
Simulation Foundations, Methods and Applications

For further volumes:
www.springer.com/series/10128

Murilo G. Coutinho

Guide to Dynamic Simulations of Rigid Bodies and Particle Systems

 Springer

Murilo G. Coutinho
Sony Pictures Imageworks
Los Angeles, CA, USA

Series Editor

Prof. Louis G. Birta
School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada

ISSN 2195-2817

Simulation Foundations, Methods and Applications

ISBN 978-1-4471-4416-8

DOI 10.1007/978-1-4471-4417-5

Springer London Heidelberg New York Dordrecht

ISSN 2195-2825 (electronic)

ISBN 978-1-4471-4417-5 (eBook)

Library of Congress Control Number: 2012950598

© Springer-Verlag London 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, 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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

To my family

Izabella, Leticia and Nicholas

Preface

Physically based modeling is widely used within the computer graphics and mechanical engineering industries as a way of achieving realistic animations and accurate simulations of complex systems. Such complex systems are usually hard to animate using scripts, and difficult to analyze using conventional mechanics theory, which makes them perfect candidates for physically based modeling and simulation techniques.

The field of physically based modeling is broad. It includes everything from modeling a ball rolling on the floor, to a car engine working, to clothing a virtual character. The theory varies from precise mathematical methods to purpose-specific approximated solutions that are mathematically incorrect, but produce realistic animations for the particular situation being considered. Depending on the case, an approximated solution might serve the purpose, however, there are times when approximations are not admissible, and the use of accurate simulation engines is a requirement. Developing and implementing physically based dynamic-simulation engines that are robust is difficult. The main reason is that it requires a breadth of knowledge in a diverse set of subjects, each of them standing alone as a broad and complex topic.

Instead of attempting to address all types of simulation engines available in the broad area of physically based modeling, this book provides in-depth coverage of the most common simulation engines. These simulation engines restrict the general case of physically based modeling to the particular case wherein the objects interacting are either particles or rigid bodies.

This book is a comprehensive introduction to the techniques needed to produce realistic simulations and animations of particle and rigid-body systems. It focuses on both the theoretical and practical aspects of developing and implementing physically based dynamic-simulation engines that can be used to generate convincing animations of physical events involving particles and rigid bodies, such as the demolition of a bridge or building with debris falling all over. It can also be used to produce accurate simulations of mechanical systems, such as a robotic parts feeder where parts are dropped on a conveyor belt and then positioned and aligned as they hit fences strategically placed on the conveyor and used to align the parts at a specific orientation.

Guide to Dynamic Simulations of Rigid Bodies and Particle Systems was written for computer graphics, computer animation, computer-aided mechanical design and

modeling software developers who want to learn to incorporate physically based dynamic-simulation features into their own systems. The goal of this book is to make the principles and methods of physically based modeling of particle and rigid-body systems accessible to a broader audience of software developers who are familiar with mainstream computer-graphics techniques, and the associated mathematics.

The book is organized into three main topics: particle systems, rigid-body systems, and articulated rigid-body systems. The first chapter is an overview of how all techniques covered in this book fit together as independent modules constituting a simulation engine. The following chapters and appendices go into more detailed explanations for each technique. The techniques developed can be used to create simulation engines capable of combining particles, rigid bodies and articulated rigid bodies into a single system. Each chapter presents many algorithms and covers them in considerable depth, yet makes their design and analysis accessible to all levels of readers. We have tried to keep explanations elementary without sacrificing depth or mathematical rigor.

The most complex mathematical algorithms are described in detail in the appendices. Our goal here is to focus the reader's attention to the details of the topic being covered, and not be distracted by mathematical issues that can be viewed as "black box" modules having specific functionality (such as a numerical integrator or a rigid-body-mass-properties computation module). Readers should be able to develop their own software implementation of a simulation engine using the techniques covered in-depth in this book, or shorten their software development effort by taking advantage of the several resources available on the Web.

Acknowledgments

I wish to thank my wife, Izabella, and my children Leticia and Nicholas for all their support, encouragement and love.

Los Angeles, CA, USA

Murilo G. Coutinho

Contents

Part I Dynamic Simulation

1	Dynamic Simulations	3
1.1	Introduction	3
1.2	Particle and Rigid-Body Systems	4
1.3	Simulation Overview	5
1.4	A Computationally Efficient Implementation	9
1.4.1	Interface with the Rendering Engine	9
1.4.2	Moving the Objects	12
1.4.3	Detecting Collisions	14
1.4.4	Determining Collision Time for Non-convex Objects	16
1.4.5	Determining Collision Time for Thin or Fast Moving Non-convex Objects	18
1.4.6	Determining Collision Time for Convex Objects	21
1.4.7	Determining Collision Time for Thin or Fast Moving Convex Objects	22
1.4.8	Resolving Collisions	23
1.5	Guide to Readers	24
1.6	Exercises	27
2	Hierarchical Representation of 3D Polyhedra	29
2.1	Introduction	29
2.2	Hierarchical Representation of Objects	31
2.2.1	Axis-Aligned Bounding Boxes	32
2.2.2	Oriented Bounding Boxes	33
2.2.3	Bounding Spheres	36
2.2.4	Convex Hull	39
2.3	Hierarchical Representation for Continuous Collision Detection	43
2.4	Hierarchical Representation of the Simulated World	44
2.4.1	Uniform Grid	45
2.4.2	Multi-level Grid	48
2.4.3	Bounding Volume for Continuous Collision Detection	53
2.5	Collision Detection Between Hierarchical Representations	53
2.5.1	Computing Hierarchy–Hierarchy Intersections	54

2.5.2	Computing Hierarchy-Self Intersections	55
2.5.3	Computing Box-Box Intersections	56
2.5.4	Computing Sphere-Sphere Intersections	59
2.5.5	Computing Triangle-Triangle Intersections	61
2.5.6	Computing Box-Sphere Intersections	64
2.5.7	Computing Box-Triangle Intersections	65
2.5.8	Computing Sphere-Triangle Intersections	66
2.5.9	Computing Line Segment-Sphere Intersections	67
2.5.10	Computing Line Segment-Triangle Intersections	68
2.5.11	Computing Line Segment-Box Intersections	69
2.5.12	Point-in-Triangle and Edge-Edge Intersection Tests	72
2.5.13	Point-in-Object Test	74
2.5.14	Vertex-in-Object Test	75
2.5.15	Computing Continuous Triangle-Triangle Intersections	76
2.5.16	Computing Continuous Sphere-Sphere Intersections	82
2.5.17	Computing Continuous Box-Box Intersections	84
2.6	Notes and Comments	85
2.7	Exercises	86
	References	87
3	Particle Systems	89
3.1	Introduction	89
3.2	Particle Dynamics	91
3.3	Basic Interaction Forces	95
3.3.1	Gravity	96
3.3.2	Viscous Drag	97
3.3.3	Damped Springs	97
3.3.4	Spatially Dependent Forces	98
3.3.5	User Interaction	103
3.4	Collision Detection	103
3.4.1	Particle-Particle Collision	104
3.4.2	Particle-Rigid Body Collision	106
3.5	Particle-Particle Collision Response	111
3.5.1	Computing Impulsive Forces for a Single Collision	112
3.5.2	Computing Impulsive Forces for Multiple Simultaneous Collisions	120
3.5.3	Computing Contact Forces for a Single Contact	124
3.5.4	Computing Contact Forces for Multiple Contacts	132
3.6	Particle-Rigid Body Collision Response	137
3.6.1	Computing Impulsive Forces	139
3.6.2	Computing Contact Forces	140
3.7	Specialized Particle Systems	140
3.7.1	Particle Emitter	141
3.7.2	User-Definable Parameters	143
3.8	Overview of Smoothed Particle Hydrodynamics	148
3.9	Notes and Comments	153

3.10 Exercises	154
References	155
4 Rigid-Body Systems	157
4.1 Introduction	157
4.2 Rigid-Body Dynamics	158
4.3 Basic Interaction Forces	166
4.3.1 Gravity	167
4.3.2 Viscous Drag	167
4.3.3 Damped Springs	168
4.3.4 User-Interaction Forces	169
4.4 Collision Detection Overview	169
4.5 Collision Detection Between Non-convex Bodies	171
4.6 Collision Detection for Thin or Fast Moving Non-convex Bodies	174
4.7 Collision Detection Between Convex Bodies	176
4.8 Collision Detection for Thin or Fast Moving Convex Bodies	177
4.9 The Voronoi Clip Algorithm for Computing Closest Points Between Convex Objects	177
4.9.1 Feature b_2 Is a Vertex	181
4.9.2 Feature b_2 Is an Edge	182
4.9.3 Feature b_2 Is a Face	194
4.9.4 Dealing with Interpenetration	194
4.9.5 Avoiding Local Minima	194
4.10 The GJK Algorithm for Computing Closest Points Between Convex Objects	196
4.10.1 Termination Condition	203
4.11 Rigid Body–Rigid Body Collision Response	204
4.11.1 Computing Impulsive Forces for a Single Collision	205
4.11.2 Computing Impulsive Forces for Multiple Collisions	217
4.11.3 Computing Contact Forces for a Single Contact	223
4.11.4 Computing Contact Forces for Multiple Contacts	231
4.12 Particle–Rigid Body Contact Revisited	236
4.13 Notes and Comments	239
4.14 Exercises	241
References	242
5 Articulated Rigid-Body Systems	245
5.1 Introduction	245
5.2 Articulated Rigid-Body Dynamics	246
5.3 Single Joint Systems	252
5.3.1 Spherical Joint	255
5.3.2 Universal Joint	256
5.3.3 Revolute Joint	258
5.3.4 Cylindrical Joint	261
5.3.5 Prismatic Joint	263
5.3.6 Rigid Joint	264

5.4	Multiple Joint Systems	265
5.5	Collision Detection	268
5.6	Collision Response	270
5.6.1	Computing Impulsive Forces for Single or Multiple External Collisions	271
5.6.2	Computing Contact Forces for Single or Multiple External Contacts	273
5.7	Notes and Comments	275
5.8	Exercises	276
	References	277

Part II Mathematical Toolset

6	Appendix A: Useful 3D Geometric Constructions	281
6.1	Introduction	281
6.2	Projection of a Point on a Line	282
6.3	Projection of a Point on a Plane	282
6.4	Intersection of a Line Segment and a Plane	283
6.5	Closest Point Between a Line and a Line Segment	284
6.6	Computing the Collision- or Contact-Local Frame from the Collision- or Contact-Normal Vector	286
6.7	Representing Cross-Products as Matrix–Vector Multiplication	287
6.8	Interpolating Positions and Orientations	288
6.9	Suggested Readings	289
	References	289
7	Appendix B: Numerical Solution of Ordinary Differential Equations of Motion	291
7.1	Introduction	291
7.2	Euler Method	294
7.2.1	Explicit Euler	294
7.2.2	Implicit Euler	295
7.3	Runge–Kutta Method	298
7.3.1	Second-Order Runge–Kutta Method	298
7.3.2	Forth-Order Runge–Kutta Method	300
7.4	Using Adaptive Time-Step Sizes to Speed Computations	302
7.5	Suggested Readings	305
	References	305
8	Appendix C: Quaternions	307
8.1	Introduction	307
8.2	Basic Quaternion Operations	308
8.2.1	Addition	308
8.2.2	Dot product	308
8.2.3	Multiplication	308
8.2.4	Conjugate	310
8.2.5	Module	310

8.2.6	Inverse	310
8.3	Unit Quaternions	311
8.3.1	Rotation-Matrix Representation Using Unit Quaternions	311
8.3.2	Advantages of Using Unit Quaternions	313
8.4	Suggested Readings	315
	References	315
9	Appendix D: Rigid-Body Mass Properties	317
9.1	Introduction	317
9.2	Mirtich’s Algorithm	318
9.2.1	Volume-Integral to Surface-Integral	321
9.2.2	Surface-Integral to Projected-Surface-Integral	322
9.2.3	Projected-Surface-Integral to Line-Integral	325
9.2.4	Computing Line Integrals from Vertex Coordinates	327
9.3	Suggested Readings	328
	References	328
10	Appendix E: Useful Time Derivatives	329
10.1	Introduction	329
10.2	Computing the Time Derivative of a Vector Attached to a Rigid Body	329
10.3	Computing the Time Derivative of a Contact-Normal Vector	330
10.3.1	Particle–Particle Contact	330
10.3.2	Rigid Body–Rigid Body Contact	331
10.4	Computing the Time Derivative of the Tangent Plane	333
10.5	Computing the Time Derivative of a Rotation Matrix	334
10.6	Computing the Time Derivative of a Unit Quaternion	335
10.7	Suggested Readings	339
	References	339
11	Appendix F: Convex Decomposition of 3D Polyhedra	341
11.1	Introduction	341
11.2	Joe’s Algorithm	343
11.2.1	Determining Candidate Cut Planes	344
11.2.2	Computing the Cut Face of a Cut Plane	346
11.2.3	Termination Conditions	348
11.3	Suggested Readings	349
	References	349
12	Appendix G: Constructing Signed Distance Fields for 3D Polyhedra	351
12.1	Introduction	351
12.2	A Memory Efficient Signed Distance Field	352
12.2.1	Computing the Grid Cell Size	353
12.2.2	Scanline Rasterization	354
12.2.3	Computing the Signed Distance at a Grid Vertex	356
12.2.4	Computing the Signed Distance at a Point	358
12.2.5	Computing the Gradient at a Point	360

12.2.6	Computing the Closest Point on the Zero Iso-contour . . .	361
12.3	Detecting Collisions	361
12.4	Resolving Collisions	362
12.5	Suggested Readings	364
	References	364
13	Appendix H: Conservative Time Advancement for Convex Objects .	365
13.1	Introduction	365
13.2	Computing the Conservative Time Advancement	366
13.3	Suggested Readings	368
	References	368
14	Appendix I: The Linear-Complementarity Problem	369
14.1	Introduction	369
14.2	Dantzig’s Algorithm: The Frictionless Case	371
14.2.1	Termination Conditions	378
14.3	Baraff’s Algorithm: Coping with Friction	379
14.3.1	Static-Friction Conditions	382
14.3.2	Dynamic Friction	390
14.3.3	Termination Conditions	391
14.4	Suggested Readings	392
	References	393
	Index	395