Optics Letters

Artificial neural network approaches for fluorescence lifetime imaging techniques

Gang Wu,[1,2] Thomas Nowotny,[2] Yongliang Zhang,[1,3] Hong-Qi Yu,[4] and David Day-Uei Li[1,*]

[1]*Centre for Biophotonics, University of Strathclyde, Glasgow G4 ORE, UK*
[2]*School of Engineering and Informatics, University of Sussex, Brighton BN1 9QJ, UK*
[3]*College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310007, China*
[4]*School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China*
Corresponding author: David. Li@strath.ac.uk

Received 24 March 2016; revised 27 April 2016; accepted 30 April 2016; posted 2 May 2016 (Doc. ID 261876); published 25 May 2016

A novel high-speed fluorescence lifetime imaging (FLIM) analysis method based on artificial neural networks (ANN) has been proposed. In terms of image generation, the proposed ANN-FLIM method does not require iterative searching procedures or initial conditions, and it can generate lifetime images at least 180-fold faster than conventional least squares curve-fitting software tools. The advantages of ANN-FLIM were demonstrated on both synthesized and experimental data, showing that it has great potential to fuel current revolutions in rapid FLIM technologies.

OCIS codes: (030.5260) Photon counting; (110.0180) Microscopy; (110.2960) Image analysis; (170.3650) Lifetime-based sensing; (170.6920) Time-resolved imaging; (100.4996) Pattern recognition, neural networks.

http://dx.doi.org/10.1364/OL.41.002561

Fluorescence lifetime imaging microscopy (FLIM) is a powerful imaging technique that not only detects fluorescence intensity but also probes the local environment of the fluorophores. For example, FLIM can monitor physiological parameters such as pH, O_2, Ca^{2+}, NAD(P)H, temperature, or live cellular processes (e.g., protein–protein interactions). This helps scientists to understand diseases or develop therapies [1–4]. There are time-domain (TD) and frequency-domain (FD) FLIM solutions. Fast FD FLIM systems have been commercially available, but time-correlated single-photon counting (TCSPC) approaches, which have superior timing resolution, remain the gold standard within the FLIM community.

Recent developments in large single-photon avalanche diode (SPAD) arrays and multi-channel systems have significantly boosted the acquisition speed, making TCSPC approaches promising for real-time applications. These advanced systems, however, create massive data throughput making FLIM analysis even more challenging [5–7]. Taking a 512×512 FLIM image as an example, if each pixel contains a fluorescence histogram generated from an 8-bit TCSPC module (256 time bins, and assume that each time bin has a capacity of 10-bit), then the data throughput would be $512 \times 512 \times 2560 = 6.7$ Gbits. Some novel data compression techniques have been introduced to reduce the data throughput [7,8], but they are based on single-exponential models and might neglect important biological information. On the other hand, traditional gating approaches have been reported to be able to provide a frame rate higher than 5 fps using only two-time gates, but only producing single-exponential approximations. Or for laboratory FLIM experiments a much larger number of gates (>10) are usually used to cover a wider range of lifetime variations within the field of view [9]. For a gate number larger than 5, FLIM data analysis usually use iterative least square methods (LSM). The analysis hence becomes much slower, as the amount of data increases.

High-speed FLIM is key to unveiling dynamic cellular processes, but emerging rapid systems make image analysis increasingly challenging. Almost all commercial FLIM analysis tools are LSM based [10,11], and the general belief of users that they are the gold standard has probably deterred applications of non-LSM approaches. However, LSM approaches usually require experienced users to supervise and provide proper initial conditions or manual interventions, and they usually need iterative curve-fitting computations, making real-time analysis impossible. In this Letter, we propose an artificial neural network (ANN) approach to tackle this problem. As we will demonstrate below, ANN does not need specific initial conditions, holding the promise for automated applications.

ANNs were first inspired by biological neural networks and have been widely used in a variety of areas, such as systems identification and control, classification, time series forecasting, robotics, and medical image analysis [12–16]. Similar to Google's famous artificial intelligence computer program AlphaGo [17], an ANN needs to be trained before it can be employed. ANNs have the ability to perform regression (also known as function approximation) [15,16], and for FLIM analysis, they can be used to approximate the function that maps histogram data as input values onto the unknown lifetime parameters. To seek faster analysis, we will apply feed-forward

0146-9592/16/112561-04 Journal © 2016 Optical Society of America

Corrected 26 May 2016

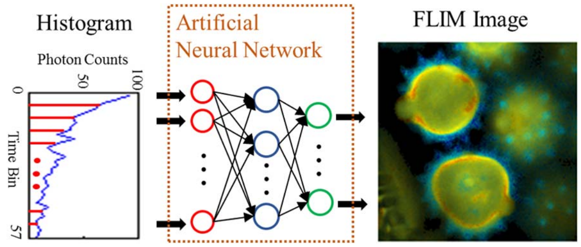

Fig. 1. Principle of the ANN-based FLIM analysis.

ANN topologies [12,16] to avoid iterative computations. To demonstrate the potential of ANN-FLIM, we will compare its precision and accuracy performances with LSM.

An ANN mimics how interconnected neurons in a human brain make sense of new external stimuli by comparing them with previous experiences [12]. Figure 1 shows a trained ANN interpreting a fluorescence histogram. The connections between neurons determine how the ANN reacts to input stimuli and are determined through supervised learning, i.e., through training on example data with known lifetime properties. Unlike LSM, which treats each image as a new problem and analyzes it from scratch, a trained ANN can directly calculate FLIM images based on the experience accumulated during training without iterations. To demonstrate how the ANN approach performs, we used a bi-exponential model and compared it against LSM [9,18]. We will also apply it to analyzing lifetime images of daisy pollens later. We assume the fluorescence decay is $f(t) = K \cdot [f_D \cdot \exp(-t/\tau_F) + (1 - f_D) \cdot \exp(-t/\tau_D)] \cdot u(t)$, where, τ_F, τ_D are the lifetimes, K the pre-scalar, f_D the proportion, and $u(t)$ the step function, as in Ref. [3]. Background correction can be easily carried out [19,20] before the ANN-FLIM analysis; it only adds minor extra efforts. The instrument response function (IRF) usually plays a crucial role, especially when the full-width half-maximum of the IRF is large. However, to quickly demonstrate the ANN approach, we will leave it to future reports.

For this study we use the simple ANN structure shown in Fig. 2. It contains one input layer, one output layer, and two hidden layers. The neurons of the output layer have linear transfer functions, whereas the other neurons in the input and hidden layers are configured with sigmoid transfer functions [14], as demonstrated in Eq. 1:

$$y_{k\_l} = x = \sum_{j=0}^{m} w_{kj} x_j$$

$$y_{k\_s} = \frac{2}{1+\exp(-2*x)} - 1 = \frac{2}{1+\exp(-2*\sum_{j=0}^{m} w_{kj} x_j)} - 1, \quad \textbf{(1)}$$

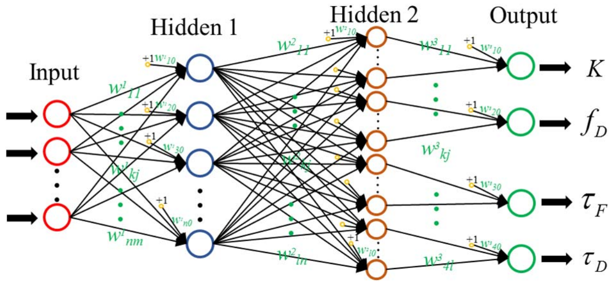

Fig. 2. Architecture of the ANN used in this Letter. Inputs are the photon counts and outputs are the parameters K, f_D, τ_F, and τ_D.

where y_{k_l} and y_{k_s} represent the output of k-th neuron with either the linear or sigmoid transfer function, respectively, w_{kj} is the weight of j-th input of k-th neuron (see Fig. 2).

As one of the most common architectures, the feed-forward network only allows the neurons in each layer to transfer information to neurons in the next layer (from the input layer to the hidden layers, and then ultimately to the output layer). The number of neurons in the input layer depends on the number of time bins in the histogram (57 time bins in this study). As for the output layer, four neurons are used (more neurons can be used in the future) to generate K, f_D, τ_F, and τ_D, respectively. The two hidden layers implement the transformation from 57 inputs to the four outputs, which include implicitly the underlying relationship among the four outputs.

The weights w_{kj} encode the mapping of histograms to outputs and are initially unknown. They can be deduced (or trained) from a large number of synthesized sample sets, i.e., synthesized histograms and matching vectors $\alpha = [K, f_D, \tau_F, \tau_D]$. Ideally, one would like to recover the true underlying α based on a noisy histogram generated based on this α, and we originally have tried to do this. However, the fully trained ANNs failed to recover the true α in the majority of cases (data not shown). This is compatible with the observation that other leading FLIM algorithms are in many circumstances also unable to recover the true α due to ambiguities caused by noise. As the next best goal to recovering the true parameters, we now train the ANN to recover the results of a maximum likelihood estimation (MLE) of the parameters obtained from the same histogram. Figure 3 illustrates the corresponding process for preparing training samples. First, we choose a set of α vectors from within the working range (the range of each parameter is usually known in advance; this is not strictly necessary, but it helps speed up the training). In order to create an ANN that is suitable for the whole working range of possible output values, we suggest to control the distributions of parameters using low-discrepancy sampling [21]. Second, photon count histograms are generated for each α vector, by adding Poisson noise [22,23], as illustrated by the green curve in the histogram block of Fig. 3. Third, and *this is the core innovation in this method*, the valid training targets α^* are obtained using MLE [24,25], which is known as one of the best FLIM analysis algorithms, on the generated synthetic histograms. In brief, to perform MLE, the definite integral of fluorescence decay $\Lambda(t)$ is given by Eq. (2), and the expected value EN_i of each time bin can be obtained as in Eq. (3). Therefore, the likelihood of the observed histogram decay is given by Eq. (4), and the preferable targets of the training samples can be acquired based on minimizing this likelihood function:

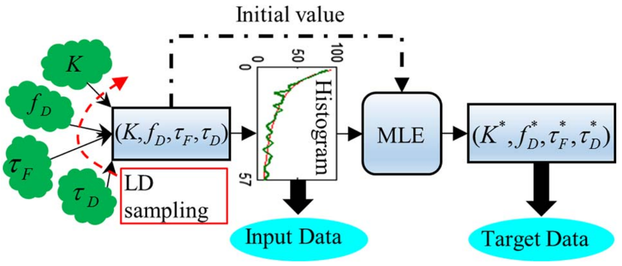

Fig. 3. Sample preparation for network training.

$$\Lambda(t) = \int_0^t f(t)\mathrm{d}t$$
$$= -A(\tau_F f_D \exp(-t/\tau_F) + \tau_D(1 - f_D)\exp(-t/\tau_D))|_0^t, \quad (2)$$

$$EN_i = \Lambda(ih) - \Lambda((i-1)h)$$
$$= K[\tau_F f_D \exp(-ih/\tau_F)(\exp(h/\tau_F) - 1)$$
$$+ \tau_D(1 - f_D)\exp(-ih/\tau_D)(\exp(h/\tau_D) - 1)], \quad (3)$$

$$L = \prod_{i=1}^{m} \frac{EN_i^{N_i}\exp(-EN_i)}{N_i!}, \quad (4)$$

where N_i is the photon count of the ith TCSPC time bin, m is the number of time bins, and h is the bin width.

The ANN is then trained by iteratively updating the weights to minimize the error between the outputs of the ANN and the target parameters α^*. Weight updates can be accomplished with the supervised backpropagation learning method [16]. The normalized mean squared error is used as the error function:

$$F_{\mathrm{mse}} = \frac{1}{N}\sum_{1}^{N}[w_i(\alpha_{Ti} - \alpha_{Oi})^2], \quad (5)$$

where N is the number of total observations, and α_{Ti} and α_{Oi} represent target output and network output, respectively.

The weights w_i can be used to emphasize one or some of the parameters in different circumstances. To train the ANN, the MATLAB neural network toolbox was used, which provides flexible and sufficient algorithms such as conjugate gradient methods that deliver faster training progress with fewer resources required, or the Levenberg–Marquardt algorithm (LM), which provides higher precision but needs more memory and takes more time. Once the ANN is trained, it is straightforward to use it to obtain lifetime estimates from input photon count histograms by applying the photon counts as inputs and propagating the activity through the network (Fig. 2) according to the learned weights. The outputs of the network are the estimated lifetime parameters.

We compared ANN-FLIM with the widely used LSM method (MATLAB nonlinear least square routine [26] with the "*trust-region-reflective*" option) using Monte-Carlo simulations. In the simulations, each synthesized histogram has less than 900 total photon counts, with $m = 57$, and $h = 333$ ps. In this case, 210,000 samples were used to train the network, and the training procedure took about 4 h. Figures 4(a), 4(c), and 4(e) show the precision (F-value [7], $F = N_C^{0.5}\sigma g/g$, N_c is the photon counts of each pixel, $g = f_D, \tau_F$, or τ_D) plots, whereas Figs. 4(b), 4(d), and 4(f) show the bias plots, for τ_F, f_D and τ_D, respectively. For Figs. 4(a)–4(d), $\tau_D = 2.5$ ns, and for Figs. 4(e) and 4(f), $\tau_F = 0.58$ ns. The optimized regions (for the F-value and the bias) of LSM are different from those of ANN. Figs. 4(a), 4(c), and 4(e) show that ANN can provide a wider optimized area for the F-value for all τ_F, τ_D, and f_D. Figs. 4(b), 4(d), and 4(f) show that LSM produces slightly less biased τ_F, whereas ANN offers wider optimized areas for both τ_D, and f_D. Although there are differences between them, their estimations are in the same order. Using MATLAB computation on a Windows PC [Intel(R) Xeon(R) E31245 processor with 16 GB memory] on 256 x 256 images, the proposed ANN-FLIM (0.9 s) is a staggering 180-fold faster than LSM (166s). Based on our previous experience in parallel FLIM analysis, using the state-of-the-art GPUs [27] may boost calculation speed an additional 30-fold (this is due to the fact that ANN mapping mainly contains matrix multiplications, suitable for parallel computing), showing that an ANN-based analysis tool has great potential to enable real-time or even video-rate FLIM imaging.

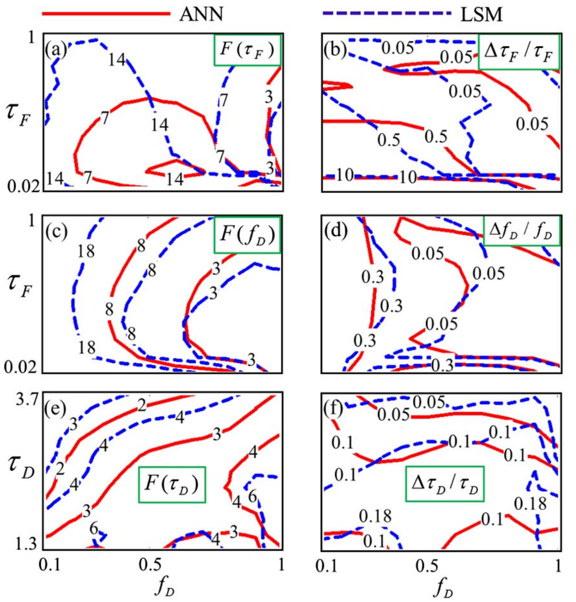

Fig. 4. Performances of ANN and LSM: F-value of (a) τ_F, (c) f_D, and (e) τ_D; the bias of (b) τ_F, (d) f_D, and (f) τ_D.

The potential of ANN-FLIM was also demonstrated by testing it on experimental data against LSM. FLIM experiments were performed on daisy pollens using the MicroTime 200 time-resolved confocal fluorescence microscope (PicoQuant, Germany). The MicroTime 200 was equipped with the standard piezo scanner (Physik Instrumente; 100 × 100 μm scan range) and a SPAD (SPCM-AQRH from Excelitas). The excitation source was a ps-pulsed diode laser (LDH-D-C-485) operating at 485 nm with the pulse frequency of 20 MHz (50 ns for the TCSPC dynamic range), which was controlled by the PDL 828 "Sepia II" laser driver. The data was acquired by the HydraHarp 400 (bin width set to 8 ps), and the image size was 400 × 400. For FLIM analysis the bin width was 0.32 ns. The intensity image of daisy pollens is shown in Fig. 5(a). Figures 5(b)–5(g) compare the τ_F and average lifetime $\tau_{\mathrm{Average}}[\tau_{\mathrm{Average}} = f_D \cdot \tau_F + (1 - f_D) \cdot \tau_D]$ maps for ANN and LSM. From these images, it is easy to see that ANN is capable of extracting the features of the sample, and it shows similar results to those of LSM, especially for the images of average lifetime. On the other hand, comparing Figs. 5(b) and 5(c) shows that in some pixels LSM failed to converge to correct estimations (dark red spots), which is due to its high sensitivity to initial conditions (to improve it might require quick estimations on f_D, τ_F, and τ_D taking more analysis time), whereas ANN provides a superior success rate. To be specific, the success rate is 99.93%, for ANN and 95.93% for LSM. Figures 5(d) and 5(e) show similar merged images (intensity and τ_{Average}) for ANN and LSM, respectively. Figures 6(a) and 6(b) also shows that ANN produces similar τ_F, τ_{Average} and f_D histograms with LSM, except that LSM has more invalid pixels around $\tau_F \sim 0$ ns and there is a slight difference in τ_D. The difference in τ_D is likely due to the different

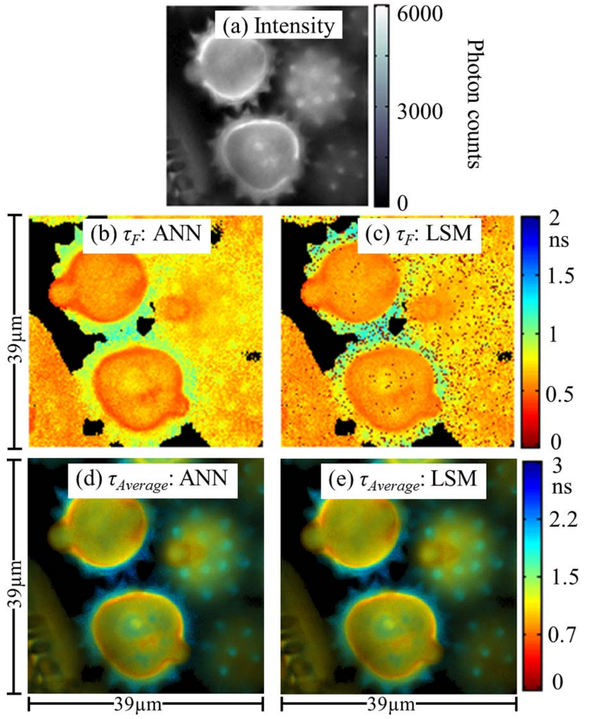

Fig. 5. (a) Intensity image, (b) ANN and (c) LSM τ_F images, (d) ANN and (e) LSM merged intensity and τ_{Average} images.

bias behaviors when f_D is closer to 1, Fig. 6(b), and $\tau_D > 3$ ns. However, LSM and ANN still show similar τ_{Average} histograms.

A more convincing feature is that ANN (1.8 s) is 566 times faster than LSM (1019.5 s). Together with the observation from the previous analysis on synthesized data, the speed of LSM analysis is subject to the choice of initial conditions, whereas ANN does not require any initial conditions. This Letter gives a quick demonstration on the potential of ANN approaches in FLIM analysis. To provide thorough assessments, more detailed analyses considering the IRF, multi-exponential decays, or complex neuron models will be included soon.

To summarize, we have proposed an ANN-based high-speed FLIM analysis method. To our knowledge, this is the first time that a machine learning algorithm has been successfully introduced into FLIM analysis. Compared with LSM, the results reveal that ANN not only provides comparable or even better performances, but also offers much faster high-throughput data analysis. Thanks to recent advances in parallel computing technologies, it promises real-time or video-rate FLIM analysis if combined with the latest GPU devices.

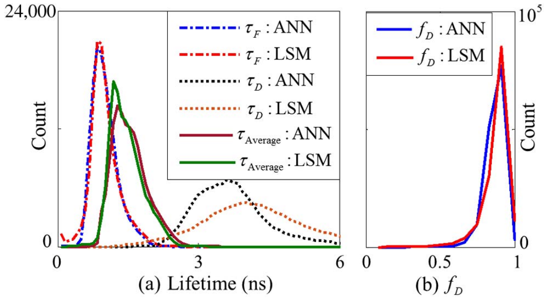

Fig. 6. (a) Lifetime and (b) f_D histograms of the experimental data.

Funding. China Scholarship Council (CSC); Royal Society (140915); Engineering and Physical Sciences Research Council (EPSRC) (EP/J019690/1); Nvidia (nVIDIA Graduate Fellowship).

Acknowledgment. We would like to thank Dr. Andreas Bülter and Dr. Volker Buschmann, PicoQuant, Germany, for their technical support.

REFERENCES

1. K. Okabe, N. Inada, C. Gota, Y. Harada, T. Funatsu, and S. Uchiyama, Nat. Commun. **3**, 705 (2012).
2. M. A. Yaseen, S. Sakadzic, W. Wu, W. Becker, K. A. Kasischke, and D. A. Boas, Biomed. Opt. Express **4**, 307 (2013).
3. A. Leray, S. Padilla-Parra, J. Roul, L. Heliot, and M. Tramier, PLoS One **8**, e69335 (2013).
4. S. Coda, A. J. Thompson, G. T. Kennedy, K. L. Roche, L. Ayaru, D. S. Bansi, G. W. Stamp, A. V. Thillainayagam, P. M. French, and C. Dunsby, Biomed. Opt. Express **5**, 515 (2014).
5. S. P. Poland, N. Krstajić, J. Monypenny, S. Coelho, D. Tyndall, R. J. Walker, V. Devauges, J. Richardson, N. Dutton, P. Barber, D. D.-U. Li, K. Suhling, T. Ng, R. Henderson, and S. Ameer-Beg, Biomed. Opt. Express **6**, 277 (2015).
6. R. M. Field, S. Realov, and K. L. Shepard, IEEE J. Solid-State Circuits **49**, 867 (2014).
7. D. D. U. Li, J. Arlt, D. Tyndall, R. Walker, J. Richardson, D. Stoppa, E. Charbon, and R. K. Henderson, J. Biomed. Opt. **16**, 096012 (2011).
8. D. D. U. Li, S. Ameer-Beg, J. Arlt, D. Tyndall, R. Walker, D. R. Matthews, V. Visitkul, J. Richardson, and R. K. Henderson, Sensors **12**, 5650 (2012).
9. T. Omer, L. Zhao, X. Intes, and J. Hahn, J. Biomed. Opt. **19**, 086023 (2014).
10. S. Chakraborty, F. S. Nian, J. W. Tsai, A. Karmenyan, and A. Chiou, Sci. Rep. **6**, 19145 (2016).
11. W. Becker, *The Bh TCSPC Handbook* (Becker & Hickl, 2015).
12. J. Jiang, P. Trundle, and J. Ren, Comput. Med. Imaging Graph. **34**, 617 (2010).
13. A. Mellit and A. M. Pavan, Solar Energy **84**, 807 (2010).
14. J. A. K. Suykens, J. Vandewalle, and B. L. R. D. Moor, *Artificial Neural Networks for Modelling and Control of Non-linear Systems* (Kluwer Academic, 1996).
15. S. S. Haykin and S. S. Haykin, *Neural Networks and Learning Machines* (Prentice-Hall, 2009).
16. M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design* (PWS, 1996).
17. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).
18. W. H. Press, *Numerical Recipes in C: the Art of Scientific Computing* (Cambridge University, 1992).
19. D. U. Li, R. Walker, J. Richardson, B. Rae, A. Buts, D. Renshaw, and R. Henderson, J. Opt. Soc. Am. A **26**, 804 (2009).
20. S. P. Poland, A. T. Erdogan, N. Krstajić, J. Levitt, V. Devauges, R. J. Walker, D. D.-U. Li, S. M. Ameer-Beg, and R. K. Henderson, Opt. Express **24**, 6899 (2016).
21. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods* (Wiley, 1964).
22. D. S. Elson, I. Munro, J. Requejo-Isidro, J. McGinty, C. Dunsby, N. Galletly, G. W. Stamp, M. A. A. Neil, M. J. Lever, P. A. Kellett, A. Dymoke-Bradshaw, J. Hares, and P. M. W. French, New J. Phys. **6**, 180 (2004).
23. P. J. Verveer, A. Squire, and P. I. Bastiaens, Biophys. J. **78**, 2127 (2000).
24. P. Hall and B. Selinger, J. Phys. Chem. **85**, 2941 (1981).
25. T. A. Laurence and B. A. Chromy, Nat. Methods **7**, 338 (2010).
26. T. F. Coleman and Y. Y. Li, SIAM J. Optim. **6**, 418 (1996).
27. N. Wilt, *The CUDA Handbook: a Comprehensive Guide to GPU Programming* (Addison-Wesley, 2013).