500 Million and Counting: Hair Rendering on *Ratatouille*

Pixar Technical Memo #07-09

David Ryu[*]
Pixar Animation Studios

Figure 1: A scene with hundreds of millions of hairs. The optimizations described in this sketch reduced peak memory consumption and rendertime both by more than 90% over the hair system used on *The Incredibles* and *Cars*.©Disney / Pixar. All rights reserved.

Featuring plush rats, well-groomed humans, and a colony of rodents numbering a thousand strong, *Ratatouille* had shots where the original scene descriptions contained many hundreds of millions of hairs. To make these shots renderable, we developed many new technologies to optimize our RenderMan-based hair rendering pipeline, including caching to speed up runtime sculpting, a technique for optimizing bounding boxes for RiProcedurals, and a smooth level-of-detail system incorporating depth-of-field and motion blur. These techniques resulted in an order of magnitude reduction in hair rendering time and memory compared to the already optimized pipeline used previously in Pixar films.

1 Hair Rendering

Pixar's hair rendering system, called gofur (developed for *Monsters, Inc.*), is a set of RiProcedural plugins designed to sculpt and efficiently feed RiCurve geometry to PhotoRealistic RenderMan (prman). It uses bounded procedurals to delay geometry specification until absolutely needed by the renderer. This provides a huge memory reduction over simply specifying the geometry in the input rib file. However, scaling gofur up to the needs of Ratatouille required many new technologies.

2 Optimizations

Stochastic Simplification A key technique was Stochastic Simplification[Cook et al.], a level of detail scheme involving smoothly reducing hair density and increasing width based on factors including size on screen, motion blur, and depth of field. The size measure involved comparing the "detail" parameter from prman with the size in world-space of the element being rendered. For motion blur, we came up with a measure that encapsulated both amount of blur and the distance from screen by projecting the world-space motion vector onto the plane perpendicular to the vector from the eye to the moving point. We used a corresponding measure for depth of field, calculating the size of the circle of confusion projected onto the plane containing the blurred point. Stochastic Simplification provided drastic memory and time reductions, and enabled us to use a single model representation for both foreground and background rats. In this sketch, we will present implementation, deployment, and approval details not present in the paper.

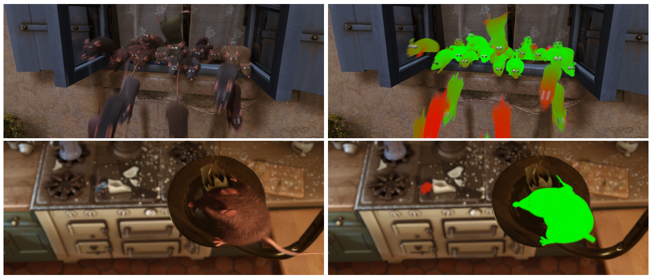

Figure 2: Motion-blur and depth-of-field measures. Red means more simplification.©Disney / Pixar. All rights reserved.

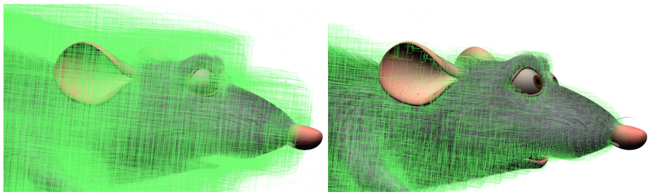

Figure 3: Perfect bounding optimizes RiProcedural bounding boxes (shown in green).©Disney / Pixar. All rights reserved.

Operator Caching gofur's hair generation has render-time procedural components (called operators) for actions like scaling, clumping, and curling. Many of these operators were fixed after completion of the groom. By caching offset vectors for the fixed operators to a file, we could bypass those expensive computations and greatly reduce hair render times.

Perfect Bounding Using RiProcedurals requires specification of a bounding box; accurate bounds greatly reduce the memory required to render the scene. Calculating even decent bounds can be difficult because of procedural sculpting (consider a scale based on low-frequency noise), animation, and simulation. Perfect bounding addresses this with a 2-pass approach. First, a proxy procedural is emitted with a conservative bound, which calculates the hair geometry but doesn't actually give it to the renderer. Instead, it emits a procedural with the exact bound it calculated, which will render the hair. This time/space tradeoff was worthwhile because operator caching made calculating hair geometry very fast, and resulted in large memory wins at a minimal cost in time, and also freed the TDs from having to endlessly tweak bounding box sizes.

Miscellanea We also found smaller tricks that made an impact: converting vertex variables (VVs) on the hair to their most compact representation before emitting geometry; culling shading related VV data in shadowmaps; sharing mesh structures for characters with identical topologies; taking shutter settings into account when computing motion blurred bounding boxes; backface culling; curve basis reduction.

References

COOK, R., HALSTEAD, J., PLANCK, M., AND RYU, D. Stochastic simplification of aggregate detail. In *Proceedings of SIGGRAPH 2007 (to appear)*.

[*] e-mail: ryu@pixar.com