

EXPERIENCE**1993-2008: DIRECTOR ENGINEERING AT MENTAL IMAGES, BERLIN, GERMANY**

My primary responsibility as Director Engineering was to lead a team of highly competent researchers and engineers to build production-grade visualization systems with >1 million lines of code each. I was responsible for the software architecture, and to shape and productize advanced research results in a way that it could serve a large professional customer base (millions of seats) over long periods of time (>10 years) without fragmenting into un-maintainable ad-hoc solutions.

In the process, I wrote about 400,000 lines of C/C++ infrastructure code such as scalable multithreaded and distributed databases, networking and rendering code, language parsers and translators, and support libraries. I also designed and built repository and code review frameworks, manuals and reference documentation, test suites, release management, and bug tracking, and managed second-line customer support.

mental ray 1

In the first six months at mental images, I was responsible for creating a translator to embed mental ray 1.7 into SOFTIMAGE 3D. The integration was successfully shown at Siggraph 1993 and became part of SOFTIMAGE 3D. mental ray is the first photorealistic 3D renderer used in industrial and film production houses that could be embedded in OEM products.

mental ray 2/3

It quickly became obvious that mental ray 1 was an “organically grown” system that would not be able to support mental images’ growing business. I proposed and was charged with rewriting mental ray from scratch. I was put in charge of the software engineering department after Siggraph 2003.

- Worked out the requirements from mental images’ strategic plans and customer analysis. Special emphasis was scalability, both for number of CPU cores and number of hosts on an IP network.
- Performed product planning: feature list, modular layered system architecture, software design, coding guides, unit test framework, regression test system, release management, quality control, and documentation.
- Wrote most of the infrastructure code, about 25% of the total, including the virtual shared database, scene storage, image I/O, scene file parser, and the job management system using a load-balancing scheduler for a cache-based storage system.
- A key concern was portability, from small PCs to supercomputers, including 32 and 64 bit architectures, up to 14 platforms including Linux, Windows, Mac OS X, SGI Irix, Sun Solaris, HP/UX, IBM AIX, and others.

Matrix movies

mental images was deeply involved in the rendering technology for the Matrix Reloaded and Matrix Revolutions movies, used by ESC Entertainment to create almost all visual effects shots in these movies. Over a hundred releases were built for ESC. I consulted on-site for several months at ESC to find and implement solutions for ESC’s requirements. This included unprecedented scene sizes and photorealistic indirect light simulations for large city models and ESC’s virtual humans.

Other movies

I have also consulted with other visual effects production houses on a smaller scale, and traveled with the CEO as a technical expert during project negotiations, including Industrial Light & Magic, PDI, Dreamworks, Digital Domain, BUF Compagnie, and others, and was then responsible for designing and helping with the implementation of customizations and extensions.

mental ray was the first commercial 3D renderer that incorporated ray tracing and indirect illumination techniques like photon mapping, caustics, and final gathering. Although I am not a rendering algorithm researcher, I built the software infrastructure to support them, worked out features sets and APIs that solve real customer problems, helped customers to understand, integrate, and use these technologies, and found ways to make the results of academic research usable and robust in a commercial production environment.

Academy Award

The Academy of Motion Pictures And Sciences has awarded me and my team a Technical Achievement Award at the 2003 technical Oscar gala in Beverly Hills, primarily for the work on scalable job management that made it possible to perform complex global illumination simulations on small machines.

Reality Server

mental images' RealityServer product makes high-end large-scale 3D visualization available on thin clients that run only a browser, and connect to a data center that performs complex interactive rendering of very large scenes for a large number of concurrent users. I was responsible for the architecture, design, and engineering of a new major system, with more than a million lines of code.

The main challenge were extreme demands on scalability over thousands of cores, many tens of gigabytes of data, rapid user interaction with very large datasets and high frame rates, and efficient dataflow and load balancing algorithms.

Sale to NVIDIA

I had been a partner and co-owner of mental images since 1994. In November 2007, mental images was acquired by NVIDIA. Subsequently, mental images went through a rapid series of reorganizations.

1988-1993: SOFTWARE MANAGER AT AURORA SYSTEMS, SANTA CLARA, CA, USA

I have joined Aurora Systems first as an intern for five months, then became a software engineer, and then software manager in charge of all software engineers. Aurora System's main product was a paint and 2.5D animation system primarily used for news graphics by CNN and others; there was also a simple 3D component.

When I joined, the system was built on Sun workstations controlling complex proprietary graphics hardware. My task as software manager was to rewrite the system from scratch on commodity hardware, such as SGI workstations and Unix PCs. This product was a layered modular system with strict hardware abstraction for a wide variety of graphics and tablet hardware, GUI and I/O layers, and rendering. This system was named Aurora Liberty and was also used for the Matrix movies, eight years after I left the company.

EDUCATION

Master's degree in Computer Science, Technical University of Berlin, 1987.

SKILLS

- Software architecture and design of complex software systems
- Leading teams of researchers and engineers to create commercial-grade software products
- large scalable multithreaded and networked tightly coupled systems using multicore and multi-CPU machines
- POSIX TCP/IP network socket and thread APIs
- documentation and books with LaTeX
- C, C++, Linux, significant experience with multiplatform portability including Windows

PUBLICATIONS

- *Rendering with mental ray*, Driemeyer; 3 editions, Springer Wien New York, ISBN 3-211-22875-6.
- *Programming mental ray*, Driemeyer, Herken; 3 editions, Springer Wien New York, ISBN3-211-24484-0.

AWARDS

- Academy of Motion Picture Arts and Sciences: "Technical Achievement Award to Thomas Driemeyer and to the Team of Mathematicians, Physicists, and Software Engineers of mental images for their contributions to the Mental Ray rendering software for motion pictures."