Mortal Kombat 11: High Fidelity Cached Simulations in Real-Time

Jason Nadro
NetherRealm Studios
jnadro@netherrealm.com

Matt Battaglia
NetherRealm Studios
mbattaglia@netherrealm.com

Aren Voorhees
NetherRealm Studios
avoorhees@netherrealm.com

Figure 1: Fatality blood effect

ABSTRACT

Mortal Kombat 11 introduces an Alembic-based asset pipeline that enables artists to leverage new workflows previously unattainable in real-time games. Blood and gore are the cornerstone of the franchise, and the art direction for our Fatalities and Fatal Blows focuses on close-up, high-fidelity, slow-motion shots showing extreme amounts of blood, which traditional sprite particles would struggle to achieve.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**.

KEYWORDS

realtime rendering, games, art production, geometry caches

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

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328187

ACM Reference Format:
Jason Nadro, Matt Battaglia, and Aren Voorhees. 2019. Mortal Kombat 11: High Fidelity Cached Simulations in Real-Time. In *Proceedings of SIGGRAPH '19 Talks*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328187

1 INTRODUCTION

Realistic fluids and physically based dynamic simulations are typically outside of the reach of real-time video games which only have 16.6 ms or 33.3 ms to render everything their game requires. Most of this time is dedicated to shadowing, lighting, accurate material models, and post processing effects. Mortal Kombat 11 renders its whole frame in 16.6 ms with every millisecond budgeted and accounted for. This leaves no extra room to run real-time dynamic simulations of fluids. By running fluid simulations offline with Houdini and efficiently caching the animation we trade memory for performance.

Our artists used Houdini, a 3D animation software application, to create FLIP-based fluid simulations and bake them into a mesh-based vertex animation called a geometry cache. Our new technology allows the direct playback of these geometry caches in real-time using a modest amount of memory suitable for games.

2 PREVIOUS WORK

In 2014 Ryse: Son of Rome [Evans et al. 2014] demonstrated that complex, physical simulations can be baked into animations for real-time applications using the Alembic file format. It was used for large scale destruction events, cloth, and facial animation. These use cases of geometry caches assume a constant topology frame-to-frame of the mesh being deformed. In other words, the vertex and triangle count of the mesh remains constant. This allows for efficient storage, compression, and playback of the data.

The fluid simulation geometry caches used for the blood effects in Mortal Kombat 11 require changing topology from frame-to-frame which previous work has not focused on. The main limiting factor of changing topology geometry caches is the prohibitively large memory usage from storing vertex and index buffers for full, discrete meshes for each frame of the animation.

3 OUR CONTRIBUTION

We will demonstrate that the vertex data in fluid simulation geometry caches can be highly quantized and the tangents and UV attributes are unnecessary to shade blood effects realistically. With these insights we achieved a 64 percent reduction in the size of the vertex data in the cache, allowing this feature to appear in hundreds of gameplay moments. For vertex attributes, our geometry caches store positions, normals, and what we call a "user channel." The user channel is a single byte that can store an animated value of the artist's choice. For our blood effects, it was used to store a mask that drove a subsurface lighting effect in the shader.

3.1 Accuracy

Table 1 summarizes the accuracy of our storage solution for vertex positions compared to using half-precision floating point numbers. While our RMS error is an order of magnitude worse than half-precision, in practice we found no significant visual differences in the resulting mesh. The error manifests as small shifts in the vertex positions which can be seen in Figure 2.

The RMS error reported below is the average of 18 different geometry cache animations of 45 frames each. The "largest error" column represents the geometry cache animation with the highest RMS error value.

Table 1: Accuracy of the 18 geometry caches in our blood library

type	average RMS error	largest error
Half-precision	0.009157	0.025724
Mortal Kombat 11	0.029941	0.089666

3.2 Use cases

We will also show multiple use cases for geometry caches, the art production techniques used, and the design considerations made to achieve these effects.

Our Alembic art pipeline also supports importing point caches of attributes such as position, velocity, color, size, and lifetime. These point caches can be imported into our engine and the attributes are used to drive traditional sprite particle systems to supplement our effects.

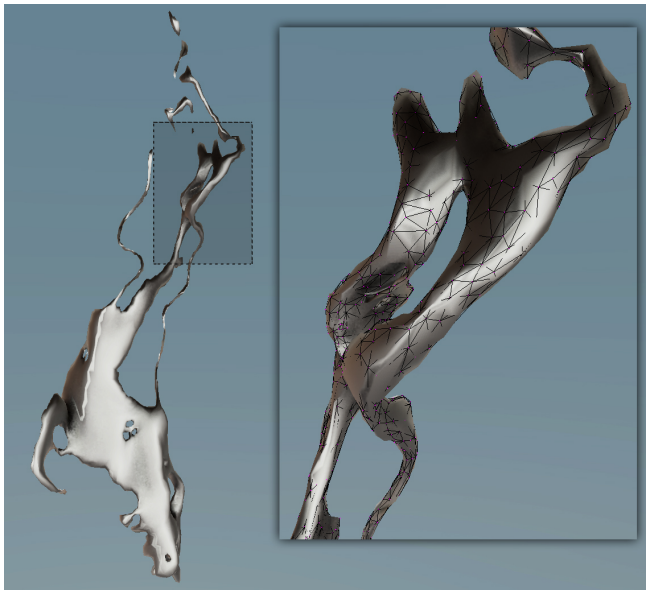

Figure 2: Comparison between source mesh with full-precision positions drawn in wireframe and Mortal Kombat 11's quantized geometry cache shaded in gray.

4 FUTURE WORK

Currently the fluid based geometry caches do not support interpolating animation between frames and motion blur. Both of these problems stem from the fact that each frame of the geometry cache is a discrete mesh and there is no correlation between the triangles frame to frame. Future work to alleviate these issues would be to store the animation as ellipsoids that approximate the fluid's surface and in real-time reconstruct the fluid in screen space. [Green 2008; Yu and Turk 2013]

5 CONCLUSION

In summary, we show that mesh-based animations provide better workflows for artists and result in higher quality effects than the alternatives available to games. We were able to address the memory limitations of mesh animations and illustrate how they can realistically be put into production. Solving these challenges allowed our artists to create the realistic, gruesome blood that is the defining feature of Mortal Kombat.

REFERENCES

Christopher Evans, Lars Martinsson, and Sascha Herfort. 2014. Building an empire: asset production in Ryse. ACM SIGGRAPH 2014 Courses.

Simon Green. 2008. Particle-based Fluid Simulation. http://developer.download.nvidia.com/presentations/2008/GDC/GDC08_ParticleFluids.pdf.

Jihun Yu and Greg Turk. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics* 32, 5 (Jan. 2013). https://doi.org/10.1145/2421636.2421641