
TomoTwin

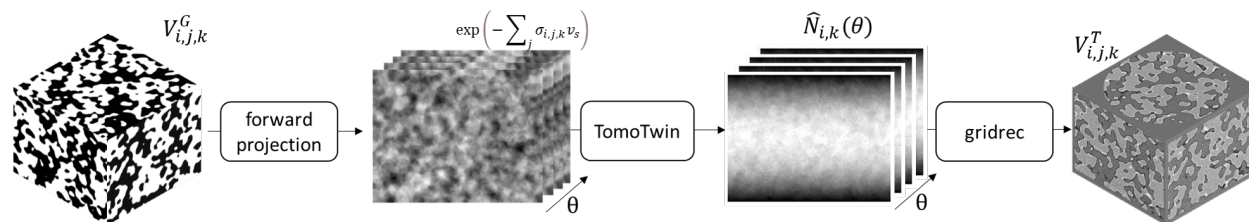
Release 1.10

Aniket Tekawade

Apr 21, 2021

CONTENTS

1	Contribute	3
2	Content	5
	Python Module Index	9
	Index	11



TomoTwin is a simple digital twin for synchrotron tomography that provides an API to generate realistic grayscale tomography datasets from ground-truth (labeled) volumes as training data for denoising and segmentation models. The grayscale data mimics imaging artifacts such as noise and in-line phase contrast with a dependence on the instrument parameters such as detector configuration and beam energy.

CONTRIBUTE

- Documentation: <https://github.com/aniketkt/TomoTwin/tree/master/doc>
- Issue Tracker: <https://github.com/aniketkt/TomoTwin/docs/issues>
- Source Code: <https://github.com/aniketkt/TomoTwin/>

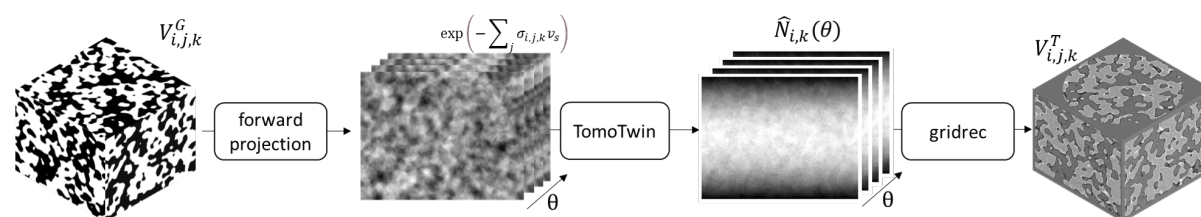
CONTENT

2.1 About

2.1.1 Description

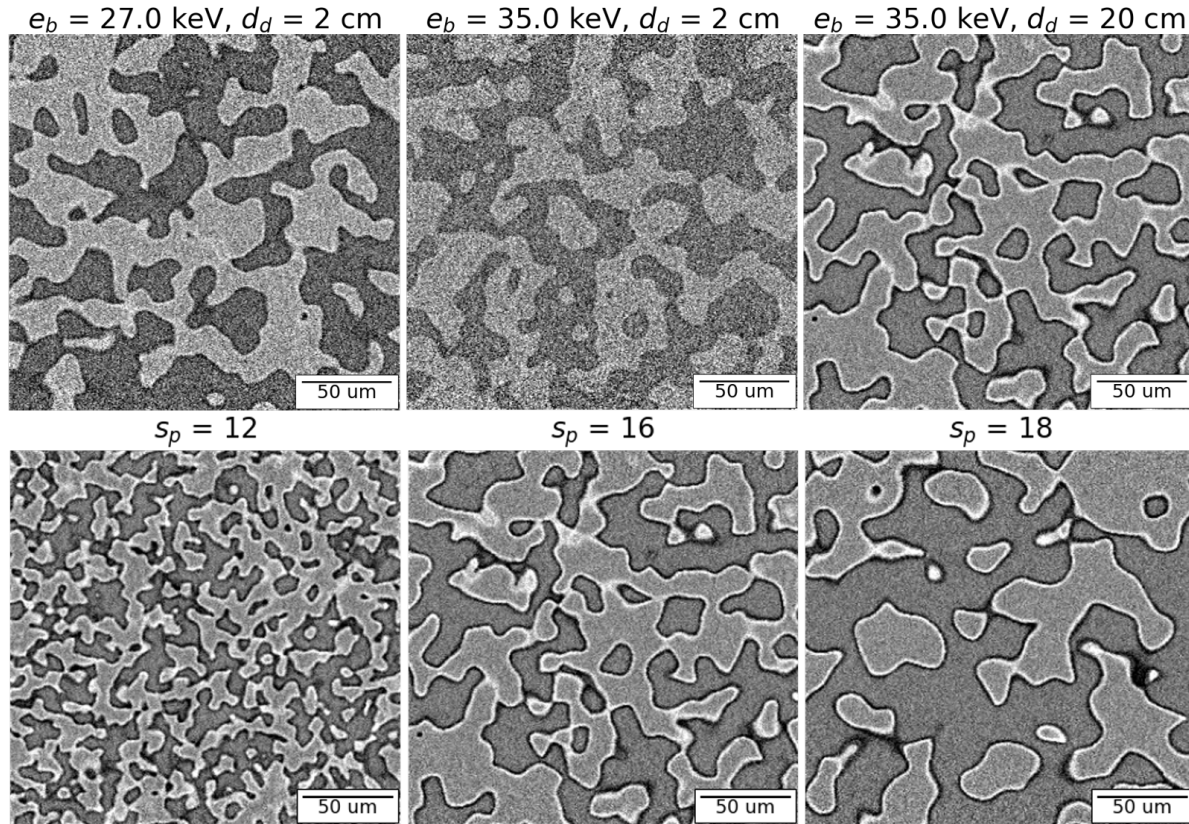
ray module

Two classes are implemented: The ‘Phantom’ class is instantiated by passing a ground-truth labeled volume and the material name and density corresponding to each label. For example, in a porous volume mimicing rock, the voxels belonging to the matrix could be labeled ‘1’ and those belonging to the pores could be labeled ‘0’. The material names and corresponding density (g/cc) would be passed as a dictionary {“air” : 0.00122, “silica” : 2.7} in that order. The ‘Material’ class is instantiated for each material by reading the attenuation data from model_data/materials folder. You may use XOP to generate new materials by following the README file in model_data/. Further details are in our paper.



gt_generators module

You may pass your own labeled volumes to the Phantom class. This file provides functions to parametrically create some labeled volumes that mimic porous materials, inclusions, etc. You could create a volume of up to 256 labels as this will be handled as 8-bit data.



jupyter notebooks

A few example images are shown above. s_p is a size parameter for the phantom while the acquisition parameters (beam energy and detector distance) affect the contrast in the grayscale images. For tutorials on how to use TomoTwin, please check out the jupyter notebooks that are provided with the installation. You can run them by clicking on the binder button at the top of the README.

Tell me more

Our paper is under review for IEEE-ICIP 2021 and will be up on arXiv very soon!

2.2 Install

2.2.1 In a new tomopy environment

Clone the package, then run make.sh.

```
git clone https://github.com/aniketkt/TomoTwin.git
cd TomoTwin
sh install.sh
```

2.2.2 In existing tomopy environment

Simply install the package using pip

```
$ pip install git+https://github.com/aniketkt/TomoTwin.git@master#egg=tomo_twin
```

in a prepared virtualenv which has tomopy pre-installed.

2.3 Usage

2.3.1 jupyter notebooks

The best way to use TomoTwin is through jupyter notebooks. Check out the notebooks on binder to get started.

2.4 API reference

TomoTwin modules:

2.4.1 ray

2.4.2 gt_generators

2.4.3 recon

2.5 Credits

2.5.1 Citations

We kindly request that you cite the following article [] if you use TomoTwin for your project.

PYTHON MODULE INDEX

t

tomo_twin, 7

INDEX

M

module
 tomo_twin, [7](#)

T

tomo_twin
 module, [7](#)