Differencing and Merging for 3D Animation Revision Control

George Madges[1] and Idris Miles[1] and Eike Falk Anderson[1]

[1]The National Centre for Computer Animation, Bournemouth University, United Kingdom

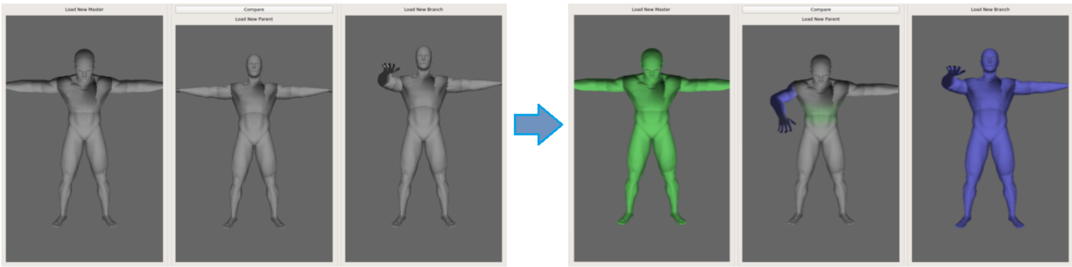

Figure 1: *Example of our application applying a 3-way difference and automatic merging of 3D animations: Left image shows the input animations, with middle viewport displaying the parent animation, left viewport branch A and right viewport branch B of the animation. Right image showcases the results of our technique, with the middle viewport showing the final merged animation, shaded to highlight the sources of the animated joints; left viewport again shows branch A and right viewport shows branch B of the animation.*

Abstract

Version Control System (VCS) techniques for managing the creation of 3D computer generated models exist for static 3D models, however, there are no solutions for revision control of the animation data from animated 3D models. A precondition for any type of VCS is the ability to compare two versions of an item, to identify the differences between them and to combine these. To this end, we propose a novel 3-way difference, merging and conflict resolution technique for 3D animation data.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Software support

1. Introduction

Version Control Systems for data created by multiple users require two types of operations to exist: A method to compare different versions of the data and to identify differences between them, and a method for merging these different versions of the data in a meaningful manner. We propose methods that would satisfy these criteria for a 3D animation data VCS (Figure 1).

2. Related Work

Several matching algorithms for calculating the differences between 3D model versions have been described by Denning and Pellacini [DP13]. Related to this, Doboš and Steed [DS12a] present a system that allows 3D model designers to compare static (non-animated) 3D models with one another to highlight differences and modifications between these, which forms the basis for a 3D model revision control system [DS12b]. While so far no differencing techniques for animations specifically targeting 3D model VCS's exist, there are techniques for motion caption data analysis applicable to comparing animation sequences [vBE09].

We have previously developed an algorithm for simple comparison of different animations of the same 3D model (AnimDiff [MMA17]) that can form the basis for a more comprehensive differencing technique. This algorithm for simple 2-way differencing of animation sequences makes a number of assumptions:

1. An animated 3D model contains an endo-skeletal hierarchical control structure, i.e. a rig of connected joints, that drives the animation of the 3D model's skinned mesh.
2. An animation consists of a set of frames $A = \{K_0, K_1, \ldots, K_n\}$.
3. A frame consists of a set of joint transformations encoding a pose of the rig $K = \{J_0, J_1, \ldots, J_n\}$.
4. A joint transformation consists of a 3D position and scaling vector and a rotation quaternion $J = \{p, s, r\}$.

This then allows us to compare two versions of the rig, calculating a 2-way difference of the master$_m$ and branch$_a$ animation

DOI: 10.2312/egp.20171038

$\Delta A = 2Way(A_m, A_a)$, where $\Delta A = \{\Delta K_0, \Delta K_1, ..., \Delta K_n\}$ is calculated by

- comparing the difference kinematic hierarchy for each frame $\Delta K_i = \{\Delta J_0, \Delta J_1, ..., \Delta J_n\}$
- that requires finding the differences between the corresponding joint transformations $\Delta J_j = \{\Delta P, \Delta S, \Delta R\}$
- which includes the differences for
  Position $\Delta P = P_a - P_m$,
  Scale $\Delta S = S_a - S_m$ and
  Rotation $\Delta R = R_a R_m^{-1}$

For merging animations, techniques that have been developed for blending animations [Per95] can be employed.

3. Differencing and Merging of Non-linearly Edited 3D Animations

In non-linear edited animations, differences of more than one edited (child) animation need to be compared with the original (parent) revision, which can then be merged, however merging may result in conflicts that need to be resolved.

3.1. Multiple (3-Way) Animation Difference

Our method starts by calculating the difference between the parent revision and each of its children, where a 3-way difference involves the parent (master$_m$) animation and two child animations (branch$_a$, branch$_b$): $\Delta A_{a,b} = 3Way(A_m, A_a, A_b)$.

Each of the 2-way differences $2Way(A_m, A_a)$ and $2Way(A_m, A_b)$ results in the individual changes that have been made in each branch. We then compare both sets of difference data, checking for cases where delta values of joint transformations are set in one branched revision and null in the other. It is comparable to the notion of an XOR operation in Boolean Algebra where the XOR between 2 sets of differenced animation data produces a new combined difference: $\Delta A_{a,b} = \Delta A_a \oplus \Delta A_b$.

When edit changes of both branch$_a$ and branch$_b$ affect the same joint at the same key-frame (the keys, i.e. values of the transformations for all joints of a rig, stored for a specific time-step of an animation), this is registered as a conflict: $Conflict = \Delta A_a \wedge \Delta A_b$.

Recognizing a conflict is important to the user as it indicates a disagreement over how that particular bone should be animated which may not be possible to resolve automatically. In this case our system highlights the conflict (Figure 2, middle panel) and lets the user decide which revision of the animation is used or to interpolate between the two revisions (Section 3.2).

A 3-way difference in our system is therefore defined as:

$$\Delta A_{a,b} = (\Delta A_a \oplus \Delta A_b) \vee User\ Resolution(\Delta A_a \wedge \Delta A_b).$$

3.2. Merging and Conflict Resolution

A useful mechanism in version control is the ability to merge together two descendants of a common ancestor into a new version that the user would then continue to work with. By employing the 3-way difference data, we can apply the delta values to the master animation to produce a new merged animation: $A' = A_m + \Delta A_{a,b}$

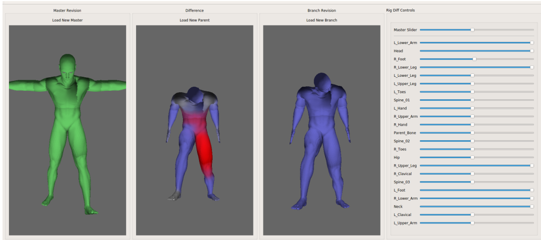

Figure 2: *A demonstration of the results of a merge conflict from the automatic merging after a 3-way difference. The middle viewport displays the merged results, parts of the mesh highlight red indicate conflicting animations.*

We play this merged animation back to the user whilst highlighting various parts of the rig and mesh to indicate the source of its animation (Figures 1 and 2).

Conflicts can be resolved by the user through manual merging, which the user interface allows in the form of a series of sliders corresponding to each joint of the animated rig. The sliders allow the user to blend the animation of that joint between the two branches. The shading of the joint and surrounding mesh are also updated from red, indicating a merge conflict, to the corresponding colour of the branch selected.

4. Discussion

Our system has the ability to compare and merge several animation revisions and provides the user with control over how much individual joints in the kinematic hierarchy are influenced in the merge process. Our approach could be extended to include differences to additional revisions by adding further 2-way difference calculations and XOR operations.

5. Acknowledgments

We would like to thank Valery Adzhiev for his support and useful suggestions.

References

[DP13] DENNING J. D., PELLACINI F.: MeshGit: Diffing and merging meshes for polygonal modeling. *ACM Trans. Graph. 32*, 4 (July 2013), 35:1–35:10. 1

[DS12a] DOBOŠ J., STEED A.: 3D Diff: An interactive approach to mesh differencing and conflict resolution. In *SIGGRAPH Asia 2012 Technical Briefs* (2012), pp. 20:1–20:4. 1

[DS12b] DOBOŠ J., STEED A.: 3D revision control framework. In *Proceedings of the 17th International Conference on 3D Web Technology* (2012), Web3D '12, pp. 121–129. 1

[MMA17] MADGES G., MILES I., ANDERSON E. F.: AnimDiff: Comparing 3d animations for revision control. In *Eurographics 2017 – Short Papers* (2017). 1

[Per95] PERLIN K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (1995), 5–15. 2

[vBE09] VAN BASTEN B. J. H., EGGES A.: Evaluating distance metrics for animation blending. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (2009), pp. 199–206. 1