Animating human model in OpenGL using data from Vicon system

Gediminas Mazrimas
Aalborg University Copenhagen
Computer Vision and Graphics
g.mazrimas@gmail.com

Algirdas Beinaravicius
Aalborg University Copenhagen
Computer Vision and Graphics
algirdux@gmail.com

Abstract

This paper explains how to animate 3D human model in OpenGL, avoiding most common problems, that occurs while dealing with various human body rigid parts transformations. The animation is associated with real person movements, while using data from Vicon motion capture system. Animating is made in lowest programming C++ OpenGL level.

Keywords: *Human animation, Vicon system, OpenGL, C++, Linear Blend Skinning*

1. Introduction

Our animation focuses on the most common and partly simple human body animation technique, that uses joints to animate human model. The joint structure, given their position and orientation, can be thought as being human body skeleton. The skin shape is associated to the joints, where as it's a 3D polygon mesh and it's the only thing that is displayed for the end-user. Due to very fast computation speeds, this technique is the most popular in animation production. On the other hand, using simple shape blending technique to deal with complex human body rigid parts transformations, there are various skin deformation problems. Typical ones are collapsing elbow, candy-wrapper joint when the arm turns 180 degrees, intersection between two adjacent bones (links) around a joint. Also such a technique don't consider many very complicated and detail human body deformations, for example dealing with muscles (stretch or bulge).

1.1. Previous works

What we've read and what was written there. References. [Linear blend skinning, for deformation problems, data formats]

1.2. Overview

What is represented in further sections?

2. Linear blend skinning

Theory of linear blend skinning, that helps avoiding skin deformations.

3. Vicon motion capture system

What's Vicon motion capture system. How it works, what from it consists?

3.1. Setting up system

How we set up this system to be able to work with it. How we prepare working space, how cameras are calibrated and similar.

3.2. Capturing data

How we capture our motion data (with costume). How we assign (label) captured data to a Vicon model. How Vicon model looks like (available joints, their connections, connections to costume sensors).

4. Motion data

What Motion data formats are we using? How do they look like?

4.1. Description of Vicon C3D format

Short information about C3D data.

4.2. Description of Biovision BVH format

The BVH format is an updated version of BioVisions BVA data format, with the addition of a hierarchical data structure representing the bones of the skeleton. The BVH file is an ASCII file and consists of two parts.

First section (HIERARCHY) is for storing hierarchy and initial pose of the skeleton, basically joint-to-joint connections and offsets. While as second section (MOTION) describes the channel motion data for each frame, that is describes the movement of individual joints.

The HIERARCHY section starts with ROOT joint and contains OFFSET, CHANNELS and children JOINTS data. Here OFFSET is followed by X, Y and Z coordinates, to determine current joint position. CHANNELS information contains the existence of corresponding XYZ data streams (i.e. rotation and translation) in the MOTION section which follows. Finally it includes other JOINTS in hierarchical order with OFFSET and CHANNEL and possible END SITE attribute, to determine body segment end.

In the MOTION section, each row contains data values for all CHANNELS which were specified in the HIERARCHY. The listing order of CHANNELS values in each row in is assumed to match their listed order from the HIERARCHY section (top down).

There are only few drawbacks of this format, for example - the lack of a full definition of the basis pose (it has only translational offsets of children segments from their parent, no rotational offset is defined) and also it lacks explicit information for how to draw the segments, but that has no bearing on the definition of the motion.

4.3. Processing captured data

Why do we need to convert C3D data to BVH? What's good and bad about them? (C3d would be faster, but it's binary, so not a human readable, also BVH gives us rotations and have good structure, that is much easier to implement in our C++ program). How do we do it using Motion builder?

While C3D gives us our marker coordinates from motion capture system, by labeling these markers at first in Vicon system and then importing them in Motion builder, in BVH then we get rotations and translations not for those markers, but for human body model joints.

[Here goes how we interpret BVH data]

5. Human body mesh model

What do we use for our animation?

5.1. Mesh model preparations in Maya

What Mesh model in Maya do we use? How do we prepare it, that it would be suitable for out program? How we cut mesh to different body parts, export to .obj format files.

5.2. Mesh model file format used in animation

Why do we use .obj file format? How does it look like?

6. Animating human body

Explain how we load human model, exported from Maya to .obj file. How we create natural primary human pose, assign meshes to joints and so on.

6.1. Loading human body model in OpenGL

How we load human body to OpenGL. How we import mesh from obj files. Joint creation. Joint connection with meshes.

6.2. Linear blend skinning relations

How we adapted linear blend skinning to our animations. Explain how our method using meshes assigned with joint rotations is similar to linear blend skinning. How we automated all the things.

7. Conclusion

Conclusion

References

[1] M. G. Alex Mohr. Building efficient, accurate character skins from example. *SIGGRAPH 2003*.

[2] S. M. K. F.S. HILL, JR. *Computer Graphics Using OpenGL.* Prentice Hall, 2007.