
Undergraduate Topics in Computer Science

Series Editor

Ian Mackie, University of Sussex, Brighton, UK

Advisory Editors

Samson Abramsky , Department of Computer Science, University of Oxford, Oxford, UK

Chris Hankin , Department of Computing, Imperial College London, London, UK

Mike Hinchey , Lero – The Irish Software Research Centre, University of Limerick, Limerick, Ireland

Dexter C. Kozen, Department of Computer Science, Cornell University, Ithaca, NY, USA

Andrew Pitts , Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

Hanne Riis Nielson , Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark

Steven S. Skiena, Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

Iain Stewart , Department of Computer Science, Durham University, Durham, UK

‘Undergraduate Topics in Computer Science’ (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems, many of which include fully worked solutions.

The UTiCS concept relies on high-quality, concise books in softback format, and generally a maximum of 275–300 pages. For undergraduate textbooks that are likely to be longer, more expository, Springer continues to offer the highly regarded Texts in Computer Science series, to which we refer potential authors.

More information about this series at <http://www.springer.com/series/7592>

Alexei Sourin

Making Images with Mathematics

Alexei Sourin
School of Computer Science
and Engineering
Nanyang Technological University
Singapore, Singapore

ISSN 1863-7310 ISSN 2197-1781 (electronic)
Undergraduate Topics in Computer Science
ISBN 978-3-030-69834-8 ISBN 978-3-030-69835-5 (eBook)
<https://doi.org/10.1007/978-3-030-69835-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are solely and exclusively licensed 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.

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.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my wife Olga

Preface

Aim of the Book

Information visualization creates images from abstract data by interpreting them geometrically. It may require skills of making images with raw mathematics while the graphics content is now mostly created by using sophisticated licensed software. As a result, fewer and fewer developers are capable of using procedurally based visualization in which mathematical formulas are used for defining complex geometric shapes, transformations, and motions as well as for coloring the geometry.

The book explains how to see geometry and colors beyond simple mathematical formulas and teaches how to represent geometric shapes and motions from first principles by digital sampling mathematical functions.

The book may serve as a self-contained text for a one-semester computer graphics and visualization course for computer science and engineering students, as well as a reference manual for researchers and developers.

Book Organization

The book has seven chapters.

The Chap. 1 “From Ancient Greeks to Pixels” explains how we see the world and how the computer makes images. Beginning with Ancient Greek Geometry, it travels to modern geometry, introduces the subject of computer graphics and visualization, explains how the graphics pipeline works, and how a geometric point turns into a color spot on the computer screen.

The Chap. 2 “Defining Geometric Shapes” presents the mathematical foundations of shape modeling. Curves, surfaces, and solid objects are considered as a set of points which are obtained by sampling various types of mathematical functions. Using the concept of sweeping, many varieties of shapes are defined based on only a few simple foundation principles.

The Chap. 3 “Transformations” considers how the same formulas, used for making shapes, can define their transformations. The rationale for using matrix transformations is explained, and affine and projection matrix transformations are presented. Generalization of geometric sweeping implemented with matrices is further discussed.

The Chap. 4 “Motions” explains how the previously introduced mathematical formulas, defining shapes and transformation matrices, can be extended to time-dependent models of moving shapes. Motions of rigid shapes and shape morphing transformations are considered. Besides pseudo-physical motions, definitions based on Newtonian physics are also introduced.

In the Chap. 5 “Adding Visual Appearance to Geometry”, we consider how colors can be added to geometry, and how its photorealistic appearance can be achieved. The formulas, previously used for defining geometry, now will define variable colors as a new modality of immersion into the world of geometric definitions.

In the Chap. 6 “Putting Everything Together”, the ways of making interactive, real-time, and immersive visualization environments are considered including technical and physiological design and implementation issues. Still the same transformations, and actually the same basic mathematical principles, will be used in the fast visualization methods.

Finally, the Chap. 7 “Let’s Draw” introduces to the reader a few commonly used freeware software tools—OpenGL, POV-Ray, VRML, and X3D—which will let the readers apply theoretical principles into practice without requesting expensive hardware and software solutions. Also, the readers will learn how immersive visual mathematics can be implemented using the function-based extension of VRML and X3D, which allows for defining geometric shapes and their appearances with analytical functions. Finally, the Shape Explorer tool will be presented to the reader as a multi-platform companion viewer for all the examples used in the book.

Acknowledgments

I am very grateful to the School of Computer Science and Engineering of Nanyang Technological University for the pleasure of doing research and teaching, which eventually resulted in writing this book. I am very thankful to my students and colleagues for their comments and suggestions.

Last but not least, I would like to thank Helen Desmond, Editor for Computer Science of Springer Nature, who offered continuous encouragement and support.

Singapore, Singapore
2021

Alexei Sourin

Contents

1	From Ancient Greeks to Pixels	1
1.1	Drawing with Computer	1
1.1.1	How We See the World	1
1.1.2	Displaying Images	2
1.1.3	Computer Monitors	3
1.1.4	Storing Images in Computers	5
1.2	From “Earth Measuring” to Computer Graphics	6
1.2.1	Evolution of Geometry	6
1.2.2	Computer Graphics and Beyond	8
1.3	We Need Digits to Draw with Computer	10
1.3.1	2D Cartesian Coordinates	11
1.3.2	Polar Coordinates	11
1.3.3	3D Cartesian Coordinates	14
1.3.4	Cylindrical Coordinates	15
1.3.5	Spherical Coordinates	16
1.3.6	Visualization Pipeline	16
1.4	Geometric Algebra	19
1.4.1	Geometric Modeling	19
1.4.2	Rendering Geometry to Images	20
1.4.3	Mathematical Functions in Geometric Modeling	20
1.5	Summary	22
	References	23
2	Geometric Shapes	25
2.1	Sampling Geometry	25
2.2	Points	26
2.3	Curves	26
2.3.1	Straight Line	27
2.3.2	Circle	30
2.3.3	Ellipse	33
2.3.4	Plethora of Curves	37
2.3.5	Three-Dimensional Curves	40

2.4	Surfaces	42
2.4.1	Plane	43
2.4.2	Polygons	48
2.4.3	Bilinear Surfaces	49
2.4.4	Quadrics	51
2.4.5	Making Surfaces by Sweeping Curves	55
2.5	Solid Objects	63
2.5.1	Defining Solids by Parametric Functions	64
2.5.2	Constructive Solid Geometry by Functions	73
2.6	Summary	82
	References	83
3	Transformations	85
3.1	Mathematics of Transformations	85
3.2	Matrix Representation of Affine Transformations	87
3.2.1	Homogeneous Coordinates	87
3.2.2	Identity Transformation	88
3.2.3	Scaling and Reflection	89
3.2.4	Shear	91
3.2.5	Rotation	91
3.2.6	Translation	94
3.3	Composition of Transformations	94
3.3.1	Rotation About a Point	95
3.3.2	Scaling and Reflection About Points and Lines in 2D Space	95
3.3.3	Deriving Matrix of an Arbitrary 2D Affine Transformation	97
3.3.4	Rotation About an Axis	97
3.3.5	Reflections About Any Point, Axis, or Plane in 3D Space	102
3.3.6	Deriving Matrix of an Arbitrary 3D Affine Transformation	104
3.3.7	Matrix Algebra Laws	105
3.3.8	Definition of Sweeping by Transformation Matrices	106
3.4	Projection Transformations	107
3.4.1	Implementations of Projection Transformations	108
3.4.2	Classifications of Projection Transformations	109
3.4.3	Parallel Orthographic Projections	112
3.4.4	Axonometric Parallel Projections	112
3.4.5	Perspective Projections	112
3.4.6	Axonometric Perspective Projections	116

3.4.7	Projections on Any Plane	117
3.4.8	Viewing Frustum	118
3.4.9	Stereo Projection	118
3.5	Summary	120
	References	121
4	Motions	123
4.1	Animating Geometry	123
4.1.1	Motion of Points	123
4.1.2	Animating Shape Definitions	127
4.1.3	Time-Dependent Affine Transformations	128
4.1.4	Shape Morphing	130
4.2	Physically Based Motion Specifications	131
4.2.1	Motion with Constant Acceleration	131
4.2.2	Motion Under Gravity	134
4.2.3	Rotation Motion	138
4.3	Summary	141
	References	141
5	Adding Visual Appearance to Geometry	143
5.1	Illumination	143
5.2	Lighting	144
5.2.1	Ambient Reflection	146
5.2.2	Diffuse Reflection	146
5.2.3	Specular Reflection	147
5.2.4	Phong Illumination	148
5.3	Shading	150
5.3.1	Flat Shading	150
5.3.2	Gouraud Shading	151
5.3.3	Phong Shading	152
5.4	Shadows and Transparency	153
5.5	Textures	154
5.6	Colors by Functions	157
5.7	Summary	158
	References	160
6	Putting Everything Together	161
6.1	Interactive and Real-Time Rendering	161
6.2	Positioning the Observer	161
6.2.1	Direction Cosines	162
6.2.2	Fixed Angles	164
6.2.3	Euler Angles	166

6.3	Fast Rendering of Large Scenes	167
6.3.1	Hierarchical Representation	167
6.3.2	Current Transformation Matrix	168
6.3.3	Logical and Spatial Organizations	169
6.3.4	Bounding Boxes	171
6.3.5	Level of Detail	172
6.4	Summary	175
	References	176
7	Let's Draw	177
7.1	Programing Computer Graphics and Visualization	177
7.2	Drawing with OpenGL	177
7.2.1	Introduction to OpenGL	177
7.2.2	Interaction with GLUT	181
7.2.3	Drawing Parametric Shapes with OpenGL	182
7.2.4	Animation and Surface Morphing with OpenGL and GLUT	193
7.2.5	Interactive Solid Modeling with OpenGL and GLUT	196
7.3	Drawing with POV-Ray	212
7.3.1	The Persistence of Vision Ray Tracer	212
7.3.2	Function-Based Shape Modeling with POV-Ray	218
7.4	Drawing with VRML/X3D	221
7.4.1	Introduction to VRML	221
7.4.2	Introduction to X3D	226
7.4.3	Function-Based Extension of VRML	226
7.4.4	Function-Based Extension of X3D	232
7.5	Drawing with Shape Explorer	240
7.6	Summary	243
	References	243
	Index	245