Custom Rigging System

Adam Mally

Advisors: Norman I. Badler, Aline Normoyle

University of Pennsylvania

ABSTRACT

In computer animation, character models need to be prepared with various deformers before an animator can bring them to life, as it were. As such, a well-rigged character is vital to the quality of an animation. There is a variety of commercially available software with which users may rig models, but such software often sacrifices additional rigging functions and tools in favor of functionality in other areas, such as animation or modeling. While a broader range of general capabilities may help sell a product, it occasionally leaves users with more specific needs wanting. This project attempts to improve upon basic functionality already present in commercial rigging software to create a program in which users may more easily and more robustly rig character models.

Project Blog: http://amallyseniorcapstone.blogspot.com

1. INTRODUCTION

In the computer animation industry, a rig is a set of virtual controls used by an animator to set the position and motion of a character model. Often, these rigs include a virtual skeleton that provides the bulk of the primary mesh deformation. A skeleton-based rig is a method of controlling a character via a hierarchy of transforming (though often just rotating) nodes that influence vertices on the mesh bound to them. When a node transforms, all nodes that are children of it reflect its transformation, and all points on the mesh influenced by those nodes will also reflect the transformation, simulating the way a skeleton moves in the physical world. A large issue with using a skeleton-based rig, however, is the process of assigning bone influence to the vertices of a mesh. In most modern rigging software, the program assigns default weighting to vertices based on a general distance function, leaving the user to spend time assigning appropriate bone weights.

Skeleton-based rigs will also often use an inverse kinematics algorithm to determine bone positioning to allow the user to more easily animate limbs. Traditional forward kinematics require the user to rotate a joint tree node by node until the entire tree is in the desired position. Inverse kinematics work by taking a joint position desired by the user and transforming the parent joints so that their combined transformations place the end joint in the input location.

Since a skeleton alone will not provide completely realistic character deformation, most animators will use additional rig components such as lattices, bend deformers, squashers, or stretchers to ensure their character jiggles, bends, bounces, and squashes how they want it to. These deformers work differently from a virtual skeleton in that they are not hierarchical in structure and tend to only provide one type of deformation as opposed to the rotation, scale, and translation of a skeleton.

This project aims to provide riggers with a set of tools that allow them to quickly build an accurate character rig. This will be accomplished by providing faster means of bone weight assignation through more refined automatic weighting functions, incorporate multiple types of mesh deformers into the traditional virtual skeleton tree. This tool set will be realized as a standalone program capable of exporting rig files in formats readable by other programs, such as the COLLADA format.

1.1 Design Goals

The target audience with this project is anyone with the need to make a character rig. The hope for this project is to provide said rigger with the tools he or she needs to quickly and accurately create a rig for a character while also providing a varied toolbox of deformers to apply to a character. By including deformers such as squash/stretch and lattice as part of a skeleton tree, we can allow the user to create rigs capable of more extreme and cartoonish animation than a standard skeleton rig would allow.

1.2 Projects Proposed Features and Functionality

This program will allow users to import polygonal meshes from .OBJ files and create animation rigs for them using deformers such as virtual skeleton joints, lattice deformers, benders, squashers, and blend shapes. It will provide an interface that is easy and relatively intuitive to use for applying the deformers to the mesh.

2. RELATED WORK

[ZB94] Jianmin Zhao's and Dr. Badler's paper "Inverse kinematics positioning using nonlinear programming for highly articulated figures" outlines a method for positioning an articulated figure via inverse kinematics computed through nonlinear programming techniques. Their technique implements a numerical method for computing the minimum of a nonlinear function, which is an efficient andf accurate method of solving for joint angles inan inverse kinematics chain. Zhao and Badler point out that their method is susceptible to becoming caught at local minima of a function, but they show that users can interactively adjust joints so that the skeleton becomes unstuck. This paper is relevant to my project since it covers an efficient and accurate method of computing inverse kinematics, which I wish to implement as part of my program.

[BP07] Ilya Baran's and Jovan Popovic's paper describes a method for automatically binding a 3D mesh to a given skeleton and then animating said skeleton from a library of pre-made motions. The skin binding is implemented by computing joint positions inside the character mesh by minimizing a penalty function. The skinning algorithm is iterative in that is begins with a less detailed, discretized representation of the character mesh and then repeats the process with more and more detailed mesh representations. This paper is relevant to my project since it deals with accurately weighting a skin for a skeleton-based rig, which I want to include as a feature in my program.

[BJD*12] This paper introduces algorithms for sketch-based 3D modeling that allows users to model and rig in parallel. It covers a series of steps in character creation and rigging, starting from a 2D sketch that is transformed into a 3D mesh and skeleton and going to a set of modeling tools used to expand on the model created from the 2D image. This paper is relevant since it describes a method for computing a skeleton for a given mesh, which is a feature I would like to try to implement.

[KS12] Kavan's and Sorkine's paper introduces an algorithm for generating skin weights that approximates nonlinear elastic deformations while remaining efficient. Its method is based on the concept of joint-based deformers, but unlike traditional linear blend skinning, this method can describe complex non-linear spatial deformations because it is based on the relative rotation of a parent joint and its child's rotation. This is relevant to mu project since I would like to implement a method of skinning that is more robust than classic linear weight blending.

[TGB00] This paper outlines an inverse kinematics algorithm suitable for animating an anthropomorphic arm or leg. It uses a combination of analyitical and numerical methods to solve inverse kinematics problems such as position, orientation, and aiming constraints. These methods are also faster than traditional Jabcobian and optimization-based algorithms. This paper is relevant since I will implement inverse kinematics in my program.

[LY12] Li's and Yueqi's paper describes a method for distributing skin weights for a humanoid mesh by generating a skeleton from the mesh then using joint positioning and linear interpolation to weight skin vertices. This paper is relevant since I will implement sutomatic skin binding.

3. PROJECT PROPOSAL

I plan on implementing a standalone application that allows users to quickly and efficiently rig a character model with various deformers. It will include context-based skin weighting algorithms for skeletal animation. It will also include the ability to export files that can be read by commercially available programs such as Autodesk Maya.

3.1 Anticipated Approach

The user will begin with a character mesh, which will be a polygonal shell mesh. Once the mesh has been read into my program, the user can set about rigging it. The user will have access to a variety of rigging components: a bone and joint system, nonlinear deformers (bend, squash, etc.), lattice deformers, inverse kinematics tools, and blend shapes.

The user may also choose to use an automatic system for rigging his character, such as a skeleton creator that uses the mesh imported by the user as a guide and an automatic skin weighting system that uses a heat-diffusion model to assign bone weights to vertices. Once a skeleton or skin weights have been created, the user can then manually refine the skeleton or weights to his liking.

Finally, once the user's rig is complete the user can export his model in a format that animation programs such as Autodesk Maya or Blender can read. The user can then animate his rigged model using other software.

3.2 Target Platforms

This toolkit will be implemented in C++ as a standalone application. It will use Qt for its user interface design.

3.3 Evaluation Criteria

The character rigging system can be tested by comparing rigging capabilities and ease of use to other rigging software like Autodesk Maya or Blender. If the rigging system is faster or more accurate for rigging than other software then the implementation was a success.

The interface itself can be tested simply by having naïve users attempt to rig a character with no prior instruction on how to use the toolkit. If the users are able to successfully navigate the interface and rig a character, then the toolkit was simple enough to understand from a layman’s point of view.

4. RESEARCH TIMELINE

Project Milestone Report (Alpha Version)

- Completed all background reading
- Implemented Obj loader
- Have completed a basic user interface
- Implemented a joint-based deformer system

Project Final Deliverables

- User capable of rigging meshes using joints, nonlinear deformers, blend shapes, inverse kinematics, and lattices
- Program capable of exporting rigs in formats readable by animation software
- Documentation

Project Future Tasks

- Improve the skinning algorithms even more, perhaps implement context-sensitive skinning.

References

[ZB94] ZHAO J. and BADLER, N. I. 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*. 13, 4, Pages 313-336 (October 1994).

[BP07] BARAN I., POPOVIC J.: Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics. Article No. 72.* (July 2007).

[BJD*12] BOROSAN P., JIN M., DECARLO, D., GINGOLD, Y., NEALEN, A.: RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation. *ACM Transactions on Graphics*, 31, 6, Article No. 198 (November 2012).

[KS12] KAVAN L., SORKINE, O.: Elasticity-Inspired Deformers for Character Articulation. *ACM Transactions on Graphics.* 31, 6, Article No. 196 (November 2012).

[TGB00] TOLANI D, GOSWAMI A., BADLER, N.I.: Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graphical Models,* 62(5), September 2000, pages 353-388.

[LY12] LI, D., YUEQI, Z.: 3D-Human Skin Weight Distribution Based On Linear Interpolation. *Computer Science and Education*, July 2012.

Schedule

	Jan		Feb				March				April		
	Week 3	Week 4	Week 1	Week 2	Week 3 (α)	Week 4	Week 1	Week 2	Week 3	Week 4 (β)	Wee k 1	Wee k 2	Week 3 (Due!)
Background reading	■	■	■	■									
User Interface		■	■	■	■	■	■	■	■	■	■	■	■
OBJ importer		■	■	■									
Skeleton system				■	■	■	■						
Other deformers						■	■	■	■	■			
Automatic skin bind-ing									■	■	■	■	■
Automatic skeleton generation									■	■	■	■	■