

国家科学技术学术著作出版基金资助出版

Hujun Bao
Wei Hua

Real-Time Graphics Rendering Engine

With 66 figures, 11 of them in color



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社



Springer

**ADVANCED TOPICS
IN SCIENCE AND TECHNOLOGY IN CHINA**

ADVANCED TOPICS IN SCIENCE AND TECHNOLOGY IN CHINA

Zhejiang University is one of the leading universities in China. In *Advanced Topics in Science and Technology in China*, Zhejiang University Press and Springer jointly publish monographs by Chinese scholars and professors, as well as invited authors and editors from abroad who are outstanding experts and scholars in their fields. This series will be of interest to researchers, lecturers, and graduate students alike.

Advanced Topics in Science and Technology in China aims to present the latest and most cutting-edge theories, techniques, and methodologies in various research areas in China. It covers all disciplines in the fields of natural science and technology, including but not limited to, computer science, materials science, life sciences, engineering, environmental sciences, mathematics, and physics.

Hujun Bao
Wei Hua

Real-Time Graphics Rendering Engine

With 66 figures, 11 of them in color

 ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

 Springer

Authors

Prof. Hujun Bao
State key Lab of Computer
Aided Design and Computer
Graphics at Zhejiang University,
Hangzhou 310058, China
E-mail: bao@cad.zju.edu.cn

Dr. Wei Hua
State key Lab of Computer
Aided Design and Computer
Graphics at Zhejiang University,
Hangzhou 310058, China
E-mail: huawei@cad.zju.edu.cn

ISSN 1995-6819 e-ISSN 1995-6827
Advanced Topics in Science and Technology in China

ISBN 978-7-308-08133-7
Zhejiang University Press, Hangzhou

ISBN 978-3-642-18341-6 e-ISBN 978-3-642-18342-3
Springer Heidelberg Dordrecht London New York

© Zhejiang University Press, Hangzhou and Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

A real-time graphics rendering engine is a middleware, and plays a fundamental role in various real-time or interactive graphics applications, such as video games, scientific computation visualization systems, CAD systems, flight simulation, etc. There are various rendering engines, but in this book we focus on a 3D real-time photorealistic graphics rendering engine, which takes 3D graphics primitives as the input and generates photorealistic images as the output. Here, the phrase “real-time” indicates that the image is generated online and the rate of generation is fast enough for the image sequence to be looked like a smooth animation. For conciseness, we use the rendering engine to represent a 3D real-time photorealistic graphics rendering engine throughout this book.

As a rendering engine is a middleware, users are mainly application developers. For application developers, a rendering engine is a software development kit. More precisely, a rendering engine consists of a set of reusable modules such as static or dynamic link libraries. By using these libraries, developers can concentrate on the application’s business logic, not diverting attention to rather complicated graphics rendering issues, like how to handle textures or how to calculate the shadings of objects. In most cases, a professional rendering engine usually does rendering tasks better than the programs written by application developers who are not computer graphics professionals. Meanwhile, adopting a good rendering engine in application development projects can reduce the development period, since lots of complex work is done by the rendering engine and, consequently, development costs and risks are alleviated.

In this book we are going to reveal the modern rendering engine’s architecture and the main techniques used in rendering engines. We hope this book can be good guidance for developers who are interested in building their own rendering engines.

The chapters are arranged in the following way. In Chapter 1, we introduce the main parts of a rendering engine and briefly their functionality. In Chapter 2, basic knowledge related to developing real-time rendering is introduced. This covers the rendering pipeline, the visual appearance and shading and lighting models. Chapter 3 is the main part of this book. It unveils the architecture of the rendering engine through analyzing the Visionix system, the rendering engine developed by

the authors' team. Lots of details about implementation are also presented in Chapter 3. In Chapter 4, a distributed parallel rendering system for a multi-screen display, which is based on Visionix, is introduced.

In Chapters 5 and 6, two particular techniques for real-time rendering that could be integrated into rendering engines are presented. Chapter 5 presents an overview of real-time rendering approaches for a large-scale terrain, and a new approach based on the asymptotic fractional Brownian motion. Chapter 6 presents a variation approach to a computer oriented bounding box tree for solid objects, which is helpful in visibility culling and collision detection.

This book is supported by the National Basic Research Program of China, also called "973" program (Grant Nos. 2002CB312100 and 2009CB320800) and the National Natural Science Foundation of China (Grant No. 60773184). Additionally, several contributors have helped the authors to create this book.

Dr. Hongxin Zhang and Dr. Rui Wang from CAD&CG State Key Lab, Zhejiang University, China, have made key contributions to Chapter 2 "Basics of Real-time Rendering". Ying Tang from Zhejiang Technology University, China, has done lots of work on tailoring contents, translating and polishing this book.

Special thanks to Chongshan Sheng from Ranovae Technologies, Hangzhou, China, as one of the main designers, for providing a lot of design documents and implementation details of the Visionix system, which is a collaboration between Ranovae Technologies and the CAD&CG State Key Lab.

Many people who work, or have ever studied, at the CAD&CG State Key Lab, Zhejiang University, provided help and support for this book: Rui Wang, Huaisheng Zhang, Feng Liu, Ruijian Yang, Guang Hu, Fengming He, Wei Zhang, Gaofeng Xu, Ze Liang, Yifei Zhu, Yaqian Wei, En Li, and Zhi He.

Hujun Bao
Wei Hua
Hangzhou, China
October, 2010

Contents

1	Introduction	1
1.1	Scene Graph Management	2
1.2	Scene Graph Traverse	4
1.3	Rendering Queue	5
1.4	Rendering Module	5
2	Basics of Real-Time Rendering	7
2.1	Rendering Pipeline	7
2.1.1	Conceptual Rendering Phases	9
2.1.2	Programmable Rendering Pipeline	10
2.1.3	Geometry Transforms	11
2.2	Shading	12
2.2.1	Rendering Equation	12
2.2.2	Lighting	14
2.2.3	BRDF	15
2.2.4	Light Transport	17
2.3	Summary	19
	References	19
3	Architecture of Real-Time Rendering Engine	21
3.1	Overview	21
3.2	Basic Data Type	22
3.2.1	Single-Field Data Type	22
3.2.2	Multiple-Field Data Type	25
3.2.3	Persistent Pointer: TAddress<>	26
3.3	Basics of Scene Model	26
3.4	Entity	28
3.5	Feature	29
3.5.1	IAttributedObject and IFeature	29
3.5.2	IBoundedObject	31

3.5.3	IChildFeature.....	31
3.5.4	Subclasses of IGroupingFeature	32
3.5.5	Subclasses of IShapeFeature	33
3.5.6	IAnimatedFeature	38
3.5.7	Subclasses of ILightFeature.....	40
3.5.8	Subclasses of IBindableFeature.....	40
3.5.9	IGeometryFeature.....	42
3.5.10	IAppearanceFeature and Related Features	55
3.6	Scene Graph.....	73
3.7	Spatial Index.....	75
3.7.1	Relation Schema A	77
3.7.2	Relation Schema B	79
3.8	Scene Model Schema.....	79
3.9	Scene Model Interface and Implementation	82
3.9.1	Scope of Name and ID	82
3.9.2	Transaction	82
3.9.3	Scene Storage	82
3.9.4	Reference and Garbage Collection	83
3.9.5	Data Visit and Cache	84
3.9.6	Out-of-Core Entity.....	85
3.9.7	ISceneModel.....	86
3.9.8	ISceneStorage.....	89
3.9.9	Implementation of ISceneModel and ISceneStorage.....	91
3.10	Scene Manipulator.....	93
3.10.1	Manipulator Functions.....	94
3.10.2	Usage of Scene Model Manipulator	97
3.11	Traversing Scene Model	98
3.11.1	Traverse via Iterator.....	98
3.11.2	Traverse via Visitor.....	107
3.12	Rendering Engine	115
3.12.1	CRenderingEngine	115
3.12.2	The Composition of the CRenderingEngine.....	119
3.13	Render Queue and Its Manager	122
3.14	Camera Manager.....	123
3.15	GPU Resources and Its Manipulator	124
3.15.1	Texture Resource	125
3.15.2	Buffer Resource	126
3.15.3	Shader Program	128
3.15.4	GPU Resource Manipulator.....	128
3.16	Render Target and Its Manager.....	131
3.17	Render Control Unit	134
3.18	Pre-render and Its Manager	137

3.18.1	IPreRender.....	137
3.18.2	CPreRenderManager	140
3.19	Render Pipelines and Its Manager	142
3.19.1	IRenderPipeLine.....	142
3.19.2	Modular Render Pipeline.....	147
3.19.3	Render Module	157
3.19.4	CRenderPipelineManager.....	160
3.20	Examples of Pre-render	161
3.20.1	CVFCullingPreRender	161
3.20.2	CMirrorPreRender	163
3.20.3	COoCEntityLoader.....	165
3.20.4	CFeatureTypeClassifier	169
3.20.5	CRenderQueueElementProcessor.....	171
3.20.6	CLightCullingPreRender.....	173
3.21	Examples of Modular Render Pipeline and Render Module	174
3.21.1	CShapeRenderPipeline	175
3.21.2	CShapeRenderModule.....	176
3.22	Implementation Details of CRenderingEngine.....	186
3.22.1	Configure.....	186
3.22.2	Initialize.....	189
3.22.3	DoRendering	190
3.22.4	OpenSceneModel	190
3.23	Conclusion.....	191
	References.....	192
4	Rendering System for Multichannel Display.....	193
4.1	The Overview of Parallel Rendering	193
4.1.1	Client-Server	195
4.1.2	Master-Slave.....	196
4.2	The Architecture of a Cluster-Based Rendering System.....	196
4.3	Rendering System Interface.....	197
4.3.1	vxIRenderingSystem	199
4.3.2	vxIModel	201
4.3.3	vxIUI	218
4.3.4	The Basic Example.....	231
4.4	Server Manager.....	233
4.4.1	Functionality.....	233
4.4.2	Structure	233
4.4.3	CServerManager.....	236
4.4.4	CServiceRequestManager	236
4.4.5	CServiceRequestTranslator	238
4.4.6	CServiceRequestSender	238

4.4.7	CSystemStateManager, CScreenState and CRenderServerState	240
4.4.8	CServiceRequestSRThreadPool	242
4.4.9	IServiceRequest and Subclasses	243
4.5	Implementation of Rendering System Interface	245
4.5.1	Implementation Principles	245
4.5.2	Example 1: Startup System	246
4.5.3	Example 2: Open Scene Model	247
4.5.4	Example 3: Do Rendering and Swap Buffer	248
4.6	Render Server and Server Interface	250
4.7	Application: the Immersive Presentation System for Urban Planning	251
4.7.1	System Deployment	253
4.7.2	Functionality	254
	References	256
5	Optimal Representation and Rendering for Large-Scale Terrain	257
5.1	Overview	258
5.1.1	LOD Model of Terrain	258
5.1.2	Out-of-Core Techniques	262
5.2	Procedural Terrain Rendering	263
5.2.1	An Overview of Asymptotic Fractional Brownian Motion Tree	265
5.2.2	afBm-Tree Construction	268
5.2.3	Procedural Terrain Rendering	270
5.2.4	Application	275
5.3	Conclusion	277
	References	277
6	Variational OBB-Tree Approximation for Solid Objects	281
6.1	Related Work	282
6.2	The Approximation Problem of an OBB Tree	283
6.3	Solver for OBB Tree	285
6.3.1	Computation of Outside Volume for Single Bounding Box	285
6.3.2	Solver for OBB Tree	287
6.4	Experiments and Results	290
6.5	Conclusion	291
	References	292
	Index	295