Excercises on deconvolution

Alessandra Griffa

September 1, 2010

Exercise 1: 3D widefield PSF

Start the PSFGenerator plugin in ImageJ. Generate a 3D PSF for a widefield microscope with following parameters:

- ni =.1.518
- NA = 1.4
- W040 = 0
- λ =500nm
- amplitude = 255
- background = 0
- SNR= 100
- x0 = y0 = 32
- z0 = 64
- $\Delta r = \Delta z$ = 50nm
- Nx = Ny = 64
- Nz =128

You can discard the image called *PSF for deconvolution*. Browse through the PSF using the VolumeViewer, the 3D Viewer and the Orthogonal Views tools of ImageJ. (hint: change the lookup table of the optic PSF into Fire to better appreciate its shape)

Change the numerical aperture (e.g. NA = 0.8) and/or the wavelength (e.g. λ = 400 or 600 nm). Generate the corresponding PSFs and describe the influence of these parameters.

Go back to the initial set of parameters. W040 controls the amount of spherical aberrations; set its value to 500 nm. Describe the effect on the PSF. What are possible causes of spherical aberrations?

____

____

____

____

Exercise 2: 2D inverse filtering

Load the images *FluorescentCells.tif* and *PSFDefocus.tif*. Launch the Deconvolution plugin.

2.1 Noise-free inverse filtering

We will first create an artificially blurred image using the defocusing blur kernel.
In the PSF section of the plugin, select the image *PSFDefocus.tif*. Make sure that Normalize PSF is selected. In the Algorithm section, select Convolution. Make sure that Add noise is deselected. Click the Run button. (note: to apply the plugin's functionalities to a specific image, you have to make sure that it was the last active image before clicking Run)

Try to deconvolve the widefield image you simulated using the inverse filtering method. In the Algorithm section, select Direct inversion and click Run. Comment the result.

____

____

____

____

2.2 Noisy inverse and Wiener filtering

We will create now an artificially blurred and noisy image (more realistic simulation).
Go back to Convolution in the Algorithm section. Activate the Add noise checkbox. Play around with both noise models at various noise levels (e.g. between 10, 30 and 60 dB), to get a feeling for this functionality. What are the principal sources of Gaussian and Poisson noise in microscopy images?

Choose the Gaussian model with a noise level such that you cannot visually distinguish the image from the one obtained at point 2.1. Apply the inverse filtering to restore the image and comment the results.

Try now the inverse filtering with regularization (Wiener filtering) on the noisy and blurred image. Select Regularized Direct Inversion in the Algorithm section and run the deconvolution with different values of the regularization parameter Lambda. How does the regularization parameter influence the result?

<u>Exercise 3: 2D iterative deconvolution</u>

Choose the Poisson model with a noise level of 30 dB. Select the Richardson-Lucy deconvolution algorithm, which is one of the simplest deconvolution algorithms for shot-noise limited imaging. Try out the algorithm with different numbers of iterations (no more than 100). What is the optimal number of iterations?

Suggestion: you can obtain a quantitative assessment of your prediction as follows (keep in mind that this is a synthetic experiment!): under Log, choose Normal; under SER, choose

Process SER and select *new reference* as the reference image. The plugin will indicate the error (in dB) of the current estimate with respect to the original image.
What do you observe? How can you explain it?

Perform 50 iterations of the Richardson-Lucy algorithm with TV regularization on the image corrupted by Poisson noise, with λ=0.0005. Comment the result both qualitatively and quantitatively (using the same error measurement as before). Compare the Richardson-Lucy deconvolution results at 50 iterations, with and without the TV regularization (hint: subtract the two images to highlight the differences).

Exercise 4: 3D deconvolution

Close all images and the Deconvolution plugin. Load the file *Microtubules.tif* into ImageJ: this is a widefield stack of a biological sample that we will deconvolve. In this exercise, we generate a synthetic PSF; another option would be to use a PSF obtained from an experimental measurement (typically by imaging sub-resolution fluorescent beads). Launch the PSFGenerator plugin and generate a PSF using the following parameters (which correspond to the acquisition settings):

- ni =.1.518
- NA = 1.4
- W040 = 0
- λ = 517 nm
- amplitude = 255
- background = 0

- SNR= 100
- x0 = y0 = 144
- z0 = 192
- Δr = 89 nm
- Δz = 290 nm
- Nx = 288
- Ny = 384
- Nz = 32
-

Perform a visual verification of the *Optic PSF* stack; after that you can discard it.

Note that the voxel size of the acquired image is coherent with the resolution of an objective with 1.4 NA, for the used wavelength.

Open the Deconvolution plugin. In the PSF section, select *PSF for deconvolution*. In the Algorithm section, select the Richardson-Lucy algorithm with 20 iterations. Compute maximum intensity projections of the original and deconvolved stacks. Compare them.

If there is some time left: try to perform a deconvolution with another algorithm (e.g. the Tikhonov-Miller algorithm) and comment the results. You may want to activate Test the algorithm on a small rectangular selection. You could also study the influence of using a PSF with wrong parameters (for example, change the NA value or introduce spherical aberrations).