miller MI, Thomas LJ, McNally JG. Regularized linear method for reconstruction of three-dimensional microscopic objects from optical sections. *J Opt Soc Am A* 1992a;9:219–228. [PubMed: 1542061]
- Preza C, Miller MI, Conchello J. Image reconstruction for 3-D light microscopy with a regularized linear method incorporating a smoothness prior. *SPIE Ser* 1992b;1905:129–139.
- Schaefer LH, Schuster D, Herz H. Generalized approach for accelerated maximum likelihood based image restoration applied to three-dimensional fluorescence microscopy. *J Microsc* 2001;204:99–107. [PubMed: 11737543]
- Starck JL, Pantin E, Murtagh F. Deconvolution in astronomy: a review. *Publ Astron Soc Pac* 2002;114:1051–1069.
- Verveer PJ, Jovin TM. Acceleration of the ICTM image restoration algorithm. *J Microsc* 1997;188:191–195.
- Verveer PJ, Jovin TM. Image restoration based on Good's roughness penalty with application to fluorescence microscopy. *J Opt Soc Am A* 1998:1077–1083.
- Verveer PJ, Gemkow MJ, Jovin TM. A comparison of image restoration approaches applied to three-dimensional confocal and wide-field fluorescence microscopy. *J Microsc* 1999;193:50–61. [PubMed: 12558687]

Fig. 1.

A flow chart of ML-LWID: the estimated object is usually initialized to be the observed image; the regularization parameter λ_R is calculated for pre-conditioning both the PSF and the observed image in the frequency domain; in the main deconvolution loop, the estimated object is updated iteratively in the frequency domain and then transformed back to the spatial domain, where the non-negative constraint is applied.

Fig. 2.

A flow chart of ML-CGID: the estimated object is usually initialized to be the observed image; the regularization parameter λ_R is calculated for pre-conditioning both the PSF and the observed image in the frequency domain; the intensity penalty parameter is calculated if the intensity regularization is applied; both the steepest descent (r) and conjugate gradient (d) searching directions are initialized; in the main deconvolution loop, the step sizes α , β , the steepest descent, conjugate gradient searching directions and the estimated object are all updated iteratively.

Fig. 3.

A flow chart of ML-EMID: the estimated object is usually initialized to be the observed image; the intensity penalty parameter is calculated if the intensity regularization is applied; in the main deconvolution loop, the estimated object is updated iteratively in the spatial domain, where the non-negative constraint and the intensity regularization are applied.

Fig. 4.

The simulated object (a), the synthetic image (b) and deconvolved images obtained using ML-LWID with $N_{PN} = 5$ for pre-conditioning (c), ML-CGID with $N_{PN} = 5$ for pre-conditioning and applying the intensity regularization iteratively (d) and ML-EMID with applying the intensity regularization every 40 iterations, (e) for better visualization and comparison, all of the volumes were cropped from $256 \times 256 \times 64$ to $96 \times 96 \times 32$; an index along each dimension of a volume starts from 0; resolutions along X-, Y- and Z-axis are 0.2, 0.2 and 1.0 μm per voxel, respectively.

Fig. 5.

The mean squared error between the normalized deconvolved object and the normalized simulated object as a function of the iteration number $e^{(k)}$ calculated in deconvolution of the synthetic image using ML-LWID, ML-CGID and ML-EMID versus iteration number k .

Fig. 6.

The norm of the difference between the current and last estimated objects weighted by the norm of the current estimated object as a function of iteration number $\delta^{(k)}$ calculated in deconvolution of the synthetic image using ML-LWID, ML-CGID and ML-EMID versus iteration number k .

Fig. 7.

The raw fluorescently labelled FaDu cell sectioning images (a), and deconvolved images obtained using ML-LWID with $N_{PN} = 5$ for pre-conditioning (b), ML-CGID with $N_{PN} = 5$ for pre-conditioning and applying the intensity regularization iteratively (c), and ML-EMID with applying the intensity regularization every 50 iterations, (d) an index along each dimension of a volume starts from 0; resolutions along X -, Y - and Z -axis are 0.67, 0.59 and 1.0 μm per voxel, respectively.

Table 1Criteria and performance characteristics of deconvolution of the Synthetic Image data set ($256 \times 256 \times 64$).^{*}

Deconvolution Method	ML-LWID (c in Fig. 4)	ML-CGID (d in Fig. 4)	ML-EMID (e in Fig. 4)
Stop criterion	1.0E-7	1.0E-7	1.0E-6
Total iterations	441	278	218
Time/iteration	0.85 s	2.54 s	2.20 s
Total time	7 min	13 min	8 min
Memory cost	105 Mb	124 Mb	97 Mb

^{*} Deconvolution programs were compiled using GCC and run in Linux on a 2.8 GHz Intel Pentium4 workstation.

Table 2

Criteria and performance characteristics of deconvolution of the fluorescently labelled FaDu cell image data set ($128 \times 128 \times 128$).^{*}

Deconvolution Method	ML-LWID (b in Fig. 7)	ML-CGID (c in Fig. 7)	ML-EMID (d in Fig. 7)
Stop criterion	1.0E-7	1.0E-7	1.0E-7
Total iterations	732	348	986
Time/iteration	0.49 s	1.38 s	0.79 s
Total time	6 min	8 min	13 min
Memory cost	52 Mb	61 Mb	48 Mb

^{*} Deconvolution programs were compiled using GCC and run in Linux on a 2.8 GHz Intel Pentium4 workstation.