



图灵原版计算机科学系列

3D游戏开发圣经



3D Game Engine Design

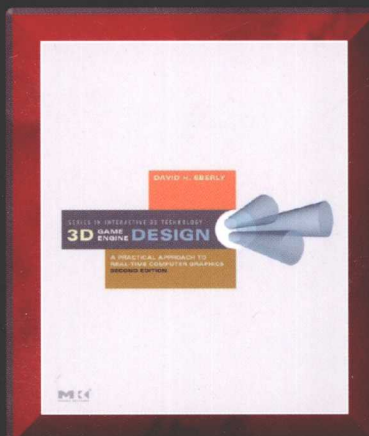
A Practical Approach to Real-Time Computer Graphics
Second Edition

3D游戏引擎设计

实时计算机图形学的应用方法

(英文版·第2版)

[美] David H. Eberly 著



人民邮电出版社
POSTS & TELECOM PRESS

TP391.41/Y39

2009.

TURING

图灵原版计算机科学系列

3D Game Engine Design

A Practical Approach to Real-Time Computer Graphics

Second Edition

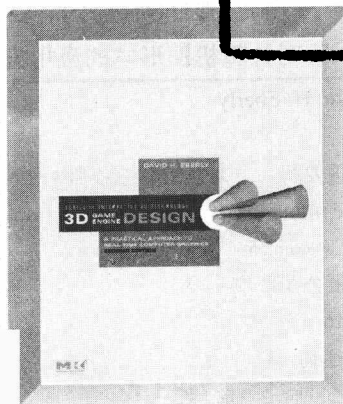
3D游戏引擎设计

实时计算机图形学的应用方法

(英文版·第2版)

[美] David H. Eberly 著

江苏工业学院图书馆
藏书章



人民邮电出版社
北京

图书在版编目(CIP)数据

3D游戏引擎设计: 实时计算机图形学的应用方法=3D
Game Engine Design: A Practical Approach to Real-Time
Computer Graphics, Second Edition; 第2版: 英文/(美)
埃伯利(Eberly, D. H.)著. —北京: 人民邮电出版社,
2009.2

(图灵原版计算机科学系列)

ISBN 978-7-115-19553-1

I. 3… II. 埃… III. ①三维—动画—计算机图形学—教材—英文 ②游戏—软件设计—教材—英文 IV. TP391.41
TP311.5

中国版本图书馆CIP数据核字(2008)第204792号

内 容 提 要

本书深入剖析了3D游戏引擎的设计, 书中许多内容对于更好地理解3D计算机图形学也极有帮助。本书首先介绍了几何转换和坐标系统等较基础的内容, 然后介绍曲线、渲染、效果等高级知识。本书基于作者自身在游戏产业中的工作、研究经验, 提供了算法、编程技术、代码等大量实用信息, 对于游戏设计者及相应的编程人员来说, 是一本非常有价值的参考书。

本书适合高等院校相关专业的师生、接受游戏软件开发培训的学生、相关技术人员及游戏开发人员阅读。

图灵原版计算机科学系列

3D游戏引擎设计: 实时计算机图形学的应用方法(英文版·第2版)

-
- ◆ 著 [美] David H. Eberly
责任编辑 杨海玲
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 64.75 彩插: 8
字数: 1243千字 2009年2月第1版
印数: 1-2 000册 2009年2月河北第1次印刷

著作权合同登记号 图字: 01-2008-5826号

ISBN 978-7-115-19553-1/TP

定价: 138.00元

读者服务热线: (010) 88593802 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

版 权 声 明

3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, Second Edition by David H. Eberly, ISBN: 978-0-12-447751-3.

Copyright © 2007 by Elsevier. All rights reserved.

Authorized English language reprint edition published by the Proprietor.

ISBN: 978-981-272-315-4

Copyright © 2009 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Elsevier (Singapore) Pte Ltd.

3 Killiney Road

#08-01 Winsland House I

Singapore 239519

Tel: (65)6349-0200

Fax: (65)6733-1817

First Published 2009

2009年初版

Printed in China by POSTS & TELECOM PRESS under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由Elsevier (Singapore) Pte Ltd. 授权人民邮电出版社在中华人民共和国境内（不包括香港特别行政区和台湾地区）出版发行。未经许可之出口，视为违反著作权法，将受法律之制裁。

TRADEMARKS

The following trademarks, mentioned in this book and the accompanying CD-ROM, are the property of the following organizations:

- Altivec is a trademark of Freescale Semiconductor.
- DirectX, Direct3D, Visual C++, Windows, Xbox, and Xbox 360 are trademarks of Microsoft Corporation.
- GameCube is a trademark of Nintendo.
- GeForce, Riva TNT, and the Cg Language are trademarks of NVIDIA Corporation.
- Java 3D is a trademark of Sun Microsystems.
- Macintosh is a trademark of Apple Corporation.
- Morrowind and The Elder Scrolls are trademarks of Bethesda Softworks, LLC.
- NetImmerse and Gamebryo are trademarks of Emergent Game Technologies.
- OpenGL is a trademark of Silicon Graphics, Inc.
- Pentium and Streaming SIMD Extensions (SSE) are trademarks of Intel Corporation.
- PhysX is a trademark of Ageia Technologies, Inc.
- Playstation 2 and Playstation 3 are trademarks of Sony Corporation.
- PowerPC is a trademark of IBM.
- Prince of Persia 3D is a trademark of Brøderbund Software, Inc.
- 3DNow! is a trademark of Advanced Micro Devices.
- 3D Studio Max is a trademark of Autodesk, Inc.

PREFACE

The first edition of *3D Game Engine Design* appeared in print over six years ago (September 2000). At that time, shader programming did not exist on consumer graphics hardware. All rendering was performed using the fixed-function pipeline, which consisted of setting render states in order to control how the geometric data was affected by the drawing pass.

The first edition contained a CDROM with the source code for Wild Magic Version 0.1, which included 1,015 source files and 17 sample applications, for a total of 101,293 lines of code. The distribution contained support only for computers running the Microsoft Windows operating system; the renderer was built on top of OpenGL; and project files were provided for Microsoft Visual C++ 6. Over the years, the source code evolved to Wild Magic Version 3.9, which contained additional support for Linux and Macintosh platforms, had OpenGL and Direct3D renderers, and included some support for shader programming. However, the design of the engine was still based on a fixed-function pipeline. The distribution also included support for multiple versions of Microsoft's compilers, support for other compilers on the various platforms, and contained some tools such as importers and exporters for processing of art assets.

This is the second edition of *3D Game Engine Design*. It is much enhanced, describing the foundations for shader programming and how an engine can support it. The second edition is about twice the size of the first. The majority of the increase is due to a more detailed description of all aspects of the graphics system, particularly about how shaders fit into the geometric pipeline. The material on scene graphs and their management is also greatly expanded. The second edition has more figures^① and less emphasis on the mathematical aspects of an engine.

The second edition contains a CDROM^② with the source code for Wild Magic Version 4.0, which includes 1,587 source files and 105 sample applications, for a total of 249,860 lines of code. The Windows, Linux, and Macintosh platforms are still supported, using OpenGL renderers. The Windows platform also has a Direct3D renderer whose performance is comparable to that of the OpenGL renderer. Multiple versions of Microsoft's C++ compilers are supported—versions 6, 7.0, 7.1, and 8.0 (Professional and Express Editions). The MINGW compiler and MSYS environment are also supported on the Windows platform. The Linux platform uses the g++ compiler, and the Macintosh platform uses Apple's Xcode tools.

① 凡是编号后标有“*”的图都配有对应的彩色插图。——编者注

② 本书出版并未随书附带光盘，凡书中提到CDROM中的Wild Magic Version 4.0的源代码可从Geometric Tools公司网站 (www.geometrictools.com) 下载。——编者注

The graphics system of Wild Magic Version 4.0 is fully based on shader programming and relies on NVIDIA's Cg programming language. The precompiled shader programs were created using the arbvp1 and arbfp1 profiles for OpenGL and using the vs_2_0 and ps_2_0 profiles for Direct3D, so your graphics hardware must support these in order to run the sample applications. If your graphics hardware supports only lesser profiles such as vs_1_1 and ps_1_1, you must recompile the shader programs with these profiles and use the outputs instead of what is shipped on the CDROM. The distribution also contains a fully featured, shader-based software renderer to illustrate all aspects of the geometric pipeline, not just the vertex and pixel shader components.

The replacement of the fixed-function approach by a shader-based approach has made Wild Magic Version 4 a much more powerful graphics engine for use in all graphics applications, not just in games. Much effort went into making the engine easier to use and to extend, and into improving the performance of the renderers. I hope you enjoy this new manifestation of Wild Magic!

A book is never just the product of the author alone. Many people were involved in making this book as good as it possibly can be. Thanks to the reviewers for providing valuable and insightful feedback about the first edition regarding how to improve it for a second edition. A special thanks goes to Marc Olano (University of Maryland, Baltimore County) for taking the time to provide me with detailed comments based on his experience using the first edition as a textbook. Thanks to Elisabeth Beller, the production editor and project manager for all of my Morgan Kaufmann Publisher books, for assembling yet another fine group of people who have the superb ability to take my unattractive book drafts and make them look really good. And, as always, thanks to my editor Tim Cox for his patience and help in producing yet another book for Morgan Kaufmann Publishers.

GEOMETRIC TOOLS, INC. LICENSE AGREEMENTS

Wild Magic has two license agreements, one for the Foundation library source code and one for the remaining source code. These agreements are located in the directory `GeometricTools/WildMagic4` of the CD-ROM. The Foundation library agreement is `Wm4FoundationLicense.pdf` and the other agreement is `Wm4RestrictedLicense.pdf`.

INSTALLATION OF THE SOFTWARE

The installation manual and release notes are located in the directory `GeometricTools/WildMagic4` of the CD-ROM. The document name is `Wm4InstallationRelease.pdf`. Please read this document before installing the software.

CONTENTS

CHAPTER

1

INTRODUCTION

1.1	THE EVOLUTION OF GRAPHICS HARDWARE AND GAMES	1
1.2	THE EVOLUTION OF THIS BOOK AND ITS SOFTWARE	2
1.3	A SUMMARY OF THE CHAPTERS	3

CHAPTER

2

THE GRAPHICS SYSTEM

2.1	THE FOUNDATION	8
2.1.1	Coordinate Systems	9
2.1.2	Handedness and Cross Products	10
2.1.3	Points and Vectors	15
2.2	TRANSFORMATIONS	18
2.2.1	Linear Transformations	18
2.2.2	Affine Transformations	29
2.2.3	Projective Transformations	31
2.2.4	Properties of Perspective Projection	35
2.2.5	Homogeneous Points and Matrices	40
2.3	CAMERAS	43
2.3.1	The Perspective Camera Model	43
2.3.2	Model or Object Space	48
2.3.3	World Space	48
2.3.4	View, Camera, or Eye Space	50
2.3.5	Clip, Projection, or Homogeneous Space	52
2.3.6	Window Space	56
2.3.7	Putting Them All Together	58
2.4	CULLING AND CLIPPING	66
2.4.1	Object Culling	66

2.4.2	Back-Face Culling	67
2.4.3	Clipping to the View Frustum	70
2.5	RASTERIZING	77
2.5.1	Line Segments	77
2.5.2	Circles	82
2.5.3	Ellipses	84
2.5.4	Triangles	89
2.6	VERTEX ATTRIBUTES	92
2.6.1	Colors	92
2.6.2	Lighting and Materials	92
2.6.3	Textures	99
2.6.4	Transparency, Opacity, and Blending	117
2.6.5	Fog	122
2.6.6	And Many More	123
2.6.7	Rasterizing Attributes	124
2.7	ISSUES OF SOFTWARE, HARDWARE, AND APIS	125
2.7.1	A General Discussion	125
2.7.2	Portability versus Performance	127
2.8	API CONVENTIONS	128
2.8.1	Matrix Representation and Storage	129
2.8.2	Matrix Composition	134
2.8.3	View Matrices	134
2.8.4	Projection Matrices	136
2.8.5	Window Handedness	139
2.8.6	Rotations	140
2.8.7	Fast Computations Using the Graphics API	143

CHAPTER

3

RENDERERS	147
3.1	SOFTWARE RENDERING
3.1.1	Vertex Shaders
3.1.2	Back-Face Culling
3.1.3	Clipping
3.1.4	Rasterizing
3.1.5	Edge Buffers
3.1.6	Scan Line Processing
3.1.7	Pixel Shaders
3.1.8	Stencil Buffering
3.1.9	Depth Buffering
3.1.10	Alpha Blending

3.1.11	Color Masking	171
3.1.12	Texture Sampling	171
3.1.13	Frame Buffers	172
3.2	HARDWARE RENDERING	173
3.3	AN ABSTRACT RENDERING API	175
3.3.1	Construction and Destruction	175
3.3.2	Camera Management	176
3.3.3	Global-State Management	177
3.3.4	Buffer Clearing	178
3.3.5	Object Drawing	179
3.3.6	Text and 2D Drawing	180
3.3.7	Miscellaneous	180
3.3.8	Resource Management	182
3.4	THE HEART OF THE RENDERER	194
3.4.1	Drawing a Scene	195
3.4.2	Drawing a Geometric Primitive	198
3.4.3	Applying an Effect	199
3.4.4	Loading and Parsing Shader Programs	201
3.4.5	Validation of Shader Programs	213

CHAPTER

4

SCENE GRAPHS 217

4.1	SCENE GRAPH DESIGN ISSUES	217
4.1.1	The Core Classes	221
4.1.2	Spatial Hierarchy Design	226
4.1.3	Sharing of Objects	230
4.2	GEOMETRIC STATE	233
4.2.1	Vertex Buffers and Index Buffers	233
4.2.2	Transformations	234
4.2.3	Bounding Volumes	244
4.2.4	Geometric Types	251
4.3	RENDER STATE	259
4.3.1	Global State	259
4.3.2	Lights	261
4.3.3	Effects	266
4.4	THE UPDATE PASS	268
4.4.1	Geometric-State Updates	268
4.4.2	Render-State Updates	280

4.5	THE CULLING PASS	289
4.5.1	Hierarchical Culling	293
4.5.2	Sorted Culling	296
4.6	THE DRAWING PASS	297
4.6.1	Single-Pass Drawing	298
4.6.2	Single-Effect, Multipass Drawing	302
4.6.3	Multiple-Effect Drawing	304
4.7	SCENE GRAPH COMPILERS	305
4.7.1	A Scene Graph as an Expression	307
4.7.2	Semantics of Compilation	311

CHAPTER

5

CONTROLLER-BASED ANIMATION 315

5.1	KEYFRAME ANIMATION	317
5.1.1	Interpolation of Position	317
5.1.2	Interpolation of Orientation	318
5.1.3	Interpolation of Scale	318
5.2	KEYFRAME COMPRESSION	320
5.2.1	Fitting Points with a B-Spline Curve	321
5.2.2	Evaluation of a B-Spline Curve	325
5.2.3	Optimized Evaluation for Degree 3	333
5.3	INVERSE KINEMATICS	339
5.3.1	Numerical Solution by Jacobian Methods	341
5.3.2	Numerical Solution by Nonlinear Optimization	342
5.3.3	Numerical Solution by Cyclic Coordinate Descent	342
5.4	SKINNING	347
5.5	VERTEX MORPHING	349
5.6	PARTICLE SYSTEMS	350

CHAPTER

6

SPATIAL SORTING 353

6.1	BINARY SPACE PARTITIONING TREES	354
6.1.1	BSP Tree Construction	355
6.1.2	BSP Tree Usage	357
6.2	NODE-BASED SORTING	365
6.3	PORTALS	366

6.4	USER-DEFINED MAPS	375
6.5	OCCCLUSION CULLING	375

CHAPTER

7

LEVEL OF DETAIL	377
-----------------	-----

7.1	SPRITES AND BILLBOARDS	378
7.2	DISCRETE LEVEL OF DETAIL	379
7.3	CONTINUOUS LEVEL OF DETAIL	380
7.3.1	Simplification Using Quadric Error Metrics	380
7.3.2	Reordering of Vertices and Indices	385
7.3.3	Terrain	386
7.4	INFINITE LEVEL OF DETAIL	387

CHAPTER

8

COLLISION DETECTION	389
---------------------	-----

8.1	THE METHOD OF SEPARATING AXES	393
8.1.1	Extrema of Convex Polygons or Convex Polyhedra	394
8.1.2	Stationary Objects	404
8.1.3	Objects Moving with Constant Linear Velocity	412
8.1.4	Oriented Bounding Boxes	436
8.2	FINDING COLLISIONS BETWEEN MOVING OBJECTS	444
8.2.1	Pseudodistance	444
8.2.2	Contact between Moving Intervals	446
8.2.3	Computing the First Time of Contact	448
8.2.4	Estimating the First Derivative	453
8.3	A DYNAMIC COLLISION DETECTION SYSTEM	455
8.3.1	The Abstract Base Class	455
8.3.2	Pseudodistances for Specific Pairs of Object Types	461
8.3.3	Collision Culling with Axis-Aligned Bounding Boxes	465
8.4	OBJECT PICKING	472
8.4.1	Constructing a Pick Ray	472
8.4.2	Scene Graph Support	475
8.4.3	Staying on Top of Things	479
8.4.4	Staying Out of Things	481
8.5	PATHFINDING TO AVOID COLLISIONS	481
8.5.1	Environments, Levels, and Rooms	482

8.5.2	Moving between Rooms	486
8.5.3	Moving between Levels	486
8.5.4	Moving through the Outdoor Environment	488
8.5.5	Blueprints	488
8.5.6	Visibility Graphs	489
8.5.7	Envelope Construction	494
8.5.8	Basic Data Structures	503
8.5.9	Efficient Calculation of the Visibility Graph	504

CHAPTER**9****PHYSICS** 507

9.1	PARTICLE SYSTEMS	508
9.2	MASS-SPRING SYSTEMS	510
9.2.1	Curve Masses	510
9.2.2	Surface Masses	513
9.2.3	Volume Masses	516
9.2.4	Arbitrary Configurations	519
9.3	DEFORMABLE BODIES	521
9.4	RIGID BODIES	522
9.4.1	The Rigid Body Class	525
9.4.2	Computing the Inertia Tensor	527

CHAPTER**10****STANDARD OBJECTS** 529

10.1	LINEAR COMPONENTS	529
10.2	PLANAR COMPONENTS	532
10.3	BOXES	534
10.4	QUADRICS	535
10.4.1	Spheres	535
10.4.2	Ellipsoids	535
10.4.3	Cylinders	537
10.4.4	Cones	537
10.5	SPHERE-SWEPT VOLUMES	538
10.5.1	Capsules	539
10.5.2	Lozenges	539

CHAPTER

11

CURVES	541
11.1 DEFINITIONS	542
11.2 REPARAMETERIZATION BY ARC LENGTH	543
11.3 BÉZIER CURVES	545
11.3.1 Definitions	545
11.3.2 Evaluation	545
11.3.3 Degree Elevation	546
11.3.4 Degree Reduction	546
11.4 NATURAL, CLAMPED, AND CLOSED CUBIC SPLINES	548
11.4.1 Natural Splines	550
11.4.2 Clamped Splines	550
11.4.3 Closed Splines	550
11.5 B-SPLINE CURVES	551
11.5.1 Types of Knot Vectors	552
11.5.2 Evaluation	553
11.5.3 Local Control	558
11.5.4 Closed Curves	558
11.6 NURBS CURVES	560
11.7 TENSION-CONTINUITY-BIAS SPLINES	562
11.8 PARAMETRIC SUBDIVISION	566
11.8.1 Subdivision by Uniform Sampling	566
11.8.2 Subdivision by Arc Length	566
11.8.3 Subdivision by Midpoint Distance	567
11.8.4 Fast Subdivision for Cubic Curves	568
11.9 ORIENTATION OF OBJECTS ON CURVED PATHS	570
11.9.1 Orientation Using the Frenet Frame	571
11.9.2 Orientation Using a Fixed Up-Vector	571

CHAPTER

12

SURFACES	573
12.1 INTRODUCTION	573
12.2 BÉZIER RECTANGLE PATCHES	574
12.2.1 Definitions	574
12.2.2 Evaluation	575

12.2.3	Degree Elevation	575
12.2.4	Degree Reduction	576
12.3	BÉZIER TRIANGLE PATCHES	578
12.3.1	Definitions	578
12.3.2	Evaluation	578
12.3.3	Degree Elevation	580
12.3.4	Degree Reduction	580
12.4	B-SPLINE RECTANGLE PATCHES	582
12.5	NURBS RECTANGLE PATCHES	583
12.6	SURFACES BUILT FROM CURVES	584
12.6.1	Cylinder Surfaces	584
12.6.2	Generalized Cylinder Surfaces	585
12.6.3	Revolution Surfaces	586
12.6.4	Tube Surfaces	586
12.7	PARAMETRIC SUBDIVISION	587
12.7.1	Subdivision of Rectangle Patches	587
12.7.2	Subdivision of Triangle Patches	602

CHAPTER**13**

CONTAINMENT METHODS		609
13.1	SPHERES	609
13.1.1	Point in Sphere	609
13.1.2	Sphere Containing Points	610
13.1.3	Merging Spheres	616
13.2	BOXES	617
13.2.1	Point in Box	617
13.2.2	Box Containing Points	618
13.2.3	Merging Boxes	625
13.3	CAPSULES	627
13.3.1	Point in Capsule	627
13.3.2	Capsule Containing Points	628
13.3.3	Merging Capsules	629
13.4	LOZENGES	630
13.4.1	Point in Lozenge	631
13.4.2	Lozenge Containing Points	631
13.4.3	Merging Lozenges	633
13.5	CYLINDERS	634
13.5.1	Point in Cylinder	634

13.5.2	Cylinder Containing Points	634
13.5.3	Least-Squares Line Moved to Minimum-Area Center	635
13.5.4	Merging Cylinders	635
13.6	ELLIPSOIDS	636
13.6.1	Point in Ellipsoid	636
13.6.2	Ellipsoid Containing Points	637
13.6.3	Merging Ellipsoids	638

CHAPTER

14

DISTANCE METHODS		639
14.1	POINT TO LINEAR COMPONENT	639
14.1.1	Point to Line	640
14.1.2	Point to Ray	640
14.1.3	Point to Segment	641
14.2	LINEAR COMPONENT TO LINEAR COMPONENT	642
14.2.1	Line to Line	642
14.2.2	Line to Ray	643
14.2.3	Line to Segment	644
14.2.4	Ray to Ray	645
14.2.5	Ray to Segment	645
14.2.6	Segment to Segment	645
14.3	POINT TO TRIANGLE	646
14.4	LINEAR COMPONENT TO TRIANGLE	651
14.4.1	Line to Triangle	651
14.4.2	Ray to Triangle	654
14.4.3	Segment to Triangle	654
14.5	POINT TO RECTANGLE	655
14.6	LINEAR COMPONENT TO RECTANGLE	657
14.6.1	Line to Rectangle	657
14.6.2	Ray to Rectangle	659
14.6.3	Segment to Rectangle	660
14.7	TRIANGLE OR RECTANGLE TO TRIANGLE OR RECTANGLE	661
14.8	POINT TO ORIENTED BOX	663
14.9	LINEAR COMPONENT TO ORIENTED BOX	663
14.9.1	Line to Oriented Box	664
14.9.2	Ray to Oriented Box	666
14.9.3	Segment to Oriented Box	666
14.10	TRIANGLE TO ORIENTED BOX	667

14.11	RECTANGLE TO ORIENTED BOX	669
14.12	ORIENTED BOX TO ORIENTED BOX	670
14.13	MISCELLANEOUS	672
14.13.1	Point to Ellipse	672
14.13.2	Point to Ellipsoid	673
14.13.3	Point to Quadratic Curve or to Quadric Surface	674
14.13.4	Point to Circle in 3D	675
14.13.5	Circle to Circle in 3D	676

CHAPTER**15**

	INTERSECTION METHODS	681
15.1	LINEAR COMPONENTS AND CONVEX OBJECTS	681
15.2	LINEAR COMPONENT AND PLANAR COMPONENT	684
15.3	LINEAR COMPONENT AND ORIENTED BOX	686
15.3.1	Test-Intersection Query	686
15.3.2	Find-Intersection Query	693
15.4	LINEAR COMPONENT AND SPHERE	698
15.4.1	Line and Sphere	698
15.4.2	Ray and Sphere	700
15.4.3	Segment and Sphere	701
15.5	LINE AND SPHERE-SWEPT VOLUME	703
15.5.1	Line and Capsule	703
15.5.2	Line and Lozenge	708
15.6	LINE AND QUADRIC SURFACE	709
15.6.1	Line and Ellipsoid	709
15.6.2	Line and Cylinder	710
15.6.3	Line and Cone	710
15.7	CULLING OBJECTS BY PLANES	710
15.7.1	Oriented Boxes	711
15.7.2	Spheres	712
15.7.3	Capsules	712
15.7.4	Lozenges	713
15.7.5	Ellipsoids	713
15.7.6	Cylinders	715
15.7.7	Cones	716
15.7.8	Convex Polygons or Convex Polyhedra	717