The Py3DViewer project:
a Python library for fast prototyping in geometry processing

G. Cherchi, L. Pitzalis, G. L. Frongia & R. Scateni
Py3DViewer

Python library for fast prototyping in Geometry Processing

Manipulation and Visualization of:

- Surface meshes
- Volume meshes
- Skeletons
- Others (asap)
Jupyter environment

Jupyter Lab
• Interactive Python IDE in a browser

You can easily install jupyter: jupyter.org/install.html
Why Py3DViewer?

- CGAL
- OpenMesh
- libigl
- PyMesh
- cinolib
- OpenVolumeMesh
- ParaView
C++ vs Python

- Verbose but fast
- Efficient loops
- Not ideal for prototyping
- Troublesome library installation

- Compact but slow, “interpreted pseudo code”
- Efficient tensorial operations
- Ideal for prototyping
- Easy package manager (pip)
Python... but fast?

NumPy

Numba

SciPy
The library structure

Data structure module
- meshes
- skeleton
- others...

Visualization module
- viewer
- GUI
- drawable

Utility module
- metrics
- load/save op
- color maps
- others...

Algorithm module
- work in progress
Data structures module

Index-based storage

TriMesh
- face2face
- add_vertices(...)
- remove_vertices(...)
- add_faces(...)
- remove_faces(...)
- show()

QuadMesh
- face2face
- add_vertices(...)
- remove_vertices(...)
- add_faces(...)
- remove_faces(...)
- show()

AbstractMesh
- vertices
- faces
- bounding_box
- vtx2vtx
- vtx2face

TetMesh
- tetrahedra
- tet2tet
- vtx2tet
- face2tet
- add_vertices(...)
- remove_vertices(...)
- add_tets(...)
- remove_tets(...)
- show()

HexMesh
- hexahedra
- hex2hex
- vtx2hex
- face2hex
- add_vertices(...)
- remove_vertices(...)
- add_hexas(...)
- remove_hexas(...)
- show()
Visualization module

**Viewer**
- GUI
- reactive
- drawable geometries
- update()

**GUI**
- geometry
- widgets

**Drawable**
- geometry
- as_drawable()
- colors
- materials
Responsive viewer

```python
while True:
    rot = Rotation.from_euler('y', .5, degrees=True)
    mesh.vertices = rot.apply(mesh.vertices)
    sleep(1/20)
```
Reactive dataflow

GUI

observes

Viewer

observes

Drawable

observes

Geometry

observes

Geometric attributes

Observable arrays (Numpy)
The GUI

![Py3DViewer Project Image]

The Py3DViewer project...
Algorithms module

Work in progress module

At least one for the main GP problems like:

• Smoothing
• Hole filling
• Triangulation
• Decimation
• and so on...
Active development

We developed and improved a lot from the paper submission...

<table>
<thead>
<tr>
<th>Model</th>
<th>#Vertices</th>
<th>#Simplices</th>
<th>Time [Liv19]</th>
<th>Time ours (paper)</th>
<th>Time ours (now)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tri-mesh</td>
<td>14.2k</td>
<td>25.5k</td>
<td>0.25s</td>
<td>1.05s</td>
<td>0.35s</td>
</tr>
<tr>
<td>quad-mesh</td>
<td>9.9k</td>
<td>9.6k</td>
<td>0.11s</td>
<td>0.49s</td>
<td>0.15s</td>
</tr>
<tr>
<td>tet-mesh</td>
<td>4.6k</td>
<td>15.4k</td>
<td>0.52s</td>
<td>0.58s</td>
<td>0.32s</td>
</tr>
<tr>
<td>hex-mesh</td>
<td>18.9k</td>
<td>15.2k</td>
<td>0.61s</td>
<td>0.61s</td>
<td>0.58s</td>
</tr>
</tbody>
</table>

Loading times (with adjacencies)
Let’s code!

tinyurl.com/py3dviewer
How to install and use Py3DViewer

• GitHub repo:
  github.com/cg3hci/py3DViewer

• You can easily install Py3DViewer by using pip:
  pip install git+https://github.com/cg3hci/py3DViewer

• Documentation and examples available on GitHub or:
  py3dviewer.readthedocs.io/en/latest

• Interactive example on binder:
  tinyurl.com/py3dviewer
Thanks for your attention

github.com/cg3hci/py3DViewer