

# Eric Tabellion

(415)-609-1359 - [eric@tabellion.org](mailto:eric@tabellion.org) - [www.tabellion.org/et](http://www.tabellion.org/et)

---

## Summary

Senior graphics software lead engineer and research scientist. Broad graphics research background, strong software design, C/C++ programming, technical leadership and management skills. Extensive experience architecting solutions and leading teams through complex engineering challenges, within a large-scale production software environment. Avid learner, self-motivated, with long history of deploying efficient tools for technologists and artists alike, enabling unprecedented high-end computer generated imagery.

---

## 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.

---

## Skills

Languages, APIs: C/C++, STL, ISPC, Python, Lua, OpenGL, GLSL, Java, HTML5, Javascript.  
System: Multi-threading, SIMD / SPMD vectorization, Protocol buffers RPC, Posix sockets.  
Graphics Software: Maya, Houdini, RenderMan, Arnold, Nuke, Photoshop, Illustrator, etc.  
Spoken Languages: Fluent in English, French and Spanish. Comfortable in Portuguese.

---

## Experience

### **Google, Inc. - Software Engineer**

Mountain View, CA. - 2017 to Present

Manager and technical lead of a team of 5 engineers, focusing on video search. Performed extensive prototyping and collaboration with back-end data and research teams. Built and launched features helping billions of users find video content on the web.

### **PDI / DreamWorks Animation - Technology Fellow**

Redwood City, CA. - 1999 to 2017

Team-leading, architecting, designing, writing and deploying core graphics software used by lighting and FX animators during the production of animated feature films, live-action film-effects and commercials.

*Rendering:*

- Lead a team of 10 engineers, developing our latest generation high-end and high-performance **Monte-Carlo ray tracing** system. Initially prototyped, architected and designed core integrators and physically based shading and lighting components. Scaled a new team, to build this rendering system from the ground up. Architected and co-designed major key components, such as a new physically based shading infrastructure suitable to brute-force ray tracing, and a new parallel geometry procedural system. Participated in designing its integration into our latest generation lighting tool. Making use of heavily threaded and vectorized code, and relying on state of the art open-source lower-level components (Embree, OpenImageIO, OpenSubdiv).
- Designed and developed our in-house **point-based global illumination** system, used to light and render all feature films from “*How to Train Your Dragon*” on. Developed software core libraries, and handled user-facing tool and pipeline integration. Collaborated with other engineers on various innovations, embracing physically based shading and making the system scalable to very complex scenes.
- Team lead, participated in the design, development and support of DreamWorks’ second-generation high quality **film production 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 and more recently point-based approaches. 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 photorealistic reflections and refractions. Implemented adaptive sampling and ray differentials for high quality anti-aliasing, as well as 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.
- 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.

## **PDI / DreamWorks Animation (continued)**

### *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 image post-processing tool, that became the new studio standard for efficiently rendering **motion-blur and depth-of-field** effects. 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.
- 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 a proprietary image I/O library, supporting a variety of image formats.
- Enforced gamma-correction consistency across most of the applications throughout the 3D and 2D pipeline.

### *Miscellaneous:*

- 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 low-level system libraries (pool memory allocation schemes, particle file format standardization, etc.). Enhanced existing libraries through code re-factoring.

## **Animation Science - Graphics Software Engineer**

Paris, France / 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 sampling algorithms and variance reduction techniques.
- Adapted the sampling capabilities of the particle systems engine.
- Implemented a photometric curves viewer in a Motif/OpenGL environment.

## **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. - Software Engineer**

Paris, France - 1991 to 1993

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

---

## Awards

- **Academy Award - Technical Achievement:** *for the creation of a computer graphics bounce lighting methodology that is practical at feature film scale.* 83<sup>rd</sup> Academy Awards, 2010.
- **Film credits:** *Shrek, Sinbad, Madagascar, Over the Hedge, Flushed Away, Bee-Movie, Kung-Fu Panda, Monsters vs. Aliens, How to Train Your Dragon, Megamind, Puss in Boots, Rise of the Guardians, The Croods, Turbo, Mr. Peabody and Sherman, Penguins of Madagascar, Home, Trolls.*

---

## Publications and Patents

- Mark Lee, Brian Green, Feng Xie, Eric Tabellion: *Vectorized Production Path Tracing.* High Performance Graphics 2017.
- Sven Woop, Carsten Benthin, Ingo Wald, Gregory S. Johnson, Eric Tabellion: *Exploiting Local Orientation Similarity for Efficient Ray Traversal of Hair and Fur.* High Performance Graphics 2014.
- Eric Tabellion: *Point-based global illumination directional importance mapping.* Patent: 9460553, 2012.
- Janne Kontkanen, Eric Tabellion, Ryan S. Overbeck: *Coherent Out-of-Core Point-Based Global Illumination.* Eurographics Symposium on Rendering 2011. Patent: 8780112.
- Stephen Yuen, Eric Tabellion: *Infinite Complexity Deep Frame-Buffer Rendering.* Patent: 8497876, 2009
- Feng Xie, Eric Tabellion, Andrew Pearce: *Soft Shadows by Ray-Tracing Multilayer Transparent Shadow Maps.* Eurographics Symposium on Rendering 2007.
- Eric Tabellion, Arnauld Lamorlette: *An Approximate Global Illumination System for Computer Generated Films.* Siggraph 2004.
- Eyal Cohen, Olivier Maury, Eric Tabellion: *Method for allowing a particle to follow a path.* Patent: 6014151, 1997

---

## Conference Talks

- Siggraph 2016 panel guest speaker. *What Makes a Production Renderer in 2016.* Brent Burley, Marcos Fajardo, Alexander Keller, Philippe Leprince, Maurice van Swaaij, Eric Tabellion.
- High Performance Graphics 2013 panel guest speaker. *High-Performance Graphics in Film.* Matt Pharr, Eric Tabellion, Luca Fascione, David Adler, Rasmus Tamstorf, Thiago Ize, Marcos Fajardo.
- FMX 2013 talk: *Point-Based Global-Illumination for Feature Animation.*
- Siggraph 2012 talk: *Point-Based Global Illumination Directional Importance Mapping.*
- Eurographics Symposium on Rendering 2012, guest keynote speaker.
- Siggraph 2011 talk: *Coherent Out-of-Core Point-Based Global Illumination.*
- San Jose State University, guest speaker, 2011: *The Science of Film-Production Rendering.*
- Siggraph 2010 course: *Global Illumination Across Industries.* Jaroslav Krivanek, Anton Kaplayan, Michael Bunnell, Peter-Pike Sloan, Eric Tabellion, Per H. Christensen, Marcos Fajardo, John Hutchinson.
- Siggraph 2008 course: *Practical Global Illumination with Irradiance Caching.* Jaroslav Krivanek, Pascal Gautron, Gregory J. Ward, Henrik Wann Jensen, Per Christensen, Eric Tabellion.
- Siggraph 2004 technical paper: *An Approximate Global Illumination System for Computer Generated Films.* Eric Tabellion.