## Countless Characters and Clovers: Interpreting Dr. Seuss' Style with 3D Fur

<table>
<thead>
<tr>
<th>Eric Maurer</th>
<th>Sean Palmer</th>
<th>Alen Lai</th>
<th>Jamie Williams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Sky Studios</td>
<td>Blue Sky Studios</td>
<td>Blue Sky Studios</td>
<td>Blue Sky Studios</td>
</tr>
<tr>
<td><a href="mailto:ericm@blueskystudios.com">ericm@blueskystudios.com</a></td>
<td><a href="mailto:palmer@blueskystudios.com">palmer@blueskystudios.com</a></td>
<td><a href="mailto:alen@blueskystudios.com">alen@blueskystudios.com</a></td>
<td><a href="mailto:jamie@blueskystudios.com">jamie@blueskystudios.com</a></td>
</tr>
</tbody>
</table>

Translating Dr. Seuss' *Horton Hears a Who!* from the printed page to an animated feature presented stylistic and technical challenges for Blue Sky's ray-traced voxelized fur technology. The directors embraced our fur tools to interpret Seuss' brush strokes, challenging us to address the scope of our furriest feature film to date. We furred and feathered jungle animals, groomed clothing for Whos in Whoville, emulated Seuss’ swooshes for trees and bushes, and created landscapes with windswept grasses and clovers.

### basic fur tools

We procedurally generate each individual hair, point by point, allowing explicit articulation and contrasting with other systems which interpolate guide hairs. Samples are generated on host geometry and pruned according to minimum distances and probability signals. Curves are drawn at each sample from intrically defined world space vector fields and transformed into surface space. Our grooming interface is analogous to modular analog synthesis: raw signals are generated, mixed, modulated and filtered creating alternately harmonious and dissonant results.

Fur data is written to files, which allows us to create elaborate descriptions without concern for time constraints, since the networks are only solved during the design cycle, and the final results are stored in a groom file. Animation of character and prop fur is achieved using a proprietary follow-through algorithm that produces offsets for the static fur that are also written to files. Fur is rendered by accumulating curves of position, transparency, color, and other data into a voxel space. The system represents complex organic forms with less memory and aliasing issues than geometry.

### furry crowds of Whos

The Whos were designed to be covered in fur which is groomed into clothing. To shape otherwise organic fur grooms into contrived clothing we developed rectilinear vector fields akin to the weft and warp of the implied weave of the cloth. Similar tools described elaborate hairdos using arrays of curves to articulate the basic contours as well as braiding, twisting and clumping. Crowd sequences with thousands of Whos presented new problems since an average Who took ten minutes to build two million hairs, five thousand would have taken over eight hundred hours to build ten billion hairs. We solved this by creating 'cheap' grooms with as little as one percent of the original hairs. We pre-processed crowd shots to determine a character’s maximum screen space resolution in order to decide which fur files to load. Subsequently thousands of characters were built in hours instead of days.

### fields of clovers and LOD

The scope of vast furry landscapes proved to be impossible for grooming and storing every single hair. While an average character contained 2-8 million hairs, even using only 1,000 hairs to represent each clover would put the field's hair count at several billion. File storage and i/o times were prohibitive. The solution was to groom a single hair representing each clover's stem and write these to disk. This gave artistic control of spacing, height and placement and could be reused in every shot. Using these as guides, clovers were built procedurally, generating hairs without the intermediate storage. This dynamic building allowed near-limitless amounts of fur.

Variable hair counts for clovers were created based on the camera's position and built up or down as the camera moved. A level of detail (LOD) method managed these transitions. The screen area of each clover determined the number of hairs needed to create a full look at that distance. Our system could handle extreme camera moves that would have been prohibitive with a predetermined clover resolution. Dynamically building from 5 to 50,000 hairs, the resultant clovers had enough detail up close while reducing generation time significantly at a distance. The final number of hairs per shot varied from a few thousand to 2.5 billion. This flexible solution became part of an extensible tool set used to create generators for dandelions, grass and particle systems.

### bridging fur to effects

The effects department leveraged on this work with the dandelions, the first effect in production requiring procedurally animated fur. We modified the generators to use world space transforms, derived from particle motion, as offsets to static fur from files, allowing for more dynamic uses of voxels. To direct clover piles eroding, we sampled the source positions of the clovers and stored relative height and depth signals to direct motion and interaction with characters. Our follow-through technology was adapted to create procedural wind on the field. An important consideration in integrating the effects and fur pipelines was to avoid redundant efforts. A hybrid system that used an fx node network design on the surface and called fur scripts within bridged the two systems.

From millions of hairs in Ice Age 2 to billions in Horton, our fur technology evolved to realize Seuss’ vision. With an effective inter-department pipeline, it is now possible to design new tools and creative controls for a range of voxel applications at Blue Sky.