

Eric Tabellion

2606 Read Ave - Belmont, CA. 94002 - (415) 609-1359 - eric@tabellion.org

Summary

Graphics software engineer working for over 10 years in the animation industry. Strong software design, C/C++ programming and technical leadership skills. Extensive experience within a large-scale production software development environment. Fast learner, highly motivated doing research and development of innovative graphics techniques.

Experience

PDI/DreamWorks

Redwood City, CA. - 1999 to Present

Senior R&D Staff

Design, write and support core graphics software, used by lighting and FX animators during the production of animated feature films, live-action film-effects and television commercials.

Film credits: Shrek, Sinbad, Shrek 2, Madagascar, Over the Hedge, Flushed Away, Shrek the Third, Bee-Movie.

Rendering: Technical Lead within a group of eight developers

- Participated in the design, development and support of DreamWorks' latest generation high-quality **film renderer**. Based on the REYES micro-polygon rendering architecture, it supports distributed interactive re-shading within a proprietary lighting tool.
- Designed and developed the **global illumination** toolset used to light and render all of DreamWorks' feature films since "Shrek 2". Wrote the light gathering engine and many optimizations, including irradiance caching and importance sampling. Handled the tight integration with existing shaders and implemented shader-specific optimizations. Provided flexible controls enabling art-direction of global illumination effects using the interactive lighting tool. Provided training and support to all the lighters.
- Designed the shader API that allows developers to use **ray-tracing** capabilities in their shaders. Enhanced the core ray-tracing libraries and developed shaders allowing lighters to easily render reflections and refractions. Implemented adaptive sampling and ray differentials for high quality anti-aliasing, as well as Ward-BRDF importance sampling for efficient glossy reflections and refractions.
- Developed **shadow algorithms**, based on shadow-maps percentage closer filtering. Wrote a two-pass approach, able to render adaptive soft shadows, using single layer and multiple layer depth-maps.
- Developed an image post-processing tool, that became the new studio standard for efficiently rendering **motion-blur and depth-of-field**. Researched and developed crack-less post-process motion-blur and depth-of-field algorithms based on line integral convolution. Used vector-field divergence, crack detection and compositing techniques, to eliminate cracks while preserving motion-blurred and out of focus soft edges.
- Enhanced the **particle and volume rendering** tools to enable rendering separate layers. Developed a technique based on motion-blurred cutouts, to achieve compositing with the main rendering layers after color-correction.

Procedural modeling:

- Wrote the set of tools used to render water, mud, beer and milk effects in the feature films "Shrek" and "Shrek 2". Researched and designed a general **surface reconstruction** process, which converts a particle-set animated by a fluid dynamics system, into an animated smooth surface. The latter can be used to render realistic looking fluids. Implemented several computational geometry algorithms: delaunay tetrahedrization, alpha shape extraction, topology processing and triangle mesh smoothing. Oversaw the toolset integration in the production pipeline.

Image processing and visualization:

- Developed an interactive image files playback tool, handling in-memory image caching and processing. Designed the playback user interaction, user interface components and widgets adopted by all of DreamWorks' proprietary playback tools. Enhanced the proprietary image I/O library, which supports a variety of image formats.
- Enforced gamma-correction consistency across most of the applications throughout the 3D and 2D pipeline.

General:

- Enhanced and supported the core scene-graph library capabilities. Added the ability to decompose individual scene objects' information into show, sequence, and shot-level data.
- Designed and developed general core libraries (specific memory allocation schemes, particle file format standardization, etc.). Also enhanced existing libraries by refactoring code.

Animation Science (Formerly **ArSciMed** - Paris, France)
Computer Graphics Software Engineer

Sunnyvale, CA. - 1996 to 1999

As a member of the research and development team, designed and developed major parts of the particle systems and character animation technology. Solved algorithmic problems leading to innovative solutions and participated in the release of several software products while interacting with animators and the marketing department.

Crowd simulation: lead developer for Rampage/SDK

- Designed and implemented the underlying 2D behavior model based on a perception and reaction scheme. Solved character avoidance problems. Developed dynamics rules adapted to character motion.
- Enhanced obstacle collision detection and avoidance, based on earlier work on 3D collision detection.
- Multi-threaded the simulation engine.

Particle systems: developer for Kinema/SDK and Outburst, a plug-in for 3D Studio Max

- Enhanced and optimized space-partitioning data structure computation for collision detection purposes, designing a 3D triangle rasterization algorithm.
- Participated in the design and development of Outburst main features, including particle trail management, path following, customizable force fields with mathematical expressions and undo-redo management.

Rendering: research and development

- Researched global illumination and physically-based rendering algorithms such as Metropolis light transport and Bidirectional path tracing. Implemented the latter writing a Monte-Carlo rendering prototype.
- Developed a scan-line and ray-tracing rendering engine. Integrated an existing particle splatting algorithm, solving compositing problems between particles and surfaces.

Photometric simulation: designer and lead developer of Kinema/Lighting

- MS thesis research on Monte-Carlo algorithms and variance reduction techniques.
- Adapted the sampling capabilities of the particle systems engine.
- Implemented a photometric curves viewer in a Motif/OpenGL environment.

A.S.T.S. (Association Science Technologie et Société)
Teacher

Paris, France - Summer 1995 and 1996

Taught an introductory hands-on course on digital image manipulation, as part of a cultural project open to the public, focused on computer graphics, photography, video and semiology.

I.B.M.

Paris, France - 1991 to 1993

Project Leader Assistant

Designed and developed internal software projects involving system programming in C, on OS/2 operating system. Trained in project management, Merise design methodology and C/C++ programming.

Skills

Languages / APIs: C/C++, STL, OpenGL, X11/Motif, 3D-Studio Max/SDK.
System and network: Multi-threading, Unix system calls, sockets, MPI, RPC, PVM.
3D Software: Maya, PrMan, Mental Ray, 3D Studio Max, Houdini, Softimage.
Operating systems: Linux/Unix, Windows, MacOS.
Spoken Languages: Fluent in French, English and Spanish. Spoken Portuguese.
Misc: Project management, excellent communication and presentation skills.

Education

MS Computer Science (1996), Université de Marne-la-Vallée - Paris, France

Computer graphics and image synthesis. Emphasis on radiosity algorithms. *Thesis:* Nuclear medicine image synthesis.

Publications

- Feng Xie, Eric Tabellion, Andrew Pearce: *Soft Shadows by Ray-Tracing Multilayer Transparent Shadow Maps*. Eurographics Symposium on Rendering 2007.
- Eric Tabellion, Arnaud Lamarlette: *An Approximate Global Illumination System for Computer Generated Films*. Siggraph 2004.