cellsino Documentation

Release 0.5.0

Paul Müller

Apr 12, 2021

CONTENTS

Cellsino is a Python library for generating sinograms (phase and fluorescence) of cell phantoms for testing tomographic reconstruction algorithms. This is the documentation of cellsino version 0.5.0.

CHAPTER

ONE

INTRODUCTION

1.1 What is cellsino?

Cellsino is a tool for generating in-silico fluorescence and refractive index sinograms of predefined cell phantoms. The computation of the sinogram data is done with semi-analytical approaches (e.g. Rytov approximation for spheres [[MSG+18]]) and is thus comparatively fast and accurate.

CHAPTER

TWO

GETTING STARTED

2.1 Installation

cellsino is written in pure Python and supports Python version 3.6 and later. To install, run

```
pip install cellsino
```

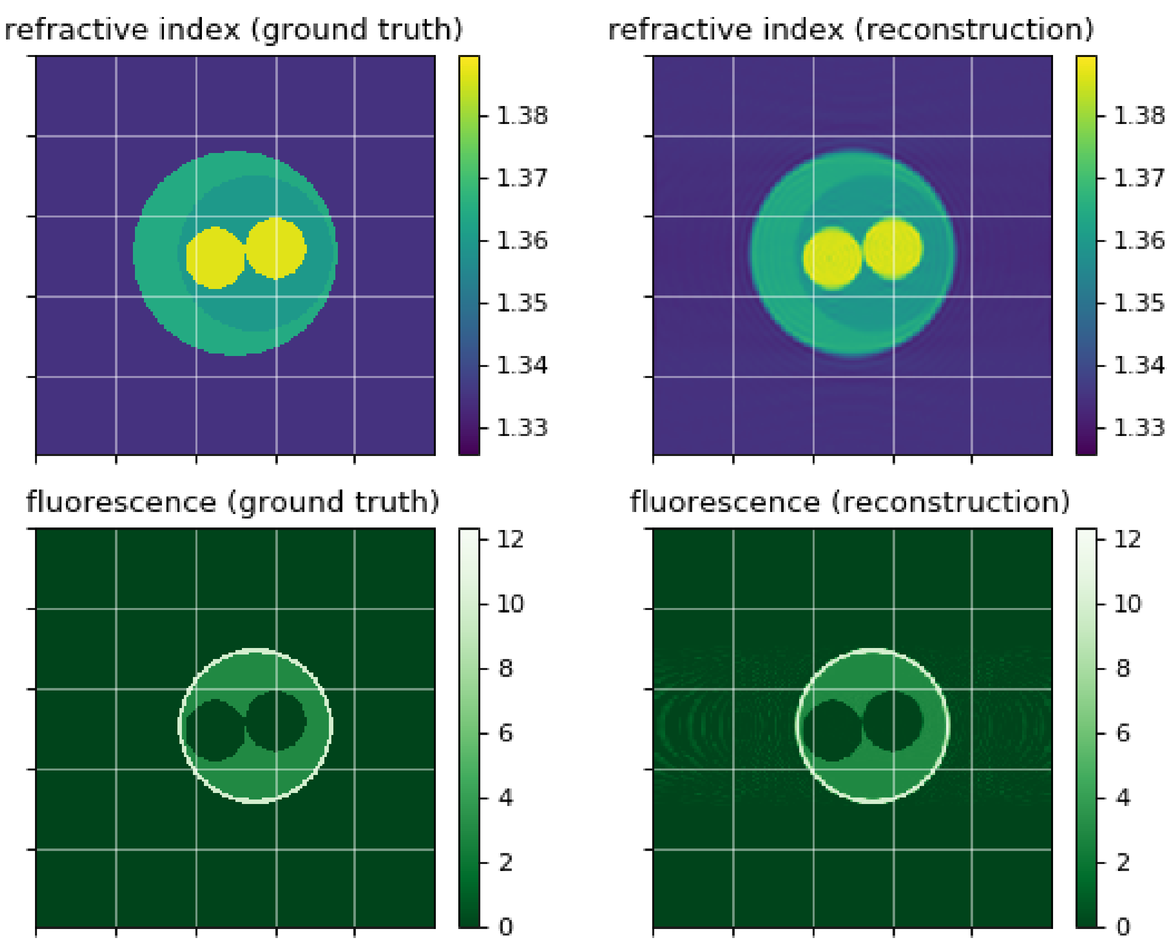
CHAPTER

THREE

CODE EXAMPLES

3.1 Reconstruction of a simple cell phantom

This example uses ODTbrain and radontea for the reconstruction of the refractive index and the fluorescence sinogram of the simple cell phantom. The reconstruction is compared to the ground truth of the cell phantom.



simple_cell.py

```python
import cellsino
import matplotlib.pylab as plt
import numpy as np
import odtbrain as odt
import radontea as rt


# number of sinogram angles
num_ang = 160
# sinogram acquisition angles
angles = np.linspace(0, 2*np.pi, num_ang, endpoint=False)
# detector grid size
grid_size = (250, 250)
# vacuum wavelength [m]
wavelength = 550e-9
# pixel size [m]
pixel_size = 0.08e-6
# refractive index of the surrounding medium
medium_index = 1.335

# initialize cell phantom
phantom = cellsino.phantoms.SimpleCell()

# initialize sinogram with geometric parameters
sino = cellsino.Sinogram(phantom=phantom,
                         wavelength=wavelength,
                         pixel_size=pixel_size,
                         grid_size=grid_size)

# compute sinogram (field according to Rytov approximation and fluorescence)
sino_field, sino_fluor = sino.compute(angles=angles, propagator="rytov")

# reconstruction of refractive index
sino_rytov = odt.sinogram_as_rytov(sino_field)
potential = odt.backpropagate_3d(uSin=sino_rytov,
                                 angles=angles,
                                 res=wavelength/pixel_size,
                                 nm=medium_index)
ri = odt.odt_to_ri(f=potential,
                   res=wavelength/pixel_size,
                   nm=medium_index)

# reconstruction of fluorescence
fl = rt.backproject_3d(sinogram=sino_fluor,
                       angles=angles)

# reference for comparison
rimod, flmod = phantom.draw(grid_size=ri.shape,
                            pixel_size=pixel_size)

# plotting
idx = 150

plt.figure(figsize=(7, 5.5))

plotkwri = {"vmax": ri.real.max(),
            "vmin": ri.real.min(),
```

(continues on next page)

(continued from previous page)

```
             "interpolation": "none",
             "cmap": "viridis",
             }

plotkwfl = {"vmax": fl.max(),
            "vmin": 0,
            "interpolation": "none",
            "cmap": "Greens_r",
            }

ax1 = plt.subplot(221, title="refractive index (ground truth)")
mapper = ax1.imshow(rimod[idx].real, **plotkwri)
plt.colorbar(mappable=mapper, ax=ax1)

ax2 = plt.subplot(222, title="refractive index (reconstruction)")
mapper = ax2.imshow(ri[idx].real, **plotkwri)
plt.colorbar(mappable=mapper, ax=ax2)

ax3 = plt.subplot(223, title="fluorescence (ground truth)")
mapper = ax3.imshow(flmod[idx], **plotkwfl)
plt.colorbar(mappable=mapper, ax=ax3)

ax4 = plt.subplot(224, title="fluorescence (reconstruction)")
mapper = ax4.imshow(fl[idx], **plotkwfl)
plt.colorbar(mappable=mapper, ax=ax4)

for ax in [ax1, ax2, ax3, ax4]:
    ax.grid(color="w", alpha=.5)
    ax.set_xticks(np.arange(0, grid_size[0], 50))
    ax.set_yticks(np.arange(0, grid_size[0], 50))
    ax.set_xticklabels([])
    ax.set_yticklabels([])

plt.tight_layout()

plt.show()
```

CHAPTER

FOUR

CODE REFERENCE

TODO

CHAPTER

FIVE

CHANGELOG

List of changes in-between cellsino releases.

5.1 version 0.5.0

- BREAKING CHANGE: switch from longdouble to double precision in propagator
- build: migrate from travisCI to GH Actions
- build: setup.py test is deprecated
- docs: refurbish docs

5.2 version 0.4.0

- BREAKING CHANGE: revert previous breaking change, because this way the reconstruction with ODT-brain/radontea remains straight-forward
- enh: support str in *mode* when computing sinograms

5.3 version 0.3.0

- feat: new options for sinogram generation
  - measurement duration (or specific times per angle)
  - specify simulated imaging modalities
  - simulate photobleaching and background fluorescence
- BREAKING CHANGE: changed definition of angles in sinogram generation

5.4 version 0.2.1

- ref: migrate to flimage

5.5 version 0.2.0

- feat: add option to roll rotational axis (in-plane) during sinogram generation
- feat: add capability to track progress of sinogram generation
- feat: add possibility to add (random) displacements for each sinogram slice
- fix: medium index not stored in exported sinogram file

5.6 version 0.1.0

- add basic example to docs
- drawing refractive index and fluorescence phantoms
- sinogram generation
- basic functionalities (sphere element):
  - simple cell phantom
  - propagators: Rytov and projection
  - fluorescence projector

5.7 version 0.0.1

- initial setup

CHAPTER

SIX

BILBLIOGRAPHY

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[MSG+18] P. Müller, M. Schürmann, S. Girardo, G. Cojoc, and Guck J. Accurate evaluation of size and refractive index for spherical objects in quantitative phase imaging. *Optics Express*, 26(8):10729–10743, 2018. doi:10.1364/OE.26.010729.