Real-time GPU-based 3D Deconvolution

Marc A. Bruce\(^1\) and Manish J. Butte\(^1,2,\ast\)

\(^1\)Stanford Immunology, Stanford University, Stanford, California 94305, USA
\(^2\)Department of Pediatrics, Stanford University, Stanford, California 94305 USA
*manish.butte@stanford.edu

Abstract: Confocal microscopy is an oft-used technique in biology. Deconvolution of 3D images reduces blurring from out-of-focus light and enables quantitative analyses, but existing software for deconvolution is slow and expensive. We present a parallelized software method that runs within ImageJ and deconvolves 3D images ~100 times faster than conventional software (few seconds per image) by running on a low-cost graphics processor board (GPU). We demonstrate the utility of this software by analyzing microclusters of T cell receptors in the immunological synapse of a CD4 + T cell and dendritic cell. This software provides a low-cost and rapid way to improve the accuracy of 3D microscopic images obtained by any method.

©2013 Optical Society of America

OCIS codes: (180.2520) Fluorescence microscopy; (100.1830) Deconvolution.

References and links

1. F. Sedarat, E. Lin, E. D. W. Moore, and G. F. Tibbits, “Deconvolution of confocal images of dihydropyridine and ryanodine receptors in developing cardiomyocytes,” J. Appl. Physiol. 97(3), 1098–1103 (2004).
2. B. Storrie, T. Starr, and K. Forsten-Williams, “Using quantitative fluorescence microscopy to probe organelle assembly and membrane trafficking,” Methods Mol. Biol. 457, 179–192 (2008).
3. K. Mehta, A. D. Hoppe, R. Kainkaryam, P. J. Wooff, and J. J. Linderman, “A computational approach to inferring cellular protein-binding affinities from quantitative fluorescence resonance energy transfer imaging,” Proteomics 9(23), 5371–5383 (2009).
4. A. Edelstein, N. Amodaj, K. Hoover, R. Vale, and N. Stuurman, Computer Control of Microscopes Using \(\mu\)Manager, Current Protocols in Molecular Biology (John Wiley & Sons, Inc., 2010).
5. J. Hoferock and N. Bell, “Thrust: A Parallel Template Library,” (2010), retrieved from http://www.meganewtons.com/.
6. J. B. Pawley, Handbook of Biological Confocal Microscopy (Kluwer Academic Publishers, 1995).
7. J. B. Sibirita, “Deconvolution microscopy,” Microscopy Techniques, 1288–1291 (2005).
8. Y. Hiruka, J. W. Sedat, and D. A. Agard, “Determination of three-dimensional imaging properties of a light microscope system. Partial confocal behavior in epifluorescence microscopy,” Biophys. J. 57(2), 325–335 (1990).
9. J. G. McNally, T. Karpova, J. Cooper, and J. A. Conchello, “Three-dimensional imaging by deconvolution microscopy,” Methods 19(3), 373–385 (1999).
10. R. P. Dougherty, “Diffraction PSF 3D,” (2005), retrieved from http://www.optinav.com/Diffraction-PSF-3D.htm.
11. L. Lucy, “An iterative technique for the rectification of observed distributions,” Astron. J. 79, 745 (1974).
12. W. Richardson, “Bayesian-based iterative method of image reconstruction,” J. Opt. Soc. Am. 62(1), 55–59 (1972).
13. D. S. Biggs, “3D Deconvolution Microscopy,” in Current Protocols in Cytometry (Wiley Online Library, 2010), Chap. 12, Unit 12 19, pp. 11–20.
14. A. Griffa, N. Garin, and D. Sage, “Hollow Bars,” retrieved from http://bigwww.epfl.ch/deconvolution/?p=bars.
15. A. Griffa, N. Garin, and D. Sage, “C. elegans embryo” retrieved from http://bigwww.epfl.ch/deconvolution/?p=bio.
16. S. C. Bunnell, D. I. Hong, J. R. Kardon, T. Yamazaki, C. J. McGlade, V. A. Barr, and L. E. Samelson, “T cell receptor ligation induces the formation of dynamically regulated signaling assemblies,” J. Cell Biol. 158(7), 1263–1275 (2002).
17. P. J. Lu, P. A. Sims, H. Oki, J. B. Macarthur, and D. A. Weitz, “Target-locking acquisition with real-time confocal (TARC) microscopy,” Opt. Express 15(14), 8702–8712 (2007).

1. Introduction

Confocal microscopy has become an important and standard tool in biology because it enables detailed three-dimensional examination of cellular specimens. In optical microscopy, the diffraction-limited nature of the optics reduces the ability to precisely localize photons, resulting in blurring, the major source of distortion. The use of the pinhole in confocal
microscopy improves localization but reduces the amount of light available for detection, thus increasing the contaminative effect of shot noise. Blurring is modeled in software as a point spread function (PSF), essentially a mathematically defined cone of light that a single fluorophore makes. Deconvolution corrects blurring and reduces the effect of noise by applying an inverse of the PSF to each point of the measured image. Because blurring affects downstream calculations, deconvolution is essential for proper analysis of colocalization measurements [1], for membrane trafficking studies [2], and for FRET studies [3].

Deconvolution and generation of the PSF are computationally costly because they entail numerous calculations and multiplication of large matrices of complex numbers. We present a method to almost instantaneously deconvolve 3D images obtained by any microscopy technique, including confocal, wide-field, or two-photon, by utilizing a low-cost graphics processor unit (GPU), leveraging its multitude of computational cores and capability to accelerate fast Fourier transforms (FFTs). We implemented our software to run on NVIDIA GPUs that support CUDA (Compute Unified Device Architecture).

2. Materials and methods

Confocal microscopy data sets

T cells were prepared from the spleen of an OT-II TCR transgenic mouse (Jackson Labs). These CD4+ T cells, which recognize the ovalbumin 323-339 peptide antigen, were purified by CD4 positive-selection with magnetic bead separation technology (Miltenyi Biotec). Dendritic cells (DCs) were obtained from the spleen of a C57BL/6 mouse (Jackson Labs) by magnetic bead purification for CD11c+ cells, activated with lipopolysaccharide (100 ng/mL, SigmaAldrich) for 6 hours. DCs and T cells were plated on glass-bottom Petri dishes in media (comprising RPMI 1640, 10% fetal calf serum, 10 mM HEPES, Pen/Strep), then pulsed with ovalbumin peptide. After 1 hour, the plate was moved to ice, and cells were surface stained with anti-TCR-beta-AlexaFluor 647 (BioLegend, clone H57-597) and anti-I-A/I-E-Alexa Fluor 488 (BioLegend, clone M5/114.15.2) for 30 min. The cells were then fixed with 4% paraformaldehyde plus 4% sucrose at room temperature for 10 minutes, washed, permeabilized with 0.1% Triton X-100 for one minute, washed, stained with AlexaFluor 568-labeled phalloidin (Invitrogen), washed again, and imaged in SlowFade Gold (Invitrogen).

Images were collected using a Nikon Ti-E microscope through a 1.4 NA PlanApo 100x objective using a Yokagawa spinning disk confocal. MicroManager was used to operate the microscope and collect the images [4]. Specimens were illuminated using 488 nm, 561 nm, and 641 nm laser excitation. Z sections were collected using a piezo z-stage (MadCity Labs), taking steps of 100 nm between sections. Images for this paper were processed in our software in ImageJ.

PSF and Deconvolution Software

Software for deconvolution and generation of the PSF was written in C++ using the NVIDIA CUDA libraries, version 5.0. The software requires CUDA compute capability 1.0 or better, supporting any card with CUDA capabilities. We used the Thrust library version 1.6 to facilitate CUDA coding [5]. The interface to ImageJ was written in Java and C++.

Timings for the generation of PSFs and deconvolved images were obtained running on an NVIDIA GTX580, driver version 304.54, running in Ubuntu 11.04 64-bit on a quad-core 2.8 GHz Intel Core i7-930 processor with 24 GB RAM. Benchmark tests were run in up-to-date versions of Fiji. This same computer was used for the comparisons where GPU-based calculations were compared with CPU-based ones. Deconvolution software can be downloaded from http://tcell.stanford.edu.

3. Point spread function

To begin, we developed GPU-based software to generate a PSF using the Fraunhofer approximation of diffraction through a pinhole [6] that is customized to the dimensions of the image; the numerical aperture of the objective, and the emission wavelength of light. The
cylindrical symmetries in the equations allowed us to accelerate generation of the PSF by using parallel threads to calculate values of the PSF equation at integer radii. First, we created a grid of thread blocks of width \( v \) and height \( u \) (Fig. 1(a)), with each block holding 512 threads. These threads calculated the integral of the PSF at each position in the \( u-v \) slice. We then used parallel processing blocks to interpolate from the integer positions, which greatly reduced the number of calculations and memory accesses as compared with direct calculations at each point (Fig. 1(b)). Despite the potential for accuracy loss due to the interpolation, these PSFs were nearly identical to those calculated directly (root mean square error (RMSE) \( \sim 0.016\% \)). Because of octant symmetry, we needed to calculate only one octant of the PSF and the others are generated by rotation and translation. To compare our method against CPU-based software, we timed the generation of PSFs of typical sizes and found our software ran more than 300 times faster than CPU-based software (Fig. 1(b)). For example, calculation of a PSF for a very large image \((1024 \times 1024 \times 32 \text{ pixels})\) took 0.33 s using our GPU-based software, as compared to 99 s using the CPU. Wide-field PSFs are calculated in a similar way, based on the understanding that confocal PSFs are the square of wide-field PSFs. Our software also allows the user to provide their own PSF as obtained by empirical measurement or another software tool. Huygens provides such a tool to derive a PSF from 3D images of sub-diffraction beads. Also, many papers describe the underlying process [7–9].

3. Deconvolution results

Our deconvolution software uses a standard Richardson-Lucy maximum likelihood algorithm (R-L) [11, 12]. The iterative method of R-L reduces noise contamination such as the Poisson noise of photon counting, commonly observed in confocal imaging techniques with low photon fluxes. The algorithm calculates corrected images iteratively, using the following formula (where \( o \) is the original image, \( h \) is the PSF, \( h^* \) is the complex conjugate of the PSF, and \( i_k \) is the \( k^{th} \) iteration):

\[
i_{k+1} = i_k h^* \left( \frac{o}{i_k \otimes h} \right)
\]

The noise model for wide-field microscopy with its larger photon fluxes approaches Gaussian noise, but wide-field images can still be deconvolved using R-L [13]. With an appropriate PSF, this algorithm can thus be applied to multiple modes of microscopy. The largest speed gains were accomplished by implementing the convolution steps of the R-L algorithm on the GPU, noting that a convolution in real space is equivalent to a pointwise multiplication in Fourier space. The CUFFT library provided by NVIDIA is a highly tuned FFT implementation that is generalized to arbitrary dimensions, allowing our software to operate efficiently on images of any size. Processing all arithmetic operations in parallel on the GPU further accelerates software execution.

Our software automatically terminates iterating when the change in mean square error between subsequent iterations falls below 2.5% (comparing the original image to the current iteration). Alternatively, the software allows the user to exactly specify the number of iterations performed.
Fig. 1. (a) Parallel algorithm for calculation of the point spread function (PSF). The PSF has radial symmetry, so we start by calculating one radial slice of the PSF at integer radii using cylindrical coordinates. Each point of the radius-depth slice grid entails numerical integration of the Airy function, which is performed in parallel. The inset cube is the final PSF, and the points of one octant are calculated by interpolating from the cylindrical coordinates. Then the octants are mapped using symmetry operations. (b) Timing to calculate the point spread function (PSF) for a variety of image sizes and using our GPU-based parallel, interpolation algorithm; using a CPU-based publically available package, Diffraction PSF 3D [10], to calculate.

To analyze the performance of our software, we tested our program on three images. We first analyzed a standardized test set, a synthetic image comprising five hollow cylinders. Gaussian (10 dB signal-to-noise ratio (SNR)) and Poisson noise (30 dB SNR) functions had been applied to the cylinders [14]. Our software was able to produce an excellent recreation of the true image after 25 iterations (Fig. 2(a)), which took 0.67 s to complete. We compared the performance of the GPU-based code against near-identical CPU-based code by replacing all CUFFT calls with FFTW and parallel arithmetic with OpenMP. We found an increase in speed using the GPU of over 150x (Fig. 2(b)). In a test image of biological data (a C. elegans...
embryo [15]), we found that GPU calculation of a PSF required ~300 ms less than transfer of a pre-computed PSF, a savings of ~5%. Depending on image size, memory bandwidth, system configuration and other factors, there may be no speed penalty (or advantage) to using an empirical PSF as compared to a calculate PSF.

![Image](image.png)

**Fig. 2.** Deconvolution of test data. (a) Richardson-Lucy deconvolution is applied to the noisy bars data set (left) for 25 iterations (middle), compared to the underlying true image (right). (b) Timing of deconvolution for given number of iterations to deconvolve the bars data set by CPU-based and GPU-based software.

To deconvolve real-world data, we processed a confocal, two-color image of a helper T cell interacting with an antigen-pulsed dendritic cell, stained to reveal actin. The dimensions of the image are 400 x 400 x 96 x 3 colors. Twenty iterations of the R-L algorithm, taking a total of 2.1 s (including the time required to calculate the three PSFs), or approximately 0.7 s per color, were sufficient to produce an excellent image essentially free of lateral or axial blurring. Deconvolution eliminates excess noise to reveal that actin localizes in the T cell to a ring shape in the region of the immunological synapse (IS), and in the dendritic cell at three anchoring points in addition to the IS (blue, Fig. 3(a)). A slice of this image at the immunological synapse shows that TCR microclusters are apparent in the interface (Fig. 3(b)). To evaluate the improved resolution after deconvolution, the full width at half maximum of a single CD4 microcluster was calculated to be 0.6 µm (in agreement with previous estimates [16]), whereas the raw image shows lower signal to noise and a wider estimate of the microcluster size (Fig. 3(c)). To analyze the performance of our deconvolution software versus CPU-based software, we deconvolved test images for various numbers of iterations and of various sizes (Fig. 4(a)), and found a typical speed-up of more than 150-fold. These examples show that our software can be used to deconvolve real-world 3D microscopy images in a fraction of a second.
Fig. 3. Deconvolution of T cell-Dendritic cell immunological synapse. Deconvolved image shows two T cells, where right-most one is interacting with antigen presenting cell, fixed, permeabilized and stained with AlexaFluor 568-labeled phalloidin to reveal the distribution of intracellular actin. (a) Shows 3D rendering of cells, where red portrays TCR staining, green I-A\(^{\alpha}\) (MHC class II), and blue phalloidin. (b) Shows a slice at the synapse, revealing the improvement in visualization of structures in the immunological synapse. (c) Line scan from (b) shows improved visualization of TCR microclusters in the interface.
4. Conclusions

Deconvolution has historically been slow because of the computational intensity of the algorithms. Note that to collect a typical z-frame of a spinning-disk confocal 3D image may take 0.1 s, and a typical 30-stack z-series may require 3 or more seconds. Our software deconvolves such images in under a second, ~10 times faster than they can be collected, and about as fast as they can be written to the hard drive. This speed allows the possibility of the microscope control software to display deconvolved images as fast as the data are collected. By showing deconvolved images immediately, the user could adjust experimental parameters on the fly. For example, laser powers in a FRET experiment could be altered in real-time to reduce the risk of photobleaching or to ensure that adequately bright data were captured. Software that tracks objects in 3D using confocal microscopy could be similarly improved by real-time deconvolution [17]. Commercial deconvolution software is expensive (generally, thousands of dollars per year) and runs on restricted operating system platforms, which may...
not suit labs in academia, in the non-for-profit sector, or in resource-poor settings. Our software will be widely available, free to academics and non-profits, and runs in the widely used and freely available environment of ImageJ (Fig. 4(b)) on all major platforms (Windows, MacOS X, Linux). No significant differences in runtime have been observed between operating systems on the same hardware, making the software truly multi-platform. The software even automatically deconvolves each color of multi-color images. We envision our software will broadly improve the accuracy of 3D fluorescence microscopy across many applications in cell biology.

Acknowledgments

We are grateful for the gift of a video card from NVIDIA through their Academic Partnership Program. Funding for this project came from the NIAID/NIH (K08 AI079268) to M.J.B. and to the Stanford Graduate Fellowship (to M.A.B.).