TECHNISCHE UNIVERSITÄT CHEMNITZ

BACHELOR THESIS

Inspection and comparison of automated methods for embedding skeletons and motion retargeting for 3D Scans

Faculty of Computer Science

Author:
Mick KÖRNER

Examiner:
Prof. Dr. Guido BRUNNETT
Supervisor:
Dr.-Ing. Thomas KRONFELD

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

November 17, 2024

Declaration of Authorship

I, Mick KÖRNER, declare that this thesis titled, "Inspection and comparison of automated methods for embedding skeletons and motion retargeting for 3D Scans" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF TECHNOLOGY CHEMNITZ

Abstract

Professorship of Computer Graphics and Visualization

Bachelor of Science

Inspection and comparison of automated methods for embedding skeletons and motion retargeting for 3D Scans

by Mick KÖRNER

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

Physical Constants

Speed of Light $c_0 = 2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

Introduction

1.1 Motivation

Virtual Humans have been a major Part of Computer Graphics because of its wide range applications, spanning multiple research domains.

Creating a realistic Virtual Human is still a challenge today. Digital Reconstruction techniques like Strucute-from-Motion can create a very Detailed Surface replication of a Person. However, this Mesh is static. If it is desired to animate this Scan with Motion Capture Data, the Mesh does not contain any Information on how to apply these.

While Motion-Capture techniques like Shape-from-Silhouette exist, which are creating an Animation by storing a 4D Mesh. The use Cases for these Results are limited because the Motion and Virtual Character are coupled.

Simplifying the Virtual Human problem to decouple Motion- and Surface Data has naturally developed to be the standard today, not only for Realistic Virtual Humans, but also heavily stylized ones in Movies and Games.

- Another important Motivation was to provide an easy to access and open source tool for motion retargeting, all widely used retargeting tools either require payment or an account login. Notibly there do no exist solid free motion retargeting Solutions.

- no basic tool for simple customizable motion retargeting

- while ik is already a common tool for animators to quickly get a desired pose, a well implemented and accessable motion retargeting can further improve an animators workflow by posing as a starting base for a desired pose using other motion editing tools

A deeper look into existing tools for these Problems reveals that many of them are sub-optimal or require some form of payment. Either in form of Currency or User Data.

1.2 Objectives and Scope

To facilitate the option to use a large set of Motion Data with Rigged Characters popular Tools like Mixamo use standardized Human like Skeleton to simplify the Process by moving the Motion Retargeting Problem to a Auto-Rigging Problem. Thus for a scalable system, the underlying Skeleton should be abstractable and independent of Motion Data. This is however not easy.

The primary Goal is a Tool which automates or streamlines the process of creating a Virtual Character just from a Scan. This includes the Implementation of Interfaces to easily add new methods for Autorigging and Motion Retargeting.

To further support Scalability for Future use. The proposed Tool should be interactive in order to test and compare algorithms more easily for correctness and potential drawbacks.

1.3 Summary of the Work

Firstly we will go over all Related Works in Chapter 2. This includes a Recap of how Computer Animation works and their basics. Then we go over Inverse Kinematics, Constraints up to Motion Retargeting and AutoRigging in Chapter 2.

In Chapter 3 the Design and Implementation of the Automation Tool is explained. As well as details specific Implementations of Motion Retargeting and Autorigging Methods or API interfaces.

Chapter 2

Related Work

2.1 3D Animation Basics (0.5 Basics)

2.1.1 Joints

2.1.2 Skeletal Format

2.1.3 Pose Space vs. Work Space

2.1.4 Forward Kinematics

2.2 Inverse Kinematics (1. Inverse Kinematics)

In the previous Section we learned that Forward Kinematics takes Input from the Configuration Space of a Rigged Model and gives us Working Space Coordinates we can use to do things like Rendering or Collision.

An natural desire is to know a Configuration to target Points in Working Space.

Inverse Kinematics

2.2.1 Analytical Methods

The analytical approach tries to solve the system of equations spanned by inverting the Forward Kinematics formula of the corresponding armature.

While this solution would be Ideal because it is very fast and numerically perfect. Solving the system for more than two Joints becomes with each additional Joint harder.

This is because already in 3D compared two 2D, two joints yield an infinite amount of solutions for a reachable Point in space.

2.2.2 Jacobian Methods

The Jacobian Inverse Method for solving Inverse Kinematics represents the first Iterative Approach.

2.2.3 CCD

- first heuristic method

2.2.4 FABRIK

- builds and optimizes apon ccd

2.2.5 Other Methods

2.3 Constraints (2. Constraints)

2.3.1 Constraint Types

2.3.2 Jacobian Constraints

2.3.3 CCD Constraints

2.3.4 FABRIK Constraints

2.3.5 iTASC

2.4 Motion Retargeting (4. Motion Retargeting)

2.4.1 Available Tools

2.4.2 Naive Retargeting

2.4.3 Limb based Retargeting

[3] - describes the problem to be hard to solve mathematically because of how to define the quality of a motion - require basic features of motion identified as constraints

Limb based Motion Retargeting approaches abstract Joints into Joint Chains, where each Chain is retargeted individually.

[2]

2.4.4 Jacobian based

Choi and Ko use Inverse Rate Control, which is the Jacobian Inverse Method of Inverse Kinematics, and extend it to be applicaple to tree structures instead of chains without branches. [1]

Choi and Ko have also showed a way to imitate joint angles of the source motion by incorporating them as a secondary goal.

2.4.5 Machine Learning Approaches

2.4.6 Other approaches

2.5 Automated Rigging (5. Autorigging)

2.5.1 Machine Learning Approaches

2.5.2 Thinning Approaches (TODO genauer anschauen für mögliche impl?)

2.5.3 Skin Matching Approaches

2.5.4 SMPL fitting

2.5.5 Re-Meshing

Chapter 3

Motion Retarget Editor (6. Editor)

3.1 Chosen Tools

- Also having an open source foundation opens up community improvements and helps CrossForge mature by protyping features and incorporating them if deemed useful
- a

3.1.1 Imgui Integration

3.2 Classes and Scene Management

3.3 Animation System

3.3.1 Sequencer

3.3.2 Picking

3.3.3 Editing Tools (Restore Restpose, apply Transform etc.)

3.4 Inverse Kinematics Implementation

3.4.1 Jacobian Method

3.4.2 CCD

3.4.3 FABRIK

3.5 Skeleton Matching

3.6 Motion Retargeting

3.7 Constraints Implementation

3.8 foreign tool linking

3.8.1 Rignet

3.8.2 deep-motion-retargetin

3.9 Import and Export

3.9.1 Model Data

3.9.2 Animation Data

Chapter 4

Conclusion and Future Work (7. Future)

4.1 Editor Improvements

4.2 Utilizing Skinning Alternatives

4.3 Other Useful Tools

4.4 Clothing

4.5 Motion Blending

4.6 Blender Addon

Bibliography

[1] Kwang-Jin Choi and Hyeong-Seok Ko. "On-Line Motion Retargetting". In: (1999).

[2] Yann Pinczon Du Sel, Nicolas Chaverou, and Michaël Rouillé. "Motion Retargeting for Crowd Simulation". In: *Proceedings of the 2015 Symposium on Digital Production*. DigiPro '15: The Digital Production Symposium. Los Angeles California: ACM, Aug. 8, 2015, pp. 9–14. ISBN: 978-1-4503-3718-2. DOI: 10.1145/2791261.2791264. URL: https://dl.acm.org/doi/10.1145/2791261.2791264.

[3] Michael Gleicher. "Retargetting Motion to New Characters". In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*. The 25th Annual Conference. Not Known: ACM Press, 1998, pp. 33–42. ISBN: 978-0-89791-999-9. DOI: 10.1145/280814.280820. URL: http://portal.acm.org/citation.cfm?doid=280814.280820.