Fast, Interpolationless Character Animation Through "Ephemeral" Rigging

Raf Anzovin
raf@anzovin.com
Florence, Massachusetts

Figure 1: A jumping cat, animated interpolationlessly with an ephemeral rig.

ABSTRACT

I present an alternative CG character animation methodology that eschews both keyframes and conventional hierarchical rigging. Primary rig controls have no hierarchy or built-in behavior–instead the animator calls for "ephemeral" rig behavior as needed. The system also facilitates "interpolationless" animation by removing keyframes as we know them, replacing them with discrete poses and inbetweening tools.

CCS CONCEPTS

• **Applied computing** → **Media arts**.

KEYWORDS

animation, rigging

ACM Reference Format:
Raf Anzovin. 2019. Fast, Interpolationless Character Animation Through "Ephemeral" Rigging. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328165

1 INTRODUCTION

Hierarchical rigging and keyframe animation are so ubiquitous that they are sometimes seen as synonymous with the idea of CG animation itself. Nevertheless, they present significant problems for the character animator, stemming from the indirectness of the approach.

Multiple combinations of rig state and keyframe placement can result in the same apparent movement on screen, greatly increasing the cognitive load associated with understanding and editing motion. Unwanted behavior frequently emerges "mysteriously" from interference between the motion of different levels of control in a hierarchical rig, and can become quite difficult to fix. Animators are forced to manually manage rig state transitions (such as space and FK/IK switching), again increasing the animator's cognitive load and focusing attention away from the character's performance and onto technical considerations. It becomes quite easy for the animator to paint themself into a corner where some alterations are easy and others prohibitively difficult.

SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6317-4/19/07…$15.00
https://doi.org/10.1145/3306307.3328165

2 PRIOR WORK

Animators have attempted to work around these issues by using a "Pose to Pose" or "Blocking Plus" approach to keyframing for decades, as exemplified by Keith Lango's 2006 article, "Breakdowns Can Be Such a Drag." [Lango 2006] Valve's Source Filmmaker [Valve 2012] presented a comprehensive interface for editing dense animation data, but with limited rigging tools. Motionbuilder [Kaydara 1994] decoupled rig manipulation and rig interpolation, but retained keyframes and interpolation as its central model. Raffaele Fragapane has advocated moving some rig behavior into a "tool" on his stream Cult of Rig [Fragapane 2019], but evaluates most interaction between primary rig controls conventionally.

3 EPHEMERAL RIGGING

In order to allow the animator full freedom to pose the rig without managing its state, I have designed a system in which all rig behavior of primary controls is temporary (ie. "Ephemeral"). The system is implemented as a dependency graph that operates outside Maya's internal DG. Absent the ephemeral behavior, all primary controls in this system are simply transforms in world space with no relationship to each other. These transforms drive a lower level deformation rig that deforms the mesh conventionally through the Maya DG.

In many respects the ephemeral graph resembles Maya's DG, or any other DG-based rigging system. It is a network of nodes evaluated in order of dependency, complete with familiar rigging concepts such as "constraints" that evaluate the node's new transformation in response to its dependencies. However, the graph does not persist between user interactions. Instead, it is continually discarded and rebuilt as needed to provide the currently desired rig behavior.

This system subsumes disparate rigging concepts that are usually implemented in an ad-hoc fashion (such as "space switching," "FK/IK switching," and "reverse foot rigs") into a single concept, as any rigging behavior supported by the system can be called for on any node without being "pre-built" into the rig. Rebuilding the graph also allows for bidirectional behavior that would create a circular dependency in a conventional rigging graph, such as interchangeable forward and "backward" kinematics.

Animators may manually establish temporary dependencies between nodes, or use a variety of predefined modes that rebuild the rig into useful interaction states. Binding these modes to keys allows for very fast character posing.

The current implementation supports rigs that combine ephemeral and conventional controls for performance reasons; face and finger controls are, for instance, commonly conventional. I suspect this will not be necessary in future iterations of the system.

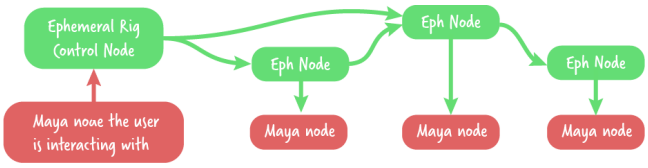

Figure 2: The ephemeral dependency graph is triggered when the user interacts with a node, and pushes its results back into the Maya DG

4 INTERPOLATIONLESS ANIMATION

Interpolationless animation treats the character's motion purely as a series of discrete poses. Since rig state does not persist, this creates a one-to-one relationship between what the animator sees and the controls that create that motion. This workflow is very friendly to traditional animators, and broadly mimics the behavior of 2D animation tools.

Once created, poses require no evaluation, which makes it easy for the system to read or write them regardless of what frame the play head happens to be at. To create an inbetween between two key poses the system can pull the data of past and future poses and construct an ephemeral graph that interpolates past and future pose data. All rotation is interpolated with quaternions, removing any concern for rotate order. In addition to creating inbetweens, this technique may be used to "smooth" poses with each other to remove discontinuities in motion.

Crucially, this method of creating inbetweens does not create any persistant relationship between poses; it simply generates new ones. Hence, it may be called "interpolationless" despite using interpolation as part of its process.

In practice, this produces a workflow similar to the "pose and breakdown" methodology already common among animators, but without "splining." In place of a graph editor or motion trail, motion is visualized through an onion skinning tool written by Christoph Lendenfeld[Lendenfeld 2018].

This technique works extremely well for cartoony or stylized animation production. My own experience animating with the system suggests productivity gains as high as 200 to 300 percent are possible when working in a compatible style. Further research is needed to verify this conclusion.

5 PERFORMANCE

By building only the graph necessary for a given rig interaction, ephemeral rigging may prune control rig complexity by an order of magnitude or more compared to conventional control rigs. During scrubbing and playback, no ephemeral control rig evaluation is necessary at all.

The current ephemeral rigging prototype is implemented entirely in Python with no effort made at optimization. Nevertheless, the reduction in complexity is such that an ephemeral graph including all primary rig controls for a standard biped evaluates in only 0.324 milliseconds. Pushing the results of that graph back into the Maya scene via Open Maya 2 costs 11 millseconds, accounting for the vast majority of the system's cost. An implementation using Maya's C++ api might reduce the performance cost of the entire ephemeral system to a fraction of a millisecond, even if single-threaded.

6 FUTURE WORK

The current system is less effective than more traditional methods at subtle or realistic motion. To allow the ephemeral-interpolationless approach to be used for a broader range of animation styles would require more advanced inbetweening tools and multiple-pose editing tools. Directly editing dense animation data is a promising subject for future research, as is an optimized version of the ephemeral rig graph. More data from a wider range of animators would also be helpful in establishing the degree of productivity gain.

Ephemeral rigging could also be decoupled from interpolationless animation, by using an ephemeral rig as a posing tool for an underlying hierarchical rig that handles interpolation.

ACKNOWLEDGMENTS

Thanks to Raffaele Fragapane, Tagore Smith, and Bay Raitt for an introduction to previously unknown concepts, Brian Kendall for timely advice, Brad Clark for encouragement, and Christoph Lendenfeld for saving me from the need to write my own onion skin tool.

REFERENCES

Raffaele Fragapane. 2019. Pilot Season - Day 13 - Scripting Version of fk ik Switch. Retrieved February 10, 2019 from http://www.cultofrig.com/2017/07/08/pilot-season-day-13-scripting-version-of-fkik-switch

Kaydara. 1994. Motionbuilder. https://www.autodesk.com/products/motionbuilder

Keith Lango. 2006. Breakdowns Can Be Such a Drag. Retrieved February 10, 2019 from http://www.keithlango.com/tutorials/overlap/overlap.html

Christoph Lendenfeld. 2018. Onion Skin Renderer for Autodesk Maya. Retrieved February 10, 2019 from https://github.com/Viele/onionSkinRenderer

Valve. 2012. Source Filmmaker. http://www.sourcefilmmaker.com