

# Introduction

---

Developing a production 3D rendering system is an intimidating task. The skills required are both broad and deep, requiring the mastery of practically every aspect of computer graphics from curves and surfaces through to compositing and image file formats. However, it doesn't stop there; the developer needs to be familiar with many aspects of computer science and engineering typically not considered to be graphics related: compiler development, processor architecture, signal processing, virtual machines, and perhaps most importantly of all, the software engineering skills to bring together such a diverse range of techniques into a coherent and manageable body of code.

While many books have been written about rendering, they are typically limited to either the basics of ray tracing or they specialize on a certain aspect of rendering research. When I began developing a renderer of my own, I rapidly found that these texts told me very little about *real* rendering, as it is used in film and video production. Rather than ray tracing spheres and polygons with Phong shading I needed to know how to parse RIB files, and execute compiled shaders, on a range of complex surfaces. I learned about shading engines by examining the compiled shaders from other rendering systems.

As my own rendering system (Angel) developed I found that my experiences were not unique, and I was lucky enough to meet others who were delighted to share their own ideas on dicing strategies, grid sizes, optimization techniques and other minutiae that enable a renderer to tackle the complex scenes found in production. Rendering systems rarely seem to be developed by large teams, but rather by enthusiastic individuals, who are always happy to share and learn from each other. Many renderers which started out as experimental projects, much like mine, have grown into commercial products used across the world.

A few years later when I was invited to contribute a chapter to the *Handbook of Computer Animation* (Vince, 2002) I realized this was an opportunity to share my experience of developing Angel, and present a more realistic introduction to how a production renderer works. While successful, one chapter could only provide an overview of such a huge subject and there was clearly scope for a more in-depth treatment.

Having established that there was a need for this book, rather than attempt to write it myself, I decided to invite the experts I had met to each contribute a chapter on a subject of their choice. I was very pleased when every one of them said yes, and with only a little rearrangement and coercion the rendering pipeline was carved up between us, to produce a comprehensive and in-depth study of how a production render is designed and implemented.

Chapter 1 is an extended and updated version of my original chapter that appeared in the *Handbook of Computer Animation*. The intention is that this provides a roadmap for the later chapters, introducing the concepts and ideas which will be expanded upon later. Though the chapters are presented in a logical sequence they can be read in any order, using the first chapter as a guide.

Chapter 2 is written by Rick LaMont, and deals with the overall structure of the rendering system, showing how the scene is created and represented within the renderer, and how the objects in the scene can be passed through to the various rendering stages. Rick is CTO of Dot C Software, and the lead developer of the RenderDotC renderer.

Chapter 3 deals more specifically with the geometry types typically found in a production render, explaining how surfaces can be evaluated and manipulated. It was written as a team effort by myself, Paul, Scott and Rick.

Having prepared the geometry, it must then be shaded by a procedural shading engine. In Chapter 4 Mark Elendt explains how this can be implemented. Mark was the very first employee of Side Effects Software Inc, developers of the Houdini animation system, where he holds the position of Senior Mathematician. In addition to his contributions to Houdini itself (for which he has received a Technical Achievement Award from the Academy of Motion Picture Arts and Sciences) he is the chief architect of Side Effects' Mantra rendering system, and designer of its VEX shading language.

Matthew Bentham is a programmer at ART VPS, where he develops the shader compiler for their RenderDrive and PURE range of hardware rendering products. In Chapter 5 Matthew describes how a compiler can be written to convert a high level language such as RenderMan SL into a format suitable for use by the shading engine.

While historically most systems capable of handling scenes of the complexity required by commercial production used scanline techniques for efficiency, more recently ray tracing and global illumination have been integrated, to create hybrid renderers. Ray tracing and global illumination are the subjects of Chapters 6 and 7 respectively. In Chapter 6 Scott Iverson considers the problems of adding ray tracing support to a render without sacrificing the performance and flexibility that users expect in a production system. Scott is the founder of SiTex Graphics where he has developed the AIR rendering system, and supporting tools.

Jacopo Pantaleoni, developer of Lightflow Rendering Tools, is the author of Chapter 7, which discusses the problem of global illumination. Jacopo studied mathematics at the University of Padova, has worked at NVIDIA as an intern in the OpenGL group, and is currently working as a Lighting Technical Director for Valiant Productions in London.

Once the surfaces have been shaded they must be assembled into a final image, as described by Paul Gregory in Chapter 8. Paul was the original architect of the Aqsis rendering system and, following its release as an open source project, leads its development team.

Chapter 9, which concludes the book, is a collection of thoughts, information and useful fragments of code contributed by myself, Rick, Scott, Mark and Paul. Though not necessarily fitting neatly within the structure of the book as a

whole, we hope you find the contents of this final chapter valuable or at least interesting.

Constructing a high quality renderer requires such a diverse set of skills that this book could never have been written by one person. Each of us has learnt something by reading the others' chapters, and I would like to thank each of the authors for their outstanding contributions.

Ian Stephenson  
National Centre for Computer Animation  
Bournemouth University



<http://www.springer.com/978-1-85233-821-3>

Production Rendering

Design and Implementation

Stephenson, I. (Ed.)

2005, XIII, 302 p. 115 illus., 32 illus. in color., Hardcover

ISBN: 978-1-85233-821-3