Character creation:
From Blender to Humanoid Path Planner

Mylène Campana (mcampana@laas.fr)
LAAS-CNRS, Toulouse, France

Contents

1 Introduction

This paper presents the technical framework for the creation and the motion planning of an arbitrary character. The involved softwares for these tasks are:

- Blender, version ≥ 2.70
- Humanoid Path Planner[1] (abbreviated as 'HPP')

2 Creation of the articulated character (URDF)

HPP has a ROS[2] parser to read URDF[3] files. Characters have to be defined following this formality. Initially, the character can be represented in two ways:

1. The character contains separated bodies, such as a skeleton, a robot or an object.
2. The character is composed of one mesh which is deformed, e.g. a humanoid or an animal.

In the case 2., it will be necessary to cut the mesh in bodies to come back to the case 1., because the URDF model handle a kinematic chain made of bodies and joints.
Many free 3D meshes can be found on the following websites:

- http://tf3dm.com
- http://archive3d.net

Note that only some formats can be imported in Blender:

- Collada (.dae)
- Wavefront (.obj)
- 3D Studio (.3ds)
- Stl (.stl)

Otherwise, it will be necessary to use another software to convert the mesh.

2.1 Mesh cutting and body origins

This section presents how a mesh can be cut with Blender into mesh bodies, and how body origins can be located for the future URDF file.
Important note: the frames of the bodies will be oriented along the global frame, this method does not handle frame re-orientation.
These are the steps, keyboard shortcuts in Blender are indicated:

[1] http://humanoid-path-planner.github.io/hpp-doc/index.html
[2] Robot Operating System http://www.ros.org
[3] Unified Robot Description Format http://wiki.ros.org/urdf

1. Import the mesh in Blender.
2. If necessary, re-orient the mesh (e.g. to follow the $\mathbf{e}_x$ direction).
3. If necessary, deform the mesh so that the frame orientation fits the joint directions.
4. Cut the bodies in edit mode.
   (a) Use `z` for the Wireframe view (otherwise, back part of the mesh will not be cut).
   (b) Cut with the straight knife `K` + `Z`.
   (c) Select the cut vertices with `Space`.
   (d) Hide the vertices `H`.
   (e) Select the body to separe from the rest of the mesh with `L`.
   (f) Separate with `P`.
   (g) Un-hide the vertices `Alt` + `H`.
5. Relocate the origin of the body where the joint will be (note the joint will *act* on the current body).
   (a) Place the 3D cursor at the selection `Shift` + `S`.
   (b) Be in object mode with the relevant body selected.
   (c) Define the new origin `Ctrl` + `Alt` +`Shift` + `C`.
6. One again, note that the joint will be actuated on the given origin, around axes parallel to the global axes.
7. When having their origin located at $(0,0,0)$ in Blender, export each mesh-body with a Stl or Collada (.dae) format in one folder.

2.2 URDF creation

When the robot geometry is expressed in bodies, its kinematic chain (with joints) will be described using an URDF file. We refer to the ROS documentation to build the URDF. The only specificity regarding cut meshes is the definition of the origins:

- Set all the body origins at $(0,0,0)$.
- Use the origins of the Blender mesh-bodies (set previously) for the origins of the joints. Origins must be measured according to their parent frame (i.e. the parent origin must be in $(0,0,0)$ when reading the child origin values). To move the body, use `Shift` + `S`.

The cut mesh-bodies can simply be inserted in the URDF as links, specifying the path of the mesh folder and the mesh filename in the *mesh* xml tag.
Note that HPP requires also a SRDF file that specifies the disable collision pairs (e.g. two following bodies that are in contact).

3 Motion planning with HPP

We refer to the HPP documentation for the basics[4]. The trajectory is planned with and can be exported with automatic tools. Following the initial shape of the character, the animation process differs for each time-frame:

1. If the character contains separated bodies, each body transformation is written in a text file.
2. If the character is composed of one deformable mesh, each joint value is written in a text file.

4 Smooth animation with Blender

In Blender with the full model (separated bodies or one mesh), use an automatic tool to import the motions for each frame.

Before this, for the deformable mesh, it is necessary to create an armature and to use it a a skeleton to deform the mesh:

1. The armature base is located at the origin of the character root.
2. The bone connections are located at the joint/body origins.
3. The names of the bones following the bodies are not important.
4. For each connection with at least one joint, it is necessary to create one intermediate bone that will fix the bone frame (because by default, bone frames are oriented with the $\mathbf{e}_y$ direction along the bone).
    - Set the intermediate bone name as the joint *root* name in the URDF (*root* because a suffix '_ry' can be added in the URDF).
    - Set The *Roll* value to 0.
    - Set the same position values for the Head and the Tail of the bone, except for the Y value which should be +0.001 or -0.001 depending on the wanted orientation for the joint (resp. $\mathbf{e}_y$ of the joint following the global $\mathbf{e}_y$ or the global $-\mathbf{e}_y$).
5. Link the armature to the mesh using `Crtl + P`.
6. Test the joints with the pose mode.
7. Additional bones can be added to the armature to correctly follow the mesh.

[4] http://humanoid-path-planner.github.io/hpp-doc/tutorials.html

5 Examples - existing files - functions

Examples of cut meshes can be found in the package hpp-rbprm-corba of Mylene Campana's repository on Github[5]. Refer to the most recent branch.

- In **data/blender** folder, Blender files including the full mesh, cut meshes and armatures.
- In **data/urdf** folder, URDF files for HPP.
- In **data/srdf** folder, SRDF files for HPP.
- In **data/meshes** folder, Cut meshes to be used by the URDF files.

Scripts for HPP and Blender can also be found:

- In **scripts/test** folder, HPP scripts to launch the motion planning scenarios. Functions are:
  - `writePathSamples` for characters containing separated bodies.
  - `pathJointConfigsToFile` for characters with one deformable mesh.
- In **scripts/blender** folder, the unique script `fullAnimationSkinning.py` is a library to export motions from HPP files to Blender. Main functions are:
  - `importYamlMotion` for characters containing separated bodies.
  - `importArmatureMotion` for characters with one deformable mesh.

[5] https://github.com/mylene-campana