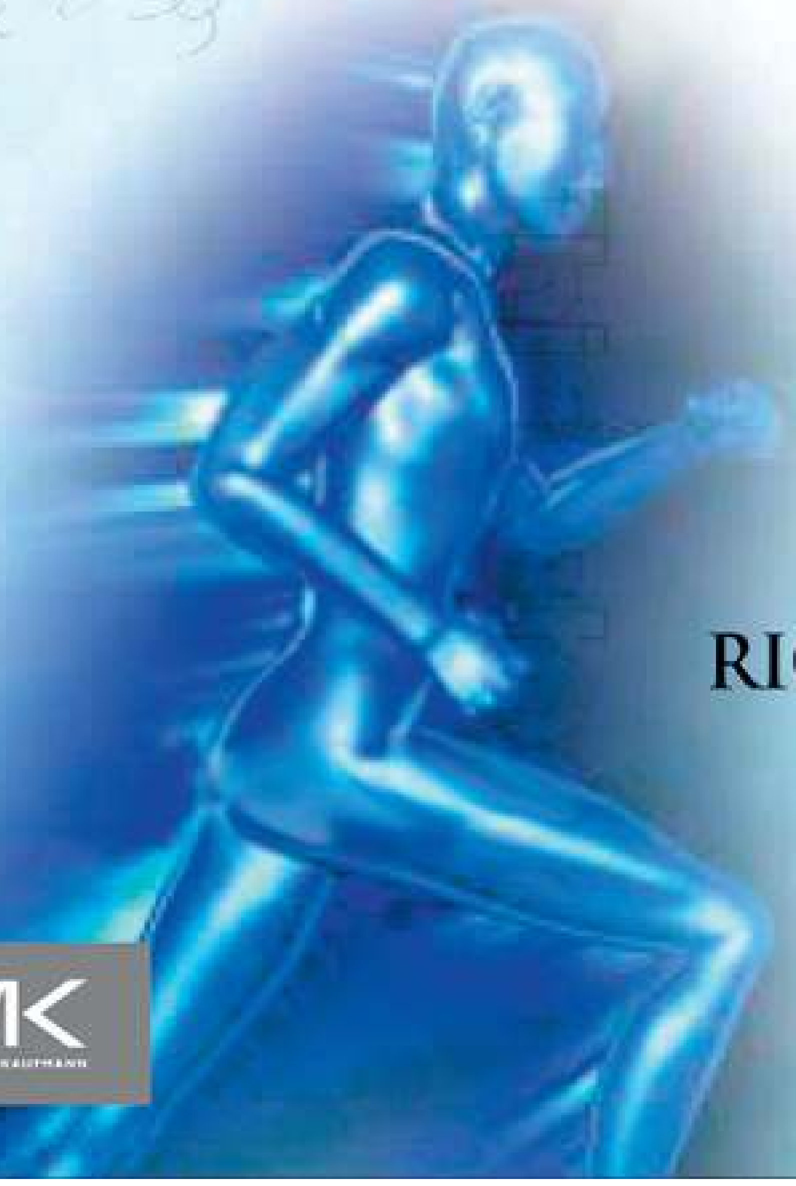
COMPUTER ANIMATION

ALGORITHMS & TECHNIQUES

THIRD EDITION



RICK PARENT

MK MORGAN KAUFMANN

Computer Animation

Intentionally left as blank

Computer Animation

Algorithms and Techniques

Third Edition

Rick Parent

Ohio State University

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier

Acquiring Editor: Steven Elliot
Development Editor: Robyn Day
Project Manager: Paul Gottehrer
Designer: Joanne Blank

Morgan Kaufmann is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

Library of Congress Cataloging-in-Publication Data
Application submitted

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

ISBN: 978-0-12-415842-9

Printed in the United States of America
12 13 14 16 16 10 9 8 7 6 5 4 3 2 1

To Kim and John, for teaching me to keep things in perspective.
And to my wife, Arlene, for her attention-to-detail approach to life, especially when juxtaposed to my 'big picture' way of doing things.

Intentionally left as blank

Contents

Preface xiii
About the Author xvii

CHAPTER 1 Introduction 1
1.1 Motion perception 2
1.2 The heritage of animation 4
1.2.1 Early devices 4
1.2.2 The early days of "conventional" animation 6
1.2.3 Disney 7
1.2.4 Contributions of others 8
1.2.5 Other media for animation 8
1.3 Animation production 9
1.3.1 Principles of animation 10
1.3.2 Principles of filmmaking 12
1.3.3 Sound 14
1.4 Computer animation production 15
1.4.1 Computer animation production tasks 16
1.4.2 Digital editing 18
1.4.3 Digital video 20
1.4.4 Digital audio 21
1.5 A brief history of computer animation 22
1.5.1 Early activity (pre-1980) 22
1.5.2 The middle years (the 1980s) 25
1.5.3 Animation comes of age (the mid-1980s and beyond) 26
1.6 Summary 29

CHAPTER 2 Technical Background 33
2.1 Spaces and transformations 33
2.1.1 The display pipeline 34
2.1.2 Homogeneous coordinates and the transformation matrix 38
2.1.3 Concatenating transformations: multiplying transformation matrices 40
2.1.4 Basic transformations 40
2.1.5 Representing an arbitrary orientation 42
2.1.6 Extracting transformations from a matrix 46
2.1.7 Description of transformations in the display pipeline 47
2.1.8 Error considerations 48

2.2 Orientation representation ... 52
2.2.1 Fixed-angle representation ... 54
2.2.2 Euler angle representation ... 56
2.2.3 Angle and axis representation ... 57
2.2.4 Quaternion representation ... 58
2.2.5 Exponential map representation ... 60
2.3 Summary ... 60

CHAPTER 3 Interpolating Values ... 61
3.1 Interpolation ... 61
3.1.1 The appropriate function ... 62
3.1.2 Summary ... 65
3.2 Controlling the motion of a point along a curve ... 65
3.2.1 Computing arc length ... 66
3.2.2 Speed control ... 78
3.2.3 Ease-in/ease-out ... 80
3.2.4 General distance-time functions ... 86
3.2.5 Curve fitting to position-time pairs ... 90
3.3 Interpolation of orientations ... 91
3.3.1 Interpolating quaternions ... 91
3.4 Working with paths ... 96
3.4.1 Path following ... 96
3.4.2 Orientation along a path ... 96
3.4.3 Smoothing a path ... 100
3.4.4 Determining a path along a surface ... 106
3.4.5 Path finding ... 108
3.5 Chapter summary ... 108

CHAPTER 4 Interpolation-Based Animation ... 111
4.1 Key-frame systems ... 111
4.2 Animation languages ... 115
4.2.1 Artist-oriented animation languages ... 116
4.2.2 Full-featured programming languages for animation ... 116
4.2.3 Articulation variables ... 117
4.2.4 Graphical languages ... 117
4.2.5 Actor-based animation languages ... 118
4.3 Deforming objects ... 119
4.3.1 Picking and pulling ... 119
4.3.2 Deforming an embedding space ... 121

4.4 Three-dimensional shape interpolation 135
4.4.1 Matching topology 136
4.4.2 Star-shaped polyhedra 137
4.4.3 Axial slices 137
4.4.4 Map to sphere 139
4.4.5 Recursive subdivision 145
4.5 Morphing (two-dimensional) 147
4.5.1 Coordinate grid approach 147
4.5.2 Feature-based morphing 153
4.6 Chapter summary 159

CHAPTER 5 Kinematic Linkages 161

5.1 Hierarchical modeling 162
5.1.1 Data structure for hierarchical modeling 164
5.1.2 Local coordinate frames 170
5.2 Forward kinematics 171
5.3 Inverse kinematics 172
5.3.1 Solving a simple system by analysis 173
5.3.2 The Jacobian 174
5.3.3 Numeric solutions to IK 178
5.3.4 Summary 185
5.4 Chapter summary 185

CHAPTER 6 Motion Capture 187

6.1 Motion capture technologies 187
6.2 Processing the images 188
6.3 Camera calibration 190
6.4 Three-dimensional position reconstruction 191
6.4.1 Multiple markers 192
6.4.2 Multiple cameras 192
6.5 Fitting to the skeleton 193
6.6 Output from motion capture systems 195
6.7 Manipulating motion capture data 196
6.7.1 Processing the signals 196
6.7.2 Retargeting the motion 197
6.7.3 Combining motions 197
6.8 Chapter summary 198

CHAPTER 7 Physically Based Animation 199

7.1 Basic physics—a review 200
7.1.1 Spring-damper pair 202

7.2 Spring animation examples 202
7.2.1 Flexible objects 202
7.2.2 Virtual springs 205
7.3 Particle systems 205
7.3.1 Particle generation 206
7.3.2 Particle attributes 207
7.3.3 Particle termination 207
7.3.4 Particle animation 207
7.3.5 Particle rendering 207
7.3.6 Particle system representation 208
7.3.7 Forces on particles 208
7.3.8 Particle life span 209
7.4 Rigid body simulation 209
7.4.1 Bodies in free fall 210
7.4.2 Bodies in collision 219
7.4.3 Dynamics of linked hierarchies 232
7.5 Cloth 235
7.5.1 Direct modeling of folds 237
7.5.2 Physically based modeling 240
7.6 Enforcing soft and hard constraints 244
7.6.1 Energy minimization 244
7.6.2 Space-time constraints 247
7.7 Chapter summary 249

CHAPTER 8 Fluids: Liquids and Gases 251

8.1 Specific fluid models 251
8.1.1 Models of water 251
8.1.2 Modeling and animating clouds 262
8.1.3 Modeling and animating fire 268
8.1.4 Summary 270
8.2 Computational fluid dynamics 270
8.2.1 General approaches to modeling fluids 271
8.2.2 CFD equations 272
8.2.3 Grid-based approach 276
8.2.4 Particle-based approaches including smoothed particle hydrodynamics 277
8.3 Chapter summary 280

CHAPTER 9 Modeling and Animating Human Figures 283

9.1 Overview of virtual human representation 283
9.1.1 Representing body geometry 284
9.1.2 Geometry data acquisition 285

9.1.3 Geometry deformation ... 286
9.1.4 Surface detail ... 286
9.1.5 Layered approach to human figure modeling ... 287
9.2 Reaching and grasping ... 290
9.2.1 Modeling the arm ... 290
9.2.2 The shoulder joint ... 293
9.2.3 The hand ... 293
9.2.4 Coordinated movement ... 295
9.2.5 Reaching around obstacles ... 296
9.2.6 Strength ... 297
9.3 Walking ... 298
9.3.1 The mechanics of locomotion ... 298
9.3.2 The kinematics of the walk ... 303
9.3.3 Using dynamics to help produce realistic motion ... 303
9.3.4 Forward dynamic control ... 308
9.3.5 Summary ... 308
9.4 Coverings ... 309
9.4.1 Clothing ... 309
9.4.4 Hair ... 309
9.5 Chapter summary ... 311

CHAPTER 10 Facial Animation ... 317

10.1 The human face ... 317
10.1.1 Anatomic structure ... 317
10.1.2 The facial action coding system ... 319
10.2 Facial models ... 320
10.2.1 Creating a continuous surface model ... 322
10.2.2 Textures ... 325
10.3 Animating the face ... 327
10.3.1 Parameterized models ... 327
10.3.2 Blend shapes ... 327
10.3.3 Muscle models ... 329
10.3.4 Expressions ... 332
10.3.5 Summary ... 332
10.4 Lip-sync animation ... 333
10.4.1 Articulators of speech ... 333
10.4.2 Phonemes ... 334
10.4.3 Coarticulation ... 335
10.4.4 Prosody ... 335
10.5 Chapter summary ... 335

CHAPTER 11 Behavioral Animation **339**
11.1 Primitive behaviors 342
11.1.1 Flocking behavior 342
11.1.2 Prey–predator behavior 351
11.2 Knowledge of the environment 352
11.2.1 Vision 352
11.2.2 Memory 353
11.3 Modeling intelligent behavior 354
11.3.1 Autonomous behavior 354
11.3.2 Expressions and gestures 356
11.3.3 Modeling individuality: personality and emotions 357
11.4 Crowds 358
11.4.1 Crowd behaviors 359
11.4.2 Internal structure 359
11.4.3 Crowd control 360
11.4.4 Managing *n*-squared complexity 360
11.4.5 Appearance 361
11.5 Chapter summary 361

CHAPTER 12 Special Models for Animation **365**
12.1 Implicit surfaces 365
12.1.1 Basic implicit surface formulation 365
12.1.2 Animation using implicitly defined objects 367
12.1.3 Collision detection 368
12.1.4 Deforming the implicit surface as a result of collision 368
12.1.5 Level set methods 371
12.1.6 Summary 372
12.2 Plants 372
12.2.1 A little bit of botany 372
12.2.2 L-systems 374
12.2.3 Animating plant growth 379
12.2.4 Summary 381
12.3 Subdivision surfaces 382
12.4 Chapter summary 384

Appendix A Rendering Issues 387
Appendix B Background Information and Techniques 407
Index 503

Preface

Overview

This book surveys computer algorithms and programming techniques for specifying and generating motion for graphical objects, that is, *computer animation*. It is primarily concerned with three-dimensional (3D) computer animation. The main audience is advanced undergraduate or beginning graduate students in Computer Science. Computer graphics programmers who want to learn the basics of computer animation programming and artists who use software packages to generate computer animation (*digital animators*) who want to better understand the underlying computational issues of animation software will also benefit from this book.

It should come as no surprise to anyone reading this book that activity in Computer Animation has exploded in recent years - as a research area, as an academic field of study, as a career, and even as a hobby. Feature length films are now often stored digitally and incorporate digital special effects (often referred to as *computer generated imagery* and abbreviated *CGI*). As listed by the Internet Movie Database (imdb.com) [1] as of March 2012, all of the top 10 U.S. films (All-Time U.S. Box Office) depend on extensive use of CGI. Computer animated films have become top box office attractions - according to the same movie database, 2 of the top 10 feature length films are computer animations (*Shrek 2* and *Toy Story 3*) with a third having a significant computer animation component (*Avatar*). Recent *Technical Achievement* and *Scientific and Engineering* awards from the Motion Picture Academy of Arts and Sciences have been for digital image technology including render queue management, facial motion retargeting, tools to review digital effects, and efficient rendering of volumetric effects, just to name a few [2]. And, of course, the computer game industry has exploded. The Entertainment Software Association estimate that, in 2010, consumers spent $25.1 billion on video games, hardware and accessories [3].

Computer animation is more accessible that ever. Desktop, high-quality, computer animation is now possible because of sophisticated off-the-shelf animation software, cheap CPU cycles, and cheap storage coupled with digital video recording. Many technical programs and computer science departments now offer courses in computer animation and the proliferating artistic programs train digital artists in the use of off-the-shelf animation software. There are now major technical conferences and journals that archive developments in computer animation and video game algorithms and techniques.

This book addresses practical issues, provides accessible techniques, and offers straightforward implementations. Purely theoretical discussions have been avoided except to point out avenues of current and future research. In some cases, programming examples are complete working code segments—in C, which can be copied, compiled, and run to produce basic examples of the algorithms discussed; other programming examples are C-like pseudocode that can be translated into working code. C was chosen because it forms the common basis for languages such as C++ and Java, and it lends itself to illustrating the step-by-step nature of algorithms. The Appendixes cover basic material that the reader may find useful as a refresher as well as specific algorithms for use in implementations.

This text is not intended for animators using off-the-shelf animation software (except to the extent that it might help in understanding the underlying computations required for a particular technique). It does not attempt to cover the theory of computer animation, address the aesthetics of computer animation, or discuss the artistic issues involved in designing animations. It does not detail the production issues in the actual commercial enterprise of producing a finished piece of animation. And, finally, it does not address the issue of *computer-assisted animation,* which, for our purposes, is taken to mean the computerization of conventional hand-drawn techniques; for the most part, that area has its own set of separate issues [4] [5]. The book does concentrate on full 3D computer animation and identifies the useful algorithms and techniques that animators and programmers can use to move objects in interesting ways. While 3D techniques are the emphasis, 2D is not completely ignored.

The fundamental objective of computer animation programming is to select techniques and design tools that are expressive enough for animators to specify what they intend, yet at the same time are powerful enough to relieve animators from specifying any details they are not interested in. Obviously, no one tool is going to be right for every animator, for every animation, or even for every scene in a single animation. The appropriateness of a particular animation tool depends on the effect desired and the control required by the animator. An artistic piece of animation will usually require tools different from those required by an animation that simulates reality or educates a patient. In this spirit, alternative approaches are presented whenever possible.

Organization of the Book

This book presents background information in the first couple of chapters. Techniques that directly specify motion (*kinematic* - not based on underlying forces) are presented in the next 4 chapters followed by 2 chapters that cover force-based (*dynamics*) animation. Character animation is then covered in 3 chapters. The last chapter covers special geometric models. Appendices provide extensive support material. More detail about the chapters is given below.

Chapter 1 discusses general issues related to animation, including motion perception, the heritage of conventional animation paying particular attention to its technological innovations, overviews of animation production and computer animation production, and a snapshot of the ever-evolving history of computer animation. These provide a broad perspective of the art and craft that is animation.

Chapter 2 presents background material and reviews the basics of computer graphics necessary for animation. It reviews computational issues in computer graphics to ensure a solid background in the techniques that are important in understanding the remainder of the book. This includes a review of the rendering pipeline and a discussion of the ordering of transformations to reduce round-off errors that can creep into a series of calculations as one builds on another. A detailed section on quaternion representation of orientation is presented in this chapter as well. If the reader is well versed in computer graphics, this chapter may be skimmed to pick up relevant terminology or skipped altogether.

Chapters 3 and 4 cover interpolation. Chapter 3 presents the fundamentals. It introduces time-space curves, arc-length parameterization of a curve, and speed control along a curve. Interpolation of orientation with an emphasis on using quaternions is then covered. Various ways to work with paths are then presented. Chapter 4 presents animation techniques based on interpolation including key frame interpolation, animation languages, shape deformation, and shape interpolation including morphing.

Chapters 5 and 6 are primarily concerned with kinematic control of articulated figures. Chapter 5 is concerned with kinematics of linked appendages. It covers both forward and inverse kinematics. Chapter 6 covers the basics of motion capture (*mocap*). First, the basic technology is reviewed. Then the chapter discusses how the images are processed to reconstruct articulated figure kinematics, including some techniques to modify the resultant mocap data.

Chapters 7 and 8 cover animation that is more concerned with simulating real-world (e.g. physics-based) processes. Chapter 7 covers physics-based animation as well as mass-spring-damper systems, particle systems, rigid body dynamics, and enforcing constraints. It has an additional section on ways to model cloth. Chapter 8 covers the modeling and animation of fluids. It first covers models that handle specific macro-features of fluids and then covers computational fluid dynamics (CFD) as it relates to computer animation.

Chapters 9 through 11cover animation concerned with people and other critters. Chapter 9 covers human figure animation: modeling, reaching, walking, clothing, and hair. Chapter 10 covers facial animation: facial modeling, expressions, and lip-sync animation. Chapter 11 covers behavioral animation including flocking, predator–prey models, intelligent behavior and crowd behavior.

Finally, Chapter 12 covers a few special models that are useful to animation: implicit surfaces, L-systems, and subdivision surfaces.

Appendix A presents rendering issues often involved in producing images for computer animation: double buffering, compositing, computing motion blur, drop shadows, and billboarding. It assumes a general knowledge of the use of frame buffers, how a *z*-buffer display algorithm works, and aliasing.

Appendix B is a collection of relevant material from a variety of disciplines. It contains a survey of interpolation and approximation techniques, vector algebra and matrices, quaternion conversion code, the first principles of physics, several useful numeric techniques, optimization, and attributes of film, video, and image formats, and a few other topics.

The Web page associated with the book, containing images, code, and figures can be found at *textbooks.elsevier.com/9780125320009*.

Acknowledgments

Many people contributed in various ways to this book. First and foremost, I'd like to thank my wife, Arlene, who both tolerated my preoccupation with this project and greatly increased the readability of the book.

In general, I owe much to the students I have had the pleasure of knowing and/or working with over the years and whose collective interest in, knowledge of, and enthusiasm for the field has fueled my own. Their contribution cannot be overstated. These include Doug Roble, John Chadwick, Dave Haumann, Dave Ebert, Matt Lewis, Karan Singh, Steve May, James Hahn, Ferdi Scheepers, Dave Miller, Beth Hofer, Madhavi Muppala, Domin Lee, Kevin Rogers, Brent Watkins, Brad Winemiller, Meg Geroch, Lawson Wade, Arun Somasundaram, Scott King, and Scott (Slim) Whitman (apologies to anyone left out).

I would also like to thank the readers who, over the years, have given me feedback (both good and bad) concerning the book. In particular, I would like to note the critical contributions of Dr. Philip Schlup of Colorado State University, Dr. Brian Wyvill of the University of Calgary, and Dr. Przemyslaw Kiciak of the University of Warsaw.

And finally, I would like to acknowledge the support of the Department of Computer Science and Engineering at Ohio State (Xiaodong Zhang, Chair) as well as the support of the Advanced Computing Center for Art and Design (Maria Palazzi, Director). I would also like to thank Morgan Kaufmann's reviewers and multiple editors for seeing this project through.

References

[1] All Time Grossing Movies. In: The Internet Movie Database (IMDB). IMDb.com, Inc; 2012. Web. 26 March 2012. http://www.imdb.com/boxoffice/alltimegross.

[2] Scientific and Technical Awards. In: Academy of Motion Picture Arts and Sciences. 2012. Web. 26 March 2012. http://www.oscars.org/awards/scitech/index.html.

[3] Industry facts. In: The Entertainment Software Association (ESA). 2012. Web. 26 March 2012. http://www.theesa.com/facts/index.asp.

[4] Catmull E. The Problems of Computer-Assisted Animation. In: Computer Graphics. Proceedings of SIGGRAPH 78, vol. 12(3). August Atlanta, Ga.; 1978. p. 348–53.

[5] Levoy M. A Color Animation System Based on the Multiplane Technique. In: George J, editor. Computer Graphics. Proceedings of SIGGRAPH 77, vol 11(2). San Jose, Calif.; July 1977. p. 65–71.

About the Author

Rick Parent is a Professor in the Computer Science and Engineering Department of Ohio State University (OSU). As a graduate student, Rick worked at the Computer Graphics Research Group (CGRG) at OSU under the direction of Charles Csuri. In 1977, he received his Ph.D. from the Computer and Information Science (CIS) Department, majoring in Artificial Intelligence. For the next three years, he worked at CGRG first as a Research Associate, and then as Associate Director. In 1980 he co-founded and was President of The Computer Animation Company. In 1985, he joined the faculty of the CIS Department (now the Department of Computer -Science and Engineering) at Ohio State. Rick's research interests include various aspects of computer animation with special focus on animation of the human figure. Currently, he is working on facial animation and on using model-based techniques to track human figures in video.

Intentionally left as blank

CHAPTER

Introduction

1

Computer animation, for many people, is synonymous with big-screen events such as *Star Wars*, *Toy Story*, and *Avatar*. But not all, or arguably even most, computer animation is done in Hollywood. It is not unusual for Saturday morning cartoons to be entirely computer generated. Computer games take advantage of state-of-the-art computer graphics techniques and have become a major motivating force driving research in computer animation. Real-time performance-driven computer animation has appeared at SIGGRAPH[1] and on *Sesame Street*. Desktop computer animation is now possible at a reasonable cost. Computer animation on the Web is routine. Digital simulators for training pilots, SWAT teams, and nuclear reactor operators are commonplace. The distinguishing characteristics of these various venues are the cost, the image quality desired, and the amount and type of interaction allowed. This book does not address the issues concerned with a particular venue, but it does present algorithms and techniques used to do computer animation in all of them.

Computer animation, as used here, refers to any computer-based computation used in producing images intended to create the perception of motion. The emphasis in this book is on algorithms and techniques that process three-dimensional graphical data. In general, any value that can be changed can be animated. An object's position and orientation are obvious candidates for animation, but all of the following can be animated as well: the object's shape, its shading parameters, its texture coordinates, the light source parameters, and the camera parameters.

This book is organized as follows. To lay a firm foundation for the rest of the book, Chapter 2 surveys the technical background of computer graphics relevant to computer animation. This includes the fundamental geometric transformations and associated representations of graphical data. It can be skipped by those well versed in the mathematics of the computer graphics display pipeline. Chapters 3–11 cover various computer animation algorithms and techniques: Chapters 3–5 deal with directly specifying motion (kinematics), Chapter 6 covers digitizing motion (motion capture), Chapters 7 and 8 consider physically based animation (dynamics), and Chapters 9–11 concentrate on (mostly human) figure animation. Finally, Chapter 12 surveys some modeling techniques that have been used in computer animation. The appendices provide ancillary material. Appendix A covers rendering issues that are relevant for animation, and Appendix B provides detail of the mathematics used in the text.

In considering computer animation techniques, there are basically three general approaches to motion control. The first is *artistic animation* in which the animator has the prime responsibility for crafting the motion. The foundation of artistic animation is interpolation. Various animation

[1]SIGGRAPH is the Association for Computing Machinery's (ACM) special interest group on computer graphics. The ACM is the main professional group for computer scientists.

techniques based on interpolation are concentrated in the early chapters (Chapters 3–5). The second is *data-driven animation* in which live motion is digitized and then mapped onto graphical objects. The primary technology for data-driven animation is referred to as *motion capture* and is the topic of Chapter 6. The third is *procedural animation*, in which there is a computational model that is used to control the motion. Usually, this is in the form of setting initial conditions for some type of physical or behavioral simulation. Procedural animation is concentrated in the later chapters (Chapters 7–11).

To set the context for computer animation, it is important to understand its heritage, its history, and certain relevant concepts. The rest of this chapter discusses motion perception, the technical evolution of animation, animation production, and notable works in computer animation. It provides a grounding in computer animation as a field of endeavor.

1.1 Motion perception

A picture can quickly convey a large amount of information because the human visual system is a sophisticated information processor. It follows, then, that moving images have the potential to convey even more information in a short time. Indeed, the human visual system has evolved to provide for survival in an ever-changing world; it is designed to notice and interpret movement.

It is widely recognized that a series of images, when displayed in rapid succession, are perceived by an observer as a single moving image. This is possible because the eye–brain complex has the ability, under sufficient viewing conditions and within certain playback rates, to create a sensation of continuous imagery from such a sequence of still images. A commonly held view is that this experience is due to *persistence of vision*, whereby the eye retains a visual imprint of an image for a brief instant once the stimulus is removed. It is argued that these imprints, called *positive afterimages* of the individual stills, fill in the gaps between the images to produce the perception of a continuously changing image. Peter Roget (of *Roget's Thesaurus* fame) presented the idea of impressions of light being retained on the retina in 1824 [35]. But persistence of vision is not the same as perception of motion. Rotating a white light source fast enough will create the impression of a stationary white ring. Although this effect can be attributed to persistence of vision, the result is static. The sequential illumination of a group of lights typical of a movie theater marquee produces the illusion of a lighted object circling the signage. Motion is perceived, yet persistence of vision does not appear to be involved because no individual images are present. Recently, the causality of the (physiological) persistence of vision mechanism has been called into question and the perception of motion has been attributed to a (psychological) mechanism known as the *phi phenomenon* (as is the case in the movie marquee example given above). A related phenomenon, for example the apparent motion of a disk traveling between two flickering disks, is referred to as *beta movement* [1] [2] [13] [39].

Whatever the underlying mechanism is, the result is that in both film and video, a sequence of images can be displayed at rates fast enough to fool the eye into interpreting it as continuous imagery. When the perception of continuous imagery fails to be created, the display is said to *flicker*. In this case, the animation appears as a rapid sequence of still images to the eye–brain complex. Depending on conditions such as room lighting and viewing distance, the rate at which individual images must be played back in order to maintain the perception of continuous imagery varies. This rate is referred to as the *critical flicker frequency* [8].

While perception of motion addresses the lower limits for establishing the perception of continuous imagery, there are also upper limits to what the eye can perceive. The receptors in the eye continually sample light in the environment. The limitations on motion perception are determined, in part, by the reaction time of those sensors and by other mechanical limitations such as blinking and tracking. If an object moves too quickly with respect to the viewer, then the receptors in the eye will not be able to respond fast enough for the brain to distinguish sharply defined, individual detail; *motion blur* results [11]. In a sequence of still images, motion blur is produced by a combination of the object's speed and the time interval over which the scene is sampled. In a still camera, a fast-moving object will not blur if the shutter speed is fast enough relative to the object's speed. In computer graphics, motion blur will never result if the scene is sampled at a precise instant in time; to compute motion blur, the scene needs to be sampled over an interval of time or manipulated to appear as though it were [21] [32]. (See Appendix A.3 for a discussion of motion blur calculations.) If motion blur is not calculated, then images of a fast-moving object can appear disjointed, similar to viewing live action under the effects of a strobe light. This effect is often referred to as *strobing*. In hand-drawn animation, fast-moving objects are typically stretched in the direction of travel so that the object's images in adjacent frames overlap [49], reducing the strobing effect.

As reflected in the previous discussion, there are actually two rates of concern. One is the *playback* or *refresh* rate—the number of images per second displayed in the viewing process. The other is the *sampling* or *update* rate—the number of different images that occur per second. The playback rate is the rate related to flicker. The sampling rate determines how jerky the motion appears. For example, a television signal conforming to the National Television Standards Committee (NTSC) format displays images at a rate of roughly 30 frames per second (fps),[2] but because it is *interlaced*,[3] *fields* are played at 60 frames per second to prevent flicker under normal viewing conditions [34]. In some programs (e.g., some Saturday morning cartoons) there may be only six different images per second, with each image repeatedly displayed five times. Often, lip-sync animation is drawn *on twos* (every other frame) because drawing it *on ones* (animating it in every frame) makes it appear too hectic. Film is typically shown in movie theatres at playback rates of 24 fps (in the United States) but, to reduce the flicker, each frame is actually displayed twice (*double-shuttered*) to obtain an effective refresh rate of 48 fps. On the other hand, because an NTSC television signal is interlaced, smoother motion can be produced by sampling the scene every 60th of a second even though the complete frames are only played back at 30 fps [8]. Computer displays are typically progressive scan (*noninterlaced*) devices with refresh rates above 70 fps [34]. See Appendix B.10 for some details concerning various film and video formats.

The display and perception of animation using a sequence of still images imposes certain requirements on how those images are computed and played effectively. Understanding the operation, limits, and trade-offs of the human visual system are essential when making intelligent decisions about designing any type of visual and auditory content, including computer animation.

---

[2]More accurately, the format for broadcast television system, established by the NTSC, specifies a frame rate of 29.97 fps [29].

[3]An *interlaced* display is one in which a frame is divided into two *fields*. Each field consists of odd or even numbered scanlines. The odd and even fields are displayed in alternate scans of the display device [8].

1.2 The heritage of animation

In the most general sense, *animate*[4] means "give life to" and includes live-action puppetry such as that found on *Sesame Street* and the use of electromechanical devices to move puppets, such as *animatronics*. History is replete with attempts to bring objects to life. This history is a combination of myth, deception, entertainment, science, and medicine. Many of the references to animation are in the form of stories about conjuring a life force into some humanoid form: from Pygmalion to Prometheus to Wagner's homunculus in Goethe's *Faust* to Shelley's Dr. Frankenstein. Some of the history is about trying to create mechanical devices that mimic certain human activity: from Jacque Vaucanson's mechanical flute player, drummer, and defecating duck in the 1730s to Wolfgang von Kempelen's chess player in 1769 to Pierre Jaquet-Droz's writing automaton of 1774 to the electromechanical humanoid robots (*animatronics*) popular today. The early mechanisms from the 1700s and 1800s were set in the milieu of scientific debate over the mechanistic nature of the human body (e.g., *L'Homme Machine*, translated as *Man a Machine*, was written by Julien Offray de La Mettrie in 1747 and was quite controversial). This activity in humanoid mechanical devices was propelled by a confluence of talents contributed by magicians, clock makers, philosophers, scientists, artists, anatomists, glove makers, and surgeons (see Gaby Wood's book for an entertaining survey on the quest for mechanical life [50]). Here, however, the focus is on devices that use a sequence of individual still images to create the effect of a single moving image, because these devices have a closer relationship to hand-drawn animation.

1.2.1 Early devices

Persistence of vision and the ability to interpret a series of stills as a moving image were actively investigated in the 1800s [5], well before the film camera was invented. The recognition and subsequent investigation of this effect led to a variety of devices intended as parlor toys [23] [38]. Perhaps the simplest of these early devices is the *thaumatrope*, a flat disk with images drawn on both sides with two strings connected opposite each other on the rim of the disk (see Figure 1.1). The disk could be quickly flipped back and forth by twirling the strings. When flipped rapidly enough, the two images appear to be superimposed. The classic example uses the image of a bird on one side and the image of a birdcage on the other; the rotating disk visually places the bird inside the birdcage. An equally primitive technique is the *flip book*, a tablet of paper with an individual drawing on each page. When the pages are flipped rapidly, the viewer has the impression of motion.

One of the most well known early animation devices is the *zoetrope*, or wheel of life. The zoetrope has a short fat cylinder that rotates on its axis of symmetry. Around the inside of the cylinder is a sequence of drawings, each one slightly different from the ones next to it. The cylinder has long vertical slits cut into its side between each adjacent pair of images so that when it is spun on its axis each slit allows the eye to see the image on the opposite wall of the cylinder (see Figure 1.2). The sequence of slits passing in front of the eye as the cylinder is spun on its axis presents a sequence of images to the eye, creating the illusion of motion.

[4] A more restricted definition of *animation*, also found in the literature, requires the use of a sequence of stills to create the visual impression of motion. The restricted definition does not admit techniques such as animatronics or shadow puppets under the rubric animation.

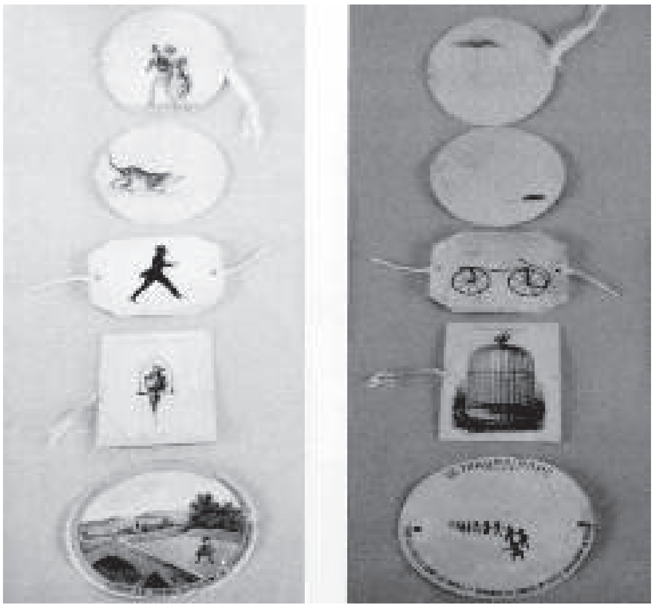


FIGURE 1.1

A thaumatrope.

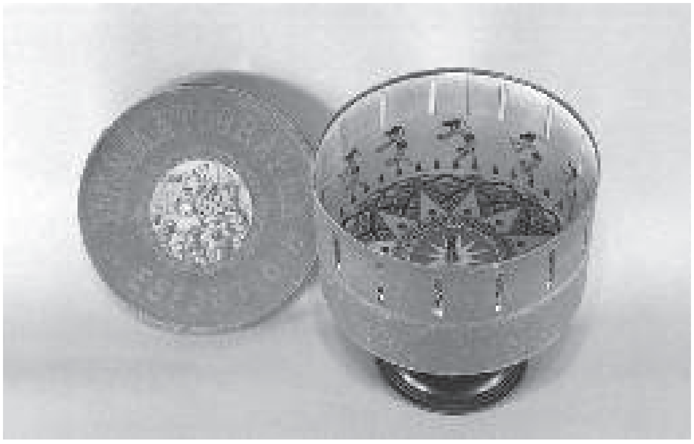


FIGURE 1.2

A zoetrope.

Related gizmos that use a rotating mechanism to present a sequence of stills to the viewer are the *phenakistoscope* and the *praxinoscope*. The phenakistoscope also uses a series of rotating slots to present a sequence of images to the viewer by positioning two disks rotating in unison on an axis; one disk has slits, and the other contains images facing the slits. One sights along the axis of rotation so the slits pass in front of the eye, which can thus view a sequence of images from the other disk.