



Original Software Publication

ConfocalGN: A minimalistic confocal image generator

Serge Dmitrieff*, François Nédélec

EMBL, Cell Biology and Biophysics, Meyerhofstrasse 1, Heidelberg, Germany



ARTICLE INFO

Article history:

Received 24 February 2017
 Received in revised form 24 August 2017
 Accepted 15 September 2017

Keywords:

Synthetic image
 Image analysis
 Bioinformatics

ABSTRACT

Validating image analysis pipelines and training machine-learning segmentation algorithms require images with known features. Synthetic images can be used for this purpose, with the advantage that large reference sets can be produced easily. It is however essential to obtain images that are as realistic as possible in terms of noise and resolution, which is challenging in the field of microscopy. We describe ConfocalGN, a user-friendly software that can generate synthetic microscopy stacks from a ground truth (i.e. the observed object) specified as a 3D bitmap or a list of fluorophore coordinates. This software can analyze a real microscope image stack to set the noise parameters and directly generate new images of the object with noise characteristics similar to that of the sample image. With a minimal input from the user and a modular architecture, ConfocalGN is easily integrated with existing image analysis solutions.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version	v1.1
Permanent link to code/repository used for this code version	https://github.com/SergeDmi/ConfocalGN/
Legal Code License	GPL3.0
Code versioning system used	Git
Software code languages, tools, and services used	MATLAB
Operating environments	MATLAB
If available Link to developer documentation/manual	https://github.com/SergeDmi/ConfocalGN/blob/master/README.md

1. Motivation and significance

Fluorescence microscopy is widely used in the bioscience community to observe biological sample. Methods such as light sheet microscopy and confocal microscopy are favored due to their good spatial resolution and high contrast compared to conventional widefield microscopy. However, these methods still suffer from the diffraction limit, detector shot noise, background fluorescence, tagging heterogeneity and fluorophore stochasticity [1,2]. Moreover, the resolution is anisotropic, and typically several times worse in the direction of the optical axis compared to the other directions for a confocal microscope. All these effects limit our ability to infer the shape of an object observed through the microscope.

Image analysis is the standard approach to extract information from microscopy data. It usually involves image processing, such as background subtraction and removal [3], or deconvolution [4,5]. Additional steps can perform segmentation or directly attempt to reconstruct the objects using a variety of techniques, such as

model-based analysis, feature classification [6–9], etc. Image analysis tools tend to include elaborate pipelines with adjustable parameters and for this reason can introduce artifacts with possibly systematic bias [10], that may remain unnoticed. It is thus essential to assess the quality of the analysis procedure, and this can be done by running the analysis pipeline with synthetic images, for which the ground truth (“digital phantom” [11]) is known. In fluorescence microscopy, this ground truth should ideally be the list of fluorophore coordinates and their intensity, or the geometry of the object under scrutiny. However, most published image analysis pipelines are not systematically validated, perhaps because existing synthetic image generators are difficult to use or insufficiently known. This prompted us to develop an easily accessible software allowing the generation of synthetic microscopy images.

To simulate a confocal image, one needs to convolve the ground truth with the point spread function (PSF) of the microscope, adding noise both before and after convolution, according to the nature of the different sources of noise. The PSF used for this simulation can be either predicted from the optical path using tools such as *Huygens* or *Icy/Fiji PSF Generator* [12], or measured from experimental images using programs such as *Huygens* or

* Corresponding author.

E-mail address: serge.dmitrieff@embl.de (S. Dmitrieff).

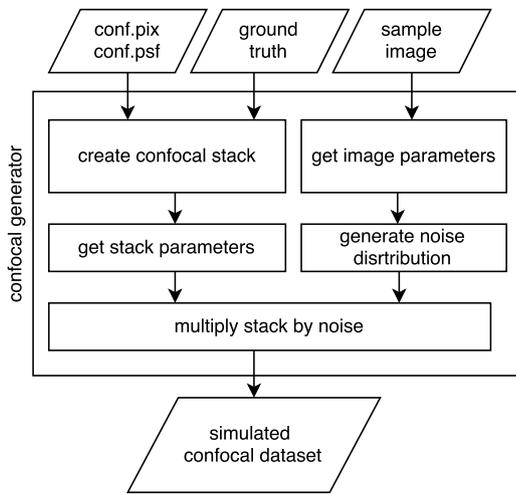


Fig. 1. Flow chart of ConfocalGN.

Mosaic [13]. Different sources of noise can be simulated bottom up : detector sensitivity, shot noise [14] and background noise [15]. Generally the operation involves the convolution of different types of distributions with the PSF, which can be computed analytically only in idealized experimental conditions [16].

Several software allow the generation of simulated image from a known PSF and/or known noise sources, including DeconvolutionLab2 [17], Huygens professional, BlurLab [18], and SMEAGOL for single-particle tracking [19]. Some software offer very advanced functionality, perhaps at the price of operative simplicity, code modularity or re-usability. We offer ConfocalGN, a user-friendly, modular and well documented software allowing the generation of realistic synthetic confocal images, that can be readily expanded to other kinds of microscopy. All the necessary operations are performed by MATLAB core functions, requiring no toolbox. It offers in particular the possibility to automatically calibrate the level of noise from an existing microscopy image. Thus requiring very little input, ConfocalGN can readily be used without specialist knowledge.

2. Software functionalities

ConfocalGN uses a top-down approach. It aims at generating realistic synthetic microscopy images with minimal input from its user. The input is organized in three fields, Fig. 1:

- The ground truth, provided as a high resolution image or as a list of 3D fluorophore coordinates.
- The microscope characteristics : PSF and voxel size.
- The noise distribution and mean signal value, provided either as numeric values or extracted automatically from a (real) sample image.

With this input specified, ConfocalGN will automatically generate a synthetic microscopy image without additional user intervention. ConfocalGN is a modular program and the different steps that are necessary to generate images (Fig. 1) are performed by separate routines that are all included within the distribution :

- Generate a ground truth image from a set of fluorophore coordinates
- Extract background noise properties and mean signal value from a (real) microscopy image
- Convolve a ground truth image with a PSF
- Extract a confocal stack from the convolved image

- Reproduce the noise properties and signal-to-noise ratio obtained experimentally

The PSF can be approximated by a Gaussian, which is usually quite accurate for a confocal microscope [20], or can be an image provided by the user. ConfocalGN does not provide a mean to compute the parameters of the PSF, but this task can be done with Huygens, PSF generator, PSF lab, etc. [12,21]. By default, the synthetic noise is generated using a gamma distribution, which is a good approximation of reality [16], but Poisson and Gaussian noise are supported. We limited ourselves to a uniform description of the image background for simplicity, since a spatially variable background cannot be generic enough for all kinds of samples and experimental setups.

While the software is focused on confocal microscopy, its ability to use a generic PSF makes it ready to simulate other kinds of microscopy, such as light sheet. The ability to extract noise and signal information from a (real) microscopy image streamlines the process of image generation. ConfocalGN can easily be modified to alter the segmentation function, the nature and uniformity of the background, or the convolution method.

3. Software architecture

ConfocalGN relies on a set of MATLAB-implemented functions that can be scripted (as in `cgn_example.m`) or called from the user's image analysis pipeline. The main function is

```
1 confocal_generator(truth, conf, sample)
```

Here `truth.img` specifies the ground truth as a MATLAB matrix or the path to a TIFF file, and `truth.pix` is the physical size of a ground truth pixel. The ground truth can also be given as `truth.source`, the coordinates of the fluorophores (either a matlab matrix or the path to the file containing the coordinates). In this case, `truth.img` and `truth.pix` are generated using `make_ground_truth`. By default, `truth.img` is a gray level image, i.e. a matrix of doubles or floats, but it can also be provided as a binary image in order to test image segmentation algorithms. `truth.img` should be of high resolution, so that one confocal voxel contains a large number of `truth` pixels. The second argument `conf` should contain two fields. `conf.pix` is a 3×1 vector specifying for each dimension the size of one confocal voxel (in physical units, such as nanometers). `conf.psf` can be either an image, to be convolved with the ground truth image, or a 3×1 vector specifying for each dimension the spread of the PSF in physical units, in the case where the PSF is Gaussian. The last argument, `sample` contains the noise and signal information : either a sample (real) microscopy image `sample.pix` or the signal and noise characteristics `sample.sig` and `sample.noise`. Then, `confocal_generator` calls `generate_stacks` to create the synthetic image.

Image parameters estimation. Typically, `sample.pix` should be a real microscopy image, from which `get_img_params` extracts the signal and noise characteristics. `sample.noise` is a 3×1 vector containing the moments of the background voxel values distribution, and `sample.sig` is the mean value of signal pixels. To extract these values, `sample` is segmented using the procedure specified in `provide_image_mask`. This can only be performed if the segmentation is good enough to not significantly bias the background pixel distribution. In other cases, the user can also directly specify `sample.noise` and `sample.sig` instead of providing `sample.img`.

Stack generation. The routine `generate_stacks` first convolves `truth` with the PSF using the function `convolve_with_psf`. From this convolved high-resolution image, the voxel values of the simulated image `stack` are extracted by reading the intensity at the voxel position. Noise then needs to be added to reproduce the experimental signal-to-background ratio $S_n^* = S^*/B^*$, with $S^* = \text{sample.sig}$ (the mean signal value) and B^* is the first element of `sample.noise` (the mean background value).

Stack parameters estimation. The noiseless synthetic confocal stack is segmented using `get_img_params` to estimate the mean intensity of the signal voxels S and of the background voxels B .

Noise distribution generation. We generate a multiplicative noise array `px_noise` from `sample.noise`, the desired moments of the background voxels. According to `sample.noise`, we use either a Poisson, Gaussian, or a gamma distribution through the function `pixel_distribution`.

Noise multiplication. We then need to do an affine transformation on `stack` to reach the correct signal and background value. Defining \times as the element-wise matrix multiplication, we can write :

$$\text{stack} = a(\text{stack} + b) \times \text{px_noise}, \quad (1)$$

$$\text{with} \quad b = \frac{S - BS_n^*}{S_n^* - 1}, \quad a = \frac{1}{B + b}. \quad (2)$$

This approach allowed us to generate a background distribution highly similar to that measured on the sample, see Fig. 2.

```
1 get_img_params(image, options)
```

This routine determines `sig` and `noise` from `image`. It uses `provide_image_mask` to separate between signal and background voxels in `image`. The moments of the voxels values are then calculated to obtain `sig` and `noise`.

```
1 provide_image_mask(image, options)
```

This wrapper function calls the image segmentation code used by ConfocalGN. By default, it calls `segment_image`, a simple, generic image segmentation tool that uses Otsu thresholding [22] after Gaussian blurring. However, the user can edit `provide_image_mask` to use another procedure that provides the segmentation mask.

```
1 convolve_with_psf(image, psf)
```

This wrapper function returns the convolution of `image` with the PSF. By default, this function calls `cgn_convolve`, itself calling `fftconvn` if `psf` is an image, or `convolve_with_gaussian_psf` if `psf` is a 3×1 vector. However, the user can edit `convolve_with_psf` to use another convolution function.

```
1 convolve_with_gaussian_psf(image, psf)
```

This wrapper function uses the fast Gaussian convolution implementation `gauss3filter` [23], which can be further accelerated using MATLAB's parallel computing toolbox.

```
1 make_ground_truth(source, options)
```

This function generates a ground truth image in TIFF format from the coordinates of the fluorophores `source`, either a MATLAB matrix or the path to a coordinate file. This file will be read by `load_points` to load the coordinates. The ground truth image integrates the activity of all fluorophores, which can have a constant or stochastic emission (e.g. Poisson), as defined in the options `gtm_options`

```
1 load_points(source, options)
```

This wrapper function calls `import_from_text` to import coordinates from a file, supporting various file formats, including `.txt` and `.obj`. `load_points` can be edited to call a function specific to a given file format.

4. Applications

ConfocalGN can be used for a variety of applications. In particular, it can be used to quantitatively validate an image analysis pipeline, for example to verify that the geometry of an object extracted from a microscopy image does match the geometry of the actual object (the digital phantom in the case of a synthetic image). For this a set of ground truths needs to be generated, which can be done manually or using a simulation software. A variety of software is available to simulate image cytometry [11], and specialized software exists to simulate specific cellular objects, such as the open-source software Cytosim for the cytoskeleton [25]. From such simulations, a (microscope-independent) ground truth image (“digital phantom”) can be generated, which ConfocalGN can automatically post-process to match a particular imaging setup.

In a previous work, we used Cytosim to simulate the ring of microtubules found in some blood cells [24]. Cytosim exported coordinates into text files, which ConfocalGN directly read to generate synthetic images. Based on those, we were able to calibrate the image analysis program used to extract the contour length of the ring from live imaging, see Fig. 2. We illustrate the necessary steps in the following example.

5. Illustrative example

ConfocalGN is distributed with a test set. After running `cgn_startup` to add the required functions to MATLAB's path, `cgn_example` will execute:

```
1 truth.source='ground_truth.txt';
2 conf.pix=[150 150 600];
3 conf.psf='gauss_psf.tiff';
4 sample.img='sample_image.tiff';
5 [res, truth, sample]=confocal_generator
6 (truth, conf, sample);
```

In this example, the ground truth is specified as a list of fluorophore coordinates `ground_truth.txt`, from which a ground truth image will be generated using `make_ground_truth`. In the second step, the microscope properties (PSF and size of a voxel, in physical units) are specified directly, with `conf.psf` being the path to the PSF image. `sample_image.tiff` contains a real microscopy image, from which noise characteristics are to be extracted. This image, as well as the PSF file, and the ground truth coordinates, are included in the distribution. Finally, `confocal_generator` will return :

- `res` : the confocal simulation result. `res.stack` : simulated confocal dataset. `res.sig` : mean value of simulated signal voxels. `res.noise` : distribution of simulated background voxels values. `res.img` : image obtained by segmenting `res.stack`. `res.offset` : 3×1 vector describing the translation (in the ground truth image pixel size) between the simulated image and ground truth.
- `truth` : ground truth properties. `truth.points` : fluorophore coordinates of the ground truth. `truth.img` : ground truth image. `truth.pix` : size of ground truth image pixels, in physical units.
- `sample` : properties of the sample image. `sample.sig` : mean value of sample signal voxels. `sample.noise` : distribution of sample background voxels values.

`cgn_example` also calls `plot_simul_results` to plot the simulated stacks over the ground truth image.

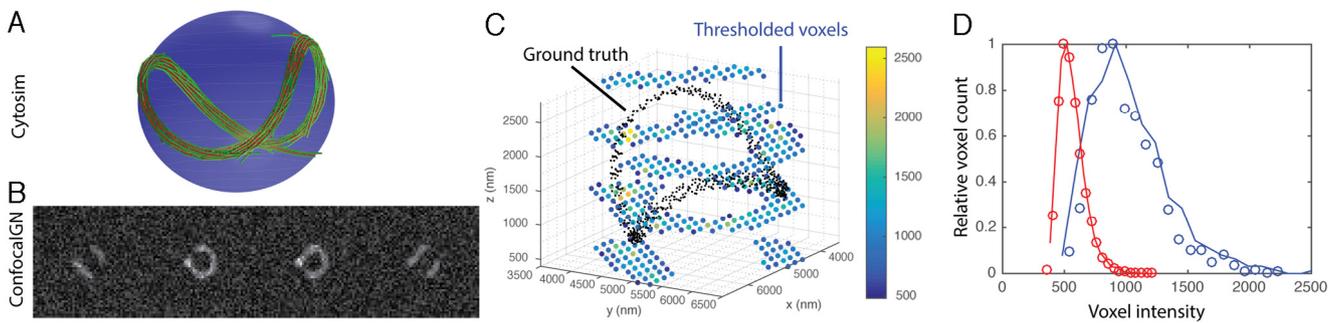


Fig. 2. A : Model of microtubules in a platelet simulated in Cytosim [24]. B : synthetic confocal stack corresponding to A, where microtubule coordinates were directly imported from Cytosim into ConfocalGN. Images show xy -planes at different z -positions within the stack. C : Black : coordinates of the fluorophores distributed on the microtubules from A. The signal voxels are shown in blue (low intensity) to yellow (high intensity). D : histogram of voxel values for the confocal microscopy image (dots) and for a simulated microscopy image (lines). The voxels have been categorized following the Otsu method, after Gaussian blurring, as background (red, 30 000 voxels for the sample image and 10 000 voxels for the simulated image) or signal (blue, 1000 voxels for the sample and 400 voxels for the simulated image). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Impact

Automated segmentation and analysis of microscopy images is paramount to modern quantitative biology, but the validation of image analysis pipeline remains an arduous task because the ground truth often cannot be known with certainty. This has encouraged the development of microscopy simulation software. However, current software are often highly specific (see [11] for a review), complex to use and difficult to adapt to the various needs of different users. ConfocalGN offers a simple alternative, while maintaining the level of realism that is required to quantitatively test image analysis pipelines.

Within our laboratory, ConfocalGN was used to verify the conclusions obtained from confocal experiments, in particular how the measured features of microscopic objects such as size and shape are altered by the limited resolution of the microscope (Fig. 2A). We share ConfocalGN in the hope that it will allow people in other laboratories to estimate the possible bias of their analysis pipeline, and eventually reach a higher confidence in their quantitative results. ConfocalGN can also be used for another purpose, to generate training sets for machine-learning based image analysis. It is a complete solution that can be used as distributed, and it is easily modified or integrated in a more elaborate computing pipeline.

7. Conclusions

ConfocalGN does not aim at being a bottom-up microscope simulator, but rather uses an effective description of the background and sensor noise, and a point spread function obtained from third party. Its input is consequently limited to (1) a ground truth image, (2) microscope characteristics (PSF and voxel size), and (3a) a sample image or (3b) the moments of the noise and signal distributions. Starting from measured PSF and noise characteristics it will produce simulated confocal image stacks with realistic characteristics. This simplicity allows it to be readily used without extensive training. ConfocalGN helped us to anticipate the quality of analysis as a function of the level of noise, and we found it irreplaceable to validate our image analysis procedures.

Acknowledgments

We thank Aastha Mathur for data and comments, Joran Deschamps for critical reading, Aliaksandr Halavatyi for testing, and the EMBL image analysis community for suggestions.

References

- [1] Tsien RY, Waggoner A. Fluorophores for confocal microscopy. In: *Handbook of biological confocal microscopy*. Springer; 1995. p. 267–79.
- [2] Pawley JB. Fundamental limits in confocal microscopy. In: *Handbook of biological confocal microscopy*. Springer; 2006. p. 20–42.
- [3] Hwang H, Haddad RA. Adaptive median filters: new algorithms and results. *IEEE Trans. image Process.* 1995;4(4):499–502.
- [4] McNally JG, Karpova T, Cooper J, Conchello JA. Three-dimensional imaging by deconvolution microscopy. *Methods* 1999;19(3):373–85.
- [5] Wallace W, Schaefer LH, Swedlow JR. A workingperson's guide to deconvolution in light microscopy. *Biotechniques* 2001;31(5):1076–97.
- [6] Meijering E. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Process. Mag.* 2012;29(5):140–5.
- [7] Tomer R, Denes AS, Tessmar-Raible K, Arendt D. Profiling by image registration reveals common origin of annelid mushroom bodies and vertebrate pallium. *Cell* 2010;142(5):800–9.
- [8] Sommer C, Straehle C, Köthe U, Hamprecht FA. Ilastik: interactive learning and segmentation toolkit. In: *2011 IEEE international symposium on biomedical imaging: From nano to macro*. IEEE; 2011. p. 230–3.
- [9] He L, Peng Z, Everding B, Wang X, Han CY, Weiss KL, Wee WG. A comparative study of deformable contour methods on medical image segmentation. *Image Vis. Comput.* 2008;26(2):141–63.
- [10] Pham DL, Xu C, Prince JL. Current methods in medical image segmentation 1. *Annu. Rev. Biomed. Eng.* 2000;2(1):315–37.
- [11] Ulman V, Svoboda D, Nykter M, Kozubek M, Ruusuvoori P. Virtual cell imaging: A review on simulation methods employed in image cytometry. *Cytometry Part A* 2016.
- [12] Kirshner H, Aguet F, Sage D, Unser M. 3-D PSF fitting for fluorescence microscopy: implementation and localization application. *J. Microsc.* 2013;249(1):13–25.
- [13] Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* 2005;151(2):182–95.
- [14] Krishnaswami V, Van Noorden CJ, Manders EM, Hoebe RA. Towards digital photon counting cameras for single-molecule optical nanoscopy. *Opt. Nanoscopy* 2014;3(1):1.
- [15] Sheppard CJ, Gan X, Gu M, Roy M. Signal-to-noise ratio in confocal microscopes. In: *Handbook of biological confocal microscopy*. Springer; 2006. p. 442–52.
- [16] Herberich G, Windoffer R, Leube RE, Aach T. Signal and noise modeling in confocal laser scanning fluorescence microscopy. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer; 2012. p. 381–8.
- [17] Sage D, Donati L, Soulez F, Fortun D, Schmit G, Seitz A, Guiet R, Vonesch C, Unser M. Deconvolutionlab2: An open-source software for deconvolution microscopy. *Methods* 2017.
- [18] Ursell T, Huang K. Resolution limits of optical microscopy and the mind. *Biomed. Comput. Rev.* 2011;7:27–9.
- [19] Lindén M, Čurić V, Boucharin A, Fange D, Elf J. Simulated single molecule microscopy with smeagol. *Bioinformatics* 2016;btw109.
- [20] Zhang B, Zerubia J, Olivo-Marin J-C. Gaussian approximations of fluorescence microscope point-spread function models. *Appl. Opt.* 2007;46(10):1819–29.
- [21] Nasse MJ, Woehl JC. Realistic modeling of the illumination point spread function in confocal scanning optical microscopy. *J. Opt. Soc. Am. A* 2010;27(2):295–302.

- [22] Otsu N. A threshold selection method from gray-level histograms. *Automatica* 1975;11(285–296):23–7.
- [23] Law MW, Chung AC. Efficient implementation for spherical flux computation and its application to vascular segmentation. *IEEE Trans. image process.* 2009;18(3):596–612.
- [24] Dmitrieff S, Alsina A, Mathur A, Nédélec FJ. Balance of microtubule stiffness and cortical tension determines the size of blood cells with marginal band across species. *Proc. Natl. Acad. Sci.* 2017;114(17):4418–23.
- [25] Nedelec F, Foethke D. Collective langevin dynamics of flexible cytoskeletal fibers. *New J. Phys.* 2007;9(11):427.