1 Additional Notes on Simulation

There were several other layers of simulated sand, rocks, and debris in addition to the flowing sand. Chunks of hard packed sand and bits of debris were also simulated as particles and made to flow along the surface in a similar manner to the sand. At render time, the sand instancer placed modeled objects stored as rib archives centered at the particles. Orientations for these pieces were calculated relative to a surface coordinate frame using the surface normal and tangents so that as an object moved over the surface it would rotate accordingly.

For the large pieces of debris such as the parts of the demolished dump truck, we went one step further and tracked the movement of these objects over Big’s surface. The frame by frame positions were transformed into UV texture space and stamped with circles containing an exponential falloff in value from white in the center to black at the edges. Positions older than the current frame had less width and less value. This map was used to animate self-healing gouges in the surface as the debris traveled over it.

2 Additional Notes on Lighting

The primary challenge with lighting Big was to keep his look consistent with Sandman’s previous iterations while at the same time making him even more monstrous than before. Specifically, even though he had additional construction material and debris in his body, he still had to feel like sand. We were able to make use of our existing sand pipeline that had been developed throughout the course of the show, which meant that FX artists and lighters would already be familiar with the tools and methodology. Regardless of whether or not sand was being rendered as a surface or particles, it always relied on the same shader to keep both the look and the lighting model consistent. When shading the surface of Big, a procedural noise function was used to simulate a finely packed body of sand.

Since sand could exist as either particles or procedurally textured surface, we needed a way for these two representations to match. As particles flowed over the surface, the particles inherited texture coordinates and spatial reference coordinate frames inherited from the polygonal mesh using barycentric interpolation of the vertex attributes of the closest triangle.

To add visual complexity to our creature, we relied on a combination of both procedural displacements as well traditionally painted texture maps. Because particles were able to inherit texture coordinates of the underlying surface, sand flowing across the surface or into cracks could then pick up the painted color or surface displacement accordingly.

In order to push these complex and heavy data sets through our render farm, Big was divided into multiple layers. Each beauty render generated a motion blurred deepshadow which would then be fed into the next layer as a holdout. Heavy particle layers needed to be tiled, or subdivided in screen space, then stitched back together. The sand instancer used the tiled Renderman cropwindow to cull any particles that weren’t currently being rendered. This allowed us to push through renders that would have otherwise had enormous memory requirements.

Our lighting setup relied primarily on traditional spot lights. The shading model relied on a camera biased normal in order to keep the shading consistent between the body and any sand falling or sheeting off the surface. This meant that very little environment light could be used since there was no proper direction to use as a lookup to an HDRI map. Lighters were able to mix back in some of the original surface normal to try and preserve detail. Strong rim lights and heavy under lighting were used to help sell the sense of scale.
(a) A still from early in the sequence as sandman is being created from a pile of debris.

(b) Sand and debris constantly drip off of sandman as he animates.

(c) Close-up as Big reaches for Harry.