Bats, Birds, and Boggans: The Simulated Armies of *Epic*

Thierry Dervieux-Lecocq, David Gatenby, Mark Adams, and Justin Bisceglio

Blue Sky Studios*

Figure 1: *Crowd Shots, Epic © 2013 Twentieth Century Fox Film Corporation. All Rights Reserved.*

1 Introduction

In *Epic* (2013), crowds are integral to the narrative and form a character as a whole. This required a new type of crowd at Blue Sky Studios, one that permits dynamic interaction between crowd characters and the environments around them in addition to supporting the high-resolution geometry with fur, deformation rigs, and material complexities needed for shots where the crowd is close to camera. Our crowd framework centers around the choice to separate the simulation process from the technique used to render the crowd. This meant we could use different simulators for different shots. At times, the crowd exceeded 100,000 characters, far more than in any of our previous films. To manage all this data we store only per character joint animation instead of deformed geometry. This compact format allows us to both display art direct-able representations of the crowd in real-time and to defer evaluation of the expensive parts of the rig until render time. To render the crowd with our in-house ray-tracing renderer, *CGIStudio™*, we build a custom, optimized deformation system that supports rendering of both deformed geometry and deformed voxels.

2 Up, Up, and Away

Sixty percent of the simulated crowd shots in *Epic* featured characters that walked, ran, and climbed on the ground or on walls. We turned to Massive exclusively for simulation on these shots to take advantage of its brain editing toolset for controlling the motion and behavior of individual agents. The other forty percent of shots consisted of highly choreographed flocking bats, grackles, and hummingbirds with riders on their backs. For these crowds we decided on a hybrid approach using Houdini for the main flocking motion and finishing in Massive to overlay animation cycles and secondary motion. Hummingbirds and bats each have their own unique, stylized way of flying. To simulate the specialized flight patterns of our animals we created an expanded version of the "Boids" flocking model in a Houdini DOP network. We modeled the erratic flight of bats using noisy figure eight cycles and for the hummingbirds overlaid sharp lateral xy-plane bursts. Once the motion and flow of each flock was approved we would import this data into Massive to control the agents using a custom API plug-in. Through this plugin we could send arbitrary attributes created in Houdini to act as inputs to the Massive agent's brain. This allowed us to easily trigger actions, choose weapon types, or add secondary motion without having to create complex new brains modules inside Massive.

3 Creating, Viewing, and Editing

Having developed our own render-time deformation system, we could save just the skeletal joint animation and not create geometry caches at simulation time. This decreased the file footprint on disk while also increasing display and simulation speeds. To accurately visualize the crowd during post-simulation editing and director approval, we created a real-time crowd viewing engine. Because many departments at Blue Sky use Maya, we incorporated our viewing engine into Maya's existing viewport so the crowds could be seen in context with the rest of the scene and available for other departments, without needing to render. Various geometry resolutions were available to the user, each displaying accurate material representation. This allowed the directors to evaluate and make notes confidently knowing what they are seeing will hold true through render. The engine also provides a GUI that allows for art directing the crowd, including changing character and garment variation, LOD, and transformations. Most editable features of the GUI are also provided as render-time handles. The engine uses OpenGL vertex buffer objects for each character that are transformed using the simulation data. For a crowd of 100,000, the data is about 1,000 times smaller than if we used a geo-cache, making it easy to fit in the GPU.

4 Render Optimization

In rendering shots with crowds, taking an aggressive approach to minimizing the storage of geometric data had handsome payoffs. Rather than storing geometry for every deformed crowd character on disk, we developed a rendering system in *CGIStudio™* where only the skeletal animation and key-framed attribute data was required. Each crowd character species shares one rig and each variation in a species shares one mesh. For each crowd character, individual animation is supplied to the rig and evaluation of the rig generates a uniquely deformed mesh. We found that reading this compact animation data from disk combined with evaluating a rig for each character was faster than reading cached geometric point data from disk by a ratio of more than 1000:1. We built support for rendering the crowd characters as either geometry or as voxels. Using a sparse voxel octree, our voxel data requires less memory than geometry. Moreover, when rendering the crowd as voxels we are able to entirely eliminate geometric data from main memory while ray-tracking. This is because our voxel data can be deformed directly by the rig. Overall, this integrated render-time deformation system allowed us to streamline both the rendering and the pipeline of our crowd assets for *Epic*.

*E-mail: {thierry,davidg,marka,justinb}@blueskystudios.com

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07