

Analyse de données de capture de mouvement

Synthèse d'images et animation
2011-2012

Le but de ce TP est d'analyser des données issues de la capture de mouvement. Ces données seront par la suite utilisées pour animer un personnage.

1 Description de l'interface

La lecture des données de capture de mouvement est basée sur la librairie `qglviewer` que vous avez utilisée l'an dernier pour le lancer de rayons.

Les données de capture de mouvement sont stockées dans l'attribut `_root` de type `Skeleton` de la classe `Viewer`.

La classe `Skeleton` permet de représenter un squelette d'animation. Un objet de type `Skeleton` a :

- un nom correspondant au nom de l'articulation
- un offset correspondant à sa translation par rapport à son père
- une transformation par rapport à son père qui varie au cours du temps. Elle est représentée par une courbe d'animation par degré de liberté. Une courbe d'animation est représentée par le nom du degré de liberté et par les valeurs au cours du temps de ce degré de liberté
- les valeurs courantes des degrés de liberté
- des fils eux-mêmes de type `Skeleton`.

Les données de capture de mouvements sont stockées sous forme de fichier `.bvh`. On peut charger un fichier `.bvh` à partir de l'interface en pressant la touche `L`. La touche `Entrée` permet de lancer ou interrompre l'animation. Dès qu'un fichier `.bvh` est chargé, le graphe de scène s'affiche dans la console de sortie.

Nous souhaitons analyser une séance de capture de mouvements pour déterminer les degrés de liberté d'un corps humain. En effet, les données lues donnent 3 degrés de liberté en rotation à chaque articulation. En pratique, en raison de contraintes biomécaniques, certaines articulations ont moins de degrés de liberté.

2 Représentation des rotations

Comme vu en cours, les angles d'Euler ne sont pas forcément la représentation la plus appropriée des rotations dans l'espace. Une rotation peut toujours être définie comme une rotation autour d'un axe 3D. Complétez les méthodes nécessaires pour faire fonctionner `Skeleton::eulerToAxisAngle(rx, ry, rz, rorder, vaa)`.

Cette méthode permet de passer d'une représentation en angles d'Euler à une représentation axe-angle. Dans notre cas, on choisit une représentation compacte où la rotation représentée par le `qglviewer::Vec vaa` a comme axe de rotation `vaa/vaa.norm()` et comme angle de rotation `vaa.norm()`.

Bien sûr, on n'utilisera pas les méthodes de `qglviewer` mises à disposition pour ce genre de calcul. On pourra cependant les utiliser pour vérifier les calculs effectués. La page wikipedia [Rotation representation](#) est une bonne source pour comprendre les différentes représentations des rotations.

3 Détermination des degrés de liberté

L'avantage de la représentation axe-angle est qu'elle facilite la détection des articulations à moins de 2 degrés de liberté. En effet, une rotation avec 1 seul degré de liberté est une rotation dont l'axe de rotation est constant au cours du temps. Complétez la méthode `Skeleton::nbDofs()` qui détermine si une articulation a 0, 1 ou plus d'1 degré de liberté.