A Very Good Kitty, Indeed

DreamWorks Animation's *Puss in Boots* Uses Intel® Math Kernel Library to Help Create Dazzling Special Effects

BY GARRET ROMAINE

THE WORKING RELATIONSHIP BETWEEN INTEL AND DREAMWORKS ANIMATION is focused on both supplying powerful hardware systems and developing elegant software solutions that tee up the creative product. Both sides of this relationship add to the amazing visual quality of a DreamWorks Animation production, and its latest animated blockbuster, *Puss in Boots*, which to date has grossed over USD 480 million at the worldwide box office, is a case in point—especially in 3D.

Sure, the Intel® multi-core processors in the servers and workstations were crucial in supporting 69 million hours of rendering time: an unprecedented number for a DreamWorks Animation production. But to be effective those big, muscular systems need smart, problem-solving software. The studio's technology teams worked closely with Intel engineers to develop effective, proprietary applications to take advantage of the Intel® hardware. The result was a new level of richness in an animated film that has garnered immense critical acclaim and commercial success.

The strategic alliance between Intel and DreamWorks Animation began in 2008 and continues to evolve. The engineering team at DreamWorks Animation has created state-of-the-art applications that help deliver better graphics and faster rendering, and aid in the completion of other tasks. With encouragement from the company's unique group of physicists grounded in fluid dynamics and aeronautical engineering, Intel application engineers worked hard to optimize those applications using a tool that doesn't normally get a lot of fanfare: Intel® Math Kernel Library (Intel® MKL). Intel MKL is a key collection of pre-written code and subroutines, and it's one of the crucial tools behind the technology of *Puss in Boots*.

A Very Good Library, Indeed

Intel MKL is a computing math library of highly optimized, extensively threaded math routines for applications that require maximum performance as they perform calculations. Core math functions include linear algebra, fast Fourier transforms (FFTs), vector math, statistics, and more. DreamWorks Animation leveraged Intel MKL to improve the performance of the fluid simulation framework in *Puss in Boots*, which is used heavily to create environmental effects such as smoke, fire, dust, and explosions. Special effects often require artists to set up and manipulate fluid simulations that tend to consume a significant amount of time. Improvements to artists' tool sets for these kinds of simulations can therefore have a positive impact on the overall production schedule.

John O'Neill, application engineer in the Intel® Software and Services Group, worked closely with DreamWorks Animation to optimize their applications and take advantage of Intel® tools. "The key was the MKL Poisson solver they used," O'Neill said. "The team studied the different algorithms and found the FFT-based solution best matched their theoretical expectations for thread scaling. Intel® MKL uses SIMD (Single-Instruction Multiple Data) so that the implementation was highly efficient."

The Fluid Modeling Dream Team

O'Neill's main contact during the production of *Puss in Boots* was Ron Henderson, who runs the effects, character effects, and geometry software development teams. Henderson, a

Poisson

Siméon Denis Poisson (1781–1840) was an influential French mathematician and physicist who from an early age dominated the field of mathematics. A prolific writer and lecturer, he published more than 300 works, treatises, and memoirs dealing with mathematics and physics. His mathematical expertise supplied important advances in planetary theory and celestial mechanics, provided mathematical foundations for the theory of electricity and magnetism, and contributed to the theory of attraction.

Most important for modern computer-generated animation, Poisson made important advances in the study of the Fourier series, involving linear differential equations with constant coefficients. Using probability theory and statistics, the Poisson distribution (also known as the Poisson law of small numbers) is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space. The Poisson distribution is most commonly used to explain the timing of rare events that take place randomly but "uniformly" over time. The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area, or volume, and is extremely useful for realistically modeling fluids such as smoke and clouds. Perhaps the only stain on Poisson's impressive body of work is his insistence on explaining light as particle-based, rather than wave-based, as we now know it. Still, his contributions to science were immense, and the Poisson solver is an important component of Intel® Math Kernel Library.

"It was a joy collaborating with DreamWorks Animation to enhance the Poisson solver in Intel® Math Kernel Library. Animation developers can now benefit from the work we did together."

— SERGEY GOLOLOBOV, ENGINEERING MANAGER, INTEL® MATH KERNEL LIBRARY

senior manager in Research and Development for DreamWorks Animation, has a PhD in Mechanical and Aerospace engineering from Princeton University and spent five years on the research faculty at the California Institute of Technology. Back then, he specialized in physics, high-performance computing, and simulation for fluid turbulence; he thought it would be interesting to apply those techniques in the entertainment world.

Ten years later, Henderson and his team add stunning visualization to hair, feathers, fluids, and explosions within DreamWorks Animation films. They often build proprietary animation tools from the ground up, but they also extend existing libraries and third-party tools. Henderson's team also spends a great deal of time explaining what they're doing in layman's terms. "Sometimes we help the artists, who are technical but might not understand the numerics involved," he said. "We help them analyze how they need to think about these problems and how they can try to control them."

In addition to computer science, Henderson's team has a diverse academic background in fields such as quantum chemistry, physics, and engineering. Henderson added, "Many of us have a deep technical background, and DreamWorks Animation gives us the unique opportunity to use our problem-solving skills and expertise to help create art. The most thrilling thing for me is when an artist takes a tool that we have worked on and does something with it that we had never even imagined was possible. That happens a lot. The environment here is a great fusion of high technology and high art."

Danger—Tornado Warning

Puss in Boots has many amazing scenes—the fast-growing beanstalk, the dreamy clouds, and the whirlpool have all drawn acclaim from critics and audiences across the globe. But for Henderson, one sequence stands out. "My favorite is the sequence where Puss, Kitty, and Humpty plant the magic beans, and a giant tornado descends from the sky just before the beanstalk erupts from the ground. The tornado is animated using a huge fluid simulation, one of the biggest fluid simulations we've ever run at DreamWorks Animation. The tornado is about 116 meters (over 380 feet) tall, and to get the level of detail

Tornado visual development resolutions:100, 150, 200, and 300 dpi (LEFT to RIGHT).

we wanted for the movie we needed to simulate it down to a scale of about 15 cm (about 6 inches). That means we model the volume of the full tornado with about 110 million small volumes, and we solve a set of equations for the fluid motion in each of those small volumes. That is a very large simulation for a single effect, and it is just one element of the full sequence. Many other elements—animated characters, rocks, trees, and ground fog—all need to be rendered and combined for the final effect. Just a few years ago, a single simulated element like the tornado would not have been possible."

Henderson's team used Intel MKL to build a fluid solver that could simulate an effect such as the tornado superfast, relying on the Intel MKL Poisson solver for one of the most expensive computational kernels. "We want solid building blocks that we know will be robust and have optimal performance," Henderson said, "Intel MKL provides that. We can start understanding the artistic benefits of a complex algorithm more quickly if we don't have to build every component of a system from scratch."

DreamWorks Animation and Intel also collaborated on performance optimization and threading for other systems. Henderson's team picked a few core systems, such as liquid simulation, and set up benchmark problems so they could analyze the issues. They met each week with Intel engineers and walked through the performance analysis together, running simulations using tools such as Intel® VTune™ Amplifier XE. "It's a great performance analysis package but it doesn't actually fix any problems," Henderson said, "It presents us with the information, and our team decides what to do with that information." It helps when that team consists of engineers with PhDs, of course, and once Henderson had real numbers for performance analysis, solutions began to emerge.

Puss in Boots: Facts and Figures

- The production of *Puss in Boots* required 69 million render hours and consumed 109 terabytes of disk storage.
- The movie comprises 120,000 (119,626 actual) individual computer-generated frames.
- The DreamWorks Animation render farm spans five geographic sites, and it is the most powerful ever used for a studio production. It consists of almost 22,000 Intel® processor cores.
- *Puss in Boots* hit a one-day peak rendering usage of almost 23,000 (22,783 actual) render units using 16,000 (15,981 actual) cores.
- *Puss in Boots* is the studio's twenty-third animated movie and its sixth in stereoscopic 3D.
- There are 314,918 grains of sand in the hole that Puss and Kitty dig to plant the magic beans.
- The tornado is over 380 feet (116 meters) tall—the height of a 30-story skyscraper.
- The beanstalk shoots out of the ground at a speed of 746 mph (1,200 kilometers per hour), or just 20 mph below the sound barrier.
- The beanstalk grows a total of 6.32 miles (about 10 kilometers) before reaching the clouds.
- Mama Goose is 80 feet (about 24 meters) tall and has feathers that are more than 10 feet (about 3.5 meters) in length.
- When the bridge is destroyed (over a period of 4 shots), the process employs 23 fluid simulations and 96 particle simulations, generating:
  - 1,562 fractures
  - 10,523 rock fragments
  - 22,492,558 dust particles falling off the bridge's bricks
  - 11,922,425 clumps of dirt

"Many of us have a deep technical background, and DreamWorks Animation gives us the unique opportunity to use our problem-solving skills and expertise to help create art. The most thrilling thing for me is when an artist takes a tool that we have worked on and does something with it that we had never even imagined was possible. That happens a lot. The environment here is a great fusion of high technology and high art."

– RON HENDERSON, SENIOR MANAGER, RESEARCH & DEVELOPMENT, DREAMWORKS ANIMATION

A Classic Race Condition

Every DreamWorks Animation feature film aims to push the boundaries of visual complexity to continue delivering to the audience a richer and more dynamic movie-going experience. For simulations, more realism generally means higher resolution. Henderson sees the solution as a function of faster processors, more memory, and new algorithms. "The techniques used for special-effects systems often scale very well with shared memory multi-core processing," he said. This is because many problems in this area amount to applying some algorithm repeatedly to a large data set, like updating the fluid velocity in 100 million volume elements. Henderson and his team found it was relatively straightforward to write new systems that exploit Intel's multi-core architecture using tools such as Intel® Threading Building Blocks and the OpenMP* support in the Intel® Compiler.

Intel MKL is optimized for the latest processors. Henderson believes that every tool his team creates should be optimized for multi-core systems. "The rule of thumb today is that software designs must all have a strategy for parallelization. In today's world, efficient algorithms that cannot be threaded are doomed to be slow. We are looking forward to being able to run on even larger numbers of cores. We like the look of the Intel® Many Integrated Core (Intel® MIC) architecture roadmap and believe we will be able to deploy applications on that architecture in the future."

But even current-generation systems have so much processing power, and allow such large problems to be processed quickly, that the bottleneck is starting to become the amount of memory and storage space needed to deal with simulation data in other parts of the artistic process. "We can continue to allocate additional cores to specific technical challenges," Henderson said. "The goal for the next generation of tools is to find an optimal way to organize and store the data." For example, it is possible to create a high-resolution volumetric model of a cloud using a box divided into a billion small volumes. But if we focus on the volumes on the edges of the cloud, since the rest are invisible, we can write efficient threaded algorithms with good scalability that don't require us to store unnecessary data."

Ultimately, Henderson and his team are working toward finding a balance between representing things the right way and having the computing horsepower to bring it all to life quickly. "We use volumes because they are very flexible, but they consume more memory. We use particle representations because they don't consume much memory, and they are very flexible, but they don't always give you the right level of realism. We could simulate something like a smoke cloud as a particle effect, but it would have a very different look than doing the same thing with a volume. Or we can simulate it as a volume, but it might get very heavy in terms of memory. Coming up with data structures and algorithms that combine the best of the two approaches is the Holy Grail for us in helping to bring the artistic vision of the film to life on-screen." ■

About the Author

Garret Romaine is a senior writer, working for RH+M3 from Beaverton, Oregon. Garret started in gaming as a beta tester for Epic Megagames and has been a columnist, editor, and reviewer ever since. Garret is a Fellow in the Society for Technical Communication, and he teaches technical communication at Portland State University.