

3D Engine Design for Virtual Globes

Patrick Cozzi and Kevin Ring

Editorial, Sales, and Customer Service Office

A K Peters, Ltd.
5 Commonwealth Road, Suite 2C
Natick, MA 01760
www.akpeters.com

Copyright © 2011 by A K Peters, Ltd.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

Library of Congress Cataloging-in-Publication Data

To be determined

Printed in the United States of America
15 14 13 12 11

10 9 8 7 6 5 4 3 2 1

Contents

Foreword	vii
Preface	vii
1 Introduction	1
1.1 Rendering Challenges in Virtual Globes	1
1.2 Contents Overview	5
1.3 OpenGlobe Architecture	8
1.4 Conventions	10
I Fundamentals	11
2 Math Foundations	13
2.1 Virtual Globe Coordinate Systems	13
2.2 Ellipsoid Basics	17
2.3 Coordinate Transformations	22
2.4 Curves on an Ellipsoid	34
2.5 Resources	39
3 Renderer Design	41
3.1 The Need for a Renderer	42
3.2 Bird's-Eye View	46
3.3 State Management	50
3.4 Shaders	63
3.5 Vertex Data	84
3.6 Textures	101
3.7 Framebuffers	112
3.8 Putting It All Together: Rendering a Triangle	115
3.9 Resources	119

4	Globe Rendering	121
4.1	Tessellation	121
4.2	Shading	133
4.3	GPU Ray Casting	149
4.4	Resources	154
II	Precision	155
5	Vertex Transform Precision	157
5.1	Jittering Explained	158
5.2	Rendering Relative to Center	164
5.3	Rendering Relative to Eye Using the CPU	169
5.4	Rendering Relative to Eye Using the GPU	171
5.5	Recommendations	177
5.6	Resources	180
6	Depth Buffer Precision	181
6.1	Causes of Depth Buffer Errors	181
6.2	Basic Solutions	188
6.3	Complementary Depth Buffering	189
6.4	Logarithmic Depth Buffer	191
6.5	Rendering with Multiple Frustums	194
6.6	W-Buffer	198
6.7	Algorithms Summary	198
6.8	Resources	199
III	Vector Data	201
7	Vector Data and Polylines	203
7.1	Sources of Vector Data	203
7.2	Combating Z-Fighting	204
7.3	Polylines	207
7.4	Resources	219
8	Polygons	221
8.1	Render to Texture	221
8.2	Tessellating Polygons	222
8.3	Polygons on Terrain	241
8.4	Resources	250

9	Billboards	251
9.1	Basic Rendering	252
9.2	Minimizing Texture Switches	258
9.3	Origins and Offsets	267
9.4	Rendering Text	271
9.5	Resources	273
10	Exploiting Parallelism in Resource Preparation	275
10.1	Parallelism Everywhere	275
10.2	Task-Level Parallelism in Virtual Globes	278
10.3	Architectures for Multithreading	280
10.4	Multithreading with OpenGL	292
10.5	Resources	304
IV	Terrain	305
11	Terrain Basics	307
11.1	Terrain Representations	308
11.2	Rendering Height Maps	313
11.3	Computing Normals	335
11.4	Shading	343
11.5	Resources	363
12	Massive-Terrain Rendering	365
12.1	Level of Detail	367
12.2	Preprocessing	376
12.3	Out-of-Core Rendering	381
12.4	Culling	390
12.5	Resources	400
13	Geometry Clipmapping	403
13.1	The Clipmap Pyramid	406
13.2	Vertex Buffers	408
13.3	Vertex and Fragment Shaders	411
13.4	Blending	414
13.5	Clipmap Update	417
13.6	Shading	435
13.7	Geometry Clipmapping on a Globe	436
13.8	Resources	443

14	Chunked LOD	445
14.1	Chunks	447
14.2	Selection	448
14.3	Cracks between Chunks	449
14.4	Switching	450
14.5	Generation	452
14.6	Shading	459
14.7	Out-of-Core Rendering	460
14.8	Chunked LOD on a Globe	462
14.9	Chunked LOD Compared to Geometry Clipmapping . . .	463
14.10	Resources	465
A	Implementing a Message Queue	467
	Bibliography	477
	Index	491