DICOM-Numpy Documentation

Release 0.1.1

J. David Giese

Jan 29, 2021

Contents

This python module provides a set of utilities for extracting data contained in DICOM files into Numpy ndarrays. It is a higher-level library that builds on the excellent lower-level pydicom library.

The library is quite small at the moment, however, if you have a DICOM-related utility function that you think would be appropriate to include, create a Github Issue!

CHAPTER 1

Dependencies

- Python 2.7 or Python 3.5+
- Numpy
- PyDicom

CHAPTER 2

Installation

```
pip install dicom_numpy
```

CHAPTER 3

Source Code

The source code is hosted on Github.

CHAPTER 4

Combine DICOM Slices

The DICOM standard stores MR, CT, and PET scans as a series of images saved in a separate files. A common task is to combine all of the images that make up a single 3D image into a single scan.

The function that performs this task is *combine_slices*. Since this library builds on pydicom, *combine_slices* takes an list of pydicom datasets.

4.1 Example

```python
import dicom
import dicom_numpy

def extract_voxel_data(list_of_dicom_files):
    datasets = [dicom.read_file(f) for f in list_of_dicom_files]
    try:
        voxel_ndarray, ijk_to_xyz = dicom_numpy.combine_slices(datasets)
    except dicom_numpy.DicomImportException as e:
        # invalid DICOM data
        raise
    return voxel_ndarray
```

4.2 Details

`dicom_numpy.`**`combine_slices`**`(`*datasets*, *rescale=None*`)`

Given a list of pydicom datasets for an image series, stitch them together into a three-dimensional numpy array. Also calculate a 4x4 affine transformation matrix that converts the ijk-pixel-indices into the xyz-coordinates in the DICOM patient's coordinate system.

Returns a two-tuple containing the 3D-ndarray and the affine matrix.

If *rescale* is set to *None* (the default), then the image array dtype will be preserved, unless any of the DICOM images contain either the Rescale Slope or the Rescale Intercept attributes. If either of these attributes are present, they will be applied to each slice individually.

If *rescale* is *True* the voxels will be cast to *float32*, if set to *False*, the original dtype will be preserved even if DICOM rescaling information is present.

The returned array has the column-major byte-order.

Datasets produced by reading DICOMDIR files are ignored.

This function requires that the datasets:

- Be in same series (have the same Series Instance UID, Modality, and SOP Class UID).
- The binary storage of each slice must be the same (have the same Bits Allocated and Pixel Representation).
- The image slice must approximately form a grid. This means there can not be any missing internal slices (missing slices on the ends of the dataset are not detected).
- It also means that each slice must have the same Rows, Columns, Samples Per Pixel, Pixel Spacing, and Image Orientation (Patient) attribute values.
- The direction cosines derived from the Image Orientation (Patient) attribute must, within 1e-4, have a magnitude of 1. The cosines must also be approximately perpendicular (their dot-product must be within 1e-4 of 0). Warnings are displayed if any of these approximations are below 1e-8, however, since we have seen real datasets with values up to 1e-4, we let them pass.
- The Image Position (Patient) values must approximately form a line.

If any of these conditions are not met, a *dicom_numpy.DicomImportException* is raised.

`dicom_numpy.`**`sort_by_slice_position`**(*slice_datasets*)

Given a list of pydicom Datasets, return the datasets sorted in the image orientation direction.

This does not require *pixel_array* to be present, and so may be used to associate instance Datasets with the voxels returned from ‘combine_slices.

CHAPTER 5

Change Log

5.1 Version 0.5.0

- Export *sort_by_slice_position*

5.2 Version 0.4.0

- Ignore DICOMDIR files
- Fix bug that was triggered when using *from dicom_numpy import **
- Make *combine_slices* work with a single slice
- Add support for “channeled slices” (e.g., RGB slices)
- Allow HighBit and BitsStored DICOM attributes to be non-uniform
- Drop support for Python 3.4; test Python 3.7
- Require the SamplesPerPixel DICOM attribute to be invariant among the slices

5.3 Version 0.3.0

- Reverted slice ordering change from v0.2.0, since the DICOM standard defines the Z-axis direction to be increasing in the direction of the head.
- Added support for both PyDicom 0.X and 1.X

5.4 Version 0.2.0

- Changed the behavior of *combine_slices* to stack slices from head (slice 0) to foot (slice -1). Note that this is the reverse of the behavior in v0.1.*.

5.5 Version 0.1.5

- Added the *rescale* option to *combine_slices*
- Made *combine_slices*'s returned ndarray use column-major ordering

CHAPTER 6

Contributing

6.1 Process

Contributions are welcome. Please create a Github issue describing the change you would like to make so that you can discuss your approach with the maintainers. Assuming the maintainers like your approach, then create a pull request.

6.2 Tests

Most new functionality will require unit tests.

Run all of the tests for each supported python version using:

```
tox
```

Run all of the tests for the currently active python version using:

```
pytest
```

6.3 Other Contributors

Additional contributions made by:

- Jonathan Daniel

Thank you!

CHAPTER 7

About Innolitics

Innolitics is a team of talented software developers with medical and engineering backgrounds. We help companies produce top quality medical imaging and workflow applications. If you work with DICOM frequently, our DICOM Standard Browser may be useful to you.

If you could use help with DICOM, let us know! We offer training sessions and can provide advice or development services.

CHAPTER 8

Licenses

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

Index