From the beginning, *Horton Hears a Who* tested the limits of our fur pipeline. Developed for character work in Ice Age: The Meltdown, it was now being used for props, plants and entire sets. In one significant moment of the film, Horton searches through a field of clover flowers described as “a hundred miles wide.” Seuss used repetition of a single, simple form to give a sense of awe at the scale of the field and the task Horton faces. We had no existing solutions to address the volume and scope of shots needed for the field. Faithfully translating Seuss’ depiction into 3D was both an inspiring and intimidating challenge.

**The Fur Work-Flow**

We found early on that our fur system emulated Seuss’ iconic illustration style of fluid, repeated brush strokes. The “hero” clover Horton carries in the film uses it for this reason. Our fur is rendered by accumulating curves of position, transparency, color, and other data into a voxel space [van Swaaij 2006]. Voxel solutions are adept at representing complex and vast organic environments with less memory and aliasing problems than geometry [Neyret 1998]. For stylistic and technical reasons, therefore, the fur system was the chosen technology for the field.

Fur data is created by a grooming process that gives control to the artist to decide placement of hairs. We initially used our groom tools to build the entire field as though it was a large animal. An average character contains 2-8 million hairs (the hero clover itself had about 800,000) and memory limits in Maya 7 prevented more than 15 million hairs per groom. The problem was that even a reduced clover, using only 1,000 curves to represent the stem and flower, would still put the field's total hair count at several billion. Even with more RAM available, file storage and i/o times would be prohibitive. Worse, many shots would show these clovers next to hero clovers, so they needed to match in look and complexity.

**The Clover Generator and LOD**

Procedurally generating data is a common process in CG, but our system needed to create millions of clovers while keeping them manageable in expansive shots but sufficiently detailed when seen up close. The solution was to groom a single hair representing each clover's stem and write these to disk. This gave artistic control of spacing and placement and could be reused in every shot. Using these hairs as guides, much like other fur systems, clovers could be built procedurally at render time, generating hairs without the intermediate storage. Our voxel system is designed to accept any source of fur generation, so we developed C++ plug-ins to create the additional fur data. Dynamically building clovers used less resources and allowed near-limitless amounts of fur. The focus was now to build the field as efficiently as possible.

Fur is voxelized per frame if it, or the camera, moves. In exchange for this expense, variable builds for each clover could be created based on the camera's position. Voxel systems using hierarchical subdivisions can track rays to reduced resolutions depending on the detail required, but they must first build the voxel space to a predetermined maximum resolution [Neyret 1998]. Using a table of vector data, we could create unique clovers per frame, rather than instancing texels, and have them build up or down as the camera moved. A level of detail (LOD) method was developed to manage these transitions. LOD is traditionally used to reduce polygonal geometry relative to camera distance. Using a similar concept, the screen area of each clover determines the number of hairs needed to create a “full” look at that distance. Fading on groups of hairs as a clover's area increases gives the different detail levels. Our LOD system could handle extreme camera moves that would have been prohibitive with a preset maximum voxel/clover resolution. Dynamically building from 5 to 50,000 hairs, the resultant clovers had enough detail up close to match the “hero” look while reducing generation time significantly at a distance.

Custom classes gave controls for wind on the field and allowed transform data to drive clovers with particle simulations. Additional flexibility came from a method to evaluate procedural materials within the generator. A signal applied to the ground could change the state of each clover growing on it, creating picked stems, plucked clovers, and empty patches. This facilitated accents such as the pick-line Horton creates as he searches the field and debris around each clover pile. Changes were made to the procedure without modifying the original field groom.

**Results and Future Work**

The final field contained guides for around 500 million potential clovers. Because of LOD, the number of hairs generated per shot varied, from only a few thousand to 2.5 billion. Using this quantity of data with the voxel system revealed visual artifacts including discontinuities in specular reflection as distant voxels' normal data averaged. We are addressing this and working to reduce computation times, but in all the generator proved to be a flexible and robust solution to a demanding problem. It has become part of an extensible tool set used to create other plug-ins, making it possible to use the code in generators for dandelions, grass and particle systems. From millions of hairs in Ice Age 2 to billions in Horton, Blue Sky's fur technology evolved to realize Seuss' vision.

**References**
