1 Introduction

Because Fido has a strong focus on character and creature animations, a robust and fast workflow for our Hair and Fur pipeline is essential for our daily work. With our previous off-the-shelf Hair and Fur system we had experienced a lot of problems with storage utilization and network traffic when we pushed it to its limits.

Our goal in creating a new Hair and Fur system was to enable our artists to work procedurally in real-time with grooming, shading and rendering and at the same time minimize the amount of data that is saved to disk. To fulfill these criteria we are developing SpeedFur, a node-based, procedural hair system integrated as a plug-in for Maya. SpeedFur is capable of rendering complex hair styles on both the GPU in real-time and on the CPU for offline rendering on our render farm, and its procedural approach keeps any disk based files and caches to a minimum.

2 Architecture

To give our technical artists as much flexibility and creative freedom as possible we are building SpeedFur as a procedural node-based network. All nodes share a common code-base and new features can easily be added, making our system extensible and easy to maintain. The nodes can change the strands in arbitrary ways and can even grow new hair strands or remove hair, giving us huge flexibility when developing new nodes.

For offline rendering, SpeedFur is evaluated as a procedural primitive in RenderMan which significantly reduces the stress on our network infrastructure.

Fido currently uses a traditional CPU-based renderfarm solution. To eliminate the need for larger investments in specialized hardware, SpeedFur is designed to run on both the CPU and the GPU. In order to achieve this, we compile the code for both platforms, thus ensuring the geometry in the real-time preview is identical to one in the offline rendering.

3 Workflow

Guide curves are used to setup the main hair layout and control the direction of the hair. The procedural node network can then be used to interpolate and style the hair using a combination of our highly specialized node types (frizz, clumping, length, noise, etc.). Kim and Neumann [2002] used generalized cylinders specified with different functions which change the style at various detail levels. Using our approach, we can achieve a similar effect by using specialized nodes for each function, affecting the hair at any specific point in the node network.

Each node type exposes a set of parameters and the user can even utilize many of Maya’s built in tools, e.g. texture painting or dynamics, to control the result of a specific node.

One of the major advantages of SpeedFur is the ability to give fast and accurate feedback to the artist by evaluating the procedural hair network on the GPU. This allows for very fast iterations and enables our artists to reach the desired result in a fraction of the time previously spent waiting for offline renders.

4 GPU

Once the guide curves are moved to the graphics memory, the node network can be fully evaluated using NVIDIA’s CUDA API, and then rendered in the viewport using OpenGL without moving any data back and forth to the main memory. Thus a very large number of interpolated hair strands can be rendered in real-time as the artist is changing the different node parameters. Since OpenGL is used for the preview, we created a GLSL hair shader based on the Marschner shading model [Marschner et al. 2003] to create a more realistic preview.

Sometimes an artist’s hair system has more hair then could be currently kept in memory on a commodity GPU, but we enabled the user to filter parts of the hair system so they can view only segments of the fur in full detail. Alternatively, the user can control the preview fur density to be lower than the render density to be able to view the full hair style but with lower fidelity.

5 Results

Thanks to SpeedFur’s procedural approach, we have been able to minimize the impact on our infrastructure when rendering, and at the same time reduce the time needed for look development utilizing our GPU accelerated real-time preview. SpeedFur’s integration with Maya means that we are able to do the shading, lighting, rendering and editing of hundreds of thousands of shaded hair strands in real-time, in one easy workflow. The support for GPU rendering means that our artists can also keep their existing workflow using Renderman and that setups done in our previous hair and fur system can be carried over to SpeedFur.

References
