Creating Ralphzilla

Moshpit, Skeleton Library and Automation Framework

Dong Joo Byun
Walt Disney Animation Studios

Alberto Luceño Ros
Walt Disney Animation Studios

Alexander Moaveni
Walt Disney Animation Studios

Marc Bryant
Walt Disney Animation Studios

Joyce Le Tong
Walt Disney Animation Studios

Moe El-Ali
Walt Disney Animation Studios

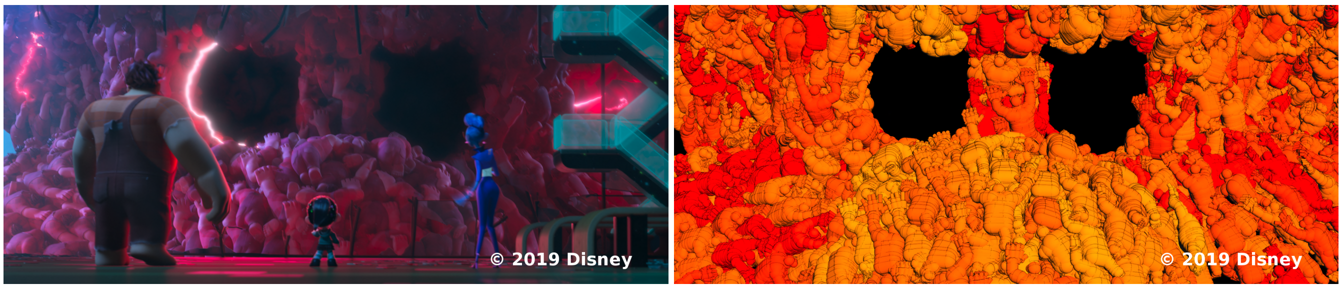


Figure 1: Left: Final render with all layers. Right: proxy geometry in Moshpit simulation.

ABSTRACT

Composed of over 550,000 crowd agents, Ralphzilla of ***Ralph Breaks the Internet***, one of the largest movie monsters ever created and presented a huge technical and artistic challenge.

We introduce a new crowd solver, Moshpit, which performs high resolution inter-body collision among crowd agents. We will also explain how Moshpit was incorporated into Disney's Skeleton Library (SL) and proprietary pipeline automation framework.

CCS CONCEPTS

• **Computing methodologies** → **Simulation by animation**; **Motion processing**; **Collision detection**.

KEYWORDS

crowd, ragdoll, bullet solver, skeleton library, automation

ACM Reference Format:
Dong Joo Byun, Alberto Luceño Ros, Alexander Moaveni, Marc Bryant, Joyce Le Tong, and Moe El-Ali. 2019. Creating Ralphzilla: Moshpit, Skeleton Library and Automation Framework. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328179

1 INTRODUCTION

Ralphzilla is made up of densely packed crowds that dictate the character's shape and movement. From the creative development, it was evident that the crowds making up Ralphzilla needed to follow specific textural movement to reflect the character's emotion, while showing detailed collision among individual crowd agents. At the same time, the sheer number of shots and complexity of the Ralphzilla's task required an automation.

We needed a new solver that supported higher resolution collisions between crowd agents and their limbs while keeping input reference animation. We will talk about the creation of this new solver, Moshpit and its integration with Skeleton Library. We will also demonstrate how the Ralphzilla rig was designed with our automation framework to ensure a consistent look while dealing with complexity.

2 MOSHPIT

2.1 Concept

Ragdoll physics has long been used to simulate physically accurate collisions among crowd agents and non-crowd colliders [Mulley and Bittarelli 2007]. Ragdoll physics uses point constraints and cone-twist constraints to connect the proxy geometries of crowd agents. Crowd agents move by referencing cached animation cycles; however, when using a ragdoll system, the reference animation is disconnected from the proxy geometry. While this produces physically-accurate simulation, the result often deviates from the input animation. To ensure that crowd agents follow the input reference animation, we introduced two major modifications to the solver: (1) velocity-driven transformation of proxy geometry, and (2) dynamic axis updates of the cone-twist constraints during simulation.

In the first modification, the velocity and angular velocity are calculated by taking the difference of position and rotation of the proxy geometry and the reference geometry. The velocity and angular velocity are then used to drive the proxy geometry during the rigid body simulation to achieve the reference animation.

$$v = \mathbf{p}_{proxy} - \mathbf{p}_{ref}$$

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

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328179

$$\omega = \theta_{proxy} - \theta_{ref}$$

In the second modification, the up and twist axes are calculated by taking the difference of each joint's relative rotation to their parent joint from the previous time step and the target relative rotation from their reference geometry. The calculated up and twist axes are used to update the cone-twist constraints during the simulation to maintain the rotation of the joints from the reference animation.

$$R_{constraint} = (R_{ref})^{-1} \times R_{target}$$

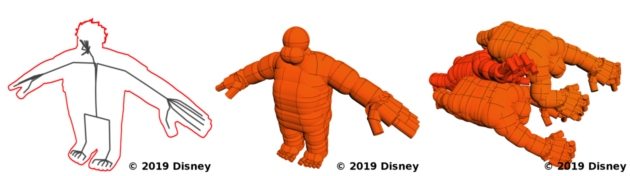
2.2 Implementation

We developed Moshpit by modifying Houdini's native Bullet dynamics. We implemented the aforementioned modifications as two microsolvers in Houdini DOPs. The first micro solver calculates the velocity and angular velocity of the proxy geometry. The micro solver refers to the target position and rotation from the reference geometry. The velocity and angular velocity are then used to drive the transformation of the proxy geometry during collision detection in the Bullet solver. The second micro solver updates the direction of cone-twist constraint axes on each time step. This micro solver also interpretes the target direction from the reference geometry. In ragdoll physics, we usually apply loose rotation limitation to constraint axes to represent each joint's flexibility. However since our second micro solver calculates the exact rotation, we were able to apply very tight rotation limitation.

In order to reduce complexity and to modularize and optimize simulation, we ran a PBD simulation [Müller et al. 2007] on sphere-shaped particles representing the individual crowd agents before running Moshpit. The translation, up, aim direction and locomotive vectors of particles were derived from reference animation. In order to control the global speed and direction of the particles, we designed flow fields in Ralphzilla's body.

3 SKELETON LIBRARY (SL) INTEGRATION

Moshpit's design provided tight integration with our proprietary crowds system called Skeleton Library (SL). Since its adoption on *Zootopia* [El-Ali et al. 2016], SL has been used in production on all subsequent Walt Disney Animation Studios features and shorts. SL provides a powerful and reliable mechanism for crowds and skeletal manipulation. Though platform independent at its core, it is tightly coupled inside Houdini, making it readily accessible. Integrating with SL allowed Moshpit to leverage all of its functionality (e.g. access to all crowds tools, etc.). The two systems connect in 4 key ways: (1) The performance and animation of the individual agents are controlled in an SL context, where all the crowds tools are available to the artist. (2) The initial PBD simulation uses the locomotion from the SL's playback of crowds animation cycles. (3) Target positions for bullet objects and the bullet constraint network are calculated from a wire representation of said animation generated through the SL (4) The output of the bullet simulation is converted into a valid SL crowds particle cache. The new animation produced by Moshpit is stored in the form of joint offsets (in essence, transformation data per joint). After this conversion process the data is fully compatible with the studio's existing crowds pipeline.



(a) Skeleton from SL **(b) Proxy geometry** **(c) Simulation**

Figure 2: Moshpit.

4 AUTOMATION

The Ralphzilla effect was featured prominently in two sequences with a total over 104 shots. Artists needed to art-direct the effect to bring out the monster's acting; however, the rig was especially unwieldy owing to the variety of pipelines that were glued together.

Considering the amount of complexity which we were dealing with, maintaining a consistent look would have been extremely time consuming if the effect was done by hand. Thus, we needed to automate the pipeline so that the artists could focus on the art direction while delivering a consistent effect. We used a newly developed rig repository and deployment framework to automate the effect. This framework's most basic functionality is to build a proper work area and scene file with an instance of the chosen rig; an artist can then open the scene file with a "blessed" pipeline already in place. Furthermore, this system has the ability to submit render dependency graphs for compatible rigs. This way we can deploy the rig to a destination shot and get back completed renders. While developing the Ralphzilla rig, we would periodically batch re-distribute / render dozens of shots at a time, testing corner cases, better informing the rig design, and getting an animation blocking pass practically for free.

5 PRODUCTION RESULTS

From idea pitch to final output, creating Ralphzilla was a huge cross departmental collaboration. 130 artists and production people from eight departments collaborated closely to accomplish this task in nine months.

Simulating accurate collision between parts of crowd agents while keeping animation fidelity is one of the most difficult problems in crowd simulation.

We not only solved this issue by introducing Moshpit, but we also integrated the solver with our pre-existing crowd toolset, SL, and our newly-developed automation framework. Leveraging SL's sophisticated geometry representation and existing toolset allowed us to focus on our specific challenge. The automation framework packaged rigs in a compatible format, supporting artists to achieve the desired artistic goal and maintain a consistent look. As a result of these efforts, the Crowds and Effects departments were able to run the same Ralphzilla automation rig for all shots.

REFERENCES

Moe El-Ali, Le Tong, Josh Richards, Tuan Nguyen, Alberto Luceno Ros, and Norman Moses Joseph. 2016. Zootopia Crowd Pipeline. In *ACM SIGGRAPH 2016 Talks (SIGGRAPH '16)*. ACM, New York, NY, USA, Article 59, 2 pages. https://doi.org/10.1145/2897839.2927467

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (April 2007), 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005

Gabe Mulley and Matt Bittarelli. 2007. Ragdoll Physics. (2007).