
VIPER – Volume Invariant Position-based Elastic Rods



Abstract

We extend the formulation of position-based rods to include elastic volumetric deformations. We achieve this by introducing an additional degree of freedom per vertex – isotropic scale (and its velocity). Including scale enriches the space of possible deformations, allowing the simulation of volumetric effects, such as a reduction in cross-sectional area when a rod is stretched. We rigorously derive the continuous formulation of its elastic energy potentials, and hence its associated position-based dynamics (PBD) updates to realize this model, enabling the simulation of up to 26000 DOFs at 140 Hz in our GPU implementation. We further show how rods can provide a compact alternative to tetrahedral meshes for the representation of complex muscle deformations, as well as providing a convenient representation for collision detection. This is achieved by modeling a muscle as a bundle of rods, for which we also introduce a technique to automatically convert a muscle surface mesh into a rods-bundle. Finally, we show how rods and/or bundles can be skinned to a surface mesh to drive its deformation, resulting in an alternative to cages for real-time volumetric deformation.

License

Except as noted below this project is released under the Apache License, Version 2. See the [LICENSE](#) file for the full text.

IMPORTANT NOTICE: The assets `data/mesh.bin` and `data/texture.bin` are proprietary and are *NOT* released under an open-source license. You are *NOT* permitted to redistribute them or to use them for any purpose other than running this demo.

How to Build

Recommended configuration

- NVIDIA GTX1080Ti or better
- Ubuntu 18.04
- CUDA 10.x

Initialize submodules

Either user 'git clone --recursive' or, after cloning, execute

```
1 git submodule init
2 git submodule update
```

Install dependencies

```
1 sudo apt-get install cmake
2 sudo apt-get install xorg-dev
3 sudo apt-get install libboost-all-dev
4 sudo apt-get install libglew-dev
5 sudo apt-get install libcgald-dev
6 sudo apt-get install libtbb-dev
```

Build & run

Ensure /usr/local/cuda exists!

```
1 export LD_LIBRARY_PATH=/usr/local/cuda/lib64
2 mkdir build
3 cd build
4 cmake ..
5 make
6 ./demo
```

Build & run with Docker

If you do not want to set up an Ubuntu 18.04 environment to build, you may use the provided docker-file to build and run in a container from a Linux host running X11.

Start by installing [nvidia-docker](#) following the instructions [here](#).

You may then build and run the demo with

```
1 sudo ./build_run_docker.sh
```

How to cite this work

```
1 @conference{angles2019vipер,  
2   title={VIPER: Volume Invariant Position-based Elastic Rods},  
3   author={Baptiste Angles, Daniel Rebain, Miles Macklin, Brian Wyvill,  
4         Loic Barthe, John Lewis, Javier von der Pahlen, Shahram Izadi,  
5         Julien Valentin, Sofien Bouaziz, Andrea Tagliasacchi},  
6   booktitle={Proceedings of Symposium on Computer Animation},  
7   year={2019}}
```