How to Build a Sixty Foot Man of Moving Sand

Jamie Pilgrim* Terrance Tornberg
Sony Pictures Imageworks

1 Introduction

In the final battle sequence of *Spider-Man 3*, the Sandman character forms himself at a construction site. Pulling foreign materials such as dirt and construction debris into himself, he forms into a monstrous, 60 foot tall version of himself which we referred to simply as "Big." Big presented a number of challenges separate from the other incarnations of the character in the film.

Big was far too large to form from particles as was done in other shots, and yet it needed to be clear that Big was made of moving sand which fed him from the ground up. Furthermore, due to script changes, we only had a few months available in which we needed to produce some 40 shots of the character. Big needed to be efficient. As soon as the look was approved, it needed to be ready to run for the entire sequence.

2 Playing with Sand

In order to make Big's surface appear to be made out of moving sand, we tried a number of tricks such as animating the shaders and adding isolated streams of sand on his body. However, neither of these approaches resulted in the desired complexity or sense of motion. Instead, we had to resort to the brute force approach of scattering sand over Big's entire body and simulating it flowing via a Houdini particle simulation.

We created a static rest position in which to simulate particle flows. Since the sand needed to generally flow up his body and then down his arms, we created the rest position such that Big was standing upright with his arms straight up. Using this rest position, as long as the particles were generally going up, they would flow up his body and down his arms. Once things were simulated on the rest position, they could be transferred to the animated model for the shot by calculating and storing the barycentric coordinates of the nearest triangle on the rest pose, and then transforming to those barycentric coordinates for the corresponding triangle upon the animated version. This also let us easily reuse simulations across several different shots.

Although there was a surface beneath the layer of flowing particles, the sheer scale of Big required us to simulate on the order of 2 to 6 million particles, which was boosted further by a factor of 20 at render time using our sand instancing rendering system.

Simulating 2-6 million particles flowing over a surface was a challenge in itself. Keeping the particles constrained to the surface proved to be the most costly part of the simulation. We decided to parallelize the work by saving that costly task until after the simulation was completed. A 3-dimensional vector field was created around the rest pose. The vectors were calculated such that they represented the up vector projected into the tangent plane of the nearest surface point. This created a flow field mimicking the gross contours of Big's body. The particles were then advected through this vector field in what was an extremely fast operation on the order of 10 seconds per frame for 2 million particles. Rather than trying to be exact, the advection merely kept the particles moving close to the surface. After this simulation was stored to disk, each frame could be read independently, snapped exactly to the surface,

Figure 1: Big Sandman stares down Spider-Man in the final battle sequence of *Spider-Man 3*.

and saved back to disk. This allowed us to run the most costly part on a farm of processors - one processor per frame. In this way, we were able to drop the time required to see a 2 million particle simulation for a 300 frame shot from 50 hours to just over 1 hour.

We found that using our physically correct sand grain scale appeared too fine at Big Sandman's size. It also made it difficult to read the flowing motion along the surface. We decided to vary the size and introduce more pebbles and rocks into the sand, which not only felt more interesting but helped sell the story point that he was always pulling fresh sand in from the rocky construction pit below. While most of our sand was rendered as RenderMan points, the larger pebbles used displaced curves that were assigned a random map in the shader to wrap them around and give us a very good approximation of geometry. Only the very largest chunks of packed sand were rendered as geometry. The sand instancer was able to blend relatively seamlessly between each level of detail and the shot lighter had control over how high they needed to push the level of detail in order to find the proper balance between quality and render efficiency.

Big was constantly dripping sand, which was also simulated in Houdini. Some of this sand fell from all over the body randomly, and some of it was jostled loose by the movement of cinder blocks, broken truck parts, and other debris caught in the flow of sand over his body. For the latter case, we computed the intersection of the debris geometry with the body and emitted sand from those regions. Levelsets and vector fields representing Big's shape and motion were computed for each frame. These fields were then used in the simulations to quickly calculate whether a sand particle had collided with the body (via a simple lookup into the signed distance function) and what the collsion response should be (based on the motion encoded in the vector field).

3 Conclusion

In the end, we were able to produce a system which allowed the artists to simulate massive quantities of particles, and to perform several iterations per day. This allowed the effects artists to finish nearly all of the sequence in a little over a month.

*e-mail: {jamie,terrance}@imageworks.com