# Grooming a lion for *Hercules*

Francesco Giordana* Sarah Macdonald† Gianluca Vatinno‡
Double Negative Double Negative Double Negative

## 1 Introduction

The film *Hercules*, with its photorealistic furry creatures, really put Dneg's creature pipeline to the test so we developed a brand new software, *Furball*, which combines the computing power of the latest GPU technology with a flexible and procedural workflow.

While most studios use the GPU in limited parts of their fur pipeline, predominantly for real-time rendering, we decided to make massively parallel computation the core of our new framework. Our solution is based on a hybrid CPU/GPU implementation, that lets us exploit the benefits of both architectures.

In this talk we discuss the major advantages and problems of this approach applied to a challenging film production and suggest that fur simulation is a good domain for the application of modern GPU technology.

**Figure 1:** *The lion's groom*

## 2 A hybrid CPU/GPU node-based framework

*Furball* is a fully procedural node based framework, where each node defines a specific operation to be applied to the curves flowing through a node graph, like many other fur systems presented in the past. What is unique about our implementation is that each node can have a GPU accelerated implementation and the framework is capable of falling back automatically to CPU computation in case of errors at any point in the graph. In production, this proved to be a vital feature, since the resources on the graphics card are quite limited and the fur system would always have to compete for them with many other applications.

Being able to target a specific architecture at a node level allowed us to better exploit the advantages of each one by leaving us the freedom to run certain operations exclusively on the CPU in a serialized way. In addition, the automatic fallback to CPU-only computation allowed us to execute the same graphs on a render farm, where graphics cards are not available.

To ensure that the final rendering output always matches the in-viewport previews, we need the CPU and GPU computations to consistently produce exactly the same results. This requirement forced us to structure both the CPU and GPU code in a very similar manner, dealing with problematic areas like random number generation, which don't translate well to a parallel approach, exclusively on the CPU.

A big limitation in hybrid CPU/GPU systems is the overhead added by transferring data from the CPU to the GPU and back. We designed a transparent but very efficient way of handling data transfers: each data object flowing through the graph has a dual implementation, one for the CPU and one of the GPU. Only one exists at any given time and the conversion is triggered by the framework only when we compute on a different architecture. As an added benefit, cleaning up allocated resources on the GPU, after the computation is finished, becomes trivial.

## 3 Results

For *Hercules* we were able to create a lion, a boar, a cerberus and three wolves, all photorealistic and featured in close-up shots, with complex dynamic interactions. The lion alone was covered by more than 10 million individual hairs, all simulated and rendered at very high resolutions. The use of the GPU allowed us to increase the computation speed by a factor of 10x on many operations (compared to a multithreaded CPU implementation on a 12-core PC), allowing the artists to tweak the look of the groom in real-time.

The one area where computation is still relatively slow is physics simulation, which was performed in Houdini on a selected set of guide curves and then remapped onto the groom via interpolation.

## 4 Conclusion

While the use of procedural networks for the creation of visual effects is not something new, we believe that we offer an interesting insight on how the advent of GPU technology can help in the generation and simulation of digital fur for film production. We believe that fur is a particularly well suited domain for parallel computation, since each fur network is composed of many similar and relatively uniform curve primitives, which can in most cases be processed independently.

## References

FAGNOU, D., AND LEANING, J. 2010. Furtility: Dynamic grooming for wolfman. In *ACM SIGGRAPH 2010 Talks*, ACM, New York, NY, USA, SIGGRAPH '10, 51:1–51:1.

GIORDANA, F., 2013. A GPU-Accelerated Node Based Framework for Hair Simulation and Rendering. GPU Technology Conference 2013 Talks.

*e-mail:fg@dneg.com
†e-mail:svm@dneg.com
‡e-mail:gv@dneg.com


SIGGRAPH 2014, August 10 – 14, 2014, Vancouver, British Columbia, Canada.

ACM 978-1-4503-2960-6/14/08