



三维数字地球引擎设计

3D Engine Design for Virtual Globes

[美] 帕特里克·卡兹 (Patrick Cozzi)

[美] 凯文·瑞恩 (Kevin Ring)

著

杨超 于荣欢 吴玲达 郭静

胡华全 郝红星 李强 谢维

译



国防工业出版社
National Defense Industry Press



CRC Press
Taylor & Francis Group



装备科技译著出版基金

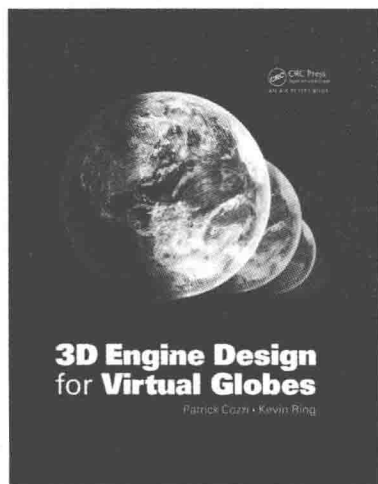
3D Engine Design for Virtual Globes

三维数字地球引擎设计

[美] 帕特里克·卡兹(Patrick Cozzi) 著

[美] 凯文·瑞恩(Kevin Ring)

杨超 于荣欢 吴玲达 郭静 译
胡华全 郝红星 李强 谢维



国防工业出版社

· 北京 ·

著作权合同登记 图字:军-2015-256号

图书在版编目(CIP)数据

三维数字地球引擎设计/(美)帕特里克·卡兹(Patrick Cozzi),
(美)凯文·瑞恩(Kevin Ring)著;杨超等译. —北京:国防工业出版社,2017.11

书名原文:3D Engine Design for Virtual Globes

ISBN 978-7-118-11374-7

I. ①三… II. ①帕… ②凯… ③杨… III. ①数字地球
-应用软件 IV. ①P208

中国版本图书馆CIP数据核字(2017)第261967号

Authorized translation from English language edition published by CRC Press, an imprint of Taylor&Francis Group LLC.

All rights Reserved.

版权所有,侵权必究.

National Defense Industry Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale throughout Mainland of China. No part of the publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher. 本书中文简体翻译版授权由国防工业出版社独家出版并限在中国大陆地区销售. 未经出版者书面许可,不得以任何方式复制或发行本书的任何部分.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal. 本书封面贴有 Taylor & Francis 公司防伪标签,无标签者不得销售.

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号 邮政编码100048)

北京嘉恒彩色印刷有限责任公司

新华书店经售

*

开本 710×1000 1/16 印张 28¼ 字数 515 千字

2017年11月第1版第1次印刷 印数 1—3000册 定价 145.00元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

Patrick 说：

感谢父母在 1994 年给我买了第一台计算机。老实说，我只想用它玩游戏；我没想用它能做出任何东西来。

Kevin 说：

当我 7 岁时，我发誓要写出自己的计算机游戏。将此书献给我的爸妈，他们认为那是一个不错的主意。

序 言

图形渲染技术在计算机图形学领域已经有很长的发展历史了。美国国家航空航天局(NASA)喷气推进实验室(JPL)的 Jim Blinn 早在 20 世纪 70 年代末 80 年代初就利用地球动画展示了 NASA 的某些空间任务,比如最有名的“旅行者”号探测器探测木星、土星、天王星和海王星任务。

当前,数字地球渲染技术不仅在 NASA 中进行应用,而且在许多三维游戏中也得到了应用。

目前最有名的三维数字地球包括谷歌的 Google Earth、NASA 的 World Wind、微软的 Bing Maps 3D,以及 ESRI 的 ArcGIS。这些应用能够让人们游览大量的真实影像、地形、矢量数据。

虽然,当前数字地球技术已经得到了非常广泛的应用,但是,到目前为止,还没有一本全面介绍数字地球相关技术的书籍。我们希望这本书能够通过提供对数字地球渲染算法的深入解析来弥补这一空白。本书的重点聚焦于面向构建精确三维数字地球的球面、地形、影像和矢量处理算法。

写这本书的知识主要来源于作者在 AGI 公司 STK 和 Insight3D 项目开发中的经验。STK 项目是 AGI 公司从 1993 年以来长期开发的太空、国防和情报建模与分析应用系统。Insight3D 是 AGI 公司关于太空和地理信息系统应用的三维可视化组件项目。希望我们的开发经验能够给读者提供实用帮助。

1. 潜在读者

本书主要面向于对图形渲染算法、渲染引擎设计、地理信息系统、数字地球、地形和大世界渲染感兴趣的图形开发者。本书包含的内容十分丰富,适用的读者层次范围也很广,包括开发者、研究者、学生和对数字地球感兴趣的爱好者均适合阅读本书。我们希望纵览式的说明方式能够满足对总体内容和理论比较感兴趣的读者,而家教式的源码实例能够满足实用性读者的需求。

本书不需要读者有数字地球或地形方面的任何背景知识,本书的背景基础知识中已经包含了相关的内容,比如椭球体的渲染、地形渲染和其它的高级主题,比如深度缓存精度和多线程等。

读者只需要一些基本的计算机图形学知识,比如矢量和矩阵;一些图形渲染接口的使用经验,比如 OpenGL 或 Direct3D;一些 shader 语言的使用经历。如

果读者已经理解如何将一个 shader 应用到光线跟踪算法中,肯定就能很好地理解阅读这本书。如果,读者对图形学不了解,欢迎访问我们的网站:<http://www.virtualglobebook.com/>。

这个网站同时也提供本书的实例代码下载。

最后,读者还必须具备面向对象的编程语言知识,比如 C++, C# 或者 Java。

2. 感谢

很多人为了这本书付出了自己的辛勤劳动。没有他们的帮助,这本书就达不到现在的内容和质量。

我们知道写这样一本书不是一个简单的任务。我们很大一部分应该归功于我们的雇主 AGI 公司的理解与支持。谢谢 Paul Graziani、Frank Linsalata、Jimmy Tucholski、Shashank Narayan 和 Dave Vallado。谢谢他们在这个项目开始阶段的支持。同时我们也要感谢 Deron Ohlarik、Mike Bartholomew、Tom Fili、Brett Gilbert、Frank Stoner 和 Jim Woodburn。谢谢他们的参与,包括对各章节的检查和不知疲倦地解答我们的问题。特别感谢 Deron 在项目起始阶段起的作用,非常感谢 Jim 在第二章所做的贡献。同时,我们也要感谢 Francis Kelly、Jason Martin 和 Glenn Warrington 的封面设计工作。

如果没有宾夕法尼亚州立大学好友 Norm Badler、Steve Lane 和 Joe Kider 的鼓励,我们就不会开始这本书的撰写。Norm 最开始鼓励我们这个想法,并提议由 A K Peters 进行出版,因此我们非常感谢他。事实上, Sarah Cutler、Kara Ebrahim、Alice 和 Klaus Peters 对我们的帮助贯穿整个过程。Eric Haines (Autodesk) 也为我们提供了很多帮助,让我们能够从正确的方向上着手。



非常感谢各章节测试小组,他们反馈了很多有用信息,促使我们做出很多改善。以字母表顺序,他们是 Quarup Barreirinhas (Google)、Eric Bruneton (Laboratoire Jean Kuntzmann)、Christian Dick (Technische Universität München)、Hugues Hoppe (Microsoft Research)、Jukka Jylänki (University of Oulu)、Dave Kasik

(Boeing)、Brano Kemen (Outerra)、Anton Fruhstuck Malischew (Greentube Internet Entertainment Solutions)、Emil Persson (AvalancheStudios)、Aras Pranckevicius (Unity Technologies)、Christophe Riccio (Imagination Technologies)、Ian Romanick (Intel)、Chris Thorne (VRshed)、Jan Paul Van Waveren (id Software)、Mattias Widmark (DICE)。

特别感谢两位审稿人: Dave Eberly (Geometric Tools, LLC) 从一开始就与我们一起工作, 多次审阅部分章节, 总是能够提出鼓励和建设性的反馈。Aleksandar Dimitrijevic (University of Nis) 审阅了部分章节, 他对该领域的热忱一直激励着我们。

还要感谢我们的家庭和朋友, 在很多个夜晚、周末和假期思念着我们(例如, 本节就是在圣诞夜写完的)。尤其是, 我们感谢 Kristen Ring、Peg Cozzi、Margie Cozzi、Anthony Cozzi、Judy MacIver、David Ring、Christy Rowe 和 Kota Chrome。

3. 致谢数据集提供者

数字地球之所以使人着迷, 就在于它提供了看起来无限的 GIS 数据空间, 包括地形、影像和矢量数据。谢天谢地, 很多相关数据集都是可以免费获取到的。我们在此对书中用到的数据集的提供者表示感谢。

(1) Natural Earth

Natural Earth (<http://www.naturalearthdata.com/>) 提供了公共区域的光栅化和矢量数据, 精度是 1:10、1:50 和 1:110 百万尺度。我们在图 1 中使用了这些图片, 并且贯穿全书使用了 Natural Earth 的矢量数据。

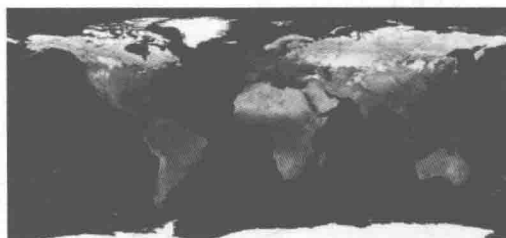


图 1 Natural Earth 的地球影像

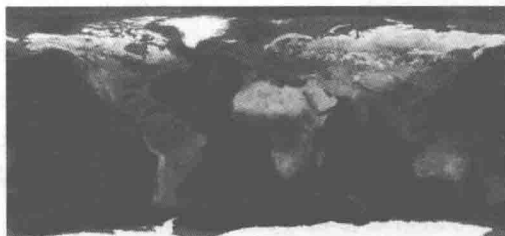
(2) NASA Visible Earth

NASA Visible Earth (<http://visibleearth.nasa.gov/>) 提供了大量的卫星影像。我们在图 2 以及全书中使用了这些图片。图 2(a) 和 (b) 是 NASA Blue Marble 集合的一部分, 归功于 NASA Earth Observatory 的 Reto Stockli。图 2(c)

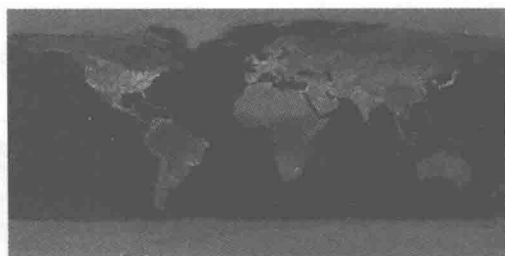
中的城市光照图像由 NASA GSFC 的 Craig Mayhew 和 Robert Simmon 提供。这些数据,要感谢 NASA GSFC 的 Marc Imhoff,以及 NOAA NGDC 的 Christopher Elvidge。



(a)



(b)



(c)

图2 NASA Visible Earth 提供的图像

(3) NASA World Wind

在地形实现中,我们使用了 NASA World Wind 的 mergedElevations 地形数据集(http://worldwindcentral.com/wiki/World_Wind_Data_Sources)。该数据集具有大多数美国地形(数据分辨率是 10m),以及世界其他部分(数据分辨率是 90m)。包括 3 个数据源:NASA Jet Propulsion Laboratory¹ 的 Shuttle Radar To-

¹ <http://www2.jpl.nasa.gov/srtm/>

pography Mission (SRTM); 美国地质调查局 (USGS)¹ 的 National Elevation Dataset (NED); 以及加州大学圣地亚哥分校斯克里普斯海洋学院地球物理和行星物理学研究所提供的 SRTM30 PLUS; SRTM30 海岸和桥梁图像。

(4) 美国国家图集

美国国家图集 (<http://www.nationalatlas.gov/atlasftp.html>) 免费提供了大量的地图数据。在我们的矢量数据渲染的讨论中, 使用了其中的机场和铁路站数据集。对于后者, 我们感谢政府的研究和创新技术, 即管理/运输统计局 (RI-TA/BTS) 国家交通地图数据库 (NTAD) 2005。

(5) 乔治亚理工学院

类似于很多开发者的地形算法, 我们使用华盛顿州 Puget Sound 的地形数据集, 如图 3 所示。这些数据集是乔治亚理工学院的大型几何模型存档 (Large Geometric Models Archive, LGMA) 的一部分 (<http://www.cc.gatech.edu/projects/large-models/ps.html>)。原始的数据集²从 USGS 获得, 华盛顿大学让它变得可用。Peter Lindstrom 和 Valerio Pascucci 抽取了其中的子集。

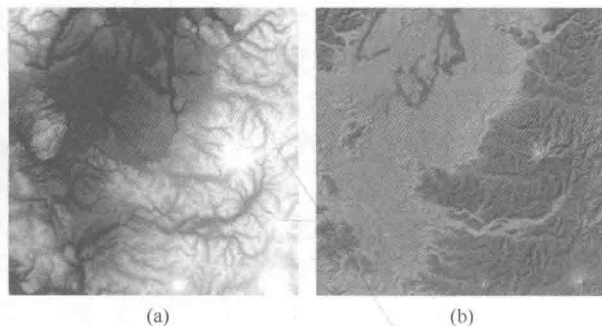


图3 (a) 高度图, (b) 乔治亚理工学院 LGMA 中 Puget Sound 的着色图 (纹理)

(6) Yusuke Kamiyamane

矢量数据渲染中使用的图标由 Yusuke Kamiyamane 创建, 在 Creative Commons Attribution 3.0 的许可下, 提供了一个大规模的图标集合 (<http://p.yusukekamiyamane.com/>)。

4. 反馈

有任何反馈、建议、更正, 欢迎与作者联系, 邮箱是 authors@virtualglobe-book.com。

¹ <http://ned.usgs.gov/>

² <http://rocky.ess.washington.edu/data/raster/tenmeter/onebytwo10/>

前 言

不要被这本书的名字所迷惑。书名只想告诉你,如果你有兴趣学习游戏中的高性能和地形渲染,那么这本书很适合你。如果你对绘图程序的特性和性能印象深刻,例如 NASA World Wind 或者 Google Earth,并且你想要知道如何编写此类程序,那么这本书也适合你。

关于计算机领域的某个主题,一些计算机类图书想要告诉读者几乎所有相关的东西,然而,那是事倍功半的,因为那些高级的描述往往让读者一知半解,而基础的细节又非常缺乏。要知道,基础的细节非常重要,它有助于在理论理解和实际源代码的鸿沟之间为读者建立起一座桥梁。在本书中,读者将获得高质量的关于数字地球和地形渲染的教程;高精度数据 3D 实时渲染的细节,包括实际源代码;以及要成为该领域专家所需要的数学基础。此外,本书还包括某些计算机主题的发展现状阐述,例如几何裁剪和 LOD 算法,这些技术对于大规模地形数据集的渲染非常重要。本书的参考文献非常广泛,让读者能够了解那些与构建数字地球相关的研究。

本书书名没有告诉你的是,为了充分利用并行化功能,书中有很多章节是介绍现代硬件的,包括多线程引擎设计、外存渲染、任务级并行化,以及处理并发、同步和共享资源的基本需求。本书内容对于数字地球渲染是必要的和有用的,同时,对于任何涉及科学计算、可视化、大数据处理的研究,本书也是非常重要的参考资料。实际上,相当于你花一本书的价钱买了两本书。我只想在办公室里摆放少量的技术书籍,因此会选择那些信息量较大而性价比较高的书。本书是其中之一。

——Dave Eberly

目 录

第一章 绪论	1
1.1 虚拟地球渲染中的挑战	1
1.2 内容简介	5
1.2.1 基础知识	5
1.2.2 精度修正	5
1.2.3 矢量数据	5
1.2.4 三维球面地形构建	6
1.3 OpenGlobe 结构	7
1.4 约定惯例	9

第一部分 基础知识

第二章 数学基础	10
2.1 虚拟地球坐标系	10
2.1.1 地理坐标系	10
2.1.2 WGS84 坐标系	12
2.2 椭球体基础	14
2.2.1 WGS84 椭球体	15
2.2.2 椭球体表面法线	16
2.2.3 大地纬度和高度	17
2.3 坐标变换	18
2.3.1 地理坐标系向 WGS84 坐标系转换	18
2.3.2 WGS84 坐标系向地理坐标系转换	21
2.4 椭球体上的曲线	30
2.5 资源	34
第三章 渲染设计	35
3.1 渲染需求	35
3.2 概要	39

3.2.1 代码结构	42
3.3 状态管理	43
3.3.1 全局状态	43
3.3.2 定义渲染状态	45
3.3.3 GL 状态与渲染状态同步	48
3.3.4 绘图状态	50
3.3.5 清除状态	51
3.3.6 通过状态排序	53
3.4 shader	56
3.4.1 编译和连接 shader	56
3.4.2 顶点属性	59
3.4.3 片元输出	62
3.4.4 uniform 单元	63
3.4.5 自动的 uniform 单元	67
3.4.6 高速缓存 shader	73
3.5 顶点数据	76
3.5.1 顶点缓冲区	77
3.5.2 引导缓冲区	82
3.5.3 顶点数组	83
3.5.4 GL 渲染操作	86
3.5.5 Direct3D 中的顶点数据	87
3.5.6 网格	87
3.6 纹理	93
3.6.1 创建纹理	93
3.6.2 采样器	98
3.6.3 用纹理渲染	100
3.6.4 GL 渲染操作	101
3.6.5 Direct3D 中的纹理	102
3.7 帧缓冲区	103
3.7.1 GL 渲染操作	105
3.7.2 Direct3D 中的帧缓冲区	106
3.8 完整流程:渲染一个三角形	106
3.9 资源	109

第四章 球面渲染	111
4.1 分格化	111
4.1.1 细分表面	111
4.1.2 细分表面操作	114
4.1.3 立方体图分格化	117
4.1.4 地理网格分格化	120
4.1.5 分格化算法比较	122
4.2 着色	122
4.2.1 光照	122
4.2.2 纹理	127
4.2.3 CPU/GPU 权衡	130
4.2.4 经纬度网格	131
4.2.5 夜间照明	134
4.3 GPU 光线投射	138
4.4 资源	142

第二部分 精度修正

第五章 顶点位置精度修正	144
5.1 抖动产生原因	145
5.1.1 浮点舍入误差	146
5.1.2 导致抖动的根本原因	147
5.1.3 为什么缩放不能解决抖动问题	150
5.2 根据中心渲染物体	150
5.3 使用 CPU 根据视点进行渲染	155
5.4 在 GPU 上根据视点进行渲染	158
5.4.1 通过 DSFUN90 提高精度	160
5.4.2 精度 LOD	162
5.5 一些建议	164
5.6 资源	166
第六章 深度缓存精度	168
6.1 深度缓存误差的原因	169
6.1.1 最小三角分隔	173
6.2 基本解决方案	174

6.3 补偿深度缓存	175
6.4 对数深度缓存	177
6.5 多视锥体渲染	179
6.5.1 性能影响	181
6.6 w 缓存	183
6.7 算法总结	183
6.8 资源	184

第三部分 矢量数据

第七章 矢量数据和折线	185
7.1 矢量数据源	186
7.2 解决 z 冲突	186
7.3 折线	188
7.3.1 批处理	189
7.3.2 静态缓存	191
7.3.3 线宽	192
7.3.4 轮廓线	196
7.3.5 采样	199
7.4 资源	200
第八章 多边形	201
8.1 纹理渲染	201
8.2 多边形镶嵌	202
8.2.1 输入整理	203
8.2.2 三角化	206
8.2.3 耳部剪除优化	213
8.2.4 细分	214
8.2.5 设置高度	218
8.2.6 绘制	218
8.2.7 管线修正	219
8.3 地形上的多边形	219
8.3.1 绘制方法	220
8.3.2 阴影体	221
8.3.3 采用阴影体绘制多边形	223

8.3.4 优化	225
8.4 资源	226
第九章 球面布告板	227
9.1 基础渲染	228
9.2 最小化纹理切换	233
9.2.1 纹理图集装箱算法	235
9.2.2 基于纹理图集的渲染	239
9.3 原点与偏移	241
9.4 文本渲染	244
9.5 资源	247
第十章 并行化资源准备	248
10.1 并行化无处不在	248
10.1.1 CPU 并行化	248
10.1.2 GPU 并行化	249
10.1.3 多线程并行化	250
10.2 数字地球中任务级的并行化	251
10.3 多线程的体系结构	252
10.3.1 消息队列	253
10.3.2 粗粒度线程	255
10.3.3 细粒度线程	260
10.3.4 异步 I/O	262
10.3.5 单线程测试/调试模式	263
10.4 OpenGL 多线程编程	263
10.4.1 一个 GL 线程,多个工作线程	263
10.4.2 多个线程,一个 GL 上下文	264
10.4.3 多线程,多内容	265
10.4.4 多线程驱动	274
10.5 参考资料	275

第四部分 三维球面地形构建

第十一章 地形基础知识	276
11.1 地形表示的基础理论	277
11.1.1 高度图	277

11.1.2 体素渲染	278
11.1.3 内隐式曲面	279
11.1.4 不规则三角网格	280
11.1.5 地形表示法小结	281
11.2 渲染高度图	282
11.2.1 创建三角形网格	283
11.2.2 顶点着色位移图	286
11.2.3 GPU 光线投射算法	288
11.2.4 高度放大技术	300
11.3 计算法向量	301
11.3.1 前向求差分算法	303
11.3.2 中心差分算法	305
11.3.3 Sobel 滤波方法	306
11.3.4 法向量计算方法小结	308
11.4 地形着色	309
11.4.1 颜色图和纹理坐标	310
11.4.2 细节图	311
11.4.3 程序着色	311
11.5 相关资料	323
第十二章 大面积地形渲染	325
12.1 细节等级(LOD)	326
12.1.1 离散 LOD	327
12.1.2 连续 LOD	327
12.1.3 层次 LOD	329
12.1.4 屏幕空间误差	330
12.1.5 Artifacts	332
12.2 预处理	334
12.2.1 高度图转换为 Mipmaps 和 Clipmaps	335
12.2.2 瓦片	336
12.2.3 网格简化	338
12.3 外存绘制	338
12.3.1 缓存层次结构	339
12.3.2 加载顺序策略	340
12.3.3 替换策略	342

12.3.4	预取	343
12.3.5	压缩	345
12.4	剔除	346
12.4.1	背面剔除	346
12.4.2	视锥体剔除	347
12.4.3	地平线剔除	347
12.4.4	硬件遮挡查询	349
12.4.5	从前往后绘制	351
12.5	资源	354
第十三章	geometry clipmapping 算法	355
13.1	clipmap 金字塔	357
13.2	顶点缓冲区	359
13.3	顶点和片断 shader	361
13.4	融合	364
13.5	clipmap 更新	367
13.5.1	环形寻址	368
13.5.2	更新区域	369
13.5.3	更新高程	372
13.5.4	更新法线	374
13.5.5	多线程 Out-of-Core 更新	376
13.5.6	上采样	379
13.5.7	替换和预取	381
13.5.8	比较和综合	382
13.6	着色	383
13.7	球体上的 geometry clipmapping 算法	385
13.7.1	在顶点 shader 中映射到椭球体表面	385
13.7.2	spherical clipmapping 算法	388
13.7.3	coordinate clipmapping 算法	389
13.8	相关资料	391
第十四章	chunked LOD 算法	392
14.1	chunks	393
14.2	选择过程	394
14.3	chunk 之间的裂痕	396
14.4	转换算法	397