

TP programmation Maya

lionel.reveret@inria.fr

2010-11

Analyse et ré-utilisation de données de capture de mouvement

Le but de ce TP est de mettre en place une lecture et une analyse de données issues de la capture de mouvement. A la fin, on s'intéressera au problème du mélange de données de capture de mouvement, c'est à dire, à partir de plusieurs séquences, comment en créer de nouvelles.

1. Lecture du format de données BVH

Le format BVH permet de stocker des séquences de capture de mouvement. Il comporte deux parties : la description topologique du squelette d'animation et le mouvement de la séquences. Pour la première, le mot clé dans le fichier est HIERARCHY, pour la seconde, MOTION. Pour chaque articulation, une liste de canaux est spécifiée et correspondent aux angles Euler de rotations dans le repère local. La partie mouvement fournit la valeur au cours du temps dans un tableau où chaque ligne correspond à un échantillon temporel. Pour chaque ligne, les canaux se présente dans l'ordre de leur déclaration dans la partie topologie.

1.1. Réaliser un parser pour le format BVH en script MEL

Un fichier *walk.bvh* et le résultat attendu *walk.mb* sont fournis. Tester ensuite sur le fichier *run.bvh*

1.2. Réaliser un parser pour le format BVH en API C++

Les 3 fichiers *motion1.bvh*, *motion2.bvh* et *motion3.bvh* sont beaucoup plus volumineux. Une implémentation en C++ permettra de faire un chargement plus rapide.

1.3. Analyse de performances

Evaluer le gain de temps pour chaque fichier bvh entre le parser MEL et le parser C++.

2. Analyse des données de rotation

Tout mouvement entre deux corps rigides liés par une articulation parfaite possède 3 degrés de libertés en rotation. Dans la pratique, en raison de contraintes biomécaniques, certaines articulations ont moins de degrés de libertés. On étudie ici ces réductions. Pour cette partie, le choix d'une solution mixte entre langage MEL et C++ est libre.

2.1. Représentation des rotations

Les angles d'Euler ne sont pas la seule représentation des rotations dans l'espace. Une rotation dans l'espace est toujours définie par un axe 3D autour duquel se fait la rotation et une valeur d'angle pour cette rotation (cf fonction `glRotate` en OpenGL). En passant par une représentation quaternion, retrouver pour chaque courbe d'animation les valeurs des rotations à chaque frame sous la forme axe-angle.

2.1. Visualisation des degrés de liberté

A partir du fichier *walk.bvh*, établir la liste des articulations.

Identifier les articulations avec un seul degré de liberté en utilisant la représentation axe-angle. Une articulation avec un seul degré de liberté aura un axe constant au cours du temps.

3. Mélange d'animation

Le but ici est d'implémenter un programme de mélange d'animation. A partir de coefficients d'interpolation entre deux ou plusieurs séquences, il faudra générer une séquence résultante. Pour cette partie, comme la précédente, le choix d'une solution mixte entre langage MEL et C++ est libre.

3.1. Interface utilisateur

Les points d'entrées du modèle sont les noms de fichier et les coefficients d'interpolation. Réaliser un interface utilisateur permettant de les spécifier.

3.2. Mélange direct

Les données à interpoler sont les valeurs des courbes d'animation, en translation pour le pelvis et en rotation pour toutes les articulations. Mettre en place une interpolation linéaire directe entre deux séquences. Commenter les résultats.

3.3. Pré-alignement avant mélange

De l'expérience précédente, on constate qu'une interpolation directe sur toute la séquence entraîne des problèmes. Identifier le découpage adéquat et les critères pour le calculer. Réaliser ainsi un nouveau schéma de mélange sans les artefacts précédents.

3.4. Interpolation dans SO3

L'interpolation linéaire des angles d'Euler peut poser problème. Réaliser une interpolation de type sphérique (SLERP) en passant à une représentation des rotations sous forme de quaternions.

3.5. Généralisation au mélange de plus de deux mouvements.

Pour l'instant on s'est limité au mélange de deux mouvements. Étendre l'interface et le calcul si on veut mélanger plus de deux séquences.

Remarque : le modèle *olaf+pose.mb*

permet de visualiser le résultat d'une animation par données de rotation sur un personnage complet. Pour lui appliquer une animation, il faut copier les clefs d'animation du squelette animé, sélectionner le nœud père du squelette cible associé au personnage puis copier les clefs.