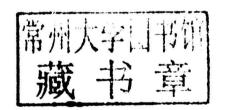


MATHEMATICAL STRUCTURES FOR COMPUTER GRAPHICS

STEVEN J. JANKE



WILEY

Copyright © 2015 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permission.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Janke, Steven J., 1947- author.

Mathematical structures for computer graphics / Steven J. Janke, Department of Mathematics, Colorado College, Colorado Spring, CO.

pages cm

Includes bibliographical references and index.

ISBN 978-1-118-71219-1 (paperback)

1. Computer graphics-Mathematics. 2. Three-dimensional imaging-Mathematics. I. Title.

T385.J355 2014 006.601'51-dc23

2014017641

Printed in Singapore

MATHEMATICAL STRUCTURES FOR COMPUTER GRAPHICS

To Deborah

PREFACE

Computer graphics includes a large range of ideas, techniques, and algorithms extending from generating animated simulations to displaying weather data to incorporating motion-capture segments in video games. Producing these images requires an array of artistic, technical, and algorithmic skills. Software can help by offering a flexible user interface, but under the hood, mathematics is orchestrating the images. Not everything in graphics begins with a mathematical result, but nearly everything is founded on mathematical ideas, because ultimately algorithms direct the computer to light up specified pixels on the screen.

The evolution of computer graphics started in the early 1970s, and since then key mathematical ideas and techniques have risen to the surface and have proved their worth in solving graphics problems. This text tries to lay out these ideas in a way that is easily accessible to those interested in a sound footing in the field and to those software engineers eager to fill in gaps where their understanding faltered.

Organized to mimic the flow of a standard graphics course, this manuscript grew from the notes for an undergraduate graphics course taught regularly over a span of 20 years. Appropriate mathematical ideas are introduced along with the details of various techniques. The style is more informal than formal, yet the approach includes thorough derivations in the hope that context and careful arguments will build confidence in constructing new approaches and new algorithms.

One or two courses in calculus should give the readers sufficient mathematical maturity to work through the text, and even if their linear algebra background is limited to matrix multiplication, they should be able to develop some useful algebraic and geometric tools. Standard mathematics courses rarely have the time to cover all the important mathematical constructs used in graphics such as the description of curves necessary for surface design, or homogeneous coordinates necessary for affine

xiv PREFACE

transformations. This text fills in those gaps and looks behind the results enough to understand how they fit into the rest of mathematics. It does not rely on the rigorous theorem/proof format, and instead uses intuition and example to develop careful results. Although the mathematics is interesting in its own right, the text hopefully does not lose sight of the ultimate goal which is to produce interesting and useful images.

There are plenty of examples and exercises to help fix the ideas and several suggestions of other directions to investigate. At the end of each chapter (except the last), there is a section titled *Complements and Details* that collects a few historical notes, several calculation details, and occasionally some ideas which may lead to interesting tangents. The text is independent of any particular graphics system, but it does have OpenGL in mind when presenting details of the viewing frustum in the chapter on visibility. Otherwise, there are programming exercises throughout, which can be done with almost any language and graphics interface.

Chapters 1–3 carefully develop vector geometry assuming very little background. They highlight the difference between vectors and points and emphasize the connection between geometry and algebra. Coordinate-free expressions and homogeneous coordinates are both introduced.

Chapters 4 and 5 examine transformations, both linear and affine. Along the way, they develop basic matrix algebra, construct various transformations including the perspective transformation, examine coordinate systems (world, local, and camera), unravel Euler angles and quaternions, and consider alternate coordinate systems.

Chapters 6 and 7 develop modeling techniques through an exploration of polygons (particularly triangles), polyhedra, parametric description of curves, Lagrange interpolation, Bézier curves, splines, nonuniform rational B-splines (NURBS), and surface construction.

Rendering is covered in Chapters 8 and 9, starting with a look at the view frustum, hidden surface algorithms, and simple ray tracing. Then an elementary lighting model is examined in detail before introducing shading, shadows, the bidirectional reflectance distribution function (BRDF), the basics of radiosity, and texture mapping.

The final chapter collects three separate mathematical techniques that represent arguably different paradigms. Bresenham's algorithm starts a discussion of pixel-based mathematics, Perlin's noise prompts a visit to random distributions, and L-systems offer an alternative algebraic description of organic forms.

When used as a course text, the first five chapters as well as selections from the last five could serve to cover an appropriate amount of material. The idea is to rely on the text for the mathematics and supplement it with algorithms perhaps specific to the available graphics systems. There are both mathematical and programming exercises in each chapter. Throughout the examples in the text, the calculation results are rounded to two or three decimal places. This still leads to round-off error, and a good exercise for the student is to reconcile any perceived discrepancies in the results.

In the way of acknowledgement, first note that most of the figures in the text were prepared using Mathematica[®]. Second, many thanks go to my graphics students over the years who prompted me to learn the nuances of the subject and who

PREFACE XV

offered constructive feedback on my courses. Thanks also go to Cory Scott whose comments on the completed manuscript were essential and to Craig Janke for cover ideas along with continued encouragement. Finally, without my wife Deborah and her unending support, this project would have dissolved on the screen.

STEVEN J. JANKE Colorado College, 2014

CONTENTS

PRI	EFACE			xiii
1	Basics			1
	1.1	Graphi	ics Pipeline, 2	
	1.2	Mathe	matical Descriptions, 4	
	1.3	Positio	on, 5	
	1.4	Distan	ce, 8	
	1.5	Compl	ements and Details, 11	
		1.5.1	Pythagorean Theorem Continued, 11	
		1.5.2	Law of Cosines Continued, 12	
		1.5.3	Law of Sines, 13	
		1.5.4	Numerical Calculations, 13	
	1.6	Exerci	ses, 14	
		1.6.1	Programming Exercises, 16	
2	Vector	Algeb	ra	17
	2.1	Basic '	Vector Characteristics, 18	
		2.1.1	Points Versus Vectors, 20	
		2.1.2	Addition, 20	
		2.1.3	Scalar Multiplication, 21	
		2.1.4	37 m 3 5 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
		2.1.5	Vector Calculations, 22	

viii CONTENTS

		2.1.6 Properties, 24	
	2.2	2.1.7 Higher Dimensions, 25	
	2.2	Two Important Products, 25	
		2.2.1 Dot Product, 25	
	2.2	2.2.2 Cross Product, 29	
	2.3	Complements and Details, 34	
		2.3.1 Vector History, 34	
		2.3.2 More about Points Versus Vectors, 35	
	2.4	2.3.3 Vector Spaces and Affine Spaces, 36	
	2.4	Exercises, 38 2.4.1 Programming Exercises, 39	
3	Vector	r Geometry	40
	3.1	Lines and Planes, 40	
		3.1.1 Vector Description of Lines, 40	
		3.1.2 Vector Description of Planes, 44	
	3.2	Distances, 46	
		3.2.1 Point to a Line, 46	
		3.2.2 Point to a Plane, 48	
		3.2.3 Parallel Planes and Line to a Plane, 48	
		3.2.4 Line to a Line, 50	
	3.3	Angles, 52	
	3.4	Intersections, 54	
		3.4.1 Intersecting Lines, 54	
		3.4.2 Lines Intersecting Planes, 56	
		3.4.3 Intersecting Planes, 57	
	3.5	Additional Key Applications, 61	
		3.5.1 Intersection of Line Segments, 61	
		3.5.2 Intersection of Line and Sphere, 65	
		3.5.3 Areas and Volumes, 66	
		3.5.4 Triangle Geometry, 68	
		3.5.5 Tetrahedron, 69	
	3.6	Homogeneous Coordinates, 71	
		3.6.1 Two Dimensions, 72	
	2.7	3.6.2 Three Dimensions, 73	
	3.7	Complements and Details, 75	
		3.7.1 Intersection of Three Planes Continued, 75	
	20	3.7.2 Homogeneous Coordinates Continued, 77 Exercises, 79	
	3.8	Non-manufacture (Manufacture (M	
		3.8.1 Programming Exercises, 82	
4	Trans	formations	83
	4.1	Types of Transformations, 84	

CONTENTS ix

4.	.2	Linear	Transformations, 85	
		4.2.1	Rotation in Two Dimensions, 88	
		4.2.2	Reflection in Two dimensions, 90	
		4.2.3	Scaling in Two Dimensions, 92	
		4.2.4	Matrix Properties, 93	
4.	.3	Three I	Dimensions, 95	
		4.3.1	Rotations in Three Dimensions, 95	
		4.3.2	Reflections in Three Dimensions, 101	
		4.3.3	Scaling and Shear in Three Dimensions, 102	
4.	4		Transformations, 103	
		4.4.1	Transforming Homogeneous Coordinates, 105	
		4.4.2	Perspective Transformations, 107	
		4.4.3	Transforming Normals, 110	
		4.4.4	Summary, 111	
4.	5		ements and Details, 112	
		4.5.1	Vector Approach to Reflection in an Arbitrary Plane, 113	
		4.5.2	Vector Approach to Arbitrary Rotations, 115	
4.	6		es, 121	
		4.6.1	Programming Exercises, 123	
0	rien	tation		124
5.			an Coordinate Systems, 125	
5.	2	Camera		
		5.2.1	Moving the Camera or Objects, 134	
		5.2.2	Euler Angles, 137	
		5.2.3	Quaternions, 141	
		5.2.4	Quaternion Algebra, 143	
		5.2.5	Rotations, 145	
		5.2.6	Interpolation: Slerp, 148	
		5.2.7	From Euler Angles and Quaternions to Rotation Matrices,	151
5	3		Coordinate Systems, 152	
		5.3.1	Non-orthogonal Axes, 152	
		5.3.2	Polar, Cylindrical, and Spherical Coordinates, 154	
		5.3.3	Barycentric Coordinates, 157	
5.4	4		ements and Details, 158	
			Historical Note: Descartes, 158	
		5.4.2	Historical Note: Hamilton, 158	
		5.4.3	Proof of Quaternion Rotation, 159	
5.5	5	Exercise		
		5.5.1	Programming Exercises, 163	
	_	200		
Po	olygo	ns and l	Polyhedra	164
6.	1	Triangle	es. 164	
			The state of the s	

x CONTENTS

	6.1.1	Paragantria Coordinates 165	
	6.1.2	Barycentric Coordinates, 165 Areas and Barycentric Coordinates, 166	
	6.1.3	Interpolation, 171	
	6.1.4	Key Points in a Triangle, 172	
6.2		ons, 178	
0.2	6.2.1	Convexity, 179	
	6.2.2	Angles and Area, 180	
	6.2.3		
	6.2.4	Triangulation, 187	
	6.2.5		
6.3		Delaunay Triangulation, 189 dra, 192	
0.3	6.3.1		
	6.3.2	Regular Polyhedra, 194	
	6.3.3	Volume of Polyhedra, 196 Euler's Formula, 200	
	6.3.4	*	
6.1		Rotational Symmetries, 202	
6.4	6.4.1	ements and Details, 205 Generalized Barycentric Coordinates, 205	
	6.4.2	•	
6.5		end the winds of the first of t	
0.5		ses, 208	
	6.5.1	Programming Exercises, 211	
Curve	2 hae ac	urfaces	212
Curve	es and S		212
7.1		Descriptions, 213	212
	Curve 7.1.1	Descriptions, 213 Lagrange Interpolation, 218	212
	Curve 7.1.1 7.1.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222	212
	Curve 7.1.1 7.1.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223	212
7.1	Curve 7.1.1 7.1.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235	212
7.1	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240	212
7.17.27.3	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242	212
7.1 7.2 7.3	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs,	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242	212
7.1 7.2 7.3 7.4 7.5	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Splir 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs, Surface	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242 , 246 es, 250	212
7.1 7.2 7.3	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs, Surfac Comple	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242 , 246 es, 250 lements and Details, 260	212
7.1 7.2 7.3 7.4 7.5	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs, Surfac Compl 7.6.1	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242 , 246 es, 250 lements and Details, 260 Adding Control Points to Bézier Curves, 260	212
7.1 7.2 7.3 7.4 7.5 7.6	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs, Surfac Compl 7.6.1 7.6.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242 , 246 es, 250 lements and Details, 260 Adding Control Points to Bézier Curves, 260 Quadratic B-Spline Blending Functions, 262	212
7.1 7.2 7.3 7.4 7.5	Curve 7.1.1 7.1.2 Bézier 7.2.1 7.2.2 7.2.3 7.2.4 B-Spli 7.3.1 7.3.2 7.3.3 7.3.4 Nurbs, Surfac Compl 7.6.1 7.6.2	Descriptions, 213 Lagrange Interpolation, 218 Matrix Form for Curves, 222 Curves, 223 Properties for Two-Dimensional Bézier Curves, 226 Joining Bézier Curve Segments, 228 Three-Dimensional Bézier Curves, 229 Rational Bézier Curves, 230 nes, 232 Linear Uniform B-Splines, 233 Quadratic Uniform B-Splines, 235 Cubic Uniform B-Splines, 240 B-Spline Properties, 242 , 246 es, 250 lements and Details, 260 Adding Control Points to Bézier Curves, 260	212

7

CONTENTS xi

8	Visibi	ility	267
	8.1	Viewing, 267	
	8.2	Perspective Transformation, 269	
		8.2.1 Clipping, 273	
		8.2.2 Interpolating the z Coordinate, 275	
	8.3	Hidden Surfaces, 278	
		8.3.1 Back Face Culling, 281	
		8.3.2 Painter's Algorithm, 283	
		8.3.3 Z-Buffer, 286	
	8.4	Ray Tracing, 287	
		8.4.1 Bounding Volumes, 289	
		8.4.2 Bounding Boxes, 289	
		8.4.3 Bounding Spheres, 291	
	8.5	Complements and Details, 293	
		8.5.1 Frustum Planes, 293	
		8.5.2 Axes for Bounding Volumes, 294	
	8.6	Exercises, 297	
		8.6.1 Programming Exercises, 298	
9	Lighti	ing	299
	_		
	9.1	Color Coordinates, 299	
	9.2	Elementary Lighting Models, 303	
		9.2.1 Gouraud and Phong Shading, 307	
		9.2.2 Shadows, 311	
	0.2	9.2.3 BRDFs in Lighting Models, 315	
	9.3	Global Illumination, 319	
		9.3.1 Ray Tracing, 319	
	9.4	9.3.2 Radiosity, 323	
	9.4	Textures, 325 9.4.1 Mapping, 325	
		11 6	
		9.4.2 Resolution, 332	
	0.5	9.4.3 Procedural Textures, 333	
	9.5	Complements and Details, 335	
		9.5.1 Conversion between RGB and HSV, 335	
		9.5.2 Shadows on Arbitrary Planes, 3369.5.3 Derivation of the Radiosity Equation, 337	
	9.6	Exercises, 339	
	9.0		
		9.6.1 Programming Exercises, 340	
10	Other	r Paradigms	341
	10.1	Pixels, 342	
		10.1.1 Bresenham Line Algorithm, 342	

xii CONTENTS

	10.1.2 Anti-Aliasing, 345	
	10.1.3 Compositing, 347	
10.2	Noise, 350	
	10.2.1 Random Number Generation, 350	
	10.2.2 Distributions, 351	
	10.2.3 Sequences of Random Numbers, 353	
	10.2.4 Uniform and Normal Distributions, 354	
	10.2.5 Terrain Generation, 356	
	10.2.6 Noise Generation, 357	
10.3	L-Systems, 361	
	10.3.1 Grammars, 362	
	10.3.2 Turtle Interpretation, 363	
	10.3.3 Analysis of Grammars, 365	
	10.3.4 Extending L-Systems, 367	
10.4	SALES	
	10.4.1 Programming Exercises, 369	
APPENDI	X A Geometry and Trigonometry	370
A.1	Triangles, 370	
A.2	Angles, 372	
	Trigonometric Functions, 373	
		256
APPENDI	X B Linear Algebra	376
B.1	Systems of Linear Equations, 376	
	B.1.1 Solving the System, 377	
B.2		
B.3		
2.0	,	
REFEREN	ICES	383
TALL DIVIN		
INDEX		387

1

BASICS

It is rather amazing that a finite rectangular array of colored dots (called *pixels* as an abbreviation of picture elements) is sufficient to display the nearly limitless collection of images we recognize as realistically or symbolically representing portions of our world. The power of combinatorics helps us to explain the situation (millions of possible colors for each pixel in the large display array), but we can hardly conceive of all the images we have already seen let alone those that are yet to be seen. From this reductionist viewpoint, the whole idea of computer graphics is to set the right pixels to the right color. Easier said than done. Yes, a plain red square is easy, but one that looks like it is made of bricks is tougher, and one that includes a human face taxes the best of known algorithms.

Of course, the computer graphics enterprise includes any and all manipulations of images. We can start from scratch and produce a photo-realistic image of a new airliner or perhaps construct a landscape design complete with a variety of plants. Maybe the challenge is to translate CAT (computerized axial tomography) scan data into an image of the brain or correct the color balance in a photo being readied for publication. To bring some order to the very long list of possibilities, it is helpful to consider two main categories: either we are generating images, or we are processing existing images. Both require mathematical tools, but the first category encompasses the broad mathematical approaches necessary to understand three-dimensional descriptions of objects and their interactions with light. The second category starts with an image and draws on the mathematics of transformations and filters necessary to convert it into a more useful visual representation. In this survey of mathematical tools that are

2 BASICS

useful in computer graphics, we will focus on the first category where we can start with the basics of mathematical descriptions and work through the generation and manipulation of objects in space.

1.1 GRAPHICS PIPELINE

As we examine the steps necessary to produce a new image on the computer screen, we are tracing what is often called the *graphics pipeline*. The pipeline analogy is intended to highlight the stages we go through both in designing images and in processing them on the computer to produce the final properly colored array of pixels on the display screen. As one frame is being completed, the next is making its way down the pipeline. Most modern hardware includes the main microprocessor (central processing unit, CPU), the graphics microprocessor (graphics processing unit, GPU), and various associated memory banks. The CPU and GPU work in parallel, as the CPU supplies descriptions of objects to the GPU which in turn processes the descriptions to determine which pixels on the screen need to be turned on. The exact order of all the required steps depends on the hardware and on the graphics software we use. However, we can make a more general description of the pipeline to enumerate the stages of image generation and set the context for understanding the associated mathematics. Our pipeline then looks like this:

- 1. Modeling. We need a mathematical description of objects, background, and light sources as well as a description of their placement in a scene. For more primitive objects such as buildings which are more or less constructed out of simple plane surfaces, the description includes a list of vertices and a list showing which vertices determine individual faces. For curved surfaces, we may attempt an accurate description (e.g., a sphere) or rely on an approximation with small flat triangles. These descriptions are, of course, just the beginning, as we need also to know the details of how the objects are placed in a scene and how light will interact with them. Mathematically, a geometric description including vertices and faces (surfaces) forms the kernel of our model, but certainly if the object is a tree or if there is fog affecting the lighting, the description may well require a deeper extension of the standard high school geometry. This modeling stage can be done with design software, allowing artists to manipulate the scene to reach the desired effect.
- 2. Transformation. Building a scene requires positioning objects relative to each other and includes rotation, scaling, and translation. Transformations reposition an object and convert its coordinate descriptions appropriately. Then, to view the scene, imagine a camera placed somewhere in space looking in a particular direction. (Alternatively, imagine your eye positioned in space looking at the scene.) Another transformation adjusts the mathematical descriptions so that they are relative to the camera position.
- 3. *Visibility*. Depending on where the camera is, we may not see the entire scene. Rather, some parts are outside the field of view and consequently can be ignored