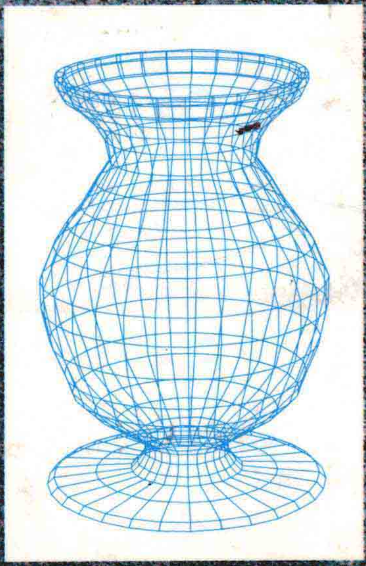
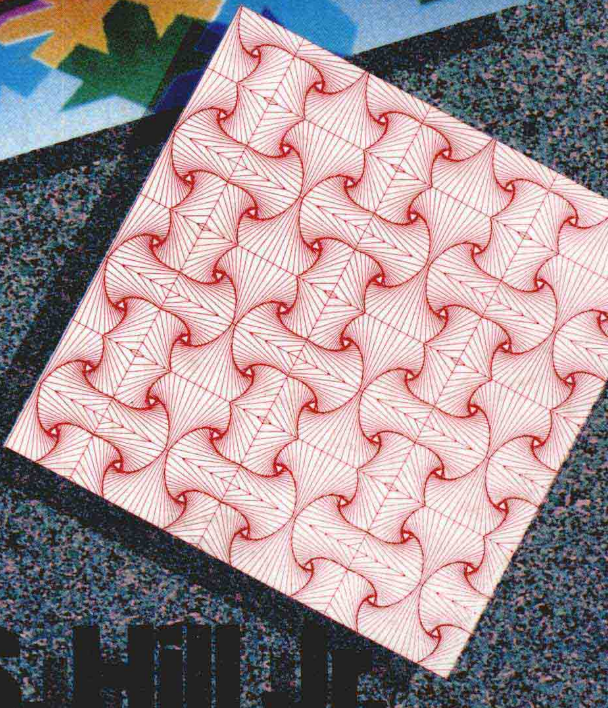
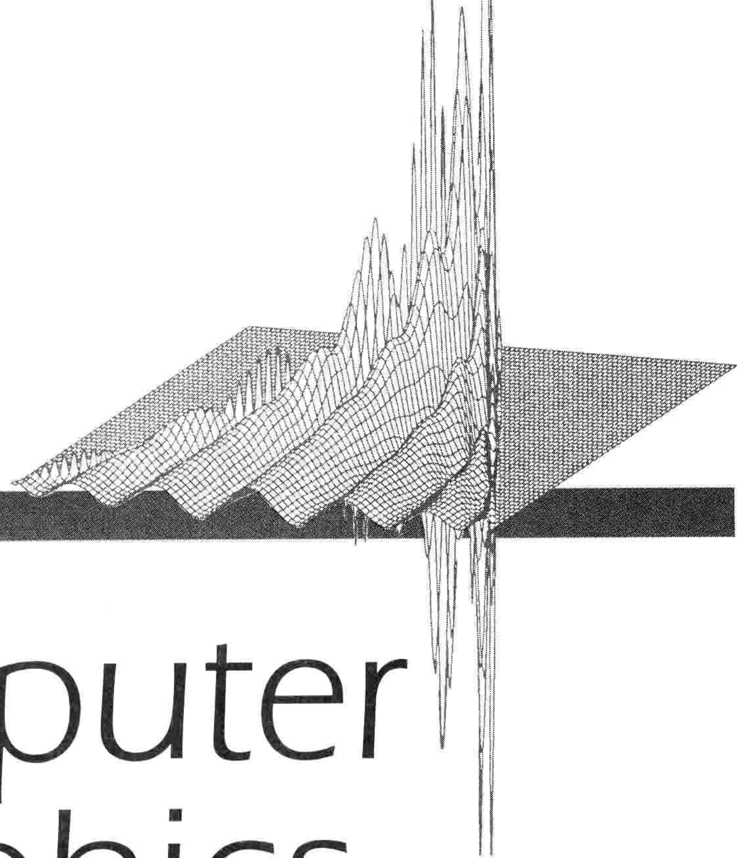
COMPUTER GRAPHICS





F.S. Hill Jr.



Computer Graphics

Francis S. Hill, Jr.

Department of Electrical and Computer Engineering

Macmillan Publishing Company

New York

Collier Macmillan Publishers

London

Editor: John Griffin
Production Supervisor: Ron Harris
Production Manager: Pam Kennedy Oborski
Text Designer: Jack Ehn
Cover Designer: Sheree L. Goodman

This book was set in 10/12 Melior by T.S.I. Graphics and printed and bound by Von Hoffmann Press. The cover was printed by Von Hoffmann Press.

Printed in the United States of America

Macmillan Publishing Company
866 Third Avenue, New York, New York 10022

Collier Macmillan Canada, Inc.

Library of Congress Cataloging-in-Publication Data
Hill, Francis S.
Computer graphics / Francis S. Hill, Jr.
p. cm.
Bibliography: p.
Includes index.
ISBN 0-02-354860-6
1. Computer graphics. I. Title.
T385.H549 1990 89-31610
006.6—dc20 CIP

Printing: 1 2 3 4 5 6 7 8 Year: 0 1 2 3 4 5 6 7 8 9

TO
Merilee
and to
Greta, Jessie, and Rosy

Preface

Computer graphics is a fascinating area of computer science. It is widely used as a tool for visualizing information in a broad variety of fields, including science and engineering, medicine, architecture, and entertainment. Interactive graphics programs let people work with computers in a natural manner: a user can supply information to the program through simple hand movements and receive information back through pictures. Computer graphics is helping to change the way people perceive and use computers, and it is available on even the lowest-cost computers today, making it readily accessible to everyone.

This book teaches the concepts and techniques of computer graphics. It teaches people how to write programs that produce graphical pictures and images of many kinds of information. It is designed as a text for either a one- or two-semester course in computer graphics at the junior, senior, or graduate level. It can also be used for self-study. It is aimed principally for students majoring in computer science or engineering but will also suit students in other fields, such as physics, mathematics, business, and art.

Prerequisites

In general, the reader should have at least one semester of experience writing computer programs in a modern language such as Pascal, C, or FORTRAN. Some experience with elementary data structures such as linked lists is desirable but not essential, since alternative approaches based on arrays are described.

The reader should have the equivalent of one year of college mathematics; knowledge of elementary algebra, geometry, and trigonometry is assumed. Some exposure to calculus and matrices is useful but not necessary, as an appendix introduces those parts of matrix theory that are used in the book.

Philosophy

The basic philosophy of this book is that graphics programming is learned by doing it: One must write and test real programs that drive real graphics devices to comprehend fully what is going on. One of the principal goals of the book is to show readers how to translate geometric concepts first into a clear mathematical expression and then into program code that works on a computer. Readers first learn how to develop simple routines to produce pictures on whatever graphics display devices they have available. Then methods for producing drawings of ever more complex objects and data are presented in a step-by-step fashion.

Code Fragments

Over 140 examples of tested code fragments, written in Pascal, serve as examples to show one way to implement an algorithm. These fragments appear throughout the text and illustrate real-life applications of the theory.

Exercises and Problems

More than 440 drill exercises appear throughout the book. These are of the "stop-and-think" variety that don't require programming and allow readers to self-test their grasp of the material. Over 180 programming problems appear at the end of chapters. These are suitable for homework assignments and range from the simple to the challenging.

Illustrations

Numerous two- and four-color drawings, photographs, and stereo drawings are used throughout the text to illustrate technique and theory. Computer graphics is of course a very visual discipline, and the figures help to fine-tune the reader's ability to visualize geometric relationships. This is particularly true for three-dimensional graphics, where the stereo pictures help enormously to reveal the geometric concepts being discussed. For suggestions on how to view the stereo pictures, please see the note that precedes the table of contents.

Device Independence

A college graphics laboratory may contain a sizable number of graphics devices, but only a few of each type or model. It can be frustrating for the instructor and student alike to write different programs for so many idiosyncratic devices. To reduce such frustration levels this book takes a device-independent approach. All the graphics in the first part of the book are built out of applications of a single short routine called *lineNDC*( ) that draws a line between two points. All programming of applications accesses the graphics capability of a device through this routine. A different version of this routine—called a device driver—is fashioned for each type of device. The student can write programs without having to know beforehand which devices will be available. To test a program the student finds the nearest available graphics device, activates the appropriate driver, and is ready to run the application.

Appendix 1 gives examples of actual simple device drivers for several classes of graphics output devices. (Additional ones are provided in the accompanying instructor's manual.) This information is useful in helping the reader or instructor over the first hurdle of producing simple pictures on whatever devices are available.

Complete, Up-to-Date Organization

Computer graphics equipment changes rapidly: Each year we see ever more dazzling displays and ever faster computations. But while pictures of equipment and the images they produce slip quickly out of date, the underlying theory and mathematics of computer graphics has become very stable. The body of knowledge required to produce effective graphics programs rests on a small set of key mathematical topics, including vectors, parametric representations, affine transformations, homogeneous coordinates, and projections. Great care has been given in this book to both clarifying and exposing the beauty of these topics.

Flexible Organization

There is much more in this book than can be covered in a one-semester course. The book has been arranged so the instructor can select different groups of chapters for close study, depending on the interests and backgrounds of a class.

Several such paths through the book are suggested here.

- For a one-semester undergraduate course: Chapters 1, 2, 3, and 4 and parts of 7, 10, 11, and 12.
- For a two-semester undergraduate course add the rest of Chapters 7, 9, 15, 16, 17, and 18.
- For a one-semester graduate course: Chapters 1, 2, 3, 4, 6, 7, 9, 10, 11, and 12.
- For a two-semester graduate course add Chapters 8, 13, 15, 16, 17, and 18.
- For those interested in emphasizing modeling and 3D graphics, this material is concentrated in Chapters 10, 12, 15, 17, and 18.
- For emphasis on raster graphics, Chapters 13, 15, 16, 17, and 18 are of particular interest.

All suggested paths include Chapters 1, 2, and 3 as fundamental. Chapter 1 can be read quickly, with the instructor focusing early on notions of NDC, device drivers, windows and viewports, and producing simple line drawings. The crucial parts of Chapter 4 are dealing with polygons and the important technique of representing a curve parametrically. The material of Chapter 5 is not essential to the development of later ideas, although many students—particularly those with an interest in the arts—vastly enjoy producing such pictures. Chapter 6 is important for gaining an understanding of interactive graphics, but if an instructor prefers to focus only on the production of pictures this chapter can be omitted.

Chapter 7 contains essential information on the use of vectors in graphics. Even students already versed in vectors should peruse it to see how valuable ordinary vector tools are in graphics algorithms. Chapter 8 makes a good project for students interested primarily in two-dimensional graphics. Chapter 9 is important for an understanding of smoothly varying surfaces but may be omitted if interest is greatest in viewing faceted models. Chapter 10 introduces the fascinating world of three-dimensional (3D) graphics based on wireframe models.

Chapter 11 discusses in detail another pivotal tool in graphics, the affine transformation, and shows how it is applied in graphics applications. And Chapter 12 moves deeper into 3D graphics, introducing the important "synthetic camera," and developing the various types of projections used in graphics.

In Chapter 13 we change gears and discuss a variety of topics peculiar to raster graphics, including scan conversion, region filling, antialiasing, and the powerful BitBlt operation. Chapter 14 discusses curve and surface design based on Bezier and B-spline methods.

Chapter 15 shows how solid objects that are modeled as "polygon meshes" can be drawn with smooth shading to achieve greater realism. The classical lighting models, which involve diffuse and specular components of light, are introduced, and Gouraud and Phong shading are described.

Methods for defining and classifying colors numerically are discussed in Chapter 16, leading to the CIE standard chromaticity diagram and its various uses. Chapter 17 attacks the surface elimination problem and presents several specific algorithms in detail.

In Chapter 18 we develop a complete working ray tracer, for achieving stunning visual realism. The development is incremental, so the reader can build it in small working pieces. The chapter also shows how to include the effects of mirrorlike and transparent objects.

Supplements

An accompanying instructor's manual provides solutions to most exercises and suggests additional projects. Complete demonstration programs of techniques developed in the text are explained and listed. Reprints of several articles on interesting geometric ideas provide the instructor with engaging background material for the student. Also included in the manual are guides to additional device drivers, to help the instructor in developing drivers for a wider variety of graphics device.

A diskette containing many of the code fragments and demonstration programs accompanies the instructor's manual. The demonstration programs are suitable for use in the classroom, and students can work with them and enhance them as well.

Acknowledgments

This book has grown out of notes used in a course I have been teaching at the University of Massachusetts for the last nine years. During this time a large number of students have helped to develop demonstrations and make suggestions for improving the courses. They have also produced many exquisite graphical samples, some of which appear here. Some students who have been particularly helpful are Tarik Abou-Raya, Earl Billingsley, Dennis Chen, Scott Davidson, Daniel Dee, Brett Diamond, Bruce Filgate, Jay Greco, Marc Infield, Tom Kopec, Adam Lavine, Tuan Le, Andreas Meyer, John Michael, David Mount, Bruce Nichol, Mike Purpura, Chris Russell, Russell Turner, Bill Verts, Shel Walker, Ken Ward, and Clay Yost.

I apologize for any inadvertent omissions.

Several colleagues have provided inspiration and guidance during the germination of the book. I am particularly grateful to Charles Hutchinson for his support in starting the graphics effort at the university, to Michael Wozny for his enthusiasm and encouragement in its development, and to Charlé Rupp for the many creative ideas in graphics he passed on to me. I would especially like to thank Daniel Bergeron, Robert Wilke, Tim Clement, and Brian Barsky, who made substantial contributions to the coherence and readability of the book. I would like to thank the following individuals, and many others who are not mentioned by name, for their advice and help: Marc P. Armstrong, University of Iowa; J. Eugene Ball, University of Delaware; Brian A. Barsky, University of California–Berkeley; Marc Berger, University of Colorado, R. Daniel Bergeron, University of New Hampshire; Tim Clement, University of Manchester; John T. Demel, The Ohio State University; Edward N. Ferguson, University of Maine; Georges Grinstein, University of Lowell; Mark W. Koch, Clarkson University; Leo J. LaFrance, New Mexico State University; Peichung F. Lai, University of Alabama; Ralph E. Lee, University of Missouri–Rolla; Joel Neisen, University of Minnesota; Chuck Nelson, University of Idaho; Spencer W. Thomas, University of Michigan; Deborah Walters, State University of New York–Buffalo; Frank G. Walters, University of Missouri–Rolla; Robert Wilke, The Ohio State University.

Special thanks to my editor, John Griffin, for his continuing guidance and encouragement during the preparation of the book, and to Ron Harris, whose expertise and care during production have markedly improved it. Anne Dolan-Niles helped enormously organizing the voluminous correspondence connected with the book, and Andy Casiello produced many of the photographs.

Finally, thanks to my parents, to my wife Merilee, and to our children Greta, Jessie, and Rosy, for all their patience and support while this book slowly took shape.

F. S. H., Jr.

Note to the reader
How to View the Stereo Pictures

It can be challenging to visualize things in three-dimensional (3D) space, and figures in a book are limited to showing projections of objects on a flat page. One way to counter this limitation dramatically is to view images stereoscopically. The eye–brain system is remarkable: If slightly different views of a scene are shown to the left and right eyes, a scene can appear to "jump out of the page" with a convincing sense of depth.

Many stereoscopic figures appear throughout the book to clarify discussions of 3D situations. They look like a pair of almost identical figures placed side by side. To gain the full value of these stereo pictures allow (coerce) your left eye to look at the left-hand one and your right eye to look at the right-hand one. This may take some practice, since we go about our lives focusing both eyes on the same points. Some people catch on quickly; others, after many bleary-eyed attempts; some people, never. Of course the figures still help to clarify the discussion even without the stereo effect.

One way to practice is to hold the index fingers of each hand upright in front of you, about 2 inches apart, and to stare "through them" at a blank wall in the distance. Each eye sees two fingers, of course, but two of the fingers seem to overlap in the middle. This overlap is precisely what is desired when looking at stereo figures: Each eye sees two figures, but the middle ones are brought into perfect overlap. When the middle ones fuse together like this, the brain constructs out of them a single 3D image. Some people find it helpful to place a piece of white cardboard between the two figures and to rest their nose on it. The cardboard barrier prevents each eye from seeing the image intended for the other eye.

An inexpensive cardboard viewer is available from the Taylor–Merchant Corporation; 212 West 35 Street; New York, NY 10001. The instructor can request that the college bookstore stock these viewers.

Brief Contents

Appendixes

Contents