Texts in Computer Science

Editors
David Gries
Fred. B. Schneider

For other titles published in this series, go to
www.springer.com/series/3191

David Salomon

The Computer Graphics Manual

Springer

Prof. David Salomon (emeritus)
Department of Computer Science
California State University
Northridge, CA 91330-8281, USA
dsalomon@csun.edu

Series Editors
David Gries
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

Fred. B. Schneider
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

ISSN 1868-0941 e-ISSN 1868-095X
ISBN 978-0-85729-885-0 e-ISBN 978-0-85729-886-7
DOI 10.1007/978-0-85729-886-7
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2011937970

Printed on acid-free paper

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

Plate A.1. Water Splash in a Green Garden (Modo).

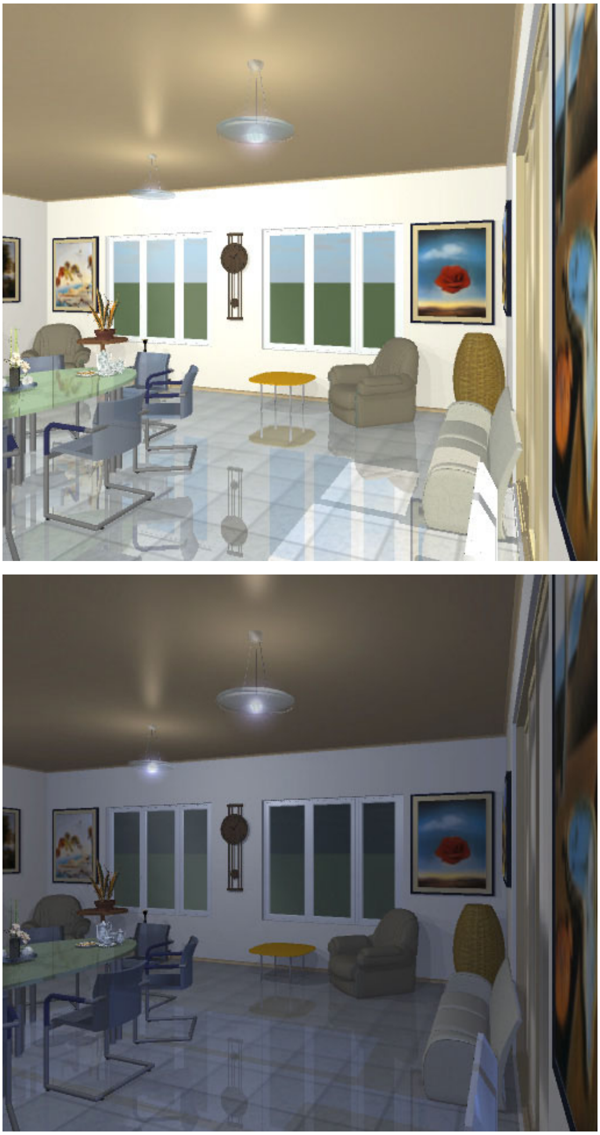


Plate A.2. A Cluttered Room, Day and Night (Live Interior).

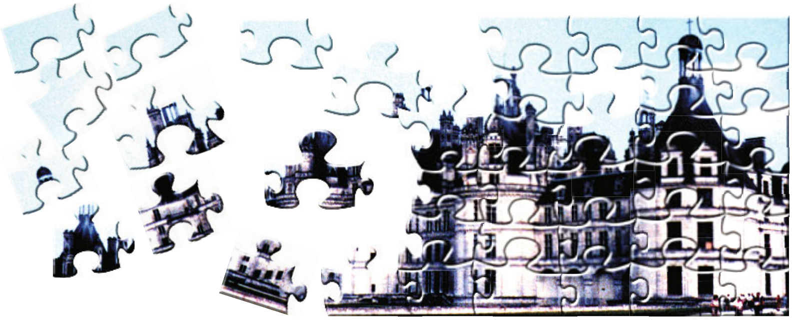


Plate A.3. A Jigsaw Puzzle of Chateau Chambord (AV brothers).

Plate A.4. The Mandelbrot Set and Two Details (Fractal Domains).

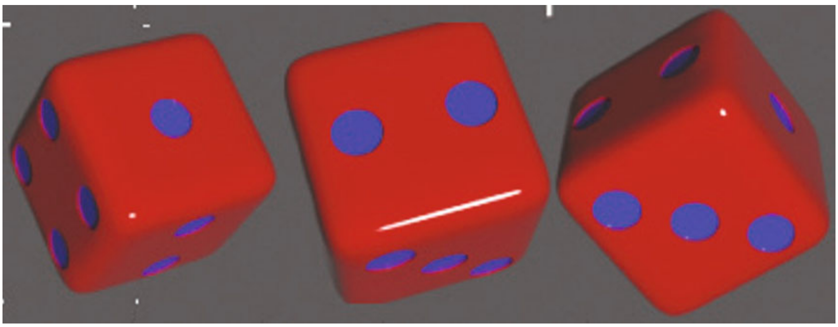


Plate A.5. Three Shiny Dice (Modo).

Plate A.6. POVRAY Textures (MegaPOV).

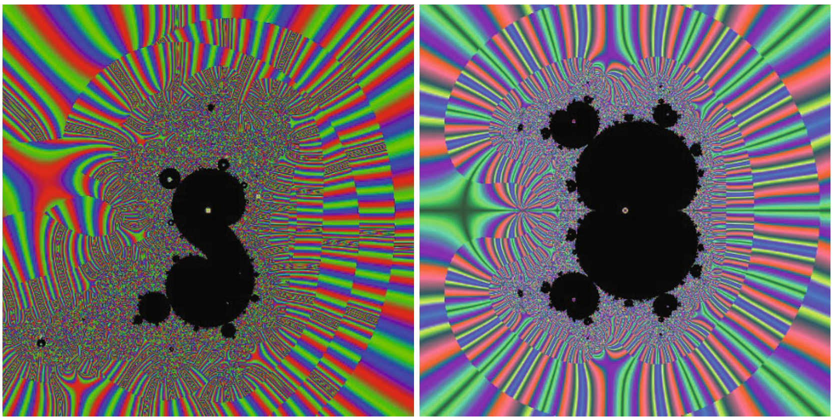
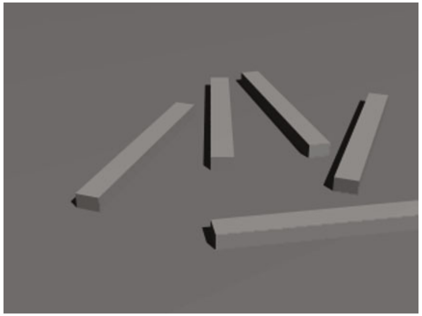
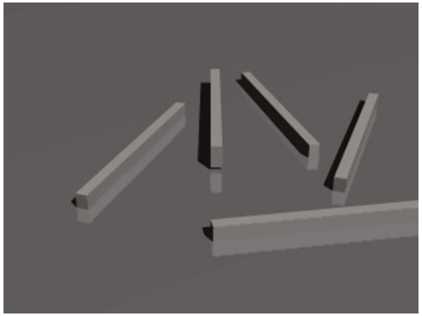


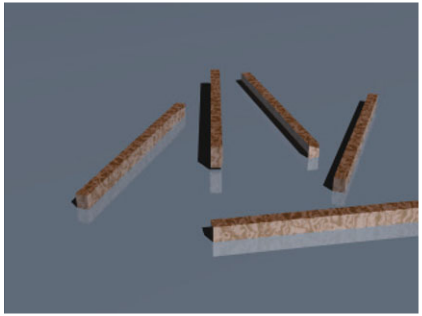
Plate A.7. Rational Fractals (Fractal Domains).



Start with a few rods



Make the floor shiny



Add texture

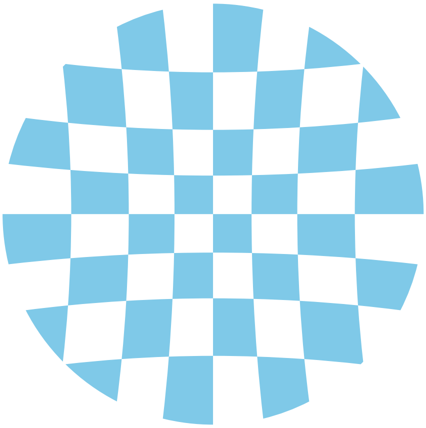
Add a wooden floor

Vary the lighting

Add a light source

Plate A.8. Steps to Visually Improve a Simple Scene (MegaPOV).

To users of computer graphics everywhere



There's no better sensation than image. It's so in-your-face!

—Jimmy Lai

Preface

The field of quantum mechanics is the cornerstone of modern physics. This field was rapidly developed in the 1920s and 1930s by a small group of (mostly young) researchers. They generally agreed that we cannot (and indeed, should not even try to) visualize atoms, photons, and elementary particles. These objects, their attributes and their behavior are best described in terms of mathematical abstractions, not pictures. Indeed, one of the first textbooks in this area *The Principles of Quantum Mechanics* by P. A. M. Dirac, does not include a single diagram in its 314 pages.

This style of writing reflects one extreme approach to graphics, namely considering it irrelevant or even detracting as a teaching tool and ignoring it. Today, of course, this approach is unthinkable. Graphics, especially computer graphics, is commonly used in texts, advertisements, and videos to illustrate concepts, to emphasize points being discussed, and to entertain.

Our approach to graphics has been completely reversed since the 1930s, and it seems that much of this change is due to the wide use of computers. Computer graphics today is a mature, successful, and growing field. It is employed by many people for many purposes and it is enjoyed by even more people. One criterion for the maturity of a field of study is its size. When a certain discipline becomes so big that no one person can keep all of it in their head, we say that that discipline has matured (or has come of age). This is what happened to computer graphics in the last decade or so. It is now a large field consisting of many subfields such as curve and surface design, rendering methods, and computer animation. Even a person who has written a book covering the entire field cannot claim that they keep all that material in their head all the time, which is precisely the reason why textbooks are being written.

In its 357 pages, *The Principles of Quantum Mechanics* featured neither a single diagram, nor an index, nor a list of references, nor suggestions for further reading.

—Graham Farmelo, *The Strangest Man*, 2009.

Overview and Goals

Today (in 2011), the power of computer-generated images is everywhere. Computer graphics has pervaded our lives to such an extent that sometimes we don't even realize that an image we are watching is artificial. The average person comes into contact with computer graphics mostly in three areas, computers, television, and electronic devices. Current computers and operating systems are based on GUI (graphical user interface). Computer programs often display results graphically. Television programs and commercials employ sophisticated, computer-generated graphics that are often hard to distinguish from the real thing. Many television programs (mostly documentaries) and recent movies mix real actors and artificial imagery to such an extent that the viewer may find it difficult to distinguish a real object or scene from a computer-generated image. (A real actor trying to outrun a computer-generated dinosaur is a common example.) More and more digital cameras, electronic devices, and instruments have small screens that display messages, options, controls, and results in color and are often touch sensitive, enabling the user to enter commands by finger gesturing instead of from the traditional keyboard. Many cell telephones even have two screens, and some new digital cameras also feature two LCD displays.

With this in mind, the goal of this manual is to present the reader with a wide picture of computer graphics, its history and its pioneers, the hardware tools it employs, and most important, the techniques, approaches, and algorithms that are at the core of this field. Thus, this textbook/reference tries to describe as many concepts and algorithms as possible, paying special attention to the important ones.

It would have been nice to include everything in this book and title it, like other texts by the same author, *Computer Graphics: The Complete Reference*, but computer graphics has grown to a point where I cannot hope to be an authority on the entire field, which is why some readers may not find every topic, term, concept, and algorithm they may be looking for.

> New material for Volume 4 will first appear in beta-test form as fascicles of approximately 128 pages each, issued approximately twice per year. These fascicles will represent my best attempt to write a comprehensive account; but computer science has grown to the point where I cannot hope to be an authority on all the material covered in these books. Therefore I'll need feedback from readers in order to prepare the official volumes later.
>
> —Donald E. Knuth.

On the other hand, those same readers may find in this manual/textbook topics they did not know existed, which might serve as compensation.

The many examples and exercises sprinkled throughout the book enhance its usefulness. By paying attention to the examples and working out the exercises, readers will gain deeper understanding of the material.

Organization and Features

This manual is large and is organized in seven parts as follows:

Part I covers the history, basic concepts, and techniques used in computer graphics. The concepts of pixel, vector scan, and raster scan are discussed. It is shown how an image given as a bitmap of pixels can be scaled (zoomed) and rotated. Many important scan-conversion methods are explained and illustrated.

Part II is devoted to transformations and projections. It starts with the important two- and three-dimensional transformations, including translation, rotation, reflection, and shearing. This is followed by the main types of projections, namely parallel, perspective, and nonlinear.

Part III is by far the largest. It includes many methods, algorithms, and techniques employed to construct curves and surfaces, which are the building blocks of solid objects. Six important interpolation and approximation methods for curves and surfaces are included, as well as sweep surfaces and subdivision methods for surface design.

Part IV goes into advanced techniques such as rendering an object, eliminating those parts that are hidden from view, and bringing objects to life (animating them) by interpolation. Several chapters included in this part discuss the nature and properties of light and color, graphics standards and graphics file formats, and fractals.

Part V describes the principles of image compression. It concentrates on two important approaches to this topic, namely orthogonal and subband (wavelet) transforms. The important JPEG image compression standard is described in detail.

Part VI is devoted to many of the important input/output graphics devices. Chapter 26 describes them and explains their operations.

Part VII consists of appendixes, most of which discuss certain mathematical concepts.

The following **features** enhance the usefulness and appearance of this textbook:

- The powerful *Mathematica*™ and Matlab software systems are used throughout the book to implement the various concepts discussed. When a figure is computed in one of these programs, the code is listed with the figure. These codes, which are available in the book's website, are meant to be readable rather than efficient and fast, and are therefore easy to read and to modify even by inexperienced *Mathematica* and Matlab users.

- The book has many **examples**. Experience shows that examples are important for a thorough understanding of the kind of material discussed in this manual. The conscientious reader should follow each example carefully and try to work out variations of the examples. Many examples also include *Mathematica* code.

- The many **exercises** sprinkled throughout the text are not a cosmetic feature. They deal with important topics, and should be worked out. Answers are provided but they should be consulted only as a last resort.

- A **quotation** is a phrase that reflects its author's profound thoughts. Quotations and epigrams enliven a book, which is why they have been generously used throughout this manual. I hope that they add to the book and make it more interesting.

> The ability to quote is a serviceable substitute for wit.
>
> —W. Somerset Maugham.

- This manual/textbook aims to be **practical**, not theoretical. After reading and understanding a topic, the reader should be able to design and implement the concepts discussed there. The few mathematical arguments found in the book are simple, and there is no attempt to present an overall theory encompassing the entire field of computer graphics.

- An important feature of this text is the attention paid to **orphans**. Those are topics that most texts on computer graphics either mention briefly or disregard completely. Examples are perspective projections, nonlinear projections, nonlinear bitmap transformations, curves, surfaces, I/O devices, and image compression. The reader will find that this manual discusses orphans in great detail, including numerous examples and exercises.

- Most of the necessary mathematical background (such as vectors and matrices) is covered in the Appendixes. However, some math concepts that are used only once (such as the mediation operator and points vs. vectors) are discussed right where they are introduced.

The Two Volumes

This textbook/reference is big because the discipline of computer graphics is big. There are simply many topics, techniques, and algorithms to discuss, explain, and illustrate by examples. Because of the large number of pages, the book has been produced in two volumes. However, this division of the book into volumes is only technical and the book should be considered a single unit. It is wrong to consider volume I as introductory and volume II as advanced, or to assume that volume I is important while volume II is not. The volumes are simply two halves of a single, large entity.

The Color Plates

This extensive manual features more than 100 color plates, placed at the very beginning, at the end, and between individual parts. They serve to liven up the book and to illustrate many of the topics discussed. It is planned to place information about these plates in the book's website, for the benefit of readers who want to recreate or extend them. The plates were prepared over several months, using a variety of graphics software. Appendix F has more information about the plates, their content, and the graphics applications used to generate them.

A Word on Notation

It is common to represent nonscalar quantities such as points, vectors, and matrices with boldface. Here are examples of the notation used in this manual:

x, y, z, t, u, v	Italics are used for scalar quantities such as coordinates and parameters.
$\mathbf{P}$, $\mathbf{Q}_i$, $\mathbf{v}$, $\mathbf{M}$	Boldface is used for points, vectors, and matrices.
$\vec{\mathrm{CP}}$	An alternative notation for vectors, used when the two endpoints of the vector are known.

$\mathbf{P}(t)$, $\mathbf{P}(u,v)$	Boldface with arguments is used for nonscalar functions such as curves and surfaces.
$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$	Parentheses (and sometimes square brackets) are used for matrices.
$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$	Vertical bars are used for determinants.
$\lvert\mathbf{v}\rvert$	The absolute value (length) of vector $\mathbf{v}$.
$\mathbf{A}^T$	The transpose of matrix $\mathbf{A}$.
x^*, $\mathbf{P}^*$	The transformed values of scalars and points.
$f^u(u)$, $\mathbf{P}^t(t)$, $\mathbf{P}^{tt}(t)$	The derivatives (first, second,...) of scalar and vector functions.
$\frac{df(u)}{du}, \frac{d\mathbf{P}(t)}{dt}$	Alternative notation for derivatives.
$\frac{df^2(u)}{du^2}, \frac{d\mathbf{P}^2(t)}{dt^2}$	Alternative notation for higher-order derivatives.
$\frac{\partial f(u,v)}{\partial u}, \frac{\partial \mathbf{P}(u,v)}{\partial v}$	Partial derivatives.
$f(x)\rvert_{x_0}$ or $f(x_0)$	Value of function $f(x)$ at point x_0.
$\sum_{i=1}^{n} x_i$	The sum $x_1 + x_2 + \cdots + x_n$.
$\prod_{i=1}^{n} x_i$	The product $x_1 x_2 \ldots x_n$.

◇ **Exercise 1:** What is the meaning of $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4)$?

Target Audiences

The material presented here has been developed over many years of teaching computer graphics. It has been revised, updated, and distilled many times, with many figures, examples, and exercises added. The text emphasizes the simplicity of the mathematics behind computer graphics and tries to show how graphics software works and how current computer graphics can generate and display realistic-looking curves, surfaces, and solid objects. The key ideas are introduced slowly, are examined, when possible, from several points of view, and are illustrated and illuminated by figures, examples, and (solved) exercises. The discussion must employ mathematics, but it is mostly non-rigorous (no theorems or proofs) and therefore easy to grasp. The mathematical background required includes polynomials, matrices, vector operations, and elementary calculus. This manual/textbook was written with three groups of readers in mind.

- **Textbook/student**. The book can serve as the primary text for a two-semester class on computer graphics for graduate and advanced undergraduate students. The many fragments of *Mathematica* code found here may serve as a core around which students can build larger programs. The exercises are especially valuable for students, and the lack of rigorous theorems and proofs should encourage those who consider computer science distinct from mathematics.

- **Reference/professional use**. Professionals in engineering, computers, and other scientific disciplines generate, watch, and process digital images all the time. They often would like to know more about how such images are generated. Artists, photographers, and publishing professionals use computer graphics routinely and may be interested in a solid background in this field. Those readers can benefit from two features of this book, namely the detailed index and the thorough and precise exposition of the principles, methods, and techniques used in computer graphics. This textbook/reference tries to cover all the important topics of the graphics field, some in more detail than others. The index is exceptionally detailed and constitutes 2.5% of the book.

- **Individuals/handy resource**. All of us, not just certain professionals, are constantly exposed to digital images, digital effects, and computer animation. Intelligent persons who try to widen their horizons may wonder how digital images are created, edited, and distributed. It is my belief that this manual can serve the needs of this group of individuals as well. The book may prove useful to them because of the simple, straightforward descriptions of graphics devices and processes. The book may also be a handy resource because of the detailed index, which makes it easy to locate any topic.

Supplementary Resources

Because of the importance of computer graphics, a vast number of supplementary resources is available. Section 1.5 has a long list of seven classes of resources and this only scratches the surface of what is available. The Internet has many thousands of resources in the form of websites dedicated to graphics, class notes, scientific publications, and software.

Please bear in mind that graphics, like any discipline associated with computers and computations, develops continually, so readers should search the Internet for new, useful, and exciting sources of information. Search items may be as general as “computer graphics,” “computer animation,” “image processing,” “computer vision,” and “computer-aided design (CAD),” or as specific as “how to twirl in Photoshop,” “what is quaternion rotation,” and “Java code for dithering.”

> Shaw’s plays are the price we pay for Shaw’s prefaces.
>
> —James Agate.

Acknowledgements

A book of this magnitude is generally written with the help, dedicated work, and encouragement of many people, and this large textbook/reference is no exception. First and foremost should be mentioned my editor, Wayne Wheeler and the copyeditor, Francesca White. They made many useful comments and suggestions, and pointed out many mistypes, errors, and stylistic blemishes. In addition, I would like to thank H. L. Scott for permission to use Figure 2.82, CH Products for permission to use Figure 26.24b, Andreas Petersik for Figure 6.61, Shinji Araya for Figure 7.27, Dick Termes for many figures and paintings, the authors of *Hardy Calculus* for the limerick at the end of Chapter 13, Bill Wilburn for many *Mathematica* notebooks, and Ari Salomon for photos and panoramas in several plates.

I welcome any comments, suggestions and corrections. They should be emailed to dsalomon@csun.edu. An errata list and other information will be maintained in the book's website http://www.davidsalomon.name/CGadvertis.

And now, to the matter at hand.

> This preface made me so impatient, being conscious of my own merits and innocence, that I was going to interrupt; when he entreated me to be silent, and thus proceeded.
>
> — Jonathan Swift, *Gulliver's Travels*

Contents

To me style is just the outside of content, and content the inside of style, like the outside and the inside of the human body both go together, they can't be separated.

— Jean-Luc Godard

Introduction

Computer graphics is a vast field with applications to presentations (slides and video), computer art, cartography, medicine, entertainment, training, visualization (of large quantities of data), image processing, design (computer aided or CAD), and many other areas. The adage "a picture is worth a thousand words" explains why this field is so important, and it also explains why this book is so big. A video (or even a single image) tells us so much more than text, but it also requires more resources and longer preparation. There simply is much to learn about computer graphics, the special input/output devices it requires, the specific approaches, techniques, and algorithms it employs, and the specialized language, concepts, and terms that are commonly used in this field. This introduction covers the chief terms used in computer graphics, the concept of graphics processing unit (GPU), and the fundamental equation of computer graphics.

Terms and Concepts

Here are a few informal definitions of the graphics field and its "relatives" (as is common with any informal definitions, certain experts and users may disagree).

- *Image processing* (more precisely, digital image processing) is the field that deals with methods, techniques, and algorithms for image manipulation, enhancement, and interpolation. Researchers in this field test, publish, and implement their algorithms to make them available to users who often may not be technically savvy.

  (A technically-savvy person is one who is proficient enough to read a user manual, run software and equipment, and notice when things go wrong, but who is not an expert, does not develop algorithms, and does not write code.)

- *Image editing* (or simply imaging) is concerned with using software (implemented by image processing professionals) to manipulate images.

- *Computer graphics* is the discipline concerned with *generating* images. An image is generated (or synthesized) from geometrical descriptions. In addition to just generating images, this field is also concerned with rendering them accurately (so they look real) and fast (so that generating a long video of computer animation does not take forever).

- *Computer vision* is that branch of engineering concerned with constructing devices that comprehend and correctly interpret real objects the way our brains do.

- *Pattern recognition* is the mathematical field that deals with finding patterns in data and signals. The data can be text, audio, still images, or video. A special case of pattern recognition is data mining, a branch of computer science concerned with finding trends in large data bases.

- *Image analysis* includes anything that has to do with the extraction of meaningful information from images. Image analysis is a wide field that often overlaps other image-related areas. Reading a bar code, for example, can be considered image analysis, but is also pattern recognition. Identifying a face in a digital image is image analysis, but may use techniques from artificial intelligence. Detecting an edge in a digital image is image analysis, but it uses algorithms from image processing.

Here is a short list of the most important concepts and terms used in computer graphics:

- *Graphics* (from the Greek *γραφικός* *graphia* or *graphikos*) are visual presentations on some surface, such as a canvas, screen, or paper. Graphics are used to inform, illustrate, or entertain. Examples of graphics presentations are paintings, photographs, drawings, digital images, graphs, diagrams, geometric designs, and maps. Graphics may be in black-and-white, grayscale, or color and may contain text.

- *Computer graphics* is the field concerned with the generation, manipulation, and storage of digital graphical data. This includes still images (two- and three-dimensional), animated graphics, and interactive images (virtual reality). In fact, most digital data that is not text, software, or audio, is graphics data.

- A *pixel* (from picture element) is the smallest unit of a digital image. In the computer, a pixel is represented by its color code, which is either a grayscale value or the three components of a color. We tend to think of a pixel as a small dot, circular or square, but Section 2.1 shows that in principle, a pixel is a mathematical, dimensionless point. Figure Intro.1 shows a small, 128×128-pixel image and some of its constituent pixels.

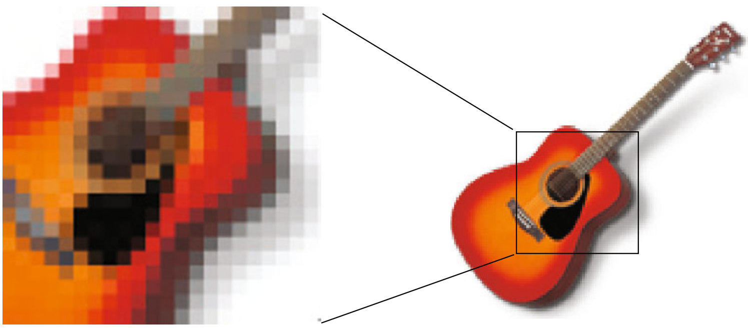


Figure Intro.1: An Image and Pixels.

- *Digital image*. Such an image is a rectangular array of pixels. Early images had only black-and-white or grayscale pixels, and featured low resolutions (only a few dozen pixels per centimeter or per inch). Current images are mostly color, where the colors are selected from huge palettes of millions of colors. Today's images can have resolutions of hundreds of pixels per centimeter, more often measured as dpi (dots per inch).

- *3D (three-dimensional) images*. Such an image contains much more information than a two-dimensional image. In order to fully view a three-dimensional image, it has to be rotated or transformed in some way. True viewing devices for such images are rare and expensive (holograms come to mind), so most three-dimensional images are projected on a two-dimensional display and have to be viewed from different directions.

- *Computer animation*. Much as a film is a sequence of still images, a digital animation is a sequence of digital images. An animation is organized in scenes, where consecutive images in a scene differ slightly in order to achieve the illusion of smooth movement.

- *Vector graphics*. Older graphics displays were of this type. In a vector graphics computer system, the program running in the computer creates a picture out of graphics components such as points, lines, and circles. A set of such components is stored in memory and becomes the description of an image. Special hardware translates each graphics component in memory into a visible part on the output (normally a display). The total of all the parts on the display constitutes the image.

- *Raster graphics*. Most current graphics displays are of this type. The screen displays an image that consists of small dots (pixels). The program running in the computer creates an image by assigning color codes to the pixels in memory. The graphics hardware scans the color codes in memory and actually paints the pixels on the display.

- *Scan conversion*. This is the process of converting a smooth geometric figure into pixels, such that the result looks as smooth as possible. Scan conversion is a very common operation, which is why scan conversion algorithms should be fast, using only logical operations, shifts, and simple arithmetic operations.

- *Transformations*. Once an image has been generated, the user often transforms it in order to view different parts of the image or to modify its shape in regular ways. Transformations are a must when a three-dimensional image is viewed on a two-dimensional display.

- *Projections*. A three-dimensional image has to be projected in order to print it or view it on a two-dimensional display. The most common projection is perspective, but parallel projections and nonlinear projections are suitable for special applications.

- *Rendering* is the process of generating a digital image from a mathematical model by means of algorithms implemented in computer programs. The mathematical model describes the objects that constitute the image in terms of points and curves. Inputs to the rendering algorithm include the orientation of the objects in space, the surface textures of the objects (including shading information), and the lighting configuration (the positions, intensities, and colors of the light sources). The term *rendering* has its origins in a painter's rendering of a scene.

- *Modeling*. A mathematical model of an object is a collection of points and curves. If the object is three-dimensional, the model also contains surface information. The first step in rendering an object is to display its mathematical model as a wireframe (Section 8.11.2).

- *Color*. An understanding of color is essential for dealing with color images and animation. Chapter 21 discusses the physical meaning and psychological implications of color, as well as the various color spaces and human vision.

- *Image compression*. Images tend to be large, compared to texts, which is why image compression is such an important field of research. Images can be compressed because their original (raw) representations are inefficient and feature much redundancy. The various image compression methods remove this redundancy in different ways, and Chapters 23 through 25 describe several approaches to image compression, especially orthogonal and subband (i.e., wavelet) transforms.

The Graphics Processing Unit (GPU)

Graphics is a computationally intensive application. The task of computing and displaying a color, high-resolution image on a large screen involves the following steps: (1) The surfaces that constitute the image have to be computed (as discussed in Part III of the book), either as polygonal surfaces or as smooth, curved surfaces that are then converted to triangles. (2) Each small triangle has to be rendered by simulating the light reflected from it (Chapter 17). The software has to know the positions, orientations, and colors of the light sources, and has to perform intensive computations for each triangle. (3) A texture is sometimes embedded in a surface. The texture is a small bitmap that is wrapped around the surface to enhance its look. (4) Once rendered, a triangle has to be projected to two dimensions (see Part II of the book). The projection can be parallel or perspective. (5) When the projection of a triangle has been determined (i.e., the two-dimensional coordinates of its three vertices are known), its visibility must be determined (Chapter 18). A triangle (a small part of a larger surface patch) may be completely or partially obscured by other surface parts that are closer to the camera (or the observer). Only those parts of the triangle that are visible to the camera should participate in the next step. (6) Those parts need to be scan converted (Chapter 3). A special algorithm is needed to determine the best pixels for each visible part of the triangle.

It has therefore been recognized in the early days of computer graphics that there is a need for special, dedicated hardware to perform most of these operations, which is why several companies started developing such hardware as early as the 1970s. The first units were special-purpose processors for creating bitmaps of text and simple graphics shapes and sending them to the video output. The 1980s saw graphics controllers for the IBM PC and the Commodore Amiga. Those devices tried to accelerate graphics operations and implement various two-dimensional graphics primitives in hardware. In the case of the Amiga computer, the graphics unit also handled scan conversion of lines, filling of regions, and performed block image transfers.

As hardware manufacturing capabilities improved during the 1990s, more graphics cards, graphics accelerators, and support devices for three-dimensional graphics operations were introduced for both personal computers and console games. In some cases, three-dimensional graphics operations became possible only if an accelerator was installed in the computer. The graphics programming language OpenGL appeared in the early 1990s and became so popular that users started demanding graphics processors that can be programmed in OpenGL.

It was only in the late 1990s that the first GPUs, made by Nvidia (which also coined the term GPU) were introduced. These devices (in combination with high-resolution display monitors and drawing and illustration software) have turned our personal computers into fast, high-resolution graphics generators, and have made it possible for anyone to create high-quality original graphics such as maps, plans, drawings, animations, games, and illustrations on a personal computer.

A GPU is a processor mounted on a special graphics card which also includes support devices, memory, and a bus. The graphics card can be plugged into the computer and later replaced with a more powerful version, but in most current personal computers the card is simply part of the main circuit board. The main functions of the GPU and its support devices are fast floating-point operations and vector operations. These operations are used all the time in surface computations, rendering, and visible surface determination.

Today (2011), GPUs feature parallel computational units that can perform their operations simultaneously. A parallel processor is notoriously difficult to program, which is why software writing for parallel GPUs is now considered an important field which is dubbed “general-purpose computing on GPU” or GPGPU.

> All problems in computer graphics can be solved with a matrix inversion.
>
> —James F. Blinn, *What’s the Deal with the DCT* (1993).

The Fundamental Equation of Computer Graphics

Mathematicians like fundamental theorems. The fundamental theorem of a field of mathematics is the theorem considered central to that field. There are fundamental theorems of calculus, algebra, arithmetic, vector analysis, Galois theory, Riemannian geometry, and more.

Theorems and proofs are rare in computer graphics, which is not as rigorous as mathematics. However, there is one equation that appears many times in this book and can be considered fundamental in computer graphics. This fundamental equation is the expression for the basic linear interpolation, Equation (9.1)

$$\mathbf{P}(t) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1,$$

where $\mathbf{P}_0$ and $\mathbf{P}_1$ can be numbers, points, curves, and even images (as, for example, in Sections 2.31.1 and 8.5). It is easy to see that $\mathbf{P}(0) = \mathbf{P}_0$ and $\mathbf{P}(1) = \mathbf{P}_1$. In general,

$\mathbf{P}(t)$ is a mixture (or a combination) of $(1-t)\%$ of $\mathbf{P}_0$ and $t\%$ of $\mathbf{P}_1$. Notice that these percentages add up to 1. When the parameter t is varied from 0 to 1, $\mathbf{P}(t)$ varies linearly from $\mathbf{P}_0$ to $\mathbf{P}_1$. (See the epigram at the end of Chapter 13.)

> You know what? I can't stand introductions. Weird coming from an author, right? It's like some committee got together and said that you've got to have an introduction in your book. Oh, and please make it long. Really long! In fact, make it so long that it will ensure no one reads introductions.
>
> —Matt Kloskowski, *The Complete Guide to Photoshop Layers,* 2011.