Numpy2Vtk Documentation

Release 0.1.0

Stefan Lau

September 20, 2016

Contents

The Numpy2Vtk library is an easy way to render numpy data in the VTK visualization framework. It is designed as a thin wrapper around VTK that returns ready-to-render VTK objects. This way a lot of boilerplate code that usually needs to be written when using VTK to render numpy data can be avoided.

If you want to report bugs and/or submit feature requests, please do so at Numpy2Vtk's github page.

Contents:

CHAPTER 1

API Documentation

1.1 Vtk Actors

1.2 Vtk PolyData

Vtk PolyData usually wraps multiple data types (e.g. Vertices, Edges) into a single data type. In VTK, they are used to define how an actor is rendered. We return the data-only representation for an object here and define the visual options in the actors module.

numpy2vtk.data.**vertices**(*points*, *z_index=0*)

Returns the VTK-representation of a number of vertices that are defined by the points array.

Parameters

- **points** (*numpy.ndarray<float> or vtk.vtkPoints*) – The points that the mesh consist of. If it's a numpy array it should be of dimensions (n,2) or (n,3)
- **z_index** (*float*) – The value the z-value of 2d-points is filled with (only applicable for (n,2) input arrays)

Returns **vertices_data** – VTK polydata representation of the vertices

Return type vtk.vtkPolyData

numpy2vtk.data.**line**(*points*, *z_index=0*, *closed=False*)

Returns the VTK-representation of a line that is build from the points in the numpy array.

Parameters

- **points** (*numpy.ndarray<float> or vtk.vtkPoints*) – The points that the line consist of. If it's a numpy array it should be of dimensions (n,2) or (n,3)
- **z_index** (*float*) – The value the z-value of 2d-points is filled with (only applicable for (n,2) input arrays)
- **closed** (*bool*) – Whether the last point of the line should be connected with the first one

Returns **line_data** – VTK polydata representation of the line

Return type vtk.vtkPolyData

numpy2vtk.data.**mesh**(*points*, *polys*, *z_index=0*)

Returns the VTK-representation of a mesh that is build by creating the patches specified by points and polys. Points are the considered points and polys consists of an array of patches (which consist of indices into the points array).

Parameters

- **points** (*numpy.ndarray<float> or vtk.vtkPoints*) – The points that the mesh consist of. If it's a numpy array it should be of dimensions (n,2) or (n,3)
- **polys** (*numpy.ndarray<int>*) – Array of patches, should be of shape nxm for n patches with m points per patch
- **z_index** (*float*) – The value the z-value of 2d-points is filled with (only applicable for (n,2) input arrays)

Returns **poly_data** – VTK polydata representation of the mesh

Return type vtk.vtkPolyData

1.3 Raw Vtk Data

The data.raw module is the lowest representation of data in VTK. These are primarily used to update the data of the actors.

numpy2vtk.data.raw.**points**(*coordinates*, *z_index=0*)

Returns the raw VTK-representation of the points in the passed numpy array

Parameters

- **coordinates** (*numpy.ndarray<float>*) – numpy.ndarray of shape (n,2) or (n,3) that contains the points
- **z_index** (*float*) – The value the z-value of 2d-points is filled with (only applicable for (n,2) input arrays)

Returns **vtk_points** – VTK representation of the points

Return type vtk.vtkPoints

numpy2vtk.data.raw.**vertices**(*indices*)

Maps a numpy ndarray of shape (n,) to an vtkCellArray of vertex indices

Parameters **indices** (*numpy.ndarray<int>*) – A numpy.ndarray of shape (n,) of indices that defines the n vertices

Returns **vtk_vertices** – VTK representation of the vertices

Return type vtk.vtkCellArray

numpy2vtk.data.raw.**edges**(*indices*)

Maps a numpy ndarray to an vtkCellArray of vtkLines

Parameters **indices** (*numpy.ndarray<int>*) – A numpy.ndarray of shape (n,2) of indices that define n edges

Returns **vtk_lines** – VTK representation of the edges

Return type vtk.vtkCellArray

numpy2vtk.data.raw.**polygons**(*indices*)

Maps a numpy ndarray to an vtkCellArray of vtkPolygons

Parameters **indices** (*numpy.ndarray<int>*) – A numpy.ndarray of shape (n,m) of indices that define n polygons with m points each

Returns **vtk_polygons** – VTK representation of the polygons

Return type vtk.vtkCellArray

CHAPTER 2

License

Numpy2Vtk is copyright © 2015 Stefan Lau and can be used under the terms of the LGPL 3.0 License, which is displayed below.

```
                   GNU LESSER GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


  This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

  0. Additional Definitions.

  As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

  "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

  An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

  A "Combined Work" is a work produced by combining or linking an
Application with the Library.  The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

  The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

  The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.
```

```
  1. Exception to Section 3 of the GNU GPL.

  You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

  2. Conveying Modified Versions.

  If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
   whatever part of its purpose remains meaningful, or

   b) under the GNU GPL, with none of the additional permissions of
   this License applicable to that copy.

  3. Object Code Incorporating Material from Library Header Files.

  The object code form of an Application may incorporate material from
a header file that is part of the Library.  You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

   a) Give prominent notice with each copy of the object code that the
   Library is used in it and that the Library and its use are
   covered by this License.

   b) Accompany the object code with a copy of the GNU GPL and this license
   document.

  4. Combined Works.

  You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

   a) Give prominent notice with each copy of the Combined Work that
   the Library is used in it and that the Library and its use are
   covered by this License.

   b) Accompany the Combined Work with a copy of the GNU GPL and this license
   document.

   c) For a Combined Work that displays copyright notices during
   execution, include the copyright notice for the Library among
   these notices, as well as a reference directing the user to the
   copies of the GNU GPL and this license document.

   d) Do one of the following:
```

```
      0) Convey the Minimal Corresponding Source under the terms of this
      License, and the Corresponding Application Code in a form
      suitable for, and under terms that permit, the user to
      recombine or relink the Application with a modified version of
      the Linked Version to produce a modified Combined Work, in the
      manner specified by section 6 of the GNU GPL for conveying
      Corresponding Source.

      1) Use a suitable shared library mechanism for linking with the
      Library.  A suitable mechanism is one that (a) uses at run time
      a copy of the Library already present on the user's computer
      system, and (b) will operate properly with a modified version
      of the Library that is interface-compatible with the Linked
      Version.

   e) Provide Installation Information, but only if you would otherwise
   be required to provide such information under section 6 of the
   GNU GPL, and only to the extent that such information is
   necessary to install and execute a modified version of the
   Combined Work produced by recombining or relinking the
   Application with a modified version of the Linked Version. (If
   you use option 4d0, the Installation Information must accompany
   the Minimal Corresponding Source and Corresponding Application
   Code. If you use option 4d1, you must provide the Installation
   Information in the manner specified by section 6 of the GNU GPL
   for conveying Corresponding Source.)

  5. Combined Libraries.

  You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

   a) Accompany the combined library with a copy of the same work based
   on the Library, uncombined with any other library facilities,
   conveyed under the terms of this License.

   b) Give prominent notice with the combined library that part of it
   is a work based on the Library, and explaining where to find the
   accompanying uncombined form of the same work.

  6. Revised Versions of the GNU Lesser General Public License.

  The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

  Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
```

```
General Public License ever published by the Free Software Foundation.

  If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

n

E

L

M

N

P

V