
cellsino Documentation

Release 0.5.0

Paul Müller

Apr 12, 2021

CONTENTS

1	Introduction	3
1.1	What is cellsino?	3
2	Getting started	5
2.1	Installation	5
3	Code examples	7
3.1	Reconstruction of a simple cell phantom	7
4	Code reference	11
5	Changelog	13
5.1	version 0.5.0	13
5.2	version 0.4.0	13
5.3	version 0.3.0	13
5.4	version 0.2.1	14
5.5	version 0.2.0	14
5.6	version 0.1.0	14
5.7	version 0.0.1	14
6	Bibliography	15
7	Indices and tables	17
	Bibliography	19

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.

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.

GETTING STARTED

2.1 Installation

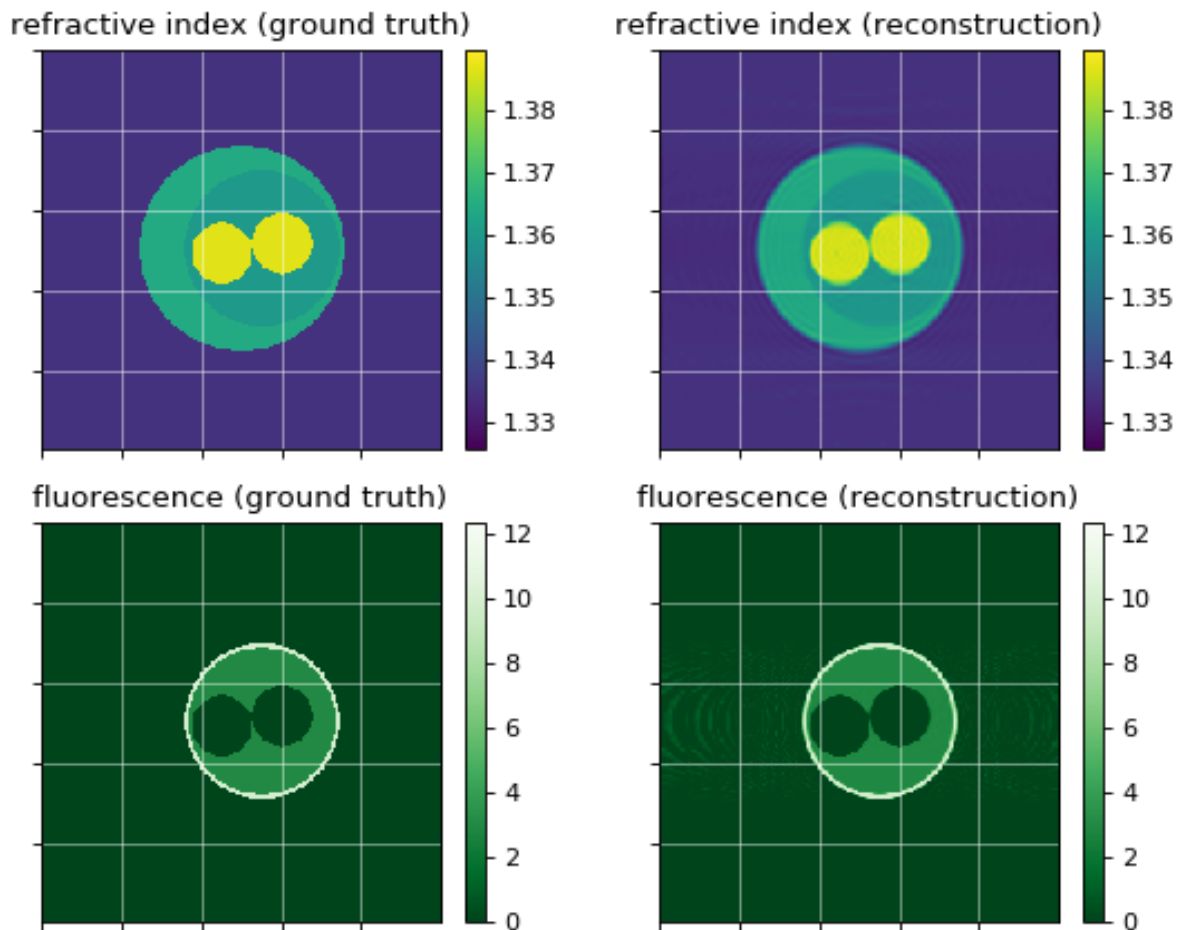
cellsino is written in pure Python and supports Python version 3.6 and later. To install, run

```
pip install cellsino
```


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`

```
1 import cellsino
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import odtbrain as odt
5 import radon Tea as rt
6
7
8 # number of sinogram angles
9 num_ang = 160
10 # sinogram acquisition angles
11 angles = np.linspace(0, 2*np.pi, num_ang, endpoint=False)
12 # detector grid size
13 grid_size = (250, 250)
14 # vacuum wavelength [m]
15 wavelength = 550e-9
16 # pixel size [m]
17 pixel_size = 0.08e-6
18 # refractive index of the surrounding medium
19 medium_index = 1.335
20
21 # initialize cell phantom
22 phantom = cellsino.phantoms.SimpleCell()
23
24 # initialize sinogram with geometric parameters
25 sino = cellsino.Sinogram(phantom=phantom,
26                          wavelength=wavelength,
27                          pixel_size=pixel_size,
28                          grid_size=grid_size)
29
30 # compute sinogram (field according to Rytov approximation and fluorescence)
31 sino_field, sino_fluor = sino.compute(angles=angles, propagator="rytov")
32
33 # reconstruction of refractive index
34 sino_rytov = odt.sinogram_as_rytov(sino_field)
35 potential = odt.backpropagate_3d(uSin=sino_rytov,
36                                 angles=angles,
37                                 res=wavelength/pixel_size,
38                                 nm=medium_index)
39 ri = odt.odt_to_ri(f=potential,
40                  res=wavelength/pixel_size,
41                  nm=medium_index)
42
43 # reconstruction of fluorescence
44 fl = rt.backproject_3d(sinogram=sino_fluor,
45                      angles=angles)
46
47 # reference for comparison
48 rimod, flmod = phantom.draw(grid_size=ri.shape,
49                             pixel_size=pixel_size)
50
51 # plotting
52 idx = 150
53
54 plt.figure(figsize=(7, 5.5))
55
56 plotkwri = {"vmax": ri.real.max(),
57            "vmin": ri.real.min(),
```

(continues on next page)

(continued from previous page)

```
58     "interpolation": "none",
59     "cmap": "viridis",
60     }
61
62 plotkwfl = {"vmax": fl.max(),
63            "vmin": 0,
64            "interpolation": "none",
65            "cmap": "Greens_r",
66            }
67
68 ax1 = plt.subplot(221, title="refractive index (ground truth)")
69 mapper = ax1.imshow(rimod[idx].real, **plotkwri)
70 plt.colorbar(mappable=mapper, ax=ax1)
71
72 ax2 = plt.subplot(222, title="refractive index (reconstruction)")
73 mapper = ax2.imshow(ri[idx].real, **plotkwri)
74 plt.colorbar(mappable=mapper, ax=ax2)
75
76 ax3 = plt.subplot(223, title="fluorescence (ground truth)")
77 mapper = ax3.imshow(flmod[idx], **plotkwfl)
78 plt.colorbar(mappable=mapper, ax=ax3)
79
80 ax4 = plt.subplot(224, title="fluorescence (reconstruction)")
81 mapper = ax4.imshow(fl[idx], **plotkwfl)
82 plt.colorbar(mappable=mapper, ax=ax4)
83
84 for ax in [ax1, ax2, ax3, ax4]:
85     ax.grid(color="w", alpha=.5)
86     ax.set_xticks(np.arange(0, grid_size[0], 50))
87     ax.set_yticks(np.arange(0, grid_size[0], 50))
88     ax.set_xticklabels([])
89     ax.set_yticklabels([])
90
91 plt.tight_layout()
92
93 plt.show()
```


CODE REFERENCE

TODO

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/radon tea 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

BILBLIOGRAPHY

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.