

Generic Cage Generation for Cage-Based Deformation

Group Members: Quincent Masters, Aaron Hornby, and Mikhail Starikov

1 Introduction

Within most animation styles, generally skeletal-based animation is used. However, when a model wants to have deformation applied or a model wants to be animated via cage models, the user must put in the time and effort to create the cage themselves.

1.1 Problem Statement

Given arbitrary 3D OBJ files, how would we be able to properly manage creating a cage mesh to apply deformation and interact with the user in an efficient manner?

2 Goals and Objectives

The primary objective of the project is to create an application, extension, or program that, when imported with an OBJ file, will generate an accurate cage mesh with appropriately weighted control points. This tool should offer the benefit of providing a lower point of entry to cage-based deformation.

3 Methodology

Preliminary methodology would require us to first find different OBJ files to render. In the rendering process, we would look at the models themselves and understand the variance in their topology. Similarly, we would like to ensure that the method for generating these cage meshes would be akin to cage meshes that have already been generated by other resources for their respective OBJ files. In doing this, we plan on implementing a base test case to ensure our application is running properly by applying MVC deformation on a preexisting model paired with its cage mesh. This would work as a basis test case moving forward as we further generalize and implement the application.

Extending from this preliminary outlook, we would then begin to incorporate an algorithm that would begin to create cage meshes for a variety of OBJ input files. Should this step be completed, we would then begin to implement user interactions for refining certain cage meshes and applying deformations.

3.1 Expected Results

Our results, in progressive order, should be the following:

- Generate efficient and accurate cage meshes for arbitrary OBJ files

- Implement rudimentary user interaction for refining cage meshes
- Implement rudimentary user interaction for selecting and applying cage-based deformations

4 Timeline

A general timeline would be to primarily achieve the above expected results in a timely manner, with the bulk of the focus of the project on the cage mesh generation functionality. As such, with regard to the presentations, we hope to have a preliminary attempt at cage mesh generation on OBJ models by the progress presentation, if not completely done.

5 Resources

Alongside the course notes, our primary resources will be accessed and found at the following links:

- <http://pers.ge.imati.cnr.it/livesu/papers/CCLS18/CCLS18.pdf>
- <https://github.com/cordafab/Cagelab2018>
- <https://pdfs.semanticscholar.org/4d1f/bfb7c9944690567c3f8cdfda5d6695516e39.pdf>
- http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/18_Deformation_1.pdf
- <https://github.com/superboubek/QMVC>

5.1 Programming Environment

For the programming environment, we would be using/utilizing the following: C++, OpenGL, GLEW, GLFW, GLM, Dear ImGui, and building for Windows/Linux.

6 Responsibilities

Joint responsibility for coding will be equally divided. The extra miscellaneous responsibilities will be distributed as follows:

Quincent: Primary and lead writer/editor for documents

Aaron: Primary and lead code manager, managing and organizing coding tasks

Mikhail: Primary resource manager and debugger