

Animation from MoCap

CS 775: Computer Graphics - Assignment 1

Due Date: 10/02/2015

1 Description

In this assignment we will animate the Transformer model from your final CS475/CS675 assignment using motion capture data from a BioVision Hierarchy (BVH) file.

The BVH file format was explained in class. You can read more about the BVH file format from here:

<http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>

2 To do

1. A basecode is provided to you for this assignment that comprises of two libraries:
 - (a) libUtil : This contains error handling and math classes
 - (b) libBVH : This contains the bvh_t, hierarchy_t, motion_t and the joint_t classes
2. The basecode sources and Doxygen documentation are available at:
http://www.cse.iitb.ac.in/~paragc/teaching/2015/cs775/assignments/A1/A1_basecode.tgz
You can generate this documentation for yourself by running “make docs.” This runs Doxygen on the basecode sources.
3. Example BVH files that your implementation should be able to read correctly can be found at:
http://www.cse.iitb.ac.in/~paragc/teaching/2015/cs775/assignments/A1/bvh_samples.tgz
4. Complete the libBVH library by implementing the methods that are marked “CS775: Implement...”

5. Add a GLFW/OpenGL interface as indicated in the library documentation to create the BVHplayer application which will render the animation contained in the BVH file.
6. The functions required to parse and read-in the BVH file are already present in libBVH and you should already be able to compile the basecode source and run it using a command like:

```
./BVHplayer -i examplefile.bvh
```

This should print out the contents of the BVH hierarchy.
7. Note: Please ensure that the rendering camera is positioned suitably with respect to the starting root position, otherwise you may not see anything on the screen.
8. Modify the hierarchy of your transformer to match the skeleton from the BVH file and match the degrees of freedom at each joint.
9. Draw your transformer so that each frame is posed as per the data in the “MOTION” section of the BVH file. Now, playing the BVH file should show your transformer animated as per the BVH file.
10. Also, create a rendering mode when the actual skeleton is drawn, instead of the transformer, by drawing thick lines between the joints. The lines can be cylinders and the joints can be spheres of various sizes. The program should be able to toggle between the two render modes.
11. Implement a camera that follows your character and one that stays stationary.

3 Marking

The assignment will be marked as follows:

- Implementing the methods to calculate the joint matrices correctly - 15 marks
- Implementing the methods to draw the hierarchy (skeleton only) in the rest pose - 10 marks
- Implementing the methods to render the animation (skeleton only) correctly - 15 marks
- Implementing the methods to render the animated transformer - 10 marks
- Implementing correct timing using glfw timers - 5 marks

- Implementing a keyboard interface to pause and move the animation forward/backward a frame at a time - 10 marks
- Implementing the cameras - 5 marks
- HTML report page - 5 marks
- Total : 75 marks
- Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., zero marks if the assignment is more than three days late after the due date.)

4 To Submit

1. A Tar-Gzipped archive of the complete source code (and only source code, no docs or bvh files). Textures may be included, if required. It should compile using the given Makefile on any Ubuntu system.
2. Include a README file in the .tgz archive containing a link to a html report page on the assignment that should contain some details about what you implemented and images of the results that you generated.
3. Submission will be done on the usual submission portal.