Tutorial 7: Generating *in-silico* microscopy image with depth-variant PSF

Subhamoy Mahajan

10 Jul, 2021

This is a resource extensive Tutorial as many PSFs need to be generated. We will focus on the depth-variant Gibson and Lanni[1], (To be published soon).

The Gibson and Lanni[1] PSF models several aspects of a microscopy, and thereby has several parameters that can be controlled. The parameters include thickness of coverslip (`tg`) and immersion oil in non-design condition, and design conditions (`tg0`, `to`); refractive index of coverslip (`meug`) and immersion oil (`meu`) in non-design and design conditions (`meug0`, `meu0`); refractive index of the specimen (`meus`), distance of object focal plane from the coverslip (`ts0`). All thickness and distances are in nanometers, while the refractive index are dimensionless. Other than these parameters, all standard parameters similar to previous **Tutorials** are used: `NA` = 1.3, `dlmn` = 0.1, 0.1, 0.2, `Plmn` = 15, 15, 25, `fs` = 530, and `lam1,lam2` = 670, 518 are used. The design condition is fixed: `tg0` = 320, `t0` 300, `meug0` = 1.522, and `meu0` = 1.51. The tutorial explores how the images change when the following non-design parameters are changed:

1. `ts0`: $10, 20$.
2. `meu`: $1.6, 1.8$.
3. `meug`: $1.6, 1.8$.
4. `meus`: $1.35, 1.4$.
5. `tg`: $300, 340$.

Note: All refractive indices should be greater than the numerical aperture `NA`.

1 Generate PSF

All the required PSF can be generated using the bash script `gen_psf.sh` using the command,

```
Tut2$ bash gen_psf.sh
```

The PSF for design condition is calculated using the code block reproduced below,

```
gen_psf (){
    echo "###################################"
    siliscopy gen_psf --method GL1991 --paramfile foo.dat --calc all\
                      --output PSF_GL$1 --multiprocess
}

#Design condition
cp parameters.dat foo.dat
gen_psf
```

First, the parameters file (contains parameters for design condition) is copied to `foo.dat` and the bash function `gen_psf` is called without any arguments (`$1` is empty). The Gibson and Lanni[1] PSF is calculated using `--method GL1991`.

To generate PSF for non-design condition, first the design parameter is changed using `sed` and then the function `gen_psf` is called. For example, to generate PSF at different `ts0` the following code block is used,

```
sed "s/ts0\s*=.*/ts0 = 10/g" parameters.dat  > foo.dat
gen_psf
sed "s/ts0\s*=.*/ts0 = 20/g" parameters.dat > foo.dat
gen_psf
```

Note that the argument to `gen_psf` is still empty. This is because PSF output is always generated with the `ts0` value in its name for depth-variant PSF (`psf_type = 1`). For other non-design changes, an argument is passed to `gen_psf` (`$1`) which is appended to the PSF output name (`PSF_GL$1`). For example, to generate PSF in non-deign condition with `meu=1.6`, the argument passed to `gen_psf` is `_meu1.6`. The code block is reproduced below.

```
sed "s/meu\s*=.*/meu = 1.6/g" parameters.dat > foo.dat
gen_psf "_meu1.6"
```

Other non-design PSF are generated in the same way.

2 Generate *in-silico* monochrome image intensities

The monochrome image intensities are generated using the bash script `gen_mono.sh` using the command,

```
Tut2$ bash  gen_mono.sh
```

The script is almost identical to `gen_psf.sh`, where the function is changed from `gen_psf` to `gen_mono`.

```
gen_mono (){
    echo "###################################"
    siliscopy gen_mono --file dp100.gro --paramfile foo.dat --psf PSF_GL$1\
                       --output GL$1_img100 --method slice
}
```

Notice the positions file is `dp100.gro`, the PSF file read is `PSF_GL$1` (identical to the generated PSF), and output header file is `img100_GL$1` (100 corresponds to the file `dp100.gro`). The script creates 22 `GL_*img100_*_fs530.dat` files.

3 Generate colored *in-silico* microscopy images

The monochrome image intensities are generated using the bash script `gen_img.sh` using the command,

```
Tut2$ bash  gen_img.sh
```

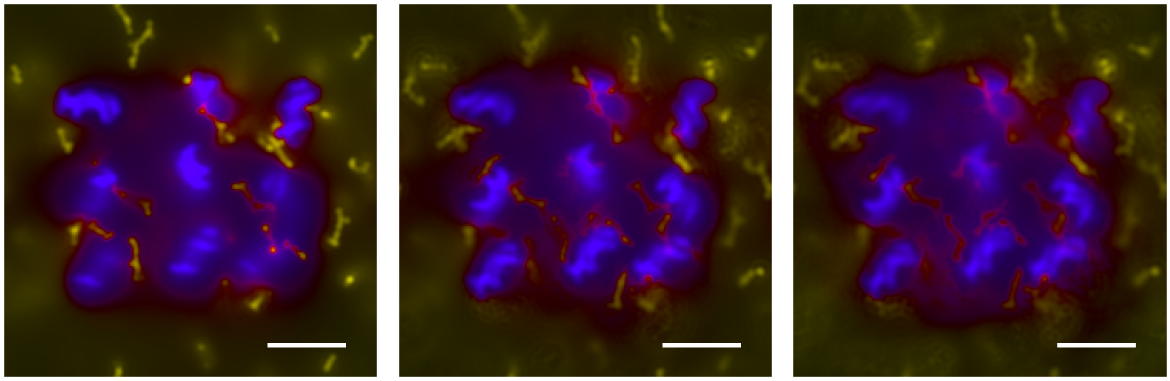
The script is almost identical to `gen_psf.sh`, where the function is changed from `gen_psf` to `gen_mono`.

```
gen_img (){
    echo "###################################"
    siliscopy plot --file GL$1_img --paramfile foo.dat --method color \
                --timestep 100 --calc specific --output GL$1_img --type jpeg
}
```

A timestep of 100 is used to read image intensity files (`--file GL$1_img`) and output colored JPEG images (`--output GL$1_img100`, `--method color`, `--type jpeg`).

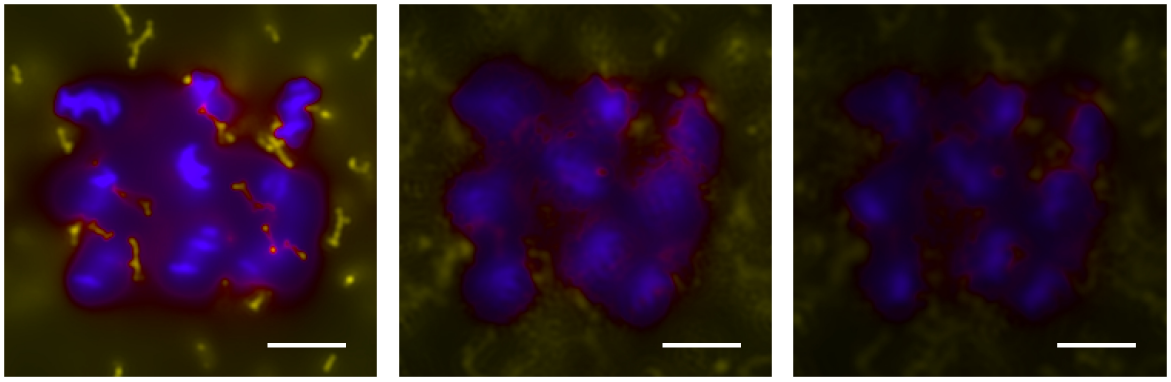
3.1 Effect of tsO

Design condtition `tsO` = 0 (left), `tsO` = 10 (middle), `tsO` = 20 (right).



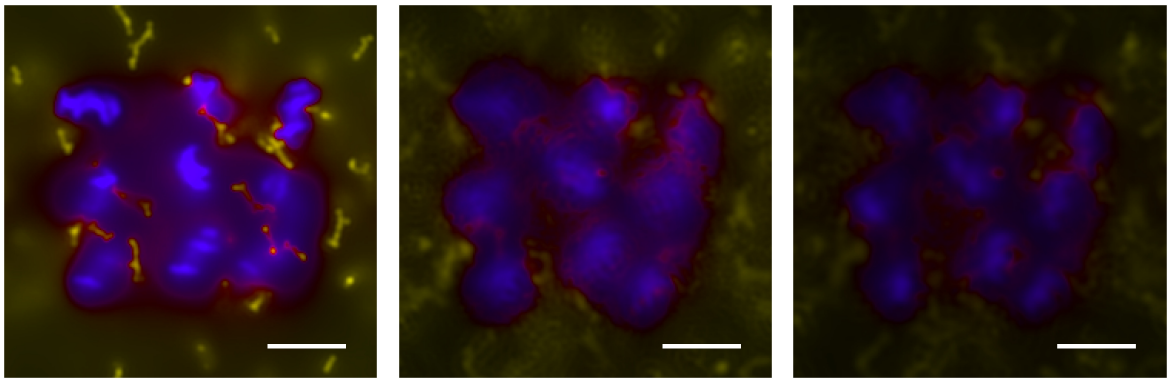
3.2 Effect of meu

Design condtition `meu` = 1.51 (left), `meu` = 1.6 (middle), `meu` = 1.8 (right).



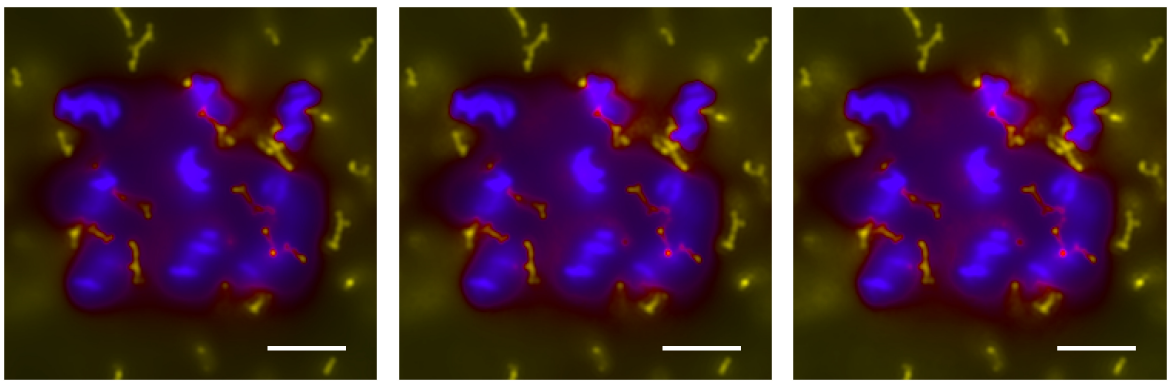
3.3 Effect of meug

Design condtition `meug` = 1.522 (left), `meug` = 1.6 (middle), `meug` = 1.8 (right).



3.4 Effect of meus

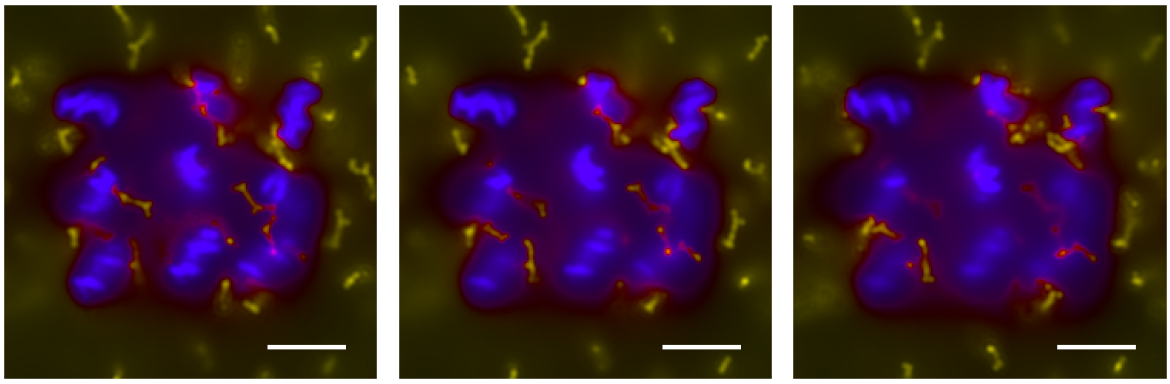
Design condtition `meus` = 1.33 (left), `meus` = 1.45 (middle), `meus` = 1.55 (right).



The changes are subtle: Notice the change in top right corner, just above the DNA (in violet-magenta).

3.5 Effect of tg

`tg` = 220 (middle), design condtition `tg` = 320 (left), `meug` = 420 (right).



Reference

[1] Gibson and Lanni PSF: S. F. Gibson, and F. Lanni, 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.