Wham! Building *Deadpool*'s Freeway Chase

Will Muto* Atomic Fiction
Marc-Antoine Paquin* Atomic Fiction
Nico Sanghrajka* Atomic Fiction
Stuart Gordon* Atomic Fiction
Monique Bradshaw† ConductorIO

Figure 1: *Breakdown of chaos and destruction in Deadpool. ©Twentieth Century Fox. All Rights Reserved.*

Abstract

The high-speed chase sequence in *Deadpool* encompassed one hundred shots of new challenges for the team at Atomic Fiction. The action had to move through many miles of densely populated fictional city, leaving a trail of destruction in its wake. The pipeline team developed novel approaches across the VFX production workflow to deal with a sequence of this scale and complexity.

Keywords: layout, animation, simulation, destruction, pipeline, cloud rendering, big data

Concepts: •**Computing methodologies** → **Animation; Procedural animation; Physical simulation; Distributed algorithms; Rendering;**

1 GPU-Optimized Layout Assembly Pipeline

With lessons learned from building New York City for *The Walk*, we developed an XML-based assembly system where layout artists could load and shift large sections of the city around using GPU-optimized models. Animators were able to load the sections of the environment that were pertinent to them, and could make any shot-specific changes to the layout if required; animation for multiple sequential shots could be published at once, and caches were automatically retimed to match the cut. Camera-based analysis was automated so downstream departments such as FX and lighting were provided culling and bounding information. The assemblies provided efficient geometry instantiation at render time, and additional optimizations such as displacement and subdivision could be applied across the cityscape quickly.

*{wmuto, mapaquin, nsanghrajka, sgordon}@atomicfiction.com
†monique@conductorio.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927419

2 RBD-Driven Character Animation

To create the initial performance of an actor being dragged outside a high speed vehicle, rigid body dynamic (RBD) simulations were used to provide a base for character animation. This was achieved by creating a dynamic simulation rig, which could collide against the ground while respecting the rotational limits of a human skeleton. The simulated performance was then retargeted to the character animation rig. That rig allowed the animators to blend in or replace custom animation on the simulated base performance.

3 Destruction: FX to Anim and Back Again

To create a realistic performance during destruction, we wanted to utilize the best of animation and the best of simulated dynamics, with minimal impact to the lookdev setup of the original asset. To do this, we developed a robust set of tools for moving back and forth between animation and FX. Initial RBD sims were used for tumbling cars and debris, which were then taken into animation where artists could refine the performance in shot context. The vehicle animation rig also utilized a real time dynamic system to simulate the vehicle suspension as well as volume preserving controls on the tires. FX could then simulate further destruction on top of this animation, breaking pieces off or completely shattering the objects. To streamline the process of delivering into lighting, we simplified the integration of simulated geometry by automatically merging parts that were pulled from assets for RBD simulation back into animation caches at publish time.

4 Anim-Friendly Traffic Toolkit

With thirty-five shots to populate with background traffic that needed to integrate with hero vehicles in a rapidly shifting layout, we needed a solution where the animators could quickly add traffic to the roadways themselves. We created a custom toolkit that allowed artists to build roads from curves extracted from the layout geometry, scatter cars along the surface, and run a simulation that could be adjusted and cached. Vehicles in the simulation reacted to surrounding traffic, switching lanes and slowing down when overtaking another vehicle. Since the road was based on a parameterized surface, layout adjustments could be made immediately. Individual car models could be hand-placed in the simulation in two clicks, and caching could be managed per car or for the traffic system as a whole. Level of detail allowed the animators to work in

real time using bounding boxes or GPU caches, or with high resolution geometry for review. On approval, the simulation was baked to a low-footprint Alembic using transforms and metadata from which vehicles were instanced in the lighting package at render time.

5 Conductor: Cloud-Based Rendering

Atomic Fiction leveraged Conductor's cloud compute system to handle the large volume of rendering iterations needed for the sequence. The sequence used 6.5 million core hours to achieve the final product. The flexible resource capabilities of the cloud avoided the constraints of static infrastructure, dynamically surging up to 32,000 simultaneous cores for weeks at a time. This provided the team the ability to shift their compute workload to the time when it would be most beneficial to the production rather than having to manage the workload into a fixed resource allocation. An additional benefit of using the cloud was the large amount of metadata generated that can be utilized to better understand and estimate render times, improving project bidding and cost management.

Contributors: Rudy Grossman & Laurent Taillefer, Atomic Fiction