

**EVERYTHING**

you need to create in 3-D

# 3-D Starter Kit

for Macintosh®

**All the Best  
Software and Advice**

Sean Wagstaff

- Build 3-D models
- Render and animate
- Adjust lighting, textures, and more
- Learn 3-D techniques
- Use working demos
- Get 3-D product info





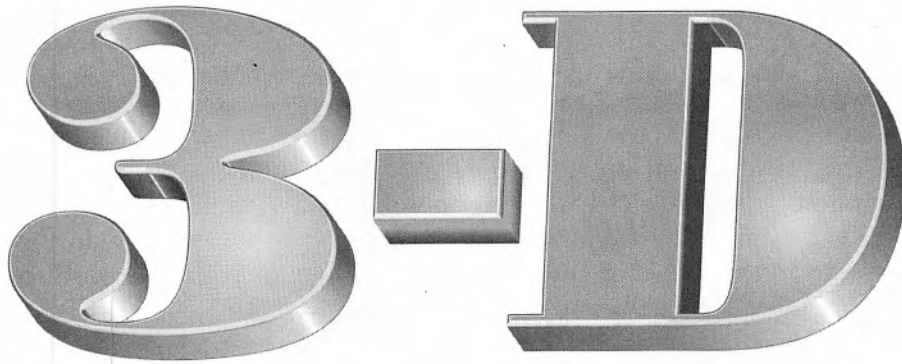
# 3-D

## **STARTER KIT FOR MACINTOSH**



Hayden  
Books





# **STARTER KIT FOR MACINTOSH**

**Second Edition of Macintosh 3-D Workshop**

Sean Wagstaff

**Hayden Books**



### **3-D Starter Kit for Macintosh**

**©1994 Hayden Books, a division of Macmillan Computer Publishing**

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission from the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than for your own personal use is a violation of United States copyright laws. For information, address Hayden Books, 201 W. 103rd Street, Indianapolis, Indiana 46290.

Library of Congress Catalog Card Number: 94-77381

ISBN: 1-56830-125-1

This book is sold *as is*, without warranty of any kind, either express or implied. While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information or instructions contained herein. It is further stated that the publisher and author are not responsible for any damage or loss to your data or your equipment that results directly or indirectly from your use of this book.

96 95 94      4 3 2

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number is the number of the book's printing. For example, a printing code of 94-1 shows that the first printing of the book occurred in 1994.

## **Trademark Acknowledgments**

All products mentioned in this book are either trademarks of the companies referenced in this book, registered trademarks of the companies referenced in this book, or neither. We strongly advise that you investigate a particular product's name thoroughly before you use the name as your own.

Apple, Mac, and Macintosh are registered trademarks of Apple Computer, Inc.

## Dedication

To Mom and Dad, for their sense of perspective.

## About the Author

### Sean Wagstaff

Since he left his three-year post as *MacWEEK*'s Associate Editor of Reviews, Sean Wagstaff has focused his vision on 3-D graphics, digital media, and writing. His reviews, tips, and analyses have appeared in *MacWEEK*, *MacUser*, *Digital Video*, and *Digital Media*. He spends his time giving shape to ideas—some of which have nothing to do with computers.

In addition to the digital press, Wagstaff's writing, photography, and editing credits include *Insight's Guides to California*, *Northern California*, *Southern California*; *Wind Surf Magazine*, *California Angler Magazine*, *Mexico Magazine*, and *Skiing Magazine*. He is currently writing the *Guide to West Coast Snowboarding* for Harper West, and illustrating an interactive children's storybook on CD-ROM.

Chronicles of his adventures in graphics, multimedia, and four dimensions—both synthetic and natural—will continue to appear in *MacWEEK*, *Digital Video*, *Digital Media*, and other prominent locations.

# **The Hayden Books Team**

## **Publisher**

David Rogelberg

## **Managing Editor**

Karen Whitehouse

## **Developmental Editor**

Brad Miser

## **Copy/Production Editor**

Meshell Dinn

## **Technical Reviewers**

Shawn Hopwood, Bob Zigon

## **Publishing Coordinator**

Stacy Kaplan

## **Interior Designer**

Barb Webster

## **Cover Designer**

Dan Armstrong

## **Production**

Dan Caparo, Brad Chinn, Kim Cofer, Lisa Daugherty, David Dean,  
Jennifer Eberhardt, Erika Millen, Beth Rago, Bobbi Satterfield, Karen Walsh,  
Robert Wolf

## **Indexer**

Greg Eldred



## Acknowledgments

Many people helped make this book possible, or at least tolerated the author's transgressions, absences, and absentminded blunders during its creation. I hope that the simple fact of a second edition serves to justify the imposition. To family, friends, and others who stubbed toes on my flotsam without a whimper, I re-extend an open invitation to dinner—I'm still cooking. At *MacWEEK*, Rick LePage exhibited extraordinary patience, and frequently turned a blind eye to my "rendering" as deadlines soared by. In that regard, Missy Roback, has also been quite patient. To my other colleagues at *MacWEEK*, I extend thanks for the years of education and discussion that led to this project. To the many people in the Mac industry who contributed to this effort, Gina Rubattino of Pixel Relations was particularly helpful. Matthew Borislausky, of View by View—renderer to the White House—also contributed invaluable insight. Peter Ryce, Sketch! product manager, was a noteworthy contributor of ideas and enthusiasm. Shawn Hopwood, instrumental in the successes of VIDI and Byte by Byte (and now evangelist for Apple's advanced 3-D technology) was the first "3D" person I ever talked to, and he made me a convert. His contribution as technical editor and advisor has made this a far more insightful and accurate book than it otherwise could have been. Ben Long has often bandied the possibility of stringing cables through yards and over fences, hedges, and telephone poles to network our Macs. He continues to match me in a game of "What Did We Learn Today?" which—more than anything—is the continuing source of ideas for this book. To the sysops and many contributors to America Online's 3DSIG, I would like to express my appreciation for the uninterrupted flow of fascinating tips and information. Neil McManus, long-time lunch mate and co-conspirator, as an editor, kindly told me where things had run completely amuck in the first edition, and his influence is still strongly felt. Brad Miser, insightful development editor at Hayden, quietly ignored my late-night rantings and kept about the business of making this a good book. Finally, Dorit Fehrensén constantly and kindly reminds me that in reality, not all things are virtual. For her, I hope the wait was worth it.

—stW

## To Our Readers

Dear Friend,

Thank you on behalf of everyone at Hayden Books for choosing *3-D Starter Kit for Macintosh* to enable you to learn about the exciting world of 3-D on the Macintosh. With the ever increasing power of hardware and 3-D applications, even novice and amateur computer users can make remarkable 3-D graphics. We have carefully crafted this book and disc to enable you to both learn about 3-D graphics and to actually make some of your own.

What you think of this book is important to our ability to better serve you in the future. If you have any comments, no matter how great or small, we'd appreciate you taking the time to send us email or a note by snail mail. Of course, we'd love to hear your book ideas.

Sincerely yours,

A handwritten signature in black ink, appearing to read "David Rogelberg". The signature is fluid and cursive, with a large, sweeping loop at the end.

David Rogelberg  
Publisher, Hayden Books and Adobe Press

You can reach Hayden Books at the following:

Hayden Books  
201 West 103rd Street  
Indianapolis, IN 46290  
(800) 428-5331 voice  
(800) 448-3804 fax

Email addresses:

America Online:	Hayden Bks
AppleLink:	hayden.books
CompuServe:	76350,3014
Internet:	hayden@hayden.com

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
	What is 3-D? .....	4
	Why <i>3-D Starter Kit for Macintosh</i> ? .....	5
	The Software .....	6
	What's Your Task? .....	7
	Illustration and Graphic Arts .....	8
	Product and Package Design .....	10
	Animation and Video .....	11
	Multimedia and Presentation Graphics .....	13
	Architectural Design and Rendering .....	14
	Virtual Reality and Walkthroughs .....	17
	What This Book is About .....	17
	What Level of 3-D User Are You? .....	18
	Beginner .....	19
	Intermediate .....	19
	Advanced .....	20
	Expert .....	20
	The CD-ROM .....	21
<b>2</b>	<b>Getting Started .....</b>	<b>23</b>
	How 3-D Works .....	24
	Modeling .....	25
	Scene Building .....	27
	Animation .....	30
	Rendering .....	31
	Hardware .....	32
	Power Macintosh .....	33
	Accelerators .....	34
	Monitor .....	35
	Display Cards .....	36
	Hard Disks .....	36
	Long-Term Storage .....	36
	Removable Hard Disks .....	38



Scanner .....	38
CD-ROM Drive .....	39
Photo CD .....	40
Drawing Tablet .....	40
3-D Digitizer .....	43
Video Animation Hardware .....	43
Animation Video Deck .....	44
Animation Controller .....	44
Digital Video Hardware .....	45
Video Encoder .....	45
Hardware Tips .....	46
<b>3 Navigating in 3-D .....</b>	<b>47</b>
The Camera .....	49
Pan, Zoom, Tilt .....	51
Lenses .....	52
Moving Camera vs. Lens Length .....	52
Absolute Position: x,y,z .....	56
The Three-Plane View .....	59
Single Perspective .....	61
Virtual Trackball .....	63
Moving and Positioning Objects .....	66
Bounding Boxes .....	66
Grids and Snaps .....	69
Reference Point .....	70
Absolutely Relative .....	70
Orientation: Pitch, Yaw, and Roll .....	71
Saving Views .....	73
<b>4 Basic Shapes and Fonts .....</b>	<b>75</b>
Simple Geometry .....	76
A Sense of Scale .....	77
3-D PostScript .....	77
Adobe Dimensions .....	78
Creating a Vase .....	78
RayDream's addDepth .....	86
Creating a "Jump!" Head .....	87

Working with Type .....	90
Pixar's Typestry .....	90
Wicked Water .....	91
StrataType 3d .....	95
Creating an Igneous Rock Logo .....	95
LogoMotion .....	98
Engraved Type .....	98
Bump-mapped Type .....	100
Embossed Type .....	100
Animated Type .....	100
<b>5 Modeling .....</b>	<b>101</b>
Additive Modeling .....	102
Joining Objects .....	102
Constructive Modeling .....	104
Extrusion .....	104
Basic Extrusions .....	106
Slabs and Blocks .....	107
A Star-Shaped Tube .....	108
Caps .....	109
Subscribing to an Illustrator Outline .....	109
Extrusion on a Path .....	114
Lathing .....	116
Lathing Techniques .....	119
Axis of Rotation .....	119
A Cylinder .....	122
Partial Lathe .....	124
Other Lathed Shapes .....	127
Lofting (or Skinning) .....	130
Sweeping .....	133
Molding and Deformation .....	135
Groups .....	138
Shapes .....	142
3-D Digitizing .....	142

Primitives .....	143
Block.....	143
Cylinder .....	143
Sphere .....	143
Ovoid .....	144
Cone.....	145
Torus .....	146
Pyramids and Similar Shapes .....	147
Bevels.....	149
Bevels in Animation.....	150
Solid Modeling .....	151
Booleans .....	152
Living Forms .....	157
Humans and Animals .....	157
Playmation .....	157
Plants and Trees .....	158
AMAP .....	158
Tree Professional .....	159
What Not to Model .....	160
Bumps and Textures .....	160
Background Details .....	161
Rendered Booleans .....	162
Displacement Mapping .....	162
3-D and CAD .....	162
Splines vs. Polygons.....	164
<b>6 Scene Building .....</b>	<b>169</b>
Setting the Stage .....	170
Using Snaps and Grids .....	170
Constraining and Linking .....	170
Parents and Children .....	171
Rotation .....	171
Joints .....	172
Stretching and Squashing .....	174
Backgrounds.....	174



Rendered Backgrounds .....	175
Bitmaps to Models .....	176
Terrainman .....	177
CyberMesh .....	178
An Island unto StrataStudio Pro .....	179
Infini-D .....	182
Showplace Terrain .....	184
Imported Backgrounds .....	184
Scenery Generators .....	184
Vista Pro .....	185
KPT Bryce .....	186
Transparent Backgrounds .....	188
Perspective Matching .....	189
Sketch! .....	189
General Perspective Matching .....	190
Shadow Casting on Backgrounds .....	190
Cameras .....	191
Orientation .....	191
Depth of Field .....	191
Atmosphere .....	191
Fog .....	192
<b>7 Materials .....</b>	<b>195</b>
Surface Quality .....	197
How We See .....	198
Color .....	199
Mirror Reflection .....	200
Specular Reflection .....	202
Diffuse Reflection .....	203
Ambient Reflection .....	204
Transparency .....	205
Refraction .....	206
Glow .....	207
Texture Maps .....	208
Color Mapping .....	209
Seamless Textures .....	209

Position .....	210
Scale .....	210
Tiling .....	211
Direction .....	212
Projections .....	212
Planar Mapping .....	213
Making a Box Label .....	213
Parametric Mapping .....	217
Cylindrical Mapping .....	217
Spherical Mapping .....	218
3-D Painting .....	219
RayPainting .....	219
Designer's Painting .....	220
The G-buffer .....	220
Bump Maps .....	221
Bump Mapping Type on a Plain Surface .....	224
Surface Maps .....	227
Glow Maps .....	228
Glowing Sign .....	229
Labels .....	232
Procedural Shading .....	233
Adding Textures .....	235
Texture Libraries .....	236
Creating Textures .....	236
Textures with Bump Maps .....	236
A Seamless Texture for Sketch! .....	237
Kai's Power Tools .....	239
TextureScape .....	243
XAOS Tools Terrazzo .....	245
Gallery Effects and Paint Alchemy .....	246

<b>8</b>	<b>Lighting .....</b>	<b>247</b>
	Daylight .....	250
	Studio Lighting .....	251
	Night Lights .....	252
	Light Types .....	253
	Point Lights .....	253
	Spotlight .....	254
	Spotlights for Highlights .....	255
	Ambient Light .....	255
	Distant Light .....	257
	Pan Light .....	257
	Back Light .....	258
	Setting Lights .....	258
	Setting Lights in Sketch! .....	259
	Visible Lights and Lamps .....	261
	Glowing Objects .....	261
	Shadows .....	263
	Gels and Gobos .....	264
	Using a Gel in StrataStudio Pro .....	265
	Electric Image Lights .....	267
	Time-of-day Lighting .....	268
<b>9</b>	<b>Animation .....</b>	<b>269</b>
	Animation Basics .....	270
	Key Frames and Events .....	271
	Scene vs. Event Key Framing .....	272
	Timelines .....	275
	A Simple Flying Logo .....	277
	Electric Image Animation System .....	283
	Presenter Professional 3.0 .....	284
	Camera Moves .....	286
	Flying Camera Logo .....	286

Sound .....	289
Presenter Pro's Microphones .....	289
Synching to Sound .....	290
Ease-in and Ease-out .....	291
Smoothing Motion .....	291
StrataStudio Bank Turns .....	292
Manually Smoothing Motion .....	296
Prepare to Morph .....	296
Morphing a Glass in Infini-D .....	297
Rotoscoping .....	300
Film Loops .....	300
Morphing for Loops .....	302
Morphing Tools .....	302
Particle Animation .....	302
Frame Rate and Resolution .....	304
Uses of 3-D Animation .....	305
Video .....	305
Multimedia .....	306
Film .....	306
Virtual Reality .....	307
<b>10 Rendering .....</b>	<b>309</b>
Rendering Methods .....	310
PostScript Blending .....	311
Wireframe .....	311
Hidden-line Rendering .....	312
Flat Shading .....	314
Gouraud Shading .....	315
Phong .....	316
Phong Reflections .....	318
Ray Tracing .....	318
Mirror Reflection .....	319
Recursion .....	319
Transparency .....	320
Refraction .....	320

Radiosity .....	321
RayPainting .....	322
“Modeling” with Transparency .....	323
Boolean Rendering .....	330
Rendering Quality .....	331
Realism .....	331
Detail .....	332
Antialiasing .....	332
Wireframe Complexity .....	333
Smoothness .....	334
Speed .....	335
Speed Factors .....	335
Rendering Type .....	335
Scene Complexity .....	336
Image Size .....	336
Mirror Reflections and Glass Refractions .....	336
Antialiasing .....	337
Lights .....	337
Atmospheric Effects .....	337
Texture and Environment Maps .....	337
Batch Rendering .....	337
<b>11 Working with Images .....</b>	<b>339</b>
Compositing .....	340
The Alpha Channel .....	342
File Formats .....	342
Color Depth .....	345
1-bit .....	345
4-bit .....	345
8-bit .....	345
16-bit .....	346
24-bit .....	347
32-bit .....	347
64-bit .....	347

Animation Formats .....	347
Numbered PICT .....	348
QuickTime .....	348
PICS .....	351
ELAS Image .....	351
Adobe Photoshop .....	352
Compositing Images .....	352
Faking a Shadow .....	356
Motion Blur .....	360
ColorIt! .....	363
Painter .....	363
DeBabelizer .....	364
Desktop Publishing .....	367
Making and Using a Clipping Path .....	368
Color Management .....	371
Animations .....	371
QuickTime .....	371
Video .....	372
Frame-by-Frame Recording .....	373
Software Animation Controllers .....	373
DQ-MAC232 and EVO-9650 .....	374
DQ-Animaq .....	375
Electric Image Animation System .....	375
Exabyte Drivers .....	376
Field Rendering .....	376
Multimedia .....	377
Authoring Tools .....	378
Color Palettes .....	379
Creating a Console .....	380
Animating Still Images .....	383
<b>12 RenderMan .....</b>	<b>385</b>
The 3-D "Printer" .....	386
Choosing MacRenderMan .....	387



RenderMan Shaders .....	388
Surface Shaders .....	389
Displacement Shaders .....	389
Volume Shaders .....	390
Looks .....	390
Looks vs. Shaders .....	391
Animated Shaders .....	392
Glimpse .....	393
Programming Shaders .....	393
The Valis Group .....	394
Shader Toolkit .....	394
<b>13 Advanced Topics.....</b>	<b>395</b>
Transporting Models .....	396
3-D File Formats .....	396
DXF .....	397
IGES .....	398
RIB .....	398
Holding Your N-Gons .....	398
Converting Polygons to Splines .....	399
Kandu's Cadmover .....	399
Smoothing .....	400
Acceleration .....	401
Faster Hardware .....	401
Power Macintosh .....	401
Power Mac "Native" .....	403
Daystar PowerPro .....	403
YARC Boards .....	403
YARC Support .....	404
Distributed Processing .....	405
How Distributed Rendering Works .....	405
NetRenderMan .....	407
Render Pro .....	408
BackBurner .....	408

DreamNet .....	408
Sketch!Net .....	408
Mac vs. Other Platforms .....	408
Apple's Advantages .....	410
Tapping into SGLs .....	412
RenderMan Unix .....	413
FlashTracer and TorqueWare .....	413
IDT CZAR .....	413
Apple's 3-D Future .....	413
3-D QuickDraw ("Escher") .....	414
3-D Hardware (SuperMac 3D) .....	415
3-D Metafile (3D PICT) .....	416
<b>A Macintosh 3-D Applications .....</b>	<b>417</b>
Acuris Inc. ....	423
Clip Model Libraries .....	423
Adobe Systems Inc. ....	423
Dimensions .....	424
Illustrator .....	424
Photoshop .....	425
Premiere .....	425
Aldus Corp. ....	426
Fetch .....	426
After Effects .....	426
After Image .....	426
Gallery Effects .....	427
Alias Research Inc. ....	427
Sketch! .....	427
The Alpha Channel BBS .....	428
America Online .....	429
3DSIG .....	429
Art of Fact .....	429
Terrainman .....	429
Artbeats .....	430

Artifice Inc. ....	430
Design Workshop .....	430
ASDG Inc. ....	430
Abekas Driver .....	431
Ashlar Inc. ....	431
Vellum 3D.....	431
Design Reality.....	431
Autodesk Inc. ....	432
AutoCAD Release 12 .....	432
Autodessys .....	432
form•Z .....	432
Byte by Byte Corp. ....	433
Sculpt 3D .....	433
Sculptor .....	434
Sculpt 4D .....	434
CompuServe .....	434
Multimedia+/Modeling & Animation .....	434
CrystalGraphics Inc. ....	434
Crystal TOPAS Mac .....	435
DayStar Inc. ....	435
PowerPro 601 .....	435
DeltaPoint Inc. ....	435
DeltaGraph Professional .....	435
Diaquest Inc. ....	436
DQ-Animaq .....	436
Mac-232 .....	436
Dynaware USA Inc. ....	437
DynaPerspective .....	437
Electric Image Inc. ....	437
Electric Image Animation System .....	437
ElectroGig Inc. ....	438
3DGIG .....	438

Equilibrium Technologies .....	438
DeBabelizer .....	438
DeBabelizer Lite .....	439
Form and Function .....	439
Wraptures and Wrapture Reels .....	439
Fractal Designs Inc. ....	439
Painter .....	439
Gibbs & Associates .....	440
Virtual Gibbs .....	440
Graphisoft .....	440
ArchiCAD .....	440
Atlantis .....	441
Zoom .....	441
Graphsoft .....	442
MiniCAD+ .....	442
Hash Inc. ....	442
Animation Master .....	443
HSC Software Inc. ....	443
Kai's Power Tools 2.0 .....	443
KPT Bryce .....	444
Immersion Corp. ....	444
Personal Digitizer .....	444
Integrated Device Technologies Inc. ....	445
CZAR .....	445
Intergraph Corp. ....	445
MicroStation Mac .....	446
Kandu Software Corp. ....	446
Cadmover .....	446
Knoll Software .....	447
CyberMesh .....	447
Missing Link .....	447
Knowledge Revolution .....	447
Working Model .....	447

## Contents

Macromedia Inc. ....	448
Swivel 3D Professional/SwivelMan .....	448
LifeForms .....	449
Macromedia Three-D.....	449
MacroModel .....	450
ModelShop II .....	450
Extreme 3D .....	450
McQ .....	451
MacAnimator Pro .....	451
Mira Imaging Inc. ....	452
HyperSpace .....	452
MicroFrontier .....	452
ColorIt! .....	452
Onyx Computing.....	452
Tree Professional .....	453
PaXar Technologies Inc. ....	453
Touch 3-D.....	453
Pixar .....	454
128 .....	454
Looks CD .....	454
MacRenderMan .....	454
NetRenderMan .....	455
Showplace.....	455
Typestry .....	455
RayDream Inc. ....	456
addDepth .....	456
DreamNet .....	456
RayDream Designer .....	457
Silicon Graphics Inc. ....	457
SketchTech.....	458
UpFront .....	458
Specular International .....	458
Backburner/YARCBurner .....	458
Collage.....	458

Infini-D .....	459
LogoMotion .....	459
Replicas.....	460
TextureScape .....	460
Strata Inc. ....	460
Fractal Terrain Modeler .....	460
Rend•X .....	460
Render Pro .....	461
StrataType .....	461
Strata Clip 3d .....	461
StrataVision 3d .....	461
Studio Pro .....	462
Strata Virtual .....	463
Torque Systems Inc. ....	463
TorqueWare .....	463
The Valis Group .....	464
PixelPutty .....	464
PixelPutty Solo .....	464
RenderMan shaders .....	465
Video Bits .....	465
Flash Tracer .....	465
VideoFusion Ltd. ....	465
VideoFusion .....	465
View by View .....	466
AMAP .....	466
Turbo 3D .....	466
Viewpoint DataLbs .....	467
3-D datasets .....	467
Virtus Inc. ....	467
Virtus vr .....	467
Virtus WalkThrough .....	467
Virtus WalkThrough Pro .....	468
Visual Information Development Inc. (VIDI) .....	468
Presenter Professional .....	468



Xaos Tools Inc. ....	469
Fresco .....	469
Terrazzo .....	470
Paint Alchemy .....	470
YARC Systes Corp. ....	470
NuSprint/MacRageous II/Zuma .....	470
Shade III .....	471
<b>B 3-D Starter Kit for Macintosh CD-ROM .....</b>	<b>473</b>
Macintosh 3-D Workshop Interactive .....	474
Alpha Channel BBS .....	474
Absynthe's Power Tools .....	475
Commercial 3-D Software .....	475
Other Try-out Software .....	476
Shareware & Freeware .....	476
Image Galleries .....	477
Sample Models .....	477
Backgrounds and Textures .....	478
Wagstaff's Favorites .....	478
The Alpha Channel .....	478
HyperCard Player .....	478
<b>Index .....</b>	<b>479</b>



**C h a p t e r**

**1**

# **Introduction**

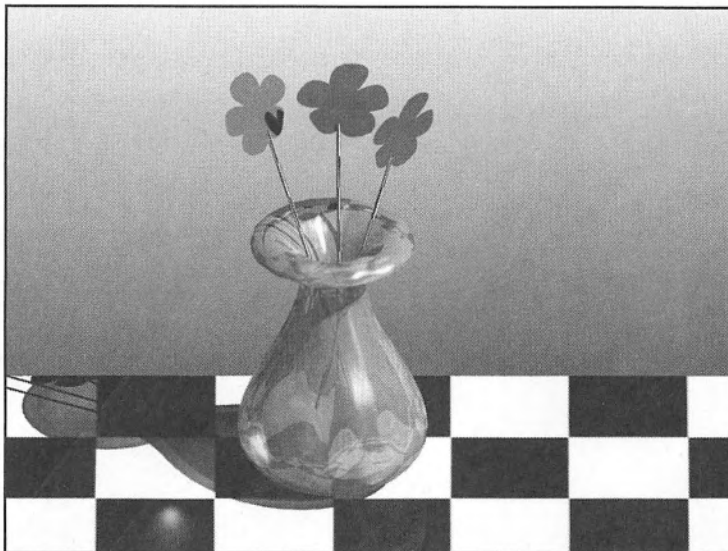
**W**hat sculpture is to painting, three-dimensional (3-D) graphics is to illustration. Three-D graphics is the modeling and rendering of 3-D objects, including all of the visual information found in the real world: perspective and depth, light and shadow, form and texture, and sometimes motion. Using these effects, a designer can produce images of imaginary worlds that are so realistic that they resemble photographs.

Three-dimensional graphics can be a powerful medium. For example, a package designer can show a perfume bottle in its display case long before the bottle is made; an illustrator can create a company logo out of chrome, complete with reflections of the surrounding landscape; a video animator can wrap a decorative map of the world around a spinning globe then send a rocket into orbit around his planet; a corporate trainer can create a realistic simulation of an airplane cockpit; an architect can set a perfectly rendered high-rise in the middle of an urban landscape.

In contrast, an artist using traditional materials must photograph intricate scale models or turn to friskets and airbrushes. Realistic animations can take the problem to the extreme; in the past, they required thousands of individual hand-painted cells. Even Macintosh 2-D techniques are taken to their limits when simulating traditional perspective design (see figure 1.1). Adobe's Illustrator and Aldus FreeHand users must draw in perspective and create many blends to simulate shadows and highlights. Users of Adobe's Photoshop and Fractal Design's Painter must create masks and paint each element of a scene to affect realism.

Using 3-D software, you need not skip details such as shadows and reflections simply because they are too difficult to paint. More importantly, 3-D software enables you to move around and change your point of view. In contrast, once you create an image in 2-D, there is little you can do if you need to see it from a new angle. If, for example, you need to see more of the front of the motorcycle you've drawn, you must start over from scratch. With 3-D, you simply move the camera that controls your point of view, or rotate the bike just as you would a real model, and render the image again. Because your models are fully-formed objects in space, you can easily view them from the top or side, front or back, inside or outside. Three-D is a vehicle users can ride into any space (see figure 1.2).

The Apple Macintosh is an ideal graphics platform for artists and designers because it is easy to use and is capable of producing a spectrum of professional artwork. It also has the advantage that it easily can move files between different programs and other users. Recent Macs include sophisticated audio and video (AV) capabilities that make them ideal for media producers as well. Yet, 3-D is one of the most demanding applications for the computer. The millions of calculations required to render light and shadow in a 3-D world taxes even the most powerful Macs.

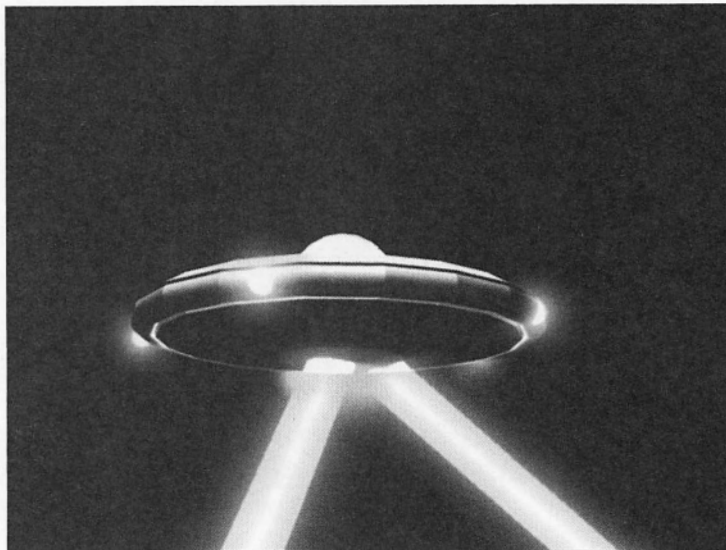


**Figure 1.1** Modeled and rendered in Alias Sketch!; notice the reflections and highlights in the glass surface

Fortunately, Apple is constantly building more powerful, less expensive computers. This has brought the computing performance required for 3-D into the range of desktop designers and illustrators as well as multimedia developers, video producers, architects, engineers, and industrial designers. The Power Macintosh, introduced earlier this year, is a milestone in the development of the Macintosh. Power Macs instantly elevated computing performance as much as ten-fold in critical performance areas. And this happened without price increases (in fact prices even decreased slightly). In a blink, the Macintosh went from being second-class 3-D grunTERS, to being one of the best 3-D solutions available. Meanwhile, add-on graphics and system accelerators, as well as distributed processing software, have also evolved to further help speed things along.

You need not despair if you don't own one of these new high-class machines. Not all 3-D programs devour Macintosh system resources. There are a number of powerful 3-D packages that can run well on low- to mid-range Macs; and system accelerators, such as those from DayStar, offer the possibility of upgrading to PowerMac performance without starting from scratch.

The first edition of the *3-D Starter Kit for Macintosh* (which was entitled *Macintosh 3-D Workshop*) provides a testimonial to the explosive growth in Macintosh 3-D graphics. In the first edition there were about 50 products that were likely to be of interest to the 3-D artist. Now, only a year or so later, there are well over 100 with more on the way.



**Figure 1.2** *Three-D makes it easy to travel to alien worlds*

For an eye-opening encounter of what can be done with 3-D on the Mac, try opening any one of the sample image galleries included on the CD-ROM (the Electric Image Animation System and Specular Infini-D animations are particularly stunning). There are dozens of fantastic examples that will give you a good idea of what's possible. Follow the directions for installing and running the "3D Starter Kit Interactive Tutorial" in appendix B, and enter the tutorial's Gallery to see many of the book's most interesting images in color, along with descriptions of how they were created.

## What is 3-D?

The field of 3-D graphics, as created on the Mac, is divided into several tasks: modeling, scene building, rendering, and animation. All are closely-related activities, but each takes highly-specialized forms so they are treated separately. Often, the different tasks require different software applications.

- **Modeling** is the creation of 3-D objects. If you've built model airplanes, worked clay into figurines, or constructed a piece of mail-order furniture ("some assembly required") you get the idea. Models are the building blocks used to create 3-D scenes.

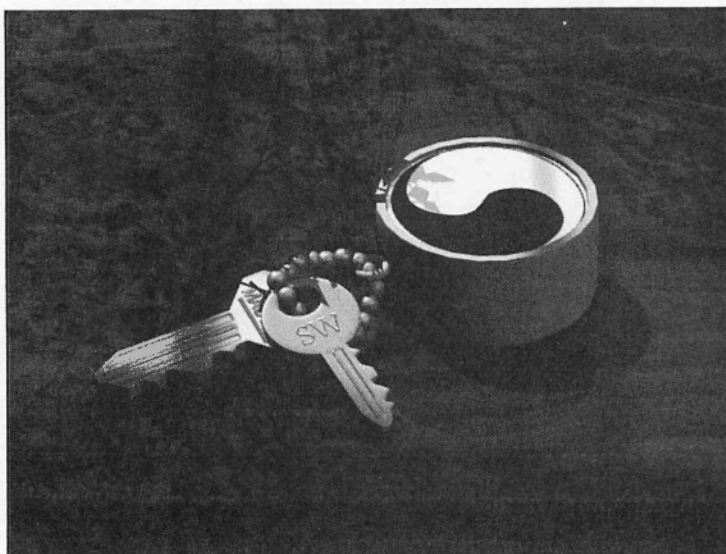
- **Scene building** is where you put the models in their places. It also is where you give them textures and physical properties, and add lighting and atmospheric effects. This is analogous to assembling a toy train set or working as both the stage director, lighting director, and costume designer in a play.
- **Animation** is scene building with the added element of time. It enables you to move objects and cameras to create movies depicting events in a 3-D world. Once you've set up your trains, animation is where you tell them where to go and what to do. If you have anything in common with Uncle Fester, you may even decide to blow up the tracks!
- **Rendering** is the final production of finished images. When making models and building scenes, you are saying, "This is how I want the picture to look." The renderer is the software that takes these directions and actually paints a picture. It takes into account the geometry and textures of models, the position and character of lights, and many other factors, and generates a realistic 3-D scene.

## Why 3-D Starter Kit for Macintosh?

Some of the reasons for which I first decided to write *3-D Starter Kit for Macintosh* now seem almost laughable. At the time, I talked about the breathtaking evolution of the Macintosh, and how it was possible for the first time to do realistic 3-D for less than the price of a new car. But at the time, a single complicated rendering may have required an entire eight-hour work day to complete; now, thanks to Power Macs, that same rendering can be done in an hour. New Macs not only have realistic 16-bit color built-in, some even support two monitors. High-speed graphics accelerators are now affordable, but with the Power Macs, they are now more of a luxury than a necessity. And even the slowest Macs on the market are fast enough to handle most 3-D software with success.

On the other hand, some things haven't changed. No matter how fast, computers still communicate with the user via flat screens, flat print, and flat video. Most computer users are comfortable thinking in only two dimensions, so 3-D requires considerable explanation. The lexicon of new terms is long and many techniques are unique to the format. However, once the hurdles are behind you, 3-D opens up a world of realistic imagery.

*3-D Starter Kit for Macintosh* is intended to be a primer in the techniques and concepts of 3-D, as well as a resource guide to help you find and sort through the many available tools. The included CD-ROM provides the software (both demos of commercial applications, and a library of shareware and freeware) to help you develop your understanding with plenty of hands-on experience.



**Figure 1.3** *Modeled and rendered in StrataStudio Pro*

For those who use 3-D software already, such as architects using CAD for design and drafting, *3-D Starter Kit for Macintosh* will help you explore the techniques needed to render 3-D designs with a high level of realism. Beginners will find an eye-popping number of new 3-D tools designed specifically to make 3-D easy for illustrators and first-time animators. In between are many thousands of graphics professionals and enthusiasts who use the Mac on a daily basis, but are looking for new ways to make unusual and interesting images and animations. For these users, Mac 3-D software offers limitless opportunity for creativity.

## **The Software**

Even more impressive than the new Macs and the cast of supporting peripheral hardware is the unprecedented development of 3-D software. At last count, there were over 100 products and content providers for Macintosh 3-D users. These include modelers, scene builders, animators, and renderers, as well as a host of supporting products: texture libraries (materials such as marble and granite that you can apply to your models); texture generators that enable you to create your own materials; 3-D illustration packages that create realistic-looking PostScript artwork; 3-D type programs; distributed rendering software; CAD programs for precision design and drafting; modeling utilities for creating unusual shapes; and many other products. Many 3-D users also make use of a gaggle of 2-D software such as Adobe Photoshop and Premiere, Kai's Power Tools, Aldus After Effects, and Fractal's Painter.



Each of the many categories of 3-D products requires a good deal of further explanation. Some modelers can construct solids into which you can carve realistic holes, while others deal only in hollow surfaces, use polygonal surfaces, or are based on smooth splines. Renderers offer many different options that may or may not render textures, reflections, refractions, and shadows. Some modelers include flat rendering, but not smooth rendering; and some scene builders offer animation, while at least one renders only still frames.

There are industrial-strength software products (such as form•Z and Electric Image Animation System) that target the professional video and film or industrial design market and sell for thousands of dollars. They also require many hours of patient study. Other products, such as RayDream's Designer or Adobe Dimensions, are aimed at the illustrator and cost less, and are as easy to use as most word processing programs.

*3-D Starter Kit for Macintosh* attempts to explain the tools offered by these diverse programs. Appendix A, "Macintosh 3-D Products," will guide you to where specific products fall within the wide spectrum of 3-D applications. Meanwhile, as many of the programs as possible receive at least a cursory introduction through tutorials in the book's chapters.

## What's Your Task?

Knowing what you want to do with 3-D will certainly help you figure out where to get started as far as selecting software and reading this book. On the other hand, the more you know, the more uses you'll discover. There are far too many 3-D applications for me to have a prayer of listing them all. The categories I've included here will serve as generalizations for the most common users of 3-D software. If your stated goal isn't on the list, don't despair. A stage designer, for example, makes use of many of the same techniques as an architect and an illustrator. Here are some categories where 3-D finds a home:

- Illustration and graphic arts
- Product and package design
- Animation and video
- Multimedia and presentation graphics
- Architectural design and rendering
- Virtual reality and walkthroughs

## Illustration and Graphic Arts

Just as you became accustomed to creating artwork on a computer in two dimensions and mastered the difficult world of blends and gradient fills, you're suddenly faced with programs such as Adobe's Dimensions, RayDream's addDepth and Designer, Alias' Sketch!, Pixar's Typestry, Specular's LogoMotion, StrataType 3d, and HSC's KPT Bryce, all of which are designed to provide easy entry into the world of 3-D. If Adobe's Illustrator, Aldus FreeHand, and Deneba's Canvas are your tools of trade, you will find help on importing and adding light and depth to PostScript art in chapter 4, "Basic Shapes and Fonts," which tackles 3-D PostScript programs.

Three-D type, also discussed in chapter 4, should appeal to almost every designer. Graphic designers, like illustrators, are under intense deadline pressure and the demands of their craft are severe. Now a designer can create 3-D beveled type quickly without friskets and airbrushes; it is a simple matter to chisel type out of marble, to wrap text around a sphere, or to turn a logo into shining gold (see figure 1.4).



**Figure 1.4** Logos and type get a new sense of light and depth from 3-D treatment

Artists used to the bitmapped painting and image editing world of Adobe Photoshop and Fractal Designs' Painter and ColorStudio can find interesting material throughout this book. It's now fairly common to find 3-D elements combined with the best elements of 2-D design. The texture and depth provided by 3-D add amazing richness to illustration.

Examples of using 3-D images in 2-D applications can be found in chapter 11, “Working with Images.” With only a few exceptions, still images from 3-D programs are generated as high-resolution bitmap, PICT, TIFF, or other standard image files. You can, for example, import 3-D images and combine them with scanned photos. Or, you can use images from your image editing and paint programs as backgrounds and textures within 3-D applications.

Many illustrators have come into computing reluctantly as work increasingly depends on instant color separations and their capability to combine earlier creations into new projects. Fortunately, some 3-D applications are simplified for illustration. For example, with Adobe’s Dimensions and RayDream’s addDepth, you can turn Bezier curves into 3-D models. You can then place lights, textures, and labels, and render images that would take hours using Illustrator’s blends and fills alone. The resulting images retain the benefits of illustrations created in a PostScript drawing package, including file formats, resolution independence when printing, and relatively compact file sizes. You can even open a 3-D image created in addDepth or Dimensions and edit it using FreeHand or Illustrator. Examples are provided in chapter 4, “Basic Shapes and Fonts.”

Alias’ Sketch! and Ray Dream’s Designer, meanwhile, are full-featured 3-D programs designed for ease of use and targeted at print illustrators and designers. Unlike Dimensions or addDepth, they are capable of creating complex models which can be matched to the perspective of scanned photographs. They both offer photorealistic ray tracing and export models to other 3-D applications, and their renderings are in bitmap format.

Three-D offers a unique opportunity to the illustrator. Three-D models can be used over and over in many new ways. A detailed model of an airplane can be duplicated and combined with models of buildings and viewed from a distant point to illustrate an airport, or you can zoom in to view the passenger compartment or cockpit. A corporate logo can be stored away and extruded, engraved, or embossed at a moment’s notice. It’s even easy to take artwork and “map” it onto new objects. A label, for instance, can easily be wrapped around a bottle, box, sphere, or far more complex shapes. Just as with 2-D computer drawing programs, 3-D lets you create one object and duplicate it if you need many copies. Imagine painting a series of pictures of a car going through a turn. In 3-D, you simply model the car once, and render it from several angles.

## Product and Package Design

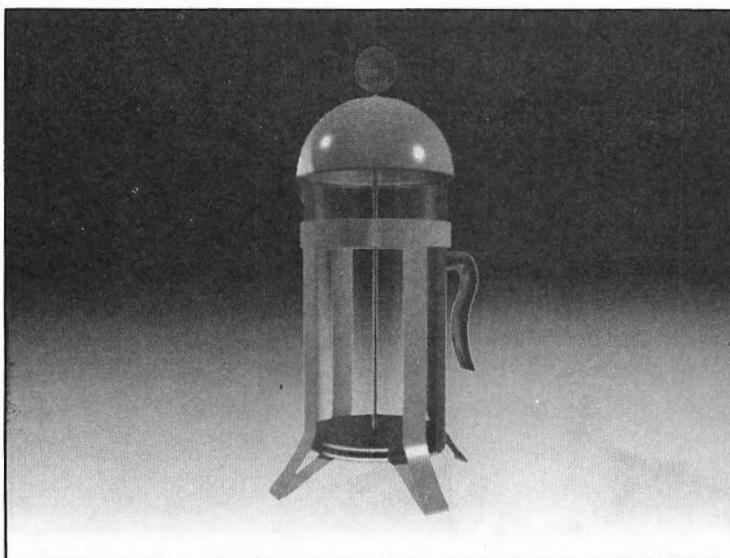
Product and package designers were among the first to embrace the power of 3-D computing. Few things will sell a concept better than a photorealistic picture or animation of a finished design (see figure 1.5). With 3-D software, a designer can build a coffee pot with accurate metal, plastic, and glass parts. The result is then placed in a flattering setting. Lights are positioned, and the image is rendered as though you were taking a photo of the package. The result is indistinguishable from a photo, with the notable exception that you don't have to go through the time and expense of actually building and photographing a model. Beyond designing the product itself, you can bring packaging and marketing concepts to life, proving to the venture capitalists that your newest invention is not only a bright idea, but a marketable one as well.

Industrial (product) design software has to be quite powerful to depict the complex shapes used in real manufactured objects, and it has to be accurate enough that designs can be taken from the modeling and rendering phase all the way to production of a finished product. On the Mac, Sculpt 3D, Presenter Professional, form•Z, Zoom, and Shade III all have the features to meet these tough demands.

CAID, or computer-aided industrial design, implies that what you're doing may eventually go from the design of a 3-D model to the actual production of plans and manufacturing specifications. Users with these needs will sometimes need to turn to so-called CAD products with special CAID modules, such as solid modeling. Other programs considered general modelers, such as form•Z, also have the capacity to produce 2-D working drawings and plans.

If you are an experienced designer, but new to the Macintosh, this whole book may prove enlightening. If you have experience with 3-D on high-end dedicated graphics workstations—as many industrial designers now do—you may also want to investigate the discussion of Power Macintosh, acceleration, and Apple's future, that explains how you can, or soon will be able to, squeeze Silicon Graphics performance out of your Macintosh system.

Production-oriented designers who intend to produce plans and other working drawings, as well as possibly taking designs to computer aided manufacturing (CAM) systems, can turn to the discussions of CAD presented in the Modeling chapter (chapter 5). A well-made 3-D rendering is virtually indistinguishable from a photograph of the actual object, but it is even better if it prevents costly prototyping and production mistakes. Virtual Gibbs, a Macintosh CAM program, for example, can actually show you step-by-step how a part will be machined before the software actually controls the machining process.



**Figure 1.5** Industrial designers convey their ideas with realistic renderings; this image was modeled in VIDI's Presenter Professional and rendered with MacRenderMan

Three-D has unlimited applications in industrial design. A popular toy-making firm, for instance, models and renders a new design in VIDI's Presenter Professional, imports the result into Macromedia Director, and adds sounds and active buttons. The company then shows the new "toy" to children, who can push the buttons and get real responses. Based on kids' reactions, the company decides whether or not to move to the next stage of development.

Industrial designers can take advantage of a variety of unique technologies. One of those is stereolithography—or 3-D printing. This uses devices that build a real 3-D model out of plastic, based on the information provided by a 3-D program.

Another interesting product, called Touch 3D, unfolds a 3-D model into flat pieces that can be printed onto cardboard, cut out, and reassembled into a real 3-D model.

## Animation and Video

Animation, whether for QuickTime movies, broadcast video, or film, is the ultimate expression of the 3-D art. Just about anything you can do in 3-D, you can set to motion. This book explains key frames, machine control, alpha channels, and field rendering—all for taking graphics to video. Also described are esoterica such as using animations as surface textures (sometimes called "rotoscoping"), beveling shapes to catch highlights, morphing, deforming, and exploding objects

(see figure 1.6). Environment maps, which are critical in animation, make moving objects seem like part of a bigger world.



**Figure 1.6** *Mac 3-D makes animation scream; this was modeled in Sketch! and animated in Electric Image Animation System*

For humans, animation is a labor-intensive task requiring 24 to 30 images for every second of film or video. Unlike a human illustrator, a computer is tireless. Build a model, specify starting and ending points (key frames) and paths for the motion of objects, lights, and cameras, and the Macintosh will render the frames between the points. Not only that, but 3-D software also enables you to automatically control acceleration, deceleration, the tension of an object's path, and other elements of animation that prevent unnatural robot-like movement.

The most recognizable purveyors of animation and special effects have used these techniques extensively in Hollywood. Television producers use Macintosh-generated fly-by titles, spinning logos, animated illustrations, three-dimensional maps, and a host of other slick graphics. In fact, viewers will be hard-pressed to find a television station that doesn't broadcast flying 3-D logos at every commercial break.

Three-D is such a staple of Sunday sports broadcasts and beer commercials that one company has released a collection of sports balls (clip models and textures of golf balls, footballs, basketballs, and so on) that producers can send flying into America's living rooms.

Even the showplace of television graphics, MTV, has taken a serious interest in Macintosh-generated 3-D animation. Work created using Electric Image Animation System and Macromedia Three-D appears regularly from this frenetic broadcaster. Nickelodian and other television shows have followed suit by hiring creative (and economical) Mac animators for special effects and animation.

While many professional animators currently use expensive workstations (particularly those from Silicon Graphics) to churn out thousands of renderings, performance differences between these computers and the Macintosh are beginning to blur. The highest-performance Power Macintosh exceeds SGI's lower-end workstations in rendering performance. And other factors, such as onscreen rendering and modeling performance, are gradually catching up on the Mac's end. Even though it's possible to do things faster—and sometimes better—on SGI's, there's no comparison in price and convenience. A fully-loaded Power Macintosh running the gamut of 3-D software and all of the other tools (such as Photoshop and other paint programs, texture generators, post-processing programs, and animation controllers) costs half the price of a 3D-ready SGI running a single 3-D animation package. Chapter 13, "Advanced Topics," has a thorough description of how these systems and MS-DOS machines compare with the Mac and what's in store for the future.

## **Multimedia and Presentation Graphics**

There will come a day, soon, when computer interfaces will transform from flat icons piled onto a flat desktop, to floating objects in virtual 3-D worlds. These objects, just like Macintosh folders today, will be "live," and you'll be able to click them to open other rooms, drag objects around, pile them into corners, and so on. Imagine a system where high-priority items are filled with "helium" and low priority "lead" objects sink to the bottom. There could be electronic equivalents of office furniture into which you could place your Rolodex files and computer manuals, of course all of this could be shrunk to minuscule sizes and stored appropriately. (Eventually, computer consulting and house-cleaning services will merge to form hybrid companies.)

But long before computer operating systems become three-dimensional, multimedia designers will have defined what a good 3-D interface will look like. Many examples of 3-D multimedia are already on the market. Iron Helix and Journeyman Project are two examples. And Myst, heralded as the first smash hit in multimedia, owes most of its success to gorgeous 3-D animation and graphics done with StrataVision 3d on the Mac.



With multimedia, a developer of training presentations can create a simulated switchboard, for example, where a train operator learns to reach for the appropriate stop and go buttons without ever stepping inside a train. This kind of simulation demands realistic 3-D renderings of the switchboard, using separate versions with warning lights turned on and off. Woe to the designer who has to paint all those switchboards by hand! But doing it with a 3-D on the Mac is relatively easy.

VIDI's Presenter Professional has a feature that enables you to attach a "speaker" to an animated 3-D object. As it flies by, you get realistic doppler effects, echoes, and other 3-D sound effects. The potential for 3-D sound effects in multimedia productions is just beginning to be explored.

Animated buttons, interactive rooms, animations synchronized with sound, and optimized color palettes are all part of the 3-D multimedia endeavor. Chapter 11, "Working with Images," describes the use of 3-D in these interactive and otherwise dazzling multimedia productions.

Presentation graphics, sometimes considered multimedia, is fast becoming filled with 3-D animations and 3-D charts and graphs. It's common to promote products that don't yet exist by including 3-D renderings of prototype designs in sales presentations.

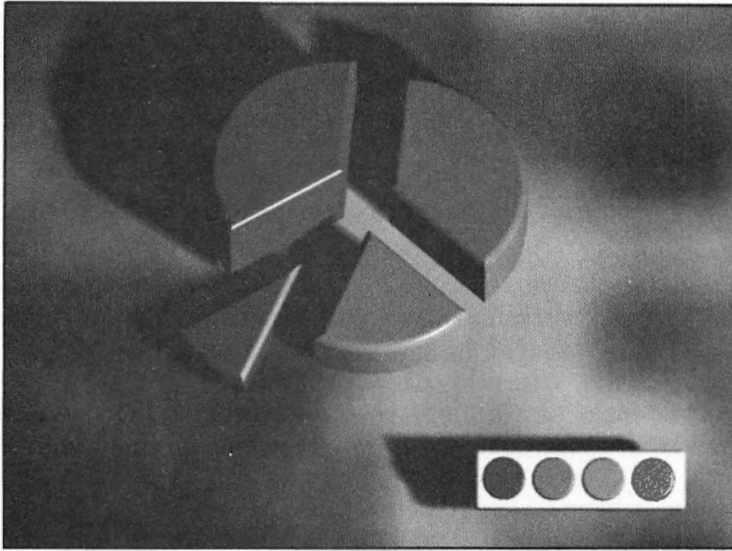
## **Architectural Design and Rendering**

The acronym CAD means literally, "computer-aided design." But as anyone can now perceive, this is a generalization that has lost much of its specific meaning. Just about every kind of design is now done on computers—from brochures and newsletters, to sculpture and stage lighting, to jet engines and high-rises. However, the term CAD is still used to denote a specific type of software—software that's extremely accurate, provides all of the tools needed to complete a given type of design, and is geared towards producing working drawings more than creating rendered images.

Architects have a whole world of 3-D CAD applications at their disposal. Three-D CAD is so powerful that 2-D-only applications are falling by the wayside. Architects that can't or won't make the move to CAD are having a hard time competing with those that do.

An architect can greatly enhance the prospects of selling a design when a client can see a photorealistic rendering of a building's sparkling interior, tastefully decorated, with sunlight streaming through windows.





**Figure 1.7** *There are endless uses for 3-D in multimedia, including high-powered business presentations*

It is now almost common to design a structure in 3-D and create realistic 3-D renderings to show clients. Some traditional architectural renderers balk at the thought of giving up pen for computer, but good artists are in more demand now than ever, simply because it's more economical to do renderings with a computer. Despite computers, it still requires a skilled artist to make artistic 3-D renderings. For the artist, the computer is a powerful tool. If a building goes through changes in plans, you can simply update your model and let the computer re-render the image.

Even landscape architects can use Macintosh 3-D graphics. Two programs, AMAP and Tree Professional, are specifically designed to generate accurate 3-D models of plants and trees. These can be included in architectural models for rendering realistic scenery.

Animated renderings and real-time walkthrough software (sometimes called "virtual reality") can take clients on a walk down hallways to enter rooms and look around, peek out windows at other parts of a compound, or view a new design in the context of surrounding landscape. It's possible to combine aerial videos of a site with animated renderings of a new structure so that they blend together perfectly. Some programs enable you to set the time of year and watch where shadows will fall in a courtyard at different times on different days.

With the help of IRIS printers and good service bureaus, it's now easy and quite economical to produce high-resolution, high-quality prints of renderings at display sizes using rag and watercolor papers. It's almost impossible to distinguish these beautiful prints from hand-painted ones—except for their extraordinary accuracy and detail.

While CAD applications are highly specialized (some for architecture, some for mechanical design, and so on), architects often use general 3-D applications to “rough out” their designs. Several 3-D programs are available specifically for this purpose. Artifice's Design Workshop and SketchTech's UpFront both feature powerful tools for quickly assembling forms and spaces.

Three-D modelers are used for creating free-form objects. Architects use these to design parts of their structures, such as decorative moldings and furniture.

Architects sometimes have to juggle between two different types of 3D—CAD and general modelers. CAD programs provide not only a wealth of tools for creating every conceivable type of architectural structure, but also tools for turning the designs into plans and blueprints. CAD programs offer high-precision mathematical accuracy, feature databases for tracking materials requirements, and even offer programming languages that enable you to customize the program to perform difficult specialized tasks. Some CAD programs offer special modules of tools, including those for architecture, electrical, plumbing, mechanical, and industrial design.

General modelers, meanwhile, are intended for rapid prototyping and visualization. They are the type described in detail in this book because they are easier to learn, and have such a wide range of uses—such as animation and graphic design. The fact that both CAD programs and modelers have many interchangeable functions makes it hard to break down the distinctions. In general, you'll have to know what you want to use a program for before you make a purchase.

CAD users will often take advantage of 3-D renderers provided by other programs (many of which are integrated with general 3-D modelers). For example, MiniCAD+ users easily can render their designs in StrataVision 3d or StrataStudio Pro, while ArchiCAD has direct links to the renderer Atlantis. Very high-end architectural renderers, such as those working on large commercial developments, and particularly those doing animated fly-by renderings, choose Electric Image as their renderer because it easily handles very large and complex 3-D models, and its renderer is quite fast.

## **Virtual Reality and Walkthroughs**

Virtual reality is still something of a science fiction term, not only on the Mac, but in general. There are several companies that offer “VR” products, but they’re not, as the movies would have you believe, a total immersion in another world. What they are is a collection of programs that enable you to model buildings, scenery and other objects, and to interactively navigate through and around the scenes in real time.

Even as I write this, however, I know that by this time next year the possibilities in VR will be totally different than they are today. The first generation of VR software offered little more than flat-shaded, plain-colored, simple models and a less than stunning fly-through performance.

New versions of software, such as Virtus vr and Strata Virtual, offer texture-mapped models and reasonable performance. By the time the programs are optimized for Power Macs and refined, you’ll probably be able to fly through complex, highly realistic models, with a high degree of clarity and smoothness. Eventually, with the help of 3-D display accelerators (see chapter 13, “Advanced Topics,”) anything you can model and animate, you may be able to fly through in real time.

Virtual reality has amazing potential—even if it seems a little clunky today. While architects can now take clients on a tour of a proposed development; real-estate agents may one day walk potential buyers through a house located 3,000 miles away. Lawyers are probably already using the software to walk juries through accident scenes, but some day VR kiosks installed in tour buses will let tourists walk through homes of the rich and famous. If today’s scenarios seem a little mundane, imagine how that will change when VR contains “hot links” to databases or multimedia productions. Suddenly, walking through a scene may become very interesting indeed. Imagine walking into a virtual record store, clicking on a virtual compact disc, seeing a picture of the cover, hearing a sample of the music, carrying it to the cash register, and the next day getting it in the mail!

Virtual reality is included in this book, not so much because of what it is today, but because it leads to intriguing possibilities in the very near future.

## **What This Book is About**

It’s my hope that a newcomer to 3-D can use this book to gain a solid understanding of 3-D graphics on the Macintosh. It is not a user’s manual for any one software package, but a guide to understanding how 3-D graphics work in general, and specific

ways to put 3-D to work for you. There are step-by-step examples of useful techniques which, in many cases, you can adapt to the 3-D software package of your choice. Where techniques are unique to a particular software package, you will find that program identified. Many techniques are general, but each 3-D application has its own approach to doing similar things.

Again, in appendix A, "Macintosh 3-D Products," you will find a listing of the many 3-D software applications available, along with a list of what categories the program falls into for each of the 3-D disciplines (modeling, scene building, rendering, and animation). While it would be nice to include a complete listing of features for each product, to do so would be a disservice both to readers and the software vendors, as features are added and modified daily. Like all software markets, 3-D is highly competitive and it is likely that the vendors of the different programs will play catch-up in features, so that a generalization that doesn't hold today will tomorrow.

Where particular programs offer unusual features, credit is given where it's due. On the other hand, I also strive to avoid boring you with endless attribution. If a feature is the rule, rather than the exception (which is often the case), it is illustrated and described in general terms.

While 3-D tools are not necessarily hard to use once the concepts are clear, the creators of software are in such a mad dash to keep up with one another that all-too-important documentation is often an afterthought. It is all the manual writers can do to keep up with new and improved features. Unfortunately for the user, the theory and clear visual examples behind the tools are often left for a later revision. With this book, I intend to fill some of these gaps.

Furthermore, this book offers many techniques and tricks of the trade used to speed work along, to bring new levels of detail and realism to renderings, and to boost the impact of animations. This is by no means an exhaustive collection. Each program offers its own pitfalls and opportunities. But my hope is that you will glean a strong enough understanding of how things are done that you will know if and when it's time to move on to more powerful tools or turn to the vendor for help.

## **What Level of 3-D User Are You?**

Just as there's a difference between replacing your car's oil and your car's engine, there's a difference of skill level in producing 3-D imagery. Here, I've tried to convey some of the things to look for, depending on whether you consider yourself a beginning, intermediate, advanced, or expert user.

## **Beginner**

You sat drop-jawed through Jurassic Park. You need a 3-D logo for your Jumping Frog newsletter. You're determined to become the first 3-D whiz on the block. Whatever the motive, the catch: you've never touched a computer. While this book expects you to understand Macintosh basics, it will guide you through building a basic geometric shape like a sphere (a crude approximation of Jupiter), creating knockout logos and completed still images, and even generating otherworldly QuickTime and video animations. If you don't know where to start, then chapter 2, "Getting Started," is a good place to find out what hardware and software you'll need to make 3-D happen.

When I started working with 3-D graphics, I had no idea how many uses it had. Even today, I'm amazed to see how people are putting it to work. The new user, totally unfamiliar with 3-D tools or geometry, can read from cover to cover with the expectation of finding a useful 3-D overview, and hopefully, some surprises as well.

## **Intermediate**

If you have some computer or graphics experience, but are just entering the world of computer 3-D (or if you need a little brushing up on geometry lessons long forgotten) you will find the basic principles of 3-D described in chapter 5, "Modeling," chapter 6, "Scene Building," and chapter 10, "Rendering." If your understanding of illumination extends only as far as the wall switch or the flash in your Instamatic camera, then chapter 8, "Lighting," will guide you safely out of the dark.

In addition to describing what 3-D is, this book describes many ways you can put it to use, such as importing a rendered image into Photoshop using the alpha channel, wrapping text around a 3-D object in QuarkXPress, or building a 3-D control panel for an interactive CD-ROM.

The intermediate 3-D user can turn to relevant chapters in time of need. The different parts of 3-D are broken into their basic components and grouped accordingly. If you cannot make heads or tails of environment maps, for example, you can find an explanation under the heading "Environment Maps," in chapter 7, "Materials." Where topics become unruly and cross boundaries—a cylinder can be extruded or lathed, for example—the heading "Cylinder" discusses the duality, while the headings "Lathing" and "Extrusion," in chapter 5, "Modeling," explain how each can be accomplished.

## Advanced

If you're already using one or more 3-D applications, the pages within will serve as a supplement to sketchy software tutorials, an introduction to new techniques, and a source of ideas. Are you building intricate models, but your renderings come up flat? Perhaps it's time to take on reflection, bump, and environment mapping. You'd like to add motion blur to a speeding bullet, but you don't know how. Maybe you're wondering how to animate RenderMan shaders or make convincing neon tubes. This book aims to shed light on these problems and many more.

If multimedia is the message, the postman delivers twice: chapter 9, "Animation," and chapter 11, "Working with Images," cover 3-D in the new media age.

If you already are a regular user of Macintosh 3-D tools, or have used 3-D on other platforms but are investigating Macintosh, you will find the basic descriptions of techniques helpful as a refresher. Many of the tips throughout the book have been contributed by expert users and some of them may be valuable additions to your tool chest. For those of you who hope to further advance your capabilities, some of the advanced topics in chapter 13 may lend new insight and ideas.

## Expert

Expert users contributed many of the ideas in this book. I spent hundreds of hours talking to them in person and conversing in places like America Online's outstanding 3DSIG. I also picked the brains of 3-D software product managers and power users, and delved into help folders and tips archives. As much as possible, I've presented the results of that research here. There may be some tips and suggestions that haven't occurred to you. Or maybe there's a 3-D package you're considering buying, but you'd like a look at the interface before you buy (the demo software on the CD-ROM is invaluable in this regard). Maybe you just need the company's phone number. There are also a number of topics covered that you may not have attempted to tackle. Here are a few:

- Inverse kinematics and physical modeling
- 3-D sound
- Solid modeling
- Splines versus DXF
- Visible lights
- Video service bureaus

- Distributed rendering and YARC boards
- 3-D QuickDraw

I readily admit that there are many expert users of 3-D who, given the time and the patience for book writing, will eventually provide more experienced, minutely-detailed guides to the individual products involved in 3-D. So what's in the *3-D Starter Kit for Macintosh* for you? In addition to being a suitable doorstop, and a primer for new employees in the graphics department, this book has benefited from several years of weekly contact with 3-D vendors and their products. Appendix A provides a thorough account, from a single source, of the capabilities of every 3-D tool currently available to the Macintosh user, as well as where to find it.

## The CD-ROM

The best reason to use the CD-ROM is that a great many of the 3-D software programs described in this book are available in demo versions on the CD. That means you can try the program out, and in some cases, follow along with the book's examples. This is far more valuable than any magazine review or product description you will find in these pages. Even demo versions of some of the 2-D programs, such as Premiere and Photoshop, are also included.

The CD-ROM also realizes another great opportunity. Color is the natural media of 3-D, but color printing all of the hundreds of illustrations included in the examples here would have driven the cost of the book into the fourth dimension. Many of the best examples have been reproduced on the disc, so you can see the images in their intended form. This also applies to sample output from many of the 3-D programs. You can get a first-hand look at rendering quality, as well as some of the ways other users have put the software to work.

Animation, a caged tiger on the printed page, is set free through the amazing medium of QuickTime. It's much easier to understand the importance of bevels in animation, when you can see them in action.

For tooling around with the software demos, there are some unique 3-D models in DXF and RIB and program-specific formats, and a collection of matching backgrounds, seamless textures, and bump maps.

The CD-ROM also contains the customized software to dial into to the author's own, graphical electronic bulletin board. This Alpha Channel BBS is not intended to compete with the excellent online forums found on America Online and CompuServe (which were the source of so many of the ideas contained herein).

Instead, I hope it will be a place where people with burning questions, criticisms or curiosity can languish in my study. (See appendix B for more information about logging on and what to expect when you get there.) This is where readers of the *3-D Starter Kit for Macintosh* will have a substantial say about what goes into the *next* edition, and there are guaranteed to be a few pleasant surprises there, too.

Directions for using the disc and a complete guide to its contents, are located in appendix B, “3-D Starter Kit for Macintosh CD-ROM.” (There’s that lead balloon again—with a herniated “disc”.)



Look for the CD-ROM icon—it highlights material that is on the disc.



**c h a p t e r**

**2**

# **Getting Started**

**M**acintosh 3-D graphics has undergone a revolution unlike any seen in Mac graphics since the emergence of desktop publishing. From a handful of expensive CAD-oriented products a few years ago, it now boasts a wide-range of 3-D tools for every graphical purpose. Much of this explosion has to do with the rapid acceleration of Macintosh hardware, from the 68000 to 68040 Motorola-based Macs, to the newest, fastest Power Macs using the PowerPC chip developed by an unlikely Apple-IBM-Motorola alliance.

The software tools now range from industrial design and broadcast video to animation programs targeting multimedia developers and programs aimed at producing new kinds of art for plain, old paper. Newer, faster, cheaper hardware has placed more powerful tools in the hands of more and more users. The software companies have responded by writing new versions with an ever-growing list of great features. For example, when it was first introduced, Pixar's Typestry was mainly intended for rendering 3-D logos. Now it features event-based animation with particle effects. Similarly, VIDI's Presenter Professional has gone from being a good spline-based modeler, to a spline-based modeler with Boolean surface trims and advanced animation effects, such as 3-D sound. The list of software that has gone through these kinds of amazing transformations is yards long.

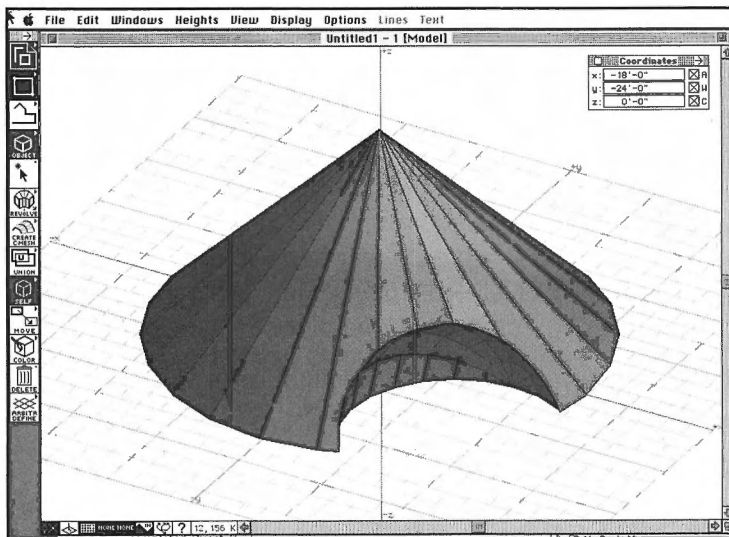
## How 3-D Works

Three-D graphics are created in stages: modeling, scene building, animating, and rendering. Many 3-D applications do more than one of the above; some do it all. Some programs combine these elements. For example, Sketch! and Sculpt enable you to model in the same space used for scene building. On the other hand, some 3-D applications take on only one of these four tasks at a time; you are required to pass a model back and forth between the different modules. Other applications are highly specialized and fill only one or two parts of the puzzle. For example, Pixar's ShowPlace is only a scene builder (with a bit of modeling); while VIDI's Modeler Professional and Byte by Byte's Sculptor specialize exclusively in model building. Because 3-D is so memory- and processor-intensive, almost all 3-D programs have a separate mode for rendering. For a complete listing of which programs offer which features, see appendix A, "Macintosh 3-D Products."

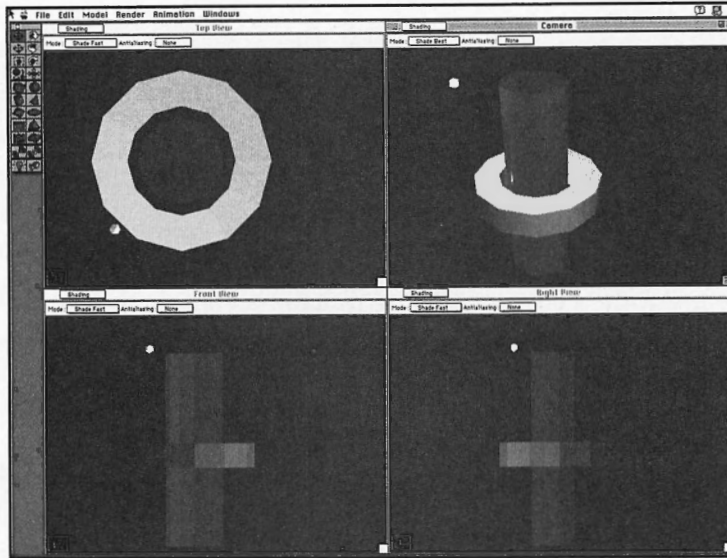
## Modeling

Modeling is the building of 3-D objects in space. Think of it as using a computer to sculpt clay. The resulting model contains 3-D *geometry* that has parts defined by height, width, and depth. Often these shapes are made up of simple polygons, such as triangles and squares. Alternatively, the geometry may be defined by *splines*, smooth three-dimensional curves that resemble the whalebone ribs in an old-fashioned corset.

You can accomplish the task of modeling while in a single-perspective view (where drawing tools “snap” to points in space), or in a multi-window mode (where each window shows the same scene from different points of view—typically top, perspective, front, and side—see figures 2.1 and 2.2). With single-perspective views, you’ll often switch between different views to see what’s happening in three dimensions. In four-plane views, changes are reflected in every window, but you have to adapt your brain to looking at the same thing from four angles simultaneously.



**Figure 2.1** *form•Z's single-perspective view closely resembles what you see in the real world; to draw in three dimensions, you have to rely on snaps and switching between views; Alias Sketch! also uses this type of view*



**Figure 2.2** *Infini-D's four-plane view; the four windows show the same scene from (counter-clockwise from upper-left) top, front, side, and perspective points of view—changes in one window are reflected in all four*

Three-D shapes can be created in a variety of ways, many of which you can combine (depending on the program) to form a finished model.

- **Additive modeling** involves adding pre-existing building blocks together to complete an object. This is a typical way to make use of models provided as libraries on floppy disk or CD-ROM. Even some programs that are not considered “modelers” per se, can do this kind of assembling of pre-built parts.
- **Constructive modeling** incorporates a number of techniques (such as extrusion, lathing, sweeping, and lofting) to construct a 3-D shape out of 2-D outlines.
- **Molding**, also called *distortion* or *deformation*, enables you to squeeze, pinch, and distort shapes into new forms. This is particularly easy to do in spline-based modelers, because the curves that make up a shape are usually easy to distort.
- **Derivative modeling** is a technique used by a growing number of programs, including form•Z, Presenter Professional, Sculpt, ZOOM, and PixelPutty Solo. This enables you to use existing shapes to “carve” other shapes. This capability is sometimes called *Boolean modeling* or *surface trimming* (depending on whether you’re using a solid or surface modeler, respectively). For

example, derivative modeling enables you do things like drill a hole in a sphere with a cone, extract a shape from the intersection of two cylinders, or knock windows out of solid walls.

- **Three-D digitizing** uses a digitizing table and pen. As you run the pen over the surface of a real object, such as a clay sculpture, the Mac recreates the object's shape in digital form. Geometry created in this way is compatible with the other types of geometry mentioned here. At the very high end of things, there are also acoustic and laser scanners, but they're beyond the scope of this book.

## **Scene Building**

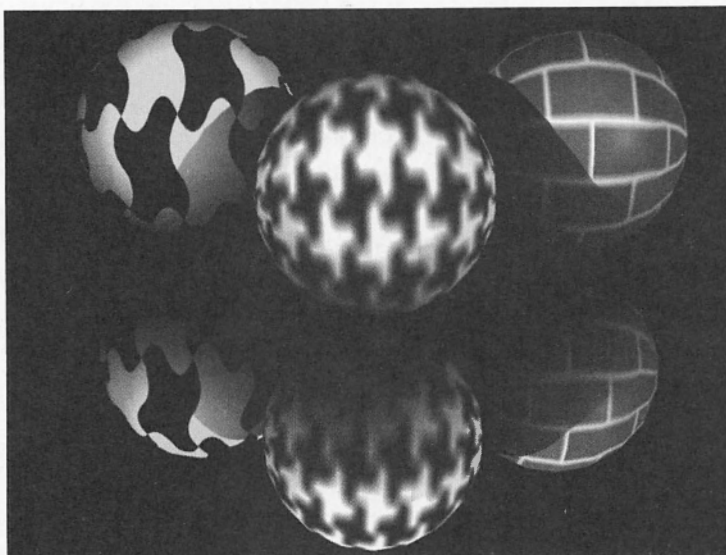
*Scene building* is the arrangement of models, textures, lights, and cameras. To some extent, scene building happens when you're building a model, in that parts are given relationships and positions relative to one another. Some programs, such as Alias' Sketch!, Shade III, Sculpt 3D, and Adobe's Dimensions make no distinction between modeling and scene building; they're completely integrated. Many programs, however, divide the tasks of modeling and scene building. VIDI's presenter Pro, for example, handles modeling in one module, but sends you to the "Presenter" module for texture mapping, the placement of lights (and microphones!), and the manipulations related to animation.

Electric Image Animation System does only scene building and animation. Models have to be built in another program. Pixar's Showplace, similarly, lacks real modeling tools (although it has a few simple modeling plug-ins). It's used to assign "Looks" shaders to objects and serve as an interface to Pixar's RenderMan rendering system.

Some scene builders enable you to constrain parts so that they will move in a particular relationship to other parts. For example, a wheel can only turn on its axle and a piston can only move within the confines of its cylinder. Swivel 3D was the first program to offer these types of joints or "constraints." Shade III, Electric Image Animation System, MacroModel, and Infini-D are other examples of programs with these kinds of tools.

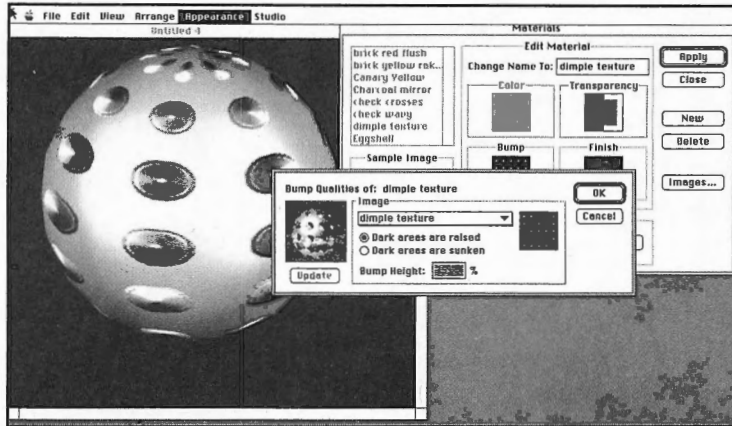
In addition to moving and placing objects, scene builders offer many of the other tools that help you create interesting renderings:

- **Texture mapping.** By attaching or mapping textures, a plain gray orb can be turned into a glowing halogen bubble, a fuzzy orange tennis ball, or a crystal oracle (see figure 2.3). Textures can be as simple as red plastic, or they can be as complex as the reflective riveted hull of an aluminum aircraft.

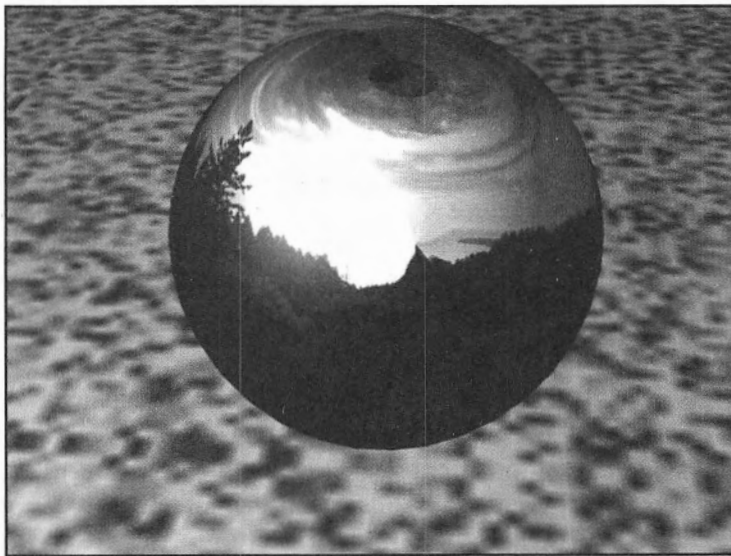


**Figure 2.3** The same sphere duplicated and rendered with several different texture maps; the reflections are provided by ray traced rendering

- **Bump maps** are a bridge between modeling and textures. Dark and light areas in a bump map are rendered as depressions and raised surfaces (see figure 2.4).
- **Label maps** are a type of texture mapping that enable you, for example, to place a wine label on a glass bottle or map a Persian rug design onto an interior floor. Because labels can have transparent parts, you can “map” them over other objects and have part of the underlying surfaces show through.
- **Environmental reflection maps**, more commonly called “environment” maps, give chrome and other reflective surfaces the appearance of reflecting the surrounding environment. While renderings done with ray tracing actually reflect surrounding objects, other rendering types, such as Phong shading, can only simulate reflections of the surroundings. This makes environment mapping very important (see figure 2.5).
- **Lighting.** Just as *lighting* sets the mood of a stage play or photograph, the placement of lights profoundly affects the appearance of your final image. Depending on the program being used, types of lights include point (such as light bulbs), distant (suns), spotlights, neon tubes, and lasers to name a few.



**Figure 2.4** This model uses a simple grid of soft-edged dots as a bump map



**Figure 2.5** A landscape photograph used as an environmental reflection map

Most scene builders have multiple lights of varying intensity and color. Light sources themselves are usually invisible, but Electric Image Animation System offers lights that glow and even have an aura or halo. Infini-D will put visible light bulbs in a scene (see figure 2.6).





**Figure 2.6** This image was rendered using Electric Image Animation System's unique visible lights

- **Cameras.** The *camera* is an object that represents your point of view. By moving or rotating the camera, you can change what appears in the final rendered image. Multiple cameras offer multiple points of view. Just like a real camera, you can change lenses for different angles of view and—using Sketch! or programs that render with RenderMan—adjust the apertures to change the depth of field.
- **Atmosphere and other environmental considerations.** In most scene builders, you can place scanned or painted backgrounds. Some programs enable you to employ atmospheric effects such as fog or haze that create the impression of depth by gradually obscuring distant objects.
- **Rendering interface.** *Renderers* are software that use a series of mathematical equations (combined with your models, textures, and the composition of your scenes) to create a realistic image. You set the user-configurable rendering parameters, such as image resolution, antialiasing, and levels of refraction, and the renderer gives you back a finished picture. There are many elements and variations of rendering, described later in this book.

## Animation

Animation is really an extension of scene building. *Animation* adds to 3-D the element of time by adding motion to objects, lights, cameras, and almost any other part of your scene.



While simple in concept, movie makers have used the technique called key framing to follow speeding spaceships through alien condominium projects. You first specify the position of an object, then create a key frame. You then move the object, and set a second key frame. The software then calculates all of the frames in between (“tweens”) the beginning and end key frames, so that the model flies smoothly from the start position to the end position.

Key framing can control much more than an object’s motion: textures, lights, and even an object’s shape can all be animated in this way. Cameras, which control your point of view, can also be animated with key framing. In some programs, objects can even be made to “morph” into one another. (Imagine glass letters magically rising out of pools of mercury, or a Fiat 500 transforming into a Cadillac El Dorado.)

It’s possible to animate surface properties such as texture and glow, and recent additions to the Mac enable you to add explosions and particle effects like sparks and rain. Playmation and Electric Image Animation System are the first programs on the Mac to enable you to animate stretch, squash, twist, bend, and other flexible properties of objects. Meanwhile, VIDI’s Presenter Professional may be the first program available on any personal computer that enables you to animate sounds. You attach a sound to a moving object, and as it flies by microphones you’ve placed in the scene, the microphones pick up the sound and add it to your animation’s sound track.

## Rendering

Rendering is the final step in creating 3-D graphics. *Rendering* is where the computer calculates the appearance of the scene, including highlights and shadows, reflections and refractions, textures, labels, and bumpiness—even special effects such as displacement, fog, and motion blur. If creating an animation, the computer does this once for every frame. The renderer can usually be set to save the resulting images in a variety of resolutions and file formats, ranging from high-resolution PICT files with alpha channel transparencies, to QuickTime movies for use in multimedia projects, or numbered PICT files for output to videotape.

Mathematically, rendering is by far the most complicated part of creating a scene. A recent 11-by-14-inch, 288 dpi ray tracing I did on my Power Mac took seven hours and required over 77 million trigonometric calculations (44 million of which were reflections). I’m glad the computer was counting instead of me!

All of these major rendering techniques are described in detail in chapter 11, "Rendering." There are many different types of rendering, including:

- **PostScript blending.** This technique generates a smoothly shaded illustration that is compatible with any image created in Adobe Illustrator or Aldus FreeHand. As with drawings created in those programs, the resulting PostScript art can be printed at any resolution, on any PostScript printing device, such as a laser printer.
- **Flat shading** gives each face of a 3-D object a single color, so objects appear solid, but faceted like a gemstone.
- **Gouraud shades** objects smoothly by blending (and blurring) adjacent surfaces together.
- **Phong shades** smoothly, but also accurately depicts surfaces, textures, and reflective properties (such as specularities) and can simulate mirror reflections.
- **Ray tracing**, a complex rendering method, can accurately render mirror effects, transparency, and refraction (bending of light through clear surfaces). It is the most realistic of the common rendering types.
- **Pixar's MacRenderMan** is a rendering engine that uses Phong shading (most of the time), although it can be used to do ray tracing. RenderMan also can provide many unusual effects, such as displacement and Booleans.
- **Radiosity** is a rendering algorithm that vastly increases the amount of rays calculated in a scene compared to ray tracing. The system calculates all of the light in a given space, such as a room, so that effects like the color of an object reflecting onto another object, the fill lighting provided by a light-colored wall, or the luminance of a translucent object with light shined through it, are accurately depicted. "Raydiosity" is Strata's version of this method; it's too slow to be used except as an experiment.

## Hardware

The Macintosh model you need to work in 3-D depends on the complexity of the images you plan to create and how fast you need to do it. If you only plan to create the occasional 3-D PostScript rendering for use in an illustration program, any standard business-level Macintosh is well-suited to the task. On the other hand, if you plan to do serious 3-D work (especially photorealistic rendering or animation), the fastest Mac money can buy will be worth the money spent. These days, that means a Power Macintosh.

## **Power Macintosh**

The introduction of the Power Macintosh in 1994 dramatically changed the landscape for 3-D graphics. Suddenly a machine with state-of-the art computing performance, and *the* state-of-the-art operating system, was placed in the hands of thousands of ordinary computer users, many of whom were not quite sure how to tap into all this new-found, simply-packaged processing horsepower. One group that had no problem knowing what to do with it, however, was the community of 3-D graphics software developers.

Specular's Infini-D was one of the first of a handful of software programs that shipped "native" on the day of Apple's rollout. ("Native" is shorthand for running under the Power Mac's standard PowerPC operating mode, instead of in "emulation," which mimics the slower 680X0 chips.) Not long after that, a whole wave of native 3-D software washed over the Mac graphics community.

Many 3-D users were first in line to buy the new Power Macs and they clamored loudly (and are still clamoring as I write this) for versions of their applications running in native mode.

So why all the commotion? Many new Macs had been delivered prior to the big rollout of Power Mac, and few people had worked themselves into such a near-religious frenzy.

With earlier Macs, using 3-D software often involved a painful amount of waiting. To understand what it was like, go back to what Shawn Hopwood, former president of VIDI, said to me long before the Power Mac was a computer in somebody's hands, "Imagine word processing and waiting 15 minutes for the cursor to return after you type a paragraph. That doesn't just affect the way you work, it affects the way you think."

The commotion at the rollout was due to the fact that the Power Mac, even in its very first generation, is as much as ten times faster than the fastest '040-based Quadra, the 840AV. For Mac 3-D artists, the issue wasn't word processing, it was rendering, and often it was a question of hours, not minutes, waiting for a single finished image. With the Power Mac, a rendering that used to take an hour can now be done in ten minutes.

But rendering was not the only task that benefited from this breakthrough in performance. The whole system is five to ten times faster. This means calculating wireframe previews and displaying them onscreen as you move a model around, as well as building and applying texture maps and shaded previews, are all much smoother and faster.

When creating a scene, the need for instant feedback is critical. If you are forced to wait for several minutes to see what you're making, the creative muse will soon become a frustrated, irritable ogre. Three-D graphics place such high demands on your computer system that every action may result in several seconds of redraw onscreen. This delay between what you do and what you see can be frustrating. If you draw a shape, you need to see it now. When you apply a texture, you want to see the result immediately.

To some extent, software developers have responded to this need by writing faster and faster software and creating interface tricks (such as pre-built previews of textures) to enhance visualization. Often, you can see real-time updates of objects and scenes as you draw. But this is far from real-time rendering, where you see a completely finished image appear as fast as you draw. Most programs force you to accept boundary boxes and wireframe views as a compromise between no feedback and real-time realism.

The Power Mac hasn't completely solved this problem, but it's a big step in the right direction. While it's possible to do amazing 3-D graphics without the help of a Power Mac, to Apple's credit, the new machines are certainly the best thing to happen to the Mac 3-D market.

## **Accelerators**

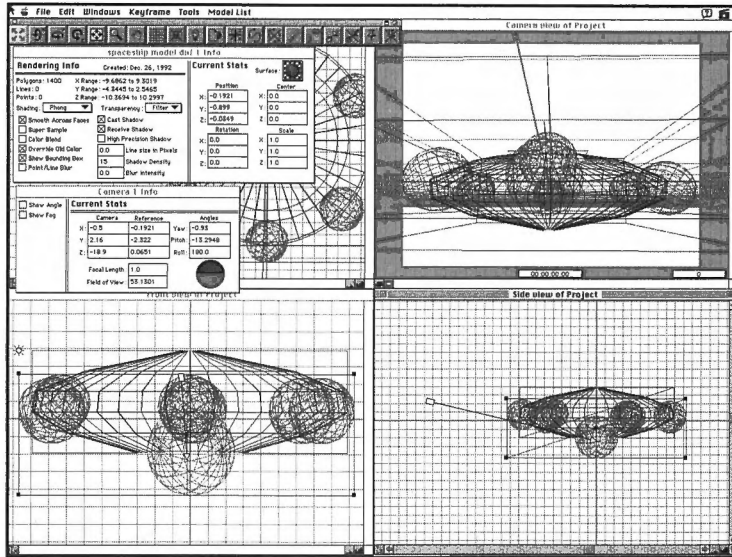
If you already own an older Mac, such as a Quadra or Centris, it is possible to accelerate your work with a variety of additional hardware and software. Accelerators that speed up your Macintosh (such as the PowerPro 601 from Daystar) can turn a tired Macintosh into a full-speed 3-D workhorse. At press time, PowerPC accelerators were only available for Mac Quadas, but by late 1994, they are projected to be widely available for most Macs.

Another option includes the '040 accelerators, made by a number of companies, such as Daystar and Mobius. Though nowhere near as fast as PowerPC-based Macs, for many users, particularly those looking to preserve their existing Mac investment, they represent a good value. Using distributed rendering (see chapter 13, "Advanced Topics"), you can get PowerPC performance by combining the resources of several Macs on a network.

Specialized accelerators, such as YARC Systems' MacRageous and Zuma boards, can accelerate Macintosh rendering to greater-than-Power Mac speeds. On the other hand, these don't accelerate all of the system functions as the PowerPC upgrade cards do, and to get any benefit out of the YARC boards, software has to be specially written to support it. Only a few products currently are. YARC's boards also are explained in more detail in chapter 13, "Advanced Topics."

## Monitor

A large color monitor, 16 to 21 inches, is practically a requirement for working in 3-D. If you plan to use a software product that displays four working windows simultaneously, one solution is to use two monitors: one that displays menus, toolbars, and positioning windows; and another that displays the preview window. Figure 2.7 clearly illustrates the need for plenty of screen space.



**Figure 2.7** A 20-inch screen (1152 by 870 resolution) crowded with the many windows used simultaneously in the Electric Image Animation System

Color is extremely helpful—almost mandatory—even when creating grayscale artwork. When modeling, it is sometimes difficult to distinguish the different layers of wireframe models, as well as which model parts are active and which are not. A color display enables you to display inactive parts in black, for example; while active objects can be displayed in red or some other equally visible color.

Dimensions and addDepth (3-D PostScript illustration programs) have simple interfaces that are not as dependent on a large color monitor. You even can get away with working on a PowerBook with these programs. Alias Sketch! is another program that does away with the dependence on having four or more windows open at one time.

## Display Cards

In programs where you are working with many different layers or groups, you often can assign different colors to different layers to further distinguish objects of interest. If you decide to use multiple monitors, you'll need at least one display card to supplement your Mac's internal video. Some of the new Power Macs even have support for two monitors built in.

While many newer Macintosh computers drive most monitors (some even in thousands or millions of colors), older Macintoshes require a separate display card if you want to use a large monitor or display photorealistic 16-bit or 24-bit color. This book was illustrated with the help of a SuperMac Thunder/24 board, which provides accelerated video and therefore makes modeling much snappier. Accelerated video cards also are available from RasterOps, Radius, Sigma, Mirror, and Envivio. Acceleration is particularly important when working on large, complex models—redrawing wireframe models onscreen without one can be painfully slow.

## Hard Disks

Three-D applications themselves do not require huge amounts of storage. The serious 3-D artist, however, may use many textures, geometry libraries, reflection maps, and so forth. Animators will require massive amounts of hard disk space to accommodate digital video and other animation formats. Three-D images, for the most part, are 32-bit bitmapped graphics: a single 640 by 480 (NTSC resolution) image occupies about 1.2 MB of disk space. Multiply this by the number of frames required for 30 seconds of video, and you will find you need about 1 GB (gigabyte) of disk space.

Although compression techniques help, a fast, high-capacity drive will make it much easier to complete your work. Drives featuring SCSI-2 and other fast technologies will significantly increase performance. On the other hand, drive capacity is far more important than speed, unless you're working in the world of digital video.

## Long-Term Storage

It's possible to develop massive libraries of textures, reflection maps, bump maps, models, animations, and images. In order to maintain these as libraries, you may want to invest in a large-capacity data storage system, such as a magneto-optical

(MO) drive. These drives currently hold anywhere from 128 MB to 1.3 GB of data on a cartridge that costs anywhere from \$20-\$120. This makes per-MB cost of media about a fifth of that of hard disks. Large format MO drives also are an ideal media for transferring animations to service bureaus (if the service bureau also has a compatible drive).

Another option for long-term storage is Digital Audio Tape (DAT). While DAT tape backup works for protecting data from loss, it doesn't provide very convenient access to images and objects when you need them. DAT drives typically store the contents of a hard disk as a backup archive. In order to regain access to files, you usually need to "restore" the entire hard disk (or at least read the entire tape, since everything is stored in a linear fashion). A remarkable program called DeskTape, from Optima Technology Inc., enables you to use a DAT or Exabyte tape drive as a mountable volume on the Macintosh desktop, giving you 2 GB, 5 GB, or even 10 GB of data storage on a tape that costs about \$10. Combined with image cataloging software, such as Canto Software's Cumulus, this makes for a tremendously effective and cheap storage solution. The drives themselves are about \$1,000, which is currently the going price of a 1.5 GB hard disk.

Floppy disks continue to be a viable media for storing single 3-D model files that are fairly compact. However, you will usually store the associated texture and environment maps with your model, which will often put floppies out of the running. Floptical drives, which store about 20 MB of data, are a relatively low-cost compromise, although 128 MB MO drives threaten the long-term future of flopticals.

Animators must plan their choice of removable media. Given that every finished frame of a video animation can require as much as a megabyte of storage space and that animations involving many layers of effects will require much more, this application definitely calls for a higher-capacity transfer mechanism. There are some creative solutions available for 3-D animators. For example, ElectricImage Animation System enables you to store animations as Abekas-format backup files on an Exabyte 8 mm DAT drive. These can be "restored" directly onto an Abekas video disk system at a service bureau.

Any animation program can now take advantage of this Exabyte-to-Abekas technology. You can copy any animation directly from your Macintosh to an Exabyte drive in Abekas format with new software from ASDG or Knoll Software.

## Removable Hard Disks

Assuming you plan to transport your images to service bureaus or elsewhere, a removable storage drive (such as Bernoulli or Syquest) can greatly facilitate moving files. In general, your choice of removable media will depend on the type of images you plan to produce, as well as what drives your favorite service bureaus have to offer. The 44 MB- and 88 MB-Syquest cartridges have been the standard mode of transport of large files for an amazingly long time for the computer business, but these drives are now on the way out. Cartridge drives from Syquest and Iomega are available in sizes ranging from 105 MB to 250 MB. If you only plan to take the occasional high-resolution still image to a service bureau for output to a film recorder or dye sublimation printer, these drives are up to the task and they can be found everywhere, making them likely to be part of the print world for a long time.

## Scanner

One of the most useful tools you can have for 3-D work is a scanner that brings photographic images into the computer. Anyone serious about doing 3-D will eventually want to tap the real world for some of its infinite textures. For example, architects will find the capability to photograph a site, produce a model, and render the proposed structure against a backdrop of its intended surroundings invaluable.

Slide and film scanners (such as the Nikon CoolScan, Microtek ScanMaker 35, and Santos Mira•35), enable you to scan negatives or slides directly into the computer. Aside from digital photography—which currently is grossly expensive or of fairly low quality—this is the fastest way to capture high-quality images and get them into your Macintosh. You can shoot a roll of film (photos of textures, for example), take them to the local 1-hour photo finisher, and scan them in for almost instant turnaround.

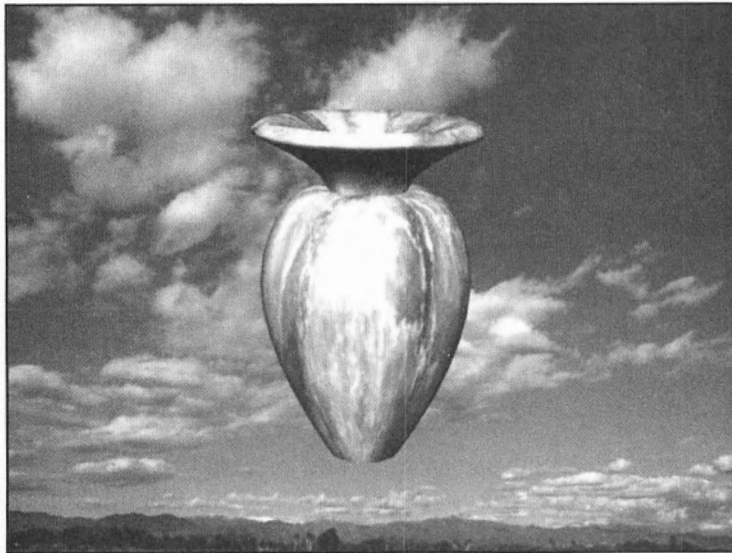
Comparatively inexpensive, flatbed scanners are readily available for scanning the prints that you get back from a photo finisher. Flatbed scanners are too numerous to mention by name, but for best results with 3-D software seek a scanner that offers 24-bit color. If you do not own a scanner already, look for a scanner that comes bundled with the regular edition of Adobe Photoshop. This software is a “must-have” for any serious 3-D user. The “Lite” version is very functional, but not as powerful in the long-run as the professional version.



Hand-held scanners are an inexpensive solution and can be useful when scanning textures you plan to use as texture “tiles.” But in general, they do not serve well as a professional graphics solution because you cannot scan large images reliably. On the other hand, a color hand-held scanner combined with a full-featured 3-D program (such as RayDream’s Designer) would comprise a very flexible system for the first-time 3-D user.

## CD-ROM Drive

There are dozens of CD-ROM titles available that ease the creation of interesting 3-D art, including the disc that comes with this book. Items that can be found on CD-ROM for the 3-D user include: print-quality photographic textures, seamless textures, video and animation loops for rotoscoping, 3-D models, and 3-D games, to name a few. While a CD-ROM drive is by no means a requirement to work in 3-D, it opens up a wealth of creative opportunities (see figure 2.8).



**Figure 2.8** The backdrop in this image is from the *Sky Volume One* collection from *CD Folios*; the marble texture on the vase is from *ArtBeats’ Marble and Granite, Volume One*

A number of vendors now provide sample 3-D models, tutorials, animations, and images—even software—on CD-ROM.

Several companies offer CD-ROM libraries of seamless photographic and synthetic textures for use in 3-D. Examples of these include the Wraptures series from Form and Function, Pixar's 128 and the Artists in Residence series from Xaos Tools, and the Marble and Granite series from ArtBeats.

The latest CD-ROM drives have "double-speed" or even "4-speed" capability and they are two- to four-times faster than older CD-ROM drives. Look for these if you're buying a new drive, because older CD-ROM drives are notoriously slow.

## Photo CD

One of the most compelling CD-ROM formats for 3-D users is Eastman-Kodak's Photo CD. You can store your own photos of textures and backgrounds permanently in electronic form on CD-ROM, without using a scanner. This is an excellent way to develop a library of textures and other images. You can put hundreds of images on a CD for the price of the cheapest flatbed scanner, and the images are usually of very high quality, depending on your originals. Many newer CD-ROM drives that read standard CD-ROMs also will read Photo CD format. The other advantage of Photo CD is that it frees you from the burden of scanning textures. You can simply shoot photos and drop the film off or mail it to a photo finisher and get the disc and negatives a few days later.

### Tip

The CD-ROM ToolKit from FWB Software Inc. accelerates existing CD-ROM drives through sophisticated caching and makes most CD-ROM drives—even old ones—Photo CD-compatible.



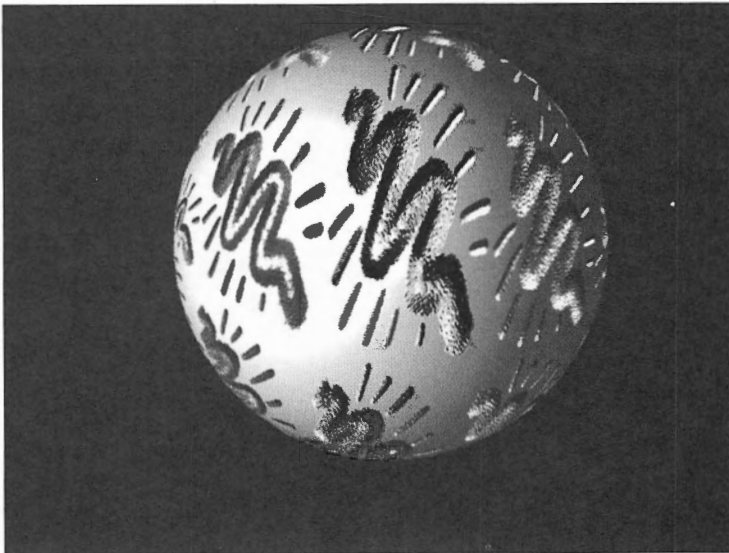
A sample of matched background and material textures from the Alpha Channel Expert Collection can be found on the included CD-ROM. Clip models, tutorials, an image gallery, and many sample animations are also on the disc.

## Drawing Tablet

Drawing or digitizing tablets are standard in the architectural and engineering CAD worlds, where handmade drawings are converted to highly-accurate vector-based artwork. These combine a tablet (anywhere from 6 inches by 9 inches to 3-feet square) with a "pen" or a "puck" (a mouse with crosshairs—which mimics a mouse in functionality). The difference is that a drawing tablet can use absolute

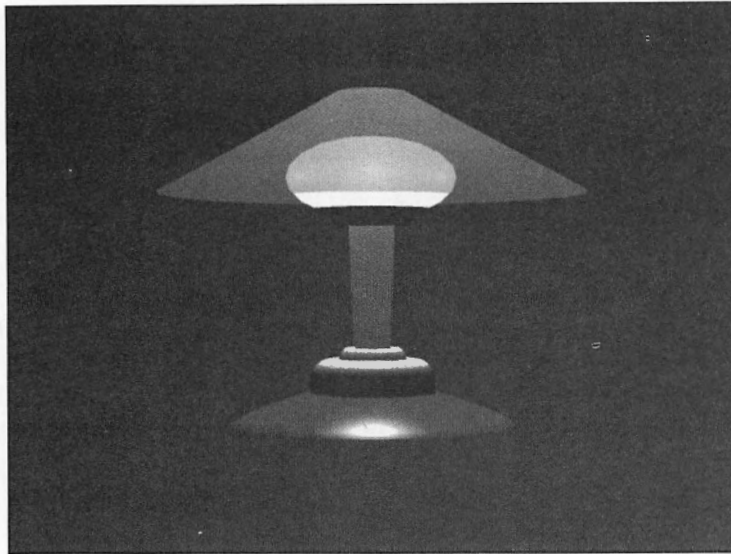
coordinates, unlike a mouse's relative position. This means that if your pen is one inch over and down from the upper left-hand corner of the tablet, your drawing cursor will be one inch over and down in your drawing. You can tape an existing drawing over the tablet and trace it with your pen to input a highly accurate copy into your 3-D application.

Wacom Inc.'s tablets have completely changed the way computer artists interact with the Macintosh. The Wacom tablets (I favor the ArtZ, which is very small and light) feature a pressure-sensitive pen that responds to human input very much like traditional brushes and paints. As you press harder, the brush stroke widens; press more lightly and it narrows. This only works with software that supports the tablets, but most paint and draw packages now do. The result of using the tablets is that you can create an infinite variety of painted, drawn, inked, and smeared textures that can bring a new level of artistic detail to 3-D images. In figure 2.9, the texture used to create the image was "painted" with a Wacom tablet in Fractal Design's Painter software—then used as a bump map in Alias' Sketch!.

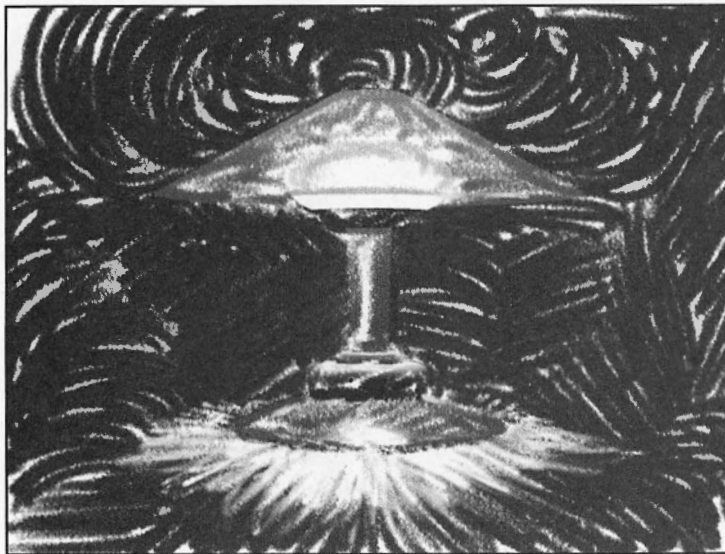


**Figure 2.9** *Texture created with Painter and a Wacom tablet*

When combined with a program like Painter, the Wacom tablet enables you to trace over a 3-D rendering using colors from the original to create an image that is accurate in terms of shadow and perspective—yet looks for all the world like it is hand-drawn or painted (see figures 2.10 and 2.11).



**Figure 2.10** A lamp modeled and rendered in 3-D



**Figure 2.11** The same lamp “cloned” in Fractal’s Painter with a Wacom pen and tablet

New tools introduced in Painter 3.0 enable you to rotoscope or paint over cells of an animation. This means you can create a 3-D animation, then very convincingly make it look like it was painted by hand.

CalComp Inc., Kurta Inc., and Summa Graphics Inc. also make pressure-sensitive tablets that work much like the Wacom tablet.

### **3-D Digitizer**

A 3-D digitizer is very much like a 2-D drawing tablet, except that a stylus is moved over the surface of the object being digitized and the tablet reports the stylus' position in three dimensions.

Many professional 3-D animators use clay models of objects; digitize the models using one of these devices; and then use the resulting computer models in their animations. Since it is sometimes easier to create a life-like organic model (such as an animal) out of clay than with 3-D software, this can be an effective technique.

Mira Imaging makes a variety of tablets that can digitize large objects to very fine levels of detail. They are all based on the same "HyperSpace" software and range in price up to \$15,000.

Immersion Systems offers a Personal Digitizer System. For about \$2,000, you get the digitizer and Mira's HyperSpace software. The unit has a stylus attached to a mechanical robot-style arm which you digitize to move around the model. Models up to 17" on a side can be digitized using this device and it is accurate to within a millimeter. As you move the stylus over the surface of the model, its position in space is saved in a standard three-coordinate format such as DXF. You can then import the file into a 3-D program and use it like a regular 3-D model.

## **Video Animation Hardware**

There are many ways to get animations on videotape. The most popular among professionals is frame-by-frame recording to an industrial animation deck, but service bureau output provides an attractive option for occasional output, and digital video—recording straight from the Mac's hard disk to tape—is now a viable method of getting animations on tape, though the recording quality is still not as good as with other methods. Animators who are determined to record their animations in-house will need to invest in a substantial amount of hardware in addition to the regular Mac system.

## Animation Video Deck

If you're going to "roll your own," you will need a frame-accurate video animation deck. Fortunately, the price of these decks is plummeting as with most electronics. It's now possible to buy a Panasonic GVR-S950 for about \$3,000. Professional-level component Beta decks, meanwhile, start at well over \$10,000, so this is not an investment to take lightly. This requirement alone is the reason most animators turn to service bureaus for final output.

An animation deck is capable of stepping, one frame at a time, through an animation and recording each frame as it goes. This seems like a simple process on the surface. But in reality, to play or record video, the tape must be moving at full speed. However, the computer cannot display the images of an animation at full speed without a high degree of compression, which reduces image quality. To solve this dilemma, the animation deck must do what is known as *pre-roll*, or back up and move forward for each successive frame of video. In an animation system, the Mac controls the animation deck, so it is not necessary for the user to manually operate the deck to record each frame.

The broadcast-quality video decks, such as three-quarter-inch component devices and Beta decks, are prohibitively expensive, but they are mandatory for broadcast video work. For previewing animations and less demanding work, however, Sony's hi-8 animation deck, the EVO-9650, and Panasonic's SVHS animation decks, produce excellent results.

## Animation Controller

Animation controllers (such as the DQ-Animaq NuBus board or the DQ-MAC232 software from Diaquest Inc.) enable your Macintosh to control a compatible animation video deck and automatically output completed animations to videotape. Essentially, the Diaquest system works as a traffic controller, sequentially stepping through and displaying the many frames of an animation, all the while telling the video deck when to move on and record the next frame.

Diaquest's DQ-MAC232 software-only animation product performs these functions, but without some of the other benefits of the NuBus boards, such as the capability to do "Fast Pass" video capture from professional animation decks. For users of low-end animation hardware, however, this product is well suited to the task.

MacAnimator Pro does essentially the same thing as DQ-MAC232.

## Digital Video Hardware

Digital Video hardware has come into its own. Several systems are available for the Macintosh that enable you to capture full-motion, full-frame video in real time, without the need to go through the often tedious task of capturing frames one-by-one. Not only can you digitize video using the VideoVision Studio, for example, but you can play back full-screen, full-motion video directly to videotape. Included in the list of things you can put on tape this way is 3-D animation. It's still difficult to get perfect video quality with digital video systems, but by the time you read this, it will be possible. Systems like Radius Telecast can lay video directly out to broadcast component video decks, and with a genlocking video buffer you can record 3-D animations directly over live video.

The digital alternative to real-time video grabbing and playback is to use a video "frame grabber." These specialize in capturing and recording one frame at a time from a video source. If used with a frame-accurate video deck and an animation controller, these boards offer reliably high image quality. The drawback to this type of system is the need for the animation controller and an animation video deck, two very expensive items.

Digital video goes both ways: as a method of getting animations from your computer to videotape, but also as a way of getting video into your animations. Video can be used as backdrops for 3-D animations, producing startling movie effects (often known as "rotoscoping"). Some programs such as Specular International Inc.'s Infini-D, ElectricImage Animation System, and StrataStudio Pro enable you to use video or animations as moving textures. For example, you can project a "movie" of a fish swimming inside a 3-D fish bowl.

## Video Encoder

In order to output to videotape, the RGB images on your Macintosh must be encoded to NTSC or another appropriate video format. (The standards in Europe are PAL and SECAM.) A *video encoder* is circuitry—often on a NuBus board—that converts your computer's RGB video signals into a format that can be displayed and recorded by video equipment. Examples include the Truevision's NuVista+ and Radius' VideoVision Studio. There also are a wide variety of external encoders that will convert the output from your standard Macintosh internal video or display board to NTSC format.

## Hardware Tips

Here are some tips for getting the most performance from hardware used for Macintosh 3-D. For more ideas, see chapter 13, “Advanced Topics.”

- **Turn off unused and superfluous system extensions.** Apple’s System 7.5 includes a control panel to do this. It enables you to create sets of startup extensions that you can use when you are performing different tasks. In particular, turn off AppleTalk and File Sharing unless they are in use. (They are required for printing and peer-to-peer networking, among other uses.)
- **Set your monitor to 256 colors using the Monitors control panel.** If you have a video board or a Mac that supports 24-bit (millions) of colors, be aware that performance will be much slower when you work in “millions-of-colors” mode. This is particularly true while modeling and scene building when the Mac is constantly redrawing the screen. Most 3-D programs will continue to work with a full 32 bits of color information, even when you’re only displaying 8 bits. Save 24-bit color for creating textures, viewing, and touching up finished images. Accelerated 24-bit video boards will greatly improve this situation, but things will still be faster in 8-bit, in general.
- **Use more than one computer.** This book was originally written and illustrated with the help of a 33-MHz Radius Rocket (68040) with RocketShare, which works essentially like a second Macintosh plugged into the NuBus. While the Mac is busy rendering, for example, you can switch over to the Rocket and continue modeling. This effectively eliminates sitting idly while you wait for things to happen. Distributed rendering systems, meanwhile, take advantage of networked Macintosh computers and (particularly since the advent of PowerPC) also are an effective way to speed things up. You can still buy Rockets, but I have since abandoned mine in favor of a Power Mac upgrade to a Quadra 950.
- **Loading your Mac with relatively inexpensive RAM** is perhaps the best hardware accelerator of all. If the Mac must constantly load code in and out of memory—or worse yet, virtual memory—the whole system will come to a virtual standstill. How much RAM is enough? This depends on the particular 3-D application you’re using and how complicated your images are. But in general, 4 MB of *free* RAM (beyond your system requirements) is plenty for PostScript rendering—a practical minimum for using programs like MacroModel and Sketch! is 8 MB; while 16 MB of RAM, or more, is required for high-resolution renderings of complex models. Like hard disk space, more RAM will almost always give you more speed. For professional-level 3-D work, 16 MB is probably a minimum amount of RAM.



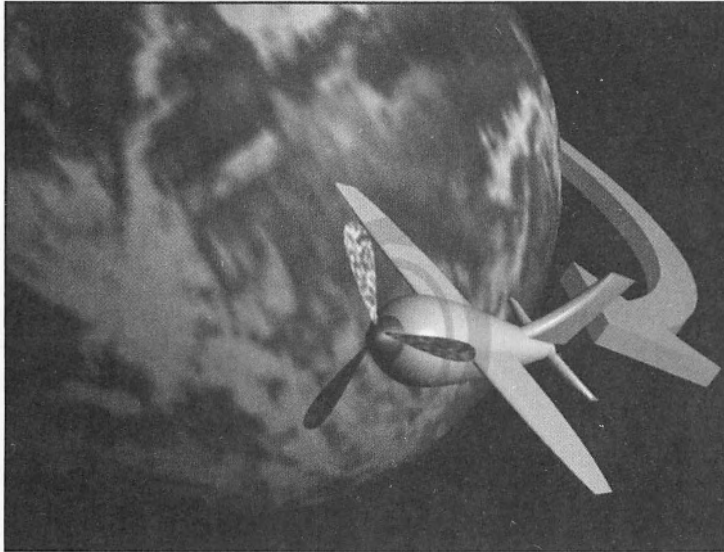
**C h a p t e r**

**3**

**Navigating  
in 3-D**

**I**magine a photographer surveying a landscape. She has the luxury of stereoscopic vision and a known point of view. This allows her to judge the length of a picket fence, the distance to a row of corn, the height of a mulberry tree, and the portliness of a grazing cow. Once she has assessed the scene, she can go about aiming her camera. By moving a few yards left, she can readjust the relationship of the cow, the fence, and the corn. If she lies on her back beneath the cow, she'll likely see a great deal of cow belly framed against blue sky, but precious little else. If she climbs the tree, she may find herself with a dizzying view of distant hills, or the wide back of the Guernsey. It all depends on her point of view.

Macintosh computers (like most others), work in two dimensions. What you see onscreen is similar to what you can print on a piece of paper. The illusion of depth in 3-D is created by perspective, highlight, shadow, and other factors (see figure 3.1).



**Figure 3.1** *Many factors create the illusion of depth*

Navigating in 3-D involves two parts: moving objects and changing your point of view. The distinction is important, although the results can sometimes appear equivalent. Take the example of a tea pot. If you are viewing the pot from the front and you lift the lid several inches, you will clearly see the gap between the two parts. However, if you view the pot from directly above, you will not be able to determine if the lid is touching the pot or floating above it. From this point of view, both states appear to be the same. Stereoscopic vision allows us to perceive depth even when other cues are missing, but this luxury is lost when working on the computer.

Now, suppose you want to see the handle of the pot framed in profile in your current point of view, but the pot happens to be poised to pour tea into a cup. You can rotate the pot so that its handle is in profile in your current view, but this will change the pot's relationship to the cup; you'll end up spilling tea on the table. Alternatively, you can leave the pot where it is and change your point of view so that you can see the pot's handle in profile; the cup and the pot will maintain their relationship, but both will also change their position in the current view.

For most people, working in 3-D is disorienting at first. After all, there's nothing normal about switching to a top view to see whether or not a table's legs are really under the table. It's sometimes helpful to think of yourself as a mosquito with super powers. You can look at your 3-D world from any point of view, and you have the power to pick things up and move them. After a while, even objects in a tri-view window (top, front, and side) will seem to take on a logical and simple sense of height, width, and depth.

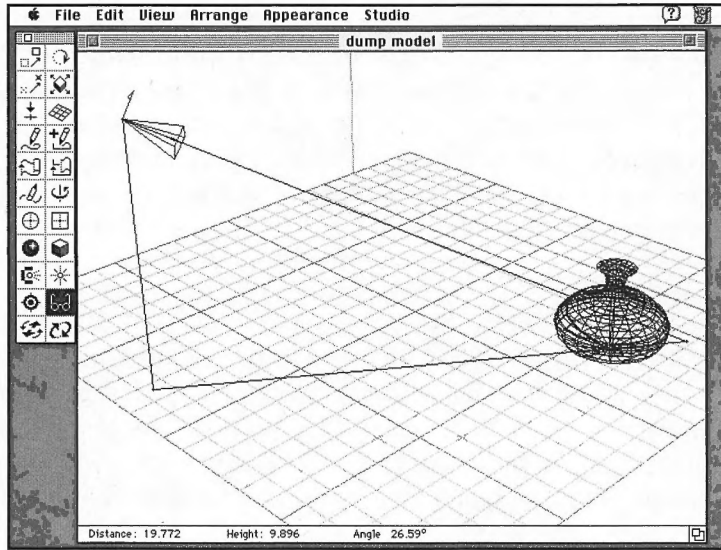
## **The Camera**

Working in three dimensions requires that you explore a virtual world much like the photographer in her landscape. While 2-D drawing limits you to a single point of view, 3-D enables you to have an infinite number of points of view. In fact, many 3-D software packages use the camera metaphor as a way to position and save different views of a scene (see figure 3.2). By aiming and placing different cameras, you can switch from view to view instantly, like a security guard monitoring a high-rise building with many hidden TV cameras. Not only can you keep your perspective, but you can keep several, and switch among them.

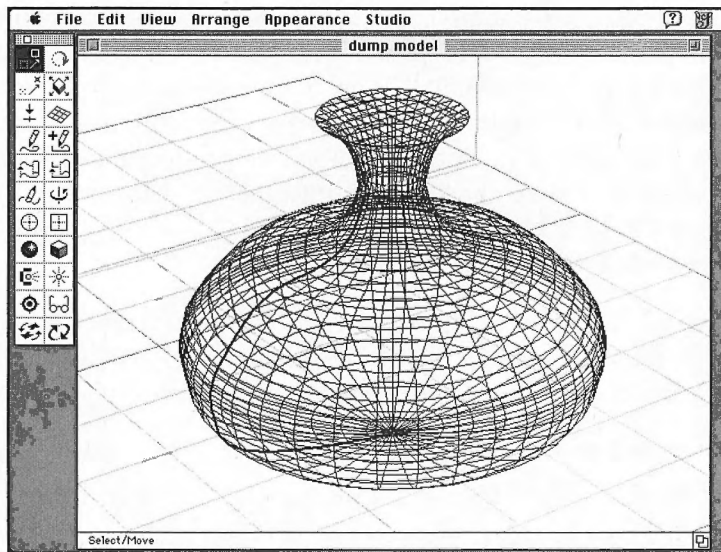
Some programs, such as Adobe Dimensions and Pixar's Typestry, offer a variety of fixed views from different perspectives in lieu of moveable cameras, but the principle is the same.

In Sketch!, the camera view shown in figure 3.2 provides the following view of the subject as shown in figure 3.3.

Some programs—particularly those that specialize in animation—enable you to move a camera through a scene. For example, you can circle an object with the camera to simulate a 360 degree fly-by. In some animation programs, the camera can attach to and follow a moving object; a feature often called *tracking*. You can also *zoom* and these cameras *dolly in or out* for realistic movie effects.



**Figure 3.2** Most 3-D programs enable you to easily position and aim a camera in its single-perspective view



**Figure 3.3** The camera view in figure 3.2 provides this point of view

The great advantage of having cameras as objects in 3-D is that you can move and reposition a camera in one view—then switch your point of view to that camera. This is critical to animators who can move a number of cameras to follow the action. Cameras can be suspended from trucks and booms, mounted to the front of a roller coaster or a stunt man's helmet, dangled out of a helicopter, or tacked to the nose of a buzzing fly.

**Tip**

You can tell a lot about the overall capabilities of a 3-D product from the type of camera it does or doesn't use. Here are many of the programs mentioned in this book, divided by camera type.

- Moveable still image cameras: Alias Sketch!, Pixar Showplace, PixelPutty Solo, Ray Dream Designer, Sculpt 3-D, Turbo 3-D
- Single still camera, moveable world: KPT Bryce, MacroModel
- Fully animated cameras: SketchTech UpFront, Animation Master, Electric Image Animation System, Infini-D, LogoMotion, Extreme 3D (speculative), Presenter Professional, Sculpt 4D (version 4.0), Shade III, StrataStudio Pro, StrataVision 3d
- Switchable fixed-position cameras: addDepth, Dimensions, Strata Type 3d, Typestry
- Virtual real-time cameras: Virtus vr, Virtus Walkthrough, Virtus Walkthrough Pro

## **Pan, Zoom, Tilt**

In 3-D programs that do not offer moveable cameras, there is usually a way to change points of view, and in most cases, to save different viewing positions.

There are a number of ways you can change your view: by dragging the world across your view window, by rotating the world, and by zooming in or out with a magnifying glass or similar tool. Even though you may not have camera objects, per se, you're still changing or moving a camera when you change your point of view. Keeping this in mind will help you visualize the changes that will bring you to the desired point of view.

## Lenses

Like all good photographers, as a 3-D user you carry a goody bag full of lenses that you can interchange as necessary. Each lens has a different focal length which provides a different angle of view. The angle of view becomes narrower as the focal length increases, so a 400-millimeter lens may zero in on a single, far-off sheep, while a 15-millimeter lens will take in the pasture, mountains, and a good deal of sky.

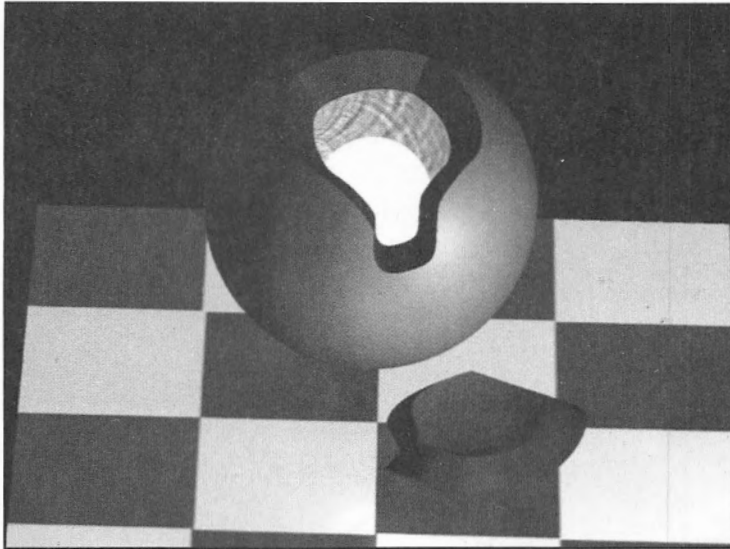
## Moving Camera vs. Lens Length

Zooming in by moving the camera closer to the subject (*dolly in*), and zooming in by changing lens length have two different effects (even though the term zoom sometimes is erroneously applied to both).

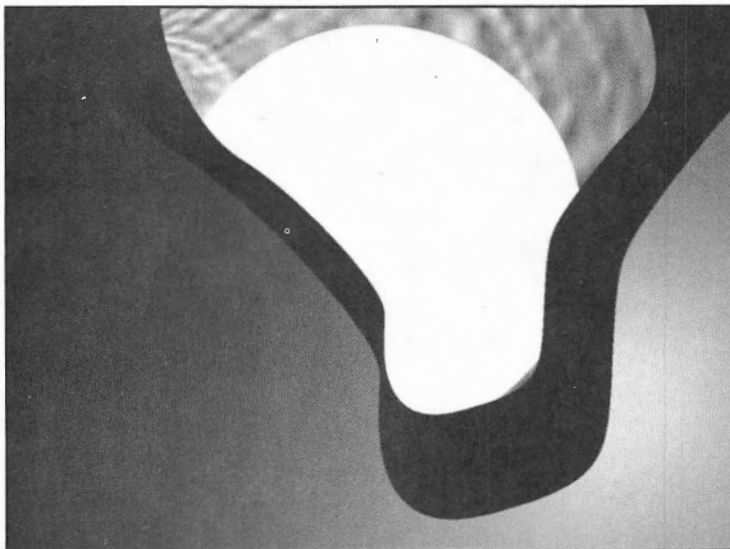
Standard human vision is approximately equal to the view provided by a 50-millimeter lens on a standard 35 mm camera. (That is why a 50 mm lens is considered normal and comes with most cameras.) When you change your lens to something other than 50 mm, you gain a sense of perspective that is not normal. With a very short (usually known as *wide*) lens of about 20 mm, you have a very broad field of view, but things begin to take on a great deal of distortion, as if you're viewing the world through the eye of a fish in a fishbowl. At the opposite extreme, long lenses of 100 mm or more (often called *telephoto*) have a very narrow field of view and cause flattening of scenes. That is, they tend to remove the sense of perspective from 3-D images. An *orthogonal projection* is the most extreme example of this flattening effect. (It is as if you aimed a telescope at a scene mounted on the moon.)

Figures 3.4 and 3.5 both show the scene shot with a 400 mm lens, but one was dollyed in to get closer to the subject. Compare this to the 6 mm lens (figure 3.8), which also moved the camera to get very close to the subject. In general, note the wide-angle distortion in the short lenses and the flattening effect with the longer ones.

Figures 3.6 through 3.9 show various lens lengths and camera positions. Except in the last figure, the camera was moved to maintain approximately the same contents in the scene.

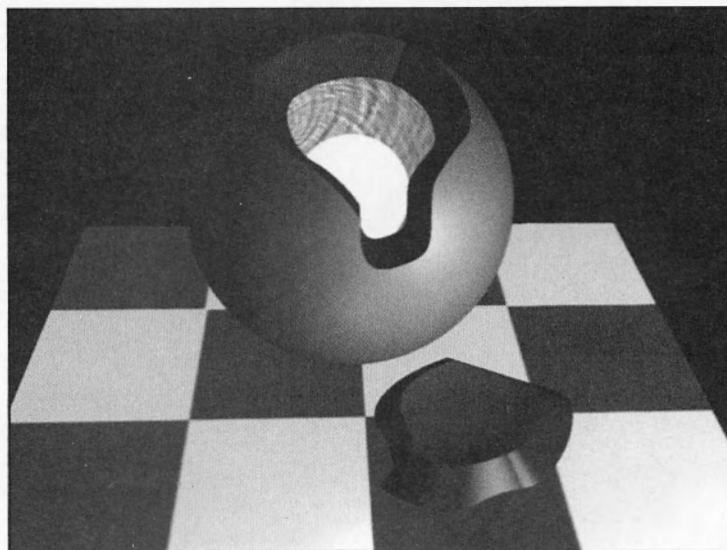


**Figure 3.4** 400 mm “telephoto” lens with the camera far away

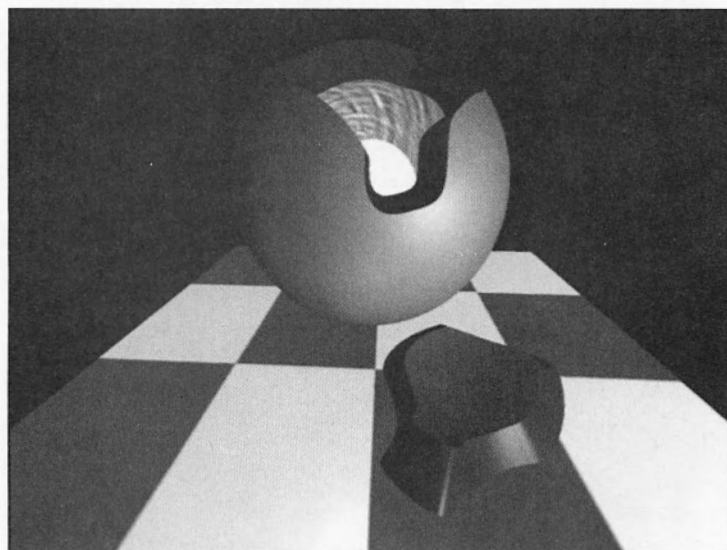


**Figure 3.5** A 400 mm lens with the camera dollied in close



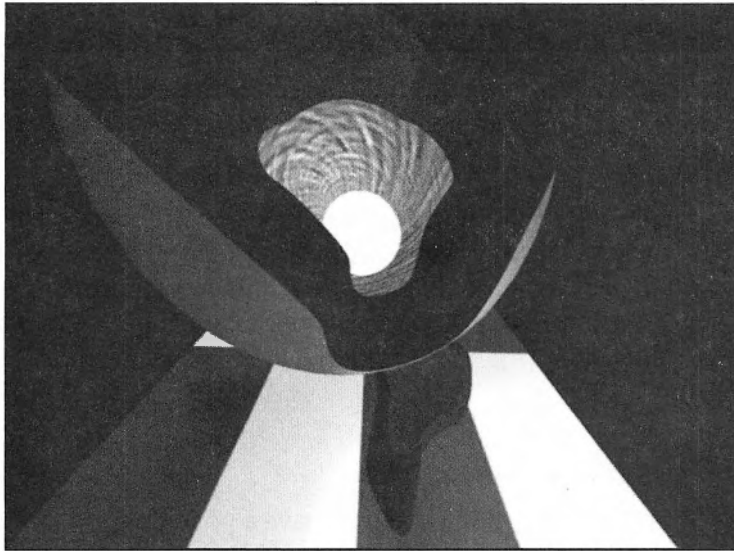


**Figure 3.6** 50 mm "normal" lens

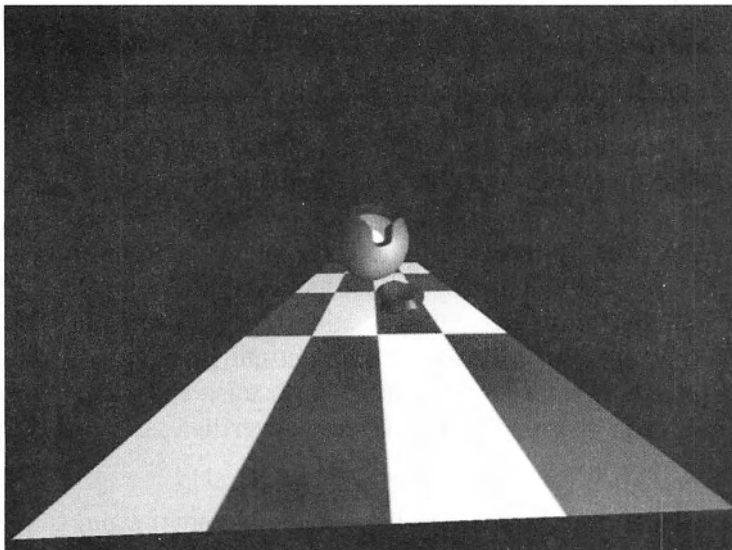


**Figure 3.7** 15 mm wide-angle lens





**Figure 3.8** 6 mm extreme wide-angle lens



**Figure 3.9** 6 mm lens camera dollied out to take in the entire scene

You can use these properties to your advantage. If you need a sense of more depth, you can use a shorter lens and move closer to the scene. This causes more wide-angle distortion, while preserving the content of the scene. On the other hand, you can use a long lens and move farther away to keep foreground and background objects proportionally sized.

Orthogonal views are usually the best for aligning objects because if things appear to be aligned, they are; this is not necessarily the case in perspective views. Some 3-D programs even offer border rulers that are visible in orthogonal views. Rulers wouldn't make any sense in perspective views because distances vary in different parts of a perspective image.

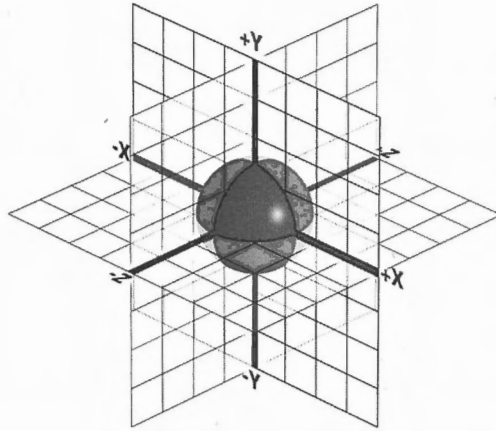
## Absolute Position: x,y,z

Both individual objects and viewpoint cameras have a position in space as well as an orientation. For example, a camera may be above, looking down. An airplane may be at 10,000 feet, flying east. These are contextual positions; that is, the camera must be above *something* and the airplane must be 10,000 feet away from *somewhere*.

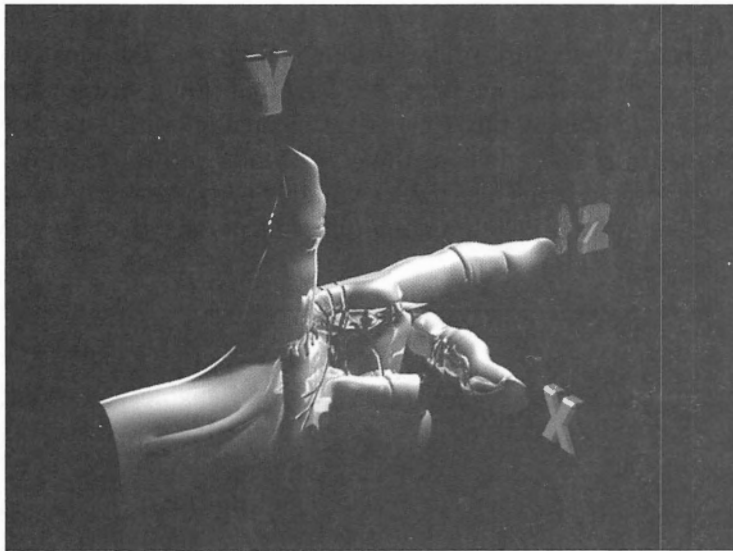
Three-D programs solve this problem by creating an "absolute world"—that is, a finite space in which all objects are located and oriented. When we refer to the position of objects, it tends to be in relationship to a known, fixed coordinate.

On maps of the real world, we use latitude and longitude to represent distance and direction from the International Dateline and the equator. In 3-D, the universal reference point is called the *origin* and everything is measured relative to this point in a space defined by three perpendicular axes which are universally referred to as x, y, and z (see figure 3.10). Just as we refer to an airplane's position relative to a given city, or an electron's position relative to its nucleus, 3-D requires that you have an origin for a given project and select an appropriate scale within which to work. (This is a generalization because several 3-D programs ignore these principles at least on the surface, nevertheless, these concepts help you to understand 3-D orientation.)

If you're like most people who loathed high school geometry and think Cartesia is an unfriendly nation, let's start with an idea you can count on three fingers. Hold your left hand in front of you, point your index finger at the person sitting across the room, your thumb into the air, and your middle finger to your right. Each finger represents an axis, x, y, or z. These fingers represent positive axes. If you move opposite the direction in which they are pointing, you will be moving in the negative direction of that axis. It's important to remember that this works only with the left hand; if you make the same gesture with the right hand, the middle finger points in the negative x direction.



**Figure 3.10** Objects are replaced in a 3-D world using known coordinates in  $x$ ,  $y$ ,  $z$  space



**Figure 3.11** The hand that rules the world: the  $x,y,z$  axes.

The Cartesian coordinate system is like the one you use to read a map by locating coordinates such as D-7 or A-9. The difference is that it includes in-out, left-right, and up-down directions. Since the map begins with  $x=0$ ,  $y=0$ , and  $z=0$  at the center of the universe, move in the direction of your pointing index finger and  $z$  increases; move up (thumb) and  $y$  increases; move right (middle finger) and  $x$  increases. Conversely, move left of center and the  $x$  value becomes negative; move down and  $y$  decreases; move in the direction opposite your pointing index finger and  $z$  decreases. (The standard notation for an object's position looks like  $(x,y,z)$ . For example, an object at  $(5,-10,7)$  is at  $x=5$ ,  $y=-10$  and  $z=7$ .)

In a three-plane view, this system enables you to position and scale objects very precisely. In a top view (looking down at the scene), moving an object changes its x and z values, but not its y value. Conversely, the front view can be used to change an object's x and y values, but leaves the z position untouched. For this reason, some programs label the top view window with an x-z, the front view window with an x-y, and the side view window with a y-z.

**Tip**

I've encountered two programs on the Mac that defy the "y is up, z is in" convention. In *Infini-D* and *PixelPutty Solo*, the y and the z axes are switched so that +z is up. To further confuse the issue, you may encounter CAD programs that switch the positive and negative z direction as well. However, the principles remain the same, you just have to rearrange your fingers.

As shown in chapter 2, the three-plane orthogonal view is the most common method of representing 3-D space in such a way that you can visualize what is going on at all times. Usually, this view is supplemented with a perspective view, so that you can get a realistic look at the scene while you work. Some programs use a single-perspective view. This enables you to view and work on objects in a natural perspective and to switch among different perspectives to stay oriented.

**Tip**

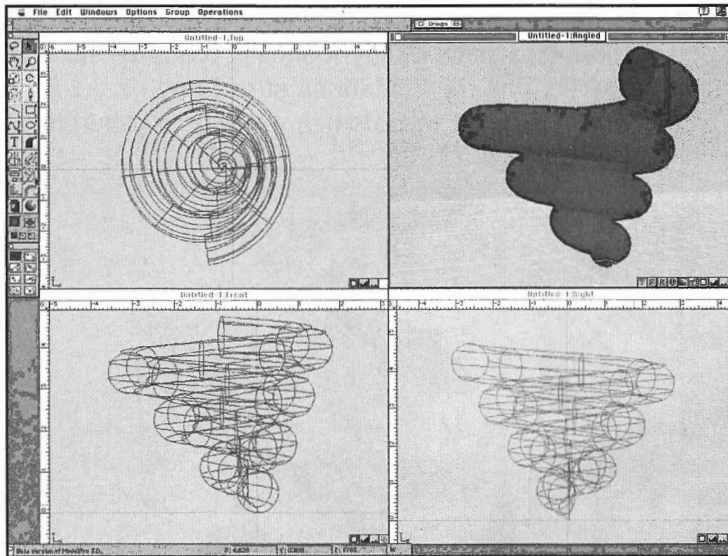
Some programs use a single perspective view and enable you to switch between front, side, top, and other views to align objects. This can make things easier to understand if well implemented. Other programs use multiple views, such as the tri-view, or variations of it, to look at the same scene simultaneously from different angles. The tri-view programs tend to make you more "aware" of the x, y, and z axes, but they're equally relevant in both. Which type of interface is better, is, of course, the topic of stormy debate. Most users are comfortable sticking with one type of program or the other. As far as I'm concerned both work and both offer advantages and disadvantages.

Examples of single-perspective-view programs include: *addDepth*, *Design Reality*, *Ray Dream Designer*, *Dimensions*, *Dyna Perspective*, *form•Z*, *MacroModel*, *Showplace*, *Sketch!*, *UpFront*, and *WalkThrough*.

Multiple-view programs include: Infini-D, LogoMotion, Playmation, Presenter Professional, Sculpt, Shade III, StrataStudio Pro, StrataVision 3d, and Turbo 3-D.

## The Three-Plane View

The natural world generally affords us only a single perspective view—that is, we see a scene from only one point, with only one physical perspective. In Macintosh 3-D, the alternative to this single-perspective is the more time-tested (although initially less intuitive) three-plane view. Basically, this approach provides three views of your model at all times: front, top, and side. Orthogonal views are most useful in this mode because they are free of lens distortion. Usually, most 3-D applications also provide a perspective preview window to show your progress.



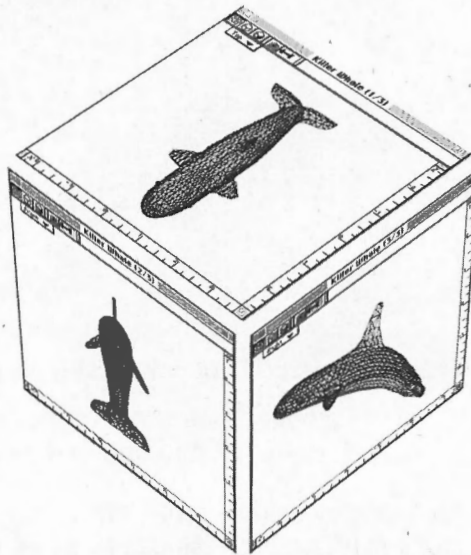
**Figure 3.12** This screen shows Presenter Professional's modeling screen, with (clockwise from top left) Top, Perspective, Right, and Front views

The advantage of the three-plane view is that you can create very precise alignments. For example, in VIDI's Modeler Professional, to draw a series of circles one atop the next (as if stacking a pile of pennies), you begin by drawing a circle in the top view. You then move to the front or side view (where your circle now looks like a mere straight line) and lock your drawing position an inch above the existing

circle. Then, you move back to the top view and draw another circle directly atop the first, and so on. This approach is very accurate, but it also requires several steps and it is not nearly as intuitive as the single perspective view, at least initially. Visualizing in three separate planes takes a good deal of practice. For instance, you have to get used to the fact that when an object in the Top view moves up and down, it's really moving forward and backward, and if it moves left and right in the Right view, it's also moving forward and backward.

This principle is commonly illustrated with a box to which the top, front, and side view windows have been glued. Each window is then looking in at the contents of the box (see figure 3.13).

In a tri-view, to draw an object that isn't flat, such as a spiral line of a spring, you'll need to frequently lock your drawing position in one window, switch to another window, continue drawing, and then smooth the lines in the different views. This can quickly become confusing. On the other hand, the precise control afforded by this approach means you can create very accurate complex shapes sometimes impossible to build in the single-window applications. Also, 3-D programs aren't limited to shapes you draw by hand. Many modeling features, such as sweeping tools, are designed to overcome the limitations of working on a 2-D screen by automatically doing what's difficult to do when drawing in different view windows.



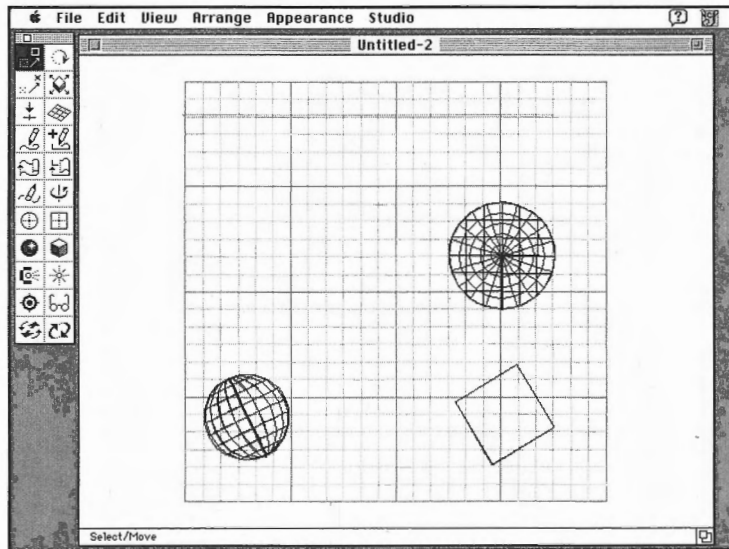
**Figure 3.13** How the tri-view windows line up: Front, Top, and Right views are left, middle, and right, respectively



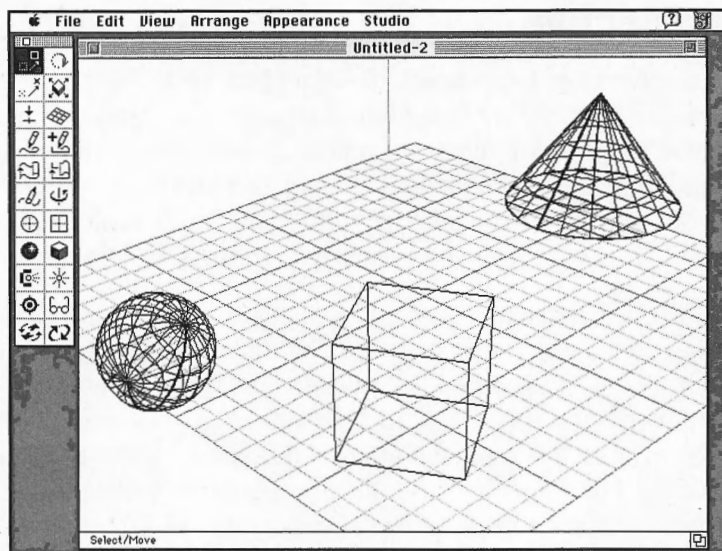
## Single Perspective

The single perspective view looks much like the place you're in right now. All objects are positioned in relation to a floor and walls (also known as working planes), which you can use for reference points. In fact, form•Z, Ray Dream Designer, MacroModel, Design Workshop, Showplace, and Sketch! show you floor plane grids instead of x,y,z coordinate axes, since it's much easier to align objects to one or more grids than to x,y,z axes when you're working in perspective.

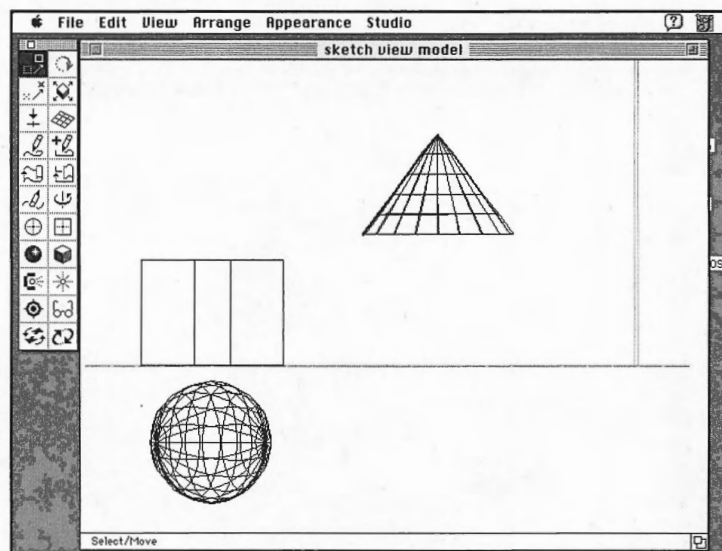
The advantage of this approach is that you can easily visualize what you're doing because your view looks like the real world. The disadvantage is that you must frequently switch points of view and make extensive use of some fairly sophisticated tools, such as guides and snaps, to help you align your work. Otherwise, what you will see in your 2-D representation of 3-D space will not necessarily coincide with reality. For example, three blocks appear to be level with each other when viewed from the top or from an elevated perspective (see figure 3.14 and figure 3.15); but when viewed from the side (see figure 3.16) or front (see figure 3.17), it is immediately apparent each is at a different elevation relative to the ground plane.



**Figure 3.14** Top view



*Figure 3.15 Elevated perspective*



*Figure 3.16 Right-side view*



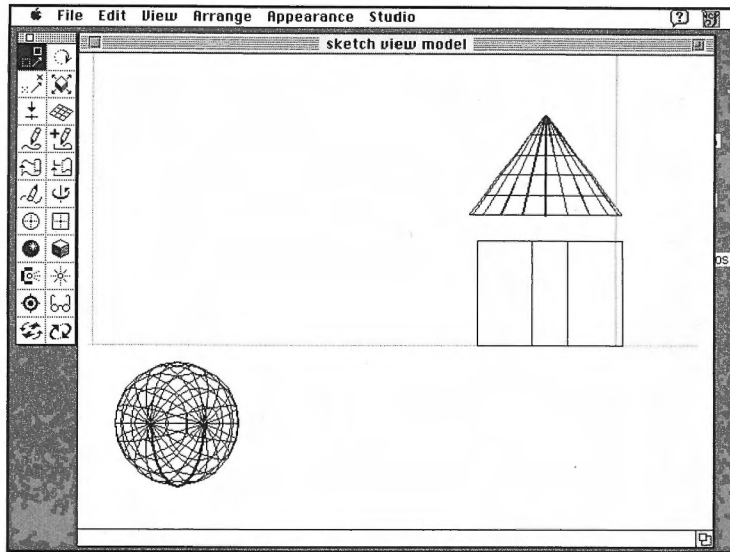
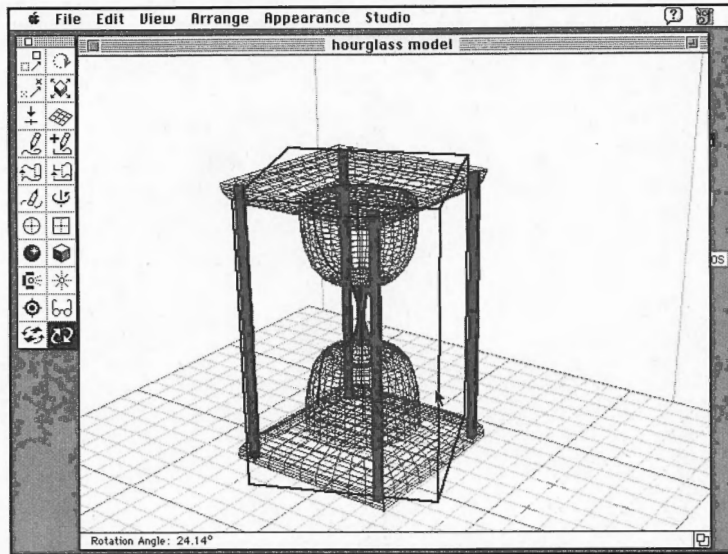


Figure 3.17 Front view

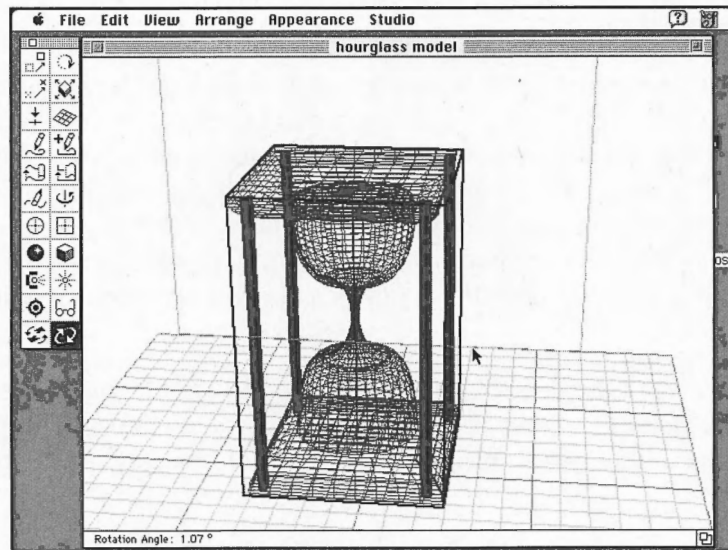
## Virtual Trackball

One of the most important tools for navigation in the single perspective is the *virtual trackball*. While this feature goes by a number of names, essentially it enables you to use the mouse as a handle that rotates your 3-D world in any direction. By “grabbing” the bottom of the screen and pushing it up, you roll the world away from you. By grabbing the top and pulling it down, you roll it towards you. By grabbing the left edge and pushing it right, you spin the world around the central axis of the screen. All of these movements are analogous to spinning a globe.

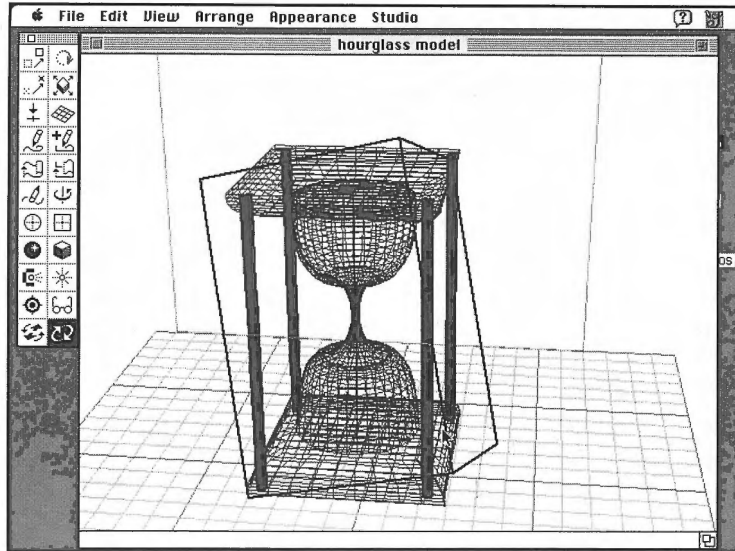
Some programs, such as Alias’ Sketch!, enable you to select single objects, or groups of objects as the center of the universe. All rotations using the virtual trackball then revolve around this object or group of objects (see figures 3.18 through 3.21).



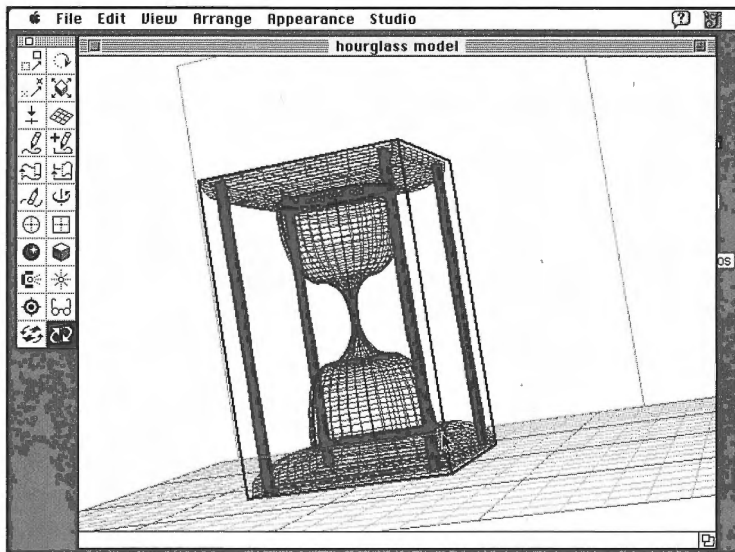
*Figure 3.18 Turning world around y-axis using a shape as a "handle"*



*Figure 3.19 The rotation in figure 3.18 results in this view*



**Figure 3.20** *Turning the world around the z-axis*



**Figure 3.21** *The rotation in figure 3.20 results in this tilted view*

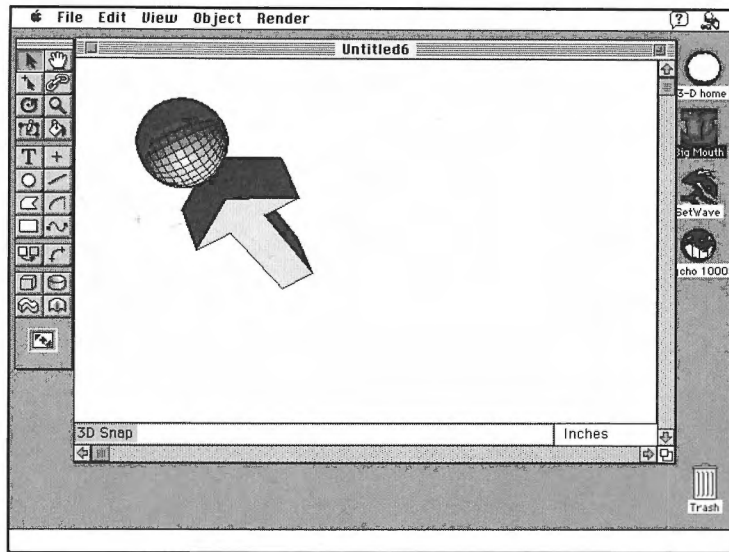
## Moving and Positioning Objects

When building models and scenes, navigation takes on new importance. Now, in addition to moving a camera to change your point and field of view, you can move objects around in space. Imagine building a car: it's important that the wheels align to the car in the proper position under the fenders—where they belong, not hovering several feet above. When viewed from directly above, the wheels may appear to be aligned properly. However, when viewed from the side, the illusion may shatter. For this reason, you will almost always need to see your objects from at least two orthogonal views in order to properly align them.

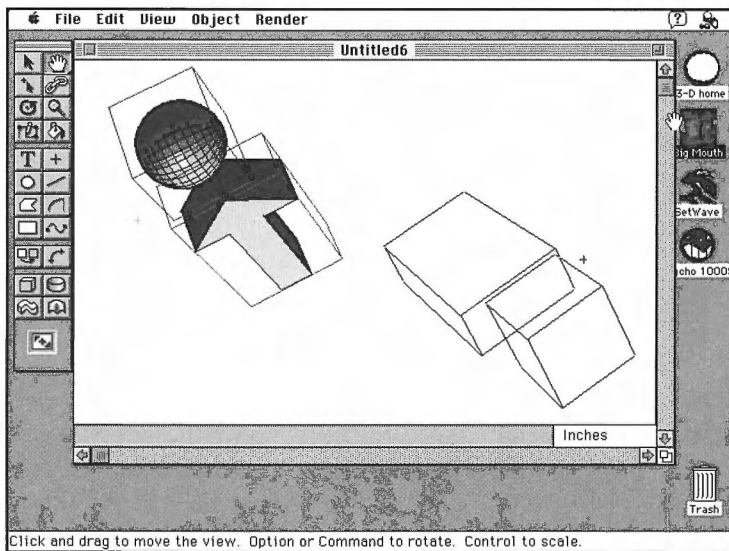
The capability to represent where objects are in 3-D space, even though their position can be displayed only in two dimensions in any one window, is a capability common to all 3-D software. The trick is to learn to “read” your software's representation of space and take advantage of its navigation tools.

### Bounding Boxes

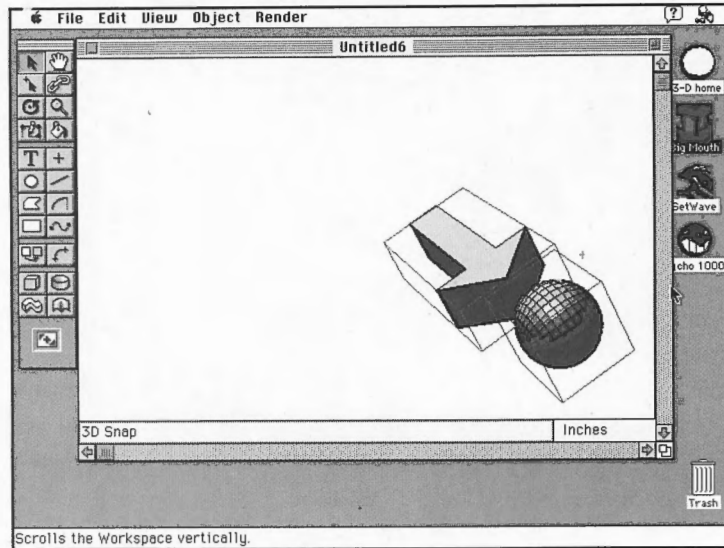
While moving an object is usually as simple as dragging it with the mouse, an important tool for understanding your 3-D screen is the *bounding box*. Because 3-D models are mathematically complex, the computer cannot redraw them nearly as fast as you can think or work. When moving or rotating an object in 3-D, programs usually employ a bounding box which is a wireframe cube that stands in for your object. Because a bounding box can be redrawn very rapidly, it is an effective tool for visualizing the alignment and orientation of objects. Bounding boxes often are employed as local 3-D trackballs for modifying and rotating individual objects. The sequence shown in figures 3.22 through 3.24 shows how a bounding box can be used to represent a group of objects as it is rotated in MacroModel.



**Figure 3.22** The object is in fully shaded preview



**Figure 3.23** When you click on the object with the rotate tool, a bounding box appears around it



**Figure 3.24** After rotation

#### Tip

When you need to see an object from a different perspective, you are usually better off changing your point of view than moving the object. You can generally get to your “home” point of view from other far flung perspectives. On the other hand, if you move an object to see a little more of its left side, for example, it may be difficult to place it back in its original orientation to other objects. Sketch! enables you to change your point of view by using an object as a sort of handle to rotate the world. Imagine that every piece of furniture in a room—the room being the 3-D world—is glued to the floor. If your omnipotent hand reaches into the room and turns the coffee table, the whole world will rotate around with it. The coffee table keeps its relationship to the world, but changes its orientation relative to you. MacroModel also uses a trackball approach; however, that program uses an actual virtual ball that you turn. The world turns in response to its movements.

## **Grids and Snaps**

*Grids* and *snaps* are navigational tools used to precisely determine your position and to help place objects. Grids are often laid out on working planes. These are flat planes positioned at a known orientation in space that are divided into a grid of squares of known size. You can use these grids to draw new objects and to place existing ones. Many programs, particularly those with a single perspective window, offer a default “floor” plane which lies on the horizontal (x-z) axis. Assuming you are working in a room, anything positioned on this plane is equivalent to resting on the room’s floor. If you draw a circle on the ground plane and extrude it upwards, you get a round column standing on the ground. Some programs, like Ray Dream Designer and Design Workshop, also offer “wall planes” (which lie on the x-y and z-y axis). The best implementations of this scheme enable you to create and position your own working planes which lie at odd angles, since not all objects are created along the three axes. Examples of these include Sketch!, form•Z, and Design Workshop.

However, tri-view programs also use grids, simply because they make it easy to align objects to precise locations.

You can use snaps to align the cursor’s position to known points in space. (Without them, you’re likely to be tricked by illusions of depth.) When drawing on a working plane, snaps can be used to confine objects to known dimensions. For example, if you want to draw a perfect square, you can snap to four equidistant corners on the reference grid as you draw. When positioning objects, you can snap them to a precise position on a grid. Some programs (particularly of the CAD persuasion), enable you to snap to points, lines, and even surfaces of objects. For instance, you can snap from the center of a circle to the endpoint of a distant line, or you can create a line that just touches (is tangent to) a circle.

In order to know where your object is in relation to the working planes and other objects, single-perspective view programs sometimes draw extent lines between the object being moved and snap points or ground and wall planes. This tells you that you are actually snapping to the desired point.

You can tell that the central figure is on the ground because the extent lines on the walls touch the floor. The extent box on the floor indicates the model’s relative distance to the walls, and the small extent boxes indicate camera and light positions.

## Reference Point

When positioning an object in 3-D space, one consideration is where to place its reference point. The *reference point* is a single x,y,z point that is often the center of an object; sometimes it is easier to think of this as an object's heart. Whenever you position an object in space, it is relative to this heart. The reference point maintains its relationship to the rest of the object, unless you intentionally change it. When you snap an object into position, it is usually this point that snaps onto the destination point.

Presenter Professional and Sculpt both enable you to pick any point on an object to snap to a destination point. You don't have to pick the object's reference point.

Some software will enable you to change an object's reference point, while other applications keep it a mystery. Your results in positioning an object in 3-D space vary greatly depending on whether you position an object's upper-left corner or its center at a given point in space. (Putting your toe on the line is much different than putting your nose there.) Using the reference point as a pivot point is useful if you want to make a hinged object rotate around one of its corners, for example. In some programs, such as StrataStudio Pro and Electric Image, it's possible to move the reference point far away from the object. You may, for instance, want to position a moon's reference point at the center of a planet so the moon orbits the planet as it rotates.

## Absolutely Relative

While it's possible to position objects absolutely in the x,y,z space, it's important to distinguish *absolute* movement and orientation about the coordinate axes from *relative* (or *local*) movement or rotation.

Absolute position is measured from the world origin, the "center of space." If you move an object using "absolute" positioning to 0,0,0, it will move to the center of the space. On the other hand, relative position is measured from the center of the object, so if you move the object to 0,0,0 using relative positioning, it won't budge at all. The same is true of rotation: if you rotate an object around the x axis using absolute rotation, it revolves around the world's horizontal x axis. If you revolve it around the x axis using relative rotation, it will revolve around the x axis that passes through its own reference point.



## Orientation: Pitch, Yaw, and Roll

It's helpful when working in 3-D to understand the concept of pitch, yaw, and roll. Even though not all programs use these terms, they apply in general to the rotation of objects in space. In simple terms, every object carries with it its own set of x, y, and z axes.

For example, if an airplane is flying through the air—as far as the pilot is concerned—it can *pitch* up or down; it can *roll* side to side; or it can make a flat turn so that it's sliding sideways (called *yaw*). The surprise comes when we learn that the airplane is upside down relative to the earth. Now, if the pilot pitches his plane upward (relative to himself) he is turning his plane towards the ground. If he yaws to the left (relative to himself) he yaws to the right relative to the earth, and so on.

### Tip

Rotation around an axis is like using the axis as a rotisserie on which you turn a chicken. Rotation around the x axis means your object is skewered by, and turning on, the x axis. In terms of global rotation, this means you are literally rotating about the coordinate x axis. (The line where y and z are equal to zero.) In the case of local rotation, the object's axes are assigned when you create the object, and they intersect at the object's reference point.

Pitch, yaw, and roll are relative to the current orientation of an object—that is, if a plane is upright and heading east and you apply 90 degrees of yaw, it will head south (if the rotation is clockwise). This is most useful when you link one object to another. In the following figures, the propeller is linked to the airplane, and the plane has “z” for its long axis. No matter what orientation the plane is in, as long as you supply a local z-axis rotation to the propeller, it will spin normally on the nose of the plane. If you were to supply a global z rotation to the prop, it would fly away from the plane.

Local rotation is affected by *rational order*; that is, if you apply pitch before roll, the resulting orientation of the object is different than if you apply the same amount of roll followed by the same amount of pitch. This is like flinging a bucket full of water and then dumping it, as opposed to dumping the bucket and then flinging it.

Assume that an airplane's original position has it moving forward and flying level. Assume also that the fuselage of the plane is on the z axis, with the plane aligned flying towards positive z. In this case, pitch, yaw, and roll can be described the following ways:

- **Pitch.** This is how much the plane turns up in the air or down at the ground. A plane taking off from a runway is said to have a high degree of pitch as it leaves the ground. (Architects also will recognize the term pitch as referring to the steepness of a roof, and it is used in this context in architectural CAD.) A dive bomber making a pass at an enemy aircraft carrier, on the other hand, has a sharply negative pitch. In the x, y, z lexicon of 3-D, pitch is the amount of rotation around the local x axis. In this case, the x axis is formed by the wings of the plane.
- **Yaw.** This is the amount of "fishtailing" (y-axis) rotation of the airplane. If the plane yaws suddenly to the left, the blast of onrushing wind will hit the right side of the airplane instead of the nose cone.
- **Roll.** Roll of the "z-oriented" airplane is when it banks or "tilts" over on either wing. From inside the cockpit, facing forward, the horizon appears to tilt to the left when the airplane rolls to the right. Roll, officially, is rotation around the local z axis.

Pitch, yaw, and roll are particularly useful when working with *linked* relationships in animation. Unlike grouped objects, where two parts are basically cemented together, linked objects have "parent-child" hierarchical relationships. The Moon, for example, is linked "child" of the Earth. Even though the Moon is constantly moving relative to the earth, the two have an unbreakable link between them. As the Earth moves around *its* parent object, the Sun, the Moon keeps its link to the Earth and moves with it.

When two objects are linked in a hierarchy, the local orientation of a child object is relative to the parent object. Modeler Professional, Swivel 3D Professional, Electric Image Animation System, MacroModel, Shade III, and Infini-D provide this type of orientation.

For instance, in the example of the solar system, if you rotate the Moon 180 degrees so that the dark side faces the Earth, the dark side will continue to face the Earth no matter how much the earth spins relative to the Sun.

A common attitude adjustment you can make to a model is a *translation* or *delta*. This is where the values you enter for a change in rotation (or scale) are relative to the position of the object prior to the change. You may use this kind of adjustment, for example, when you want to "nudge" an object in a certain direction.

## Saving Views

Many programs enable you to save views by name. You can store these views in a menu and you can switch among them. Most 3-D applications offer a view called “top” which will give an orthogonal or at least narrow-angle view of the scene from directly above. But some programs that enable you to save views also can have views such as “Top right, offset left” or “Bottom view, zoomed in.”

Switching between these views gives corresponding results. This capability can be very important when modeling and scene building and you need to see a particular relationship from different angles. It also is useful when you plan to render a scene from several points of view. Imagine that you have modeled a telescope that you need to see through. You can create a view that positions you to peer through the telescope. You can then return to it periodically while you align your planets.

In programs that support cameras as objects, named views correspond to different cameras. Changing a view by scrolling and zooming around changes the camera’s aim; while changing the camera’s aim or position alters what you see in the camera view.



**C h a p t e r**

**4**

# **Basic Shapes and Fonts**

**M**odeling is the creation of shapes. In many ways, it is like drawing on a computer in 2-D. However, drawing a simple 2-D shape is only the beginning of building a 3-D object. Once you draw or import a shape, extrusion, lathing, and lofting are just some of the operations you can perform to add the element of depth.

## Simple Geometry

The easiest models to work with are the ones someone else builds for you. Clip models are available from the makers of 3-D software, as well as many other vendors.

Although only a few high-quality 3-D model collections are currently available, the medium of CD-ROM makes it possible to distribute hundreds of objects on a single disc and it's likely to become a popular media for this purpose. Several companies listed in appendix A are already doing this.



The CD-ROM that comes with this book, for example, has many models to work with.

Several excellent 3-D forums are available to users of online services. 3DSIG on America Online, in particular, has substantial collections of 3-D clip models. The Alpha Channel, my own San Francisco BBS, also has a growing collection.

By combining two or more clip models, you can create a variety of new shapes without actually modeling at all. Typical examples of clip models include furniture and lamps for decorating interiors, cars, 3-D type, human forms, robots, and maps. Some companies are in the business of digitizing real 3-D objects and selling the resulting 3-D models. Provided you obtain models in compatible formats, it's easy to import them into 3-D programs and combine them into single objects and scenes.

If you're not using clip models, then you will need to create your own. The easiest place to start is with existing 2-D line art and type. You can bring these into most 3-D applications and they can be extruded or lathed into the third dimension.

Once you go beyond these simple measures, you can use basic geometries (called *primitives*). You can quickly lay them on a page like building blocks. Beyond that, there are many sophisticated tools available that enable you to draw and manipulate objects in three dimensions.

In general, the more complex the modeler, the more difficult it is to use; the more sophisticated the features, the more expensive the product.

## A Sense of Scale

A common mistake made by beginning modelers is failing to properly account for the scale of objects in relation to one another. For example, a clock is not as big as the desk on which it sits and a lamp generally does not reach from the floor to the ceiling. Unless you are specifically trying to create strange-looking images, keep objects in proper proportion to each other.

Set the scale in your 3-D program to reflect the real world you are representing. This way you can size your model to natural dimensions, making it easy to keep things in proper relationship.

## 3-D PostScript

A number of programs specialize in adding depth and perspective to PostScript type and line art. Ray-Dream's addDepth and Adobe's Dimensions have been designed specifically for this purpose. Unlike most 3-D applications whose output is in the form of bitmapped art, these two produce Bézier line art that you bring back into programs such as Adobe Illustrator or Aldus FreeHand. These applications quickly and painlessly create 3-D artwork.

While the advantage of PostScript 3-D programs is their extreme ease of use, they are not intended for creating complex, photorealistic scenes. The effect they create is decidedly "illustrated." However, 3-D PostScript occupies an important illustration niche. PostScript has many advantages familiar to anyone with desktop publishing experience. Most importantly, the renderings created by Dimensions and addDepth are *resolution independent*, which means that you can print them at any size and get exactly the same print quality. This is decidedly different from bitmapped 3-D renderings, which are made up of lots of small pixels; if you enlarge one of these, you end up with lots of big pixels (which don't look very realistic).

Dimensions can import a model from Swivel 3-D; Alias Sketch! can read a Dimensions file, but Dimensions doesn't support the range of geometry and other features found in photorealistic 3-D programs. Because of these limitations, these programs are sometimes referred to as "two-and-a-half-D." Admittedly, however, Dimensions, in particular, has gone a long way towards blurring that distinction.

On the other hand, both Dimensions and addDepth are very capable packages for illustrators who need the advantages of PostScript art. For illustrators using 2-D tools, these programs take the pain out of creating 3-D shapes and scenes, and the results easily can be embellished with standard 2-D tools. For example, you can

model a computer in Dimensions, add surface properties and lighting, and render the scene in a few minutes. You can then import the result into Illustrator where you can add all the artful embellishments.

Some of the other modelers mentioned in this book will import EPS line art for extrusion into 3-D, and most will export Encapsulated PostScript (EPS) bitmaps. The most important distinction is that the programs that deal in bitmapped art are capable of providing more realism; the PostScript programs, for example, don't support colored light sources or shadows. However, PostScript art is resolution independent, and it can be manipulated in a simple and efficient manner, just like other Illustrator files.

## Adobe Dimensions

In version 2.0, Adobe introduced a scaled-down set of Adobe Illustrator's excellent drawing tools to Dimensions that made Dimensions a do-everything illustration tool. But Dimensions also can import art from Illustrator (as well as Aldus' FreeHand, or Deneba's Canvas). Dimensions still shines best when combined with Illustrator's powerful artist's tools.



You can find the demo version of Dimensions on the CD-ROM (in the commercial applications folder).

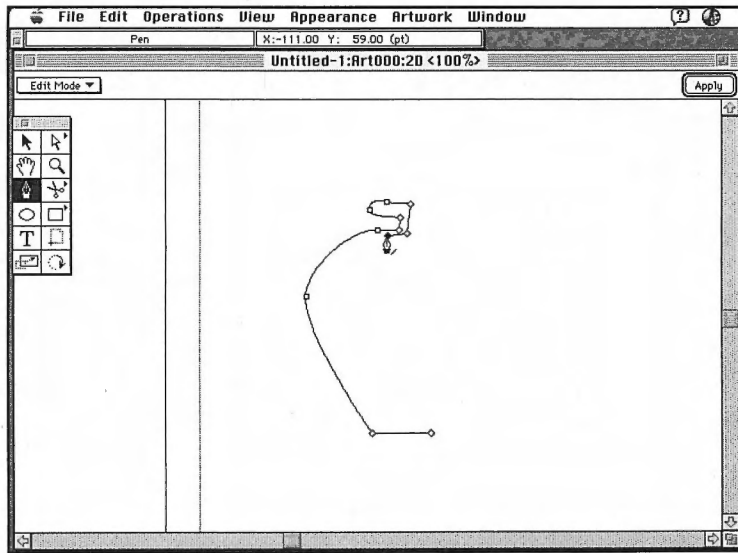
## Creating a Vase

In this example, the outline of a vase is created in Adobe Dimensions, lathed into a 3-D shape, supplied with a surface finish as well as a texture map (that was originally created in Illustrator), and then rendered. The resulting artwork is incorporated into an Adobe Illustrator project.

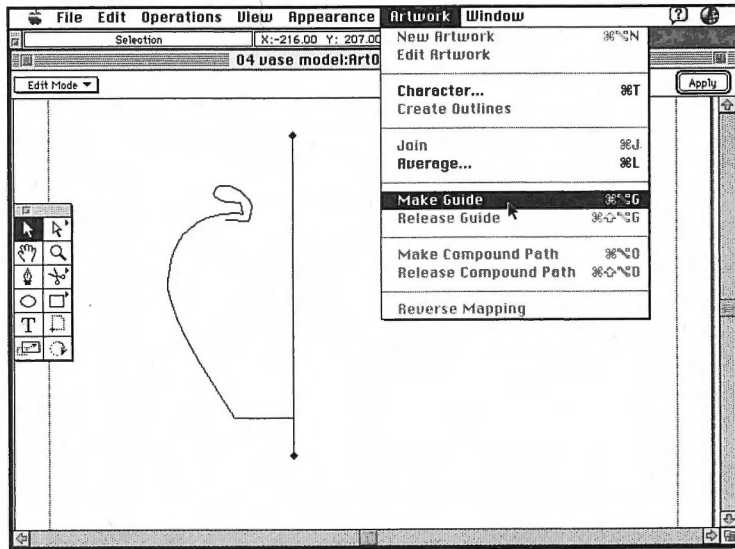
1. Launch Dimensions and use the 2-D Pen tool to draw an outline of one half of a vase. If it's not perfect the first time, you can come back and edit this outline even after you've created a 3-D shape from it (see figure 4.1).
2. Draw a vertical line and drag it over to just touch what will become the center of the vase (see figure 4.2). With this line still selected, choose Make Guide from the Artwork menu.

This will establish the line as the center axis for the Revolve step. Without this step, Dimensions uses the lower-left corner of the object as the center of rotation.





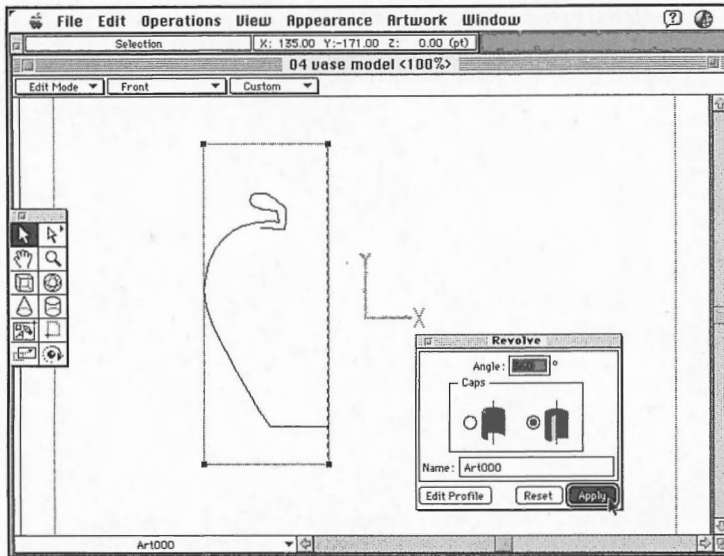
**Figure 4.1** Drawing a vase outline in the artwork mode



**Figure 4.2** Making a guide as a center of rotation

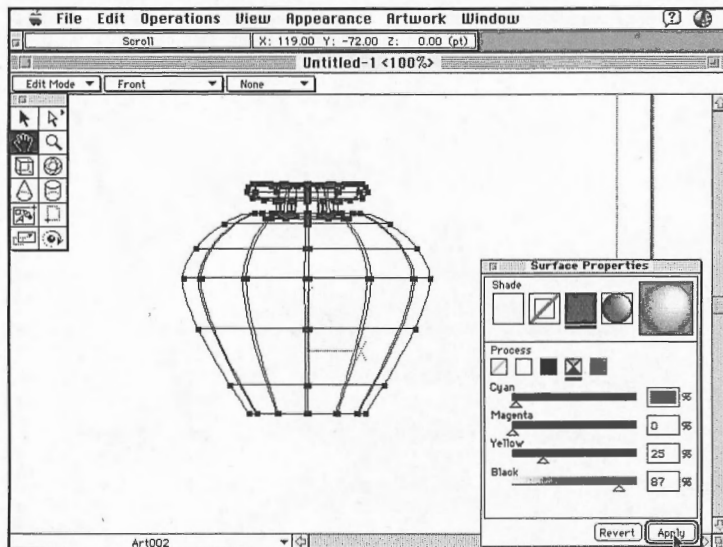
3. In the 3-D window, use the Revolve tool to lathe the shape 360 degrees.

Note that you could create a “wedge” shaped vase instead of one that’s revolved 360 degrees (see figure 4.3).



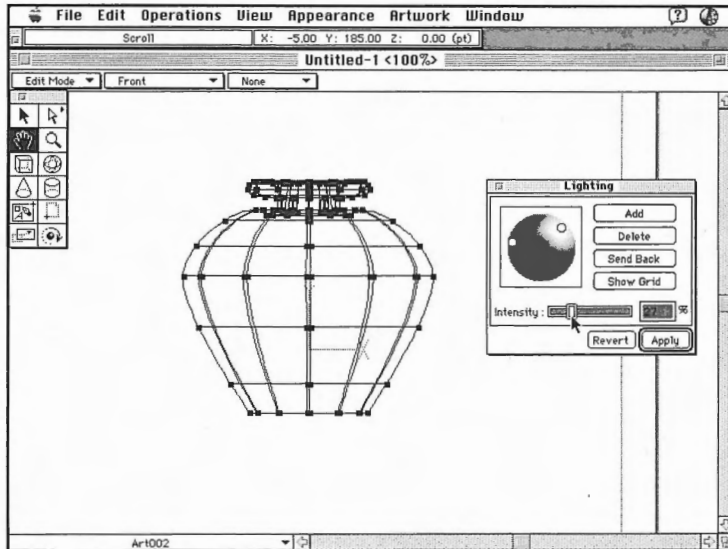
**Figure 4.3** *Revolving the shape*

4. Open the Surface Properties palette from the Appearance menu and assign a surface texture to the vase. Note that there are different settings for the fill color, stroke or outline (if any), shade color (the color that the model blends towards as it falls into shadow), and reflective properties, such as gloss and ambient reflectance, of the texture. In this case, give the vase a clay-like color and a soft matte finish (see figure 4.4).



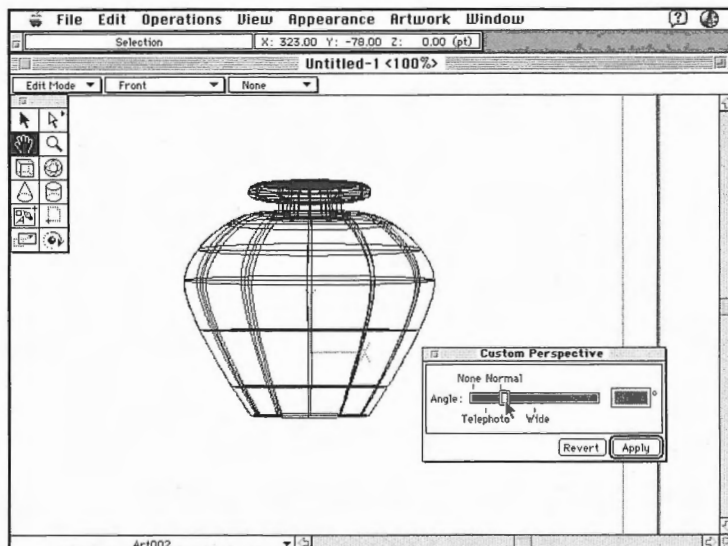
**Figure 4.4** *Assigning a clay-like texture*

5. Open the Lighting Palette and set two lights: a strong “key” light in the upper right, and a weak “fill” light on the left (see figure 4.5). See chapter 8, “Lighting,” for an explanation of key and fill lights.



**Figure 4.5** Setting the lighting

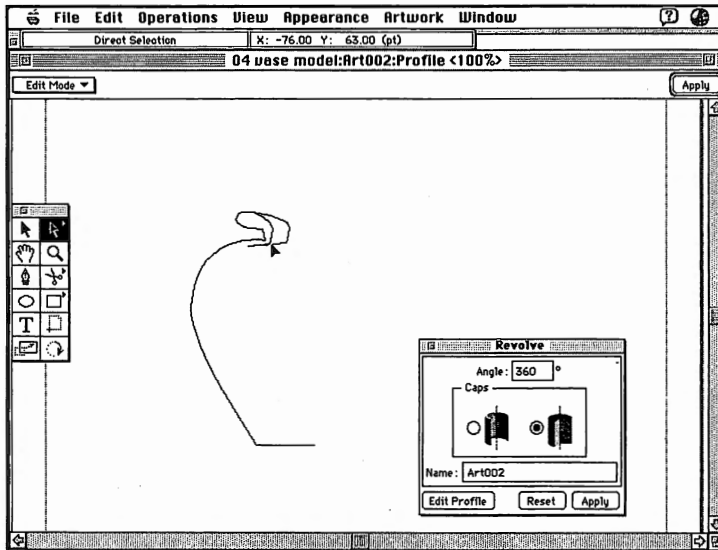
6. From the View menu, open the Custom Perspective palette, and set the perspective to “normal” with the slider (see figure 4.6).



**Figure 4.6** Setting the perspective to “normal”

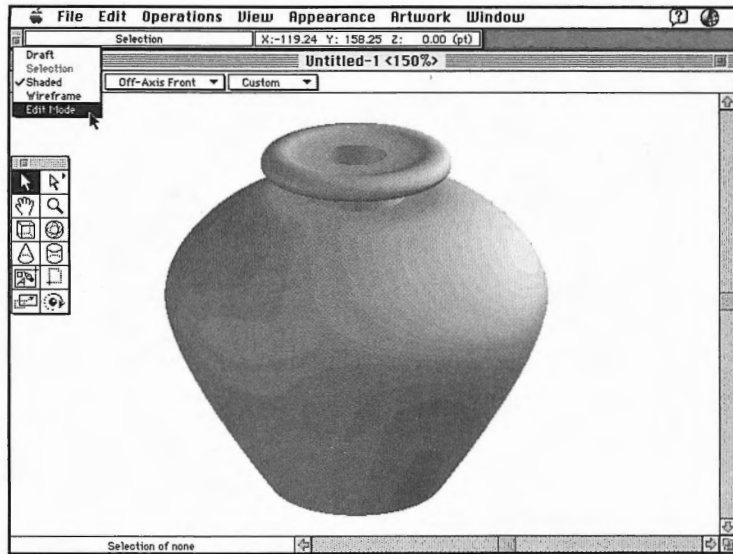
7. In the Perspective pop-up menu at the top of the work window, set the perspective to “normal” and the view angle to Off-Axis Front; this tilts your point of view slightly above and to one side of the object. In the View Mode pop-up menu, switch to Shaded.

This renders the vase—and may take a minute. In this example, the opening of the vase is very constricted and not very attractive (see figure 4.7).

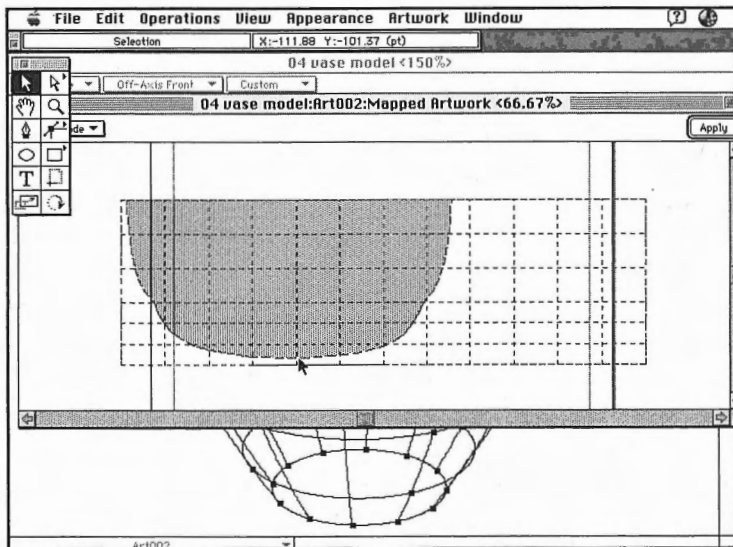


**Figure 4.7** Setting the perspective to “normal,” the view to “off-axis front,” and the shading mode to “shaded”

8. Switch back to Edit mode in the View Mode pop-up, then choose Revolve from the Operations menu. This dialog has an Edit Profile button; click it to go back into the 2-D Artwork mode and adjust the outline shape of the neck of the vase. When you click Apply in the Artwork window, the vase will be re-revolved with all of your other changes intact (see figure 4.8).
9. To use an Adobe Illustrator document as a texture for the vase, you’ll first have to choose a single surface for the vase. Do this by repeatedly pressing the Tab key until the main body of the vase is selected.
10. Select the Map Artwork command from the Artwork menu.
11. The Mapped Artwork window shows the selected surface “unwrapped.” The parts of the model currently visible in the 3-D view are shown in white on the grid; the parts that are hidden are shown in gray (see figure 4.9).

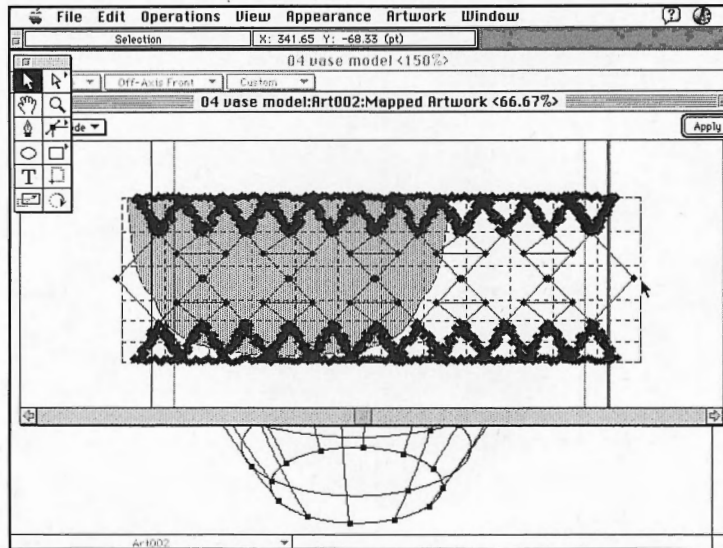


**Figure 4.8** Choose Edit Profile and fine-tune the shape



**Figure 4.9** The model's surface unfolded

12. Import the Illustrator artwork file. Use the resize tool in the tool palette to stretch the artwork until it covers the entire shape, then click the Apply button and switch back to the 3-D window (see figure 4.10).



**Figure 4.10** Mapping artwork onto the shape

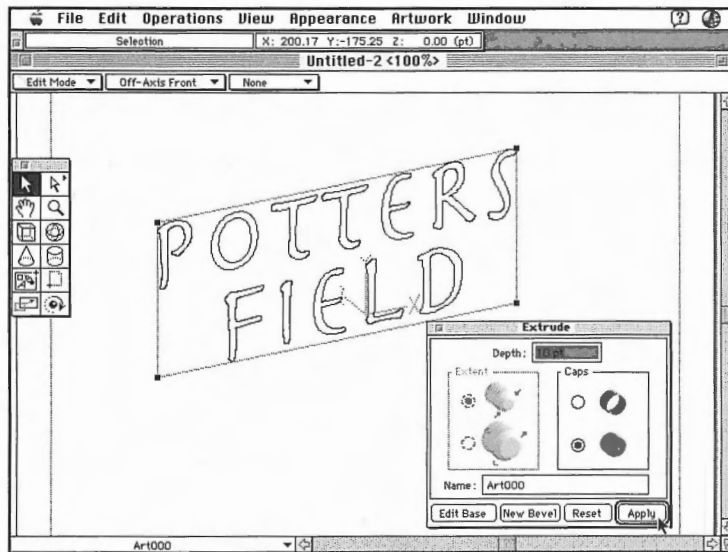
13. In the View Mode menu, select Shaded to see the final, rendered vase. Now save the vase, but also use the Export command to save it in Adobe Illustrator 5 format (see figure 4.11).



**Figure 4.11** Rendering and saving the vase

The next step in this project is to create a 3-D type logo. The process is very much like creating the vase, except that the type is created with the type tool instead of hand drawn, and it's extruded instead of revolved.

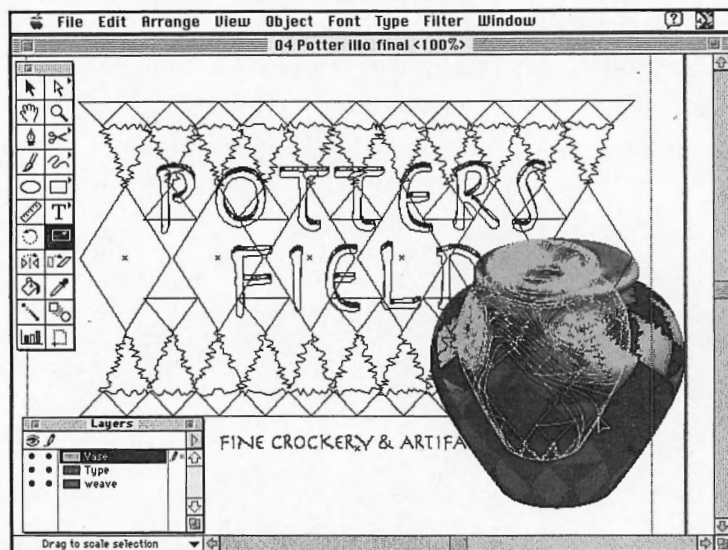
1. Create a new document in Dimensions and select the 2-D artwork tool.
2. Select the Type tool. Set the type size to 24, and choose a typeface (this example uses Adobe's Herculenum Type 1 font). Type in the words "Potters" and "Field" separately and align them one above the other. When you're finished, click the Apply button to bring you back into 3-D space.
3. Select Extrude from the Operations menu, set a depth of 10 points with no bevel (for a font this thin) (see figure 4.12). The type is extruded into 3-D. Set the angle of view and render the logo just as you did with the vase (no Artwork texture is used here). Then save the logo in Illustrator format.



**Figure 4.12** Create an extruded type logo using roughly the same series of steps used for the vase

4. In Illustrator 5 you can put art elements in their own layers. In Illustrator, open the artwork previously used as a texture on the vase.
5. Modify the artwork to give it a subtle background color, and add a border.
6. Next, using Illustrator, open the Potters Field art you created in Dimensions and Copy and Paste it into the new document. Use the scale tool to size it down to the approximate dimensions shown here.

7. Repeat the process, opening the vase file and copying and pasting the vase into the scene. As with the background art, these each can be placed on their own layers to keep things organized. For a very pleasing effect, you can use the eye dropper tool to grab colors out of the 3-D surfaces and drop them into the background elements.



**Figure 4.13** Importing the 3-D renderings into Illustrator and combining them with 2-D art for a finished project

8. The finished art has all of the qualities of a PostScript illustration, including scalability, printing device independence, and very fine control over printing details. You can even ungroup the 3-D elements and edit the parts of their blends as individual shapes.

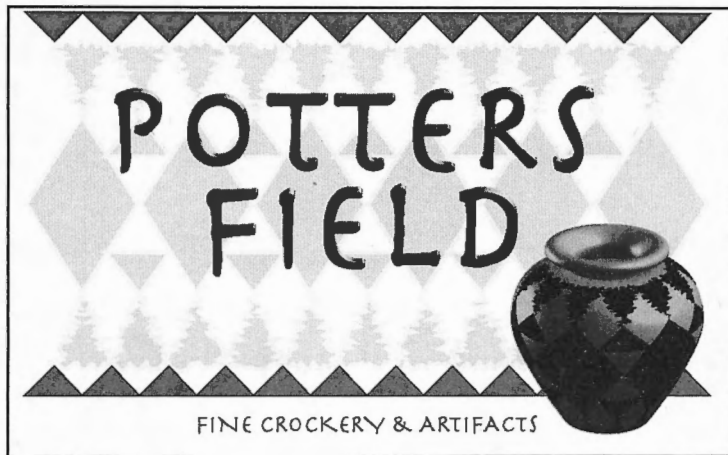
## RayDream's addDepth

Like Dimensions, addDepth can extrude existing PostScript art into 3-D. You cannot lathe objects in addDepth, so you are limited to extruding curves if you want curved surfaces. For example, if you want to create a cylinder, you must start with a circle that you would extrude upwards. If you want a sphere, however, you are out of luck.

addDepth has its own drawing tools, which means you need not use Illustrator or another drawing program as a front end, although it also supports Illustrator and FreeHand file formats. To use type (it supports both PostScript and TrueType),



you simply click the image window with the Type tool and enter the desired text. You can kern text and make other adjustments directly within the Type window.



**Figure 4.14** *The finished sign*

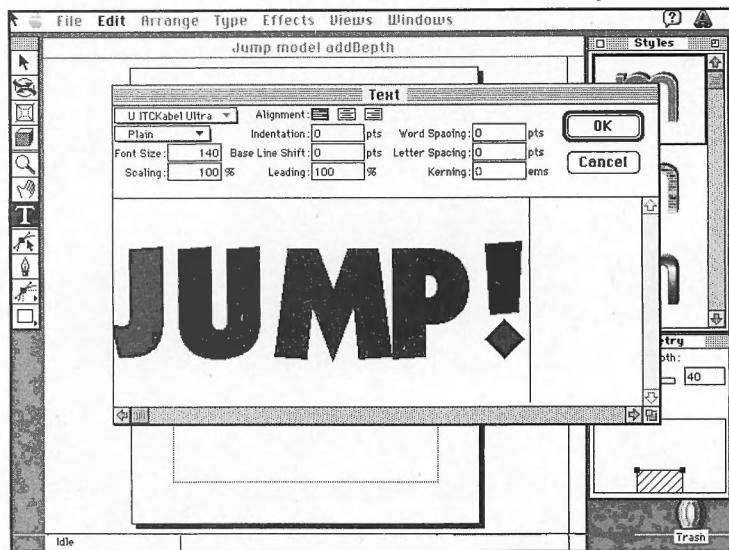
RayDream's addDepth supports System 7's Publish and Subscribe. You can create and Publish an object in one PostScript application (a chart, in Delta Graph Professional 3.5, for example) and Subscribe to it from addDepth. This object can be extruded into 3-D with all of addDepth's formatting tools. When you Publish changes to the original document, the 3-D object in addDepth is updated automatically. With Microsoft Excel, DeltaGraph, and addDepth, you can set up a three-way information exchange that will automatically create a new beveled 3-D pie chart rotated into perspective each time you change the Excel spreadsheet.

The most popular use of addDepth will undoubtedly be extruding type for use in other programs. The program offers very good control over the color and surface attributes of many of the separate elements of 3-D objects. For instance, it's easy to have a gradient fill on the face of an object, yellow on the bevels, and blue on the sides.

### **Creating a "Jump!" Head**

Creating a catchy headline of a paratrooper doing his thing required the use of Aldus FreeHand and addDepth.

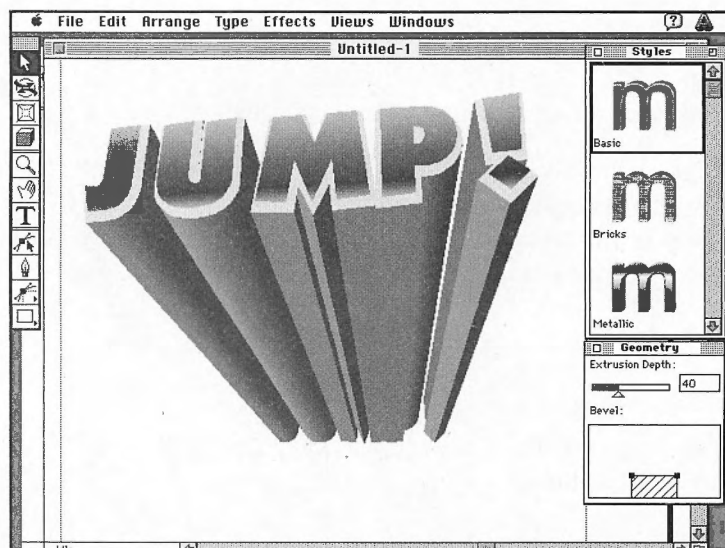
1. Enter the text "JUMP!" in addDepth's type dialog box, where you will also find kerning and other standard type controls (see figure 4.15).



**Figure 4.15** *Entering text in addDepth's Type dialog box*

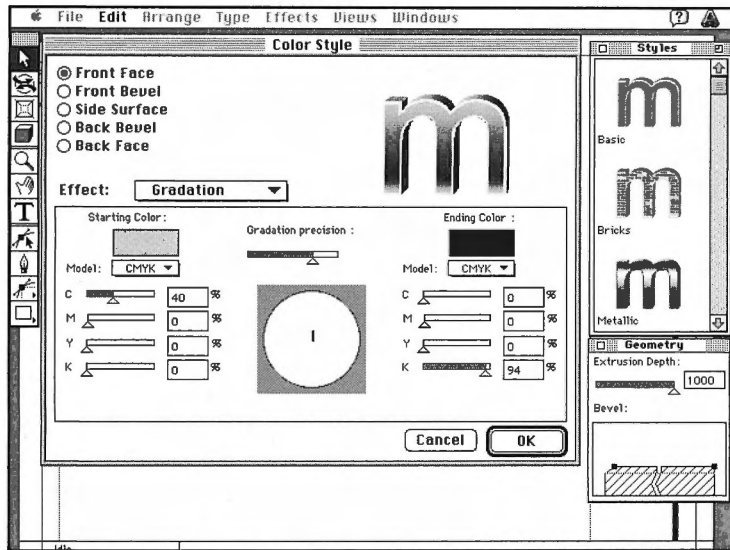
2. Extrude the resulting type using the small palette found at the lower-right corner of your window (see figure 4.16).

A bevel for the edges of the objects can be set in this window as well.



**Figure 4.16** *Setting the depth of extrusion and the size of the bevels using the Geometry floating palette*

3. After extruding the type, rotate the 3-D type into the proper perspective using addDepth's virtual trackball. You also can increase the view angle to give the object more of a sense of depth.
4. Using the Color Style dialog box, add a gradient texture to one of the default surface settings and apply it to the face of the 3-D type (see figure 4.17).



**Figure 4.17** addDepth's Color Style dialog box

5. addDepth automatically renders objects as you work. Once the rendering of your image is complete, the last step is to export the image in FreeHand format. Once you've exported the 3-D rendering in FreeHand, you can embellish it with FreeHand's drawing tools, as with this paratrooper (see figure 4.18).



**Figure 4.18** The final image in FreeHand

## Working with Type

Extruded, beveled, engraved, raised, and animated 3-D type treatments are the most popular uses of 3-D software. A number of vendors have responded to the demand by creating applications specifically for this purpose. Aside from the 3-D PostScript programs previously described, there are a number of applications that create true, exportable 3-D geometry as well as render high-quality bitmapped images using all the tricks of 3-D. These include Pixar's Typestry, Strata's StrataType 3d, and Specular's LogoMotion.

Although you can use most TrueType and Type 1 PostScript fonts with these programs, not all fonts are ideal for 3-D use. Medium-weight sans serif fonts present the fewest problems, while serif fonts with some weight to them also work well. In general, avoid waifish, fine-lined fonts; these are often too thin to handle any beveling, as the bevels will tend to overlap one another. Thin fonts work best when extruded only slightly for embossing and "foil" type, or when extruded without bevels. On the other hand, avoid extremely heavy fonts with small openings. The holes will tend to fill in when you bevel the type.

### Tip

If you have TrueType fonts and your program supports only PostScript (or vice versa), you can use programs such as FontMonger or Metamorphosis to convert one type of font to another. QuickDraw GX introduced new fonts to the Mac. These offer many advantages, such as advanced scalability. Pixar's Typestry was the first 3-D program to support this new format.

## Pixar's Typestry

Pixar's Typestry was originally devoted to creating 3-D type images and animations. Its Type tool enables you to enter a text string, specify a font, and set extrusion parameters. Its greatest strength is its reliance on RenderMan (which is built into Typestry) for rendering. This also enables you to use it with Pixar's MacRenderMan or its distributed rendering system, NetRenderMan, or a RISC-accelerated version, such as YarcRenderMan. You can also use RenderMan's creative and unusual shaders.

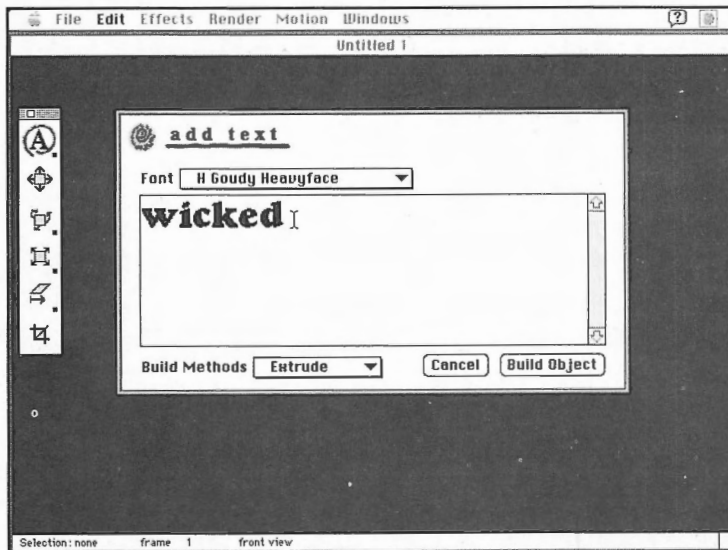
Typestry has a great variety of special effects possibilities because of its support for RenderMan shaders in the form of Looks. Unlike bitmapped textures, these are unique procedural shaders. For instance, the following example makes use of the Tropical Water shader which contains phosphorescence, among other things.

Procedural shaders differ from surface texture bitmaps used by many 3-D programs in that they are algorithms that mathematically define the appearance of a shape. RenderMan procedurals are sometimes *volumetric*, which means that they “fill” a shape, rather than just coat its surface. This means you can create realistic clouds, for example. Procedural shaders can sometimes do things like *displace* or deform a surface.

### Wicked Water

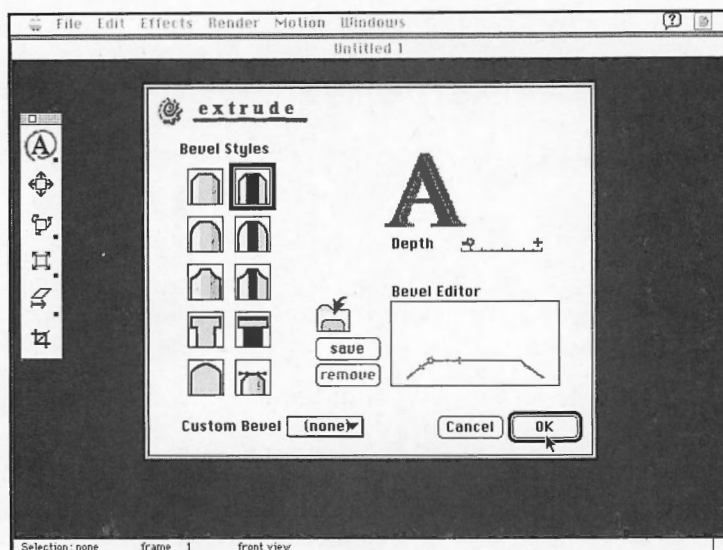
This example uses Typestry's type extrusion, bevel, and Looks editing capabilities for creating a 3-D type logo.

1. Begin by clicking the work window with the text tool. Enter the type “wicked” and set the font (see figure 4.19).

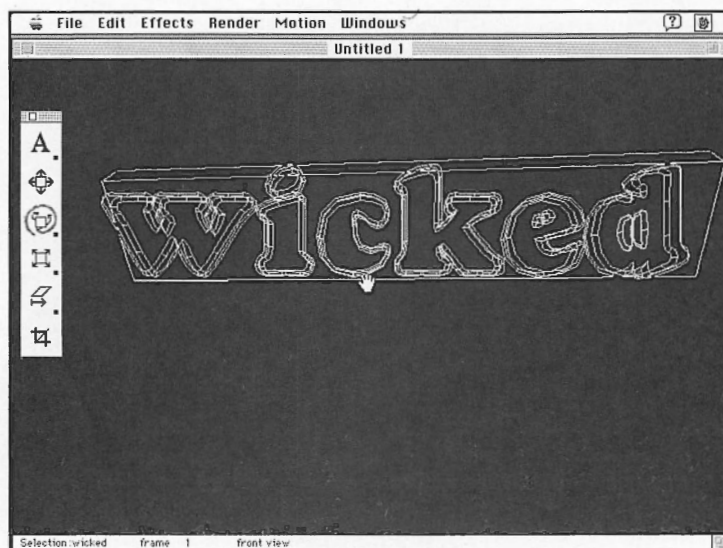


**Figure 4.19** Entering a line of text to extrude

2. Using Typestry's bevel tool, set a standard bevel. The two-color bevels indicate that you'll be able to give different textures to the faces and sides of objects (see figure 4.20).
3. Using Typestry's virtual trackball rotation tool, move your logo into a good perspective (see figure 4.21).



**Figure 4.20** Applying a bevel

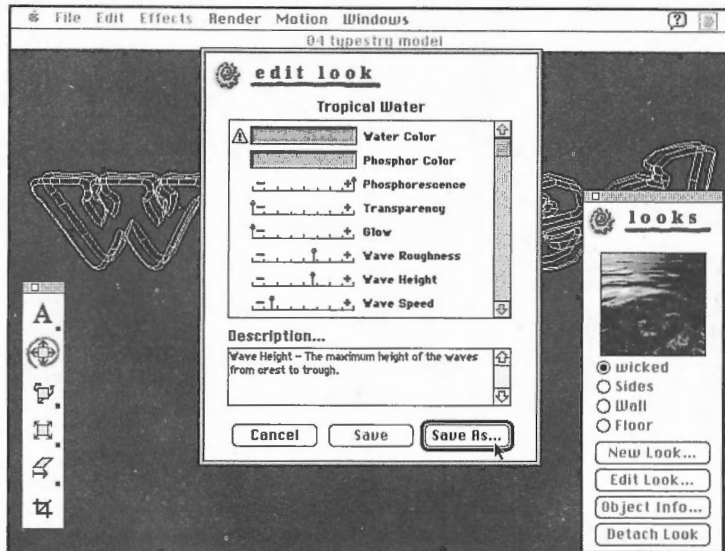


**Figure 4.21** Rotating the 3-D type into perspective

4. To apply a texture, open the Looks palette and assign one.

Note that you can apply separate textures to the faces and sides of the object (see figure 4.22). In this case, a Look called Tropical Water is applied. The Edit

Look button brings up this editing interface where you can adjust the parameters of the Look. With RenderMan products, you can often edit the *displacement* values of a texture, among other things.



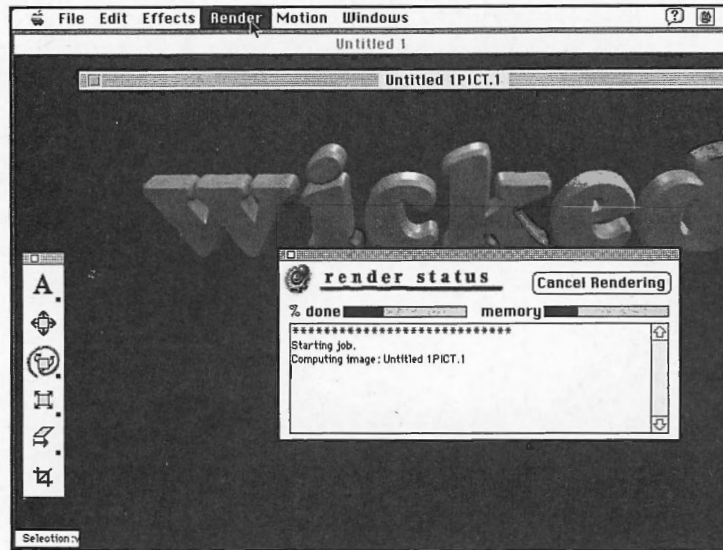
**Figure 4.22** The Looks palette and Edit Look interface

5. Generate a test render with the Quick and Dirty option in the edit menu to make sure the lighting and textures are set appropriately. Use the Render to Screen option—otherwise the image will be written to disk rather than displayed onscreen.

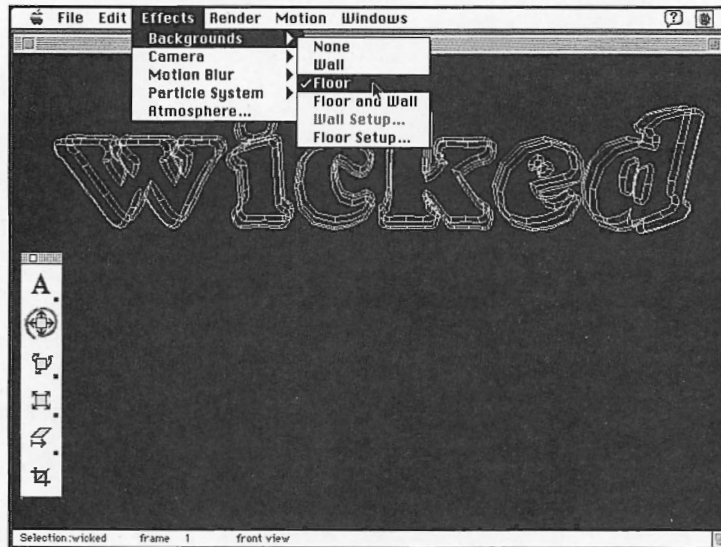
Note that the Render Status monitor used by Typestry is very similar in appearance and function to the Print Monitor in Apple's system software. Note that PrintMonitor is no longer used in System 7.5 and later.

6. To help complete the appearance of the scene, add a floor with the Add Floor option in the Backgrounds menu (see figure 4.24). You can customize the positioning of the floor in the Floor Setup dialog box.
7. Now generate a final rendering using the Reasonable or Excellent & Slow option, depending on how long you're willing to wait. On a PowerMac, this scene took about five minutes to render (see figure 4.25).



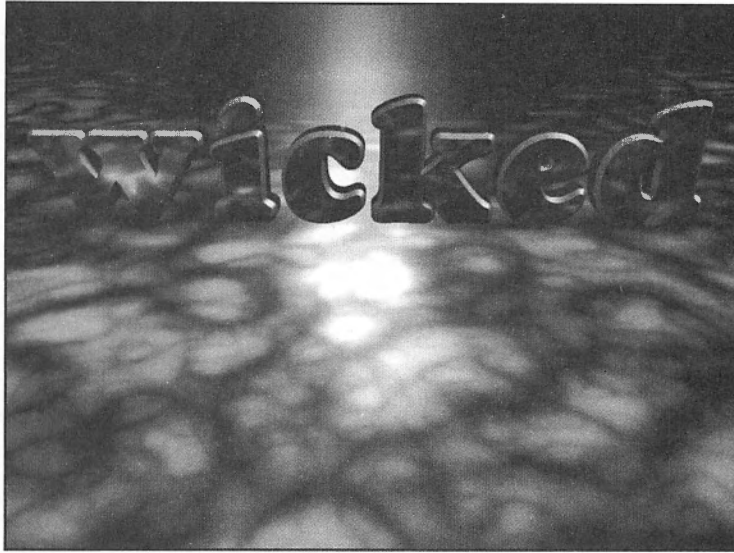


**Figure 4.23** Generating a test render



**Figure 4.24** Adding a floor to the scene





*Figure 4.25 The final rendering*

## **StrataType 3d**

StrataType 3d, from Strata Inc., works very much like Typestry, although it does not support animation. However, it is a good tool for creatively extruding and rendering type.

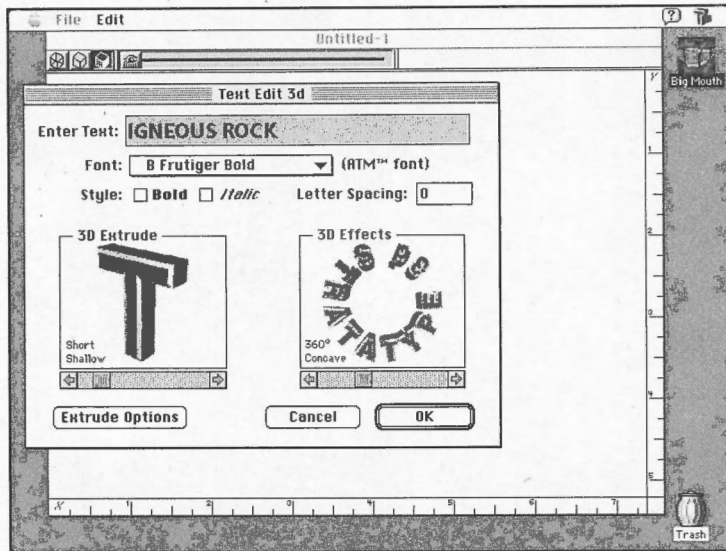
StrataType 3d offers a variety of preconfigured text layouts, extensive kerning, and other type controls. It has very good control over automatic beveling and it enables you to make your own bevels and textures. You can apply separate textures to different surfaces of an object. You can even apply chrome to the front, brass to the bevels, and granite to the sides.

Objects can be rendered with an alpha channel or PICT background for seamless compositing with other images.

## **Creating an Igneous Rock Logo**

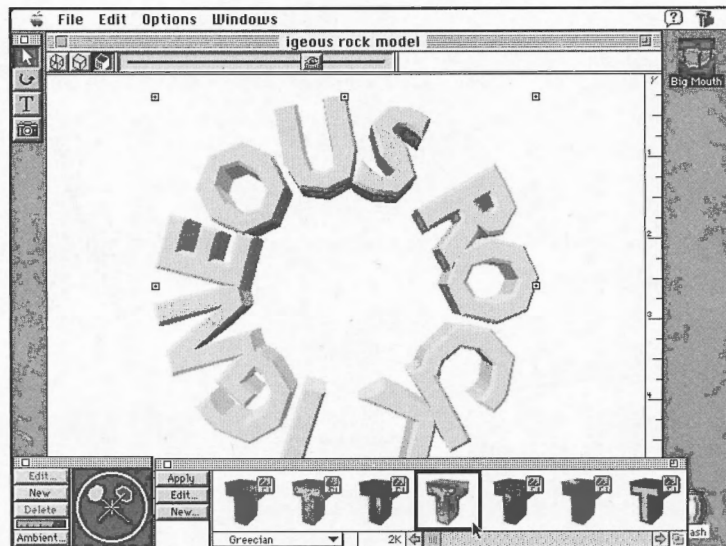
The following steps describe how you can create a cool logo in StrataType for later compositing in Adobe Photoshop.

1. Begin by entering text in StrataType 3d's Text Edit window. Select a circular type path from StrataType's library of prebuilt configurations, and a standard extrusion-and-bevel combination (see figure 4.26).



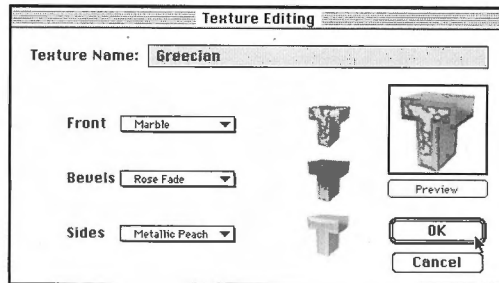
**Figure 4.26** Selecting a combination of extrusion depth and 3-D bevel options from Strata Type's library

2. If necessary, kern the letters to give them a pleasing distribution.
3. Adjust the camera's angle of view to wide angle using the slider at the top of the screen (see figure 4.27).



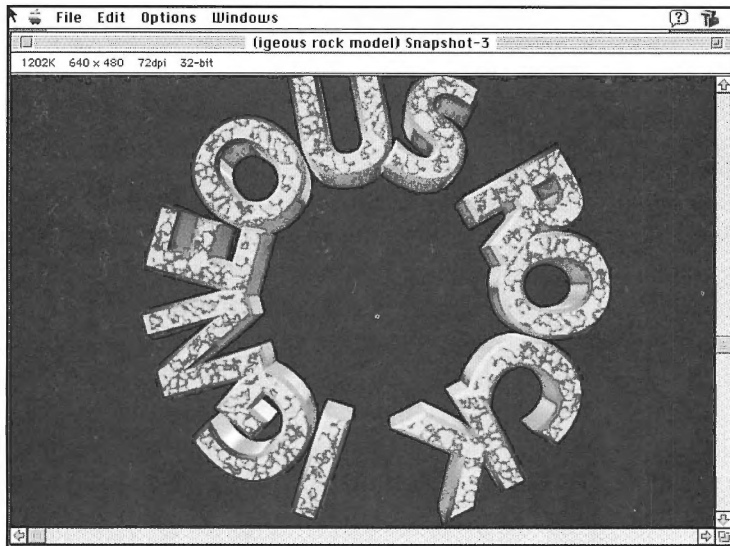
**Figure 4.27** Adjusting the camera's angle of view

4. Use Strata's Texture Editing dialog box to apply a "rock" texture with metallic sides to the object (see figure 4.28).



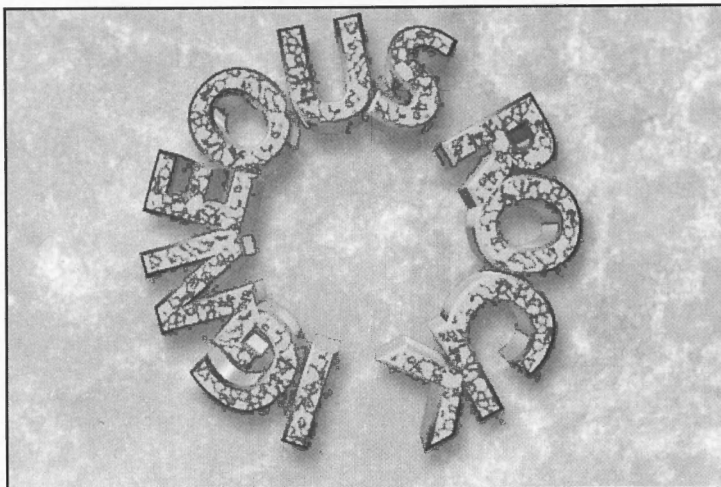
**Figure 4.28** Applying a texture to the object

5. Render the image, making sure the "alpha channel" is turned on in the render dialog. The "black" background will be transparent when composited in Photoshop (see figure 4.29).



**Figure 4.29** The "black" background is actually "transparent"

6. Save the finished rendering and open it in Photoshop. Choose the “Load Selection” menu command. Then choose Copy to render the object to the clipboard.
7. Open a background image and choose Paste. The result is shown in figure 4.30.



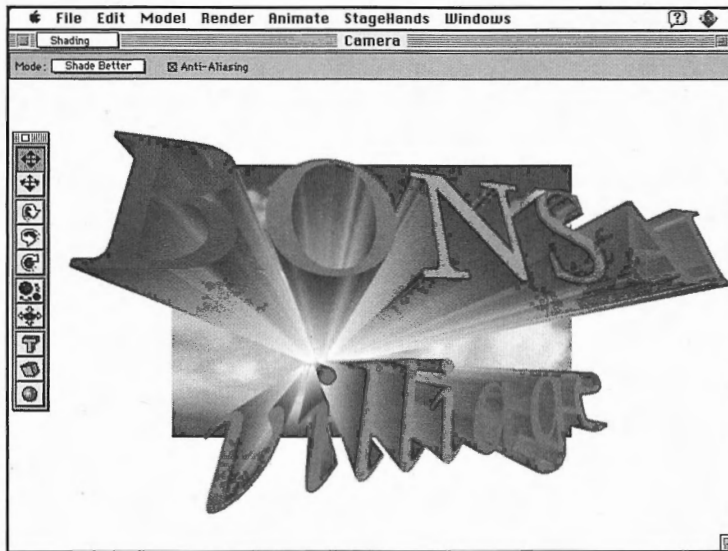
*Figure 4.30 The finished composited image*

## LogoMotion

LogoMotion is Specular International Inc.'s low-cost 3-D modeling, rendering, and animation tool. In fact, it's really a scaled-down version of Specular's high-end animation product, Infini-D. While many of LogoMotion's features are geared toward animation, it makes a very good entry-level 3-D product for those who need to extrude and render type as well. The program features Gouraud shading to achieve smooth renderings, but textures in the rendering are not nearly as detailed as Typestry's RenderMan or StrataType's Phong rendering (see figure 4.31). One advantage of LogoMotion, however, is that it's really fast, even on 680X0 Macs. For users interested in exploring 3-D animation, it is in many ways easier to use than Typestry, with significantly more control.

## Engraved Type

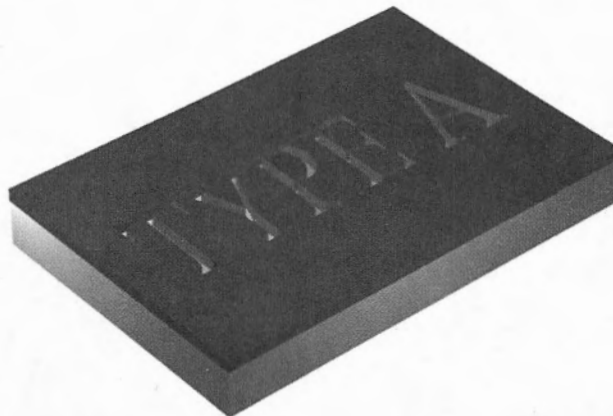
One way to create realistic engraved type (as in a tombstone or statue) is to use boolean subtraction, such as that shown in the Typestry examples.



**Figure 4.31** LogoMotion includes automatic environment maps and fog effects, which both contribute to realistic scenes

Solid modelers, meanwhile, can actually subtract one object from another, so that you can remove the intersection between an object and solid type that is embedded in the surface. form•Z, for example, has a title tool that enables you to make solid type that can be subtracted from other surfaces.

To create engraved type with a standard surface modeler that lacks booleans, you can extrude a plane that uses type for “holes” and sandwich it on top of another plane of the same size without the holes. The result looks a lot like a rubber stamp (see figure 4.32).



**Figure 4.32** A sandwich of extruded blocks creates an engraved rubber stamp

## Bump-mapped Type

Bump mapping is an extremely quick and easy way to simulate embossed type. You simply create a bump-map image containing black type and set the renderer to use black as raised areas. Light will then catch the edges of the type when you render. The same approach can be used to affect stamped or engraved type: simply switch the bump map to treat black as depressions. (For a thorough explanation, see “Bump maps” in chapter 7, “Materials.”)

## Embossed Type

To achieve a really spectacular type treatment, you can import type and give it a bright surface, extrude it minutely, and place it on another surface of duller appearance. This technique can be used to create chrome lettering on a black object, for example. Alias' Sketch! makes this extremely simple by enabling you to create surfaces from text on top of other surfaces.

A similar effect can be created in programs such as Specular's Infini-D, StrataStudio Pro, and Sculpt by using reflection mapping (which is actually much more efficient than modeling the type). By using white-on-black type as a reflection map, the white of the type will cause the surface to reflect the surroundings, but where the reflection map is black, it will not (see chapter 7, “Materials”).

## Animated Type

Animated 3-D type is a favorite use for 3-D graphics. Three-D software enables you to create a wealth of compelling 3-D video logos and titles quickly. While animation is covered in detail later in chapter 9, “Animation,” animated type deserves special mention. Most of the applications that support 3-D animation offer some means to extrude and animate type. Multimedia producers will enjoy a chance to put animated type in their QuickTime and interactive projects.

As mentioned earlier, bevels are important in animating type, particularly since they tend to outline extruded type, making it easier to read. Bevels also will catch and reflect glimmers of light, making the scene shine.

### Tip

When creating beveled type, use small, simple bevels for fonts with narrow strokes, as large bevels will tend to overlap and cause problems—particularly at sharply-angled corners. Heavyweight fonts can handle larger, fancier bevels.

**C h a p t e r**

**5**

# **Modeling**



**T**hree-D begins with the model. Just as kids build airplanes from kits to hang from their ceilings, computer-based modeling is a process of putting together many small parts. Essentially, it's not that different from the way many manufactured goods are produced. For example, take an everyday object—a standard pencil. Every school child knows that a pencil is made of five parts: a “lead,” two halves of wood, a metal ring, and an eraser. While it may be possible to form a vaguely pencil-like object in one stroke, to create the real thing, one has to take at least one step for every part. The same is true for 3-D modeling. While 3-D sometimes makes it possible to create the effect of several objects at one time, the easiest approach is additive modeling: putting lots of small parts together to form a whole.

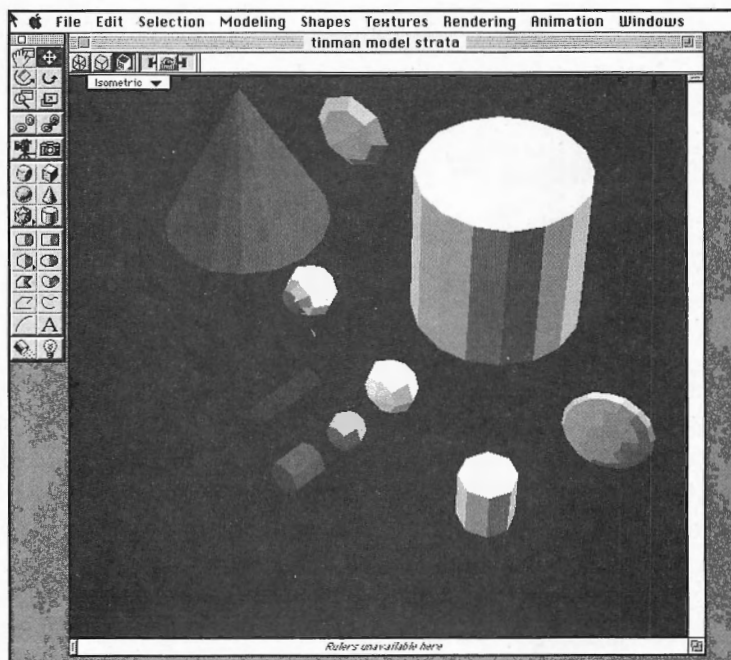
## Additive Modeling

In its simplest form, additive modeling begins with primitives—geometric shapes with very simple mathematical descriptions. Many programs make it so easy to create primitives that all you have to do is point the mouse to where you want a cone, for example, and click. Putting the pieces together merely involves moving them into position. For example, the following figures show a model built entirely from primitive objects—primarily cubes, spheres, and cylinders put together with simple additive modeling (see figure 5.1 and 5.2).

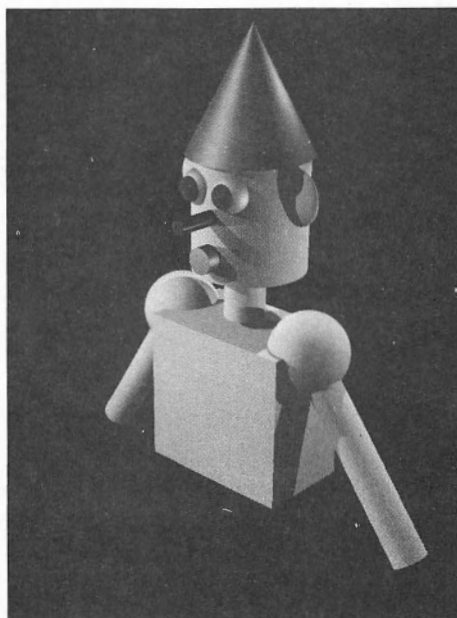
## Joining Objects

Since much 3-D model building is intended to produce a representational model, not a manufacturing-accurate design, model building is usually a fairly simple process. 3-D programs are smart about bonding objects together. Imagine attaching one of those little knobs to the top of a flagpole. In the real world, you would have to drill a hole in the knob that would exactly match the end of the flagpole, then shimmy up the pole to join the two together with screw threads or a welded seam. In 3-D, you simply put the ball on the pole; no joints are required. In fact, no glue or anti-gravity devices are required either, since there are no forces in the virtual world of 3-D to pull things apart, at least not yet. (This assumes, of course, that the hole in the ball and the threads on the pole aren't important to your design. If they are, you will want to model them.) When eyeball measurements aren't up to the task, snaps and grids help you to align objects. More elaborate and expensive modelers offer parametric inputs, that is, the capability to accept typed-in dimensions for the size, shape, and orientation of objects. Computer-aided design (CAD) systems are heavily geared towards very precise parametric modeling, which is one of the essential differences between basic modelers and full-fledged CAD. The differences are in the details.





**Figure 5.1** Model building often begins with primitives, such as cones, spheres, blocks, and cylinders



**Figure 5.2** A 3-D "stick figure" made out of the pieces in figure 5.1

Some animation programs, including Swivel 3D, Electric Image Animation System, and Shade III, enable you to constrain relationships of objects using movable joints.

## Constructive Modeling

While additive modeling is used to put parts together, it often fails to address a simple problem: what if you don't have the parts you need? Three-D software has its own techniques, analogous to those used in the real world, which can be applied to building your own components. These techniques are known as *surface* and *solid* modeling, but together, they are sometimes called *constructive* modeling.

Surface modeling uses 2-D outlines to form new 3-dimensional shapes. For example, a square can be extruded to form a post, or the outline of a vase can be lathed to form the round vase itself. Lathing and extrusion are the basis of constructive modeling. There are several advanced variations of each, and even hybrids, such as *lofting* and *sweeping*, that combine the two. Solid modeling takes this a step further by enabling you to use these shapes you've created to carve into one another or to fuse multiple shapes together.

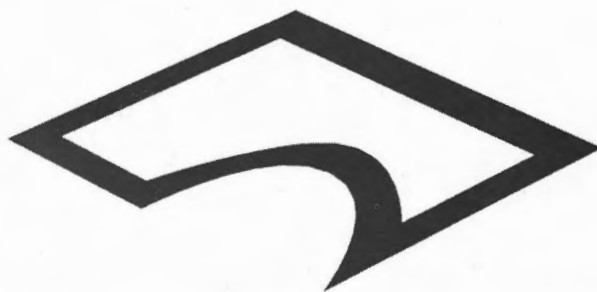
The terms solid and surface are potentially confusing because in surface modelers things still *look* solid. The basic difference is that surface modelers enable you to work only with the hollow shells of objects, while solid modelers enable you to carve into the interior surfaces.

### Extrusion

Extrusion is one of the simplest methods for creating 3-D objects. You begin with a 2-D outline, either drawn or imported, and add a third dimension. Basic extrusion is perpendicular to the plane of your 2-D object, just like a cookie cutter. If the outline of the cookie cutter (a 2-D shape) is the shape being extruded, then the cookie (the 3-D object) is the extrusion. When extruding an object, there is no practical limit to the depth of the cookie dough, so you could extrude a circle into a long straw or a thin disk.

If you extrude a surface, rather than a simple outline, you will end up with a *capped* object. (An extruded circular outline creates a pipe, while an extruded solid circle creates a solid column.) The caps on the end are often created as an option. Some 3-D programs enable you to make changes to the end caps without affecting the extruded "walls."

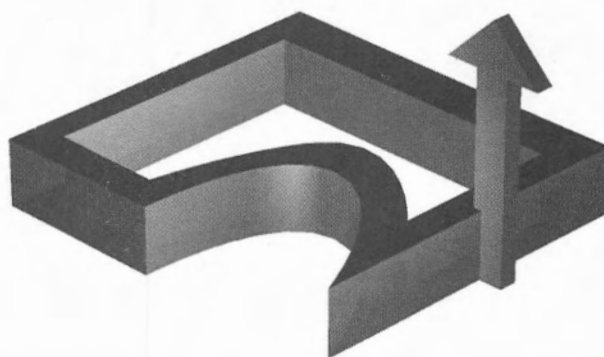
Variations of extrusion include extrusion along a path, extrusion with bevels and swept extrusion along a specific curve.



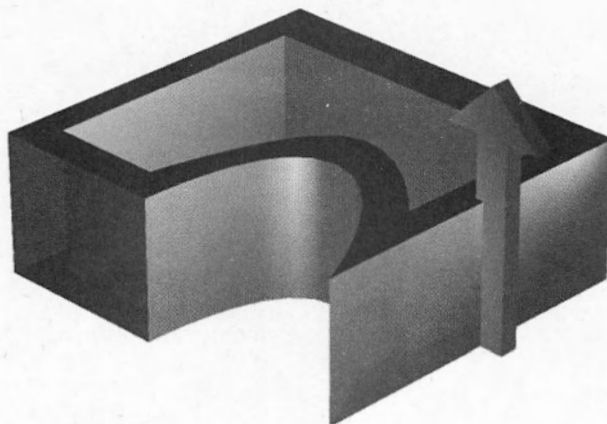
**Figure 5.3** *The 2-D outline to be extruded perpendicular to the working surface*



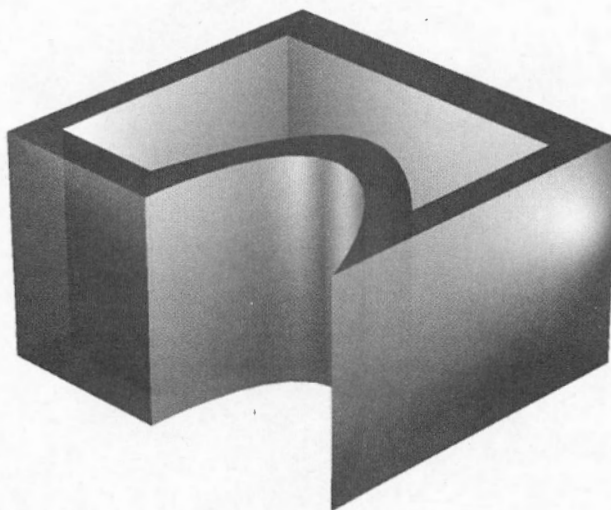
**Figure 5.4** *The outline will be extruded in the direction of the arrow*



**Figure 5.5** *The extrusion begins...*



**Figure 5.6** *Continuing the extrusion...*



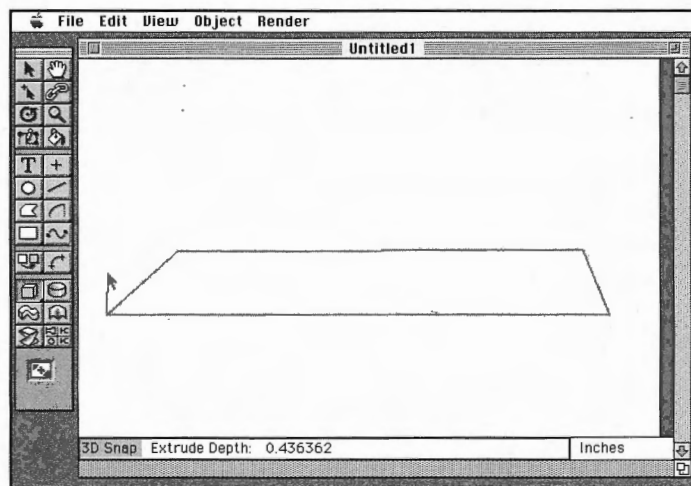
**Figure 5.7** *The finished extruded shape*

## **Basic Extrusions**

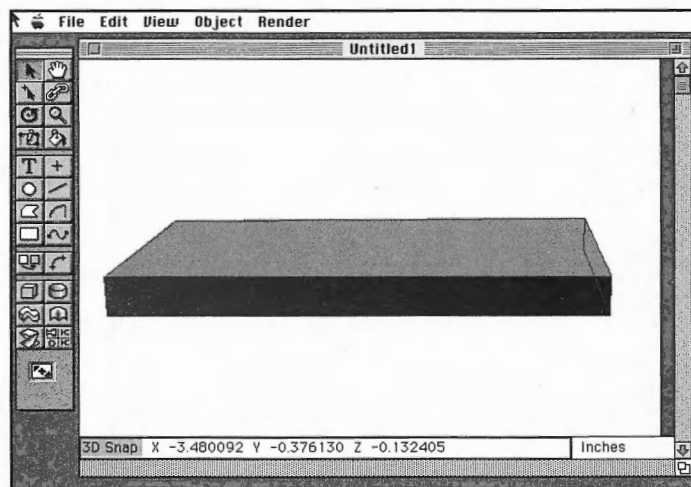
Basic extrusion is the simplest form of turning 2-D shapes into 3-D. Most 3-D programs enable you to draw your own starting profiles, but some also enable (or require) you to import outlines, such as those created in Adobe Illustrator.

## Slabs and Blocks

A block is the simplest 3-D shape. It is made by extruding a rectangular shape perpendicular its surface. If you extrude the shape at an angle other than perpendicular to its surface, you can create a trapezoidal solid. Any rectangular solid shape can be made the same way (see figures 5.8 and 5.9).



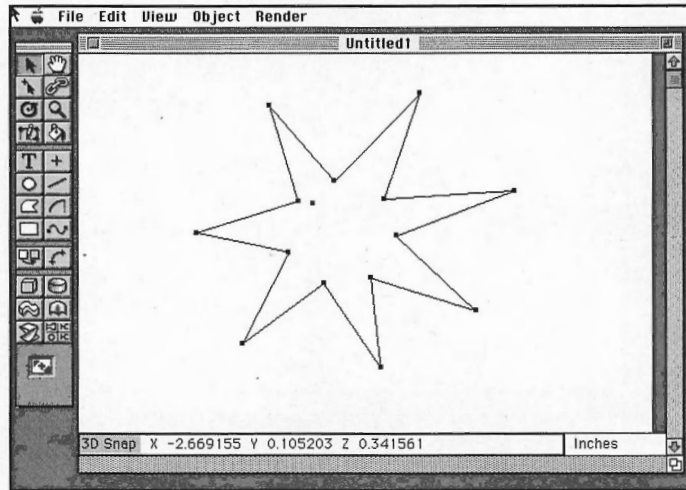
**Figure 5.8** *Preparing to extrude a slab*



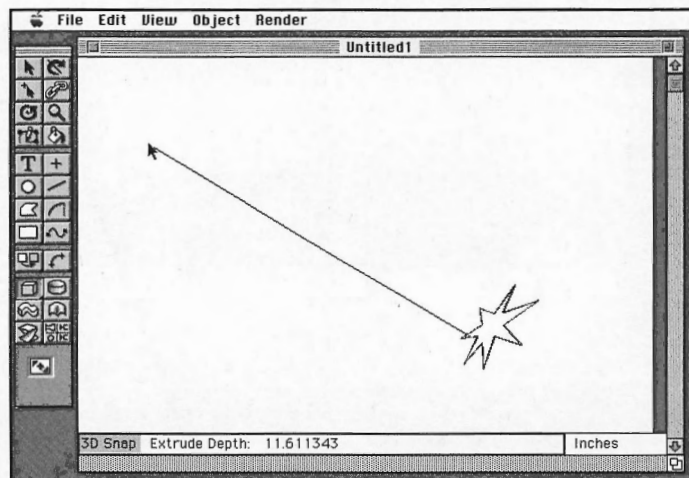
**Figure 5.9** *The extruded slab*

## A Star-Shaped Tube

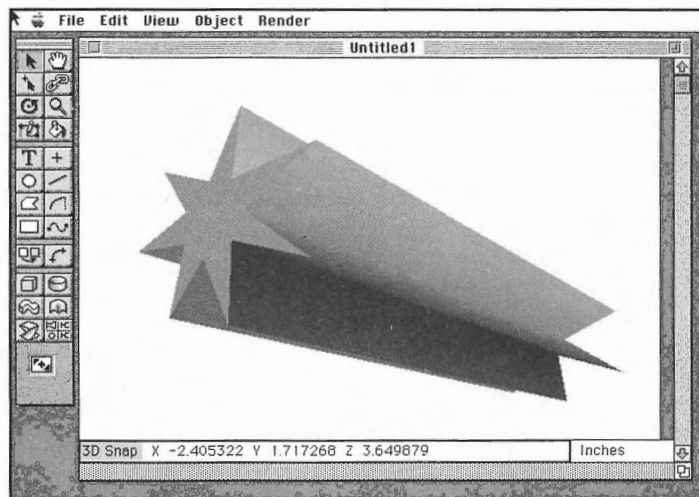
Irregularly-shaped extrusion begins with the cross sectional shape, in this case, a star shape extruded in MacroModel (see figure 5.10). The shape is extruded along the designated path. This often can be accomplished numerically, for precision, or visually (see figure 5.11). The resulting shape has a consistent star-shaped cross section (see figure 5.12).



**Figure 5.10** The 2-D star shape



**Figure 5.11** Visually defining the extrusion path



**Figure 5.12** *The extruded star*

## Caps

As mentioned earlier, it's possible to extrude shapes without end caps to create hollow surface tubes from a profile. Typical examples are extruding a circle to form a hollow pipe, or drinking straw.

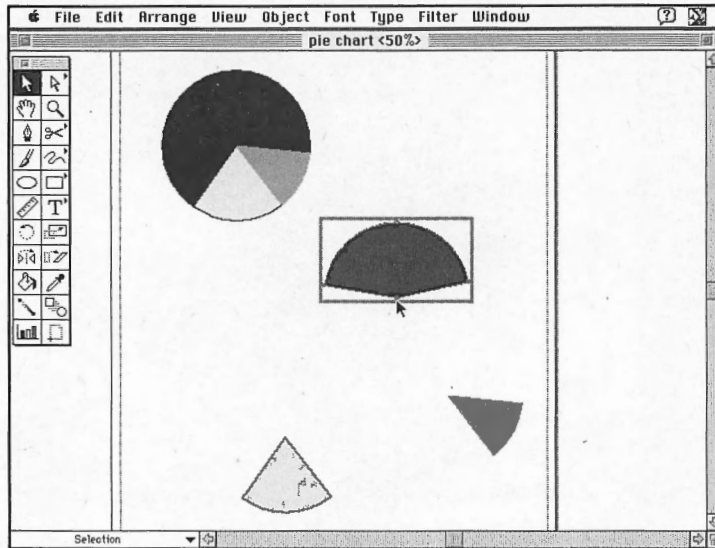
## Subscribing to an Illustrator Outline

While many programs enable you to import a finished 2-D drawing from a vector-based illustration program as a starting point for building a 3-D model, Specular's Infini-D uses a novel approach to creating an extrusion from an outline generated in Adobe Illustrator—System 7's Publish and Subscribe. To use this method effectively, you have to be running System 7 and have both Illustrator and Infini-D.

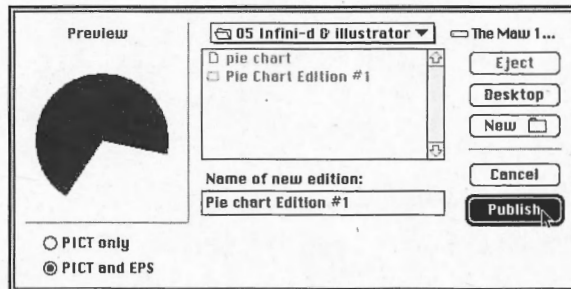
Presentation graphics can make extensive use of compelling 3-D imagery. This example shows how to create a 3-D pie chart from original art created in Adobe Illustrator (this also will work with charts created in DeltaGraph professional).

1. Begin by drawing a pie chart in Illustrator. Create copies of each of the wedges and move them far enough apart that each can have its own bounding box discrete from the others (see figure 5.13).
2. Select one of the wedges and select the Create Publisher menu option. Publish the wedge shape as "widgets" or another suitable file name (see figure 5.14).





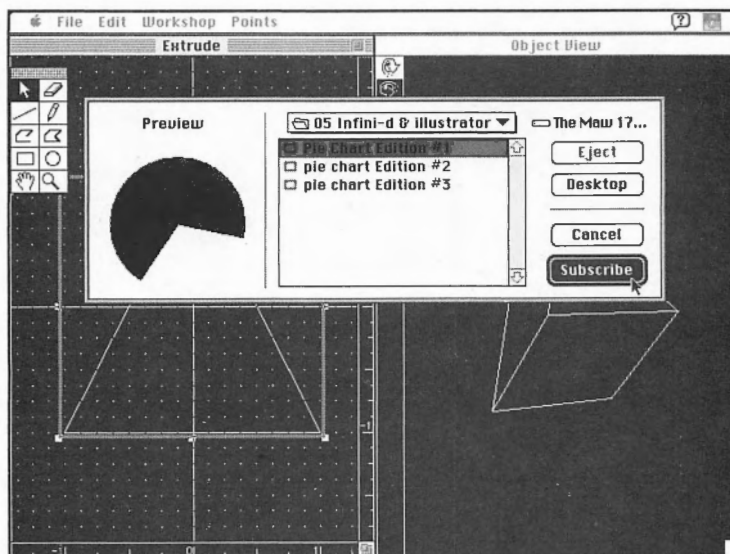
**Figure 5.13** Pie chart drawn in Illustrator and divided into parts



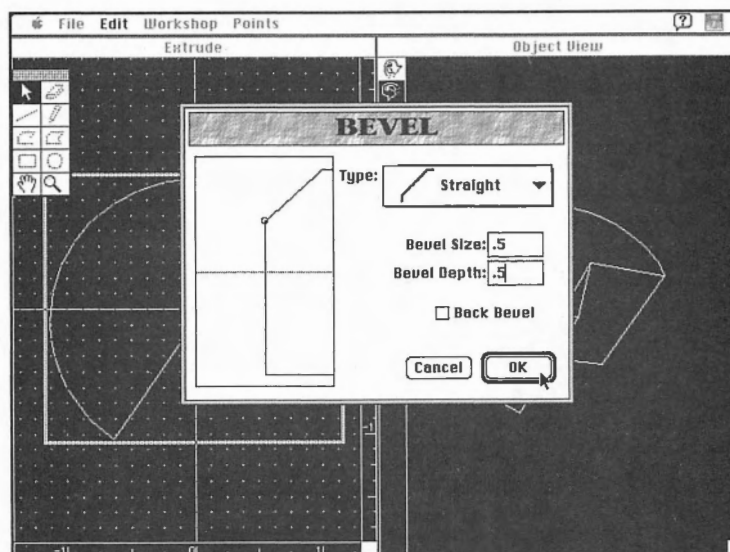
**Figure 5.14** Publish the “widgets” wedge

3. In Infini-D, place an extrude object in the scene and double-click to open the Extrusion Workshop. Use the Subscribe... menu option to Subscribe to the “pie chart edition #1” pie piece (see figure 5.15).
4. Use the Bevels... dialog box to apply a bevel to the shape, then click OK and exit the Extrusion Workshop (see figure 5.16).
5. Add two more extrusion shapes to the scene. Repeat the previous steps for each, subscribing to “Pie chart edition #2” and “Pie chart edition #3” respectively. Apply different textures to each of the objects, then move the extruded pie pieces into position. Place an infinite plane as a floor and give it a texture. Then set the Shading mode to Shade Best (see figure 5.17).

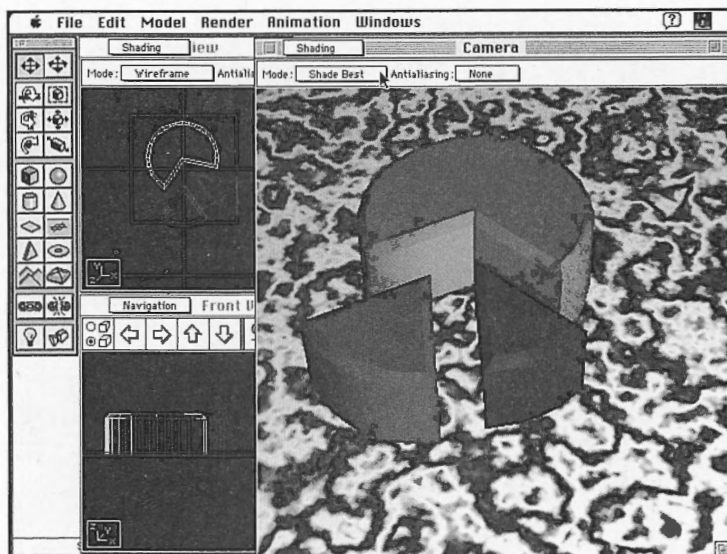




**Figure 5.15** *Subscribe to the pie shape in Infini-D*

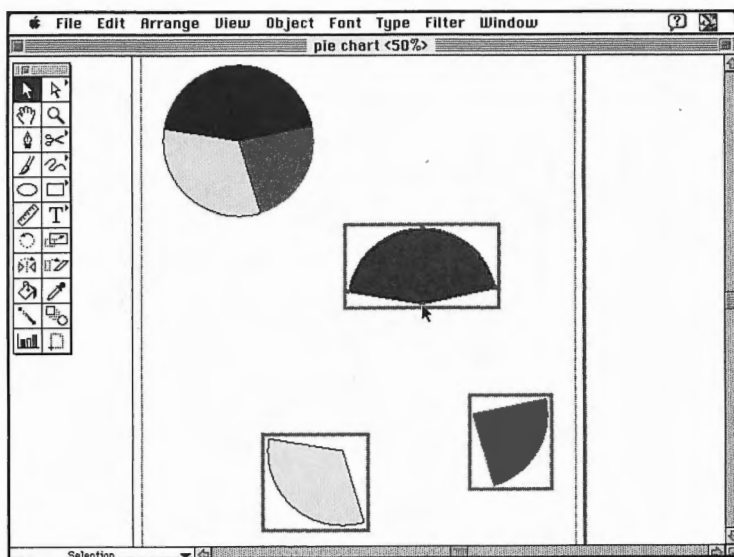


**Figure 5.16** *Apply a bevel to the shape*

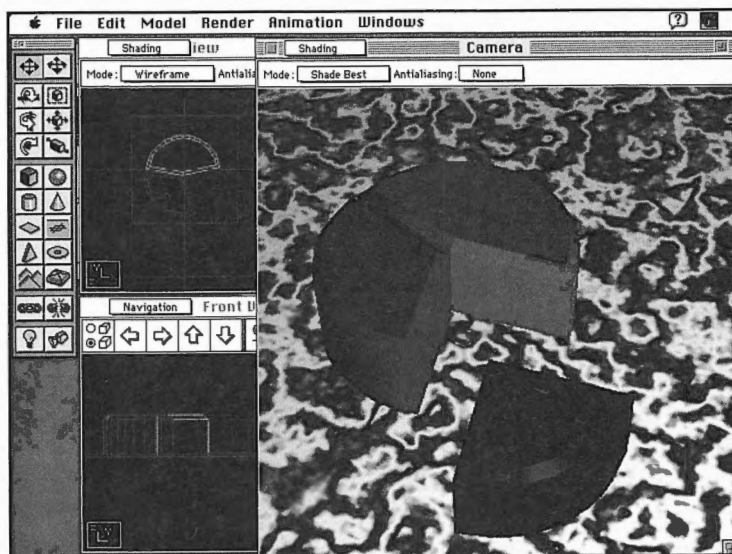


**Figure 5.17** The finished 3-D scene “linked” to the Illustrator document

6. Now open the Illustrator document to modify the proportions of the pie wedges (being sure to replace the old published wedges with the new ones) and save the file (see figure 5.18). Unless you set up the “publisher” in advance to update the editions automatically, you will be prompted whether or not you’d like to do so. This is important, for example, if the art director is waiting for the published changes, but you, as the business director, aren’t ready to send them yet. You can still work on your file and save changes without sending them off.
7. Once the new “edition” has been published, open the pie scene in Infini-D and see that all of the pie pieces have changed to reflect the changes you made to the Illustrator document (see figure 5.19). This is a great way to set up 3-D charts and other graphics that change periodically. Simple changes to the 2-D originals can be used to quickly generate new 3-D charts with all of the features of photorealistic rendering.



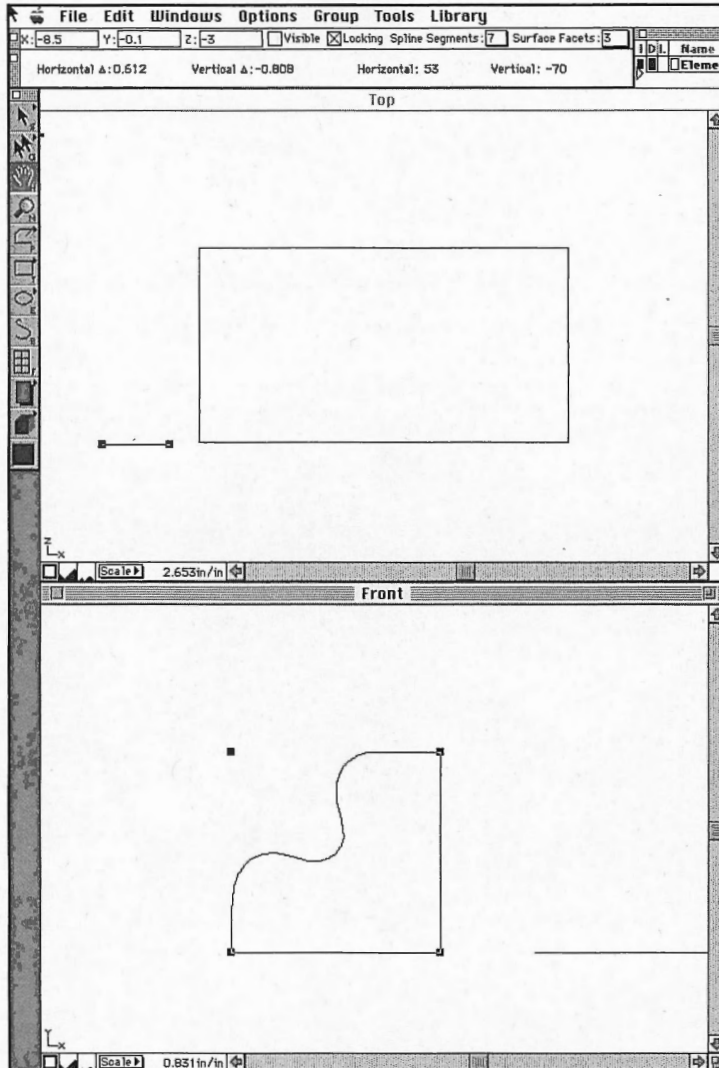
**Figure 5.18** Open the Illustrator document and re-apportion the pie



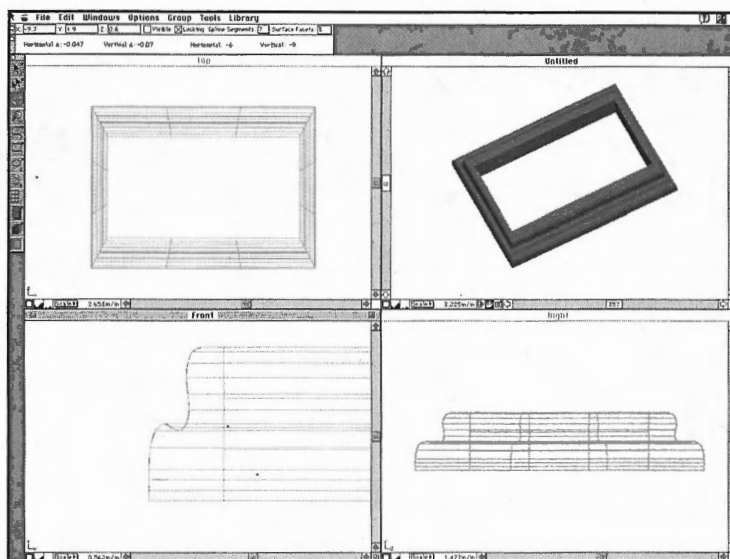
**Figure 5.19** The automatically updated chart

## Extrusion on a Path

Extrusion on a path enables you to have a shape follow a path, rather than just a straight line. If you extrude a triangular shape along a square path, you will end up with a picture frame made out of triangular “molding.” Path extrusion is a very powerful technique for creating any object that has a consistent cross section along a crooked line (see figures 5.20 and 5.21).



**Figure 5.20** The profile in the lower window to be path-extruded around the rectangular path in the upper window



**Figure 5.21** The result of the path extrusion in figure 5.20

There are many common examples of path extrusions, including bevels on non-extruded shapes, handrails, garden hoses, ropes, and drain pipes.



**Figure 5.22** This toothpaste-like series of extrusions was created in Sketch! by extruding circles around three related spline paths

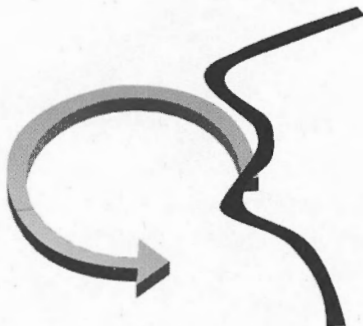
## Lathing

Lathing is the primary modeling counterpart to extrusion. In the real world, lathing is used to cut shapes from fast-spinning lengths of wood and metal. A baseball bat or a round chair leg can be turned on a lathe. A potter's wheel is the equivalent of a lathe for working with clay. As the clay spins, it slips through the potter's fingers to become a bowl or a vase. Although they can have almost any outline, lathed objects are always round when viewed along their turning axis (like a flower vase viewed from above), unless they've been squashed into an oval shape or distorted by other means. Because lathing involves revolving a shape around an axis, the terms *revolve* and *lathe* are interchangeable.

Lathing in 3-D works by turning an outline or profile of the intended object around an axis. The path followed by the profile along the way becomes the final object. Figures 5.23 through 5.27 show the process of lathing a shape.



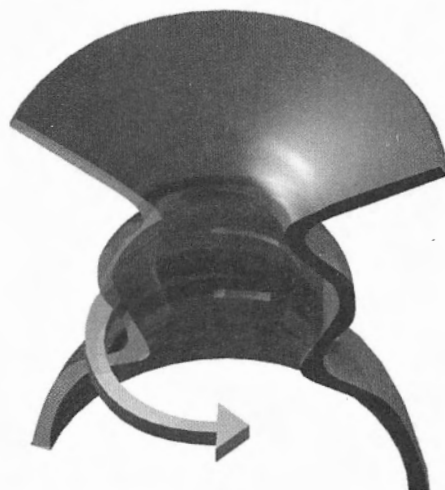
**Figure 5.23** *The 2-D shape to be lathed*



**Figure 5.24** *The shape will be rotated along the curved path*



**Figure 5.25** *Beginning the lathing*

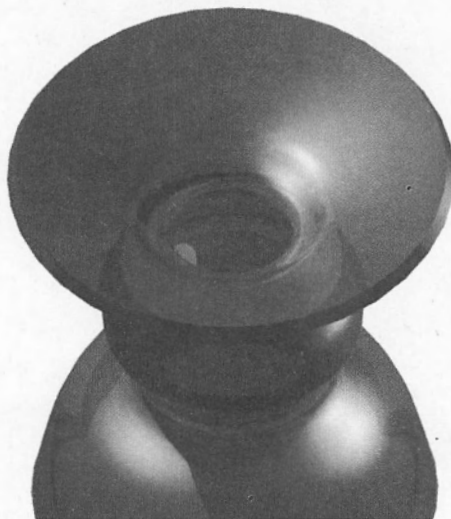


**Figure 5.26** *Continuing the lathing*

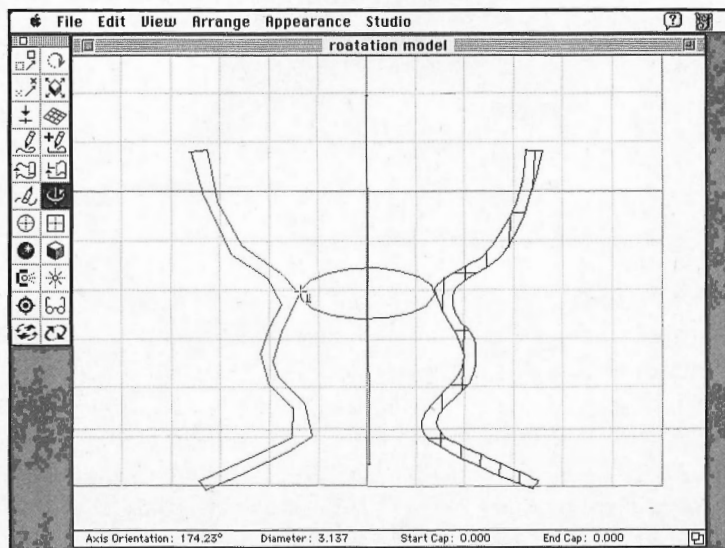
If there is a difficult concept to grasp when it comes to lathing, it's that an object can be lathed around virtually any axis in space. Some modelers, like the one offered by StrataStudio Pro, have a separate lathing module. Infini-D's interface involves a two-window view, one showing the shape to be lathed aligned to an axis, the second showing the finished shape. Since this removes you temporarily from the 3-D space, it's easy to visualize the effect of your changes. Modelers that strive to give you more flexibility and control, however, enable you to lathe objects directly in space and to move the axis of rotation. Figure 5.28 shows the lathing interface in Alias Sketch! The curved shape at the left (used to create the



previous example) is about to be revolved around the vertical axis. The curve at the right is a projection of the cross section that will result. By holding the Option key and moving the mouse, you can tilt the axis of rotation to a different angle. Releasing the mouse causes the curve to be lathed.



**Figure 5.27** *The final lathed object*



**Figure 5.28** *Lathing in Sketch!*



## Lathing Techniques

Lathing builds an object by rotating a profile around an axis in space. If you change the position or orientation of the axis relative to the profile, you will get very different shapes. For example, if the axis of revolution is parallel to a straight line, you will create a cylinder. If the axis intersects the line at an angle, you will get two cones, point to point. If it intersects an endpoint at an angle, you will create a single cone.

Here are common examples of simple lathed objects:

- Cylinder
- Vase
- Lamp shade
- Bearing
- Coffee cup
- Donut
- Pencil eraser ring

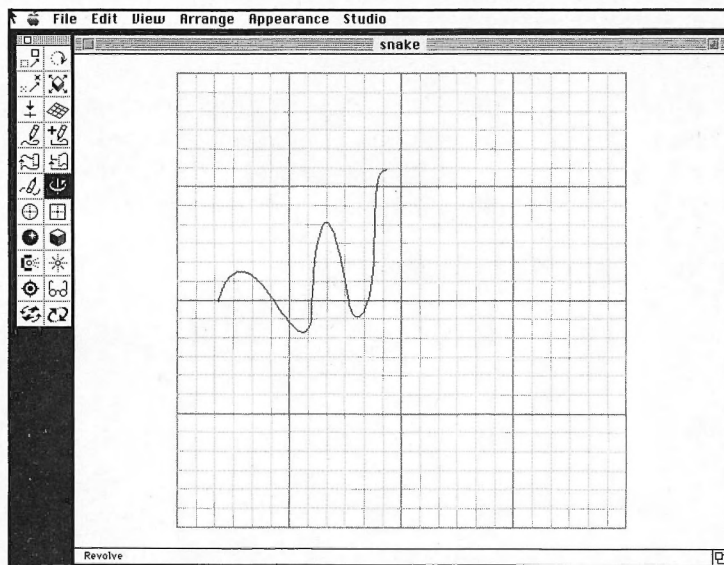
Partial lathes (a lathe of less than 360 degrees) include the following:

- Dome (a quarter circle lathed 360 degrees or a half circle lathed 180 degrees)
- Coffee cup handle (180 degree lathe)
- Arch (180 degree lathe)

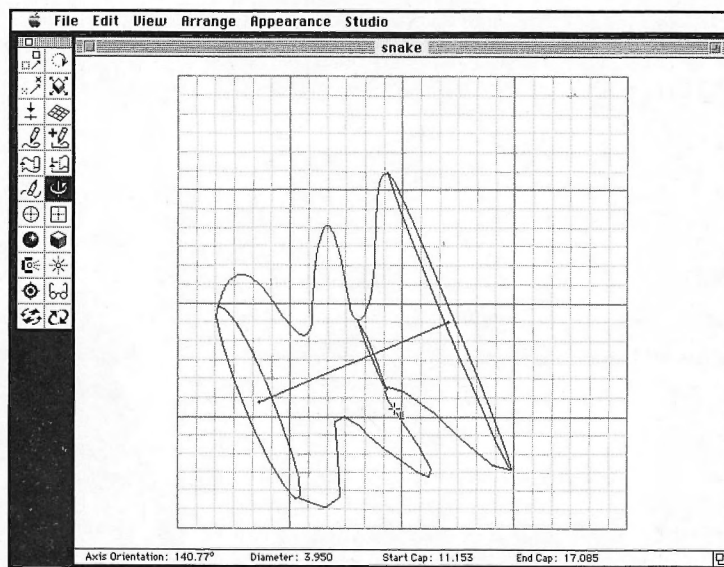
## Axis of Rotation

The following example illustrates the difference between a diagonal and vertical axis of rotation. Note that in both examples, you begin with the same squiggly line. The resulting shape, however, changes dramatically depending on what you use as an axis.

1. Begin with a line drawn on the Sketch! working plane (see figure 5.29).
2. Select the lathe tool, click the line, and drag the mouse down and to the right (see figure 5.30). Notice that this sets the “direction” of rotation (hold the Option key and drag to change the direction). The axis of rotation is perpendicular to this direction, and appears as a straight diagonal line.

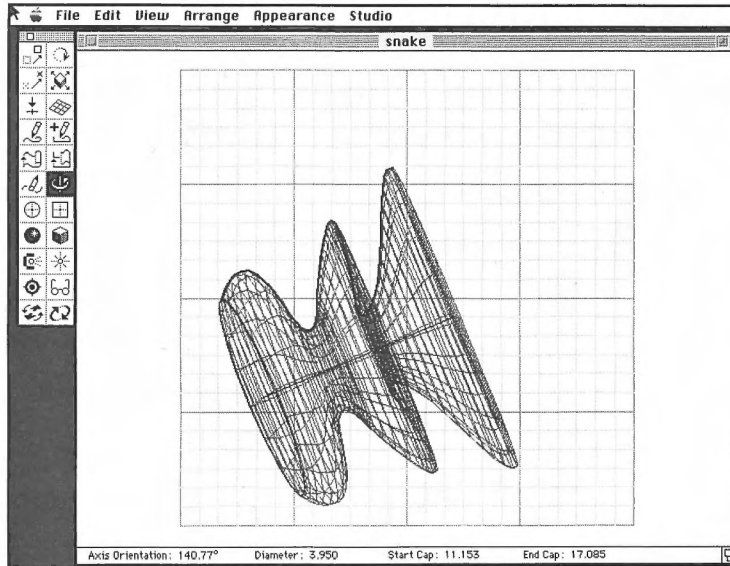


**Figure 5.29** The starting profile



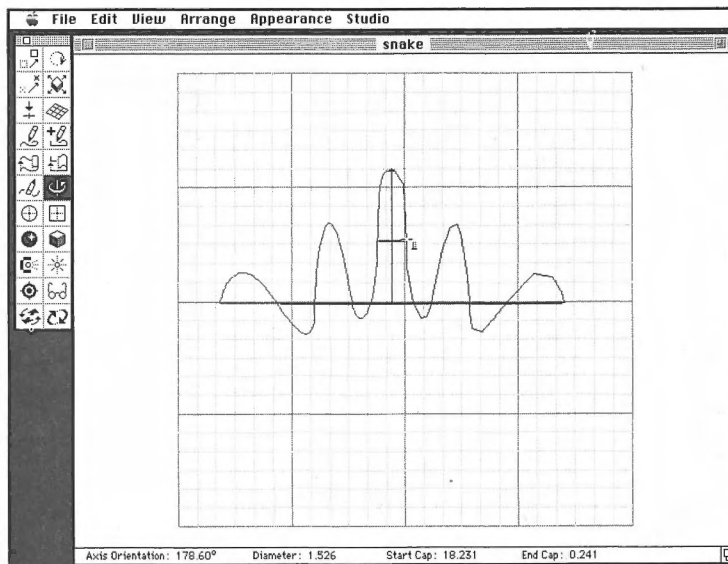
**Figure 5.30** Set the axis of rotation

3. Confirm the rotation to set the final shape (see figure 5.31).



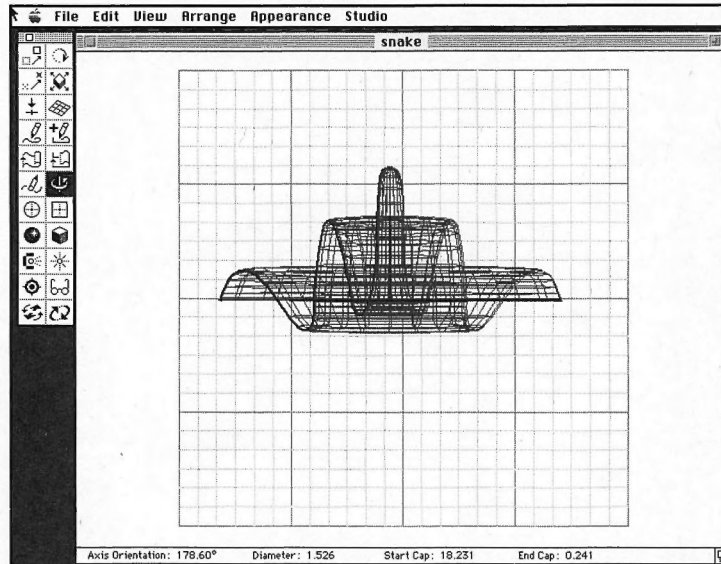
**Figure 5.31** The lathed shape

4. Undo the previous lathe command to return to just a line on the grid. This time, click and drag straight to the right a short distance. Again, you can use the Option key to change the rotation of the axis if your image doesn't match this one (see figure 5.32).



**Figure 5.32** Defining a different axis of rotation

5. Release the mouse to draw the new shape using the new axis of rotation (see figure 5.33).

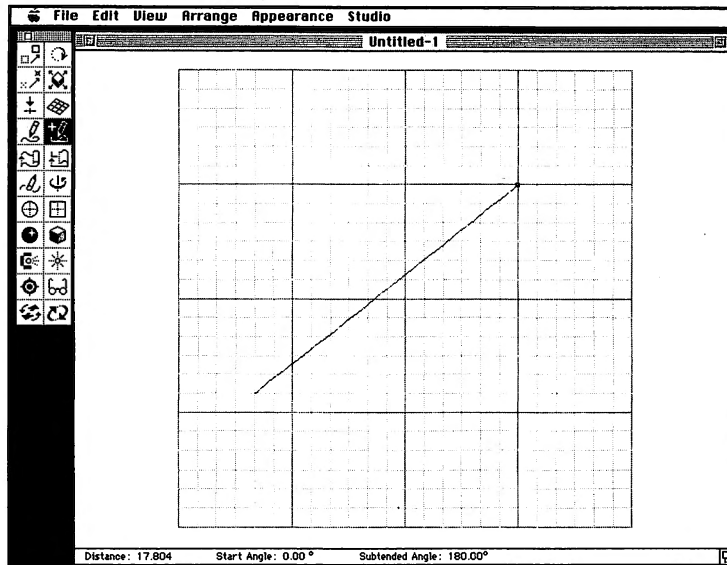


**Figure 5.33** *The original profile, revolved about a different axis, creates a very different shape*

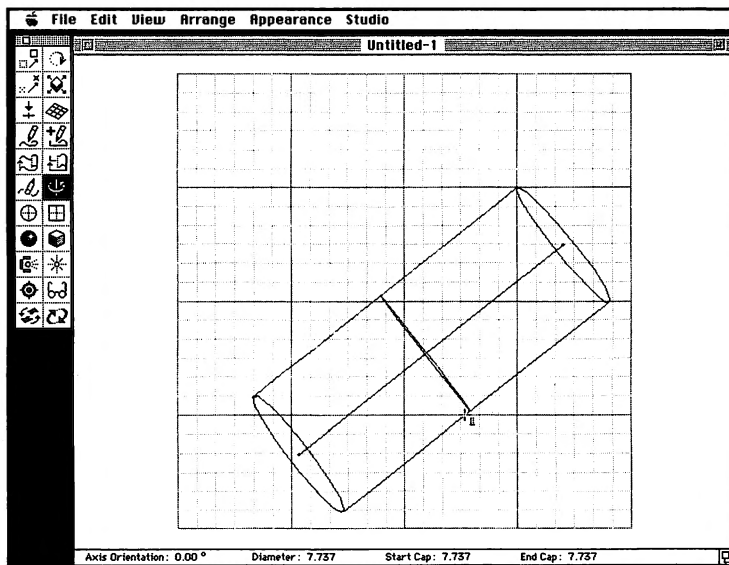
## A Cylinder

Though many 3-D programs include primitives for cylinders and other simple shapes, it may be convenient at times to make your own primitives. A cylinder is the simplest shape to lathe. It consists of a line rotated around an axis. (It's also easy to create a cylinder by extruding a circle.) You may want to lathe a cylinder instead of using a ready-made one if you want to use a specific line rotated around a specific axis or point in space. The following example illustrates lathing a cylinder in Sketch!.

1. Draw a line on the working plane (see figure 5.34).
2. Choose the lathe tool, click the center of the line, and drag away from it (see figure 5.35).

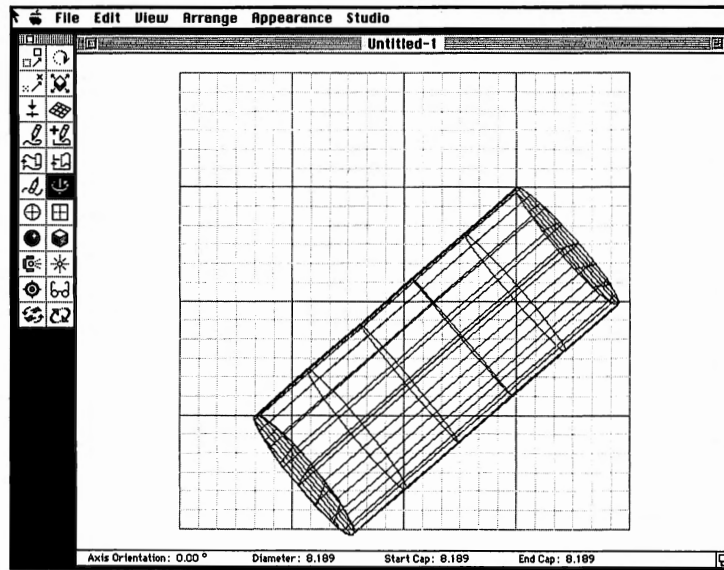


**Figure 5.34** *Drawing a line*



**Figure 5.35** *Defining the axis of rotation*

3. When the preview wireframe achieves the desired shape, release the mouse button (see figure 5.36).

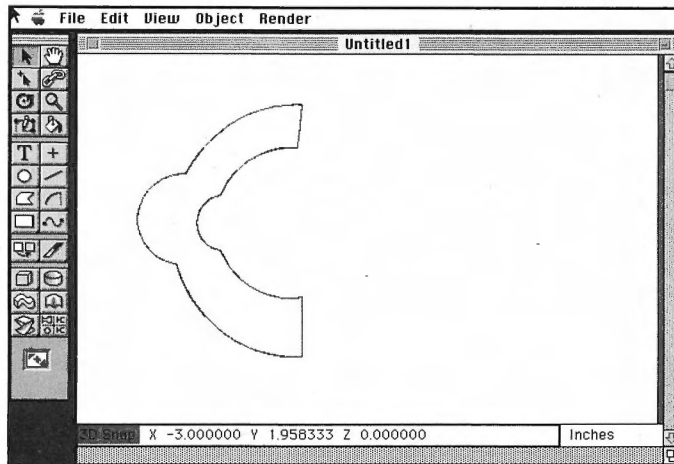


**Figure 5.36** *The lathed cylinder*

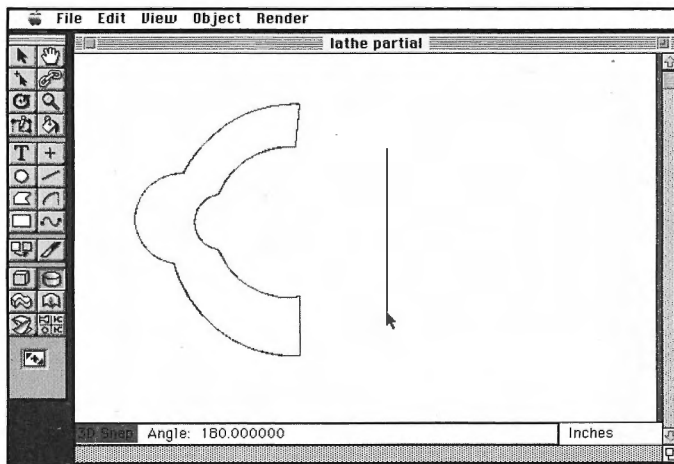
## Partial Lathe

Partial lathes are an excellent way to make many common shapes, such as pie wedges and semi-spherical bowls. The following illustration shows the lathing of an irregular shape in MacroModel (in this case, a bearing profile) around a partial circle. Instead of lathing the shape through a full revolution around the axis of rotation, this lathe goes only half-way around and stops, resulting in a semi-circular shape.

1. Using the arc and other drawing tools, draw a profile shape in MacroModel (see figure 5.37).
2. Select the lathe tool; click the drawing to tell MacroModel you want to lathe it. In the lathe degree settings box at the bottom of the window, enter a value of 180 degrees.
3. Drag the mouse to draw the axis of rotation onscreen (see figure 5.38).

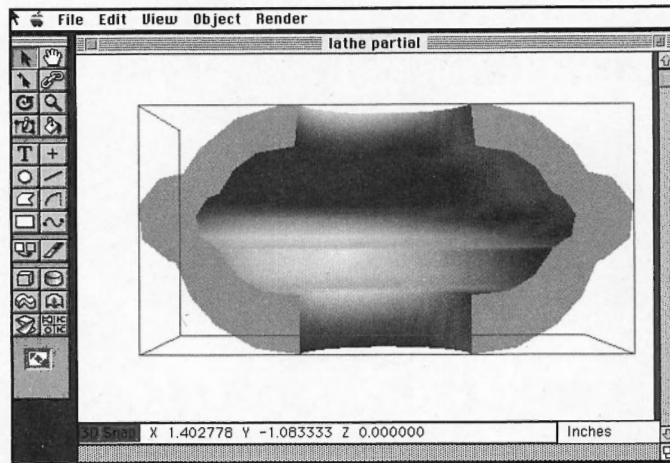


**Figure 5.37** Creating the profile shape



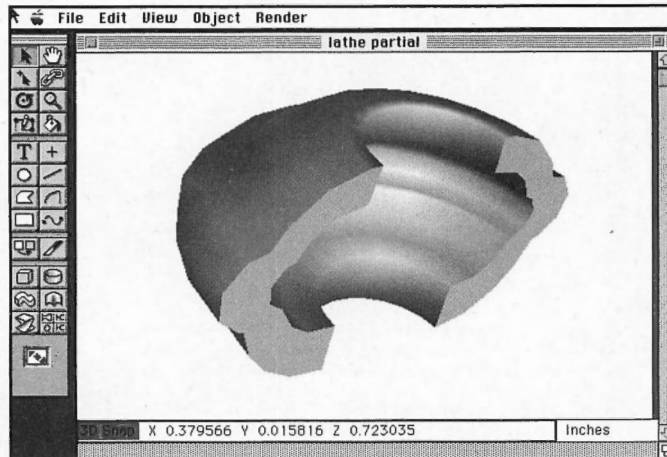
**Figure 5.38** Creating the axis of rotation

4. Release the mouse to finish the lathe and generate a 3-D shape (see figure 5.39).



**Figure 5.39** *The shape created by lathing the profile around a partial circle*

5. Rotate the model slightly to get a better idea of what you've created (see figure 5.40).

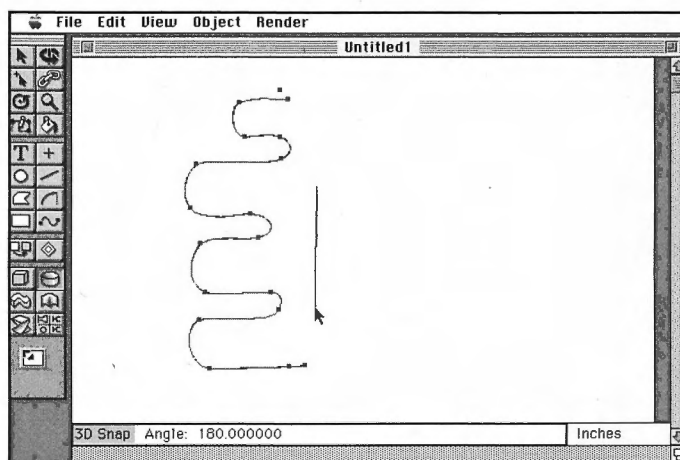


**Figure 5.40** *The same shape viewed from a slightly different angle*

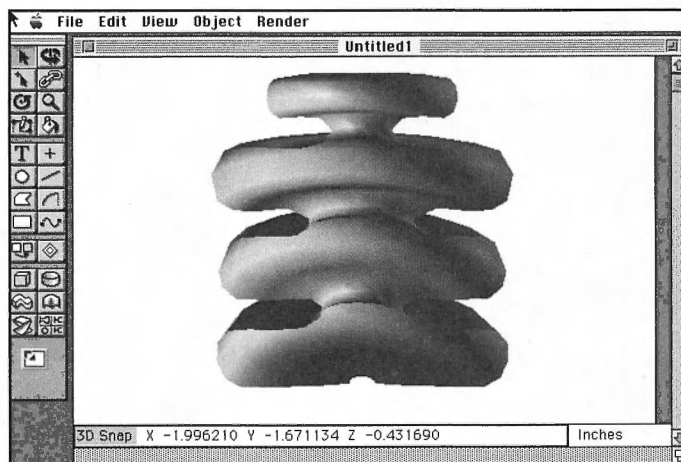


## Other Lathed Shapes

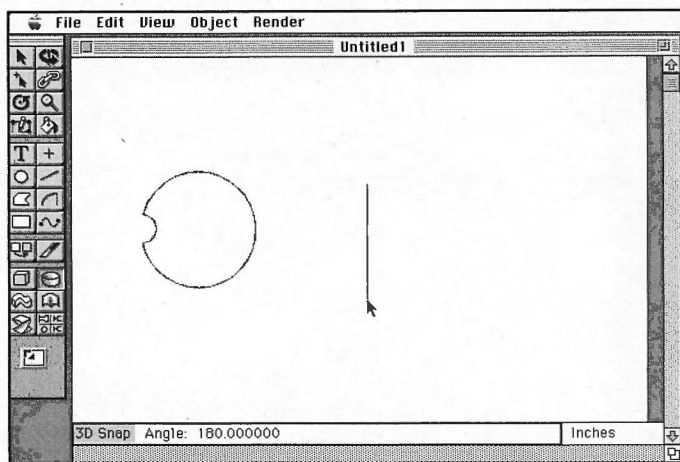
The following three pairs of images show typical lathed objects and how they were made in MacroModel.



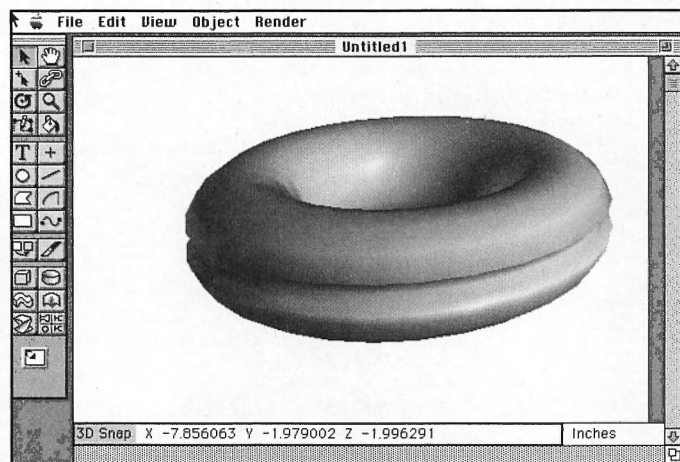
**Figure 5.41** A vase-shaped profile



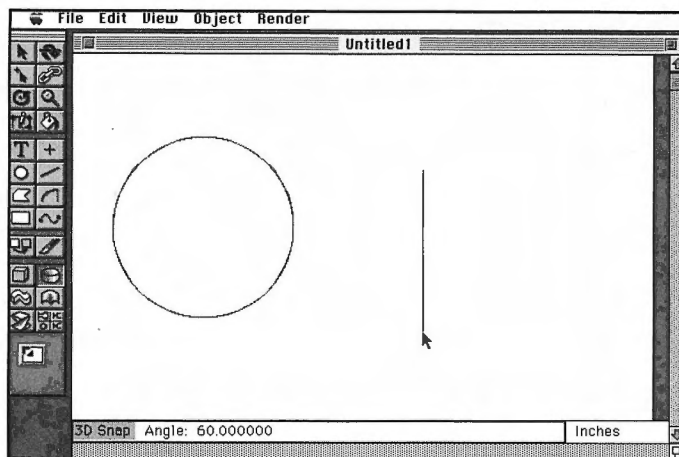
**Figure 5.42** The resulting lathe



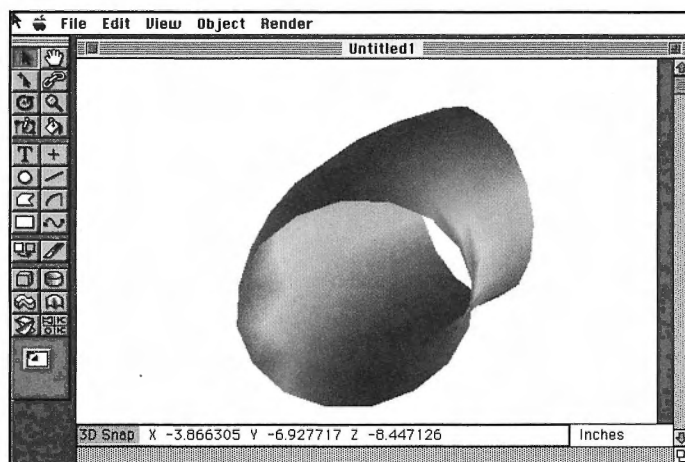
*Figure 5.43 A pulley-shaped profile being revolved around an offset axis*



*Figure 5.44 The resulting pulley shape*



**Figure 5.45** By turning off end caps and lathing a circle around a 45 degree arc...



**Figure 5.46** You can create a curved, hollow tube

## Lofting (or Skinning)

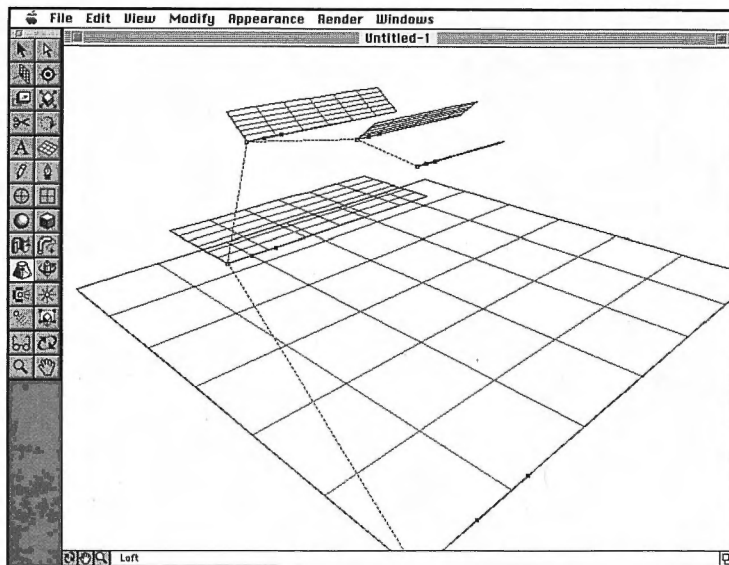
*Lofting* (also known as *skinning*) is like stretching a rubber sheet over two or more 2-D surfaces. The sheet will have the shape of each rib at the rib, while blending to the shape of the next rib. (How smoothly the sheet blends depends on the software and the smoothness settings you choose.)

This is precisely analogous to the construction of airplane wings and boat hulls in the real world, where a series of ribs or cross sections are covered with a smooth curving skin of metal or fiberglass.

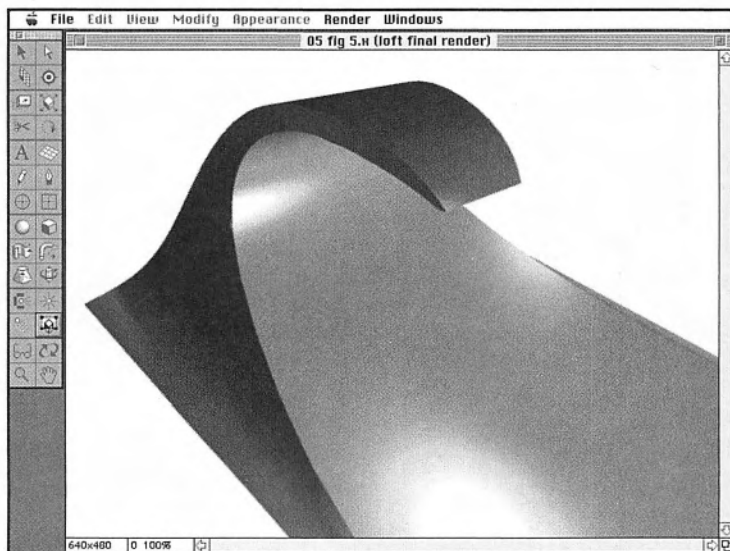
Modelers with lofting capability include the following (all of them with spline support):

- form•Z
- MacroModel
- PixelPutty Solo
- Presenter Professional
- Ray Dream Designer
- Sculpt 3D
- Sketch!
- StrataStudio Pro
- Zoom

Surfboard designers have a convention (called “one foot off”) whereby they indicate the dimensions of a board by its width and elevation one foot from the nose and one foot from the tail, as well as its width at the widest point and how far forward or back this point lies. With these measurements and a general idea of the board’s profile, they can recreate almost any surfboard, essentially skinning the known outlines by eye as they build. This same technique easily can be used to loft 3-D surfaces of many different kinds, including waves to surf on.



**Figure 5.47** A series of rectangles that have been scaled and rotated in Sketch! —the dotted line indicates the order of lofting and the small dots indicate the direction of twist

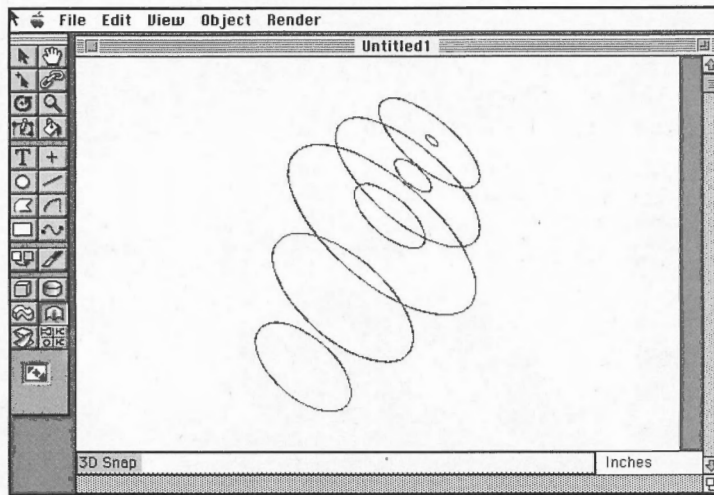


**Figure 5.48** Riding the wave

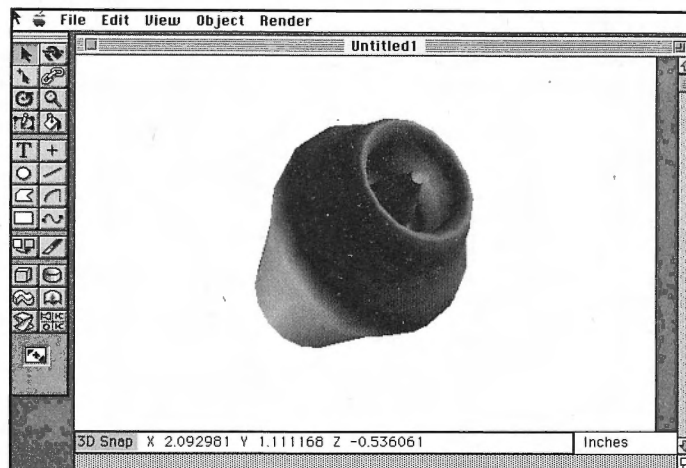
Lofting is the easiest method for creating complex organic shapes that defy description as extrusions or lathes, and it's extremely powerful. Because lofting relies on the capability to smoothly join many different shapes, it is particularly prevalent and useful in spline modelers because they specialize in making smoothly curved shapes.

For example, a bird's wing is narrow and round where it joins at the shoulder; it is wide and curved as you proceed out towards the tip, and then it is narrow and pointed at the end. By skinning these along with a few more intermediate cross sections, you can get a close approximation of a bird's wing. Of course, the more cross sections you have, the more accurate the final shape will be.

A series of concentric circles (see figure 5.49) was skinned to create the jet engine in figure 5.50. Note that MacroModel stretches the skin in the order that you click the objects.



**Figure 5.49** A series of circles is spread out along a single axis



**Figure 5.50** MacroModel generates a smooth shape by stretching skin around the “ribs”

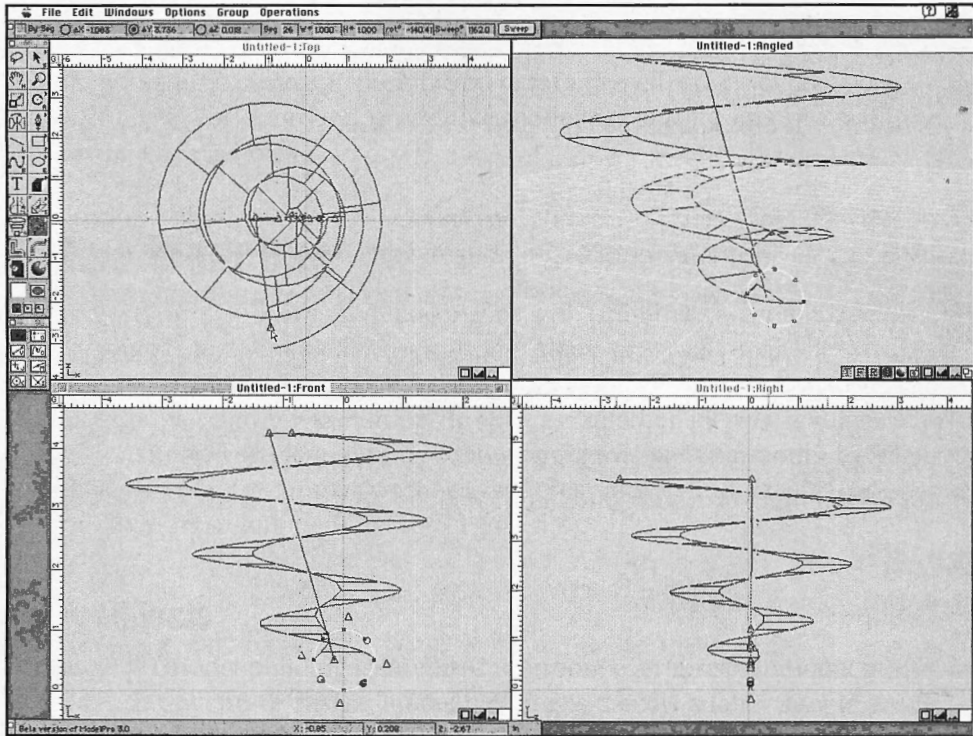
The following are examples of everyday lofted shapes:

- Airplane and automobile bodies
- Boat and submarine hulls
- Human and animal forms
- Surfboards and ski boots
- Bird wings and gun handles
- Arrowheads and sword blades
- Guitar necks and canoe paddles
- Tree trunks and kitchen sinks

## Sweeping

Sweeping is a special case of lofting (some modelers call a path extrusion “sweeping,” but I’m referring to a different animal here). Sweeping implies lofting an outline over a circular or spiral path. The result looks like a spring. Sometimes sweeping can include altering the profile along the way so, for example, it shrinks as you near the end of the swept object, as with a corkscrew. The point of sweeping is to create complex lofted shapes that would be very tedious to do by duplicating, rotating, and scaling dozens of profile shapes manually. A nautilus shell is another example of a swept object.

VIDI's Presenter Professional has an amazing sweeping tool. It enables you to interactively define a base radius and top radius for the rotation of a surface, and you can define the number and extent of coiling around the axis very much like twirling spaghetti onto a fork. This tool enables you to very quickly and easily create "spiral" shapes from tornadoes to auger shells (see figures 5.51 and 5.52).



**Figure 5.51** Presenter Pro's Sweep tool enables you to define a sweep with a simple, powerful visual interface.

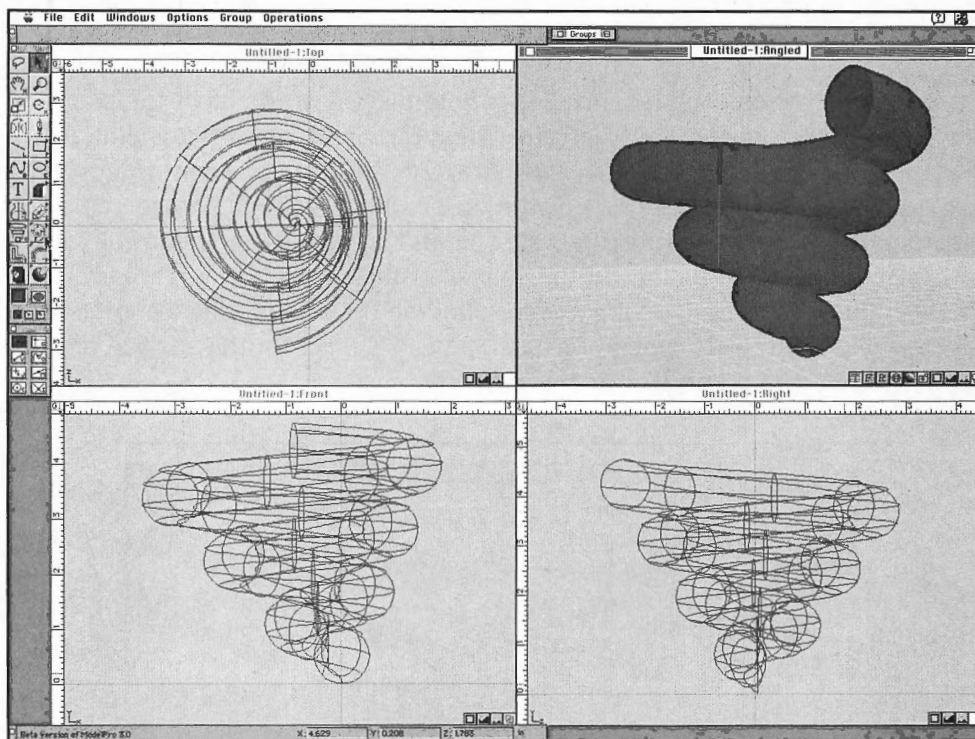
Other 3-D programs that have a sweep function, such as StrataStudio Pro, require you designate a number of steps to extrude the object, and for every step, you also specify a distance to offset the next face, and how much to rotate it from the previous step (imagine pulling a piece of taffy and twisting as you go).

Swept shapes are not the most common, but they are useful to have in your modeling toolkit. Here are a few examples:

- Telephone cord
- Spiral stair bannister



- Crepe paper wrapped around a pole
- Wrought iron
- Seashells and pig tails
- Corkscrews and coiled springs



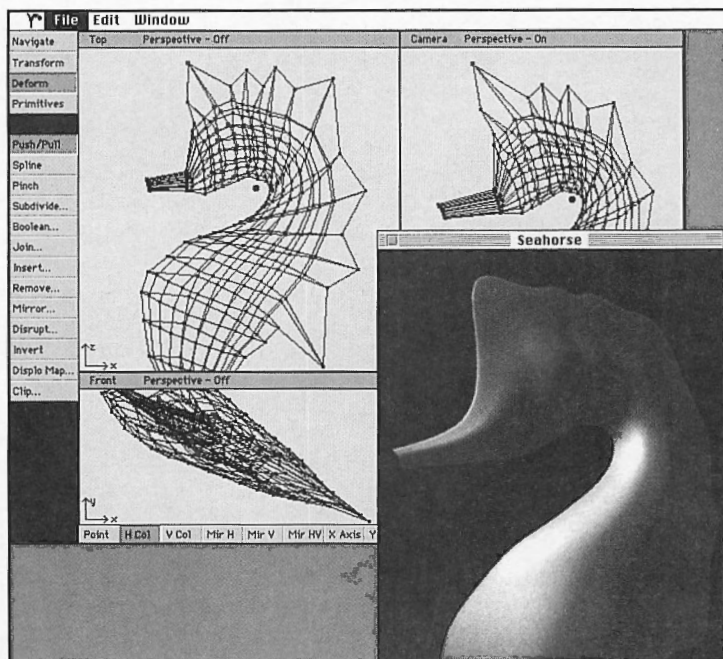
**Figure 5.52** A completed sweep can be a spring, a coil, or a tornado shape—note that the profile is tapered at the end

## Molding and Deformation

A growing number of modelers (see list later on) enable you to modify surfaces by working them like clay. Just as a glass blower may create a perfectly round pitcher and then bend a small lip outward as a pouring spout, 3-D designers often need to modify otherwise perfectly geometric shapes in small ways. Slight bends and distortions to perfectly straight and flat objects often lend a sense of realism. Organic forms, in particular, are almost never perfectly symmetrical geometric shapes.

As with lofting, molding is primarily the domain of spline modelers. In general, when you pull on a point of a surface, you don't want it to come up from the surface as a spike connected by straight lines, but rather as a smooth deformation that pulls also on the points around it (try tugging on some loose skin on your arm to see what I mean). Spline modelers are uniquely capable in this regard because they normally use the concept of surfaces joined together with tension. Polygonal modelers, on the other hand, treat points on a surface as individual entities whose movements don't affect the points around them.

Of all the modelers on the Mac, PixelPutty Solo may have the most integral implementation of this concept. Virtually everything in this program is a spline *patch* that you can push or pull. It's fairly easy to start with a shape such as a cylinder and, by molding its surfaces, end up with an octopus. While most programs convert splines to polygons for Boolean operations, PixelPutty treats the carving object like a punch that pushes in the splines of the object being "carved." PixelPutty Solo has amazing support for spline deformations; the seahorse in figure 5.53 was made very quickly by pushing and pulling on the surface of a single cylinder—note the smoothness of the rendered shape.



**Figure 5.53** PixelPutty Solo allows for very flexible deformations

Playmation and Animation Master also make extensive use of spline patches that are infinitely deformable. Also, surfaces in these programs have “skins” that act like real skin; when a model is bent over, the skin on the inside of the bend wrinkles, while the skin on the outside stretches.

These programs all offer molding tools:

- form•Z
- MacroModel
- PixelPutty and PixelPutty Solo
- Playmation and Animation Master
- Presenter Professional
- Sculpt 3D
- Shade III
- Sketch!
- StrataStudio Pro
- Zoom

There are several approaches to molding forms in 3-D. MacroModel, for example, enables you to work with Bézier-type control handles. This works just like modifying curves in 2-D illustration programs such as Adobe Illustrator or Aldus FreeHand, except the handles can modify a curved surface in 3-D, rather than just a flat line.

Some modelers offer a variety of support for NuRBS (non-uniform rational b-splines). Many varieties of NuRBs (Pixel Putty Solo and form•Z each support nine or so types) are mathematical methods of defining curves which automatically account for tension between different parts of a curve.

#### Tip

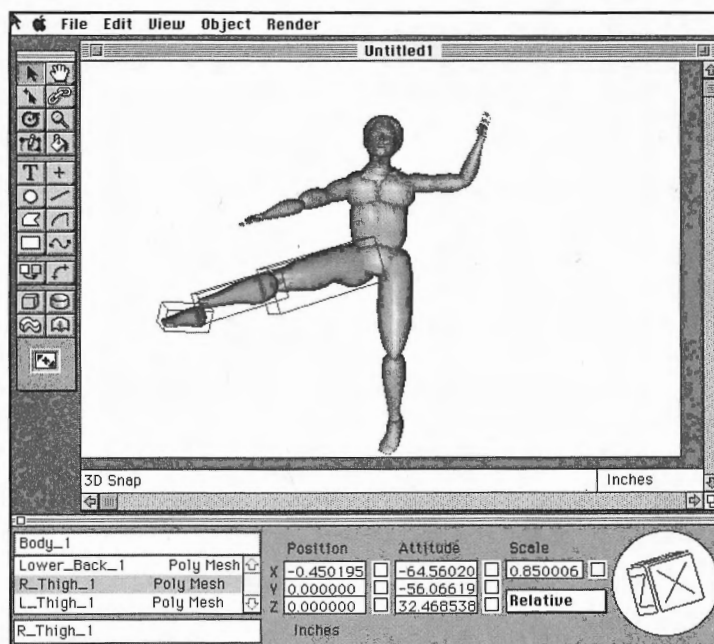
As a rule, if an object has a consistent cross section along one axis that is not circular, the object can be created by extrusion. If a cross section along one axis is round, the object can usually be made by lathing. An object with a regular cross section that follows a crooked path (such as a garden hose) is made by extrusion on a path. Objects with no regular cross sections, no matter how you slice them, are usually made by lofting, sweeping, or molding.

## Groups

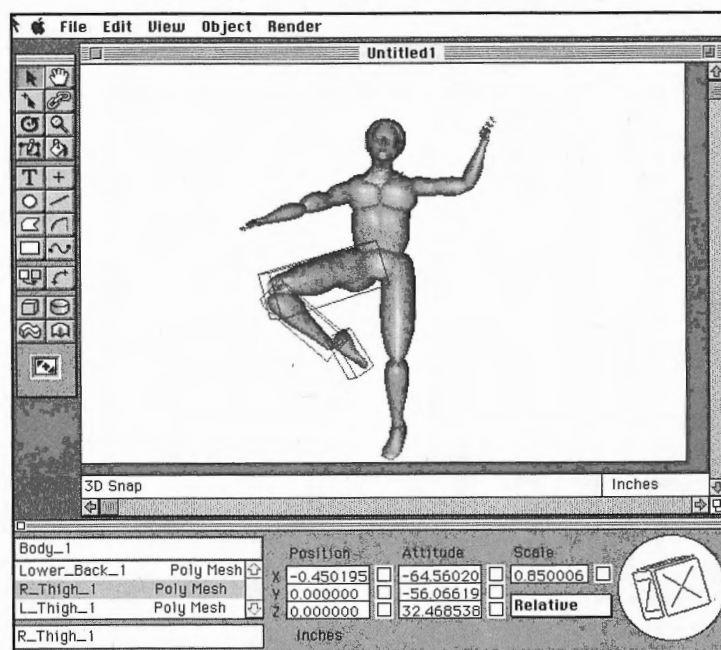
One of the most important devices offered in 3-D applications is the convention of *groups*. For those familiar with groups and layers in 2-D illustration programs such as Aldus FreeHand and Adobe Illustrator, groups should be pretty familiar. Primarily, grouping enables the user to work on several objects at the same time. Let's take the example of a human body. We know that there are three joints of the right forefinger, each belong to the same group called "right index finger." This group, along with "pinky" and "thumb," is part of a group called "right hand," which in turn is part of a group known as "right forearm," which is itself part of a group called "right arm." The right arm of course, is part of the "upper body" group, which in turn, is part of the "body." Each group is a collection of related parts. It's much easier to say "the right hand" than to say "the right index finger, and the right middle finger, and the right ring finger..." listing every part of the hand.

Groups come in handy when it comes time to perform operations such as applying textures or animating. In our example, you could move the "right hand" group, and the fingers would remain attached in the proper relationship to the hand. If you were then to move the "right forearm," the hand would keep its position in relation to the forearm, while the forearm maintains its proper relation to the "upper arm," which could then be moved relative to the "upper body." The following images show a human figure in MacroModel, imported from another one of Macromedia's products, Life Forms. Note that choosing the Right Foreleg in the Groups window selects the foreleg and all the groups attached to it (see figure 5.54). This group and then the foot group are rotated to make the leg straight; then the thigh group is selected and rotated (see figure 5.55). Also note that the placement of the rotation point of every object in this model allows it to pivot at the natural joint.

Another important use of grouping is the capability to hide and show sets of objects. When you're working on a large collection of objects in a single model, it is often hard to see what you're doing, and the screen redraws seem to take forever every time you make a change. This can make it frustrating—if not impossible—to work. By hiding everything you're not working on, you can avoid this distraction and sluggish pace. Hiding objects, also called ghosting, is very easy to do in most 3-D programs; however, if your objects are not grouped, you'll have to individually hide each one. In the example above, it's much easier to hide just the "Lower Body," than to individually hide each part of the lower body.

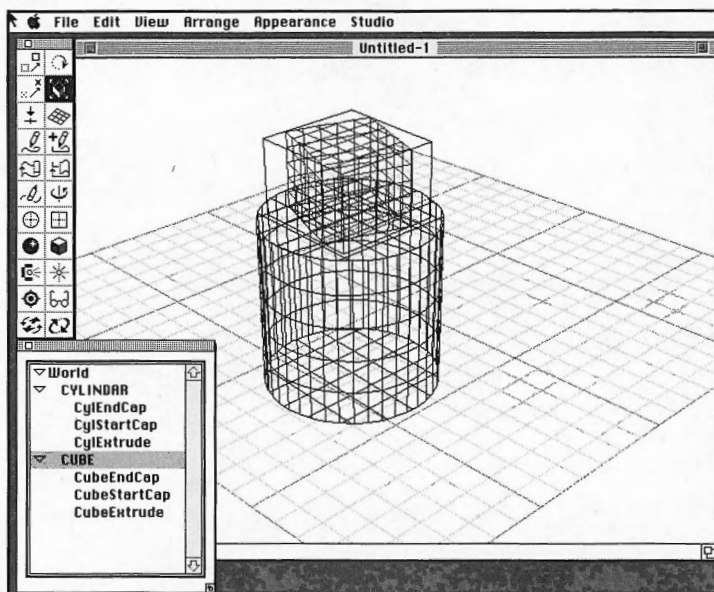


**Figure 5.54** Selecting the Right Foreleg selects all the groups attached to it

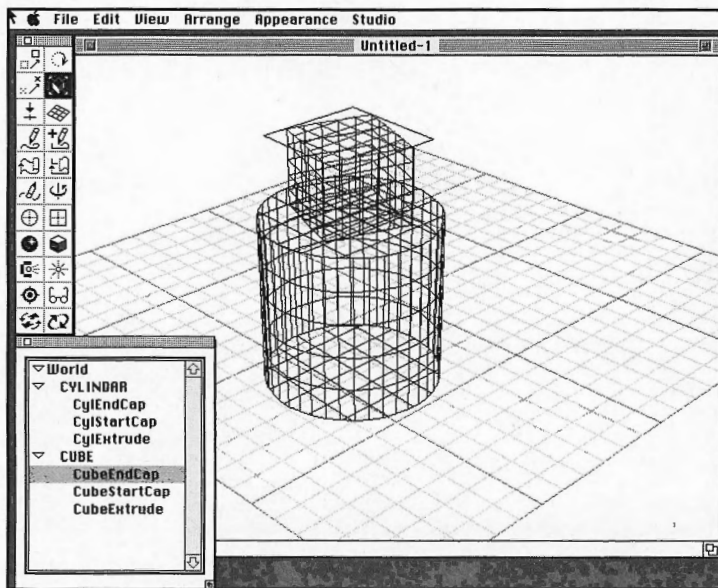


**Figure 5.55** Rotating the Right Foreleg also rotates its children objects

Group views also enable you to select objects even if their wireframes are overlapping. For example, in Sketch!, you can select just the cube in figure 5.56, or even just its top face, as in figure 5.57, even though the wireframes in both cases overlap other objects. You simply select the “cube” group from the Groups window.

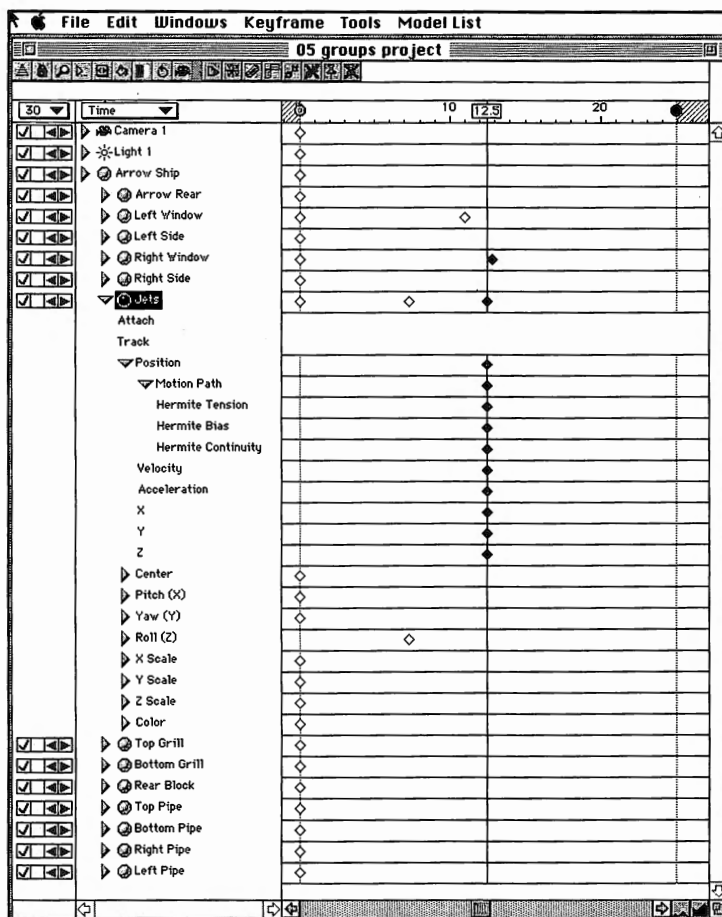


**Figure 5.56** Groups enable you to select a hidden model in a busy scene



**Figure 5.57** You can even use groups to select a face or cap of a model

Grouping is even more important for animation, because you often want an entire assembly to move—for instance, a spaceship—while parts of the assembly have their own motions going on (space shuttle probes coming and going, doors opening, and so forth). Electric Image Animation System takes the grouping concept to an extreme. Each group can be part of other groups, arranged in a hierarchical fashion, and every group can have different motion, rotation, scale, color, and many other factors at any moment in time (see figure 5.58).



**Figure 5.58** Electric Image Animation System provides a very complete environment for dealing with grouped objects in an animation

Presenter Professional has a similarly deep level of control over groups. Grouped objects are also very useful in dealing with variations in materials and shapes.



## Shapes

While it's sometimes possible to carve shapes out of a single block, it's generally easier to create the constituent shapes and add them together to form the whole.

With experience, it is easy to recognize the component shapes in everyday objects. For example, you will see that a stereo is a rectangular solid, with more solids for buttons, and a few cylinders for knobs. While it may not be obvious at the outset, a film roll is made by extruding a shape like the letter "Q" around a cylinder. A baseball bat, as in real life, is made by lathing the bat's outline. The banister of a spiral staircase is made by sweeping a single cross section of molded wood along a spiral path.

Most 3-D applications offer only a subset of the many possible tools for creating complex shapes. In many instances, however, the shape you are after can be improvised with a combination of techniques.

Some shapes are not created easily. Organic forms are the most difficult, because they shun the geometric order imposed on manufactured goods. Creating these complex shapes requires special techniques.

Some modelers (form•Z, ZOOM, Presenter Professional, Sculpt 3D, PixelPutty Solo) can *carve* one shape with another. Even with these, however, you begin with basic shapes which are additively and subtractively combined to form a single object.

## 3-D Digitizing

Some complex objects can be "scanned" using special digitizing tools that record an object's shape in three dimensions. In fact, this is the way many Hollywood animators work: an artist will sculpt a finely detailed model out of clay, then the model is in turn painstakingly digitized into 3-D using a hardware digitizing tablet. The resulting models are then articulated and animated in the 3-D software.

Immersion Systems and Mira Imaging both offer 3-D digitizers that work with the Mac (they are both based on Mira's HyperSpace software).

Using a digitizer is a fairly painstaking process. As you pass the "wand" over the surface of the model, you have to press a key to tell the computer when you're at a vertex point. The computer then assembles the information in the form of a 3-D model.



## Primitives

As painters and sculptors have long known, most objects, even the soft, organic forms of humans, are suggested by primitives. *Primitives* are the simplest of geometric shapes: spheres, cylinders, cubes, and cones. Flatten a sphere and it becomes an ovoid; square a cone and it becomes a pyramid; stretch a cube and it becomes a rectangular solid; squash a cylinder and it becomes a disk. Most 3-D programs offer simple, quick ways to create primitives, as well as ways to mold and combine them into the shapes you need.

Because primitives are mathematically very simple, many 3-D programs are fast at modeling and rendering these shapes, offering an incentive to the modeler to use them whenever possible.

### Block

A block is formed either by drawing directly using the application's primitives tools, or by extruding a rectangle or square perpendicular to its original plane.

To create a beveled cube or rectangular solid, you'll have to use the bevel extrude command offered by the 3-D application, or you can alternately use a path extrude command to trace a rectangle with a beveled line. In programs like Sketch! that enable you to modify individual splines, it's also possible to shrink the top surface of a cube slightly to create a bevel.

### Cylinder

A cylinder is like a round block. Instead of beginning from a square extruded upwards, it is formed by extruding a circle upwards. If you extrude a circle a short distance relative to its diameter, you'll get a flat disk. Extrude it a long distance relative to its diameter and you can create a rod or straight wire.

Additionally, you can form a cylinder by lathing a rectangle around one of its edges. To form a cylinder without end caps, you can lathe a straight line around a parallel axis.

To form an open-ended cylinder with thick walls, lathe a long thin rectangle around an offset axis.

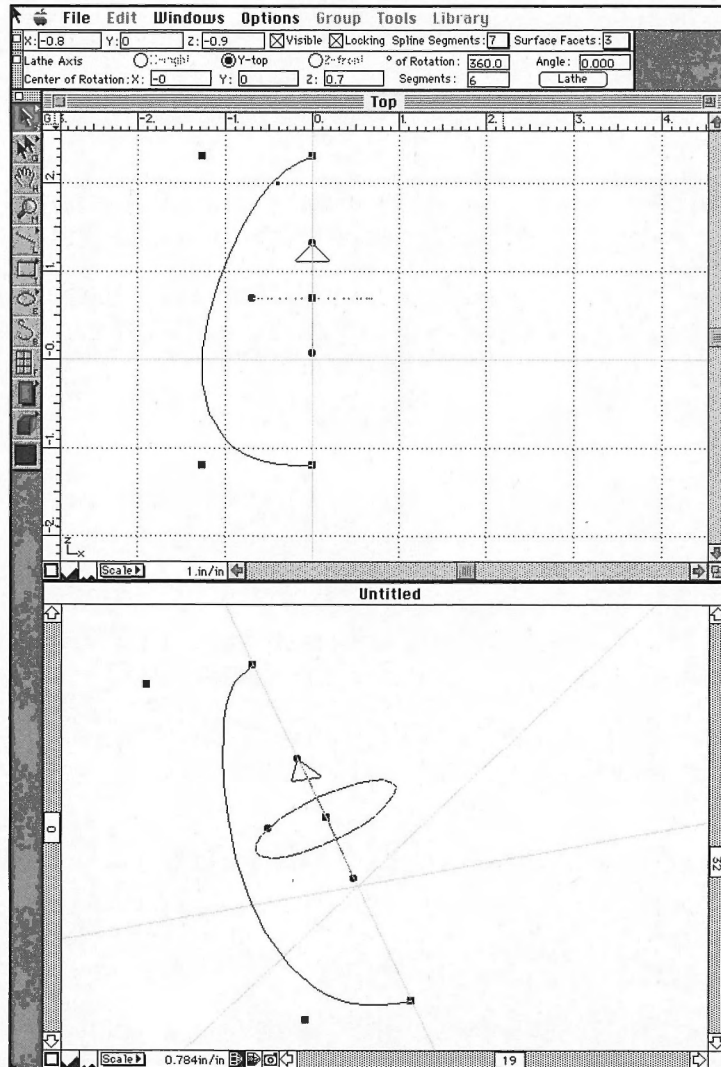
### Sphere

A sphere is formed in 3-D either by drawing directly from a tool menu choice, or by lathing a half-circle around its straight edge. To create a half-sphere, lathe a

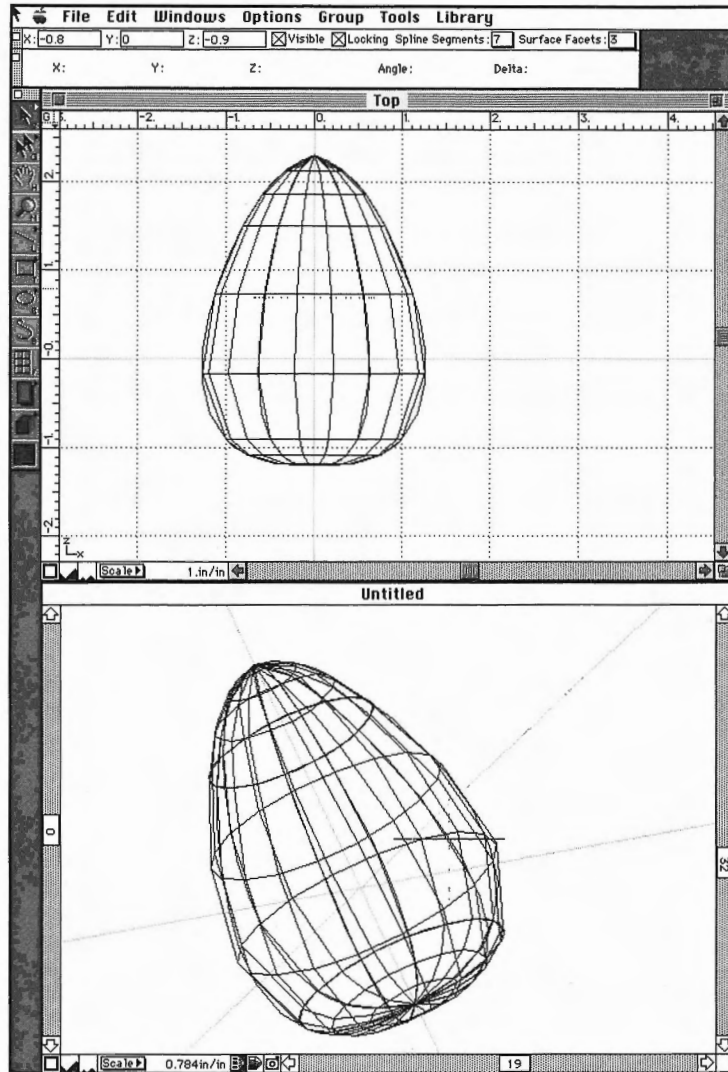
half-circle 180 degrees. Spheres are such common primitives that almost every 3-D program offers them.

## Ovoid

You can squash a sphere in one or two dimensions to form an elliptical solid (an egg shape). If you need an ovoid shape that is narrower on one end than the other, you can lathe an outline of the desired shape around a central axis (see figures 5.59 and 5.60).



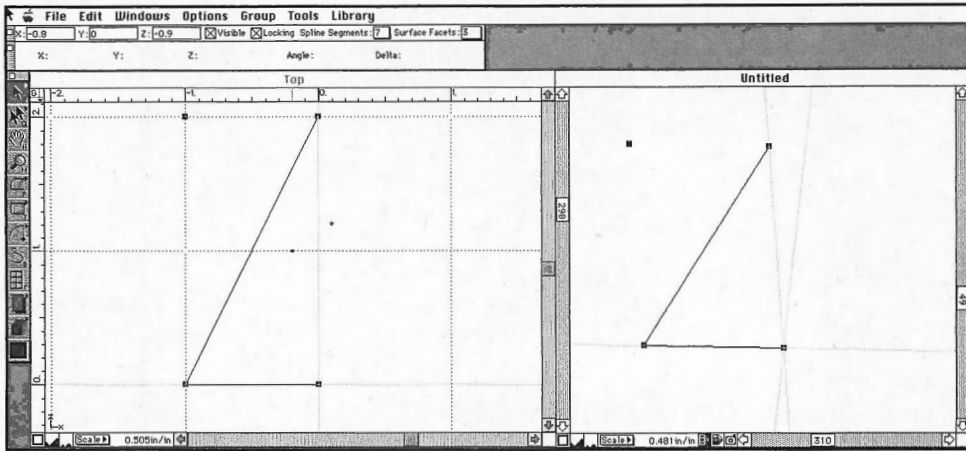
**Figure 5.59** The outlines used to lathe an ovoid



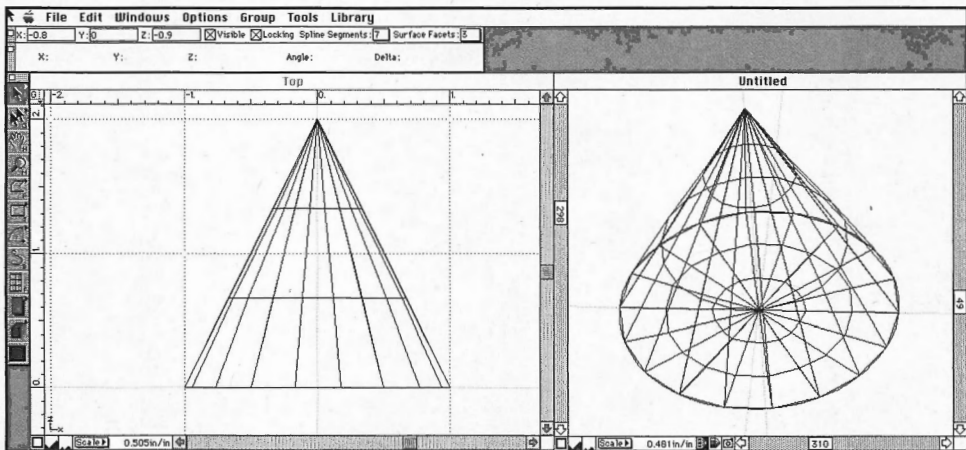
**Figure 5.60** The lathed ovoid

## Cone

A cone is created either directly through drawing from a toolbox, or as a line lathed diagonally around an axis. If you lathe a right triangle around one of its 90-degree axes (see figure 5.61), you'll get a closed cone (see figure 5.62). Lathing a diagonal line creates a cone with an open end, like an ice cream cone.



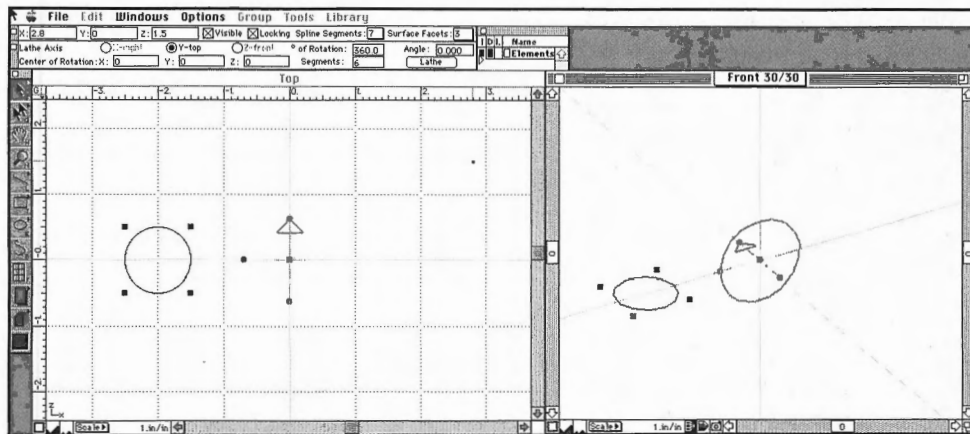
**Figure 5.61** A right triangle can be used to lathe a cone



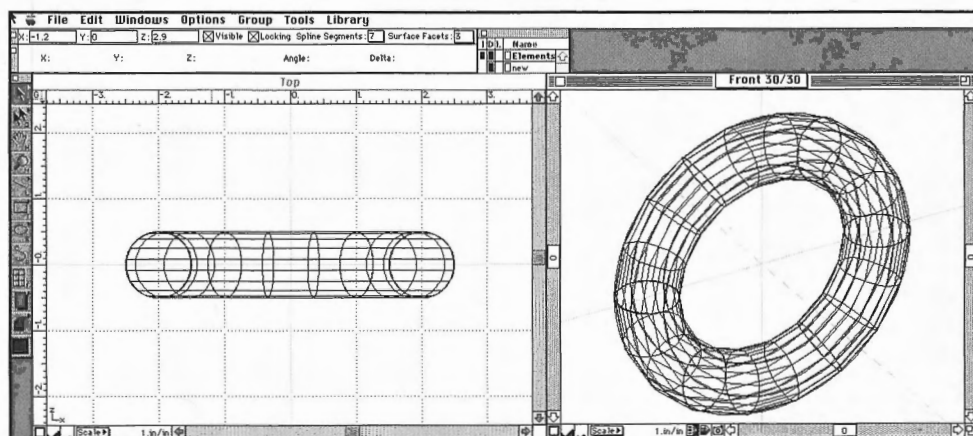
**Figure 5.62** The lower “leg” of the triangle could be omitted if you want to lathe an open cone (like an ice cream cone)

## Torus

A torus, more commonly called a “donut,” is sometimes provided as a primitive, where it has two pairs of pertinent variables: its major axes, which define the height and width of the big circle; and its minor axes, which define the diameter of the “tube’s” cross section (see figures 5.63 and 5.64).



**Figure 5.63** Defining the cross section and axis of rotation for a torus



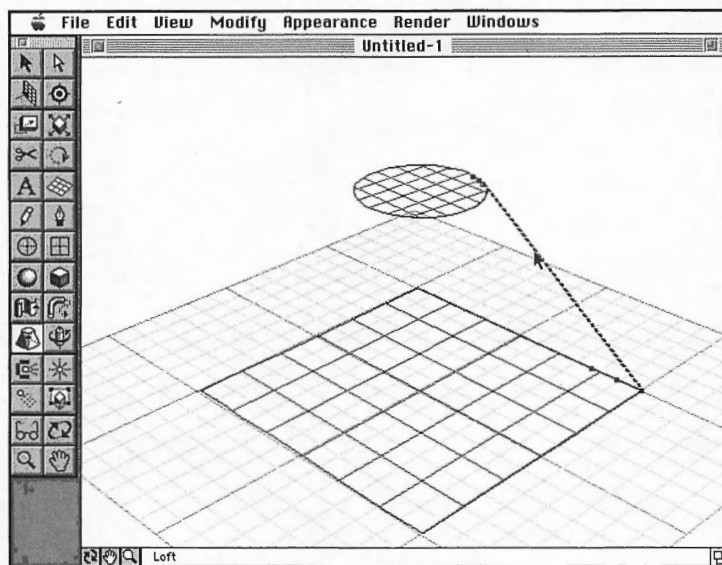
**Figure 5.64** The lathed torus

A standard torus is formed by lathing a circle around an offset axis, but similar shapes can be formed by revolving non-circles, such as squares, triangles, or the grooved outline of a bearing.

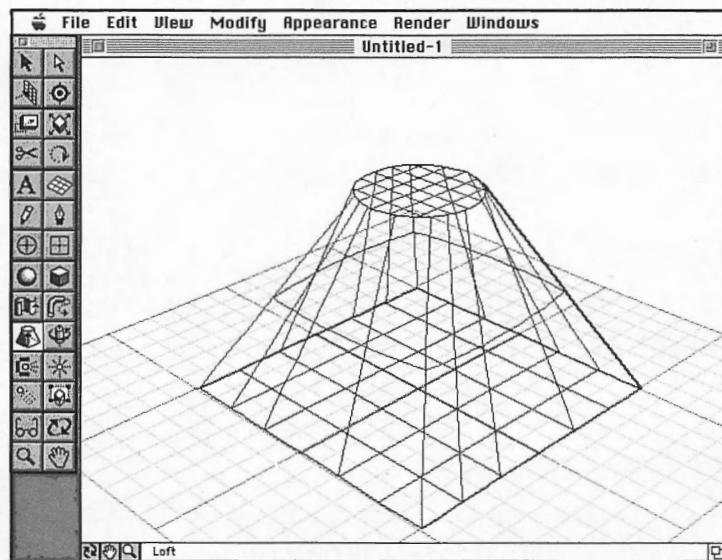
## Pyramids and Similar Shapes

Some programs support the creation of three- and four-sided pyramids. If a primitive tool isn't offered for building these shapes directly, you can try extruding the base shape, then shrinking one end to an infinitely small size. Or far more easily, you can loft between a large square and a minute one.

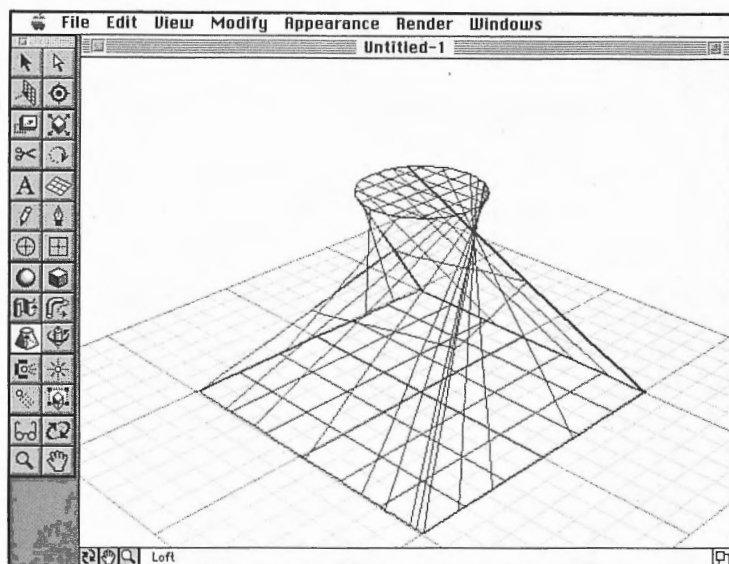
The following figures show the creation of a round-topped “pyramid” using Alias Sketch!’s loft tool (see figure 5.65 through 5.67).



**Figure 5.65** A flat circle is placed several inches directly above a flat square and both are selected with the Loft tool



**Figure 5.66** The round-topped pyramid with smooth corners at the top and sharp corners near the base



**Figure 5.67** It's important when lofting a shape to control the “twist” of the loft action—this is the same loft operation as in figure 5.70, but with a twist

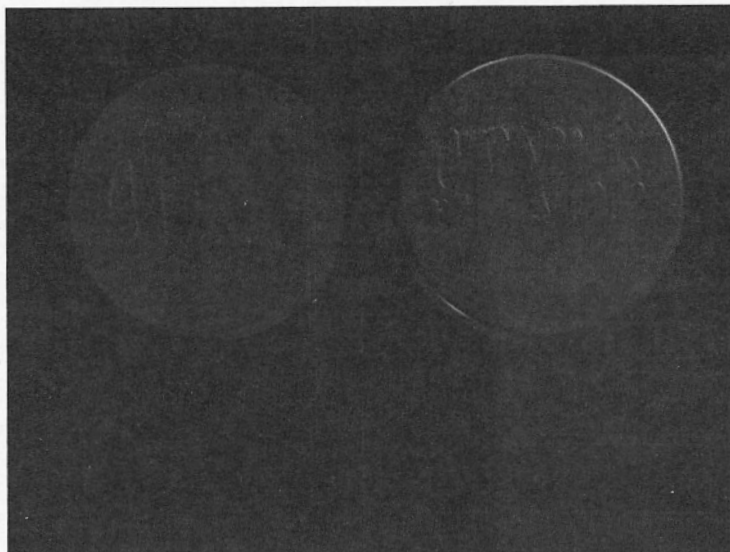
Some programs that support cones as primitives enable you to specify a number of sides for a polygonal base—instead of a circle—as a way of generating pyramids.

## Bevels

Most manufactured objects are beveled. The reason for this is simple enough: sharp edges. If television sets, stereos, furniture, and even jewelry were made with perfectly hard geometric edges, people would constantly be cutting themselves on everyday objects. Craftsmen long ago discovered that softening the edges of objects is a good idea, so if you want to make your creations look realistic, you'll have to come to grips with the need for bevels. Wooden furniture is particularly hazardous if it has sharp edges because it splinters, so carpenters often use intricate bevels which serve the extra purpose of decoration.

In 3-D, bevels serve another important function: they catch and reflect light. If you were to place a black cube against a black background it would virtually disappear. However, if you bevel the cube's edges, the cube will suddenly stand out against the background because the bevels are curves which scatter light in all directions, causing the edges of the cube to be visible (see figure 5.68).





**Figure 5.68** Both objects in this scene have identical geometry and identical textures and both are under identical lighting—the difference is in the bevels

Another consideration in creating bevels is that they help to make objects look aged. If you look at pictures of Greek or Roman ruins, you will see that the edges of great marble blocks are all worn down by weather (and lately by acid rain). If you were to model a recreation of the Parthenon and use hard (rather than beveled) edges, it will have the appearance of being new.

Another common use of bevels is in glass. Bevels in glass catch and reflect light, refracting it in many directions, like a prism. While 3-D is not yet sophisticated enough to diffract light into a rainbow, beveling glass and mirrors will enhance the sparkle and realism (and increase the complexity) in your scenes.

## Bevels in Animation

Bevels are particularly important in animation, where highlights glinting off moving bevels greatly enhance the sense of motion and sparkle in a scene. With the careful aiming and moving of spotlights, or with creative reflection mapping, you can make highlights “travel” along a beveled edge.

Many animators choose to apply a separate texture to bevels, which enhances readability as well as the sense of depth in 3-D text. However, this requires using a program that supports bevels as separate objects, such as Electric Image’s Mr. Font.



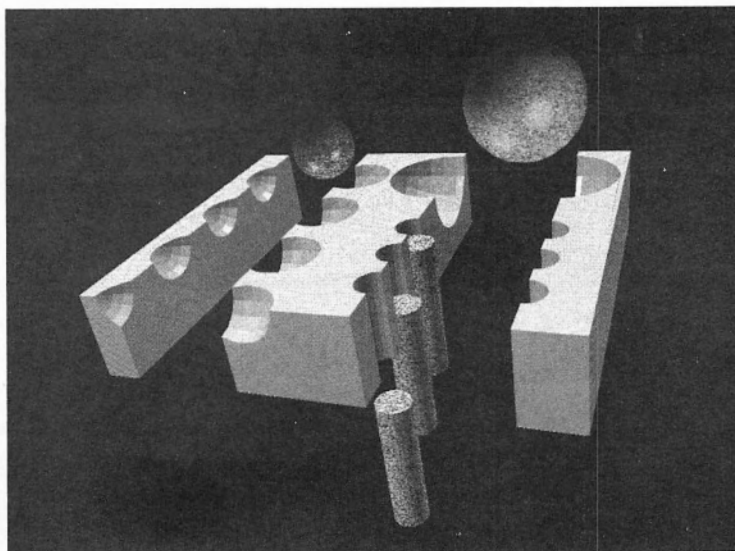
## Solid Modeling

There are currently only two programs which offer true solid modeling on the Mac: form•Z and ZOOM. 3DGIG, expected to ship for the Power Mac in the Spring of 1995, is also a solid modeler. Solid modelers differ from surface modelers in that their interior volumes, as well as their surfaces, are defined. If, for example, you slice a solid block down the middle, you'll end up with two thinner solid blocks. If you slice a block made in a surface modeler, you'll end up with two hollow boxes because the surfaces enclose empty space.

There are several significant advantages to solid modeling. Objects can have more of the properties of real objects, such as mass and volume. You could, for example, create a complex model of an engine, and determine from the model exactly how much the engine will weigh, and the inertia of its pistons.

Another advantage of solid modeling is that you can view portions of your model as cutaway sections. Slice a building design down the middle, and you can see a cross section of offices, floors, and utility shafts.

The image in figure 5.69 was made from a few primitives in form•Z: a block, three cylinders, two spheres, and a plane. The block was "carved up" with the various primitives, then sliced with the plane. It was rendered in StrataVision 3d.



**Figure 5.69** Solid modeling enables you to carve shapes with one another

## Booleans

Booleans are a group of simple logical equations that affect two objects that intersect in space, along the lines of “Cube A but not Sphere B.” The result of this operation is a cube with a spherical hole in it. It is also possible to fuse objects together with addition—for example, “add columns A and B to arch C”—to produce a complete architectural arch.

There are a rapidly growing number of modelers that offer Boolean operations. Only solids modlers, technically, can do this; Surface modlers only simulate it. However, for many purposes, the result is the same.

For example, suppose you need to create a single object that is partly extruded, partly lathed, and partly molded. The surfaces of all three parts need to flow smoothly together, with no interruptions or nonconformities, and because the objects will be semi-transparent, they can’t have overlapping protrusions that may show up in rendering. Modelers that do Booleans easily handle these situations. In addition, programs that render with RenderMan, as well as Shade III, can do Booleans at the rendering stage.

Surface modlers that offer Boolean operations include: Sketch!, Sculpt 3D, Presenter Professional 3.0, and PixelPutty Solo.

While solid modelers actually carve holes into volumes, surface modlers do *surface trimming*. For example, if you were to carve a hole in a cube using a cylinder, a surface modeler will cut two round holes in the ends of the cube, cut a rectangular slice of the cylinder, then throw the remaining pieces away. The sliced cylinder is then “stitched” to the cube at the edges of the round holes. For appearances sake, this is no different than the solids approach, but it isn’t always as seamless or as effective.

The following images illustrate the concept of Boolean modeling, in this case, with form•Z. The first image shows a block with four cones arranged at and intersecting the corners (see figure 5.70). (This arrangement would be equally easy to create in a modeler without Booleans.) The second image shows the results of subtracting the cones from the block (see figure 5.71). The third image shows the results of the Intersection Boolean. This leaves behind only the volumes that the objects have in common (see figure 5.72). The final image shows the results of subtracting the block from the cones (see figure 5.73).

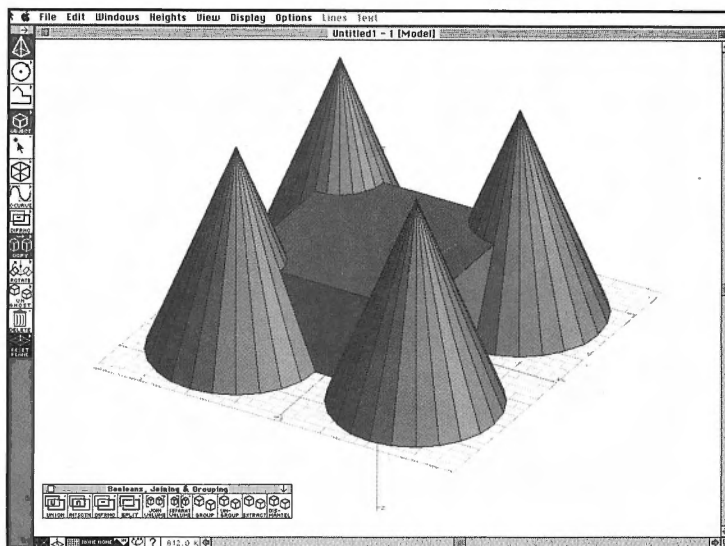


Figure 5.70 A block with four cones intersecting the corners

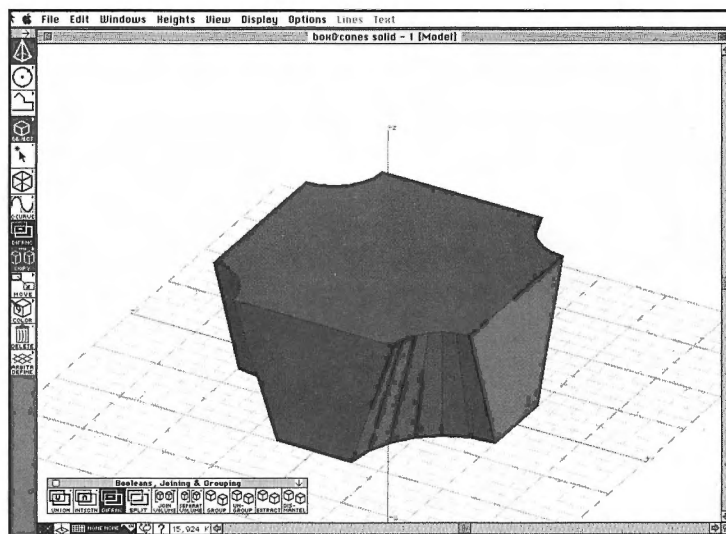
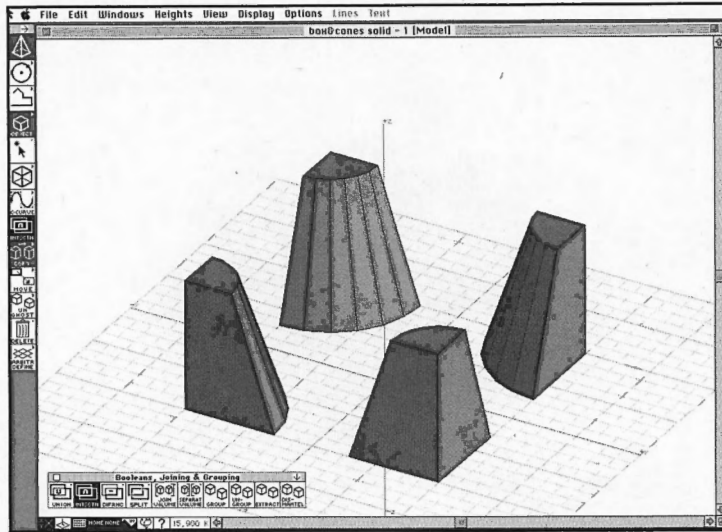
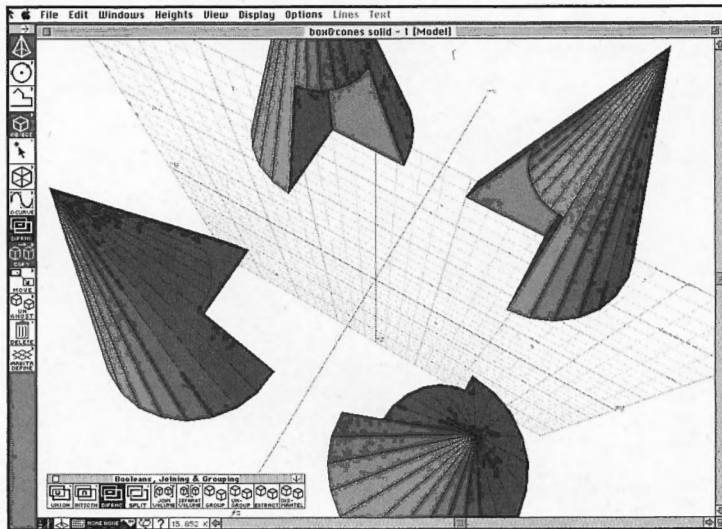


Figure 5.71 The result of subtracting the cones from the block

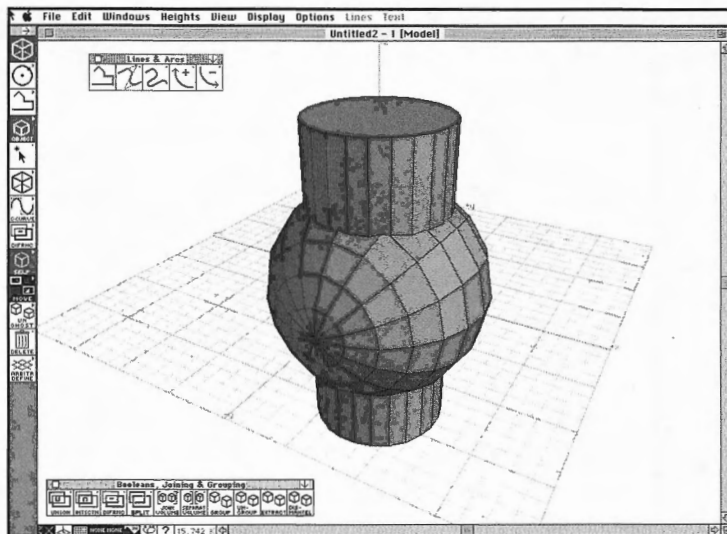


**Figure 5.72** *The intersection of the block with the cones*

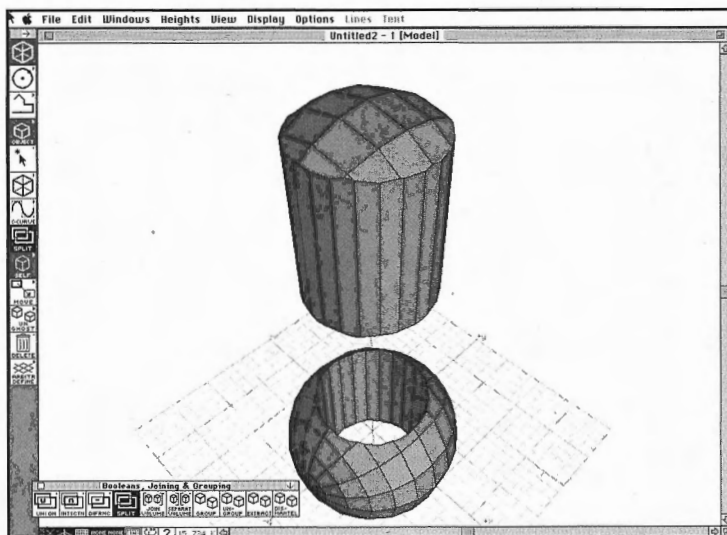


**Figure 5.73** *The result of subtracting the block from the cones*

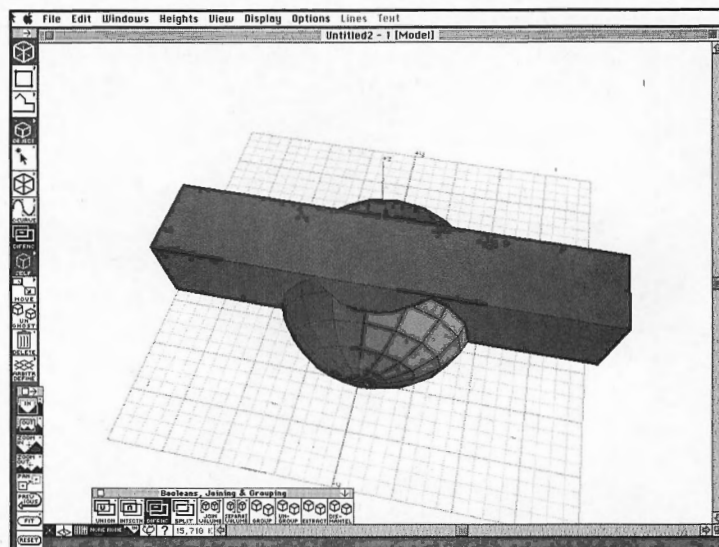
Booleans closely mimics real world manufacturing, where an object may be lathed, punched, drilled, stamped, and routed on its way through the machine shop. The following sequence (figures 5.74 through 5.77) shows a single sphere being carved up by two objects.



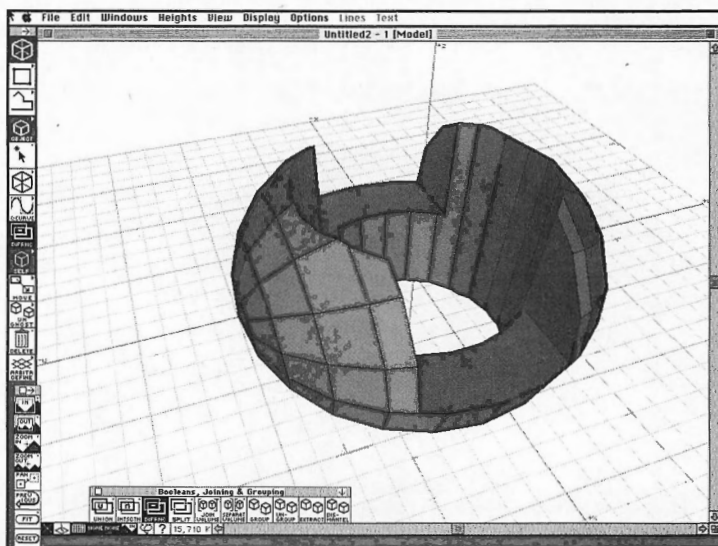
**Figure 5.74** The sphere and cylinder are carved simultaneously with the Split command—this cuts intersecting objects with their intersecting surfaces, but leaves the resulting parts unhidden



**Figure 5.75** Note the hole in the sphere and the “plug” with spherical end caps (the original cylinder ends are hidden)



**Figure 5.76** The rectangular solid will be subtracted from the remains of the sphere



**Figure 5.77** Subtracting the rectangular solid from the remains of the sphere results in this complex shape

## Living Forms

There are a few programs on the market specifically designed for modeling realistic living objects. There also are a number of sources of pre-built models including human and other natural forms.

### Humans and Animals

Macromedia Inc.'s Life Forms, originally designed for producing choreography, produces realistic animations of human figures that can be exported to other modelers and animation packages. This program, however, is limited to a single "model," which means your renderings will all begin to take on a familiar look if you frequently use this somewhat androgynous "person."

Life Forms is really more of an animation package than a modeler. See chapter 9, "Animation," for more in-depth information.

Of course, everyone would like to be able to create "Jurassic Park" on their Macintosh computers. In truth, the studios that accurately model human figures down to the twitch of a finger are using banks of high-end UNIX workstations and sophisticated digitizing stations that can handle human-sized subjects. They also are using teams of programmers and gigabytes of custom computer code. The complexity required to model and animate lifelike characters, more than any other reason, is why so many 3-D animators concentrate on vacant regions of space. While it would be great to model the animals marching two-by-two onto the Ark, or Lincoln delivering his address at Gettysburg, it's really much more practical to create geometrically-shaped robots and vaguely human-like androids. (Although the day when life comes to the Mac seems to be fast approaching!)

For creating realistic organic objects, such as people and animals, you'll eventually turn to a spline-based modeler. Be prepared for a difficult task: no one has ever satisfactorily animated human expression. Although the realistic animation of spline-based objects made inroads with the recent Macintosh release of Will Vinton's Playmation and Electric Image Animation System now has fantastic animated deformations, Macintosh software currently can't compete with workstation-level programs like SoftImage and Alias Animator.

### Playmation

Will Vinton's Playmation is a very unique product on Microsoft Windows machines, and the Macintosh version is *exactly* the same (down to the DOS filenames and the many files required to create a single scene). Problems aside, it works on the concept

of spines, muscles, and skin to create extremely organic and natural-looking characters in 3-D. It also uses sophisticated event-based animation and inverse kinematics.

This is not a “Mac-user-friendly” program, but people who have taken the time to use it generally like it. Its sole purpose in the world is really to do character animation, so it’s definitely not the right program if you’re planning on doing industrial design or lots of accurate mechanical-looking models. You’ll find that squashing, stretching, twisting, and bending are all very fluid in this program, but something as simple as drawing a square is a process requiring many steps.

## **Plants and Trees**

You may find it hard to believe that there are not one, but two, programs available specifically designed for generating 3-D plants and trees: AMAP and Tree Professional. While it’s possible to create very realistic trees with only texture mapping and simplistic models, sometimes realism calls for trees with boughs, branches, and leaves. Achieving this level of detail would demand many hours to generate a single half-believable plant. Both programs offer DXF export as their major method of file exchange, but AMAP also has direct file support for Turbo 3D, Abvent’s high-end modeling software.

### **AMAP**

AMAP is a “plant generator” by Abvent (France) and is available from View by View in the United States. It is used to build realistic 3-D models of many different kinds of plants, primarily for architectural renderings. But anyone doing 3-D graphics will appreciate the rich sense of depth that comes from having very realistic trees, bushes, and flowers in a scene.

AMAP uses a database of pre-defined tree types (including profiles of many kinds of flowers and bushes). It’s possible to set the leafiness, fruitiness, and complexity of the plants before the model is constructed. Because of AMAP’s range of plants offered, it’s a better program than Tree Pro if you need a lot of different kinds of plants. It’s quite expensive considering that it has only one purpose, but if you really need foliage, it’s a lot easier than modeling the trees by hand.



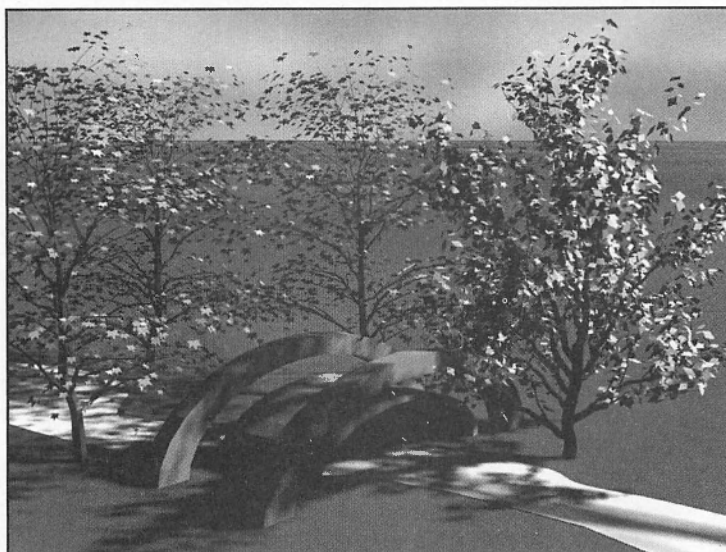


**Figure 5.78** *Bamboo trees and ferns generated in AMAP and rendered in Sketch!*

## Tree Professional

Tree Professional, from Onyx Computing, generates incredibly realistic trees, but it doesn't do flowers and shrubs like AMAP. Tree Pro gives you very detailed control over every imaginable parameter of a tree, such as the bend of the branches, the thickness of limbs, the crown height, and leaf color (it has such deep control that you can easily invent your own tree species). You can control the level of complexity in tree models themselves—most importantly, which parts of a tree you want to pare down and the level of detail you want to offer. The tradeoff is between realism and the complexity of the files created. A very large complex maple tree, for example, can easily top 50,000 polygons, which is more than most 3-D software will comfortably digest in a sitting. Fortunately, by selectively leaving out minute details, such as twigs between branches and leaves, and by using larger, less detailed leaves, it's often possible to trim your trees to about 5,000 polygons, while maintaining incredible realism.

While Tree Pro doesn't offer a rich variety of plant types, a library of predefined trees is available that sets the parameters to generate about 100 common species. The variety of trees in various stages of growth you can generate is infinite (both pine and broadleaf trees are supported). If you only need tree models, it's probably a better choice than AMAP.



**Figure 5.79** *Tree Professional trees animated and rendered in Electric Image Animation System*

## What Not to Model

Modeling can be a difficult, time-consuming process, especially if you're just learning! But there are many tricks and shortcuts.

Take the example of a dimpled golf ball. This shape is not only hard to classify—it's a sphere with lots of perfectly regular hemispherical dents—but it would be virtually impossible to model with a surface modeler. Even with a solid modeler, which could conceivably be used to subtract dozens of dimples from the surface of a sphere, creating a golf ball as geometry would be a challenge. So how come you see all of those realistic golf balls flying out of the TV on Sundays?

## Bumps and Textures

The answer is textures, in this case, bump maps. *Bump maps* can be thought of as "texture modeling." That is, they don't affect the geometry of objects to which they're applied, but they appear to do so. Bump maps are invaluable for adding texture detail to otherwise flat, plastic-looking surfaces. You can apply a grid bump map to a floor to simulate tiles, or you can bump map type on stone to create the

effect of etched or raised lettering. Bump maps are extremely effective when combined with visual textures. You also can combine a ripply bump map with a glass texture to create the effect of waves on water. (Bump maps are described in detail in Chapter 7, “Materials.”)

Another kind of geometry you can “model” with textures is transparency. Say, for example, you need a chain-link fence in your scene. You can create a 2-D image of a fence, and a *transparency map* that affects the spaces inbetween the wires, and apply this to a fence-shaped rectangle. For most purposes, this will look just as realistic as a modeled fence, but will require a fraction of the time to build and render.

Many textures, aside from bumps, can adequately substitute for all kinds of difficult modeling. Surface textures, such as a photograph of an intricate metal grate, a procedural texture, or wood grain, can provide much more detail than you’ll ever want to model. Environment maps also are an important consideration, because they save you from having to model the objects surrounding your scene.

## Background Details

Backgrounds, similarly, don’t need to be modeled as long as it’s okay that they remain a flat part of your scene. Often, photographic or painted backgrounds are used along with modeled backdrop elements to create a sense of depth and realism. For programs that support alpha channels (see chapter 11, “Working with Images”), backgrounds sometimes are not even a consideration in the 3-D environment. A 32-bit rendering with alpha channel information can be very simply composited in Photoshop or a similar program with a backdrop of any image. On the other hand, if you’re rendering objects with transparency, or you need absolutely perfect antialiasing, you will have to render with a background in place.

Animators may need to provide some level of modeled realism in their backdrops, particularly if long shots where the camera moves call for background objects to shift their relationships. However, animated flat backgrounds can do much of this work.

On the other hand, if objects are far enough off in the distance, you won’t notice a lack of depth—mountains, for example, don’t need to change position against the distant sky. In this case, only still images are required. In chapter 6, “Scene Building,” there’s a discussion of how you can match the perspective of your rendering to fit into a 3-D image, and how to cast shadows onto objects in a photographic background. If you do need 3-D models for backgrounds, there are several methods of generating 3-D terrains from grayscale images.

**Tip**

In animations, you can often use sparing levels of detail when modeling background and other non-essential objects. Since the viewer's attention will be keenly focused on the main subject, your time would be better spent on providing detail there. Often, animators use multiple copies of the main subject model (for instance, a flock of pterodactyls in flight). But to speed up rendering and reduce the level of complexity in your scene, you may be able to hide or even delete finely-modeled details in the "extras."

## Rendered Booleans

For users of RenderMan and Shade III, a good deal of "modeling" can be accomplished in the form of *Boolean rendering*. This is a rendering method that enables you to subtract or merge objects from one another during rendering. Shade III is unique in its capability to do this during ray traced renderings, while RenderMan offers the possibility of very elaborate effects with this technique.

## Displacement Mapping

In addition to Booleans, RenderMan offers "displacement mapping," which goes far beyond the capabilities of bump mapping.

While the illusion created by bump mapping vanishes at the edges of objects, displacement mapping enables RenderMan to actually move and sculpt a surface during rendering. For example, a sphere with a gravel texture can have clearly-defined pebbles appearing to cling to its edges. In animation, displacement shaders can be used to make surfaces ooze, ripple, or breathe. These effects are impossible to model (and they're one of the reasons movie studios use RenderMan for many of their special effects).

## 3-D and CAD

Industrial design, manufacturing, and architectural design typically require the use of sophisticated CAD (computer-aided design) software offering high degrees of precision. This precision enables engineers and designers to create objects to exacting specifications. Graphic artists, on the other hand, don't necessarily demand as much precision, so the question is: "How are 3-D CAD products different from, say, StrataVision 3d, and Sketch!, and how are they useful?"

In principle, there isn't much difference between a general modeler and a CAD application. A 3-D CAD application such as MiniCAD+ 5 enables you to work and build objects in much the same way as you would in a general 3-D program. However, while the interfaces aren't very different, CAD programs offer a long laundry lists of features, as well as providing access to all the numbers and pieces of data that are used to define objects. For example, while both a generalized 3-D program and a 3-D CAD program may enable you to draw an arc, the 3-D CAD program enables you to specify the exact radius and angle of the arc.

However, just as most generalized 3-D applications do not offer high degrees of numeric control, most CAD applications do not offer advanced rendering options. For this reason, engineers and industrial designers often export their geometry from their CAD program into a 3-D scene building program like StrataVision 3d for the creation of photorealistic renderings.

On the Macintosh, most 3-D programs can be either be classified as being generalized or CAD-like. In a few cases, such as Sculpt 3D and Presenter Professional, the programs can be described as being both generalized and CAD-like as they offer rendering and visualization tools, as well as precise, numeric controls.

MiniCAD+5, a 3-D CAD program, has features that include 2-D drafting with the capability to automatically create dimension lines and architectural cross-hatching. Operations like trimming lines, drawing double walls, and constructing fillets in corners, all benefit from special tools just for the job of creating 2-D plans. MiniCAD's 3-D features also are multitudinous. Walls, roofs, and floors are all provided by special tools. Window openings are created as soon as you drag a window onto a wall, and hybrid symbol libraries containing both 2-D and 3-D versions enable you to automatically drop cabinets, toilets, windows, and other features right into a scene.

Also typical of CAD packages, MiniCAD contains a database to which you can append all kinds of information for materials costs, serial numbers, sizes, and so on. As you build a project, you can refer to this database to see how much the project will cost or how many electrical outlets it will require. And finally, the program has a macro programming language that enables you to complete repetitive tasks automatically. For example, you may have a standard kind of paneling that needs to be applied to interior walls over 4' in width. You can set up a routine that will do this automatically throughout an entire building.

None of these features is unique to MiniCAD. AutoCAD, MicroStation Mac, ArchiCAD, Architron, and Vellum all have the same basic capabilities, although some, such as MicroStation and AutoCAD, offer very deep levels of customizability along with very complex tool sets. In fact, one thing that distinguishes all CAD products from general modelers is that the vast array of useful specialized tools results in an interface that takes a very long time to learn. While it's easy and

useful to jump from say, Sketch! to form•Z, to StrataStudio Pro to use their different features, you're very unlikely to see anyone jumping between MicroStation and AutoCAD in an average day.

This is one of the best reasons for CAD users to invest in a general modeler. Architects can often benefit from the simplified tools and speed of work provided by general modelers. For example, an architect may use Microstation or ArchiCAD for creating an initial design of a new office building, but furniture, lighting, and other fixtures may be much easier to create in a general modeler. It's a really good idea to investigate the file compatibility between your CAD program of choice and modeling and rendering programs before you invest in the software. MiniCAD+5, for example, exports files directly to StrataVision 3d and StrataStudio Pro. This means it's very simple to take files from MiniCad and render them with Strata's software. For software without this kind of compatibility, intermediate file formats such as DXF are essential, though not as easy to use as anyone would hope.

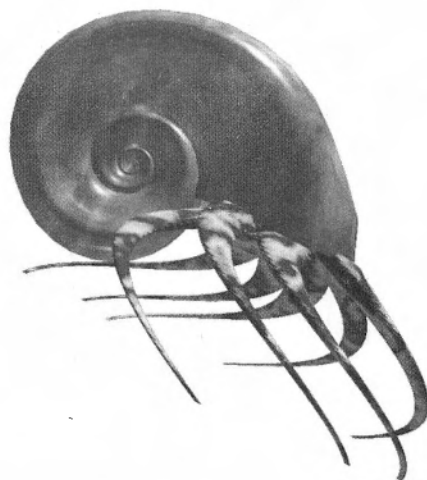
#### Tip

Kandu's CADmover software is a library of translators that can take files from virtually any 3-D program and convert them to a format useable by other 3-D programs. For users in this boat, it's better than a six-month supply of c-rations.

On the other hand, users of general modeling packages will sometimes need to create a very accurate or complex shape that is beyond the scope of a general modeling package. An animator friend of mine, who models mostly in Sketch! and form•Z, decided one day to build a cathedral on his Mac, a modest undertaking (to say the least) for which he turned to Archicad, explaining that in Sketch! it would take as long as it did to build the actual *Notre Dame de Paris*. Now that he's built it, he uses Electric Image to send light streaming through the lofty buttresses and arches (and refers to EI's visible spotlight as the "Hand of God" tool). On the other hand, the gargoyles in the cathedral, he builds in Sketch! and form•Z. If you need to create a western ghost town, a complex 3-D spaceship maze, or a very realistic Victorian apartment, you also may do well to turn to a CAD application.

## Splines vs. Polygons

At the simplistic level, splines are smooth, mathematically-defined curves and polygons are straight-edged facets such as triangles and squares. For beginning users, the differences are transparent; a sphere, for example, is a sphere no matter what the underlying geometry. However, the advantages and pitfalls of the two competing approaches become apparent with serious use.



**Figure 5.80** *Created with Presenter Professional's powerful spline tools*

Polygons, by their nature, are extremely efficient at representing straight-edged geometry, such as that which makes up the bulk of architectural design. Polygons run into trouble in the representation of curves, such as the forms that define most living creatures. In order to represent a smooth curve, a polygonal modeler has to draw many short segments or facets (a soccer ball, for example, is made up of many octagons stitched together).

The user can specify how smooth the curves will be by specifying that a circle should be drawn with a given number of facets—for instance 64 facets instead of 32. The more facets, the smoother the object, and the larger and more complex the resulting file. This results in increased RAM requirements and longer rendering times.

Splines, on the other hand, are extremely efficient at representing curves. A precise smoothly-curved spiral requires only a few points to define it. You can enlarge a spline-based object to any size without a loss of resolution; that is, the circle will continue to look like a smoothly rounded circle, no matter how large you make it or how closely you look at it.

Splines, however, have problems when you create objects that are primarily polygonal shapes. For one, they are harder to control than simple straight segments. Splines also impose an overhead in that they are not as efficient as polygonal systems at creating simple polygonal geometry.

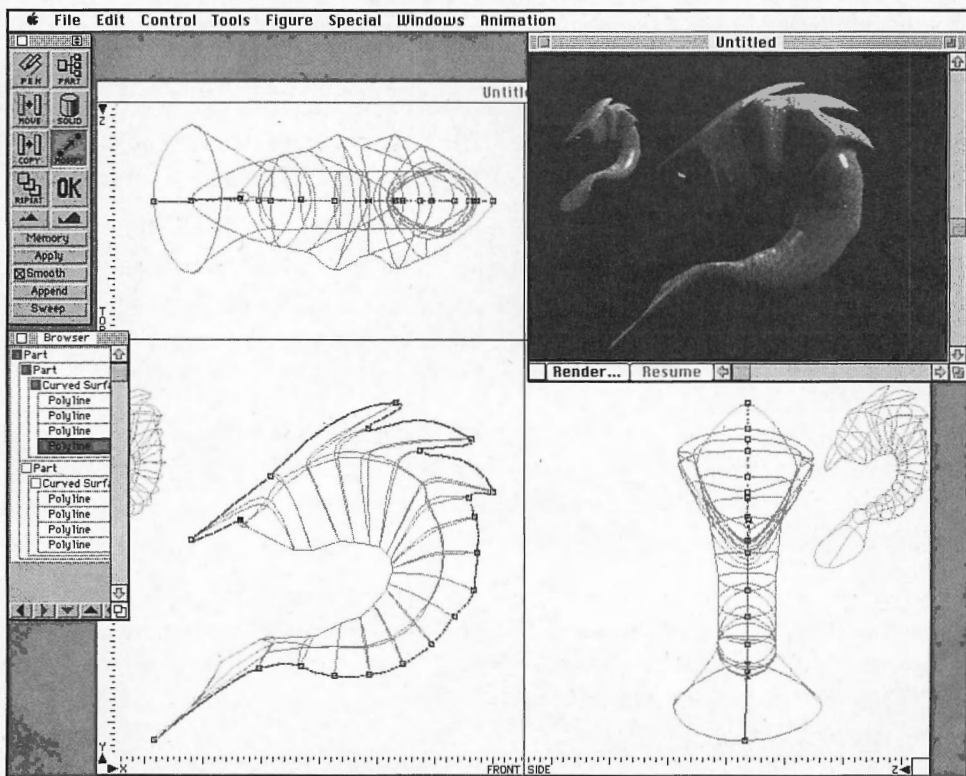
Splines come into their own when you want to do a lot of surface distortion. Because each spline point exerts a tension on adjacent segments, you can pull on one point and expect the surrounding curves to bend and change accordingly. In a



polygonal modeler, in contrast, pulling on a single point will tend to leave others in place (depending on the modeler), creating a sharp point or spike. For building anything smooth and organic looking, splines are a far better choice than polygons.

There is a trend in 3-D surface modelers, particularly those aimed at industrial design and graphics arts and animation, to forgo polygons in favor of splines. Modeler (and Presenter) Professional, Shade III, Sketch!, MacroModel, Playmation, StrataStudio Pro, form•Z, Zoom and PixelPutty all offer 3-D spline tools. Splines are popular because they make it easy to create realistic lifelike shapes and smooth flowing curves.

Polygonal systems, however, remain an industry standard. They include: Electric Image's Animation System, Infini-D, StrataVision 3d (and to a major extent, StrataStudio Pro), Design Workshop, Dyna Perspective, UpFront, Virtus vr and Virtus Walkthrough Pro.



**Figure 5.81** Shade III elevates spline modeling to high-powered simplicity—models are made of easily edited and merged “patches”



Imagine modeling a lizard's tail. If you pull sideways on the tip of the tail, you want the whole tail to bend smoothly, along a "spine." This simply isn't possible in a polygonal modeler.



c h a p t e r  
6

**Scene  
Building**

**A**nyone who has ever played with electric trains, a doll house, or toy soldiers understands the essentials of scene building. This is the phase of 3-D where you place your models on a stage, dress them up in textures, add lights, and adjust your camera so that everything appears in the proper perspective. While modeling is the most technically demanding part of creating 3-D graphics, scene building is the most creative.

*Scene building* is the application of textures, lights, and cameras so that the compelling 3-D rendering rises above the merely realistic.

Because scene building is such a specialized part of 3-D graphics, there are a number of programs specifically devoted to the task. These include animation programs such as Macromedia Three-D and Electric Image Animation System (EIAS), as well as programs that create only still images such as ShowPlace and Atlantis. In addition to providing tools to assemble and light the scene—as well as applying textures and environments—these programs include the rendering interface and engine used to create final images.

## Setting the Stage

The first order of scene building is the placement of models in a scene. In cases when the modeler and scene builder are the same application, this is merely an extension of modeling. We've already described the basics of navigation (see chapter 3); they are essentially the same in modeling and scene building—in many 3-D programs they are indistinguishable.

## Using Snaps and Grids

Scene builders, like modelers, rely on the use of snaps and grids: reference points that help the user place and align objects. Sketch!, form•Z, Designer, and MacroModel, for example, use the concept of a “working plane.” When using a floor as a working plane, you can place furniture as you would place real furniture in a real room.

## Constraining and Linking

Most scene builders—particularly those designed for animation—enable you to create physical relationships between objects. These relationships can be as simple as group hierarchies, or they can include complex spatial relationships, such as specifying how a ball joint can move within its socket. Typically, links can be established outside of the constraints of group relationships that were

established during the modeling stage. Although a foot originally may be part of a leg group, you may want to establish a link to a ball and chain, so that every time you move the ball and chain, the foot moves with it. In this example, the leg and the rest of the body may be linked to the foot so that when they move, it does too.

In most 3-D programs, group hierarchies are maintained when you import a model, but these can be rearranged. You can break a child off from a parent and link it to another object instead. Maintaining hierarchies is sometimes a trial-and-error

process. (Sometimes you want to send children to new homes, or promote children into parents.) How well hierarchies are maintained depends on how well your modeler and scene builder work together.

## **Parents and Children**

There are many instances when you will want to create parent-child relationships. As described earlier, this is a hierarchical link between two objects. Essentially, you use these relationships any time you want to move a parent and you want the child to move with the parent without changing their relationship. At the same time, moving the child does not affect the parent. You may want a car's wheels to be free to rotate independently of the car body, but you also may want the wheels to move when the car moves. If you were to merely group the car and wheels, rotating the wheels would also rotate the car! Even a light source can be the child of an object or camera; you can attach a spotlight to a camera, for example, so that when you move the camera, the spotlight moves with it. Or you can attach many lights to a spaceship so the lights travel with the ship but oscillate as the ship moves through space.

The difference between a simple group and a link is that links are like leashes that allow for the independent motion of child groups, while grouping objects is like gluing them together; when one moves so does the other.

## **Rotation**

Rotation of a child group in this scenario is relative to its original orientation to the parent group. This is important to consider when numerically transforming an object's position. If it is linked to another object, this transformation—unless otherwise specified—is usually relative to the parent, not to the world axes. Local rotation enables you to spin a propeller, for example, even while an airplane banks and turns through space. If you rotate the airplane, the propeller's orientation changes along with the plane.

Figure 6.1 shows the Orientation dialog box from Infini-D.

**Object Information**

Name:

	Position	Orientation	Dimension	Center
X	-1.947	-28.643	1.000	0.000
Y	1.417	35.401	1.000	0.000
Z	0.000	23.442	1.000	0.000

Uniform Scale:

**Options:**

☐ Invisible
 ☐ Force Backfaces  
☐ BBox only
 ☐ No Shadows  
☐ Fast Tree

Render Mode:

**Figure 6.1** Orientation, position, and dimension are set in Infini-D's Object Information dialog box

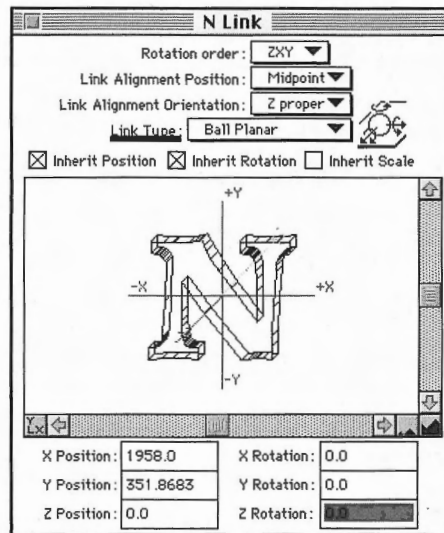
### Tip

When positioning an object in 3-D space, you must know where its reference point is located. A reference point is the point that is actually referred to when you look at an object's numerical position, and it is the point that the object rotates around (unless otherwise specified). Some programs, such as StudioPro, ELIAS, and Presenter Professional, make it easy to move this point in relationship to the model.

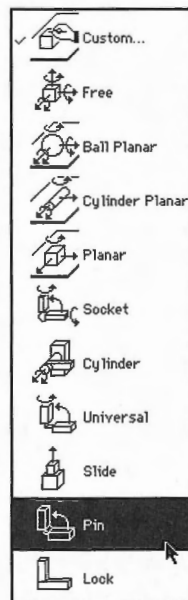
## Joints

Joints are special cases of parent-child constraints. Common examples of joints in the real world are the axles and bearings in bicycle wheels, the ball and socket in a hip joint, sliding drawers, and common door hinges. In all of these joints, the rotation and movement are constrained within certain limits (see figure 6.2). For instance, a hinge uses two pieces of metal folded around a pin. Each leaf of the hinge can rotate freely around the axis formed by the pin, but is prevented from rotating along other axes. Assuming one of the leaves of the hinge is affixed to an immovable object (such as the frame of a house), the entire assembly is also constrained to a single location in space. Of course, there is little need for joints in still images; you simply position objects and leave them in place for rendering. Joints come in handy as an

easy way to create convincing animations of constrained movement. Some of the types of links and constraints found in high-end animation applications are shown in figure 6.3.



**Figure 6.2** This image shows ELIAS' constraint window. The current object is constrained to move only with ball-type motion relative to its parent



**Figure 6.3** ELIAS offers many types of links and constraints; this kind of control is found in several of the mid-range 3-D packages—particularly those with event-based animation support

## Stretching and Squashing

Most scene builders enable you to enlarge and shrink, or stretch and squash an object. Uniformly resizing an object changes its dimensions relative to other objects in the scene, while stretching or squashing the object changes the dimension of an object along a single axis (you could flatten a cylinder to a disk).

ELIAS and Playmation both have extraordinarily cool stretching and squashing abilities that are crucial to character animation. Playmation's entire animation interface practically revolves around this capability. Not only can you bend, twist, bulge, stretch, and expand objects, but these change along spines that you designate; while the skins on the model behave in very skinlike ways, such as folding over when a model bends.

ELIAS also allows you to bend, bulge, and twist models to do sophisticated stretching and squashing. When you combine this with the program's particle effects and explosions, you can make incredibly realistic animations.

## Backgrounds

Few things in life appear against a perfectly black or white background. For this reason, all 3-D programs have some degree of support for importing or creating interesting backdrops for 3-D models. There are several ways to get backgrounds into a scene. The most common are importing a picture or movie as a flat backdrop, modeling a realistic 3-D background, or compositing your 3-D renderings after they've been rendered with a PICT background image.

An imported background is the easiest to create. You simply drop it into the scene before you render it.

On the other hand, rendered background scenery lends a strong sense of realism to an image (particularly if foreground lights and objects cast shadows into the backdrop, or if fog or depth-of-field are brought into play). The most common types of rendered background sets are rooms and landscapes, though you can easily create standard photo-studio-style seamless backdrops for things like industrial design renderings.

Imported backgrounds, or those composited with rendered objects after rendering, often look flat and "chroma-keyed" (much like the phoniest of backdrops in movie scenes). This is partly because backgrounds are not calculated in refractions when the transparent objects are rendered (which is always the case with post-composited images). More often, it's because imported backgrounds use



different lighting and they benefit from a different reflective environment than the foreground model. You can make up for this, to some extent, by careful lighting of the 3-D scene and creation of environment maps that mimic the reflections in the background image.

When rendering animations, backgrounds are particularly problematic because the motion of the background has to relate properly to the motion of the foreground objects for things to look right. If you're following a spaceship with your camera, the background needs to move in the right proportion to the foreground motion. This often requires a lot of trial and error.

A variety of very interesting options exist for creating unusual modeled backgrounds and scenery. This includes fractal terrain generators, which create convincing 3-D mountains, and bitmap-to-3-D converters that can turn a grayscale bitmap into a continuous surface model.

## **Rendered Backgrounds**

Rendered backgrounds are often as simple as a floor and a wall (with a strip of molding between them for realism). The advantage of this setup is that objects placed on this set will cast shadows onto both the floor and wall (assuming you placed your lights correctly). You can add a strong sense of realism to this scene by using a shadow gel on your main light to cast shadows onto both the foreground and background objects (see chapter 8, "Lighting"). To create even more realism, it often helps to apply appropriate textures (such as wood or tile to the floor, and stucco or wallpaper to the walls). To add a little clutter to rooms (to make them look more realistic), you may consider modeling or purchasing clip models of chairs and other furniture.

For animation, in particular, it's often desirable to create false 3-D rendered backgrounds. That is, you texture-map a background picture onto a plane or, better yet, the inside of a sphere. As your camera moves, this background will move as well, creating the impression of rendered scenery, but being much easier to build and faster to render. This works best for extreme backgrounds like sky, star fields, and distant mountains, where you can't perceive the fact that the background objects are all on the same plane. It's common to use a fairly detailed mid-ground object, like a mountain, with one of these pseudo-backgrounds in the distance.

When doing this kind of background, it's often advisable to turn off its specular and diffuse reflectance, and make it "glow" (combined with high ambient reflectance). This way, it won't be affected by the placement and movement of lights in the scene. It would look awfully fishy to have lights shining off a distant mountain range or the sky, for example.

**Tip**

Keep in mind that background scenery usually doesn't need to be as detailed as the scenery in the foreground. In fact, it's often better to maintain very simple, uncluttered backgrounds to help focus attention on the main subject. In these cases, the background serves to set the mood for foreground objects.

In cases where you need to add details to a 3-D space (but you don't want to model them), it is often possible to strike a compromise between rendered and imported backgrounds. For example, you can texture-map an air conditioning vent to the floor of a room—which is much easier than modeling one. For animation, it's possible to give simple backgrounds a realistic sense of depth by creating two background planes instead of one. For example, you can map mountains onto both layers, but make the sky in the nearest layer transparent. As you move, the foreground mountains will move in relation to the ones further back.

## Bitmaps to Models

Outdoor scenery is often much more complex than interiors. Fortunately, there are a gaggle of programs designed to make creating scenery a fun and interesting endeavor. Infini-D, StrataVision 3d, StrataStudio Pro, Sculpt 3D, and Showplace all have tools for generating fractal terrain models or other organic-looking objects based on 2-D images. Fractal terrains are particularly well suited to rough-looking cliffs and mountains.

Terrain generators generally work along the same principle: you take a grayscale image and convert the gray values in the image into heights. So, for example, a patch of white becomes a mountain top, while dark gray is a valley, and black is sea level—or even invisible. This is ideal for quickly extruding a scanned logo or other piece of artwork, as well as generating lunar landscapes. Unlike an Illustrator outline (which will be extruded to one height), extruding a grayscale bitmap enables you to extrude an image to multiple levels.

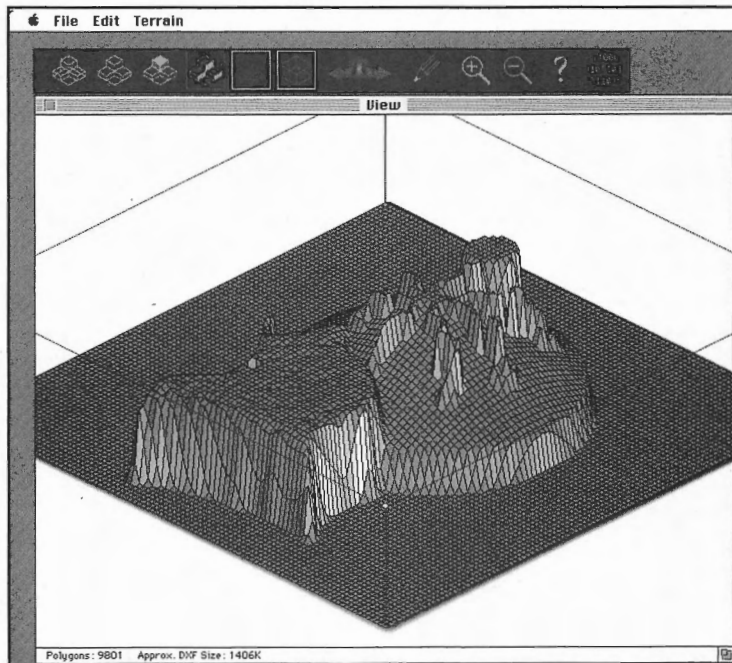
One of the beautiful parts of this approach is that as you're modeling, you can also create the texture map for the model. For example, you may want to create a rolling green valley that suddenly slopes up to a snowy mountain peak. To do this, you should create two versions of the same image: one that's a grayscale representation of the height values you're after, and another (based on the same image) that's a full-color texture map. After the model is built (and imported, if necessary), you can use planar or flat mapping to apply the texture map from the top. The color image should align perfectly with the terrain model you've created.

## Terrainman



Terrainman is “pay-another-guy-ware” that takes 2-D PICTs and “terrainificationalizes” them. One really nice feature is that you can simply drag on a bounding box to scale the terrain—you don’t have to type in numbers. Terrainman exports DXF files that you can use in almost any 3-D application.

The program was created by Joe Ashear, who graciously gave me permission to put it on the CD (and who also invented the catchy terms above). As penance for a checkered past of shareware (he’s not sure he’s paid for in full), Joe insists that if you use his cool program, you pay for the shareware. I’m repeating this message in my own small hope for absolution, and because the software certainly deserves to be paid for.



**Figure 6.4** *Terrainman not only creates great 3-D terrains from PICTs, but it’s also on the CD-ROM*

A new version of the program (to be posted on the Alpha Channel BBS as soon as it’s finished) will work with clipboard images, so that you can copy an image in one program, open Terrainman, and then generate a terrain (giving you almost instant editing powers with the paint program of your choice). It will also have lots of other cool new features, including support for many viewing angles and

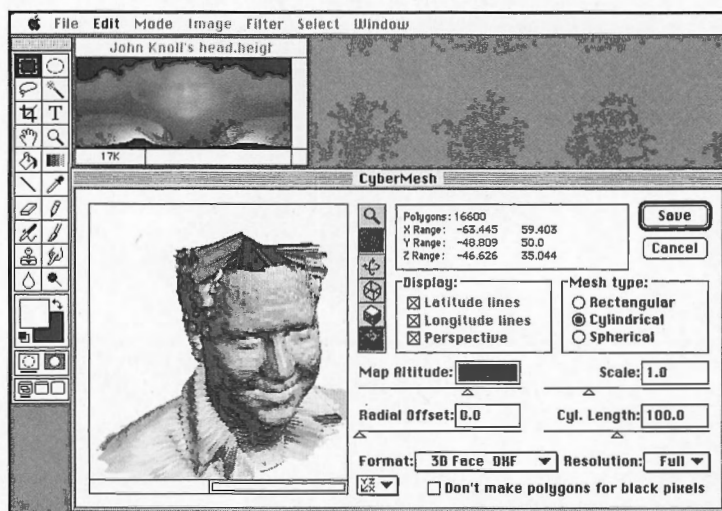
true perspective views. Joe has other plans for the program, which stands to make it a classic 3-D utility. Version 1.5.1 is included on the CD-ROM with this book. If you use it, please pay up to some deserving programmer, as Joe asks.

## CyberMesh



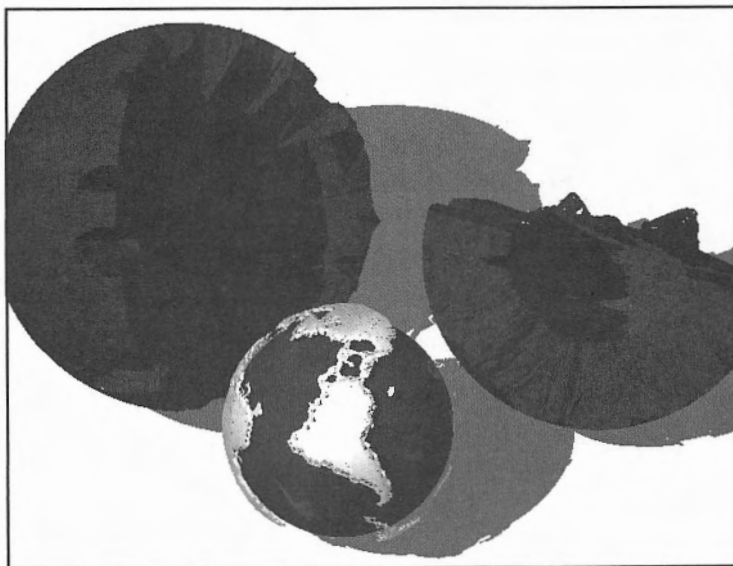
Knoll Software's CyberMesh is my all-time, second-favorite plug-in for Adobe Photoshop (next to Kai's Power Tools). CyberMesh's creator, John Knoll, shouldn't feel too bad about being in second place on my list, however, because he also is one of the creators of Photoshop, my all-time favorite piece of software—and an indispensable tool for any 3-D artist. Both Photoshop and Cybermesh demo versions are on the CD, *of course*.

CyberMesh does what other terrain modelers do, in that it converts grayscale images to 3D DXF. But it goes way beyond all the others—it can do this on a plane, as well as around a sphere or a cylinder, and it comes with an amazing level of control over file size and output format (it supports not just DXF polygons, but DXF polyline and Electric Image's FACT, as well). Figure 6.5 shows Knoll's own happy head converted from the grayscale image at the top, to the cylindrically mapped terrain in the main window. (I suspect, but haven't confirmed, that Knoll used his prodigious resources at Industrial Light and Magic to digitize his head, and that he has some "reverse-CyberMesh" utility for converting 3-D data into the grayscale image shown here.)



**Figure 6.5** A head for figures—John Knoll's software turns even strong men to mesh

Of course, the rest of us have to be content to use Cybermesh to create 3-D objects, rather than recreate them. Cybermesh has many interesting uses that just can't be duplicated with other software. Because it can map black pixels as holes, you can make things like half spheres with ragged edges and cylindrical polygon meshes filled with holes (see figure 6.6). For regular-old terrain, you can create fractal mountains that stand alone. There's really no other way to model this stuff, and you have to try it to fully appreciate it. (The first version was shareware called CyberSave, and wasn't nearly as amazing.) Knoll now charges a very minimal fee for the software, but you'll find it's worth it the first time you use it.

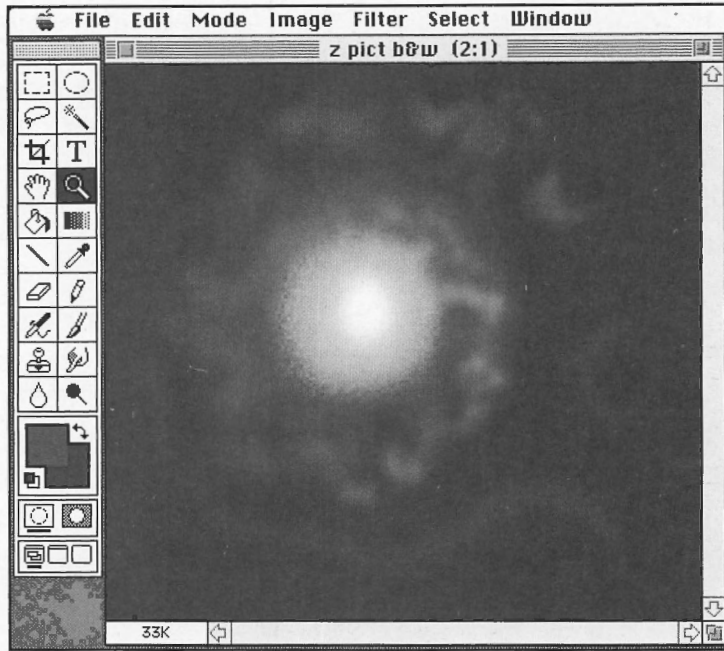


**Figure 6.6** *A nut full of worlds—modeled in Photoshop/CyberMesh, rendered in StrataStudio Pro*

### **An Island unto StrataStudio Pro**

Both StrataStudio Pro and StrataVision 3d automatically do terrain modeling when you import a grayscale PICT. (If the following example seems too mysterious, you may want to jump ahead to chapter 7, “Materials,” for an explanation of texture mapping.)

1. To create a 3-D island, begin by generating a cool color top view of an island. I used Kai's Power Tools to generate a fractal design, and then used Photoshop's tools to tweak it until the outside parts blended towards the middle. This was saved as a 400 pixels-square grayscale image (see figure 6.7).



**Figure 6.7** Create a grayscale 2-D terrain in Photoshop

2. In StrataStudio Pro, import the grayscale PICT file, which automatically converts it to geometry. Then switch to the top view (this will align it correctly) and drag a marquis with the Place shape tool to insert it into the scene (see figure 6.8).
3. In Strata's texture creation dialog, load the color version of the island PICT in the Color Map channel, exit the dialog, and then apply the newly created texture to the island by dragging it from the texture palette onto the model.
4. Open the Change Mapping dialog and position the texture on the model. Be sure to set the mapping type to Planar and the amount of tiling to None. You can periodically click the camera button to preview the results (see figure 6.9).
5. Close the Change Mapping dialog and switch the window to a perspective view. Click the camera button or choose Render from the menu to render the finished island. If the peaks are too tall or short, you'll want to switch to a front view to stretch or squash the shape down to a more desirable size (see figure 6.10).



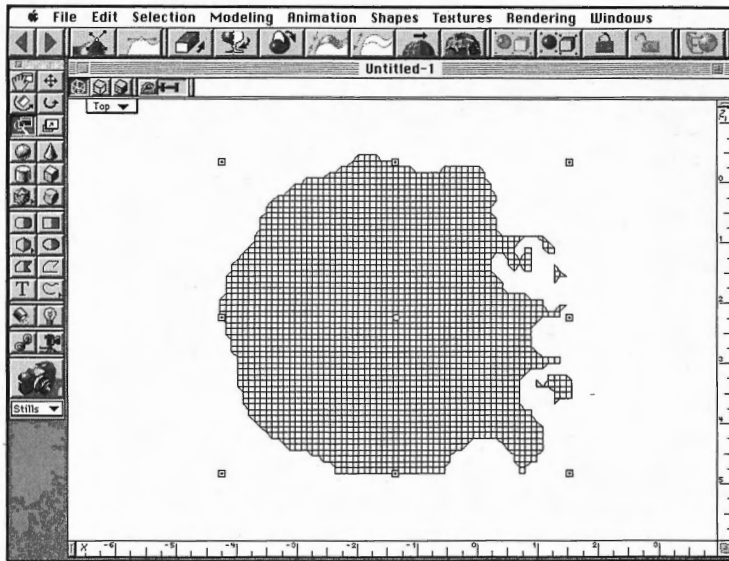


Figure 6.8 Place the island in the scene

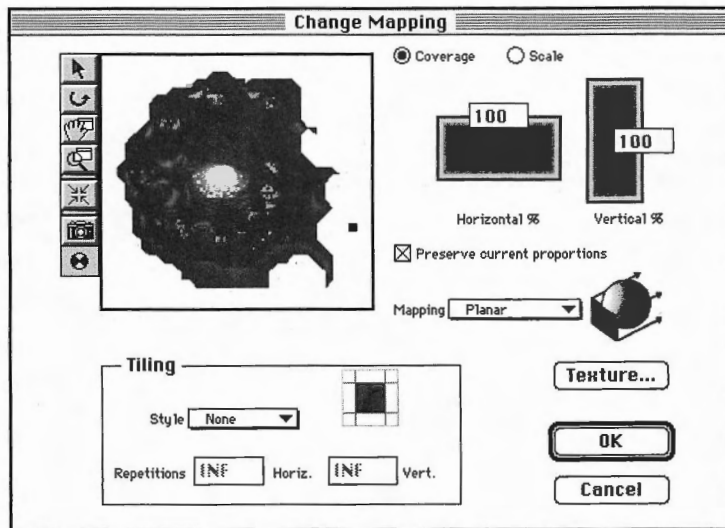
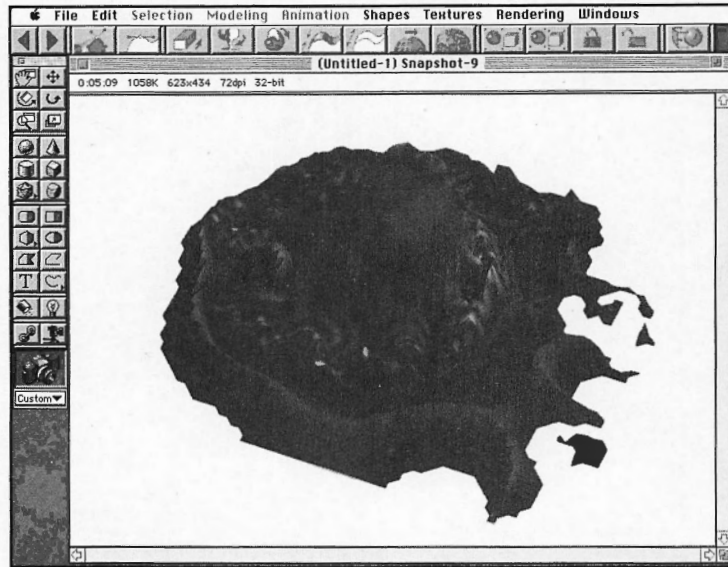


Figure 6.9 Map the texture onto the island



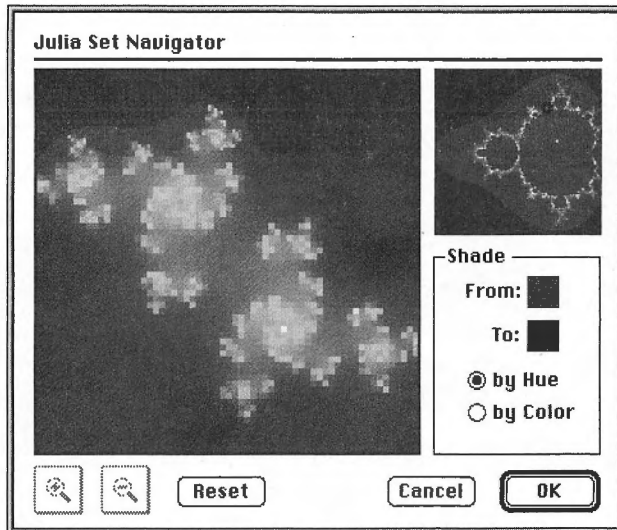
**Figure 6.10** *The island with texture applied*

## Infini-D

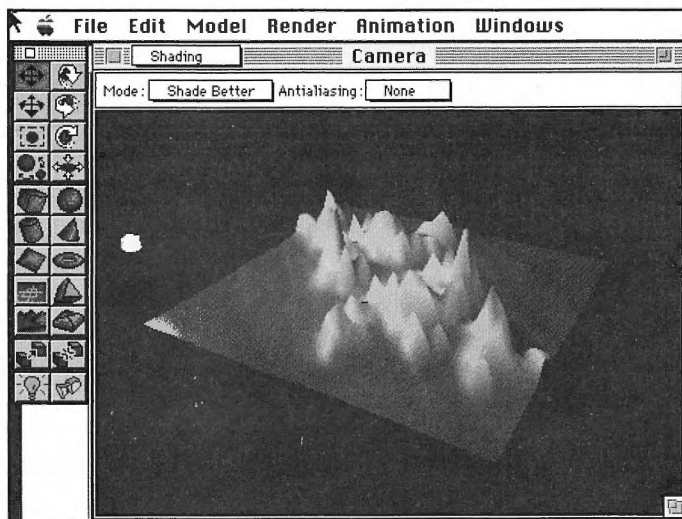
Infini-D has a very fast and easy tool you can use, called "Terrain," which generates convincing 3-D models from 2-D images. In addition, it enables you to generate fractal terrains from Mandelbrot or Julia sets, as well as automatically generating a variety of other interesting 3-D models from 2-D imagery. Fractal models are particularly useful for creating interesting mountainous landscapes (see figure 6.11). One really nice option in Infini-D's terrain tool is that it can automatically apply a color map to a terrain while it's being created.

Like StrataVision 3d and StrataStudio Pro, you can use a previously used image to create a model and apply it as a texture to that model (see figure 6.12).





**Figure 6.11** This interface enables you to select a small segment of the Julia Set to use as a model generator and texture in Infini-D



**Figure 6.12** The result after rendering

## Showplace Terrain

Pixar's Showplace has a Terrain generator much like that of Strata's and Specular's products (see figure 6.13). However, you do not have the option of importing an image to generate terrain. You can generate bump models from algorithmically generated pictures instead.



**Figure 6.13** *The Showplace Terrain generator*

## Imported Backgrounds

As we mentioned, every 3-D program allows some degree of importing backdrops. This means you can have a model of a spaceship against a starry sky without modeling a starry sky. (You simply draw one in a paint program or scan one from a public-domain NASA photo.)

Even though an imported background renders much faster than a modeled background, you will pay a significant time penalty for including it in your rendering. All renderers take a good deal of time calculating the appearance of a background—even if it is completely unaffected by the foreground image.

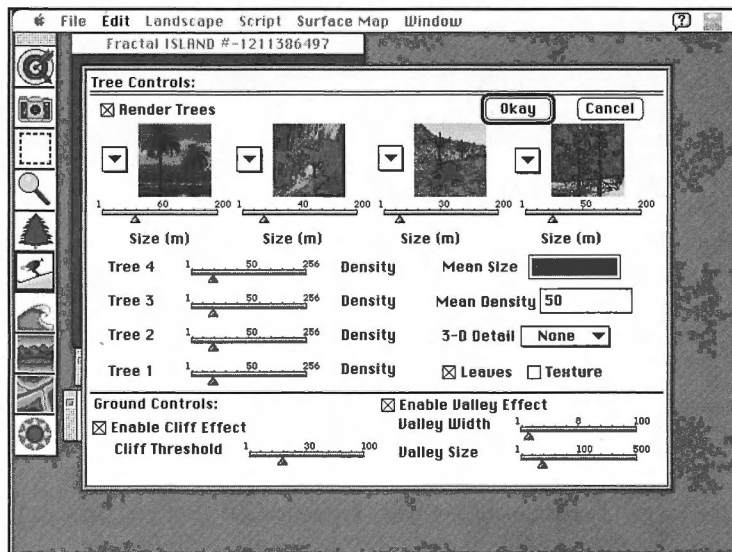
## Scenery Generators

Two brand-new products are on the market specifically for creating rendered background scenery (what will people think of next?). Vista Pro (from Virtual

Reality Laboratories) and KPT Bryce (from HSC, the makers of Kai's Power Tools, among other things) are both designed to let you build cool, realistic scenery using some standard—and some not-so-standard—techniques.

## Vista Pro

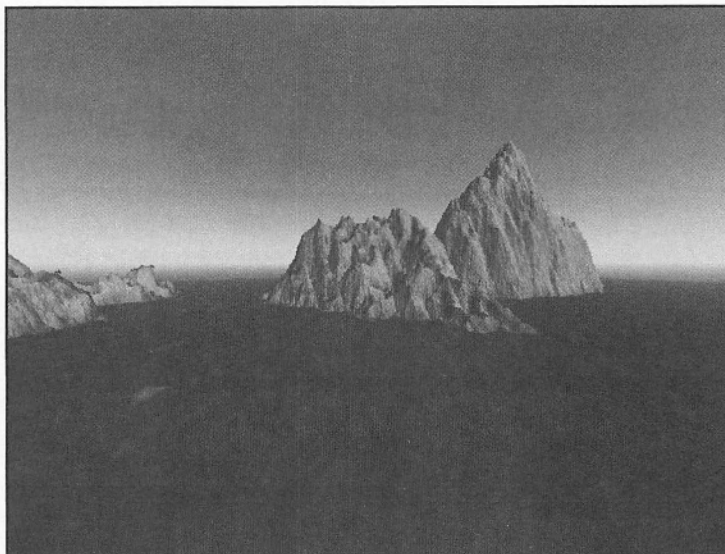
Vista Pro is a very elegant program that lets you begin with DEM geographical data (compiled by the U.S. Geological Survey) and generate very realistic scenery. The program comes with data for places like Yosemite Valley and other scenic wonders. You also can generate your own random landscapes. Vista Pro does lots of cool time-saving stuff, like automatic texture mapping and generation of trees, clouds, snow, and other natural features. All of these details are smart procedures—meaning that trees are constrained by altitude to form tree lines, you can set a snow level, and you can apply water that will fill valleys up to the level you specify.



**Figure 6.14** Vista Pro lets you set many parameters for automatic scenery effects

The program has a kind of cartoonish look and feel to it. The renderings it generates, although very accurate in detail, also have a not-quite-realistic look to them—they look more like highly detailed paintings than photorealistic renderings. The problem (which is also a good thing, depending on how you look at it) is that Vista Pro uses Gouraud Shading for its highest level of detail. This means that it doesn't render true shadows, and textures have a somewhat blurry look. On the

other hand, the program is really fast—you can render a very convincing 640 by 480 scene on a Quadra in about five minutes, and the Power Mac version is much faster.



**Figure 6.15** *Vista Pro quickly generates realistic landscapes and even islandscapes*

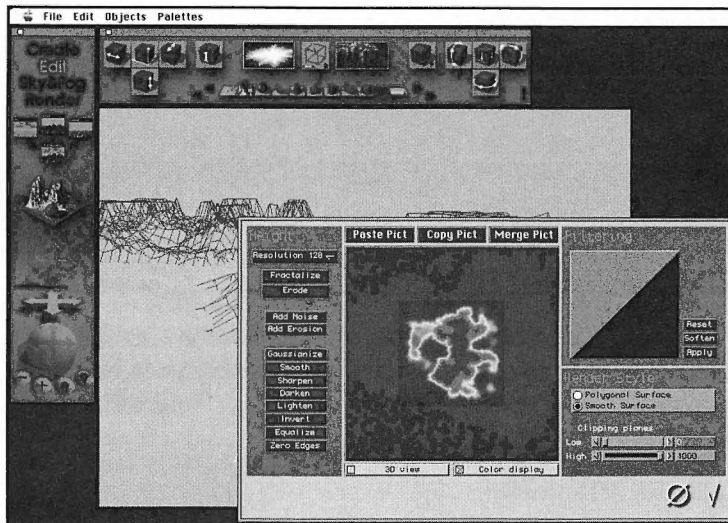
My favorite thing about Vista Pro is the capability to render actual places in very convincing detail. For example, it comes with before and after models of Mt. St. Helens. Vista Pro also exports DXF files, which means you can bring the scenery models into your other 3-D programs—a very important consideration in my book, since that's where my animation and rendering happen. Vista Pro will generate animations, but they're pretty crude compared to what you can do in a full-fledged animation program.

When you get tired of the included scenic wonders, VR Labs offers a library of scenery files you can purchase.

### **KPT Bryce**

Bryce was named after the national monument in Utah, and the eerie landscapes it creates suit its namesake. True to form for HSC, Bryce is a totally different way of doing things; the “KPT” designation indicates quite correctly that it's in the same class as Kai's Power Tools.

Unlike Vista Pro, which is oriented towards rendering real places with moderate realism, Bryce is aimed at rendering totally fabricated places with utter realism. Though Bryce is a very cool scenery generator, the tradeoff in this case is that it's also maddeningly slow. It only renders with ray tracing, so even 640 by 480 scenes have to be rendered overnight. When the Power Mac version ships, I'm sure this will improve immensely, but it's a dilemma for users of slower Macs.



**Figure 6.16** KPT Bryce has incredible control over the creation of alien worlds

Bryce has an amazing array of features, but the basic concept is to start with basic terrain objects from a library, and then modify them with filters, and even images, to create totally original terrains. Texture mapping and procedural textures are both supported, so that you can create very realistic clouds, fog, haze, and other environmental effects. It takes very little time at all to create totally realistic alien landscapes, islands, and mountains. The atmospheric effects really add a very deep level of realism.

Bryce has very cool terrain features, the best of which is "erosion." This will erode away surfaces you designate, so that you can create old crumbly mountains and weathered-looking landscapes. You can use images to apply erosion and other effects, and the changes are scaled by the gray-values of the image. Bryce also accepts plug-in textures and effects, so its creative future is limitless. If it weren't for the speed problem, I'd say this was a must-have program for creating scenery.



**Figure 6.17** *KPT Bryce has a wide array of terrain features*

One thing that would be really nice in Bryce, however, is the capability to export DXF and other 3-D model formats, since no landscape is an island.

## Transparent Backgrounds

A solution with many advantages over importing an image prior to rendering, is to render the image using the 3-D program's alpha channel setting, which creates a transparency matte. You can use this method for seamlessly integrating the rendered foreground with a background image. (The Alpha Channel is described in chapters 11 and 12, "Rendering" and "Working With Images," in much greater detail.)

One of the primary advantages of this approach is speed. Renderers take a long time to move the data in a background image through the system (and you're really not gaining anything by doing this, in most cases). On the other hand, creating an alpha channel barely slows down the rendering process. Combining images using the Alpha Channel in a program like Photoshop takes only a few seconds.

Another advantage is that you can try out many different background images until you find just what you're after. You also can move a rendering around over the background until you get it in the right position.

In addition, alpha compositing enables you to assemble many individual parts of a scene in one place. You can matte a 3-D logo over a 3-D backdrop, for example,



without rendering them at the same time. This enables you to build and render a perfect background, and composite it with the ideal logo (created separately) to get just the right effect. In chapter 8, “Lighting,” there’s a description of how you can use this technique to get perfect lighting on many objects in a scene.

The biggest disadvantage to rendering a transparent background is that transparent objects won’t transmit the background image. This is particularly crucial for refractive objects like glass, where refraction is as much a part of the surface quality as color or reflectivity. Also, be sure to set your background color to black if you’ll be rendering an alpha channel. This will maintain proper anti-aliasing when you bring your rendered objects into another program, while white backgrounds will cause a “halo” around composited objects.

## **Perspective Matching**

One of the most useful techniques in 3-D is creating and rendering a model so that it fits exactly into a pre-existing scene. What if you wanted to build a new structure in the midst of existing buildings? Ideally, you would photograph the existing site, scan the photo, and create and render a model that would fit seamlessly into the scene. Perspective matching is a technique that lets you do this.

### **Sketch!**

Alias’ Sketch! has a simple mechanism for perspective matching. You can use a perspective matching tool to drag the corners of an object to the corners of a corresponding object in the image. This changes the point of view of the camera so that the entire scene aligns to corresponding objects in the image.

When you are ready to render, import the scanned photograph of the background and, using the perspective matching tool, stretch the corners of the drawing grid to the corners of the vacant lot in the photo. This changes the perspective of the 3-D world to match exactly that of the photograph. When you render the model, it will appear to fit perfectly into place in the photo.

For example, you may build a model of a house on a rectangle known to match the lot dimensions in the photo. When you stretch the corners of the rectangle to match the corners of the lot in the photo, the model of the house and everything else in the scene are moved into this new perspective.

## General Perspective Matching

Any program that lets you view a background while you build a scene can be used to match the perspective of a photographic background.

The way you do this, without Sketch!'s tricky tool, is to model an object that corresponds to a known object in the background scene. In general, it's easiest to do this with a floor plane. Then you drag your view around—zooming, panning, and rotating until the known object lines up properly with the background. This can take a bit of trial and error, but it is not too difficult.

If accuracy is crucial, you can photograph the site from a measured distance, taking careful note of the camera lens length and the angle and elevation from which you are shooting. Programs that support moveable cameras enable you to recreate the camera's position in your model. This technique takes careful planning and a lot of time, but can yield perfect results. Usually, once you've rendered a model and composited it with the background, you'll want to do some detail work, such as placing foreground details and using Photoshop or ColorIt to paint shadows into the scene.

### Tip

One company, View by View, does amazing architectural rendering using helicopter fly-bys to shoot videos of a site. View by View aligns its architectural models to the digitized video by pinpointing recognizable objects that they've modeled into the scene (such as lamp posts) and aligning these manually at animation key frames. The results are highly accurate and incredibly convincing.

## Shadow Casting on Backgrounds

Sketch! has another really cool capability called Matte Object. This lets an object be invisible, yet catch shadows normally. After you've aligned your scene to the perspective of the background photo, you can place a matte object table exactly where a table is (in the background, for example), so that the rendered image will have a shadow in proper perspective falling on the photographed table!

You can make Matte Objects out of any modeled object, so that you can have shadows realistically cast onto all kinds of background scenery. This is important



for realism because shadows wrap around curved objects or bend at intersecting planes (such as where a floor meets a wall).

## **Cameras**

As discussed in chapter 3, “Navigating in 3-D,” cameras can be used to provide new views into 3-D space. When creating a final scene or animation, lens length (zoom), orientation, and sometimes depth of field become particularly important.

### **Orientation**

The orientation of the camera is critical to getting the picture right. You can aim a camera at any object, but you also will need to concern yourself with the roll of the camera (how much it’s tilted left or right), as well as its elevation.

### **Depth of Field**

Some renderers offer depth of field. This is an effect found in cameras, whereby the “f-stop” (or opening of the lens) determines how much of the scene will fall into focus. Objects that are too close or too far from the camera are increasingly blurred as they fall out of the focal range. As you open up the lens, the depth of field decreases. This effect is primarily useful when you want your rendering to mimic the appearance of a photograph.

The default setting for 3-D software is usually infinite depth of field (which mimics the way our eyes focus on objects in the foreground or background).

## **Atmosphere**

Atmosphere is a tricky thing to create in 3-D. Most of the time, atmosphere is thought of as clean, invisible air—until you think of terms like pea-soup fog, drizzle, haze, smoke, and puffy clouds. All of these are visible examples of atmosphere, and there are many others.

It is possible to model a tornado seen from a distance. (Make sure that you use lots of random objects flying around.) Modeling an effect like smoke is more problematic: smoke twists and curls as it rises and, at some point in its ascent from the source, breaks up into mist. This is an extremely difficult model to build.

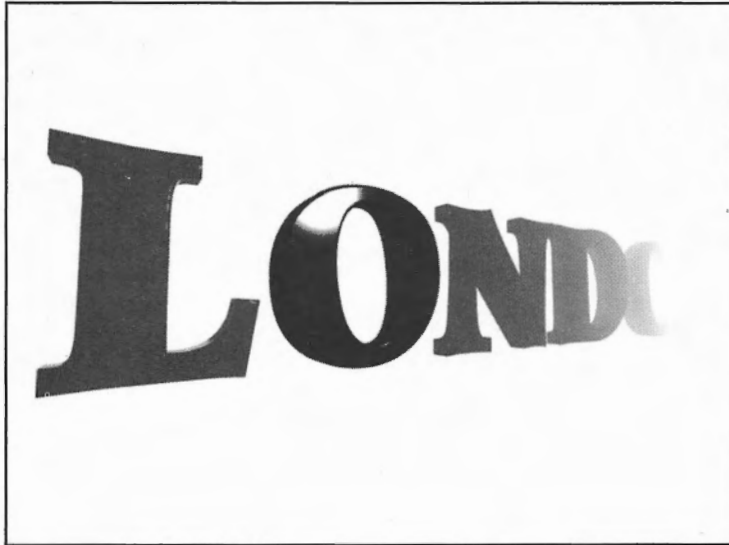
You could try building a texture—part transparent, part smoke colored—and applying it to a twisted object in the scene, but this is a tricky solution, particularly in animation. Take the example of rain; we all know what a rainy day looks like, but what does rain itself look like? You could build thousands of tiny droplets and scatter them throughout your model, but this is likely to overwhelm a renderer.

While some programs are beginning to offer cool atmospheric effects—Typestry has bumpy fog and Strata has clouds—it's often possible to use simple models in a scene and mimic atmospheric effects by texture mapping a combination of transparency and smoky glows onto the model's surface (see chapter 7, "Materials").

## Fog

One of the few atmospheric effects adequately simulated by 3-D programs is fog. It can be used to portray foggy days, but more importantly it provides *depth cueing*. When objects disappear into a distant haze, it is a strong visual clue that they are really far away; this effect enhances the sense of depth in a scene.

Unlike real fog—which is subject to the irregularities of clouds—3-D fog is an even blurring of the scene as you move further away from the camera. Fog uses a color that is increasingly dominant as you move farther away in the scene, until the background is uniformly colored (see figure 6.18). Usually, you can set the amount of fogginess, as well as the point at which you want the fog to begin affecting the scene. Thus, if your modeled airplane is 20 “feet” away in the 3-D space, and you want the fog to begin behind it, you can set the fog to begin at 30 feet. This will cause anything behind the airplane to be lost in fog: the farther away, the more it will be obscured. This is an extremely effective technique for creating the effect of distance—especially when you have scenery such as mountains, which in real life are often partially obscured by the atmosphere. Most high-quality 3-D renderers offer fog.



**Figure 6.18** This image was rendered, with lots of fog, in Electric Image Animation System

**Tip**

Unlike real fog, 3-D fog is not affected by lights in a scene. You cannot, for example, shine a red light into a fog bank and have a red streak show through the fog. To create these effects, try mapping “foggy” glows onto transparent objects. On the other hand, fog does have a color. So, if you’re showing fog at sunset, you can give it a pinkish cast; fog in the afternoon will tend to be an even light gray or white; and fog at night fades to black.



# c h a p t e r 7

## **Materials**

In this material world, it's no wonder that 3-D programs give you so many exotic substances to work with. It was the alchemists' calling to turn lead into gold, but 3-D enables you to pick your precious substances from a menu.

While it's easy enough to say that something is "made out of gold," many factors affect the surface appearance of an object. The following is only the short list:

- **Surface quality.** This includes base color, reflectivity, transparency, refractiveness, smoothness, shininess, and glow.
- **Color texture.** This includes variable or patterned colors, and the position, direction, and scale of patterns "mapped" on a surface. Typical textures include materials such as bricks or fish scales. In addition to covering the entire surface of objects, textures can be precisely mapped to specific locations. You can, for example, place a wine label on a bottle, or a photograph in a picture frame.
- **Solid texture or grain.** Picture a cut piece of wood. Along its length, the grain is straight and roughly parallel; on the ends, the grain is in concentric circles. This quality also is found in some stones, such as marble, and in textural solids, like styrene and swiss cheese.
- **Surface projection.** This property controls the manner in which surface textures are applied. Like latitude and longitude lines on a globe, the surface textures may shrink as they approach the poles; or like the label on a cola can, they may remain proportional all around. It's also possible to map a texture flat onto an object, so that it sticks to it like wallpaper, or to apply it in flat rectangular faces, such as the surface of a cardboard box.
- **Bumps.** If the whole world were smooth and glossy, everything would look like plastic. Bumps give surfaces a tactile texture. Without bumps, if you were to wrap a photograph of bricks around a perfectly smooth surface, the smooth uninterrupted gloss of the surface would appear like brick-colored plastic shelf paper. Bumps can be very regular, like the spaces between kitchen tiles or the embossing on the face of a coin, or they can be irregular, like pebbles or stucco.
- **Displacement.** When using RenderMan, a texture can actually displace an object's surface. This is a more dramatic and more tangible kind of texture than simple bumps.

Most 3-D programs make it fairly easy to use materials. Normally, it's merely a matter of selecting an object or group and making a few menu selections or clicking a palette of texture thumbnail previews. Many textures lend themselves to detailed editing and control, but often, using textures is quite simple. In most cases, you can import PICT images for use as textures, so it's easy to scan a photograph of a real-world surface to add to your materials palette. There also are many commercially available texture collections.

The exceptions to this rule are the PostScript 3-D programs that enable you to map PostScript line art (but not bitmapped artwork) onto a surface. RenderMan also imposes unusual requirements and limits on texture mapping.

## **Surface Quality**

Have you ever wondered why it's possible to look at an object and know with relative certainty what it's made of? We have come to associate certain surface qualities with particular materials. For example, a glass vase may have a very deep blue color. But if we look at it more closely, we'll find a more complex assemblage of qualities. Overall it is very dark. It doesn't reflect much of the ambient light in the room, and its deep blue color only shows in very bright direct light. There are very bright, sharp reflections of windows and other bright objects in the room on the vase's surface. But we also can see some of the internal structure of the vase, as well as some of the room's details on the other side of the glass, which seem bent and deformed as if seen through a lens. The vase is smooth and glossy, but it does not give off any light of its own, so it can't be said to glow. (However, if we heated it to a very high temperature, it would glow bright red, then orange, and finally white hot.) All of these properties make up the surface quality of an object. While we usually don't think about these properties, their presence or absence provides us with the clues we use to tell what the objects are made from.

When building objects in 3-D, you'll have to give some thought to the surface qualities of real-world materials, so that you can mimic them to create realistic textures. Primarily, the surface qualities of an object define how it reflects and transmits light. These include color, mirror reflection, diffuse reflection, specular reflection, ambient reflection, glow, transparency, and refraction.

Every material has a distinctive set of surface qualities that makes it unlike other materials. (This is independent of texture, which will be discussed in detail in the next section.) Consider that there are hundreds of kinds of plastic, dozens of variations of glass, and paints that achieve many different qualities. By experimenting with surface quality settings, you should be able to achieve the surface properties you need.

When children first learn to draw and paint, they begin to identify materials by their color—a tree is green, a house is brown, a dog is black and white. But even most adults will have a hard time telling you what color chrome is. (Try asking a friend to identify the color of a mirror or of clear glass.) In fact, such objects are colorless, or nearly so. Their surface qualities are defined by other characteristics, such as reflectivity and transparency.

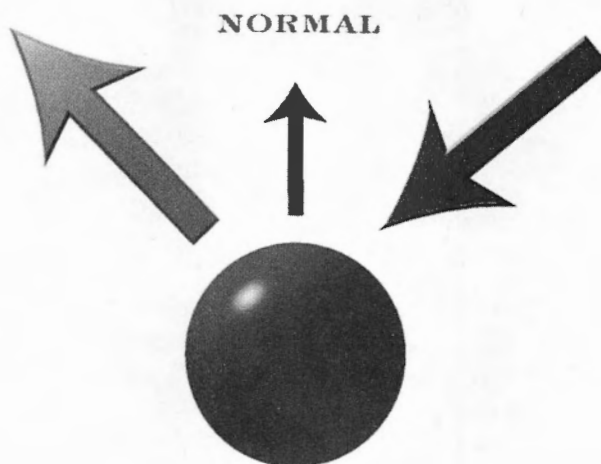
## How We See

We see most objects in terms of the light they reflect. Light from a source, such as the sun, strikes a surface, is changed to some degree, and is reflected towards the eye, which reads the light and sends the information to the brain. The 3-D world is modeled after this simple process; the only difference is that the process is calculated in reverse. The computer doesn't calculate where all the light from a given source goes. Most of that light would be reflected away and would never reach the viewer anyway. Instead, the computer traces the light rays backwards, calculating exactly what the eye sees and tracing these light rays back to the light sources. However, the end result is nearly the same either way.

The way that light is reflected from a surface, and the degree to which it changes in the process, ultimately determines the appearance of a surface. Imagine throwing a rubber ball at a 45-degree angle to the ground; a person positioned to catch the ball after it bounces would "see" the ball's reflection. The angle at which light is reflected is relative to the surface *normal*, an imaginary line that is exactly perpendicular to the surface at the point of impact.

In the case of a perfect mirror reflection (see figure 7.1) the angle that light makes relative to this line as it strikes the surface is the same both coming and going. With other materials, this angle can change as light is scattered or deflected.

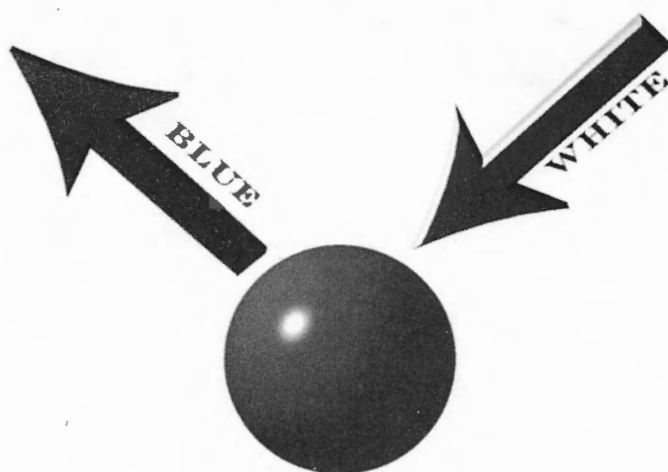




**Figure 7.1** Light from a light source, such as a lamp or another object, strikes a surface and reflects toward the viewer—when the light is reflected unmolested, we see a mirror image

## Color

Color is the most basic surface quality. The color of a surface usually is described as the light it reflects when illuminated by diffuse white light (see figure 7.2). Since white light is made up of all the visible colors in the spectrum, a material's color is usually the sum of all the colors that it doesn't absorb.



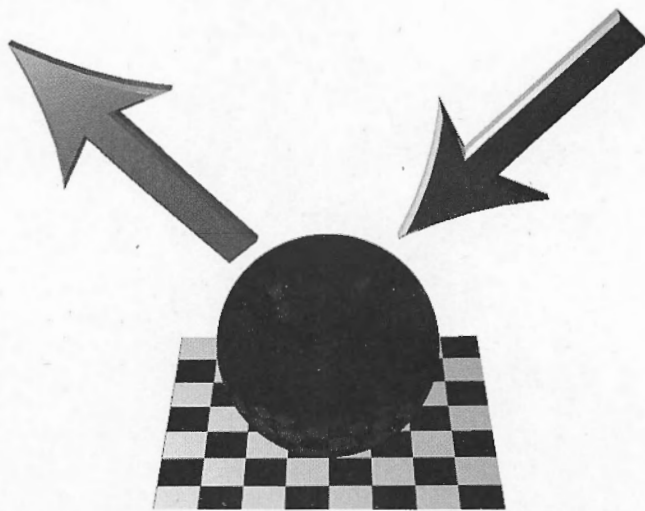
**Figure 7.2** White light strikes a surface and only blue light is reflected—this, by definition, is a “blue” surface

In situations where the light is of a single color other than white, the object may reflect no color at all, appearing black or nearly so. The color of a material is sometimes obscured by reflection of directional light or diluted by transparency. Thus, it's not always easy to determine the true color of a surface by looking at it. For example, a wafer-thin plate of dark blue glass may appear to be very pale blue, while a thick plate of the same material may appear almost black in equal light. Meanwhile, blue plastic with a matte finish that is non-reflective and completely opaque will look "blue" in most lighting conditions.

Astronomers and chemists, in fact, know that certain materials reflect and emit many different colors of light, depending on what's happening to them. They use spectrometers and other color-sensitive devices to locate and identify different materials, in both outer space and in test tubes.

## Mirror Reflection

*Mirror reflectance* (often called just "reflectivity") is a special case of directional reflectance: when light is reflected at the same angle and intensity relative to the surface normal and when the reflected light is non-diffuse, the result is a mirror image (see figure 7.3). Unlike dull-finished metal, a mirror is very smooth, and even weak directional light rays bounce off in an orderly fashion. Mirror reflection is the extent to which all light is reflected in a coherent manner from the object's surface. The capability to render mirror reflections is the main reason to use ray tracing.



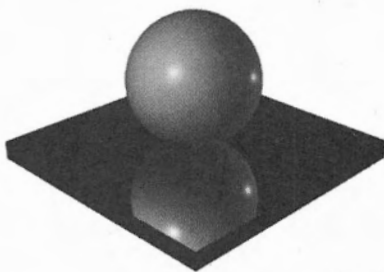
**Figure 7.3** Light striking a mirror is reflected in a coherent manner—the path of light to and from a reflective surface accounts for the fact that reflections are inverted when they reach the observer

An object with a very high degree of mirror reflectivity will reflect even dimly-lit objects, while an object with low mirror reflectivity may coherently reflect only the brightest parts of objects. Objects with high mirror reflectivity tend to be highly polished and hard, although some liquids, such as water, also can be highly reflective under certain conditions. Usually, you can express mirror reflectance in terms of a percentage: a looking glass reflects very close to 100 percent of the light that strikes it, while a plate glass window (which isn't coated with silver on one side) reflects around 10 percent.

Some surfaces are highly reflective at oblique angles—water, for example. If you look straight down into a fountain, you are likely to see all of the pennies at the bottom, but if you look towards the far edge of the fountain, you may see only a reflection of blue sky. This is generally also true of glass. Though the physics are different, this is like skipping a stone on water. If you throw the stone straight down, it will splash and disappear, but if you skim it across, it will glance off the surface and sail back into the air.

Some materials alter the color of light, even as they reflect it. A mirror made of polished copper, for example, will reflect a scene accurately, but will give the scene an orange cast. (While this is true in the real world it may or may not be true in your 3-D program. Sometimes you have to use tricks to simulate these effects.)

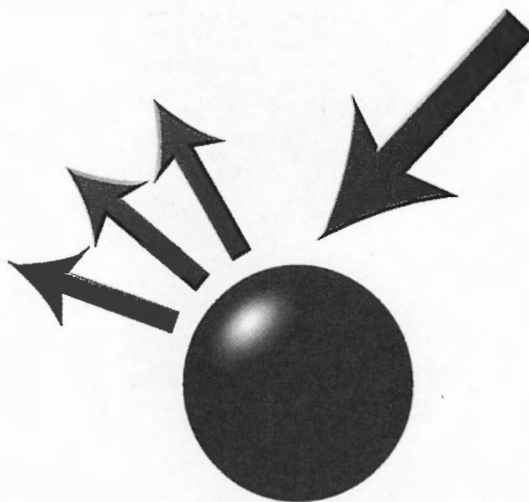
Highly reflective materials include mirrored glass or plastic, smooth chrome, silver, and silver mylar. In general, the surfaces with very good mirror reflectance have very low diffuse reflectance and tend to be much harder and smoother than highly diffuse objects (see figure 7.4). Objects with only a small amount of mirror reflectance include polished marble, white china, a television tube (when it's turned off), and glossy Scotch tape. If a surface is highly reflective, but also reflects a high percentage of diffuse light, it won't have that "mirror" look.



**Figure 7.4** This table has high mirror reflectivity and low diffuse reflectivity—note how this makes the table look hard and polished, while the sphere looks soft

## Specular Reflection

*Specular reflection* (sometimes called *glint* or *highlight*) is the reflection of a directional light source, near to but not quite in the mirror direction (see figure 7.5). The effect is that of a blurred mirror reflection. Often, it shows up as a round “hot spot” on a surface. Light sources such as square windows will cause a square specular highlight on a metal surface (although some renderers fake this effect and always use a round highlight).

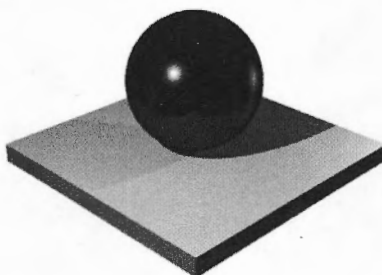


**Figure 7.5** *Specular reflectivity reflects light slightly scattered from the mirror direction*

Some surfaces, such as glass and plastic, have a specular quality that reflects light in its original color. In these materials, shining a white light on the surface results in a white highlight. A quality of metals, on the other hand, is that their specular highlights reflect the base color of the metal. Shine a white light on a 24 karat gold sphere and the specular reflection will be brilliant yellow. This is one of the distinguishing properties that defines the appearance of metals. Shine a white light on carbon steel (often called “blue steel”) and the specular highlight is blue.

To some degree specular reflection retains the shape of the light source distorted by the reflecting surface. A square light is reflected as a square highlight. This effect

is often seen in fashion photographs of people, where one or more ring-shaped or umbrella-shaped lights are reflected in the model's eyes. The smoother and harder the surface, the smaller and more perfect the specular reflection. For example, figure 7.6 shows a sphere that appears to be very hard and smooth because of the small, bright highlight.



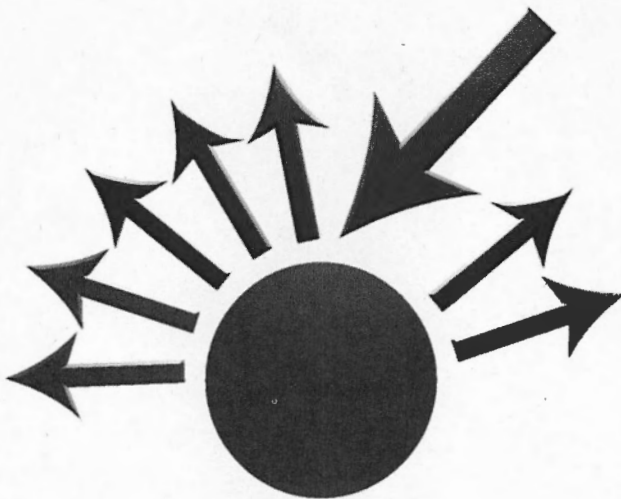
**Figure 7.6** *Specular reflection is responsible for the size and accuracy of the highlight reflection*

While specular reflection is a disorderly case of mirror reflectance, even renderers that don't use ray tracing allow for some degree of simulated specular reflection, usually in the form of a round highlight for each light source. Without specular highlights, surfaces tend to look soft and dull, while specular highlights are a strong visual clue to the character of a surface.

## Diffuse Reflection

*Diffuse reflectance* describes the amount of light that a surface scatters in all directions (see figure 7.7). High diffuse reflectance gives a surface the appearance of "lightness."

This type of reflection occurs when light penetrates a surface slightly and then is scattered back out in a random fashion. Objects with a high degree of diffuse reflectance are very evenly shaded from highlight to shadow. Objects with low diffuse reflectance appear hard and smooth and have a very abrupt transition from highlight to shadow. Low diffuse reflectance usually accounts for "dark" surfaces. Chalk, concrete, and leather are all examples of high-diffuse materials, while black velvet and polished ebony are examples of low diffusion textures. Diffuse reflection is primarily responsible for the color of objects.



**Figure 7.7** Diffuse reflection. Light striking a surface is widely scattered—the result is a uniformly blurred or diffuse surface

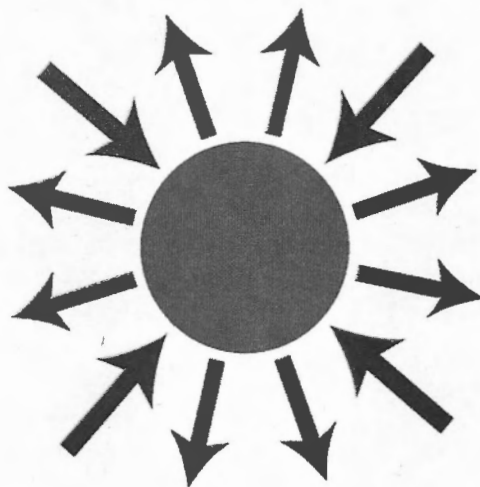
### **Ambient Reflection**

*Ambient reflectance* is the degree to which a surface reflects incidental light coming from every direction (see figure 7.8). An object with high ambient reflectance will tend to have soft or non-existent shadows, while one that does not catch ambient light will have deeper, darker shadows.

In the real world, particularly during daylight hours, there is a tremendous amount of ambient light being scattered around. That's why you can see clearly in the shade of a giant elm tree on a sunny day. Some objects, such as the pages of the book you may be reading under the tree, have a very high ambient reflectance; that is, they easily scatter and reflect even faint indirect light. On the other hand, some objects show their colors only when lighted by strong direct light.

If very even, incidental light coming from every direction can light the surface, it has high ambient reflectance (paper, cotton, and white paint are common examples). On the other hand, if the object is nearly black unless held in direct sunlight, it has a low ambient reflectance (for example, dark blue glass, or glossy black paint).

Mist—such as fog, clouds, and smoke—also have high ambient reflectance.

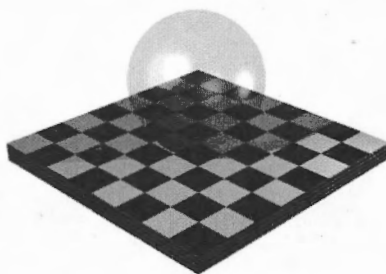


**Figure 7.8** Ambient reflectance—the degree to which an object reflects incidental light

### Transparency

*Transparency* is the capability of light to pass through a surface (see figure 7.9).

Glass, air, and water are all highly transparent. Some plastic also is transparent, as are many gemstones, including diamond. A material that is somewhat transparent but is too opaque or deeply colored to transmit light clearly is called *translucent*. A tree leaf is translucent; so is a sheet of paper, colored plastic, glass, and the skin between your fingers, if you hold it up to a bright flashlight.



**Figure 7.9** Transparency is the transmission of light through a material



As with reflection, transparent materials often change the color of light as it passes through. Stained glass is an attractive example of this effect. Try holding a colored bottle between a white wall and a white light and note the color of the shadow that strikes the surface. (In real life, the glass absorbs certain colors and transmits others; renderers simulate this effect by changing light's color and intensity as it's transmitted.)

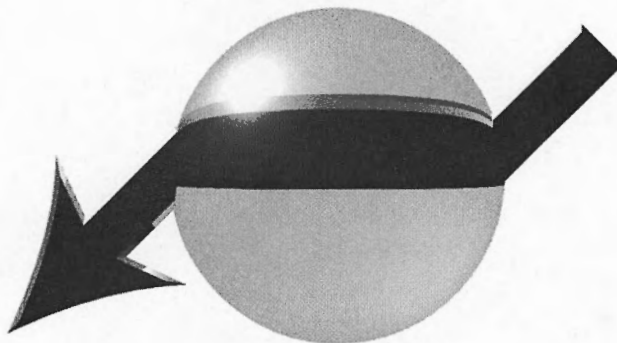
Like reflection, transparency can be specified as a percentage, with air being 100 percent transparent and plate steel being totally opaque—or 0 percent transparent.

**Tip**

In nature, total reflection plus total transparency never exceed 100 percent of the light striking a surface (unless it is glowing). In an extreme example, you can't have perfectly clear glass that also reflects like a perfect mirror. Some 3-D programs let you get away with this, but it isn't realistic. Sometimes, however, the trick can be used to add an interesting effect.

**Refraction**

*Refraction* is the bending of light as it passes through a transparent surface (see figure 7.10). This quality is responsible for the optical properties of a lens or magnifying glass. Refraction also is apparent when you look at a room through a glass full of water. Light is bent once as it passes through one side of the glass, again as it enters and exits the water, and again as it passes through the far wall of the glass.



**Figure 7.10** *Refraction changes the direction of the light*

The amount that light bends as it passes through any given material is defined by a standard number called the “index of refraction.” Air has an index of refraction



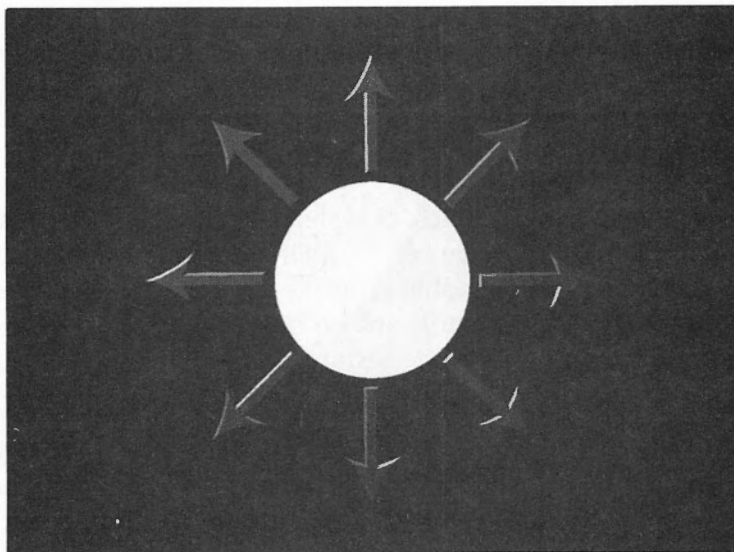
of 1.0—meaning it doesn't refract at all; glass has an index of refraction of 1.2 to 1.9; water has an index of 1.33 (1.30 for ice); and diamond has an index of 2.4. Refraction is often a fairly subtle effect, but it is an effective clue to a material's composition.

Because of its nature, refraction (the bending of light rays) is only calculated in ray-traced renderings. It is not supported in Gouraud or Phong shading.

Refraction as practiced in 3-D is really an over-simplification of the bending of light. Transparent materials to some degree also *diffract* light—that is, different colors of light bend different amounts as they pass through a surface. This is how a crystal prism works, with the result that white light passing through the prism is split into a rainbow. Even sophisticated ray tracers, to date, don't attempt to recreate this effect (but you could fake it with the help of a rainbow-colored light gel on a spotlight aimed "out" of a prism).

## Glow

*Glow* is the degree to which a surface appears to emit light (see figure 7.11). This light is added to the light reflected from the surface, often to the point where the reflected light is overwhelmed. In 3-D applications a glow is not the same as a true light for the simple reason that it does not cast light onto other objects.



**Figure 7.11** *Glow is not a form of reflection at all—objects that are heated to great temperatures, excited by electricity, or radioactive emit light of their own*

It's common practice, however, to use glows to simulate glowing light sources that also emit light (see chapter 8, "Lighting").

In Electric Image Animation System, the only Mac 3-D application with visible light sources, you can set a light to glow, but not cast light; or to cast light, but not glow. It also can be set to do both, in which case it looks like a sphere (or tube) brightly lighted in the middle and fading towards the edges. (You also can set a light to have a "negative" glow, like a black hole). EIAS, like other 3-D programs, however, does not support glowing models casting light onto other surfaces.

## Texture Maps

While some materials, particularly plastic and paint, have a uniform appearance across a surface, it is much more common for objects to have great variations in color: veins of marble, grains of granite, blades of grass, ripples of water, spatters of paint, or swirls of oil on water. Often a single "texture" is made up of many small parts (such as ceramic tiles) or an agglomeration of material (such as granite) that would be impossible to model separately. These textures exceed the descriptive power of simple color and reflectance. Even seemingly uniform surfaces like painted wood have a texture when viewed up close; the brush strokes and the imperfections in the mixing of paint conspire to make a finely detailed surface. Brushed aluminum, dented chrome, and smoke-damaged paint are all surfaces that benefit from details and textures that go beyond simple surface qualities. Consider a brick wall. It is possible to model hundreds of bricks and to stack them up with mortar, one by one, similar to a real brick layer. However, this would be terribly time-consuming. (In fact, it would probably be quicker to go out and lay real bricks yourself!)

Three-D programs offer a combination of techniques, loosely called *texture mapping*, which can accurately describe complex surfaces. In its simplest form, texture mapping stretches a photographic image as if it were a rubber sheet around an object, so that it fits to every curve and corner. This makes the object appear to be composed of the materials in the texture image.

But texture mapping involves more than choosing an image and pointing at the object to receive the texture. Bricks aren't four feet long; nor are they laid on end; and a photo of bricks may contain only eight bricks, while the object being texture mapped may need hundreds. These considerations call for the control of position, scale, direction, and tiling of textures.

In some cases you will want the bricks to wrap around a cylinder, while on a sphere you may want them laid in concentric circles like the latitude lines on a globe; or you may want them simply laid out like a sheet. These different methods are defined by mapping projections.

## **Color Mapping**

The simplest form of texture mapping is to wrap an existing 2-D image around a 3-D model. This image can be a bitmapped illustration created in a paint program, a scanned photo, or even a mathematically generated image created on the fly.

You can use color mapping any time you want a modeled surface to have a “skin” of more than a single color. The following is a small sample of things you can use color maps for:

- Painting details on a wall
- Adding stripes, polka dots, or other regular patterns to a surface
- Placing labels on boxes, cans, and other package designs
- Putting a “skin” on a surface, such as fur, scales, spots, or stripes
- Putting grass in a front yard or moss on a rolling stone

## **Seamless Textures**

A seamless texture is a texture image whose top edge precisely matches its bottom edge, and whose left edge matches the right. This enables you to place many tiles edge-to-edge without a visible seam in between them—usually your 3-D program will do this automatically. Typical seamless textures are bricks with mortar and other masonry, grass, carpeting, or hammered metal. Any surface that has a regularly repeated pattern (or such a complex pattern that you can’t tell that it’s repeated) is a good candidate for a tiled texture. Another advantage of this approach is that you can use a relatively small image file to cover all of a large model without a loss of texture quality.

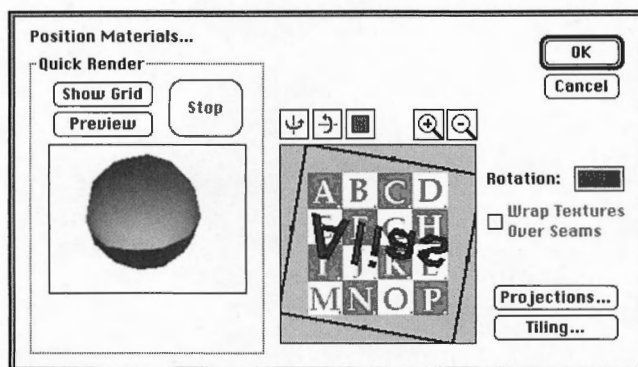
Using Adobe Photoshop and other image editing programs with a “cloning brush,” it’s fairly easy to create your own detailed seamless textures. Kai’s Power Tools, from HSC software, provides an unlimited source of abstract seamless textures automatically, as does Kaos Tool’s Terrazzo and Specular’s TextureScape.



There are a number of seamless texture collections available on CD-ROM, such as Wraptures, from Form and Function. You'll also find some seamless textures on the CD-ROM included with this book.

## Position

Most scene builders enable you to apply a texture beginning in one location on a surface and ending at another (see figure 7.12). This enables you to precisely control how much of an object is covered by a texture. Imagine, for example, that you have a texture resembling a tapestry hanging on a wall. It wouldn't be appropriate to have it wrapped around the corner between two walls, or draped over the top of the wall like a towel on a rack.



**Figure 7.12** Sketch!'s Texture mapping dialog box enables you to set direction, projection, scale, and tiling of textures

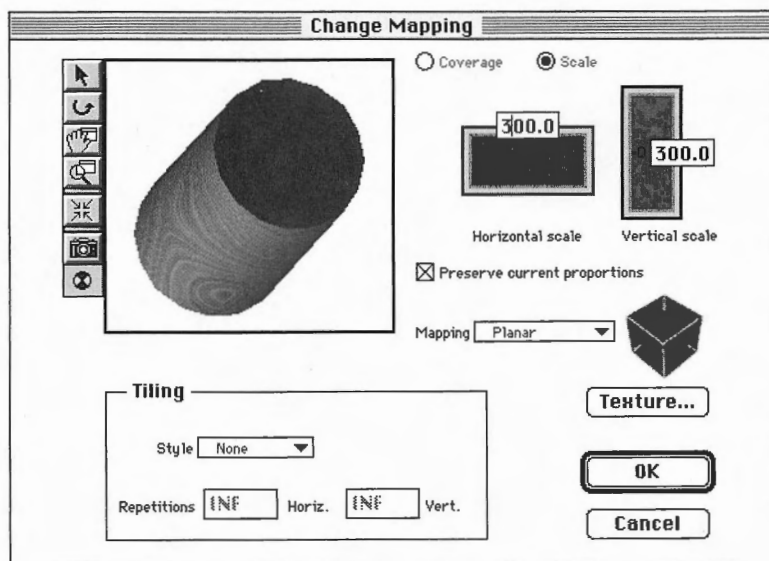
Another common example of the need for positioning textures is in industrial product designs. You might need to place a label in a precise position on a shampoo bottle, or a logo in the right spot on a stereo's front panel. Even very small details can be precisely positioned on a model.

For example, you might put rust on a pipe joint, or rivets on the seams of a ship. These details go a long way towards adding realism.

## Scale

The scale of textures in 3-D is as important as the size of fonts used in book publishing. Kitchen tiles are normally about six inches square. You'll get a strange looking counter top if you surface it with tiles that appear to be two feet square.

Likewise, gravel the size of boulders or bricks the size of postage stamps will cause confusion. Most 3-D programs that support texture mapping enable you to set the scale of a texture so that it matches the scale of your model. In some cases, scale is handled by adjusting the tiling. For example, if you specify that there should be 40 copies of the brick mapped to fill the wall, the brick texture will automatically be scaled to fit. In other instances, such as with StrataVision 3d and StrataStudio Pro, you can scale the texture specifically to fit a certain part of your model, irrespective of how many times it will be tiled (see figure 7.13).



**Figure 7.13** StrataStudio Pro has an elegant graphical interface for positioning, scaling, tiling, and orienting textures

## Tiling

It's common to use a single "seamless" texture of limited size, and to place it, over and over, edge-to-edge. *Tiling*, as this technique is called, is akin to laying many tiles on a floor or counter. This is useful if you have a texture that isn't big enough to cover a surface at a certain scale; you can tile it several times so that it covers the object completely with multiple repetitions. A tiled texture can be as simple as a single polka dot, or it can be an elaborately disguised seamless texture repeated over and over on the surface. Tiling is an extremely fast and effective method of texture mapping because the image being mapped is small and it only has to be loaded into memory once.

## Direction

A common use of textures is to map a label onto a package. It's possible to map this same label in many orientations: standing on either end, upside down, or backwards. Obviously, it's important to align it in the right direction. A more subtle application of direction is wood grain. The grain of wood always runs lengthwise. With wood and other *procedural textures*, this is particularly crucial because these textures have a grain that resembles that of real wood—that is, they are different on the sides than on the ends.

Fortunately, most 3-D programs have some sort of alignment preview which enables you to pinpoint and change a texture's alignment before you create a final rendering. Generally, a texture is applied using the position of the object in the current window as a reference, mapping the texture in normal upright orientation, but this varies from program to program.

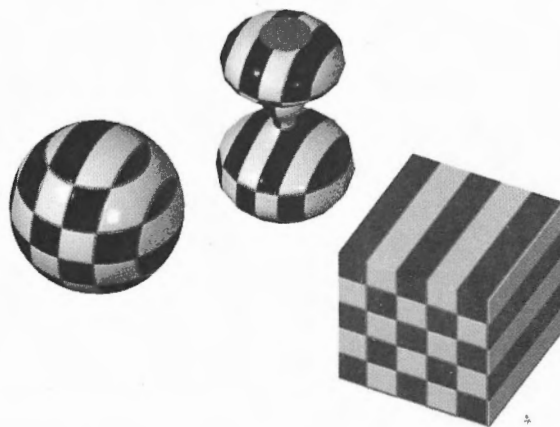
## Projections

There are many different ways to apply a texture to a surface. Essentially, techniques have one thing in common: the texture is either applied as a flexible sheet or projected as if from a slide projector. This is analogous to what map makers do when they build a globe of the earth. The rubber sheet metaphor takes several forms; the more complex projections, (like cylindrical, cubic, and spherical) treat the map like shrink wrap that is uniformly shrunk from every direction until it sticks to the model. Usually, these different techniques are used for texture mapping objects with a corresponding shape. For example, you would use spherical mapping if you wanted to apply a texture to a grapefruit, while you would use cubic mapping to apply a surface to all sides of a box. A combination of careful design when building a texture and the use of the right mapping technique makes it possible to precisely apply surfaces to objects. In the example of a cereal box, more powerful 3-D programs make it possible to create a texture with the appropriate cut-outs so that a single texture will cover all the surfaces of the box.

While projections are usually used to map surfaces onto corresponding shapes (for example, cylindrical mapping is used mostly for cylinders), it's possible to use the "wrong" mapping techniques to achieve interesting effects. Planar mapping a label onto a sphere, for example, will stretch the label onto the round surface of one side of the sphere as if the texture were projected with a slide projector.

## Planar Mapping

The simplest form of texture mapping is *planar*. This uses a flat sheet through which the model is pushed (see figure 7.14). Imagine doing this with a cube: the front face will receive an undistorted projection of the texture, as if you were projecting a slide on a screen. In general, planar mapping is reserved for putting flat objects onto more or less flat surfaces. It is often used for labels, as well as for basic mappings such as putting a painting in a frame.



**Figure 7.14** Planar mapping enables you to apply texture onto flat surfaces without distortion, as on the front face of the cube, but it also can be used to create interesting effects on surfaces that aren't flat

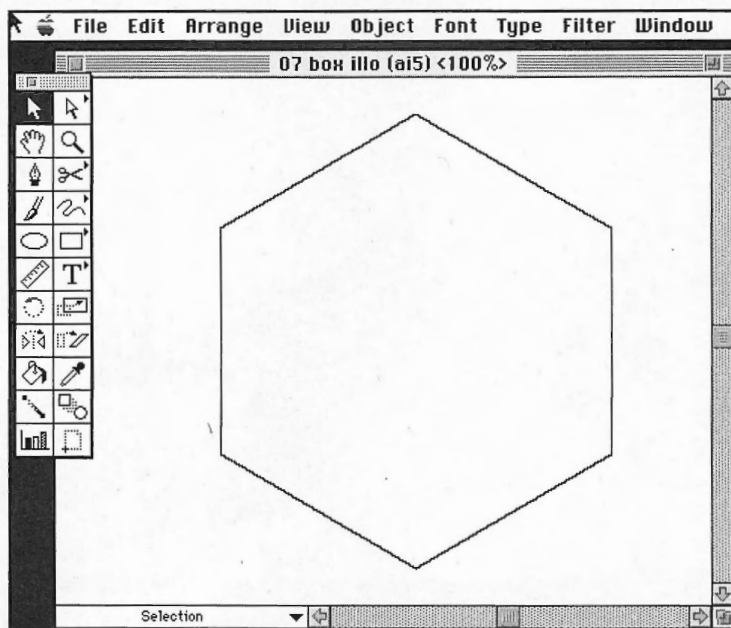
### Tip

Planar mapping enables you to apply fine details to a model, such as a wall. Begin by generating a texture that is the same shape as the surface you're covering. If you map it to the wall using planar mapping, you should be able to match details precisely to the shape of the object. This works even when there are holes or other irregularities in the surface.

## Making a Box Label

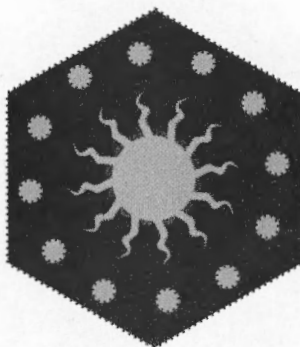
This example uses planar mapping to apply a label to a candy box. The hexagonal candy box in this example was created in StrataStudio Pro and the texture map was generated using a combination of Adobe Illustrator and Adobe Photoshop.

1. Draw a hexagon in Adobe Illustrator (version 5.5 has a filter that will do this automatically—see figure 7.15). Save this plain shape as an Illustrator outline for later use, and then decorate the shape with a border and artwork. Save this as a separate illustration.



**Figure 7.15** *Creating a hexagonal outline in Illustrator*

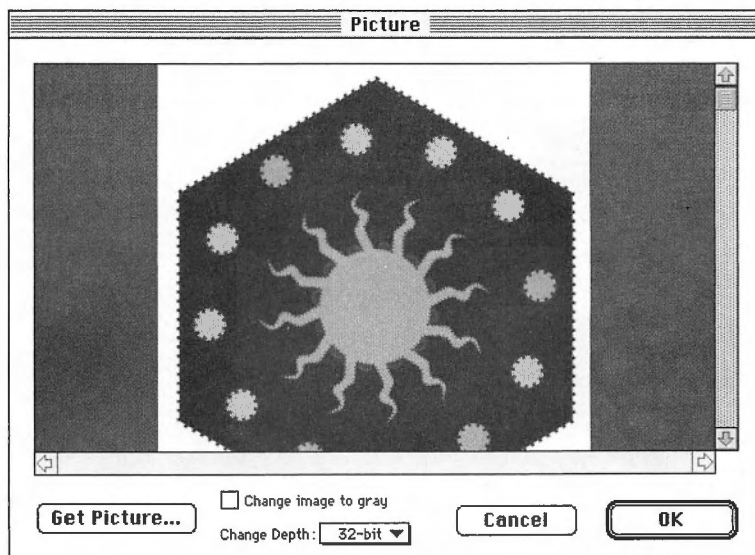
2. In Adobe Photoshop, create a 400 by 400 pixel RGB image filled with white and use the Place command in the File menu to import the decorated illustrator artwork. Crop it close to the edges and Save it as a PICT file (see figure 7.16).



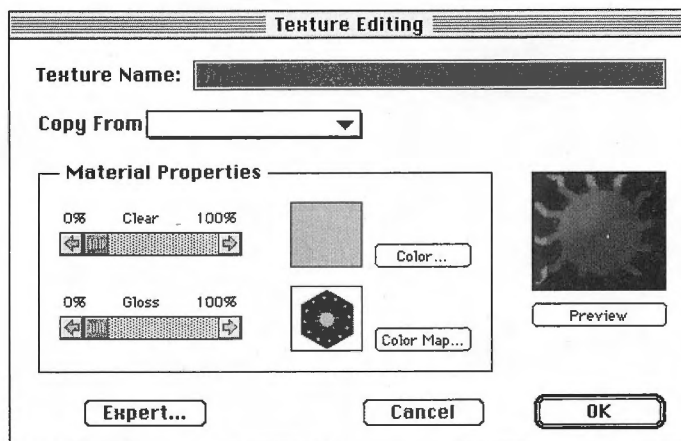
**Figure 7.16** *The finished PICT that will be used as a label*



3. In StrataStudio Pro, import the plain hexagon illustration file, convert it to a filled shape, and extrude it with a bevel. Add a table, if you'd like something to catch shadows in the image, but make sure to switch views to rotate your position instead of rotating the model.
4. Switch back to the front view, open the Textures editor, and select New Texture (see figure 7.17). Click the Color Map button; this will open the Get Picture dialog box (see figure 7.18). Now you can import the PICT file that you created in Photoshop.

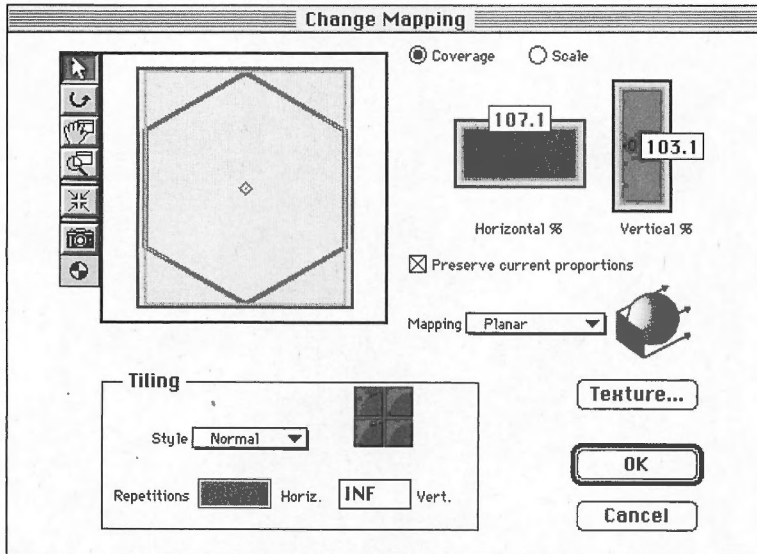


**Figure 7.17** Creating a new texture and importing the decorated PICT



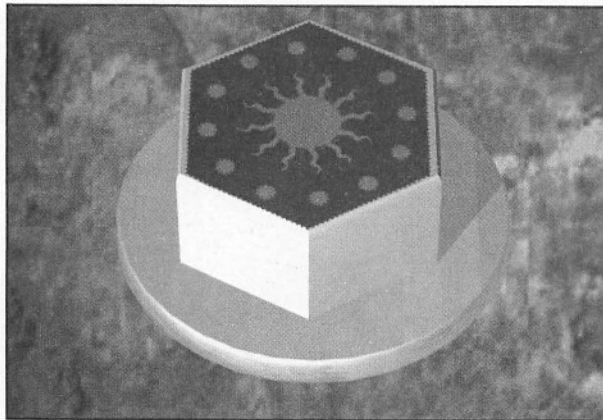
**Figure 7.18** The Texture Editing window should look like this when you're finished

5. Open the Change Mapping palette (see figure 7.19) to make sure the texture is mapped with planar projection and covers the box surface. This also is your chance to change the orientation of the texture, if necessary. You can click the camera button to get a rendered preview of the texture alignment.



**Figure 7.19** The Change Mapping palette enables you to align and scale the texture on the box's surface

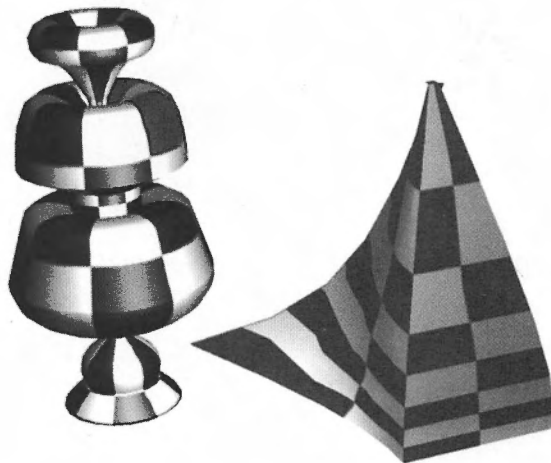
6. Position the model, set lighting, and render the image (see figure 7.20).



**Figure 7.20** The final image

## Parametric Mapping

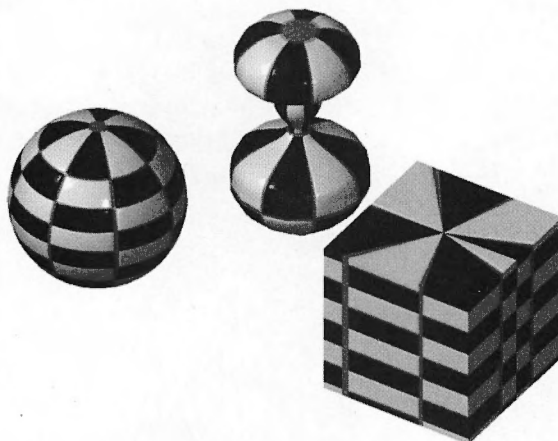
*Parametric* (sometimes called *proportional*) mapping attempts to adjust the texture to the shape of the object in different locations (see figure 7.21). This mapping method is best used when you're texturing a surface that has lots of variations in size and you want the details of the texture to vary with it.



**Figure 7.21** *Parametric or proportional mapping varies a texture with the size of a model in different areas*

## Cylindrical Mapping

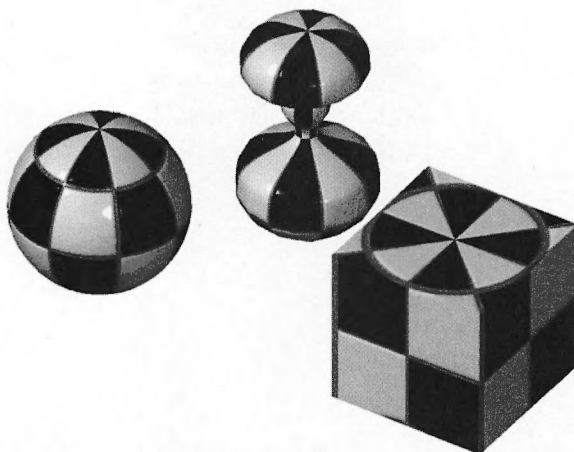
*Cylindrical* mapping wraps the texture around the object like a label around a cola can (see figure 7.22). Some programs enable you to specify cylindrical mapping with end caps, meaning that the texture is mapped around the surface in a circular fashion, and round cut-outs of the texture are placed on the ends.



**Figure 7.22** Cylindrical mapping wraps a texture around an object the way paper towels are wrapped on a roll

## Spherical Mapping

*Spherical* mapping is like that which is used to put the surface of the earth on a globe (see figure 7.23). In this method, textures shrink at the ends (or poles) and stretch at the center (or equator). This method is useful for mapping textures onto spherical or otherwise rounded objects, such as fruit or extraterrestrial planets.



**Figure 7.23** Spherical mapping applies a texture evenly around a sphere

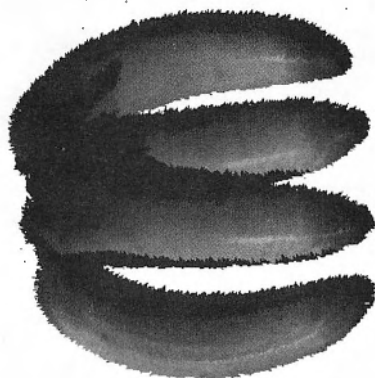
## 3-D Painting

Two 3-D programs have new approaches to texture mapping: StrataStudio Pro enables you to “paint” on a model as it’s being rendered; RayDream’s Designer, meanwhile, enables you to draw and paint on a model’s surface in real-time.

### RayPainting

Strata’s new StrataStudio Pro offers a new kind of rendering, called “RayPainting.” Instead of realistically rendering objects in a model, RayPainting renders objects with simulated strokes of paint. This is much faster than ray tracing or Phong shading because the technique uses fairly wide impressionistic strokes to render an image (see figure 7.24). One of the unique effects of this technique is that the edges of objects are displaced by the brush strokes, creating the effect of procedural fur and other rough textures.

RayPainting uses ray tracing as its starting point, so that images you create with the technique can have mirror reflections and transparency as part of the effect.



**Figure 7.24** These “fur” bananas were rendered with StrataStudio Pro’s RayPainting system. Notice that the edges of the objects are displaced, unlike the results of normal ray tracing

## Designer's Painting

After you have defined a shape in RayDream Designer, you can easily paint or draw directly on its surface. If you draw a rectangle on an irregular surface, for example, you can export the rectangle to a program like Photoshop for the addition of fine details. Designer automatically re-imports this modified image into its original position on the model.

The beauty of this approach is that once you've painted on a surface the "paint" is a separate layer to which you can apply new textures or filters, for example.



**Figure 7.25** *Ray Dream's surface painting is currently the easiest way to get a texture on a surface*

You can even use Photoshop-compatible filters directly from within Designer, so things like Kai's Power Tools textures are right at your finger tips.

## The G-buffer

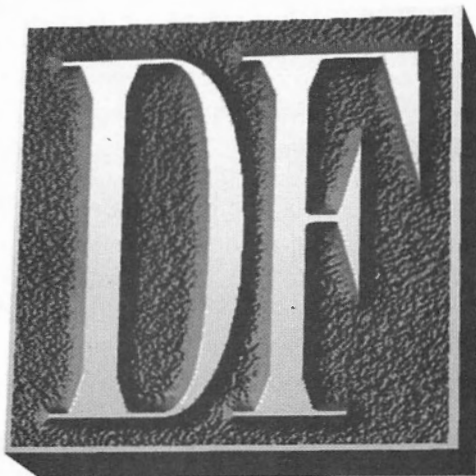
RayDream introduced a new file format into Designer 3.0 called the "G-buffer." Just as an alpha channel can be used to carry transparency information along with an image, the G-buffer uses 8-bit grayscale images to convey depth and other information. You can take your G-buffer images into any program that fully supports Adobe Photoshop's file format (with multiple channels in addition to RGB). One of the coolest uses of this capability is to use the depth channel to select your rendered scene, and then apply filters, such as Aldus Gallery Effects, to the image. The result can be like textured fog or depth of field. Another interesting tool is the Object channel; this saves a mask for each object that is different from the other objects in the scene, so that you can select it with Photoshop's Magic Wand tool. After Effects makes use of this depth information by enabling a user to paint in between objects, or to have painting affect a 3-D image differently as you paint over different "depths" of the image. You can easily add realistic fog using an airbrush tool, for example, or use a blur filter to create depth-of-field effects.

The G-buffer may even be able to carry lighting information through to a paint package, so that you can paint on an object's surface while maintaining realistic shading. Time will tell whether the creators of paint and animation programs will embrace this new format.

## Bump Maps

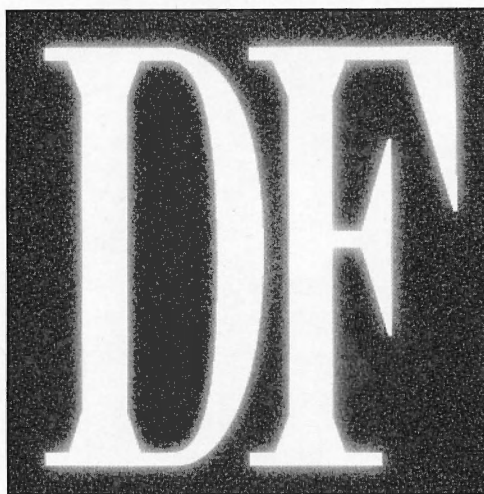
Bump maps are a special type of texture map, which are often used in unison with regular texture maps. A *bump map* creates the effect of bumps on an object's surface. Bump maps begin as simple grayscale images, such as a series of soft-edged black polka dots on a white background to simulate a golf ball texture. When you use this texture as a bump map, the renderer interprets the dark dots as depressions (or raised bumps, depending on your program) in the surface of the object. Light will reflect and cast shadows as if there were actually dents in the surface. Sometimes, you can choose to have either the light or dark areas rendered as depressions. But even if the application forces you to use one or the other, most image editing applications will enable you to create a negative (inverted) image to step around this problem.

Imagine mapping an image of kitchen tiles onto a 3-D countertop. If this is accompanied by a bump map of a grid of lines where the grout between the tiles should be, the surface will not only have the qualities of the tiled surface, but it will appear to have rough, grooved depressions between each tile.



**Figure 7.26** A grayscale bump map created in Adobe Photoshop





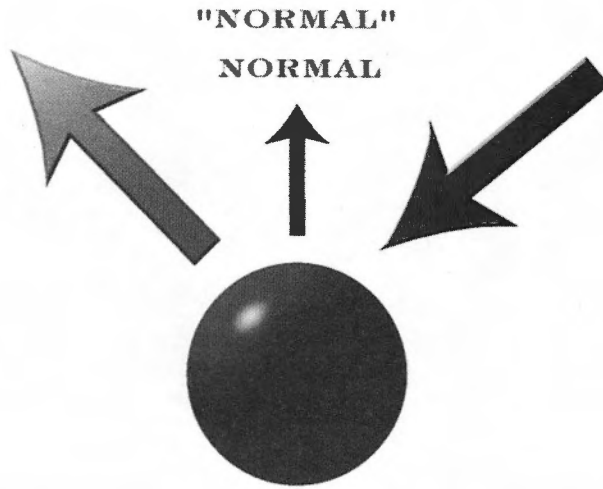
**Figure 7.27** *The result of using the bump map on the top surface of a plain cube*

In many instances, you won't need a color map at all if you use a bump map. Stucco on a wall, for example, is a continuous color of a rather flat material. By applying a bump map that simulates the rough texture of the surface, you can achieve a very realistic stucco imitation.

As described early in this chapter, renderers rely on the direction of the “normal” vectors that define a surface to determine the quality and direction of reflected light. Bump maps trick the renderer by changing the direction of the surface normals based on the gray values of the bump map (see figure 7.28). While the actual surface normal of a point on a countertop aims straight up, a bump map sets the normal to tilt more or less to one side depending on if it should be in the bump's shadow or highlight. When the renderer goes to shade this point, it does so based on the tilted normal (see figure 7.29).

For some subtle effects, the result of a bump map is every bit as realistic as rendering a modeled bump (since the renderer, in effect, doesn't know the difference). The primary limitation of bump maps compared to modeling details is that bump maps lose the illusion of modeled textures at the edges surfaces. Where you would expect to see a serrated surface at the edges of a golf ball, a bump-mapped surface instead appears perfectly smooth (see figure 7.30).

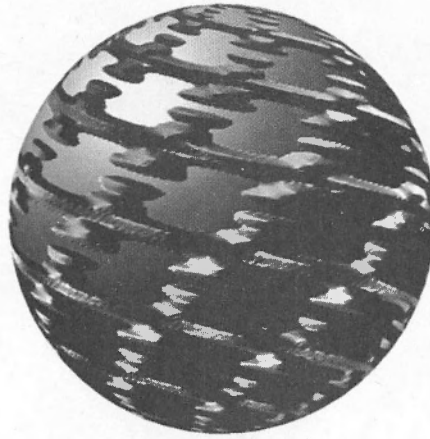




**Figure 7.28** A renderer calculated shading based on the surface normals, which are “normally” perpendicular to the object’s surface



**Figure 7.29** A bump map tilts the normals to one side or the other, depending on whether they’re on one side or the other of the “bump.” This tricks the renderer into shading a bump on the surface where none really exists



**Figure 7.30** *Even though this bump-mapped sphere looks bumpy on the surface, its smooth edges belie the fact that bump mapping is an optical trick*



The CD-ROM included with this book contains a sampling of ready-made bump maps, which match the ready-made textures that you can try in your 3-D application. These include pitted metal, brushed aluminum, golf ball dimples, riveted plates, and non-skid floor.

In fact, all the texture-mapped effects that you can apply to a surface, such as transparency, diffuse reflectivity, glow, and so-on, are created in the computer by tweaking surface normals. The effects maps you apply are used to scale these vectors in various ways. Fortunately, we, mortal 3-D users, are completely insulated from the details and mathematics of these procedures, but it's sometimes helpful to understand (and crucial if you ever want to do RenderMan programming).

## **Bump Mapping Type on a Plain Surface**

One of the simplest and most effective uses of bump mapping is the rendering of embossed or engraved type on a surface. This example uses Photoshop and StrataStudio Pro, but most 3-D programs that offer texture mapping also allow for bump mapping. Virtually any Mac paint program will enable you to create a grayscale bump map. The procedure for using a bump map is generally the same as for any other texture map.

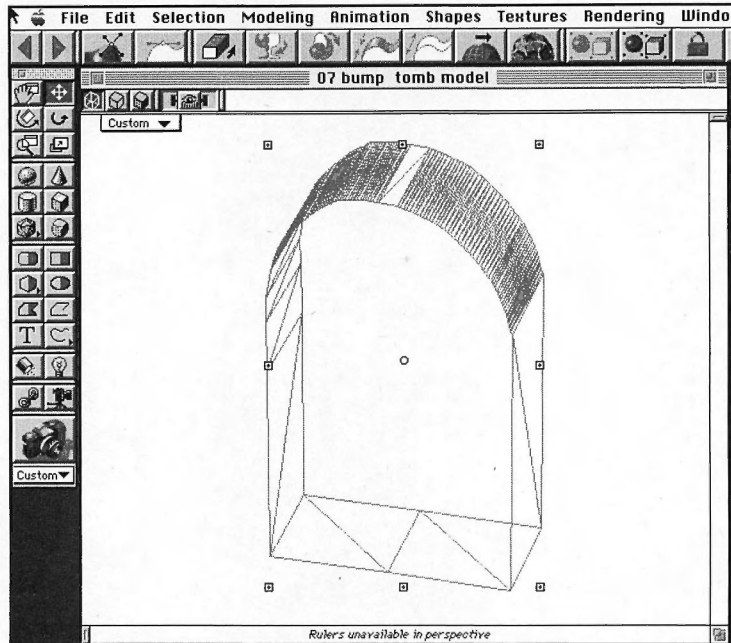
1. Begin by creating a grayscale image of black-on-white text in Photoshop (see figure 7.31). The size of the image should be the approximate size, in pixels,

that it will be when it's finally rendered. Use the Blur More filter to blur the image. This will give the bump mapped object a soft beveled edge, instead of unnaturally hard corners (you can really see this technique in action in figure 7.27). Save the image as a PICT file.

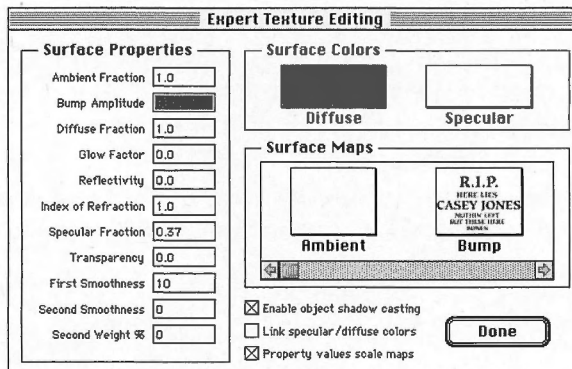
**R.I.P.  
HERE LIES  
CASEY JONES  
NUTHIN' LEFT  
BUT THESE HERE  
BONES**

**Figure 7.31** *Creating the epitaph in Photoshop and saving it as a PICT*

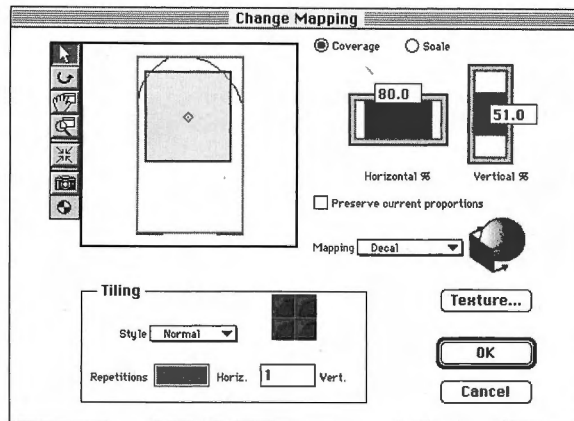
2. Open StrataStudio Pro and build a simple model by drawing an outline in the front view, clicking the Extrude button, setting an extrusion depth, and clicking OK (see figure 7.32).
3. Create a new texture in the Textures window. Apply a medium gray tombstone color. Click the Expert button (see figure 7.33) and enter 20 in the Bump Amplitude box (this scales the degree to which the bump map affects the image; the larger the number, the deeper the bumps). Click on the Bump picture box in the scrolling effects window. In the Load Image dialog, select the PICT file containing the text you created in Photoshop. Click Done to exit the Textures dialog box.
4. With the model still selected, open the Change Mapping dialog box and set the projection to Decal. Decal differs from Planar mapping in that the texture will be applied only to the front side of the object. Planar mapping, meanwhile, will push the texture all the way through to the back side as well. Adjust the scale so that the ghosted figure of the texture fills an appropriate area on the model (see figure 7.34).



**Figure 7.32** Build the tombstone by drawing the outline in the Front view and extruding the shape

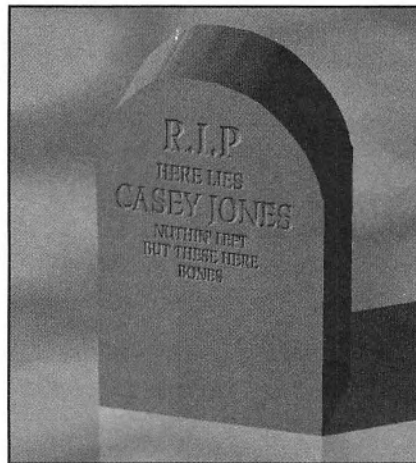


**Figure 7.33** Loading the epitaph PICT into the Bump channel and setting the Bump Amplitude to 20



**Figure 7.34** Positioning the texture on the surface of the model using the Decal map setting

5. Exit the Change Mapping dialog box. Set your lights and viewpoint, and render the image (see figure 7.35).



**Figure 7.35** Rendering the final image

## Surface Maps

Just as you can apply an image to a surface to create a texture, more powerful 3-D applications enable you to apply a texture that modifies one or more of a surface's qualities. Like bump maps, these textures are usually in the form of grayscale images. The brightness values in the image are used to increase or decrease a given quality in

the object. For example, for a reflection map, you could map a piece of white-on-black type onto a sphere's surface, and everywhere the type is mapped to the surface it will be highly reflective; where the map is black the surface won't change at all. You can apply this same map to the glow channel of the object and the surface will glow brightly everywhere that the type is mapped onto the surface.

Like bump maps, effects maps affect the degree to which a surface property is emphasized.

You could use this technique to create a globe where only the continents are opaque and the rest is transparent.

Common types of effects maps include: transparency, glow, reflectivity, specular reflectance (sometimes includes a color), diffuse reflectance, and ambient reflectance. Not all programs support all of these, although most of the better ones do.

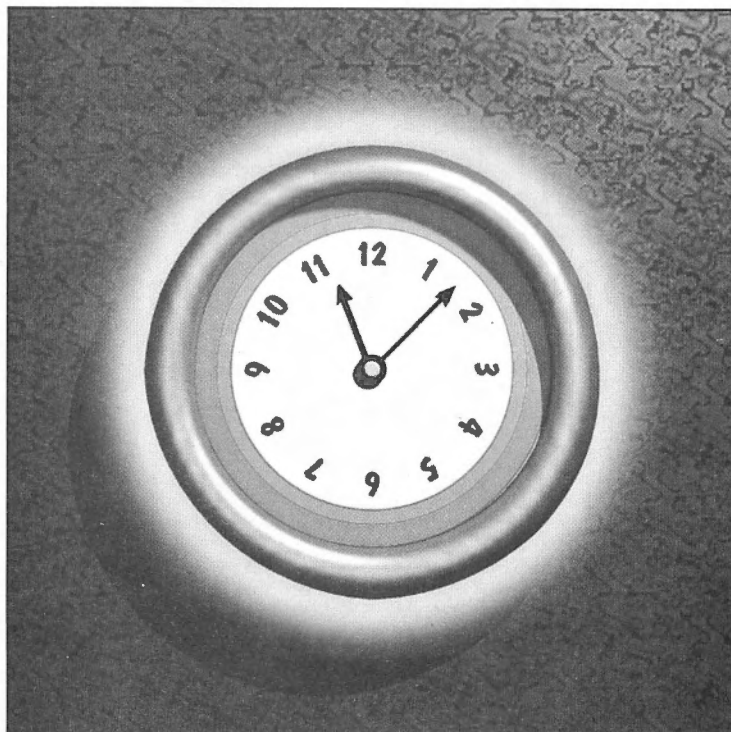
**There are a number of programs that support surface-quality mapping, including:**

- Atlantis
- Designer
- Electric Image Animation System
- Infini-D
- Presenter Professional
- Sculpt
- Shade III
- Sketch!
- StrataStudio Pro
- StrataVision 3d

## **Glow Maps**

Glow maps are incredibly useful wherever you want to create the effect of glowing objects. This can be anything from a lightbulb to the pulsing engines of a deep space probe. One of the problems with 3-D programs (RenderMan also enables you to apply shaders to a light to make it glow realistically, and ELIAS also enables you to have

glowing light sources) is that light sources are not directly visible—only the light they cast is. If you look at a lamp or a computer monitor, you'll realize that this is a major omission. Glow maps are the answer to the problem. For example, the neon clock (see figure 7.36) uses a ring-shaped glow mapped onto the wall.



**Figure 7.36** *Rendered with a glow map on the wall*

## **Glowing Sign**

To create the effect of light emitted from a neon sign in StrataVision 3d, you can map a soft-edged glow map onto a wall or other object near the neon light. It will appear as though the glow is cast by the glowing object.

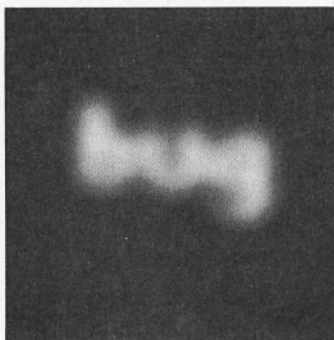
1. In Photoshop, build a black-on-white logo (its shape should roughly match the size of the wall you'll map it to). Save this for later use. Invert the image so that you have a white-on-black type logo and save it with a different file name (see figure 7.37).





**Figure 7.37** *Creating a starting logo in Photoshop*

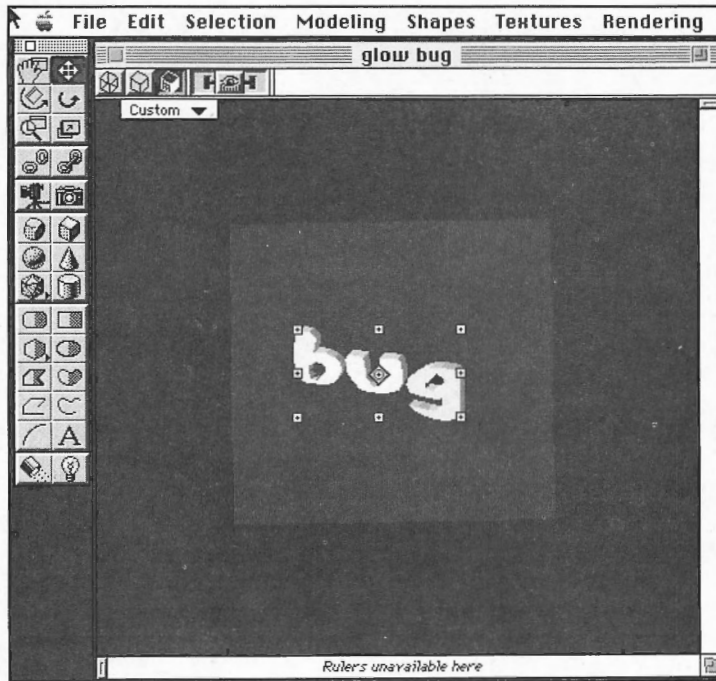
2. Blur the texture map extensively so that the white shape has a soft edge all around, and save it at the same resolution (see figure 7.38). This will be used as the glow map on your backdrop wall.



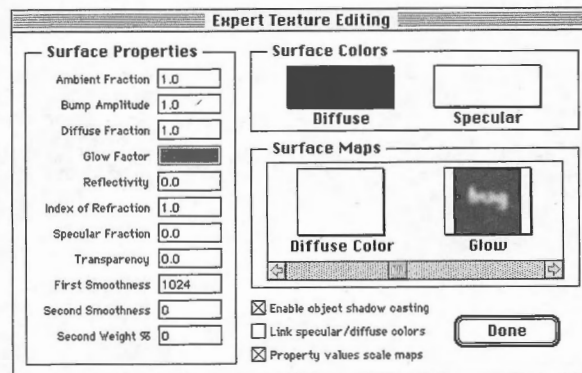
**Figure 7.38** *Blurring the logo in Photoshop*

3. In StrataVision 3d, build a simple model of a wall (in a front view) and give it a dark color. In the same view, import the black-on-white PICT. When you import it, it will automatically be converted to an extruded model. Switch to a side or top view to position this model in front of the wall (see figure 7.39).
4. Select the wall model and create a new texture. Give the texture a dark color (the darker your texture color, the more the glow map will appear to glow) and import the blurred PICT file into the Glow picture window (see figure 7.40). Set the glow factor to a value of 5.



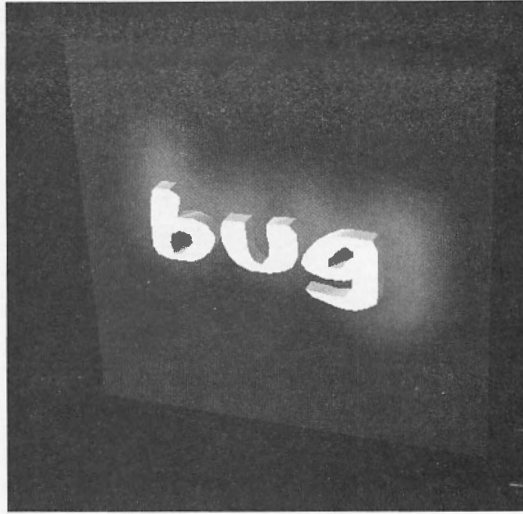


**Figure 7.39** *Importing the glow map and extruding the type*



**Figure 7.40** *Using a glow factor of 5 for the glow map*

5. Render the scene (see figure 7.41).



**Figure 7.41** *The rendered scene*

## Labels

A *label* is nothing more than a texture map that covers part of another texture. This is commonly used for packaging designs, where you want a surface applied over another surface.

### **Programs that support texture layers**

The ability to layer multiple textures, one atop the other, is extremely useful and powerful. For example, you can create a label out of gold foil type that can be overlaid on a granite surface. In programs that don't support multiple textures on a surface, you'll have to create a separate model to accommodate the gold type; this may not work if the underlying model is very complex. To date, only a handful of 3-D programs enable you to layer textures. They include:

- Electric Image Animation System
- Infini-D
- Sculpt 3-D
- RayDream Designer

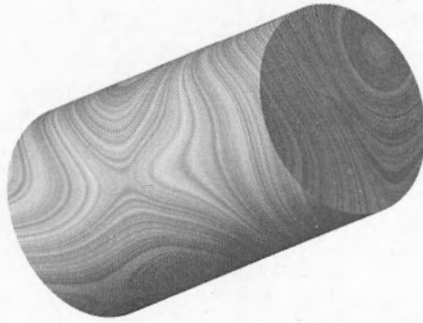
For programs that support surface quality maps, but not multiple texture layers, you can create a single texture out of composited images, and then create surface quality maps to modify parts of the texture. For example, you might use a reflection map to make part of the texture mirror-like. The most common occurrence of this is to build a model that mimics the model onto which you want to apply a label; then make this entire model invisible with a transparency map, except for the part where the label goes. Sketch, for example, makes it easy to do this because you can create surfaces on top of existing surfaces. While StrataStudio Pro has no such facility, it makes up for it by making it extremely easy to create a texture that is invisible except in one specific area.

Transparency also is an issue in this regard because some programs only support transparency when ray tracing, which may make this technique impractical for animation.

To use an illustration created in a program like Adobe Illustrator as a label, you'll first have to save it in a format that can be used by the 3-D application. (See "Making a Box Label," in this chapter.) In general this means PICT, TIFF, or JPEG, although as described in the section on 3-D PostScript, addDepth and Dimensions enable you to use an Illustrator-type EPS file as a label. To use a PICT in Electric Image Animation System, you have to first use the file translator, Transporter, or Equilibrium's DeBabelizer, to convert the PICT into an "Image" format file, meaning it's a four-step process to get an image from Illustrator onto a model in EIAS.

## Procedural Shading

*Procedural shading* is more sophisticated than simple images wrapped around surfaces. Wood, for example, has a long, wavy, relatively straight grain along the long cut of a board. On the ends, the grain is a section of concentric circles—the growth rings of the tree (see figure 7.42). Marble has a similar effect, in that the grain along the side of a piece of marble is continued from the front face to the sides of a piece of the stone. Because procedural shaders act on volumes, rather than surfaces, they are sometimes called *volume shaders* or *solid shaders*.



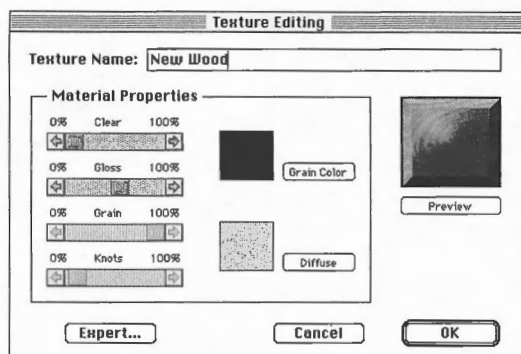
**Figure 7.42** The grain of procedural wood shaders adapts to the geometry of an object—this wood, rendered in *StrataStudio Pro*, has continuous grain around the object's corners

#### **Procedural shaders**

The following renderers have procedural shading options:

- All RenderMan renderers
- Infini-D
- Presenter Professional
- RayDream Designer
- Sculpt 3-D
- Shade III
- StrataStudio Pro
- StrataVision 3d

Fortunately, 3-D programs do not require you to understand the underlying procedures; those that support procedural shading come with a simple editing interface that enables you to modify and orient the texture. Most wood shaders enable you to modify the base and grain colors, as well as the tightness and waviness of the grain (see figure 7.43).



**Figure 7.43** *StrataStudio Pro's configuration controls for a procedural wood shader*

Pixar's RenderMan specializes in the use of procedural textures more diverse than wood and marble. RenderMan shaders exist, for example, for dragon skin, flaking paint, and gravel. Unlike bump maps, these procedural shaders create real displacement (see "RenderMan," in this chapter and chapter 12, "RenderMan").

## Adding Textures

Most 3-D applications that support sophisticated texture mapping also ship with a library of textures. Typically, these will include bricks, marble, glass, granite, chrome, and a few other stones and metals. It's inevitable, however, that you'll want to add new textures to the mix. For instance, if you're creating an outdoor scene, you may want to add crab grass, tree bark, or frog skin. None of these is likely to come with the standard package.

### Tip

There are several options for adding textures to your library shelf: you can purchase texture libraries from 3-D and clip-art vendors, modify existing textures to get new effects, or create your own textures by scanning, painting, or synthetically generating them.

It's important to consider the size and resolution of the textures you plan to use. If you are creating a 640 by 480 pixel image, the largest texture map you're likely to need is one that's as large, in pixels, as the background (640 by 480). However, many models take up only a fraction of the total image; there's little reason to use a large, high-resolution image for tiling onto your model at very small sizes. This will take up lots of extra computer resources and result in long rendering times.

Just as the size of your texture map should be determined by the size of the model in the finished rendering, the resolution should be determined by the resolution of the rendering. For 3-D objects, this often becomes a game of “guestimation.” A small extra margin of size and resolution ensures that the texture will fill the intended space without being stretched and “pixelizing.” However, different 3-D programs treat the size and resolution settings of a bitmap graphic differently, so you’ll have to consult the software documentation and vendor for specific recommendations.

Creating an environment map for reflections also requires a bit of experimenting. Often, you can imitate the size and resolutions of textures and environments provided with your 3-D application, though it really depends on your final output.

## Texture Libraries

Most 3-D vendors sell add-on texture libraries that supplement the basics. Often these will combine surfaces with bump maps, so you get a complete texture rather than just a photograph to wrap around the subject. Seamless textures sold for use as background fills are particularly well-suited to the task and can save you a lot of preparatory work. Simple photographic background collections make excellent backdrops for 3-D scenes, but they also can be called into service as the starting point for building textures. By editing supplied textures, you can create your own variations.



The CD-ROM that’s included with this book contains a wide variety of sample textures to supplement the ones that come with your software. These include a selection of matching backgrounds, textures, bump maps, and environment maps that will enable 3-D users to create a huge variety of 3-D images before they ever have to turn to a paint program.

## Creating Textures

There are several ways you can create your own 3-D textures. You can begin with a scanned-in photograph of a material; you can paint or draw a texture in a program such as Fractal’s Painter or Adobe Photoshop; or you can use a texture generator such as Kai’s Power Tools (a Photoshop plug-in).

## Textures with Bump Maps

If you plan to create a bump map to go with an image texture, you’ll often want to make them at the same time, as the information they contain is usually closely

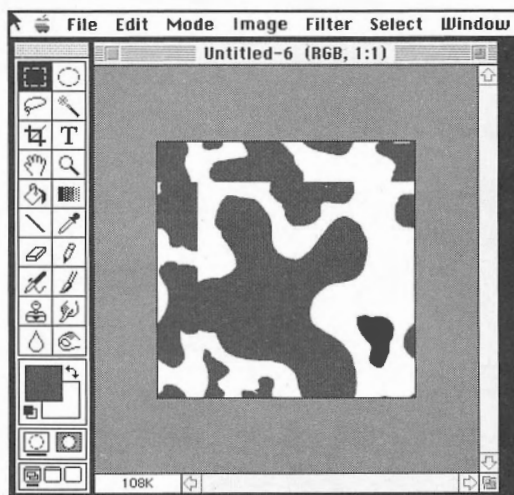
related. For example, if you are scanning in a photograph of a leaf pile to use as a texture, you may want to save a high contrast, grayscale version of the same image to use as a bump map. This will allow you to map the dark spaces between leaves as depressions in the texture's surface.

Texture generators, especially Specular's TextureScape, make it extremely easy to create matching bump and other effects maps for a texture. Photoshop plug-ins, such as Kai's Power Tools can also easily be used to create sets of textures, since it's easy to turn color textures into grayscale and play with their contrast settings and so forth.

### A Seamless Texture for Sketch!

To create your own seamless textures in Photoshop, the goal is to make an image whose left edge is the same as its right and whose top edge is the same as its bottom. This is easy if you're putting a single shape on a solid background and the shape doesn't touch the edge of the image, otherwise, use the following technique:

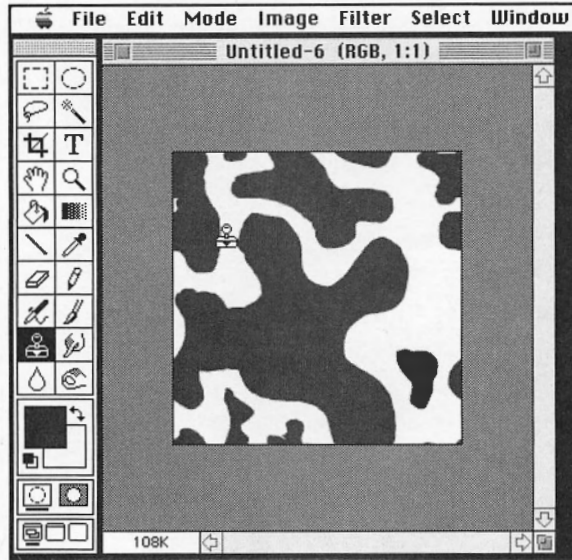
1. In Photoshop, create a square texture of 200 by 200 pixels. This can be a painted image, a scanned-in image, or a texture from a CD-ROM. Displace the entire image 20 pixels down and 20 pixels over (using Filter, Other, Offset). Be sure to use the Wrap Around option (see figure 7.44).



**Figure 7.44** Using the Offset filter with the Wrap Around option to displace the texture.

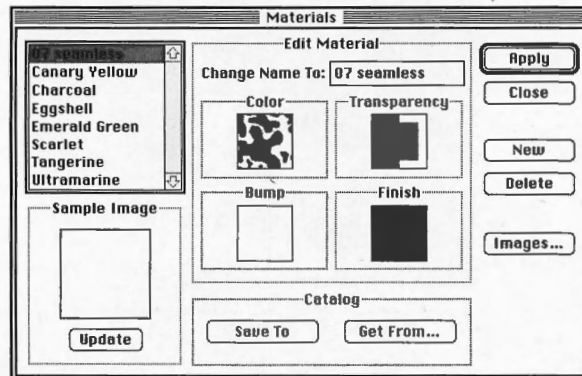


2. Use Photoshop's Rubber Stamp tool to clone away the seams created by displacement (see figure 7.45). Be sure not to touch the edges of the texture or you'll get a visible seam between tiles. Save this image as a PICT.



**Figure 7.45** Use the Rubber Stamp tool to remove the seams—but don't alter the edges!

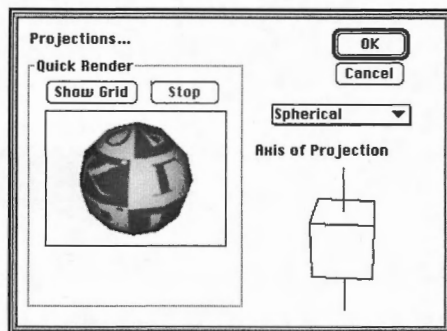
3. Create a spherical model in Sketch!. Choose the Materials command, and select New Texture. Click the color box and load the new PICT you've created (see figure 7.46).



**Figure 7.46** Loading the PICT as a new texture



4. In the Position Material dialog box, choose Mapping and select Spherical from the pull-down menu (see figure 7.47).



**Figure 7.47** *Selecting a spherical mapping for the texture*

5. Render the image (see figure 7.48).



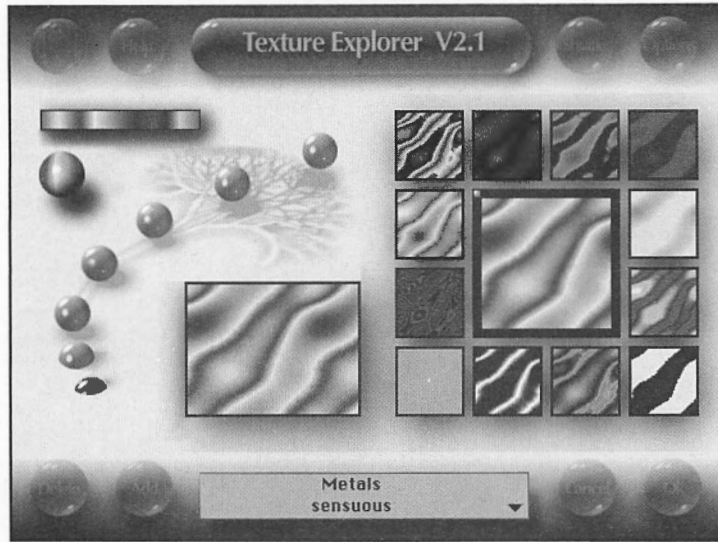
**Figure 7.48** *The final rendered image—no seams!*

## Kai's Power Tools

Kai's Power Tools (KPT) is an accessory product that runs within Photoshop (or other programs that support Photoshop plug-ins, such as Equilibrium's DeBabelizer, and even Ray Dream Designer). It's one of the most useful tools available to the 3-D user. The collection of tools enables you to easily create a huge variety of seamless textures, environments, and backgrounds. It also has the controls needed to create bump maps that neatly complement textures. Most useful among the tools is the Texture Explorer. This is a program with a unique "gene-tree" interface that enables you to create a virtually unlimited number of natural- and supernatural-looking seamless textures without spilling a drop of paint.

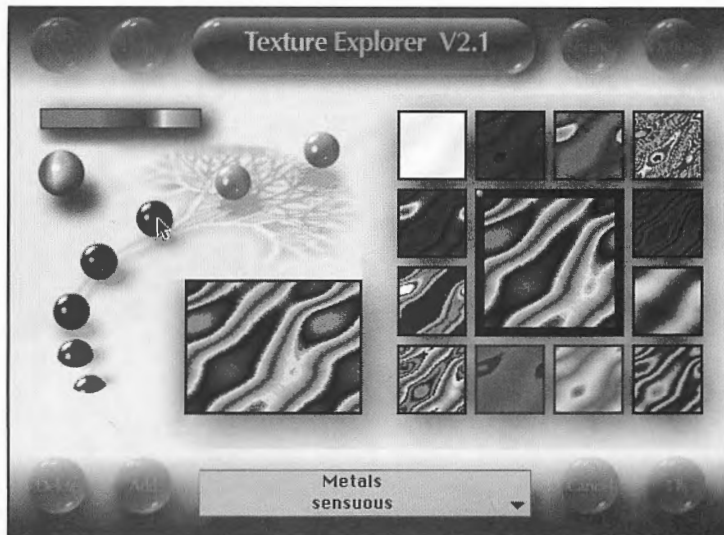
To use the KPT Texture Explorer:

1. Create a new document in Photoshop, setting the resolution to 512 pixels square. Choose KPT Texture Explorer from the Filter menu, which brings up the unusual texture picker (see figure 7.49). Choose a base texture from the menu at the bottom of the screen, in this case "Sensuous."

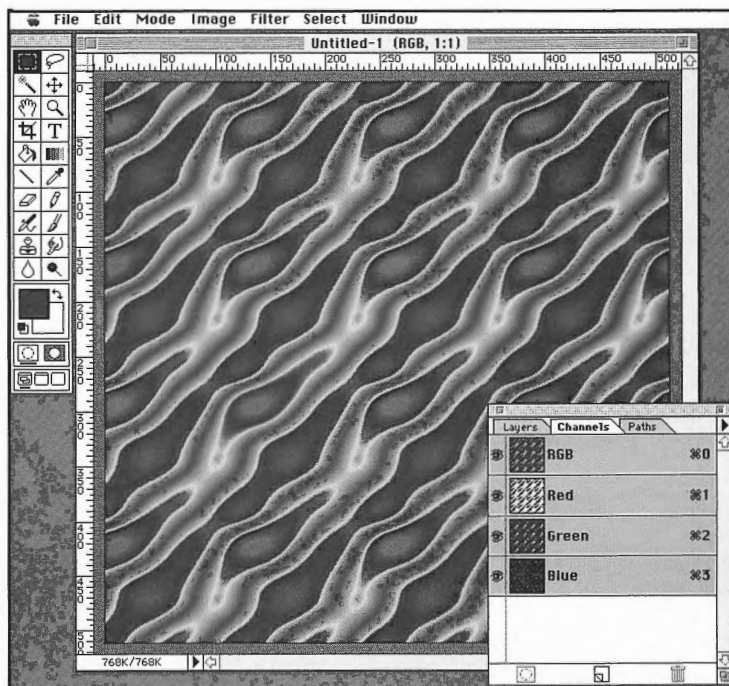


**Figure 7.49** *The Kai's Power Tools Texture Explorer*

2. Click the buttons on the "gene tree" at the left of the screen to mutate the textures until a texture appears that approximates the one you're after. (Clicking near the top of the tree causes great variations, while clicking low on the tree causes more subtle changes.) The color button, to the right of the gene tree, will change the color schemes of the textures in the texture ring without changing the selected pixel geometry (see figure 7.50).
3. Drag down on the central image tile to select 512 pixels as the tile size and click OK to fill the Photoshop image with the new texture (see figure 7.51). Notice that the texture has tiled seamlessly to fill the selection. Save the texture as a PICT.

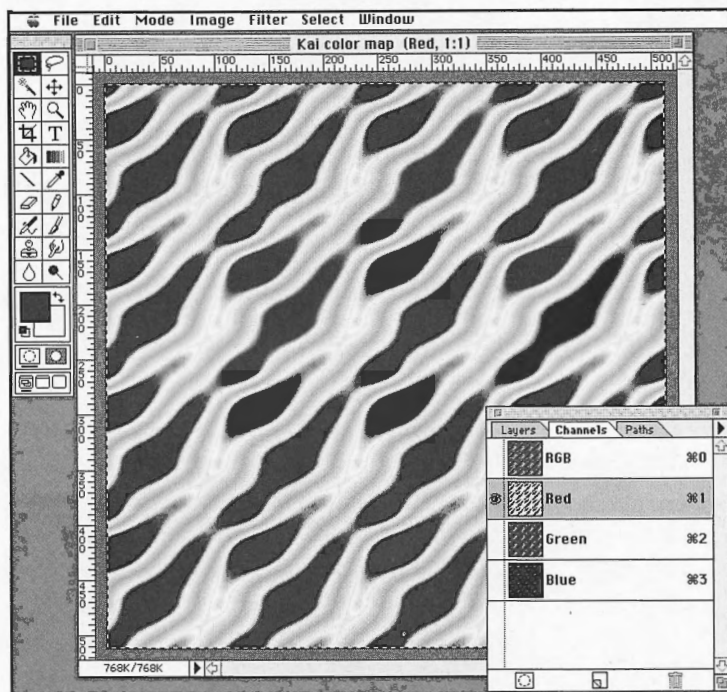


**Figure 7.50** *Mutating the chosen texture*



**Figure 7.51** *Filling the image with the selected texture*

4. Select the Red channel in the Channels palette. Note that this has very high contrast black and white shapes. Choose the Split Channels command and Save the red channel as a separate grayscale texture. You'll use this as a transparency map with the color texture.

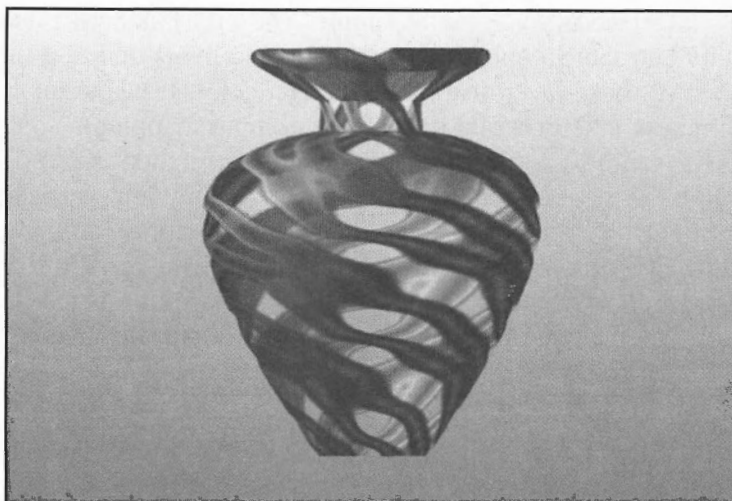


**Figure 7.52** Using the Red channel

5. In your 3-D program build a vase and then create a texture, using the color PICT as the color map and the grayscale PICT as the transparency map. Use cylindrical mapping to get a good vase texture and render the image.

**Tip**

*Moiré patterns* are distracting grid-like patterns that result when you create a “screen” of an image that already contains finely-spaced lines or grids. This effect is often visible in printed material that reproduces previously printed material. Moirés can show up in renderings when you use a texture with very finely-spaced regular lines or patterns. Try to avoid this when you’re creating textures.



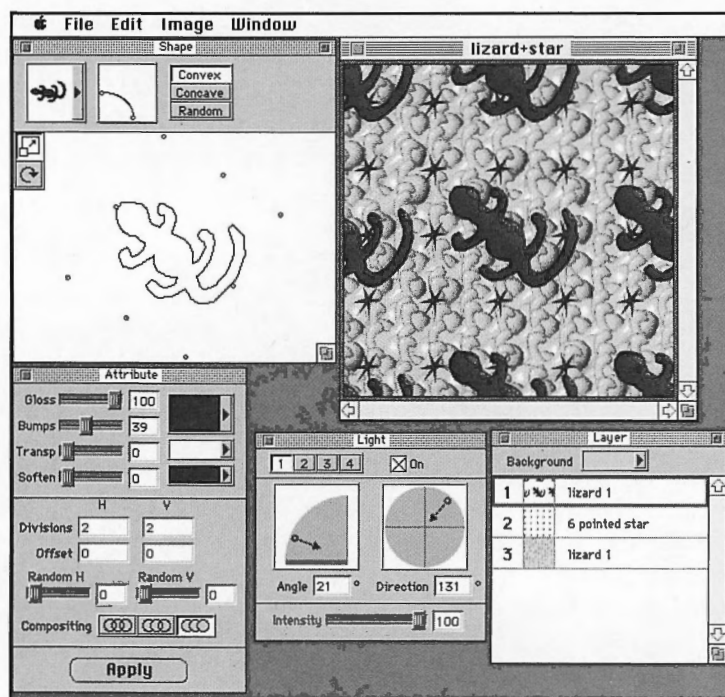
**Figure 7.53** *An image rendered with the texture created from Kai's Power Tools*

## TextureScape

The coolest texture maker to come along in the last year is a program called TextureScape, from Specular (developer of Infini-D). Unlike Kai's Power Tools, which works on the concept of a browser into infinitely deep worlds of more or less randomly generated textures, TextureScape requires you to build your textures more or less from scratch. However, it has three really huge advantages. The first is that it uses EPS outlines, such as those created in Adobe Illustrator, as the starting point for textures. This means that you can create textures out of recognizable shapes you choose. Another advantage is that textures are fully editable, so once you've started to compose a texture, it's easy to change color schemes, knockout backgrounds for creating effects maps, and totally transform the appearance of your textures. A third advantage is that you can generate animated textures; this poses all kinds of great possibilities for undulating surfaces.

Because it's so easy to edit the layers that make up a texture, it's very easy to create bump maps, transparency maps, reflection maps, and so on, that match the color texture you're creating. By creating animated versions of these effects, you can easily build textures, parts of which appear and disappear—or appear to throb with an internal glow.

TextureScape's interface is really quite simple. You add PostScript elements (clip-art fonts such as LetraSet's Fontek fonts are fantastic here) one at a time to their own layers. Each of these layers can have its own color, transparency, bump height, and shininess (the program uses its own bump mapping). You also can set a directional light source. Each of the layers can be stretched, rotated, tiled at a regular interval, or with a random offset both horizontally and vertically. With this combination of controls, you can get amazingly deep textures in only a minute or two. The starting EPS artwork is hardly important because you'll often use many layers and transparencies so that the starting shapes are unrecognizable in the finished texture. The program also ships with a pile of starter shapes.



**Figure 7.54** TextureScape makes it incredibly easy to build very complex textures quickly from existing EPS art

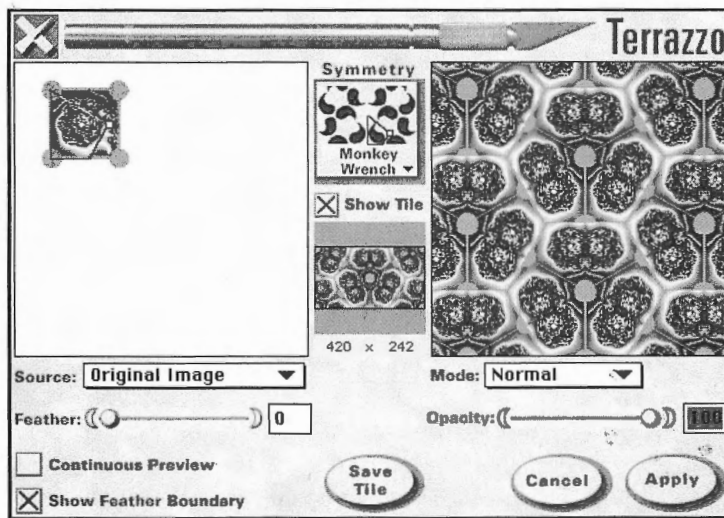


For animated textures, you simply set key frames and make changes to the layers; all of the changes are interpolated for the number of frames between key frames. The final step is to render a texture at the desired size and resolution, or to render a QuickTime movie (it even does field rendering for video—which is defined in chapter 9, “Animation”).

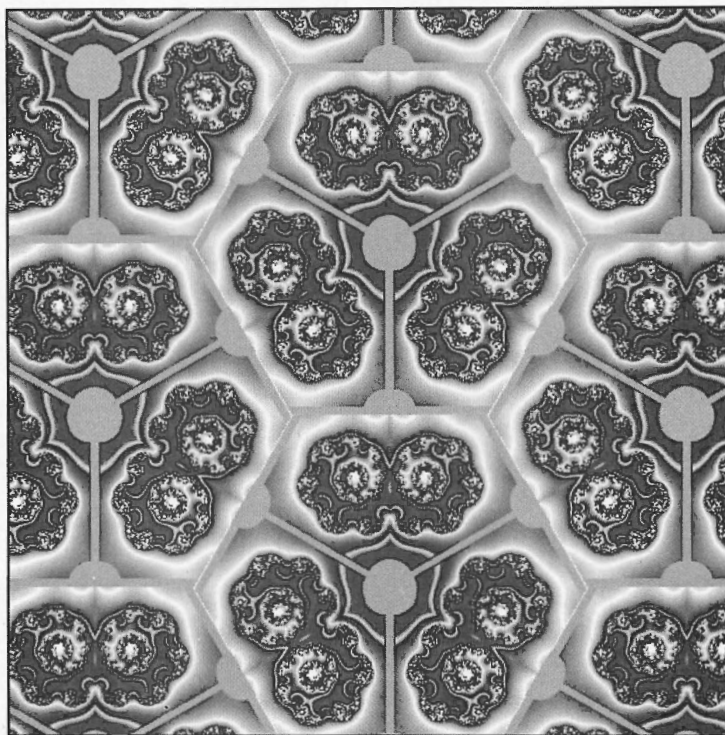
## XAOS Tools Terrazzo

Terrazzo is another Photoshop plug-in that’s specifically designed for creating textures. It has a little in common with TextureScape, in that what you create begins with original art that you create elsewhere, in this case, Photoshop. It can even be used to create cool effects with Kai’s Power Tool textures. Terrazzo generates orderly kaleidoscopic patterns from whatever you happen to have selected in the image preview window. You can adjust the rotation and transparency of tiles to quickly generate very abstract textures from ordinary things. One nice aspect of this approach is that textures automatically use the same colors that are already in your image so things tend to match well.

Terrazzo is a great tool for creating things like ceramic tiles, and it’s well worth having in your texture-making toolkit.



**Figure 7.55** Terrazzo begins with a selection from a current Photoshop image and tiles it in kaleidoscopic ways



**Figure 7.56** *Terrazzo makes it very easy to create complex tile patterns out of simple images*

### **Gallery Effects and Paint Alchemy**

While not exactly texture generators, these “filter” plug-ins are very useful for generating cool textures for use in 3-D. Each has its own strong suit: Aldus’ Gallery Effects gives you a wide variety of textural effects that you can apply to an image, while Xaos Tools Paint Alchemy gives you very fine control over the style of effects applied to an image.

Rather than actually generating textures from scratch, both of these programs make it possible to create images like charcoal, or mosaic looking textures out of existing bitmapped images. They’re especially interesting when you apply them to one of the red, green, or blue channels of an image—or when you use them in conjunction with tools like Kai’s Power Tools, Terrazzo, or TextureScape. These tools also come into their own when you want to do post-processing on rendered objects. For example, you can take a rendered house and make it look as though it was sketched with charcoal, or you can take a rendered spaceship and make it look as though it’s glowing with wild electrical charges.



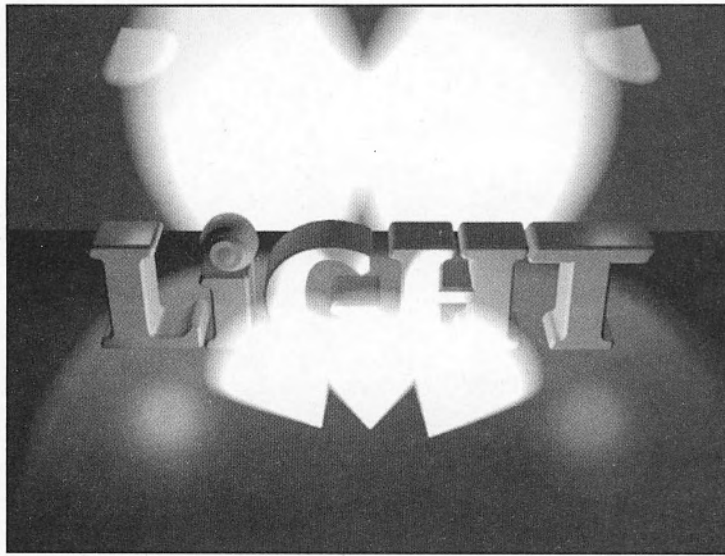
**C h a p t e r**

**8**

# **Lighting**

**M**acintosh 3-D, at least today, is more like virtual photography than virtual reality. A person who understands the principles of photography is much more likely to succeed at creating interesting and evocative artwork than one who does not. And photography, above all, requires an understanding of light.

Light does more than just make objects visible; it sets the mood of a scene. Lighting reveals that which is important—hence the origin of the word highlight—and leaves in darkness that which should be imagined. Light provides contrast and color, and brings out form and texture (see figure 8.1). Without creative lighting, objects are rendered flat and textures become dull.



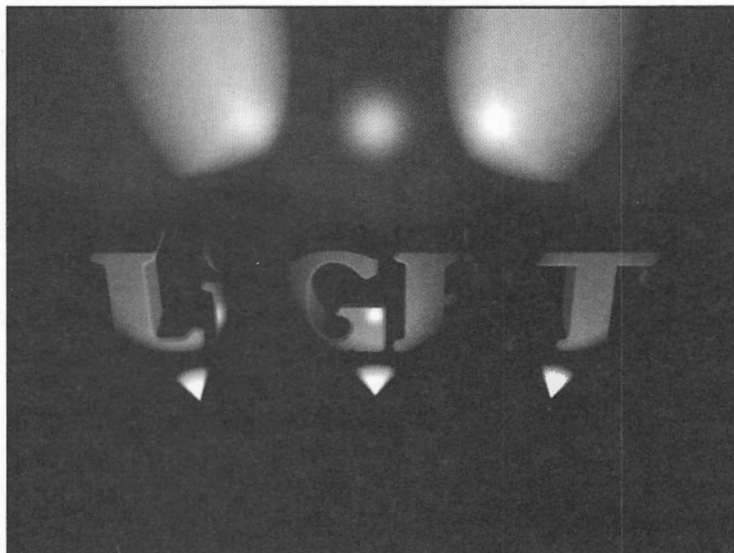
**Figure 8.1** *Rendered in Electric Image Animation System*

Think of the expressions in our everyday language: a “shadowy” figure, the “dark” ages, a “bright” future. In all cases, light—or the lack of it—is a powerful communicator.

The greatest painters, photographers, and filmmakers have long understood light. Rembrandt used “spot” lights to portray the moods of his subjects; Monet was infatuated with the color of light; Ed Weston toyed with the play of light on textures; and Ansel Adams obsessed over contrast between light and shadow.

With the thoughtful placement of lights, 3-D enables you to exploit the power of light to suit a particular task. Of course, most renderings should not look like masterpieces or dramatic productions. Often, placing an object in simple daylight or

under a basic studio light arrangement is the best solution. Choosing the right lights brings a design to life; choosing the wrong lights can give new meaning to the expressions “glaring error” or “dull image” (see figure 8.2). Imagine a table set for two, lit up by a hundred floodlights or an office in the soft glow of flickering candles.



**Figure 8.2** *Underexposed—not all lighting is good lighting*

Even sunlight is not as simple as it first seems. Compare the low, warm glow of early morning to the harsh overhead glare of midday. Successful 3-D artists spend far more time placing and testing lights than modeling. To perfect a single scene, you may do 20 renderings once you have finished modeling and scene building—placing, moving, and adjusting lights each time. The payoff is worth the effort. A well-lit scene will either pop off the page or glow with the subtle mood that fits the moment.

To complicate matters, lights themselves are invisible in nearly all 3-D applications; only the light they cast is rendered. This makes it easy to place lights without having them clutter your scene, but it is problematic when you are trying to create a fireplace with a fire in it or a flashlight with a convincing beam.

**3-D lights**

Electric Image Animation System continues to be the only Mac 3-D package with visible lights: including point, spot, tube lights, and lens flare, although you also can render glowing objects in RenderMan.

Every other 3-D program offers user-controllable lighting, usually including spot, point, and distant lights.

Typestry, StrataType 3d, Dimensions, addDepth, Design Workshop, UpFront, and Shade III are limited to distant lights.

**Daylight**

Normal daylight is the most normal, natural looking light of all. The question is, what does it look like? Direct sunlight is a distant directional light. (Directional lights have parallel rays, unlike a spotlight which “spreads.”) Sunlight, by definition, is pure white. Other lights, which deviate from the color balance of sunlight, are said to have a color cast.

Our brains tend to substitute what we know from experience for what we cannot see. Therefore, even though we may be in a room with very orange-cast tungsten light, we tend to see white walls as white, red apples as red, and a blue rug as blue. When it comes to pictures created by 3-D software, our eyes are not so forgiving (or so easily fooled, depending on how you look at it). If you put an orange light in your scene, you’re going to get a rendering where everything looks more or less, well, orange. This would be fine if we were going to show the image to people whose eyes are adjusted to orange light. But, usually, the people we intend to show the image to will see the color cast for what it is because their eyes are adjusted to normal light.

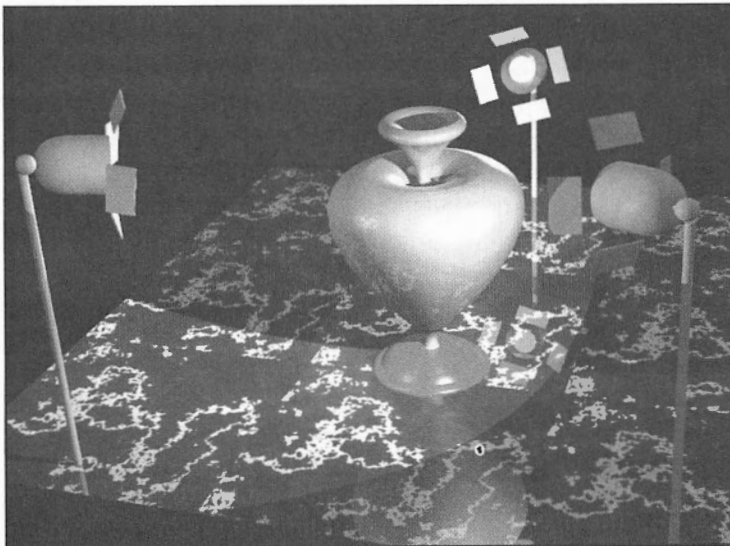
The most pleasing daylight renderings result from simulating early morning or evening sun. Here, you place a strong directional light low to one side, giving it a yellow, or even pink or orange cast, and use a medium amount of blue ambient light. This will cause long blue shadows to fall on one side of the scene with bright highlights on the other. It is a very interesting lighting arrangement.

If you are trying to create the effect of direct overhead afternoon sunlight (one of the worst conditions for photography and usually not highly recommended for 3-D imaging, either), place a strong white distant light overhead and use a large amount of ambient light.

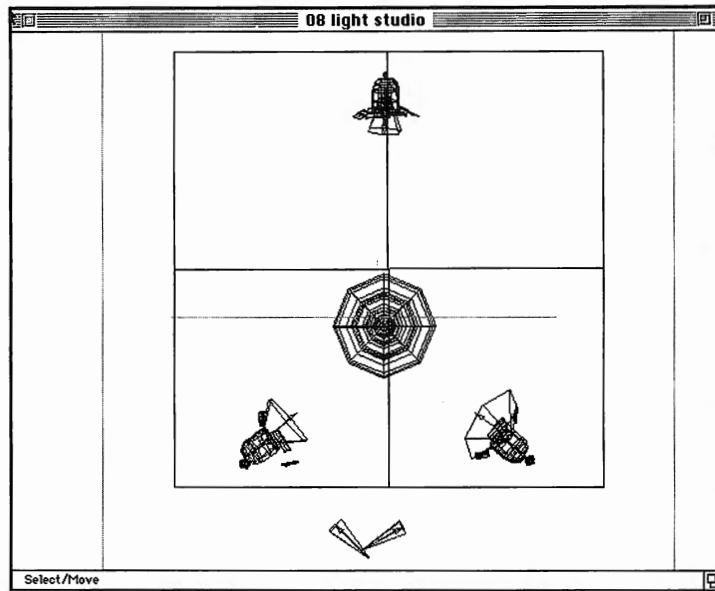
## Studio Lighting

Photographers and cinematographers have a long list of conventions they use as jumping-off points for creative work. One convention is the basic studio lighting setup: a combination of three lights that illuminates the subject, fills in harsh shadows, and provides a pleasing halo around edges (see figures 8.3 and 8.4). The *key light* is the main beam that provides most of the illumination in this setup. You usually position it in front of the subject about 45 degrees off the camera's line of view. Typically, you use a *fill light* to soften harsh shadows cast by the key light. (Vain actors appearing on TV talk shows often insist on one or more fill lights, hidden from view of the cameras, to soften their wrinkles.) In photography, the rule of thumb is to use a fill light that is half the strength of the key light. If the fill light is too weak, the key light's shadows will remain harsh. If the fill light is too strong, it will create its own highlights and shadows which appear unnatural. The final element in this triangular setup is the *back light*. Positioned directly behind the subject, this light picks up highlights on the back edges and enhances contrast between the subject and a dark background. Magazine models owe their glowing halos of hair to this trick, but it applies equally well to models of jet engines and particularly well to translucent objects.

In figure 8.3 the light in the foreground is the key light and provides most of the illumination. The fill light, about half as strong, is in the left foreground; the back light is as strong as the key light, but placed directly behind the subject. All of the lights are a little over "head" high, except for the back light, which is often hidden behind the subject or placed high enough to be out of the picture.



**Figure 8.3** *The standard studio lighting setup*



**Figure 8.4** The lighting setup in Sketch! was used to create the previous image (note the two highlight spots in the foreground [bottom] used to illuminate the light fixtures; a small amount of ambient light also was used overall)

## Night Lights

Night lighting either means artificial lights or the Moon. You can simulate the output of almost any light in most 3-D programs. The one thing that you cannot recreate easily is the visible light itself (except in EIAS). Many 3-D users artfully suggest lights by placing them in shades or other places where, while you cannot see the light itself, you can see the effect of the light on adjacent surfaces. If you want your scenes to have a night-time feel, avoid overlighting them; turn lights down low and set the decay rate realistically high so they do not reach as far. For night-time scenes to have a night-time feel, avoid using ambient light. Shiny objects will pick up a lot of highlights and make a scene sparkle. Be sure to use a bright environment map so shiny objects have something bright to reflect.

## Light Types

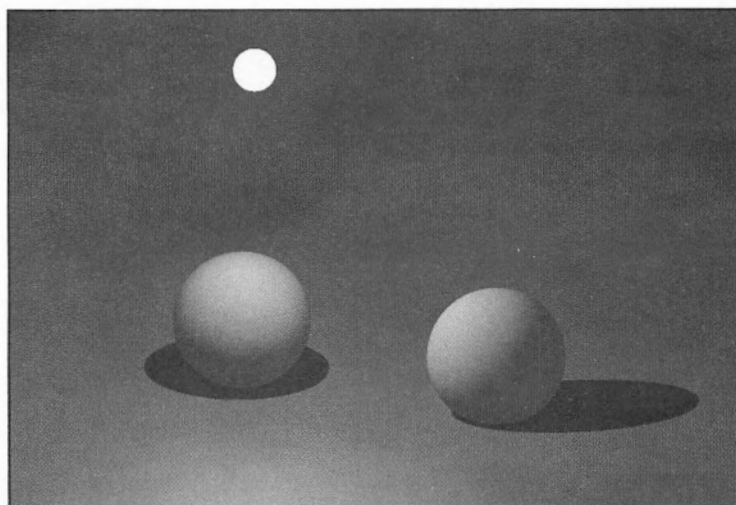
There are many different types of lights in the real world; 3-D software enables you to work with most of the basic varieties. These include point, spot, distant, and ambient light. Some programs (such as addDepth and Dimensions) only support simple white lights set at a fixed distance. Typestry and StrataType 3d enable you to add color and shadow gels to multiple lights, but they are limited essentially to distant lighting. The more sophisticated 3-D programs add the important elements of point and spotlights, as well as the capability to control the level of ambient light. This kind of control is needed for creating photorealistic scenes.

### Point Lights

*Point* lights (sometimes called *radial* lights) are the most common type of artificial light in the real world as well as in 3-D. They work more or less like a light bulb. As the name implies, light emanates outward from a single point. Unlike a physical light bulb, these do not screw in, nor do they have hot or cold spots. You simply point and click with your mouse to place a light. Point lights are useful wherever you need a light bulb, such as when you're showing that there's a lamp in a scene.

Point lights can be employed as fill lights throughout a scene, placing low-intensity illumination near any troublesome dark spot.

Because they cast light in all directions (see figure 8.5) point lights are excellent at catching bevels and curves to provide highlights.



**Figure 8.5** A point light illuminates objects in all directions



In animation, point lights can be locked to moving objects (such as the wing lights on an airplane), attached to the camera (to provide constant illumination of a subject being followed), or moved through a scene to create shimmering highlights off beveled edges.

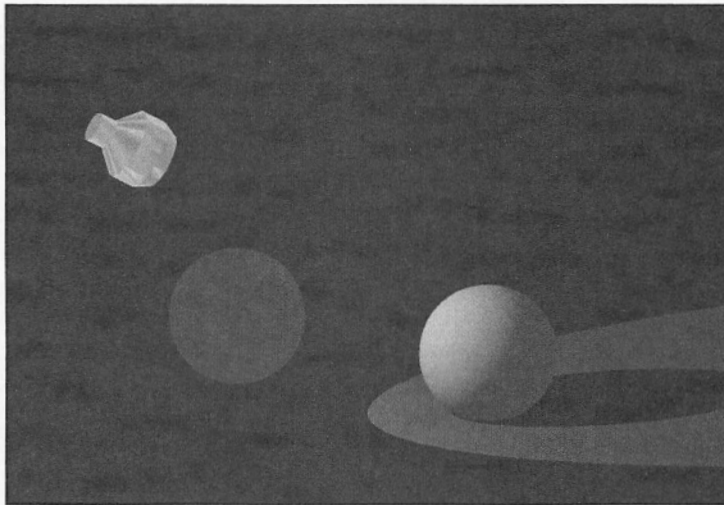
## Spotlight

A spotlight is usually placed and aimed in the same manner as a camera. It has a position in space as well as a direction. Illumination from a spotlight emanates outward in a cone with the point at the source and the wide “base” at the distant objective (see figure 8.6).

Spotlights can be aimed at your subject and their cone angle can be adjusted to illuminate a wide or narrow area. Because they can be precisely aimed, they are the light of choice for picking up small details and placing specific highlights.

Because spotlights can be focused, they also are ideal for casting gel shadows (see “Gels and Gobos,” later in this chapter). Adjusting the drop-off of a spotlight affects the “fuzziness” of its cone; adjusting the decay determines how far the light will travel before it fades away.

Animators use spotlights in many ways: tracking a subject (following an actor across a stage), locking to a subject (a car’s headlights), or even spotting an airplane.



**Figure 8.6** *A spotlight is a directional cone—it begins at a point and ends in a circle*



## Spotlights for Highlights

Lights give shiny objects their glint and dull objects defined detail. At times, however, you won't want to over-light a scene only to bring out a highlight or two, or to brighten the lights on a single subject in the scene. To bring out the highlights and detail on a spoked hubcap, for example, you can aim a small spotlight at the point of interest, but make it narrow enough to miss the rest of the car. The trick with spotlights is to position the light source accurately enough that it only catches the area you want to highlight.

In many cases, a 3-D rendering will lack a desirable highlight in small areas of detail (like the glimmer of gold off the bezel of a watch). You can position a spotlight very close to the object of interest and spread the spotlight's beam just wide enough to create the missing highlight.

### Tip

Missing highlights are usually due to the lack of imperfections and minute details in 3-D models. A slightly imperfect object actually glimmers more brightly than one with very smooth angular surfaces and edges. The most effective way to add imperfections to a surface is with bump maps. However, it's also possible to use a program like Photoshop to lightly airbrush details on an image.

## Ambient Light

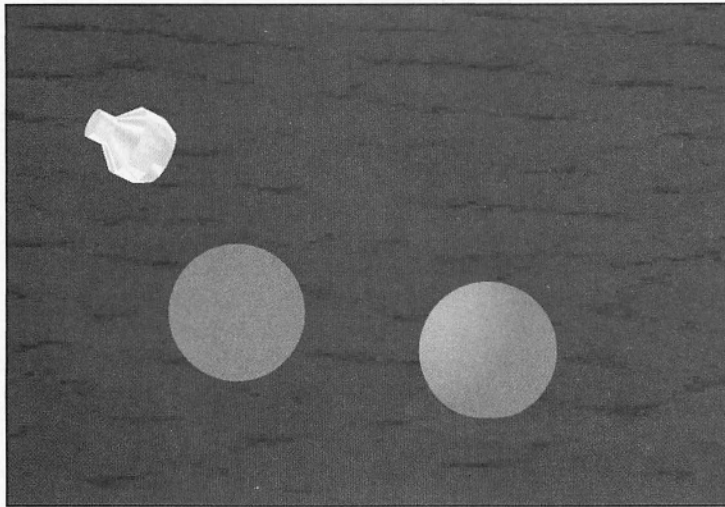
The most common example of *ambient light* is that of normal daytime shade. Even though the sun isn't shining directly on your head, you are not groping in darkness because sunlight bounces and reflects off of everything it strikes, particularly the sky. Even on cloudy days, light is coming at you from everywhere.

Ambient light is important in natural scenes. In fact, most renderers default to some amount of ambient light. Single lights by themselves are rather harsh and give the distinct impression of indoor or stage-type lighting: a reading lamp at night, a spotlight on a stand-up comic, or even a vampire looming over a candle. Ambient light makes these scenes bright and more cheerful; it fills dark shadows and softens details. In fact, very strong ambient light will completely fill even strong shadows (see figure 8.7).

It doesn't make much sense to only use ambient light, since there is rarely an instance when you'll find yourself with no directional light. An example of purely ambient light is a "whiteout" snowstorm. In this case everything appears flat and lacking detail. In general, the more ambient light in a scene, the less contrast there is among objects.

Even though the vast majority of outdoor ambient light comes from the sky (which is pale blue), normal renderings call for white ambient light. As with a colored-fill light, colored ambient light will fill shadows with a subtle cast of color.

Monet often used purples and blues when he painted shadows on haystacks—not a dark version of "haystack color." While Monet was particularly adept at discerning the true colors of light, most people (the author included), have a brain that tricks them into perceiving shade as merely "darker"—and not necessarily of a different color.



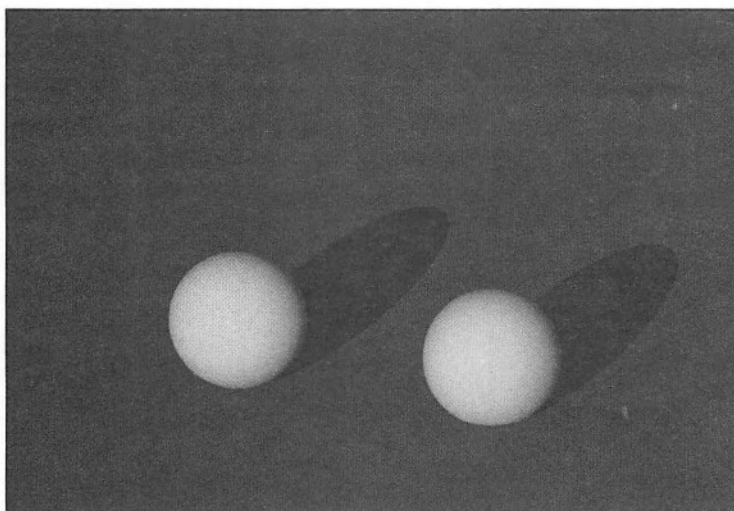
**Figure 8.7** *High ambient light fills in even the harsh shadows of a strong spotlight*

Similarly, if you're rendering an interior scene where the walls of a room are bright orange, you can set your ambient light to a pale orange color. This gives a more realistic look for the scene than plain white light. If it is an all-shade scene, white ambient light appears more natural when rendered. To create the effect of a cloudy day, use plenty of ambient light and very little directional light. This will result in very flat renderings, with little shadow detail.

## Distant Light

*Distant lights* are often referred to as “suns” for good reason: light rays coming from the sun are very nearly parallel. (The sun is a gigantic point of light so far away that Earth only gets a sliver of the rays it projects.) Placing a distant light creates the effect of a solid wall of light washing over your scene from a single direction. All shadows fall in a line and the effect is like direct sunlight (see figure 8.8).

For early morning or sunset light, set your light angle low to the horizon, adjust the color to a soft yellow or pink (or orange for a blazing sunset), and turn down the intensity. Low light often calls for long foreground shadows, so you may want to add a gel (also called a *shadow mask* or *gobo*) to the scene. Because of the harsh shadows created when you use a distant light, you will often want a fair amount of ambient light to soften them. For midday light, place the distant light directly overhead, turn up the intensity, and give the light a white color. In bright sun, the ambient light is closer to pure white than in morning and twilight hours.



**Figure 8.8** *Distant or “sun” light is directional—infinately far away—so it lights the whole scene equally*

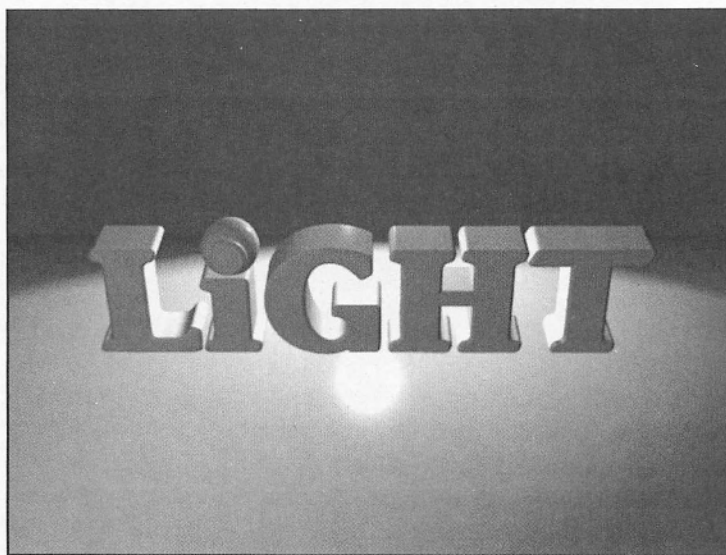
## Pan Light

*Pan lights* are not a light type in 3-D software, but I often use them to generate a broad, highly-diffused light that casts few shadows. In the real world, they are extremely useful for lighting highly reflective subjects (such as glossy cars), where you don’t want the lamp itself to be noticeably reflected in the image. Pan

lights only light one side of a subject, but the light emanates in every direction from every point on the “pan,” so shadows are extremely soft. (Sketch! enables you to set a direction for ambient light, which is essentially the same thing.) You can simulate a true pan light by evenly spreading a grid of three or four diffuse point lights in a grid.

## Back Light

Back lighting emphasizes the shapes and silhouettes of objects, rather than their surface appearance (see figure 8.9). Back lights placed above, or slightly to one side of, an object will emphasize its outline by forming a bright halo around its dark silhouette.



*Figure 8.9* Back lighting can be used to emphasize an object's silhouette

## Setting Lights

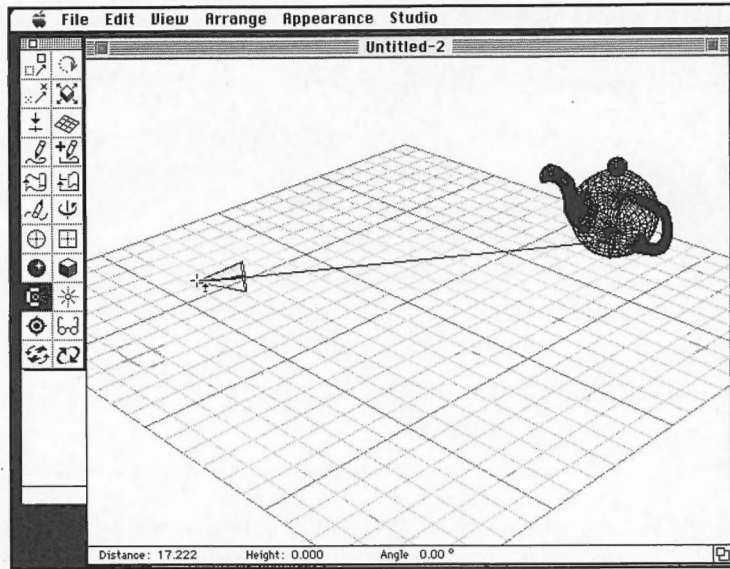
In addition to positioning and changing the type and direction of lights, there are a number of other lighting options that can affect the appearance of your scene. While every scene-building application has somewhat different lighting options, the most common include:

- **Color.** Light color directly affects the appearance of rendered objects. If you shine a yellow light on a white object, the object's likely to look yellow. Adobe Dimensions and Ray Dream addDepth lack colored lights.
- **Intensity.** Intensity or brightness of lights affects the total exposure of the image, as well as the visibility of details. If lights are too bright, the image will be overexposed; objects will appear washed out and lacking detail. Without enough light, the image will render too dark. Shining multiple lights on an object results in a combination of effects. For example, if a model is properly exposed with one light shining on it, it would be overexposed with the addition of a second light of equal intensity.
- **Drop-off** (or fall-off). This lighting option affects how much a light blurs from the center of its beam to its edge. A spotlight with low drop-off has a very sharp edge where it strikes, while a beam with high drop-off is soft or blurry at the edges.
- **Decay.** Attenuation or decay is the distance from the source at which light fades. While in life this is determined by mathematical laws as well as elements in the atmosphere, 3-D lights operate in a completely transparent ether, and the light's decay is determined by an amount you specify. You must set a distance for this effect unless you want your lights to carry all the way to the moon.

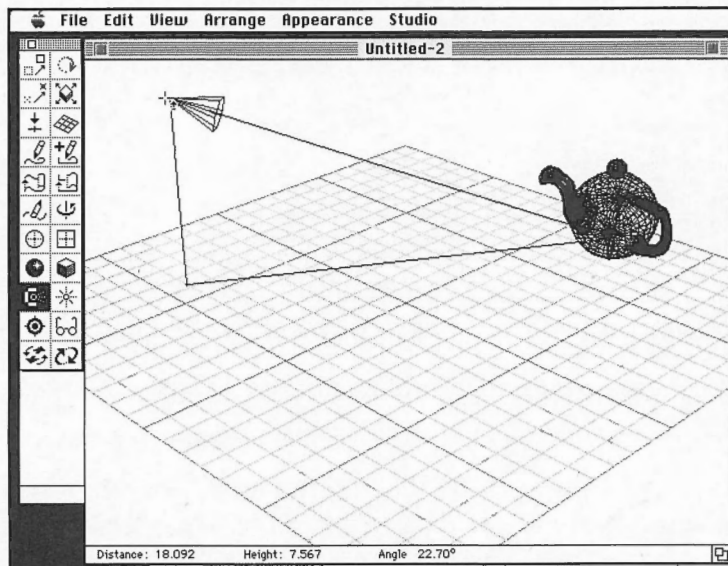
## Setting Lights in Sketch!

Sketch! has a unique and simple method for light placement. Clicking the light tool on the scene anchors the light's target; dragging the light moves the light source to a new destination while maintaining its aim at the target.

1. Select either the Spot or Point Light tool and click the target for the light.
2. Drag the light away from the target along the surface of the working plane.
3. The light automatically aims at the target as you drag it (see figure 8.10).
4. At any time, you can press the Shift key to change the light's movement away from the working plane.



**Figure 8.10** As you drag a light, it is automatically aimed at the target



**Figure 8.11** Select shift and drag to move the light up or down

## Visible Lights and Lamps

Light sources themselves are invisible in 3-D. However, you can create a realistic light bulb with an unrealistic trick: create a light bulb-shaped object with a transparent, glowing surface and hide a point light inside it. Because the surface is transparent the light will shine through. But, because it is glowing brightly, you cannot see through it.

Realistic looking lamps and other light sources are an important part of scene building. Your renderings will be much more convincing if you take the time to learn how to make visible lights.

For example, when creating a pan light, you can put your grid of light bulbs behind a plane of brightly glowing transparent glass. Not only will this look realistic, but you'll even get the right reflections off objects in the scene.

## Glowing Objects

While Electric Image has convincing glowing light sources, if you're not using this high-end rendering package, you'll need to adjust your techniques.

Sometimes the need will arise to make an object (such as a strip of neon), act as a light. To do this, you must assign a glow surface quality to the object. The difference between a glow and a true light is that even though they may shine very brightly, glowing objects do not shed light on other objects. A neon tube comprised solely of a glow will not seem very realistic unless it casts a colored glow on the wall where it is mounted.

To compensate, you can make the glowing object "transparent" and hide an appropriate type of light inside it. You can then apply a glow map to adjacent surfaces that create the effect of casting light. While there is no such thing as a glowing, completely transparent sphere in the real world, in 3-D, such an object with a point light inside it looks and acts much like a frosted light bulb. The transparency of the surface enables the internal light source to shine through.

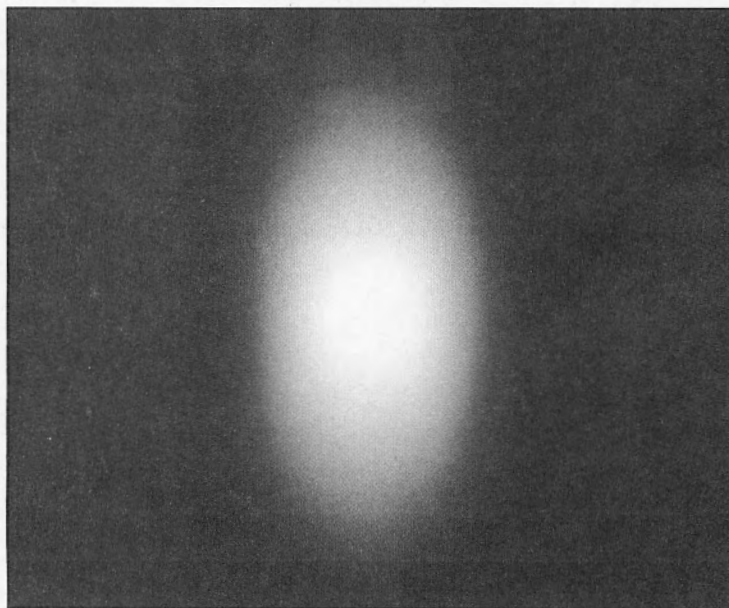
The biggest advantage of using glows, wherever possible, instead of actual light sources, is that glows are texture maps (and therefore render many times faster than adding extra lights). Rendering times are more or less proportional to the number of light sources in your scene, so you want to use them sparingly.

In figure 8.12 a "glow map" is created to simulate the appearance of light shining through a translucent lamp shade. Figure 8.13 shows the result of this technique.



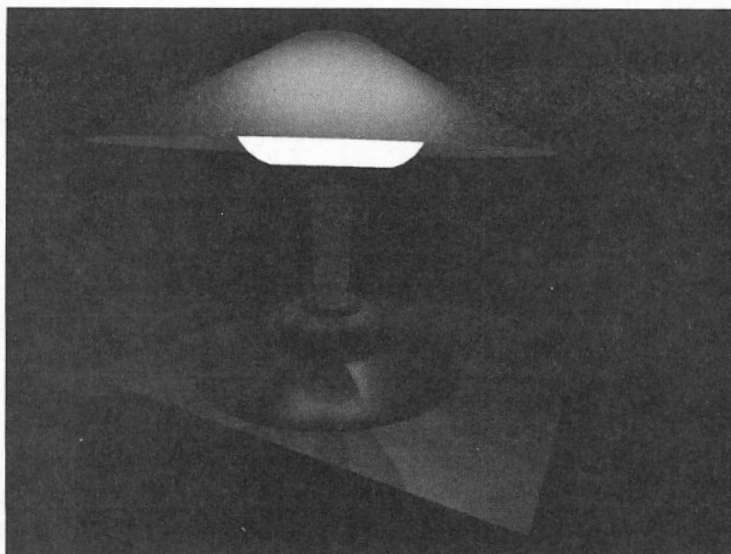
**Separate lighting**

If perfect lighting in all parts of a scene is critical, you can render parts of your image separately with alpha channels (see chapter 11, "Working with Images"), and then combine them in an image editing program. This way, you can set up one set of lights for one part of a scene and another set of lights for other parts. When you render group "A," you simply hide all of group "B"; when you render group "B," you hide all of group "A." This is an incredible time-saving technique because you don't have to fuss over the exact placement of lights, and you don't have to worry about the combined effects of all of those lights in the same scene. In terms of speed, this technique also is effective because your rendering times decrease dramatically when you hide lights. The one drawback of this system is that shadows and reflections will not fall from rendering "A" onto objects in rendering "B."



**Figure 8.12** *This glow map is used to give a realistic glow to the lampshade*





**Figure 8.13** This lamp uses a glow map on the shade, as well as a glowing transparent light bulb containing multiple point lights (note the four separate shadows on the table)

## Shadows

Light hitting an object is partially reflected and partly absorbed, and in so doing, the object “throws” a shadow onto objects behind it. (You can think of shadows as objects blocking light, or “casting” darkness.)

In the real world, objects can cast shadows from many light sources. But because the shadows from the strongest light source are the most visible, and because shadows are often as complex to render as objects, rendering is much faster if you limit the number of shadows in your scenes. While overlapping lights and shadows can provide realistic effects, be aware that every shadow in your image will significantly increase the overall rendering time. Some programs will let you turn off shadows for individual objects to reduce this problem. The capability to turn off shadows is a clue to the trickery performed by renderers. Even though shadows are disabled, shading of objects is retained. In the real world, of course, you don’t have one without the other.

Gouraud and flat shaders cannot create realistic shadows; however, with an image editing program that supports masks, it is possible to simulate simple shadows.

The most realistic shadows are rendered with radiosity (currently offered only by StrataStudio Pro). However, this rendering technique is too slow to use with any frequency.

Typically, real shadows have a soft indistinct edge, but only a few 3-D renderers have an option to achieve soft edge shadows, including Sketch!, ELIAS, and RenderMan. With these renderers, increasing the amount of drop-off also softens shadows.

## Gels and Gobos

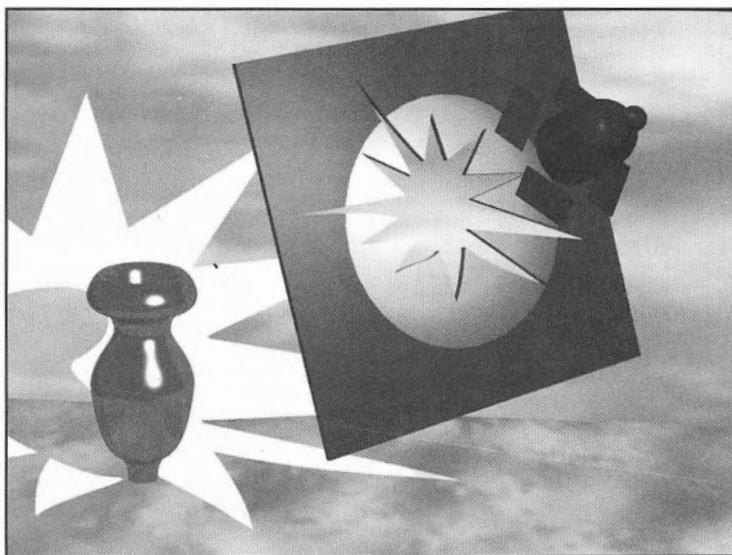
*Gels* enable you to throw environmental shadows into a scene. Some applications (such as Designer, Showplace, and StrataStudio Pro), enable you to map a texture to a light. For example, a silhouette of a window can be applied to a spotlight and projected onto a scene. The result is the realistic effect of light streaming through the window (see figure 8.14).



**Figure 8.14** A gel is a mask applied to a light—it creates the effect of light streaming through a window

If your scene builder doesn't offer gels, you can model a wall with a window (by creating a flat plane with holes cut out for windows) and place it just outside the visible part of your scene with the light shining through. Shining a light through a

cut-out in this manner is called a *gobo* in stage lighting (see figure 8.15). The same technique can be used to create light shining through curtains, bars, stars, or Venetian blinds. By shining a light through panes of stained glass, you will cast colored shadows into your scene if you are rendering with a ray tracer.



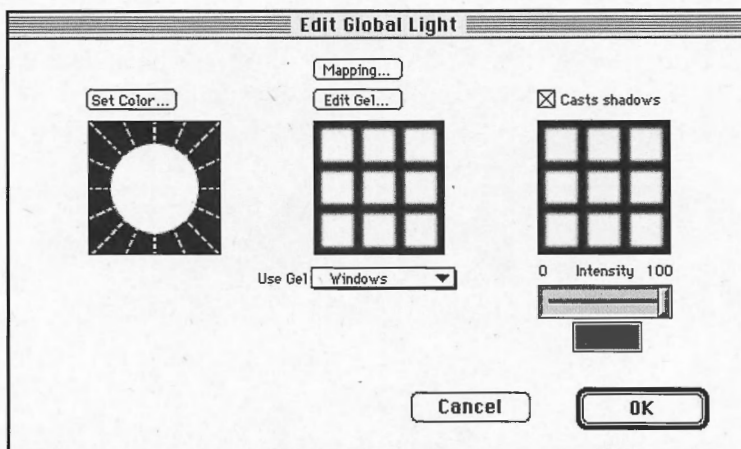
**Figure 8.15** A gobo is an object with a cut-out that you can shine a light through (this was modeled and rendered in Sketch!)

When rendering for animation, you can put light behind a revolving ceiling fan for Casablanca-style lighting. For a Bugs Bunny cartoon effect, you can use a cut-out of a looming man in a hat that casts a growing shadow as the villain approaches.

### Using a Gel in StrataStudio Pro

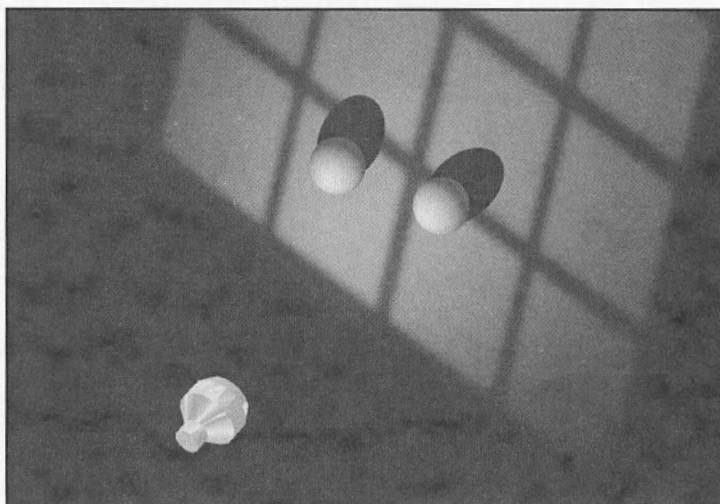
Using a gel in StratStudio Pro is almost as easy as using a light alone. Essentially, you are attaching a texture to the light source.

1. Set up a model and aim a spotlight to illuminate it (this creates a stage-lit effect; for more natural lighting, you can use a point light instead). Double-click on the lighted object to bring up your setting options found in the Edit Global Light dialog box (see figure 8.16). Select a gel in the pop-down window.

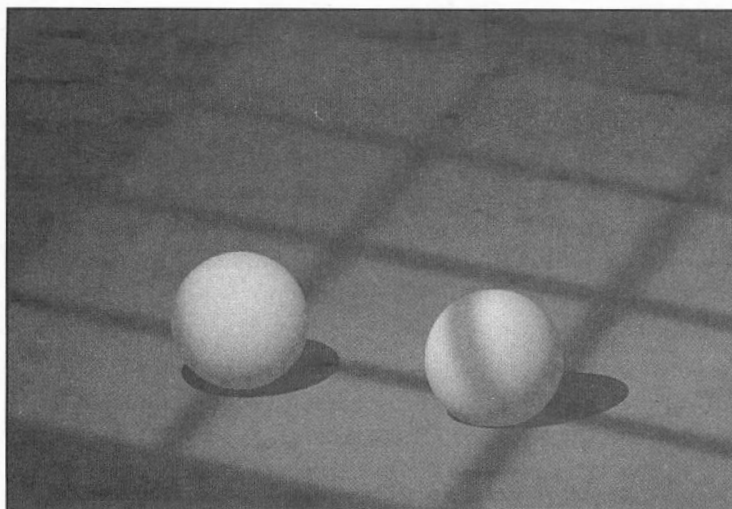


**Figure 8.16** StrataStudio Pro's Edit Global Light dialog box for setting options

2. Render the scene. The top view shows the effect of the gel from a distance (see figure 8.17).



**Figure 8.17** The use of a gel from a distance



*Figure 8.18 A typical rendering with a gel*

## Electric Image Lights

As I've mentioned several times previously, the Electric Image Animation System is the only 3-D program that offers visible lights: a point light can glow like a bulb or you can use tube lights that realistically portray straight fluorescent tubes or laser beams.

EIAS uses an adjustable inner and outer light to create the effect of an intense light source with a fuzzy haze around it. This works for point, spot, and tube-style lights. Unlike glowing objects in other 3-D programs, EIAS' glowing lights do shine light on other objects. A cool recent addition to EIAS is automatic lens flares. These are the lighting artifacts that occur when a light refracts off the surface of a lens and causes a starburst on the film; it's a very realistic lighting special effect for mimicking movie camera work.

While other Macintosh renderers, to date, cannot produce the lens flare seen in photographs, a simple filter, Lens Flare (included with Photoshop), creates this effect fairly easily, though it won't work in animation.

## Time-of-day Lighting

One interesting technique—offered primarily by programs designed for the visualization of architectural models—is lighting that corresponds to a particular time of day on a particular day of the year. UpFront, Design Workshop, and Sculpt all have a similar feature. In UpFront, when you build a model and orient it to the compass, you also can give it a latitude on the globe (see figure 8.19). The program is capable of calculating and displaying the lighting on the model based on the position of the sun at a given time of year. This enables a designer to see how a courtyard will be lit in the winter, or to see how the sun will strike the windows in the hottest part of a mid-summer's day. When generating an animation, you can set the program to calculate the sun's position at each frame so that you can see how the building looks over the course of a day.

Start Time		End Time	
Mth	Day	Hr	Min
21	7	0	0
		<input checked="" type="radio"/> AM <input type="radio"/> PM	
Sun Altitude 26.70		Sun Altitude 23.21	
Sun Azimuth 82.70		Sun Azimuth 280.58	

Latitude	
45.0	<input checked="" type="radio"/> North <input type="radio"/> South

Movie Timing	
Duration	620 Minutes
Interval	20 Minutes
Total Frames	31

☐ Display Date & Time in Movie

**Figure 8.19** The time-of-day calculator in UpFront automatically calculates sun position for lighting a model

**C h a p t e r**

9

# **Animation**



**A**lthough cell animation has been around for a century, computer-based 3-D animation has many advantages not available to animators using traditional film and cell techniques. In fact, the two can hardly be compared in any fair manner.

Two of the most important differences are the infinite level of control available in 3-D, and the fact that 3-D cameras need not bend to the limitations of depth of field and film. Three-D animators are not necessarily bound to the laws of physics, so cameras can fly through walls unscathed and dishes can fly around a kitchen possessed. You can zoom in or out at will, unconcerned with the pitfalls of changing exposures or blurred foregrounds.

Further, in digital animations, an unlimited number of layers can be superimposed without loss of image quality underneath. Traditional animators, however, are limited to four or five layers before visuals turn to mud under sheets of celluloid.

Another of the most important advantages is the labor and time saving advantages of computer 3-D animation. Traditional animations are the result of thousands of hours of tedious frame-at-a-time hand drawings, while the Macintosh can do most of the work for you when animating in 3-D. If you create a scene that needs to be changed, it is a simple matter to go back, rearrange key parts of the animation, and re-render it.

## Animation Basics

It is axiomatic that the shortest distance between two points is a straight line. In the vocabulary of 3-D, the easiest way to create an animation also is between two points. Suppose you want to create a four-second animation of a hot-air balloon leaving the ground and floating up and up until it rises out of the picture. By simply defining the beginning and ending stages of the animation (the balloon is on the ground and then the balloon is out of sight), you have the two points needed to make a movie. Three-D animation, in most ways, is identical to basic scene building. To make the hot-air balloon movie, we build the scene with the balloon on the ground, and then change the scene by dragging the balloon up and off the screen. The software keeps track of where the balloon is at the starting point, as well as where it is at the end. The software then does what you would rather not do yourself—it calculates the path and position of the balloon in all the frames in between. This technique is called *key-frame animating* and it is the basis of even the most complex and creative 3-D animation. An important extension of the key-frame concept is event-based animation, which lets you create nested key frames. There is more to animation than moving an object in a straight line between two points; you also can animate surfaces so that skin turns to stone; or animate lights so



that night becomes day; or even animate the cameras that provide the viewpoints of your scene.

## **Key Frames and Events**

Key-frame animation is based on the principle that you define the state of scenes and models at important (key) points during an animation and the computer interpolates or “tweens” all of the frames in between. This is an enormous time saver, enabling you to act as a director of action, with the computer doing all the busy work of calculating intermediate steps.

Key-frame animations are not limited to just the linear motion of models through a scene. You can define as many key frames as necessary to describe a scene. Depending on the software, you may be able to rotate, stretch, squash, or otherwise deform models; change lights and textures; and move the camera to follow and zoom in or out on the action. Virtually any property can be animated by defining its condition at the start and end points in time. Even special effects like morphing one model into another are possible in some programs.

There are a number of factors you may want to animate with key frames, including:

- Object position
- Object rotation
- Object dimensions
- Object texture
- Object shape
- Camera orientation
- Camera rotation
- Camera focal length
- Light intensity
- Light color
- Light direction
- Environments and backgrounds

Not all animation programs support all of the possible animation variations (and this list hardly covers them all), but the majority of these types are supported by higher-end programs and lots of special effects (such as explosions and sounds) that are beginning to appear on the Mac scene as well.

## Scene vs. Event Key Framing

There are actually two distinct kinds of animation available in 3-D programs. The most basic type is where a key frame describes the state of the entire world at a given time; this is called *scene-based key frame* animation. More powerful 3-D animators enable you to create more complex motions by defining *key events*. Rather than defining the state of the entire world at a given point in time, *event-based key framing* offers separate timelines for each group and each object within the group. The more powerful the animator, the finer the divisions to which you can assign separate key frames. This enables you to establish motions for objects that are independent of what is happening in the rest of the scene. For example, Electric Image, the master of event-based animation, enables you to individually animate x rotation and y rotation for the front face of a single character of a type logo. Macromedia Three-D has similar capabilities.

With event-based animation, a hot-air balloon pilot could fire his burner several times as the balloon rises off the screen and someone falls out of the balloon. This kind of nested animation can be extremely sophisticated with gears turning, pendulums swinging, fish jumping, lights flashing, and the camera moving all at the same time. With simple key frame animation, it's impossible to maintain fluid motion in one object while allowing for "sub-animations."

Scene-based key-frame animation, meanwhile, is adequate for creating "fly-bys" and other animations popular with architectural renderers. For example, you can have the camera fly around an object 360 degrees—all the while aimed at the object's center. This generates an animation as if viewed from the window of an airplane flying around the model.

### Tip



Alias Sketch!, which doesn't allow for animation at all, will enable you to batch render every view set up in a scene with its RenderQ! utility (see "Batch Rendering," in chapter 10.) When you set up your model in Sketch!, make a single camera aimed at the subject, duplicate it,

rotate the duplicate (which will stay aimed at the subject), and use auto-duplication to make one camera for every frame you want to generate. Save the model and open it with RenderQ!. Make sure you check the Render All Views dialog box before rendering. Sketch! will generate a frame number for each image it renders. You can generate a pretty passable fly-by animation of Sketch! scenes this way. MovieMaker, a shareware utility on the CD-ROM, converts numbered PICT files to QuickTime movies.

Event-based animation allows one model to go one way at its own pace, while the camera goes another way along its own timeline. This is the kind of animation you will need, for example, if you want to do a fly-by of a bouncing ball where the camera flies completely around the scene in 10 seconds, while the ball bounces on the ground several times and rolls away. The camera requires two key frames: one at zero and another at ten seconds. The bouncing ball, meanwhile, requires key events for movement up or down repeated several times at different intervals. More importantly, realistic animation requires that objects accelerate and decelerate when they stop and go, and objects need to move with their own rhythm. This kind of animation is a huge hassle without event-based key framing.

Another feature in some event-based animators is the capability to cycle parts of an animation. For example, you can select an object such as a bicycle crank, and specify that it should rotate once every half second. By telling StrataStudio Pro to cycle this animation for ten seconds, the crank turns 20 times as the bicycle rolls across the screen.

Not surprisingly, Mac 3-D programs have been flocking to event-based systems. When I first compiled the list in the following tip, two-thirds of the programs with animation performed only simple key frames. Now, well over half do event-based key framing with multiple timelines. Those that don't are primarily designed for architectural work, where the vast majority of animation is simply fly-bys and sun studies. The exceptions are old and outdated programs, or those primarily intended for still imagery.

**Tip****Key-frame animators:**

- Crystal TOPAS
- Modelshop
- Swivel 3-D Professional
- StrataVision 3-D
- Sculpt 4D 4.0

**Architectural key-frame animators:**

- DesignWorkshop
- DynaPerspective
- UpFront!
- Turbo 3-D

**Event-based animators:**

- Presenter Professional 3.0
- Shade III
- Typestry 2.1
- Electric Image Animation System
- Macromedia Three-D
- Macromedia Extreme 3D (rumored)
- StrataStudio Pro
- Playmation
- Animation Master
- Infini-D

**Virtual reality animators:**

- WalkThrough
- WalkThrough Pro
- Virtus vr
- Strata Virtual 3d

## Timelines

Nearly every 3-D animator that supports event-based animation, uses a timeline or sequencer. This is a grid that represents time across the top, with objects forming the rows on the left. Markers representing key events for each object can be placed on the grid itself. Every object has its own timeline and its own event marks (see figures 9.1 and 9.2). This approach is shared by Macromedia Extreme 3D, Infini-D, StrataStudio Pro, Typestry, Presenter Pro and EIAS. Shade III and Playmation are two animation programs that don't share this approach.

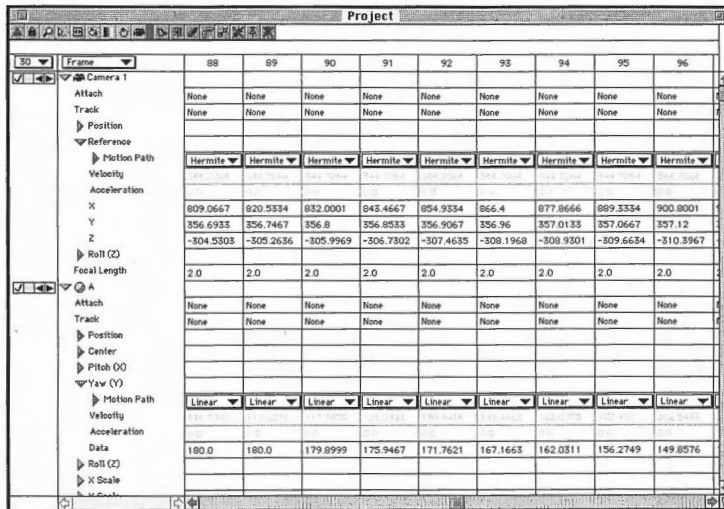


Figure 9.1 Electric Image's animation sequencer

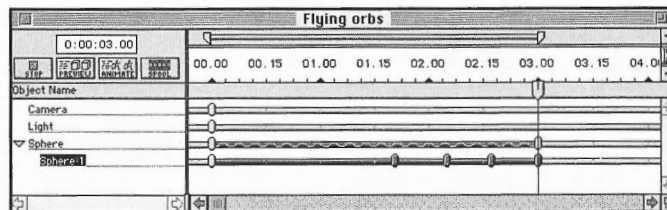


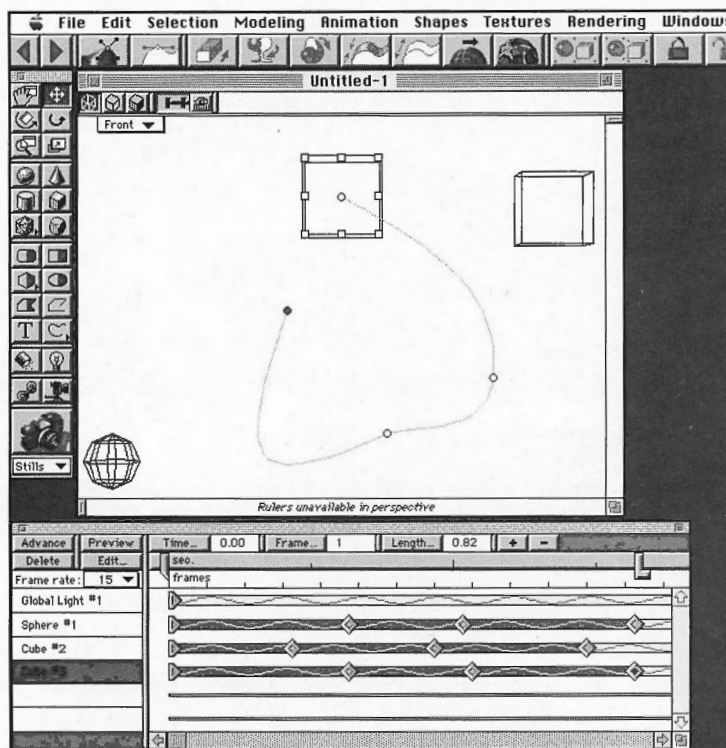
Figure 9.2 Infini-D's animation sequencer

Adding key frames in a sequencer is simple: You specify a position in time by dragging a slider or by clicking a cell. Then anything you move or change will get a new key event at that point in time.

Will Vinton's Playmotion has a unique approach to timelines; it offers separate graphs for every motion parameter (along the lines of Electric Image's), on which you can adjust an object's motion values by dragging handles on the graph against time.

Shade III enables you to animate objects completely independently of one another.

StrataStudio Pro's Animation Palette, similar to EIAS, is directly linked to the motion curves of objects. You can drag these curves around within your scene to quickly compose a satisfying motion (see figure 9.3).

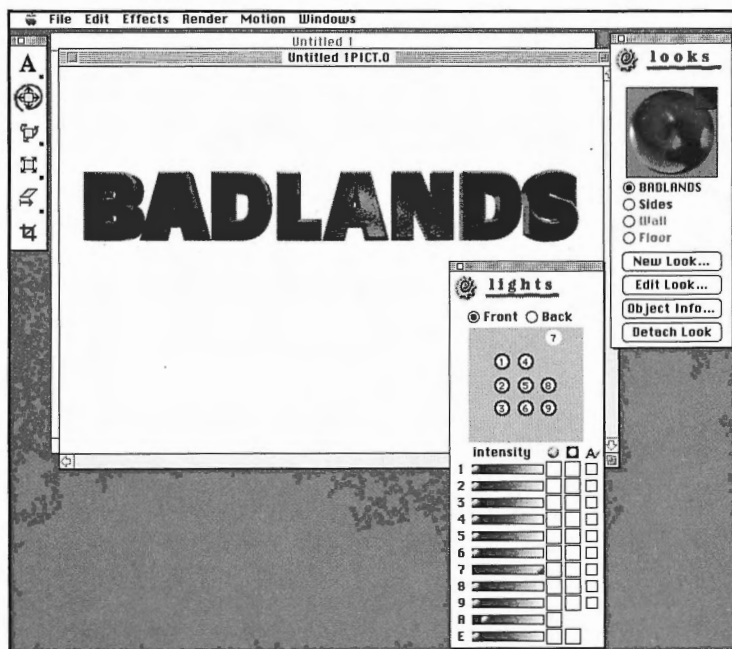


**Figure 9.3** In StrataStudio Pro, you can drag curves to set motion paths

## A Simple Flying Logo

The following sequence shows the creation of a simple flying logo in Pixar's Typestry. Nearly all of the techniques are identical to creating a static scene for rendering a still logo.

1. Generate a type logo by entering a text string.
2. Specify the font, size, bevel style, and scale it to size.
3. Apply a texture and set a light source (see figure 9.4).

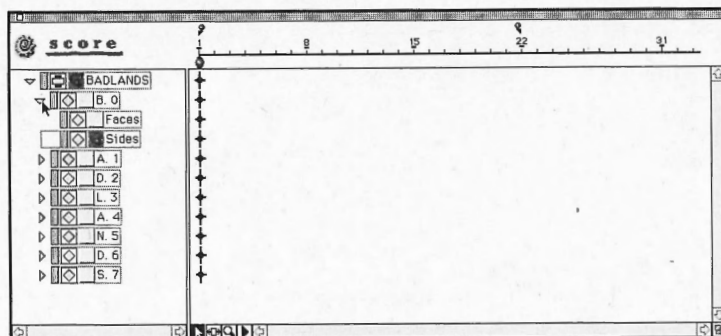


**Figure 9.4** Creating a logo

4. Open the Score window.

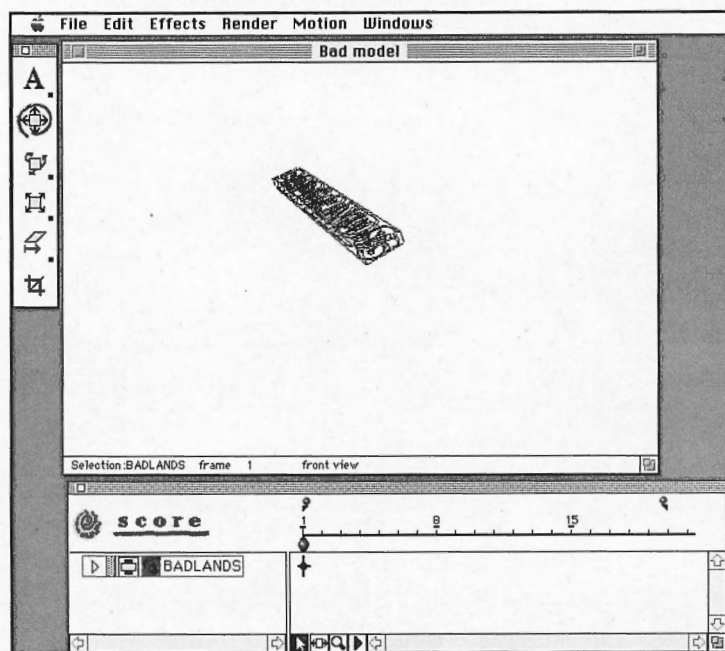
This window contains a timeline for every object in the scene (and each object within a group). Note that the timelines are arranged in an outline format, very similar to Apple's Outline mode in the Finder. You can expand a group, such as the extruded letter "B," shown in figure 9.5, to see the timelines for the individual members of the group—in this case "faces" and "sides."





**Figure 9.5** The Score window

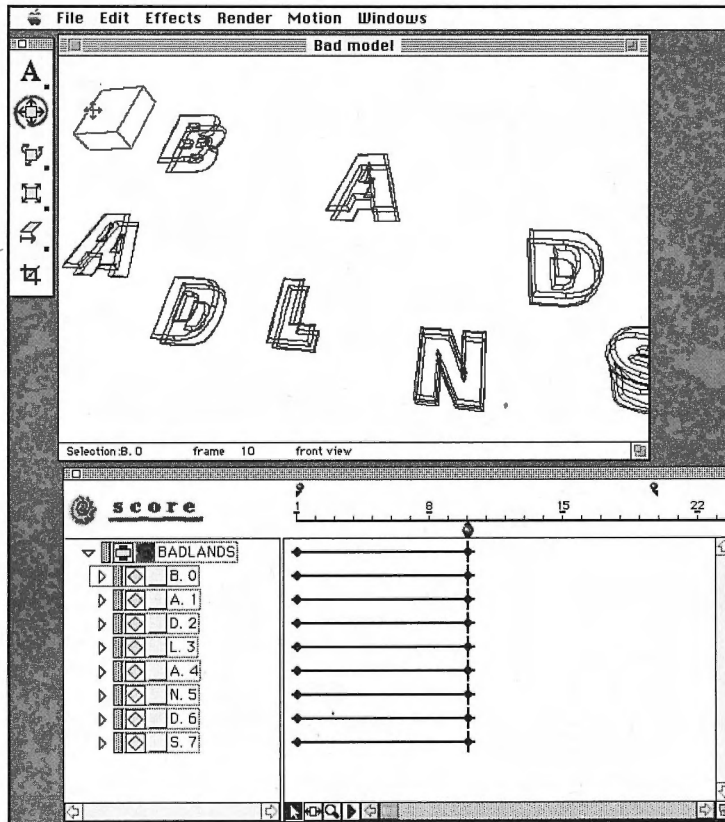
5. Collapse the entire logo in the Score window, set the time marker at 0, and move the logo into the starting position (see figure 9.6). Since this is the position of the logo at time “0,” this is the position the logo will begin in when you run the animation.



**Figure 9.6** Putting the logo in the starting position with the time set at “0”



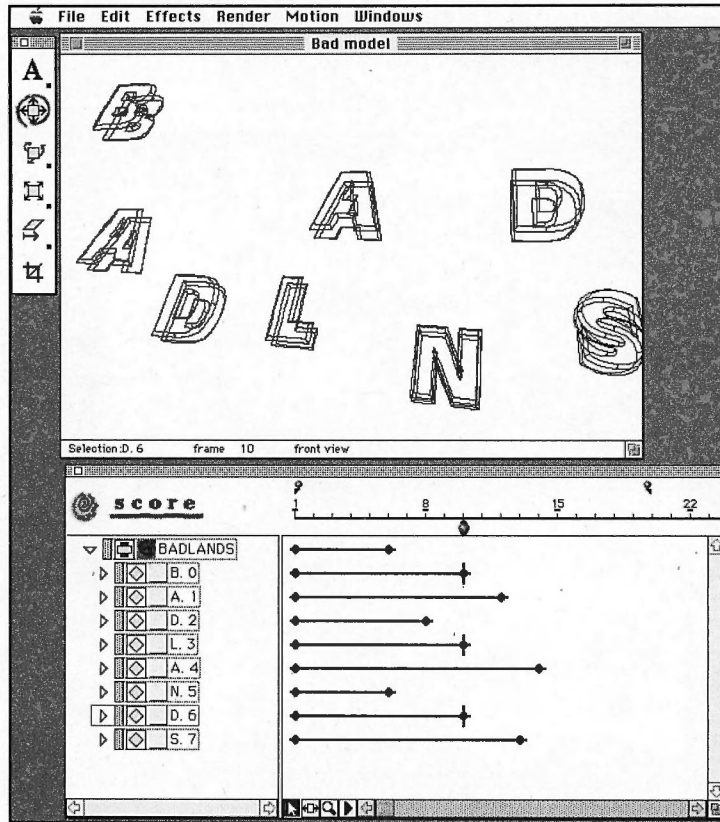
6. Set the time marker to “10” (frames) and move each of the characters to the positions shown in figure 9.7.



**Figure 9.7** *Moving the characters into position at time “10”*

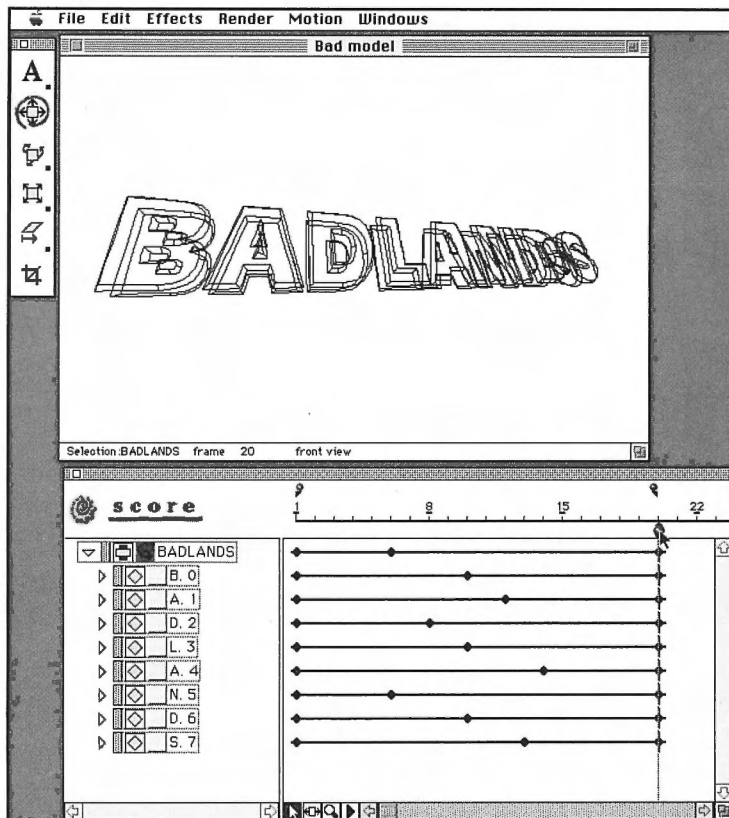
A bounding box (upper left) shows the position of the “B” character as it’s being moved. Note that it is simultaneously highlighted in the Score list of objects to indicate that it’s active. As you move each character a new event mark appears on that character’s timeline under the 10-second mark.

7. Drag the event marks for each of the characters into staggered positions so they arrive at their front position at different times; this will create the sense that they’re flying in somewhat randomly (see figure 9.8).



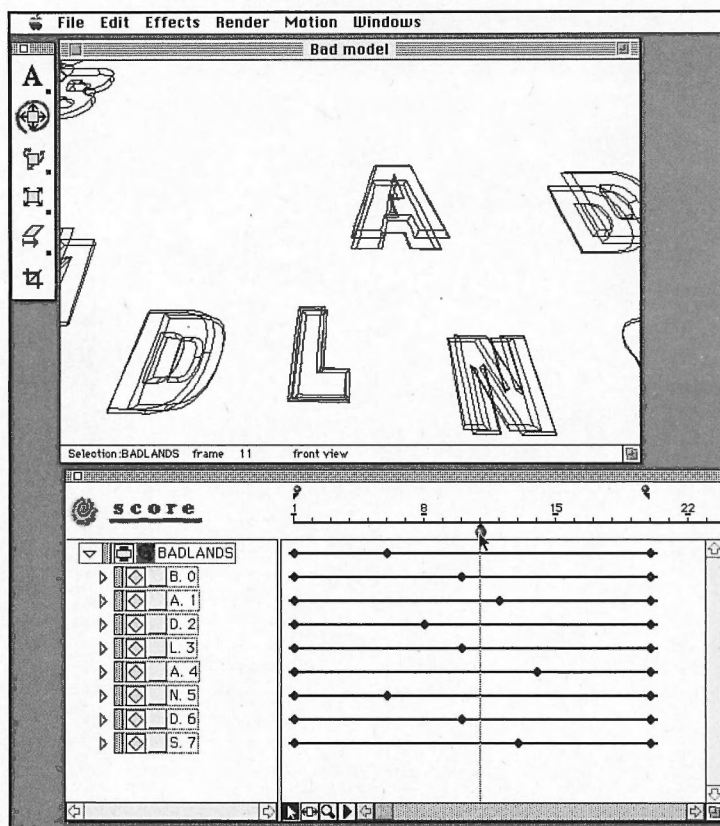
**Figure 9.8** *Stagger the event marks by dragging them*

8. Move the time marker to 20 and one-by-one, option-drag the event marks from position 0 to time 20. This will create copies of the original event marks in the new location. The logo will re-organize itself in its original form. Click the Badlands item in the Score to select the whole logo group, and in the main window, drag it into its final position (see figure 9.9).



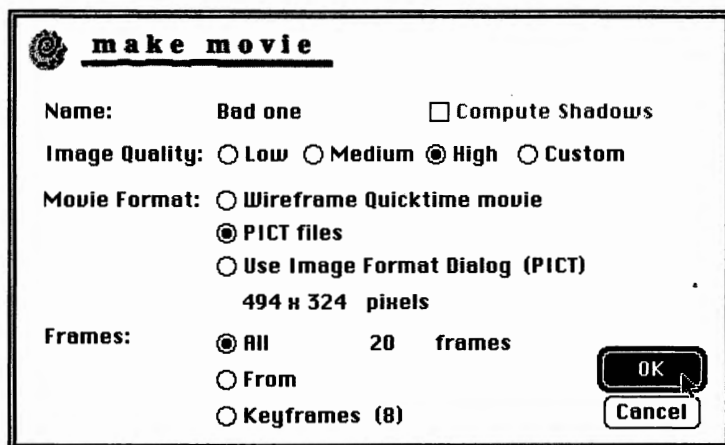
**Figure 9.9** *Reorganizing the logo and dragging it into its final position*

9. You can preview your animation at full speed with the Run Animation menu command, or you can drag the time marker to different points to see how the animation will look in wireframe (see figure 9.10).



**Figure 9.10** Drag the time marker to see the animation at different points in time

10. Use the Make Movie dialog box to set up output parameters and render the movie. Typestry will save the animation as numbered PICT files (see figure 9.11).
11. The final step is to use the Pixar Movie Tool application to convert the numbered PICT files to a QuickTime movie.



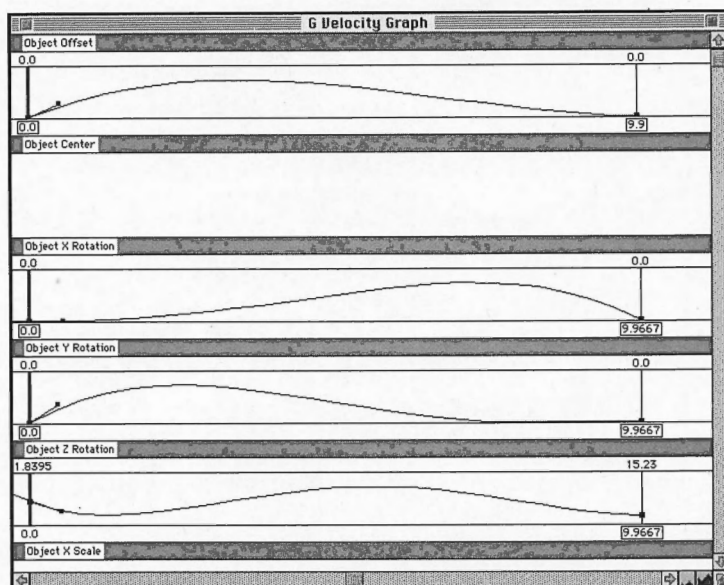
**Figure 9.11** Setting up the rendering output and rendering the animation

## Electric Image Animation System

Electric Image Animation System has very detailed control over the animation parameters for each object. For example, you can change an object's acceleration at a given point. It also has a fine level of control over the animation of visible lights; you even can create animated "lens flares."

Currently, EIAS offers far greater control over every aspect of animation than any other Mac 3-D program. Cameras and lights can point at, follow, and be attached to moving objects. Objects can appear and disappear at key points in time. The program also makes it easy to control the orientation and positioning of parent-child(ren) groups.

EIAS has extremely sophisticated, but easy-to-use, spline-based velocity controls (see figure 9.12). You can change the shape of velocity curves to simply control the speed of an object or camera over time. The program automatically limits changes to fit within parameters that you have already set. Steep sections in a curve indicate acceleration and deceleration; level sections indicate constant change. If the curve is bottomed out, it means that parameter is not changing at that point in time.



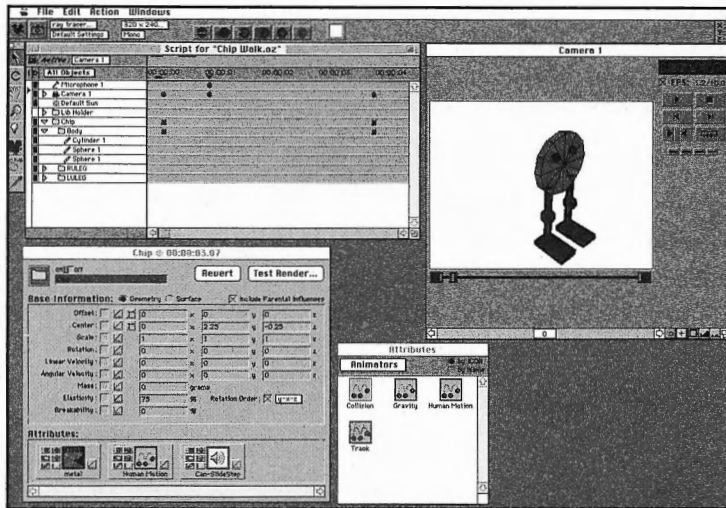
**Figure 9.12** *The Velocity Graph window in EIAS*

Version 2.0 of Electric Image Animation System offers a wide range of cool new animation controls, including:

- **Synch sound animation.** This displays a sound's waveform on the same timeline as your animation, so you can synchronize visual events to events in the sound.
- **Model deformations.** These enable you to twist, taper, bend, shear, ripple, and stretch models. They are great effects for character animations.
- **Particle effects.** Models can shoot sparks and other small particles that are affected by gravity, wind resistance, and other factors.
- **Explosions.** A plug-in to EIAS provides powerful "explosions" that can be used to simulate wind as well as atomic blasts.

### **Presenter Professional 3.0**

VIDI's Presenter Pro is the coolest recent development in animation, particularly if you like to work with RenderMan and sound. The program has a fully event-based animation palette, as well as the capability to minutely modify motion parameters for individual objects. Figure 9.13 shows the main animation elements of Presenter Professional's interface.



**Figure 9.13** The four windows clockwise from upper left: Script, Camera, Attributes, Object Information

The Script (timeline) window is fairly standard as it shows objects in a collapsible outline and each object has its own timeline. There are several types of event markers that indicate whether a model has attributes or “Non-Determined” parameters attached to it. As with other programs’ timelines, you can hide all or part of a model.

The Attributes window will show textures, RenderMan shaders, Sounds, and, in this case, Animators. Animators are “plug-ins” that provide special kinds of motion to models. Because some plug-ins, such as collision detection, are very calculation intensive, the motion they provide is “Non-Determined” until you tell the program to recalculate the motions (at which point, they’re “determined”). While VIDI plans, eventually, to put Inverse Kinematics into Presenter Pro, motion simulators, for now, are the next best thing.

You apply an attribute to a model by double clicking it to open its Object Info box (lower-left) and dragging the attribute from the Attributes window into the model’s Attributes box. In figure 9.13, the Chip model has a Metal RenderMan shader applied to it, as well as a Human Motion Animator, and a Can-SlideStep sound. Note that you can set many of the object’s animation parameters in this dialog box. By double-clicking an attribute, such as a shader, you can customize its parameters.

## Camera Moves

Just as you can animate models, you can also animate a camera so that any point of view is possible. This has an added advantage because 3-D cameras are not constrained by the limitations of film speed, changing exposure due to zooming, or depth of field.

It is possible to mimic the wildest actions of a cinematographer's movie camera: panning, zooming, and dollying side-to-side or front-to-back. It also is possible to have a camera float above a scene, follow an actor, or remain attached to an actor.

Keep in mind that perspective is your friend. Objects flying into and out of the screen, as opposed to just side-to-side, tend to create a better impression of 3-D movement.

## Flying Camera Logo

The following animation was created in Electric Image's Animation System. While the camera flies along to view different parts of the scene, objects within the scene are independently animated with different start and end points.

1. **0 seconds.** After you generate a logo model in ELIAS' Mr. Font and place it into the scene, position the camera for a top view so that it is aiming at the left end of the logo, but positioned far to the right. In the Camera Info dialog box, rotate the camera 90 degrees so that the logo forms the floor of the camera view (see figure 9.14). Rotate each of the letters 180 degrees around its own y-axis.
2. **3 seconds.** In 3/4-second intervals, the letters of the logo are revolved to their proper orientation. This gives the effect of the letters flipping into position one at a time. Move the camera so that it is nearly touching the focus point at the left end of the logo (see figure 9.15).
3. **3.5 seconds.** Move the camera away from the logo; rotate it back to its normal upright orientation; and aim it at the center of the logo. This results in the logo swinging into proper orientation and fitting completely into the camera window (see figure 9.16).



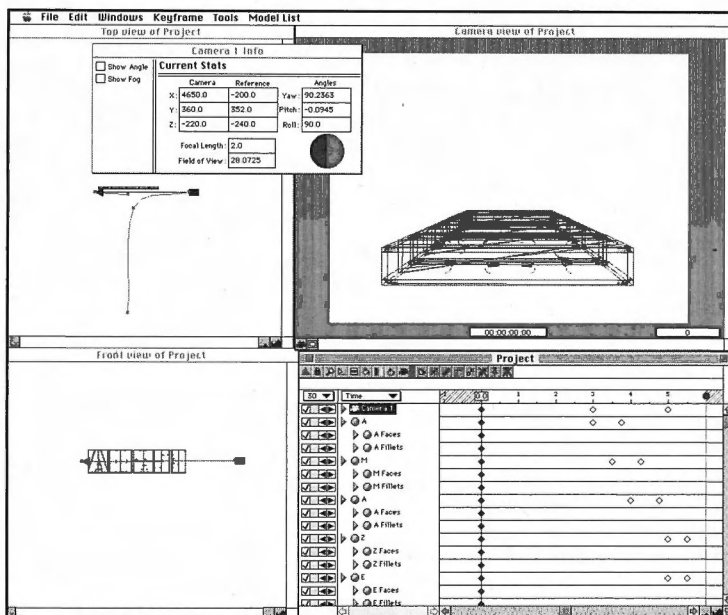


Figure 9.14 Positioning the camera in the Camera Info dialog box

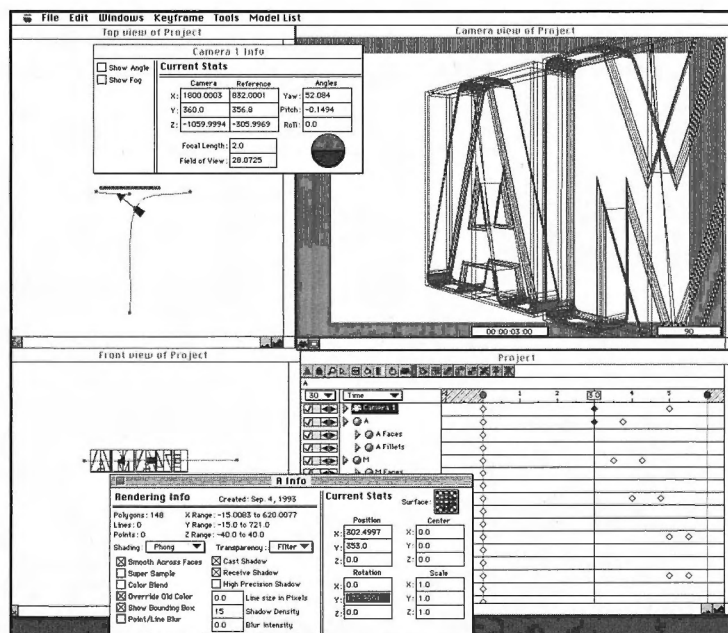
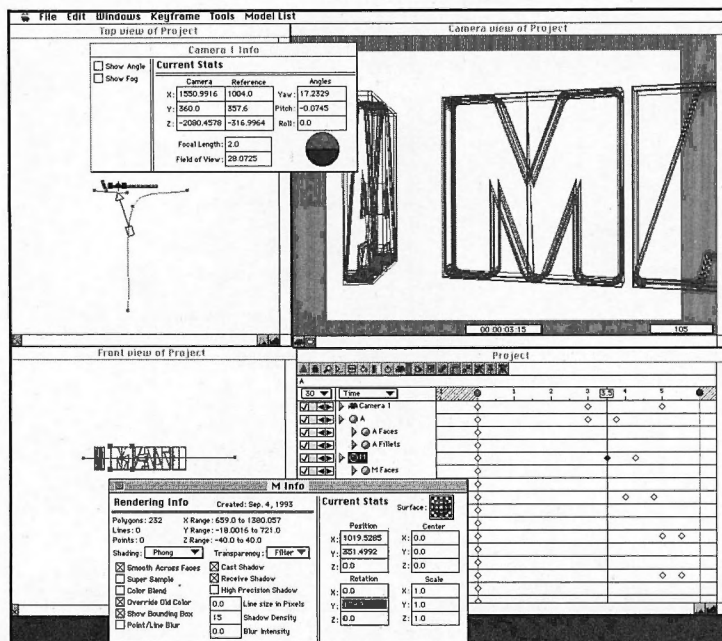
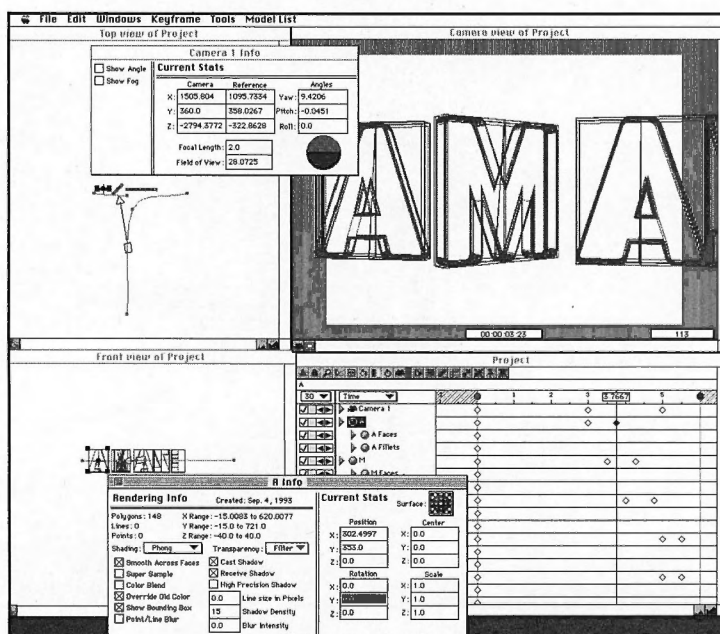


Figure 9.15 Moving the camera position again



**Figure 9.16** Moving the camera to position the entire logo in the camera window



**Figure 9.17** The final letters move into place

4. **5.5 seconds.** The final two letters of the logo settle into correct position, freezing the logo (see figure 9.17).
5. **6 seconds.** The end.

#### Tip

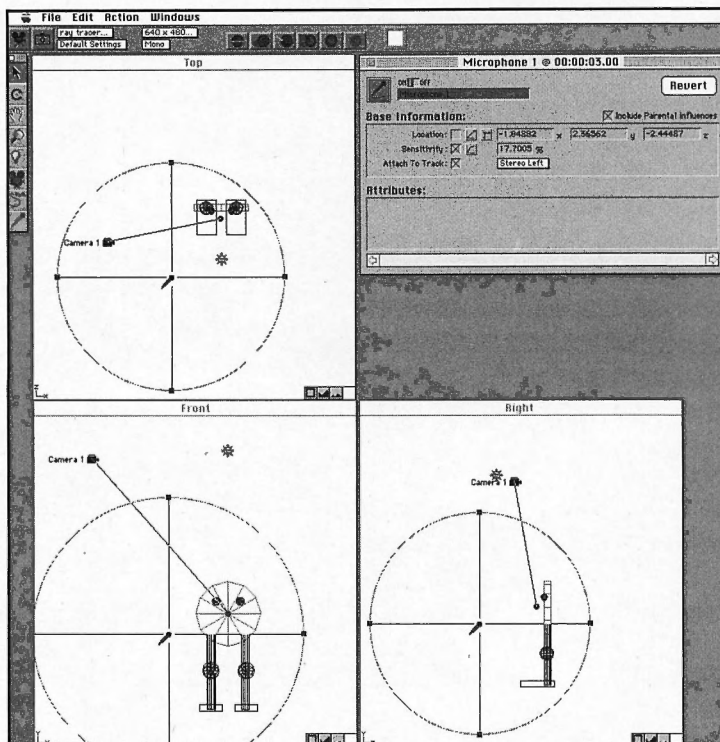
Often when building animations you will find that you want to aim a camera near an object, but that you want the object to appear near the top right of the frame, not in the center. At other times, you will want to create an animation cycle and have it repeat multiple times as you move the whole group around in the scene. The solution to both problems is simple: create an invisible object (a cube is good and small), and make it the parent of the object you want to aim at, or the cycling group you want to move. Since the object has parent status, anything it does will be reflected in its child(ren). If you aim a camera, the children will maintain their proximity to the “ghost,” but remain outside of the screen’s center. If you attach a group to the ghost object (even if this group is going through motion cycles) you can move the ghost and everything else will move along with it.

## Sound

Animation and sound go together like Road Runner and “beep-beep.” But getting your animations synched in tune with your sounds isn’t an easy matter. Several new developments help a great deal. The first is QuickTime, which means you now have an easy way to store sounds with movies. The second is really a number of different software solutions to help you get your sounds together with your pictures (these programs are described in more detail below).

### Presenter Pro’s Microphones

Presenter Professional 3.0 is the first program on the Mac to offer true 3-D sound. It’s a cool enough idea that I’m certain it will pop up in many other programs. Presenter uses microphones that you can place as objects in your 3-D world. They can have a range and sensitivity, so objects that are far from the microphone will sound faint; objects that are closer will be loud (see figure 9.18).



**Figure 9.18** Microphones in Presenter Pro pick up sounds as they enter the microphone's range (shown as a circle around the mic)

Sounds themselves are attached to objects, and as an object zooms towards a microphone, the microphone will automatically register the proper “Doppler effect.” This is what causes the sound of a car rushing towards you, then past, to start low in pitch and rise until it passes, then fall off into the distance—neeeEEE-OUWWwww!

As VIDI's implementation of microphones matures, you'll be able to apply Attributes to microphones, such as sound effects (if you want an echoed cathedral sound, for example).

## Synching to Sound

Serious character animators usually begin with a pre-recorded soundtrack. They then animate their characters to synchronize with specific events (such as dialog). In order to synch sound when working in 3-D, you will need to generate an animation script that lists events by time. The standard measure of time in video

and animation is called SMPTE Time Code, and it expresses video in terms of hours, minutes, seconds, and frames. Armed with this script, you can set key frames for each event on your 3-D software's timeline. (These may be as tedious as specifying "lips open," "lips closed," "eyebrows up," or "eyebrows down.") After you generate your animation, you can combine the sounds and video using standard compositing techniques.

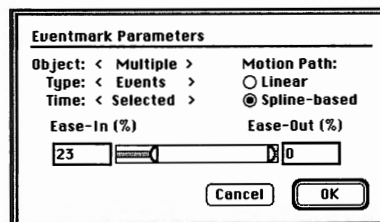
The alternative is to generate an animation and then dub in a sound track, editing parts of the animation to fit the audio. In the long run, this is less efficient, but gives the editor more material with which to work.

The practical compromise is to create the animation to fit an existing sound track, then mix in sound effects afterwards.

## Ease-in and Ease-out

Good animation accounts for the natural acceleration, deceleration, orientation, and distortion of objects as they stop and go and move around. For animators, simulating these changes is an important part of realism because in real life things do not go from "0 to 60" in zero seconds.

Infini-D, for example, uses a slider control to determine the ease-in and ease-out value of an event (see figure 9.19). To access the slider, you double-click on an event mark (object key frame). You can specify an amount you want the object to ease into its next motion, as well as how long you want it to ease out. These capabilities also are offered by LogoMotion, Macromedia Extreme 3D, Presenter Pro 3.0, StrataStudio Pro, and ELIAS.



**Figure 9.19** *Infini-D's basic motion smoothing control*

## Smoothing Motion

One of the problems of key frame animation is that you cannot see how objects will move in real time. Often you will end up with animations where the camera or objects jerk in one direction or another, or turn unnaturally. Like acceleration, it's important

to be able to adjust the smoothness of motion during directional changes, so objects conform to expectations of the real world.

Having an object maintain proper orientation as it flies is tricky. Cameras can generally be set to look at another object as they fly about. This means that if a camera is flying on a narrow oval track around a central subject, it will continue to look at the subject and turn automatically to stay aligned in the scene. With objects that usually cannot look at anything, it is a bit trickier. Again, Infini-D comes to the rescue with animation assistants that keep an object aligned along its path of motion. StrataStudio Pro enables you to easily align an object to its path of motion and make it bank like an airplane when it turns.

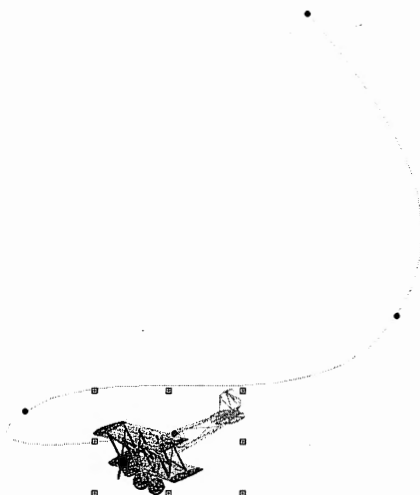
### StrataStudio Bank Turns

1. At time 0, the airplane is at rest (see figure 9.20).



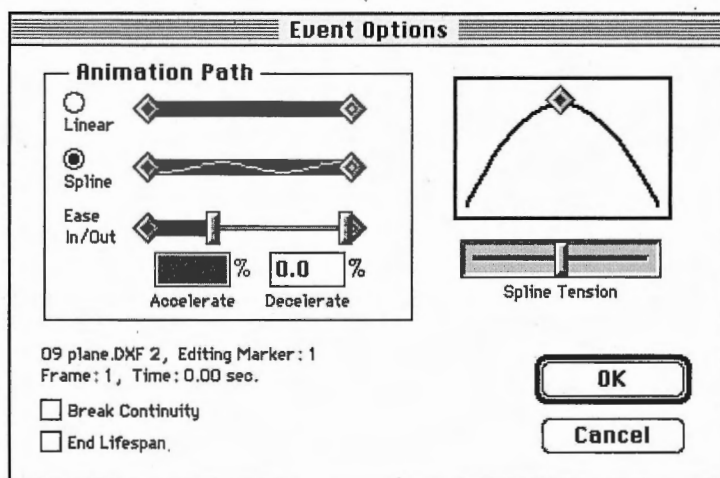
**Figure 9.20** *View of airplane at rest: time is 0 at frame 1*

2. Define a motion path with four event marks (see figure 9.21).



**Figure 9.21** *Define the motion path of the airplane by dragging the time slider forward, then moving the plane*

3. Double-click on the first event mark to open the event options dialog box. Set the ease-in value to approximately 33 percent. This determines the rate of acceleration. Notice that the spline tension slider controls the smoothness of an object's motion through a corner defined by an event mark.
4. The End Lifespan checkbox in the lower-left corner is used to snuff a model out of existence at a key point without interrupting its motion—an airplane could disappear in mid-flight, for example (see figure 9.22).



**Figure 9.22** *Setting the ease-in value for the first event mark*

5. Select the entire motion path for the airplane by dragging a marquis around all of the plane's event marks in the Sequencer. In the Animation menu, choose the Align to Path plug-in.
6. You can adjust the orientation of the model in the fast-shaded preview at the left by dragging it around with the mouse (see figure 9.23). This establishes the normal forward and upward directions for the object. Clicking the Bank Turns checkbox causes the plane to bank naturally, as well as to follow the path.

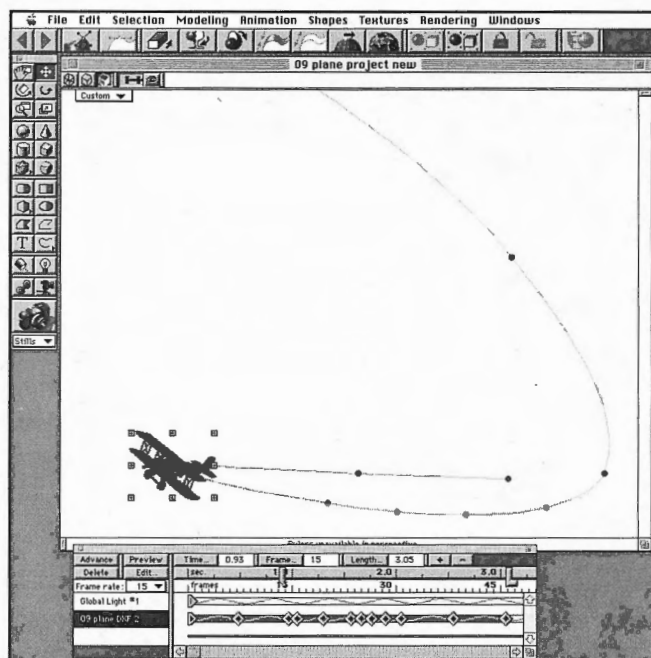




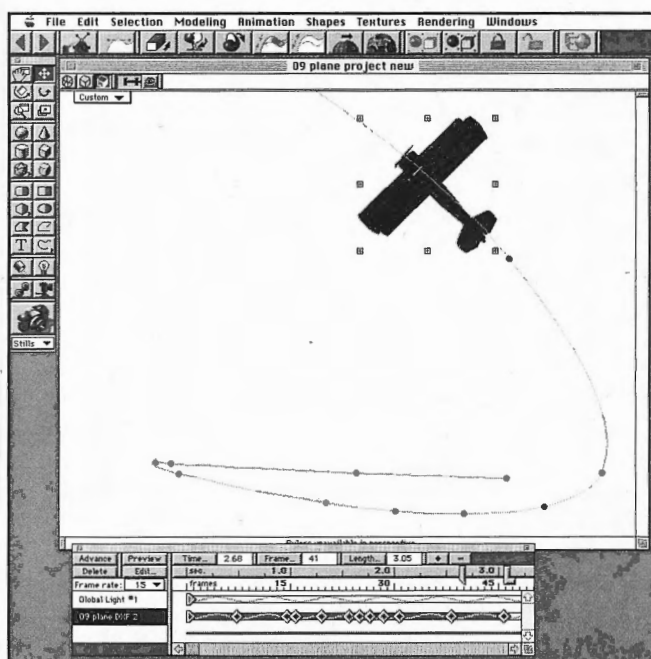
**Figure 9.23** Use the *Align to Path* dialog box to establish an “up” and “forward” direction for the plane; the *Bank Turns* checkbox ensures that the plane banks as it turns

7. Studio Pro automatically fills in the events needed to keep the plane aligned to the path and banking naturally through turns (see figure 9.24).
8. Note that the plane now “leans into” its turns as it’s animated; it also follows the animation path nose first from beginning to end (see figure 9.25).





**Figure 9.24** The plane's event sequence in the timeline is now set to make it fly naturally



**Figure 9.25** The plane leans into its turns

## Manually Smoothing Motion

With programs that do not automate banked turns, you will find yourself creating subordinate key frames to make models rotate and align within the larger overall motion. For example, if a car is flying around a narrow oval track, it will rotate almost 180 degrees at each end of the oval, but rotation will be minimal along the long straight aways of the oval. In contrast, for an object flying around a circle, you can simply specify that it should rotate 360 degrees in the time it takes to go around the loop. An object flying around a square track or other corners takes careful handling. Here, you will want the object to rotate only as it comes to a corner. You will therefore need to create two key points for every corner. The object will begin to rotate just as it approaches the corner, and finish a 90-degree rotation just after it rounds the corner. (For the effect of a fishtailing car, you can make it over-bank at the second key point; then add a third key point just beyond; and have it rotate back to 90 degrees.) The important point is not to make the car turn constantly to complete a 360-degree rotation as it rounds the square, but to adjust the rotation so it matches the actions of a real car on a genuine square track.

## Prepare to Morph

*Morph* is the popular term for metamorphosis, or transformation from one shape to another. (This is a field that is, well... rapidly changing shape.) Unlike 2-D morphing, which relies on flowing one bitmapped image into another, 3-D morphing actually distorts a model until it assumes a new shape. Almost every 3-D company has a new and improved morphing tool in the works. A couple of Mac 3-D programs have special capabilities for animating unusual properties.

You might use morphing, for example, when you want that drab-old detergent bottle to transform into a new-and-improved Super Suds! container.

Infini-D enables you to morph any object created in a Workshop into a different form of itself (see "Morphing a Glass" later in this chapter).

Crystal TOPAS, like Macromedia Extreme 3D, enables you to morph between two spline objects with the same number of spline points. Even simple key-frame animators, such as Typestry, StrataVision 3d, and Presenter Professional, enable you to squash and stretch objects over time.

StrataStudio Pro enables you to morph smoothly between spline objects with different numbers of control points. The program has also made an art out of shape morphing. Extensions to the program enable you to blow things apart and fly them back together in a new form, or simply morph smoothly between dissimilar objects without the visual blending that occurs in most 2-D morphing programs.

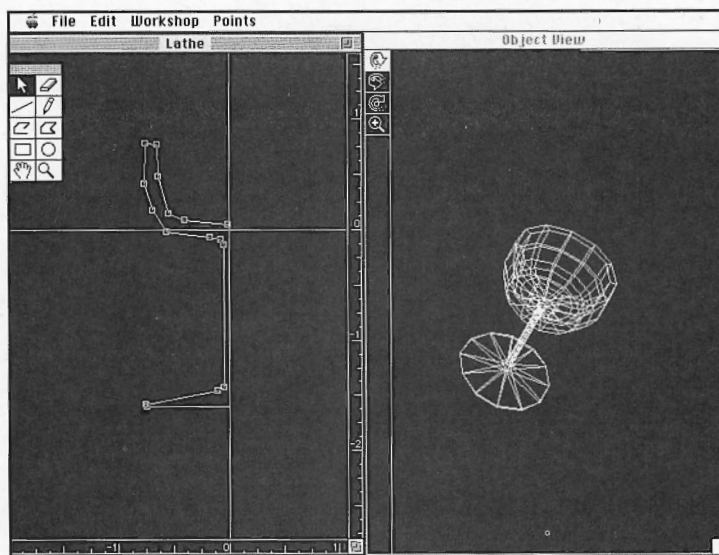
In some programs you may need to conform to some rules to accomplish smooth morphs. In Macromedia Three-D, for example, you must begin with two models created with the same number of control points. About the only practical way to do this is to build your end model by deforming your starting model (without adding points). MacroModel forms a good partnership with Macromedia Three-D in this respect, because it is fairly easy to use the Bezier spline-based modeler to derive vastly different shapes from the same starting block. Essentially, Macromedia Three-D tweens the positions of each point in the model at each frame in the sequence, using the beginning and ending states as the extremes.

Similarly, Sculpt 4D limits morphing to different forms of the same model. Infini-D's morphing and modeling are independent, so no preparation is necessary.

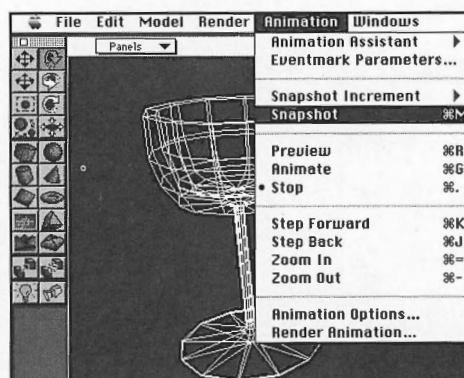
## **Morphing a Glass in Infini-D**

Infini-D's morphing is very simple to use. You can morph between any objects that can be created in the same Workshop window (Lathe, Extrude, or Freeform). But with very complex surfaces (such as 3-D type), keep in mind that surfaces will look strange when they turn inside out to accomplish a morph.

1. Model a glass in the Lathe Workshop by drawing its outline with the Polygon tool (see figure 9.26). When it is complete, choose Exit Workshop.
2. Position the model, assign a texture and lights, and select Snapshot (see figure 9.27). This freezes the model at the current location on the timeline.

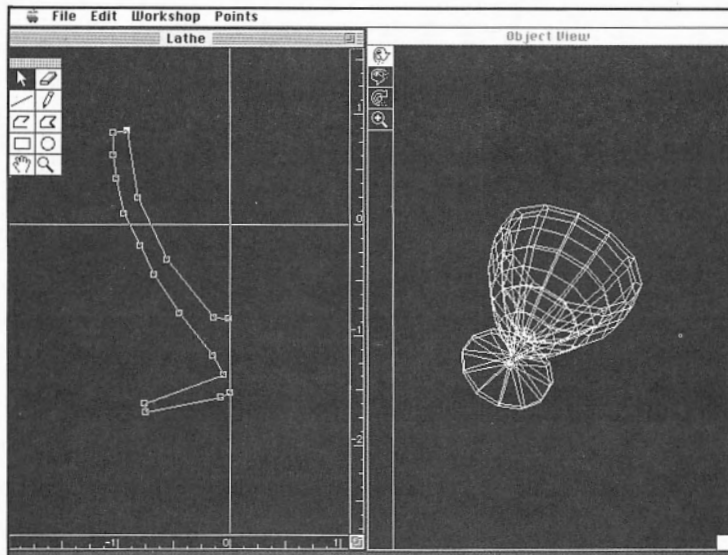


**Figure 9.26** Drawing a glass outline in the Lathe Workshop to create a glass object



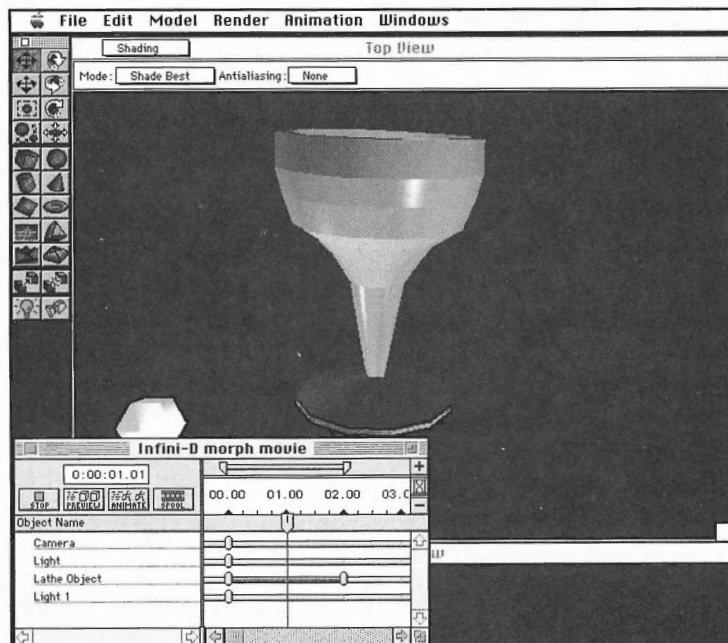
**Figure 9.27** Choosing Snapshot from the Animation menu

3. Move the timeline marker a few seconds, then move the model to a new location and rotate it. Double-click on the model to open it in the Lathe Workshop once again. Select the Reshape Outline command, and drag the control points around to reshape the glass (see figure 9.28). Exit the Workshop and note that the model has been changed to reflect your edits. Choose Snapshot again to complete the morph.



**Figure 9.28** Reshaping and rotating the glass

4. To preview the morph, drag the time slider to intermediate points in your animation (see figure 9.29).



**Figure 9.29** The completed morph

## Rotoscoping

Rotoscoping is the technique that enables you to map an animation or digitized video onto a background, or onto an object in your scene. Several 3-D programs enable you to use animated backgrounds and textures. These begin as animation files—although there is no accepted standard. Macromedia Three-D uses PICS; Infini-D uses QuickTime; Presenter Professional uses QuickTime; and EIAS requires an EIAS Image animation. Otherwise, these programs use the animation file in the same way you normally use a still image as a texture or background. For every frame in the animation, the program steps one frame through the texture movie, resulting in an animation within an animation—even an animation wrapped around a 3-D object.

You may use an animated texture to put a movie on a cinema screen, to put video on a TV screen, or even to show clouds blowing past a window in an architectural rendering.

### Tip

Several 3-D programs offer rotoscoping, including:

- Animation Master
- EIAS
- Infini-D
- Macromedia Extreme 3D
- Macromedia Three-D
- StrataStudio Pro
- StrataVision 3D

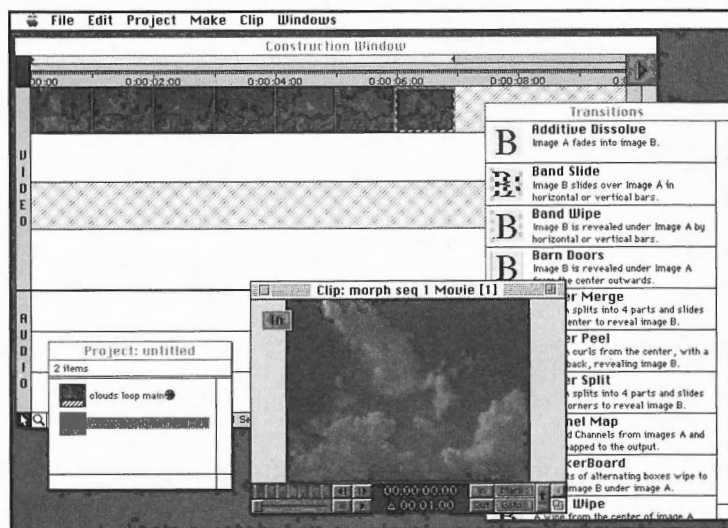
## Film Loops

A seamless animation or *loop* is a clip that loops continuously so that the same series of frames plays over and over; it is impossible to determine where it begins and ends. Just as seamless textures are more convincing, loops can create the

sense of ongoing activity in the background or on the surface of an object. To create animated loops, you can render them in 3-D, as well as turning to some digital trickery.

It is usually simple to create a 3-D animation that begins and ends in the same place—thus guaranteeing that the action in the animation continues smoothly without a break. For example, you can create an animation of machine gears grinding together and render it as an animation for use as a background. If you later composite this with an animation of a wrench dropping into the scene, you can create the effect of a wrench dropping into the machine.

For abstract moving animations, you can create an animation that moves in one direction—then reverse it and splice the two segments together. Using Kai's Power Tools' Texture Explorer, for example, you can generate two background images with a smooth transition between them. Then you can save this effect as a QuickTime movie; reverse the order of the images (but not the transition) using Premiere's Backwards filter; and generate a second QuickTime movie with the two segments spliced together. When you play the animation, the beginning frame is the same as the end and the entire movie loops seamlessly.



**Figure 9.30** Adobe's Premiere is the best of the editing tools when working seriously with QuickTime

## Morphing for Loops

It is possible to create very convincing seamless loops of natural video scenes by using one of the several new special effects programs that provide digital morphing.

Using a program like Gryphon Software's Morph, you can morph the end frame of a QuickTime video segment into the first frame. This creates the missing link between the first and last frames of a clip. With patience, this can be an effective technique, although it will probably take several attempts before the transition seems natural.

## Morphing Tools

Several programs now provide digital morphing that can be used to create special effects film loops for rotoscoping. Elastic Reality from ASDG Software and VideoFusion from Video Fusion are both good candidates. *Elastic* reality enables you to morph parts of images at different rates (event-based morphing) for effects that go beyond simple transitions. VideoFusion, meanwhile, is a full-featured QuickTime effects tool that can be used for compositing and other QuickTime production effects. The Valis Group's MetaFlo combines event-based 2-D deformation with morphing, so you can create very fluid, liquid-like morphs from one picture to another.

## Particle Animation

*Particle animation* enables you to decompose a model into points: each with a characteristic shape (such as polygonal triangles or spheres). These particles can then be blown apart, crumpled by gravity, or blown away by wind. Particle animation enables you to create effects such as a snowstorm or exploding fireworks.

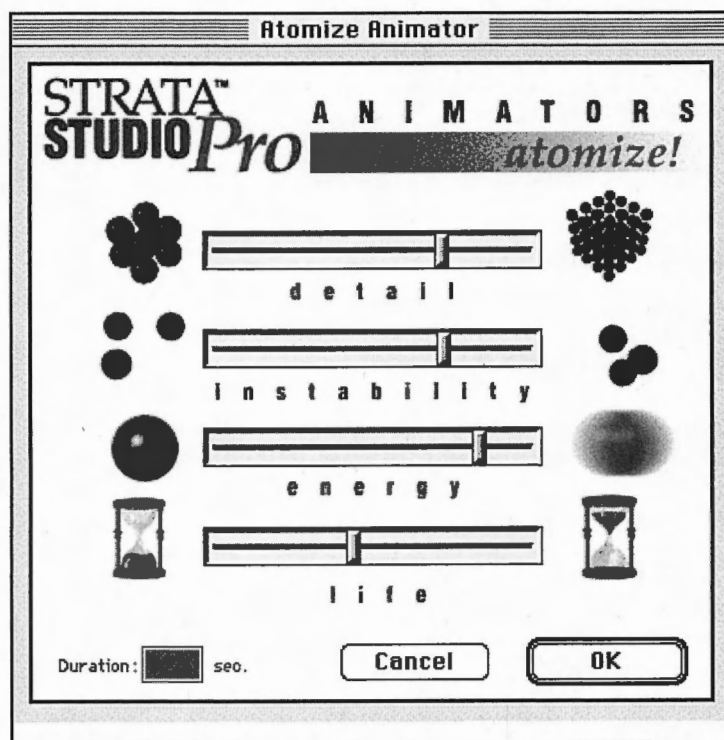
Three Macintosh 3-D programs currently offer particle animation: Electric Image Animation System, StrataStudio Pro, and Typestry.

StrataStudio Pro was the first commercially available Mac 3-D particle animator. It offers a variety of particle animation plug-ins that explode, shatter, and vaporize models. Particle morphing enables you to blow apart a model and have

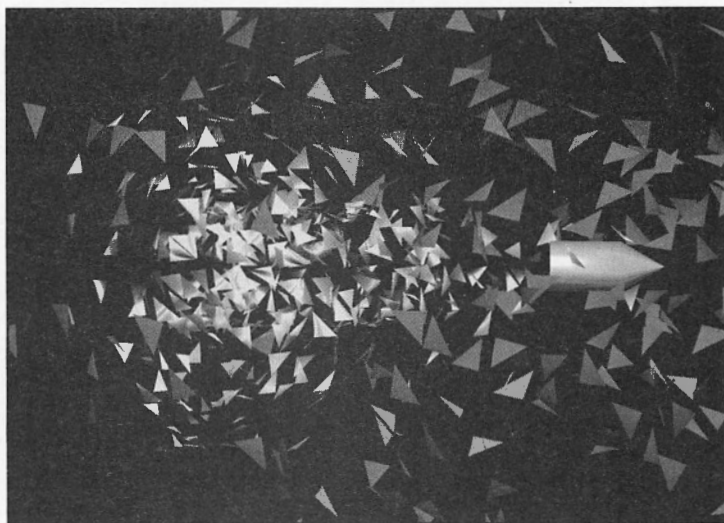


it reorganize the particles into a new model. One particle plug-in, Atomize, simply disintegrates a model into a mass of rapidly vibrating spheres before it vanishes from the scene (see figure 9.31). Two particle plug-ins, Explode and Shatter, were used to create the bullet-blasted sphere (actually four spheres using variations of the two plug-ins) in figure 9.32.

Electric Image's Animation System 2.0 ships with an "explosion" extension called Mr. Nitro, as well as a particle generator plug-in for creating sparks, fountains, and other cool particle effects.



**Figure 9.31** The Atomize plug-in for Strata StudioPro dramatically disintegrates a model



**Figure 9.32** A bullet-riddled sphere explodes and shatters with the help of two Strata plug-in particle effects

## Frame Rate and Resolution

One of the concerns for any animator is producing animations at the right screen resolution and frame rate for the task at hand. The *frame rate* of an animation is the number of frames that flash onto the screen every second to create the illusion of smooth motion. Film, for example, uses 24 frames per second; video uses 30 frames per second; and a common rate for CD-ROM-based QuickTime animations designed for playback on lower-end Macs is 15 frames per second—though 20 fps is becoming the norm. Thus, a ten-second video animation contains 300 frames, while a ten-second QuickTime clip may have only 150 or 200 frames.

The resolution of an animation determines the size (in horizontal and vertical pixels) of the screen it occupies. The American television standard (as well as the size of a Macintosh 14-inch monitor) is 640 pixels wide by 480 pixels high. This is constant—no matter how large or small the television (larger TVs just have bigger pixels). For CD-ROM QuickTime delivery, 320 by 240 pixels became manageable for slower Macs and CD-ROM drives with the introduction of QuickTime 2.0. Power Macs running QuickTime 2.0 can achieve 30 fps of full-screen video with high levels of compression. There are many other standard resolutions which depend on your intended output. Although film-resolution and upcoming HDTV animation is still primarily created on high-end dedicated systems, some Macintosh 3-D programs can render resolutions as high as 8,000 by 8,000 pixels.

## Uses of 3-D Animation

There is a booming demand for 3-D animation, but its uses cannot be summarized easily. Video and multimedia are two of the areas where you'll find scads of 3-D. There are many others, including: video games, TV commercials, presentations, industrial design, and architectural rendering and walkthroughs. Different types of users will have different demands from 3-D applications.

### Video

Video animators are interested in creating animations of 640 by 480 (NTSC) resolution or better at 30 frames per second. Users in the video field generally have a large investment in equipment and demand high-quality output.

Field rendering and legalized colors are some of the concerns of the video producer. Field rendering actually creates two frames for every frame of video. One is an even field; the other is odd. When these frames are interlaced, they play back as an animation that is effectively 60 frames per second. (If you hold your hands out, palms facing you, spread your fingers and lace them together, you'll get a good approximation of NTSC "interlacing".)

Because the color gamut (range of colors supported) of NTSC video is extremely limited compared to that which can be created and displayed on a 24-bit Macintosh system, many colors will not display properly when played back on an NTSC monitor (TV). This means you must settle for soft, muted colors if you plan to output your productions to videotape. Bright reds, for example, are notorious for *bleeding* when transferred to video; this means the large patches of red tend to blur out of their boundaries into other parts of the screen. Video producers will probably need a video encoder attached to their video board (or a separate NuBUS version) to preview 3-D renderings on an NTSC monitor to make sure colors are NTSC compatible.

Once animations are finished, you'll need a way to get them to videotape. This means either rendering to a video standard format, as Electric Image can do, or converting files with a program like DeBabelizer or Missing Link. (On the other hand, you can undertake the significant expense and complication of setting up a direct-to-disk or direct-to-tape recording system, although most video producers these days recommend Exabyte to Abekas as the best way to transfer onto tape.)

## Multimedia

Multimedia authors may create animations for video use in high-profile productions and presentations, or they may settle for smaller frames or lower frame rates, depending on the intended use. QuickTime, the standard format for delivering digital animations on the Macintosh and one of the standards in use for Intel-based PCs, has evolved into a system that can pump broadcast-quality video onto the Macintosh screen or out onto videotape. New hardware (such as Radius Inc.'s VideoVision) is bringing this capability very close to the Macintosh mainstream, although prices for the required hardware remain fairly high (for one, digital video requires multiple *Gigabytes* of super-fast disk arrays).

While full-screen video pushes the envelope of personal computer power, the medium of CD-ROM is an excellent delivery tool for smaller-sized video played at reduced frame rates. CD-ROMs are the current delivery tool of choice for commercial multimedia.

On the other hand, full-motion, full-frame video compression systems are dropping quickly in price. As this was being written, Apple had just introduced the first of a line of low-cost Macs for which you could buy a very inexpensive MPEG video board. This format may one day become the host for all the interactive television that the media has hyped recently. But more immediately, MPEG means you can put full-motion, full-screen video on a CD-ROM; 3-D animators can take advantage of this right now.

The introduction of MPEG to the Mac also means that full-screen interactive 3-D animations in kiosks and other unusual locations should no longer be prohibitively expensive to produce.

3-D animation has a long and happy future in multimedia. As the on-board support for 3-D graphics expands within system software (see chapter 13, "Advanced Topics"), concepts like real-time interactive 3-D graphics will almost certainly become a reality.

## Film

While 3-D on the Macintosh has the capability to produce renderings suitable for film, the high resolution and rapid turnaround times required in film production mean the Mac is generally too slow to produce important segments for major motion pictures. Film demands only the latest-and-greatest effects, which are still

mostly created on SGI workstations. However, Macintosh tools, such as Electric Image's Animation System, are closer than ever to crossing this barrier. As Power Mac versions of the most powerful 3-D packages emerge, it's inevitable that Mac graphics will find their way onto film.

## Virtual Reality

Three-D software has come along so quickly that virtual reality is virtually, well... reality. While the Mac has yet to fool people into thinking they are walking through another world, a number of applications now allow for real-time visualization of 3-D scenes. Virtus has a possey of 3-D walkthrough products; Strata offers a program that enables you to render your 3-D creations from Studio Pro and StrataVision 3d in real time as you navigate through them; and Macromedia is rumored to be working on a program that will enable you to create your own virtual-reality games.

The ultimate goal of 3-D graphics is to be able to do everything so effortlessly and smoothly that the software disappears and all you have in front of you is a virtual world where you can build models by waving your arms and propel them through space with the force of your will.



A Demo version of Virtus WalkThrough, and numerous sample files are on the CD-ROM. To get a feel for what VR is about, you really have to try these programs. To use WalkThrough: copy it to your hard disk, open it, and load a walkthrough file. Then simply start dragging the mouse around to see what happens. You also can experiment with the Option and Command keys to see how these affect navigation.

While virtual reality isn't quite up to the vision of science fiction writers, available programs for the Mac that lay claim to the "VR" moniker are quite useful in their own right.

Virtus WalkThrough and WalkThrough Pro, Virtus vr, and Strata Virtual are all aimed at users who need to be able to build 3-D scenes, and then get a realistic look at them in real time while they navigate around. The difference between this approach and simple animation is that you can navigate around in these virtual spaces without first rendering a movie and without being committed to a one-way animated track.

Virtus WalkThrough and WalkThrough Pro are the most elaborate of these tools. They offer full fledged modeling and scene building tools, as well as real-time navigation interfaces. WalkThrough Pro differs particularly in its capability to use texture-mapped objects in its scenes. This makes walkthroughs much more realistic than previously possible, and the PowerMac version greatly extends this feature.

The potential of these programs goes way beyond what's available today. When Apple finally introduces real-time 3-D rendering support to the Mac, virtual reality may offer real-time fly throughs of highly realistic models, complete with texture maps and rendering effects such as shadows and transparency.

And by the time this book is gathering dust on your shelf, virtual reality will undergo a transformation. With 3-D as part of the Mac's operating system, it will be possible to create multimedia authoring tools that include 3-D navigation and object selection. Imagine an interactive multimedia interface that enables you to wander around rooms and pick up and handle objects that will act as multimedia "buttons." Virtual reality at some point will certainly merge into multimedia so that the two are inseparable. I may be going out on a limb with these predictions, but the groundwork for this to happen is already under foot. Whether or not the current applications, such as WalkThrough Pro, will follow this vision is speculation, but it seems like the obvious path for virtual reality to follow.

So, back to reality: what can you do with VR today? Movie directors use it to visualize action and camera movements; architects show clients the layout of building designs in real time; and interior designers show clients where the new furniture is going to go. Another aspect of virtual reality software is that it's a lot of fun to use. It's one thing to build a cool model and render it. It's another to have it instantly transformed into a video game.

**c h a p t e r**

**10**

**Rendering**

**R**endering is where modeling, scene building, and animating payoff with the creation of realistic 3-D images. While rendering is concerned with the interaction of light and surfaces, the result of rendering is ultimately an image composed of colored pixels forming a bitmapped image. (The exception to this is 3-D PostScript.) The rendering process diverges in one crucial respect from the way lighting and visualization of scenes occurs in the real world. In nature, light travels from a light source to an object that either transmits or reflects changed light to the viewer. What you see is reflected light, modified by interaction with objects and atmosphere.

The problem with this natural approach, as far as the 3-D user is concerned, is that most of the light that enters a scene is never reflected back to the viewer, but bounces out into space in some other direction. Three-D software, therefore, calculates the process in reverse: given the viewer's position and focus, the renderer determines what surface is visible, calculates the direction and source of all the light striking that surface point, and determines how it reflects off the surface. Finally, the result of these calculations is that the renderer determines a color for the particular pixel in question.

A typical video-resolution image is 640 by 480 pixels (about 300,000 pixels), and the renderer steps through this process for each and every one of them—sometimes several times! No wonder rendering takes a long time.

Rendering remains the real bottleneck in Macintosh-based 3-D systems, but that limitation is fading quickly. Programs like Strata's Virtual 3-D and Virtus' WalkThrough Pro are proving that real-time, texture-mapped 3-D animation is a real possibility. Power Macs, as I explained in the first chapter, have turned the world on its three-dimensional ear.

While this chapter covers the basics of rendering, there are a number of advanced software and hardware options for greatly accelerating the speed and potential power of Macintosh 3-D rendering covered in chapter 13, "Advanced Topics."

## Rendering Methods

There are many types of rendering, and each has its advantages. For the Mac 3-D user, these include:

- PostScript blending
- Wireframe
- Hidden line



- Flat shading
- Gouraud shading
- Phong shading
- Ray tracing
- Radiosity
- Ray Painting
- RenderMan



The tutorial on the CD-ROM contains color examples of these rendering methods so you can get a better look at their respective advantages.

## **PostScript Blending**

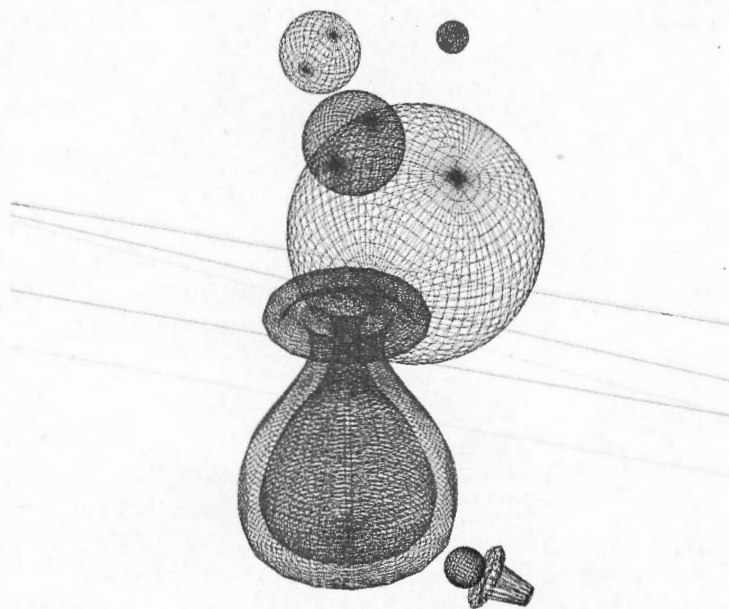
When you render with Adobe Dimensions or RayDream's addDepth, you are actually creating a rendering composed of many overlapping shapes that blend together visually. This is very similar to what you get when you create a shape blend in Adobe Illustrator or Aldus FreeHand. Each step in the gradation from black to white, and all the colors in between, is a solid block of color. The advantage of this is that a PostScript blend is infinitely scalable; the disadvantage is that as you increase the size of the image without increasing the resolution of the rendering, you get "banding" of colors. Another disadvantage of this method is that it doesn't account for realistic effects such as transparency, reflection, and others. Also, current versions limit you to white lights.

On the other hand, this is an extremely effective technique for producing artwork that needs to fit into existing production methods. PostScript blends can be used to create surprisingly beautiful illustrations. Another advantage is that this art can be easily passed around and used in different types of documents at different sizes. Although they are difficult to compare to other rendering systems, addDepth renders a typical image in under a minute on a Macintosh Quadra, and Dimensions, running on a Power Macintosh, can render a really complex texture-mapped image in a minute or so, though it's much slower on 680x0 Macs.

## **Wireframe**

Wireframe displays are the standard mode of displaying models during the modeling and scene building stages of 3-D. However, it is sometimes desirable to create a "wireframe rendering" of a scene or model (see figure 10.1). An interesting use of

wireframe renderings is the creation of renderings that appear to be part sketch, part finished image. This gives a natural, hand-made look to images.

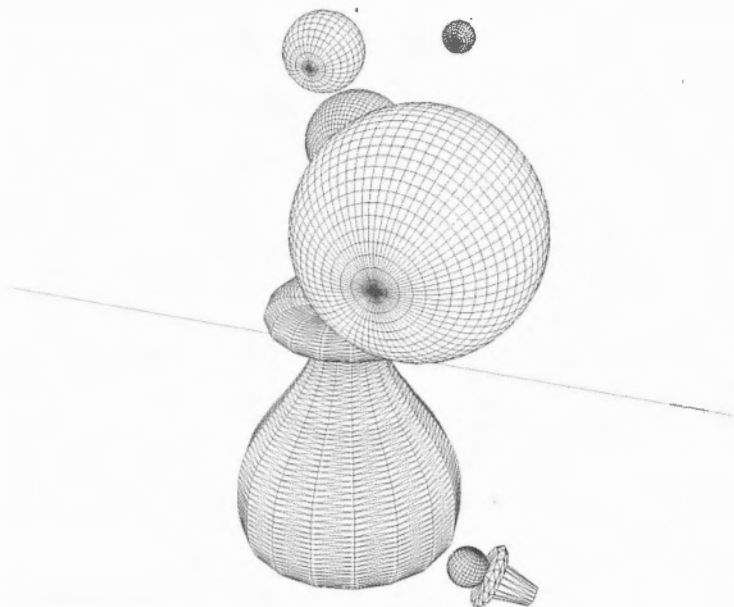


**Figure 10.1** Wireframe rendering created in StrataStudio Pro (note that the internal surface is visible in the jar)

Wireframe renderings are commonly saved as Encapsulated PostScript (EPS), a format you can incorporate into most PostScript illustration and desktop publishing packages. Some 3-D programs will export an editable EPS wireframe that you can then modify in 2-D in a program such as Illustrator or FreeHand.

## Hidden-line Rendering

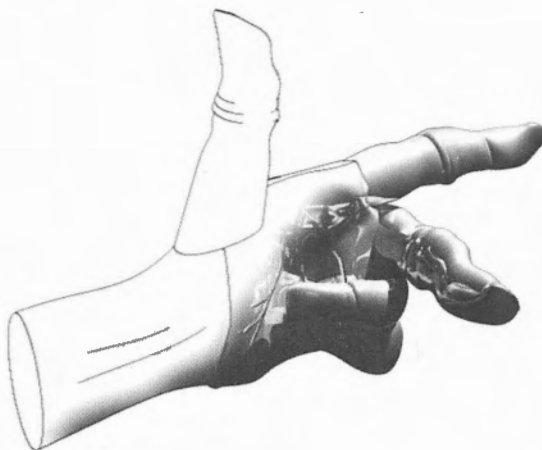
*Hidden-line rendering* is a popular architectural, engineering, and technical illustration technique. Hidden-line rendering is essentially the same as wireframe rendering, except that foreground surfaces hide the lines of background surfaces (see figure 10.2). This creates the effect of outlined solid objects without shading. You'll usually use this rendering method when you want to create an image that looks as though it were hand-drawn in perspective. A hidden-line rendering can be touched up with "watercolors" in a program such as Painter or Photoshop to create a very traditional-looking architectural rendering.



**Figure 10.2** *Hidden-line rendering*

Hidden-line renderings are a standard in technical documentation, such as service manuals, because they print well in black and white and are easy to read, even at small sizes.

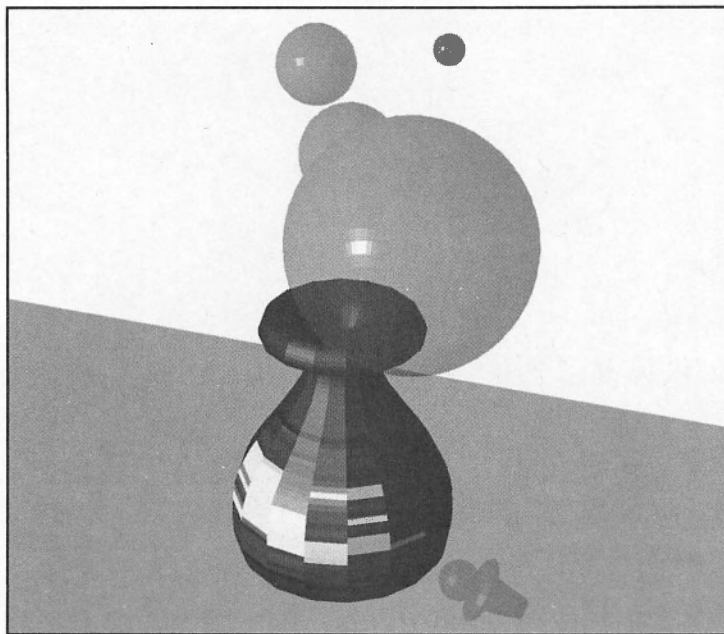
This rendering method is sometimes used when you want to create the effect of an unfinished high-tech project. It is particularly effective when composited artfully with a realistically rendered part of the same scene—for example, giving the effect of a skeleton under skin.



**Figure 10.3** *Hidden-line rendering composited with ray tracing of the same model*

## Flat Shading

*Flat shading* is the simplest and fastest method of shaded rendering. It derives its name from the faceted, polygonal appearance of the images it produces (see figure 10.4). This technique applies a single color to each of the polygons on the surface of a shape. This is the rendering technique favored by programs more concerned with the display of information than photorealism. Mathematica is a good example of a special-purpose program that uses this technique. It enables you to graph the results of mathematical equations as shaded 3-D models.



**Figure 10.4** Flat-shaded rendering created in *StrataStudio Pro*

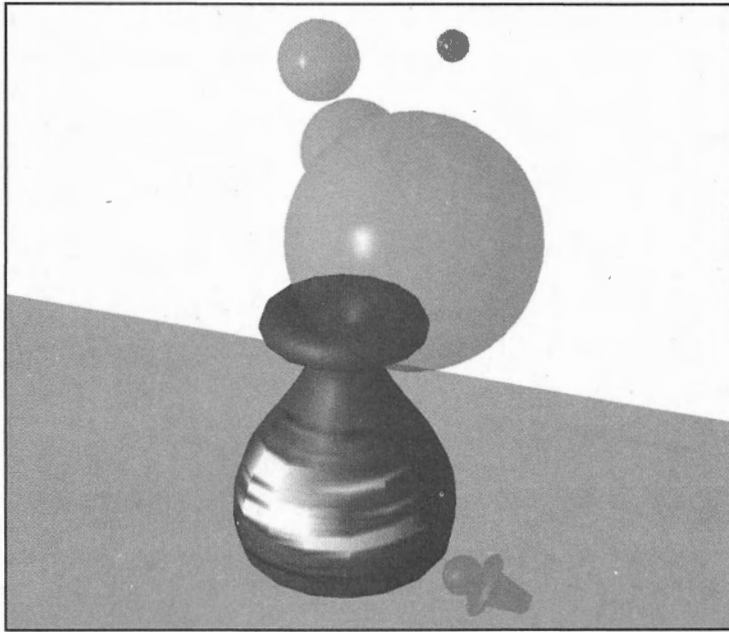
Because it is the fastest form of shaded rendering, it is commonly used by modelers to display geometry at the modeling stage. Some programs (such as VIDI's Presenter Pro and MacroModel) are fast enough at flat shading to provide rendering in near real-time.

The drawbacks of flat shading are that it doesn't allow for realistic transparency, reflection, or texture mapping. It produces renderings that are unnaturally faceted and easily identified as being computer generated.

On the other hand, flat-shaded models are great for creating modern renderings with a high-tech, video game look, and they do at least accurately convey lighting, coloring, and shading.

## Gouraud Shading

*Gouraud shading* (pronounced “juh-ro”) goes one step further than flat shading by blending the edges of polygons toward their centers. This generates a very smooth rendering. But fine details, hard edges (such as an object’s corner), and textures are “smeared” together, resulting in rather blurred shapes (see figure 10.5).



**Figure 10.5** *Gouraud-shaded rendering created in StrataStudio Pro*

This also is a very fast rendering algorithm, so it is widely used for “smooth shaded” previews, and even for some finished renderings, such as those produced by Specular’s LogoMotion. For animations where the shape of objects flying around in space is paramount, Gouraud can be a good solution.

Like other simple forms of shading, Gouraud is incapable of accurately depicting transparency, refraction, or reflection.

## Phong

*Phong rendering* (named for Bui Tuong Phong, an “illuminary” who figured out how to smooth polygonal surfaces) calculates the shading of each pixel of an image based on the brightness, color, and direction of light sources striking that pixel; however, it does not “trace” the light as it is reflected. While it is not as realistic as ray traced rendering, it is much faster in most cases. Phong shading is the preferred method for most animations since it is the simplest form of realistic rendering. EIAS, the most expensive general rendering system for the Macintosh, offers Phong shading as its “best” algorithm. With creative texture mapping and environment mapping, Phong rendering can produce images as realistic in appearance as ray tracings, although achieving realistic looking reflective surfaces requires more work on the part of the user than ray tracing (see figure 10.6).



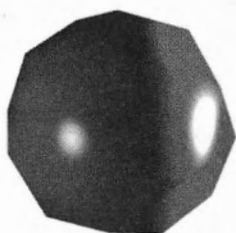
**Figure 10.6** Phong rendering created showing realistic, accurate textures

## Tip

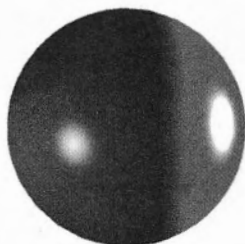
While Phong shading is a workhorse renderer, it is generally not as aesthetically pleasing as ray tracing. This is because many surfaces have subtle reflectance and transparency whose effects can only be recreated with ray tracing. Lighting and colors are sometimes also more accurately represented by ray traced rendering. Ray tracing is not just for rendering highly-polished chrome surfaces.

Unlike ray tracing, which produces shadows as a matter of course, Phong shading uses “buffered” shadows, usually increasing RAM requirements and reducing rendering speed. The two places where Phong shading completely breaks down are the mirror reflection of objects and refraction. While Phong shading does allow for environmental reflections and transparency (without refraction), mirror reflection and realistic transparency both require ray tracing for accurate depiction.

Although Phong shading generates very smooth surfaces, no renderer smooths the faceted polygonal edges of coarse wireframes, so where smooth edges are important (particularly in high-contrast foreground and background relationships) you may want to increase the wireframe detail of your models at the modeling stage (see figures 10.7 and 10.8). This is true no matter what rendering system you use.



**Figure 10.7** Phong rendering of a coarse wireframe created in Sketch!



**Figure 10.8** The same model with Phong shaded in Reflection maps



## Phong Reflections

In Phong rendering, color and brightness are calculated in reverse, from viewer to object to light source. However, only light striking a surface and bouncing towards the viewer is calculated, not light bouncing from one surface to another to the viewer; true reflections are not possible with Phong shading. This means you'll need to make creative use of environment maps to simulate reflections on surfaces.

For Phong renderings where you absolutely must have local reflections (that is, a reflection of one object on another), it is possible to render the scene from the point of view of the object and use the resulting image as an environment map for that object. This creates a "fake" reflection that won't be accurate, but may nevertheless look real. EIAS does this automatically, calling it "Automatic Environment," but the results may be slower than ray tracing.

## Ray Tracing

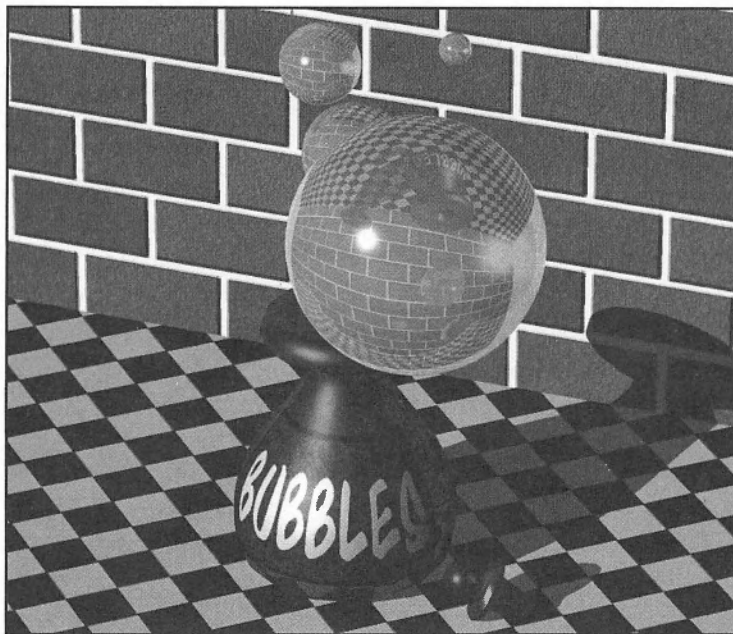
*Ray tracing* is the most realistic rendering technique commonly available; it is often referred to as "photorealistic." (However, Phong renderings can be just as convincing, depending on the scene.) Ray tracing follows the path of light rays from the light source to the surface of objects to the viewer. Unlike Phong rendering, ray tracing also accurately depicts the direction of light even after it is reflected, and the color and intensity of light as it bounces between and passes through objects.

Because ray tracing tracks the color and direction of light through a scene, it is capable of accurately depicting complex light effects. These include mirror reflection, shadows, transparency, and refraction—qualities that are only approximated by other rendering systems (see figure 10.9).

### Tip

The only reason not to use ray tracing is time constraints. Typically, a Phong rendering on a Quadra 950 may take ten minutes to an hour, depending on the complexity of the scene. The same image that takes an hour in Phong shading may take two days to ray trace if there are several mirror reflections and transparent objects with refraction. Recursive reflections in particular wreak havoc. Since a Power Mac found a home on my desk, I find myself ray tracing with more and more frequency; it now takes less time than it did to Phong render a short time ago!





**Figure 10.9** Ray traced rendering with mirror reflections (in the bottom of the jar), transparency, and refraction

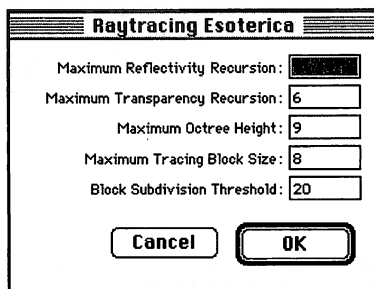
## Mirror Reflection

In ray tracing, a ray of light bounces from one object to the next, carrying color and intensity as it goes. This means that an object's appearance will bounce around in a scene until it reaches the viewer. You see not only objects, but objects reflected off of each other as well. Mirror reflectance affects much more than glass mirrors. Some of the items that require ray traced reflections for a high degree of realism include: chrome or any other polished metal, shiny plastic, polished marble and tile, glass, lacquered wood, and smooth ceramics.

## Recursion

Because objects in 3-D can have near or total reflectance, it is possible to create scenes where light rays bounce around infinitely from object to object, similar to a ball in a pinball machine with a bumper blocking every exit. This can result in a rendering taking literally forever when tracing a single infinitely bouncy light ray, and is particularly likely when there are a number of rounded reflectors in a scene (such as chrome pipes or spheres). Ray tracers, therefore, offer a setting to limit the

number of “recursions” of a single reflection (see figure 10.10). Typically, this number is set at one or two, although for extremely reflective scenes (such as a hall of mirrors), you may want to bump the number up. Just be prepared to wait a very long time for your rendering to finish—as each recursion proportionately increases rendering time for reflective surfaces.



**Figure 10.10** The Expert Settings dialog box in StrataVision 3d's textures control panel enables you to set limits on recursion

## Transparency

Ray tracing is capable of accurately depicting the transmission of light through transparent objects. The most basic effect of this is that you can see other objects behind a transparent object. This also means that light can pass through a transparent or translucent surface, affecting surfaces on the other side. As light passes through green glass, for example, the light takes on the green color.

In the case of glass, for example, a material can have transparency combined with other surface qualities (such as reflection), so that some light will pass through the surface, while some will bounce off.

## Refraction

Depending on the type of material, light bends as it passes through a surface. This quality is known as *refraction* and is one of the major visual clues that distinguishes one “clear” material from another. The most common example of refraction is that of a glass lens.

Like mirror reflection, refraction is subject to recursion. It is possible to have a series of objects that bend light into a continuous loop so that the light rays never escape to reach the viewer. An interesting pitfall of refraction recursion is that if the setting for this is too low, transparent objects will mysteriously appear to be opaque when you render.

**Tip**

When using transparent objects, consider using a rendered background instead of post-compositing your image with an alpha channel. Refraction cannot be rendered into an alpha channel; if you want your background to show through transparent objects naturally, you will need to include it in the scene when you render.

## Radiosity

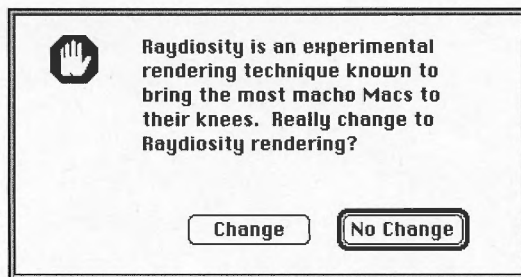
*Radiosity* is an advanced form of ray tracing that accurately depicts the interaction of light reflecting between surfaces. The original radiosity specification, developed at Cornell University, enables you to “render” a scene once—then fly around within the scene and see all of it pre-rendered (because the radiosity algorithm has already calculated all of the available light on all of the surfaces present). Radiosity has other advantages as well. Because it accurately depicts the interaction of light reflecting between surfaces, radiosity renderings are much more lifelike than ray-traced renderings.

The only radiosity rendering currently available to Mac users is provided by StrataStudio Pro, but this implementation, which Strata calls “Raydiosity,” is limited to the rendering of still frames. It doesn’t support the notion of pre-rendering an entire scene.

Since mirror-like surfaces are actually relatively simple in terms of how they reflect light (it merely bounces off), this algorithm is mostly used when you require extremely life-like matte finishes and shadows. Matte surfaces have a minute roughness that scatters light very subtly, giving the surface a soft-edge glow.

Examples of matte surfaces include: flat spray paint, asphalt, rough metal, sanded wood, and soft cotton fabric. Matte surfaces, although non-reflective, will often have a barely-perceptible glowing effect due to the interaction of light with the surface material.

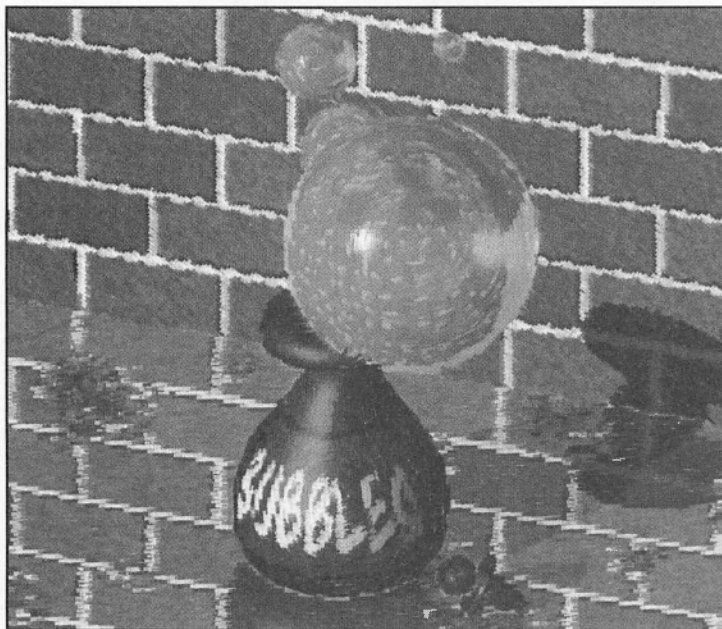
While radiosity combines the accuracy of ray tracing with the accurate depiction of surfaces, it is slower than ray tracing. Thus, it is suitable only for extremely critical renderings requiring extremely realistic results, and then only if you have the fastest possible rendering setup. It is safe to say that it has never been used in animations on the Mac. Effective use of radiosity rendering practically requires some form of rendering acceleration, perhaps distributed rendering on a network of Power Macs (see figure 10.11).



**Figure 10.11** The Warning dialog box in StrataStudio Pro, making it clear that radiosity is not for the faint-hearted

## RayPainting

StrataStudio Pro has another novel rendering method called “Ray Painting,” that paints a surface with splotches of color. This is much faster than normal ray tracing because the renderer need only render a small fraction of the total pixels in the image. It uses the results of this to smear brush strokes in an orderly fashion over an object’s surface (see figure 10.12). Unlike Photoshop plug-in style filters, Ray Painting is “smart” in that brush strokes follow the contours of objects.



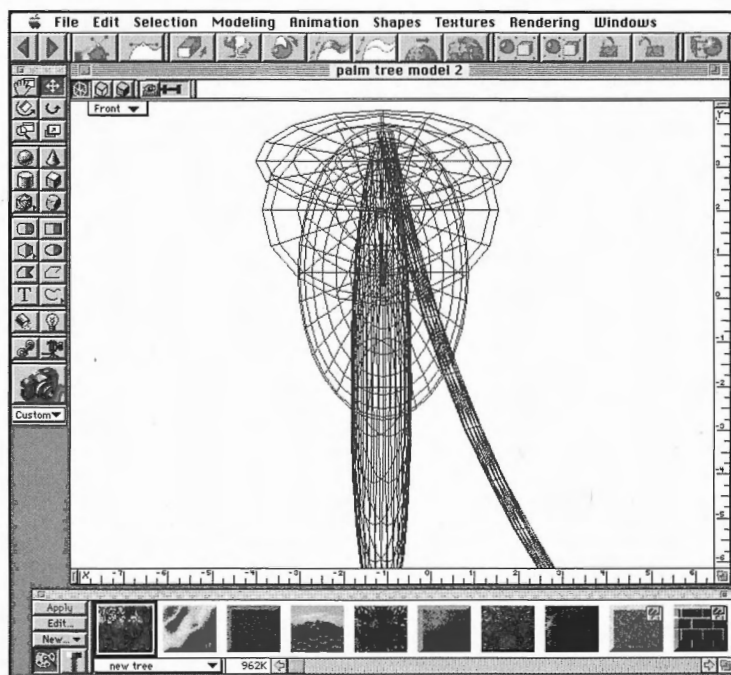
**Figure 10.12** Ray painted ray tracing created in StrataStudio Pro

## “Modeling” with Transparency

Experienced 3-D artists spend far more time texture mapping than building models. There are several reasons for this, but the most common reason is that many impossibly complex models can be simulated with creative texture mapping. Renderers also are much faster at dealing with texture maps than with hugely complex model geometry.

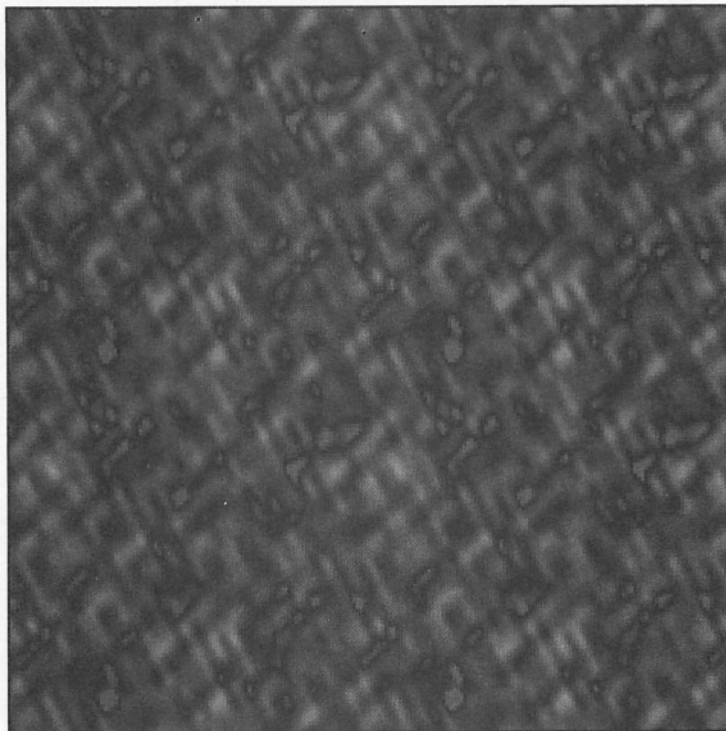
The following example shows a palm tree that appears to contain thousands of tiny parts, but is actually comprised of one lofted shape for the trunk and four spheres for the crown! This example could have easily been in the Materials chapter, because the detail is all in the texture maps. This technique can be adapted to create all kinds of objects in most 3-D programs that support effects maps.

1. The palm tree model consists of five simple shapes: first, build a series of circles and loft them together for the trunk; then create four nested spheres for the tree's crown. Each of the spheres is stretched thinner and further downward than the previous one (see figure 10.13).



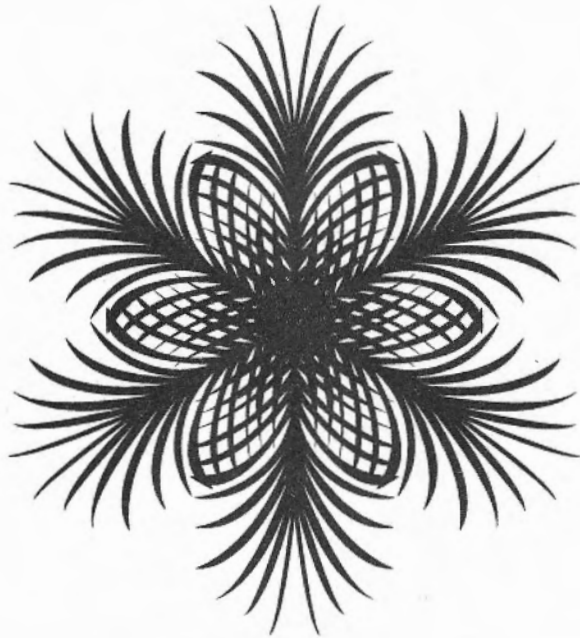
**Figure 10.13** Palm tree model consisting of a trunk and four spheres

2. The color map for the palm fronds (shown in figure 10.14) is a fairly random bright green pattern. Create it in Kai's Power Tools' Texture Explorer.

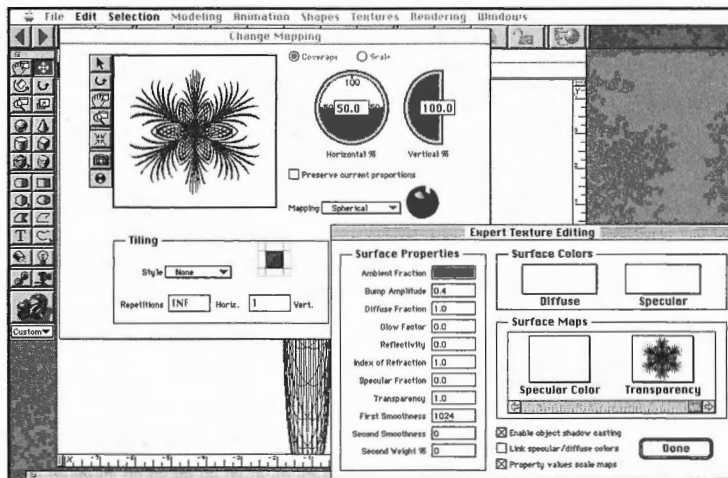


**Figure 10.14** *A simple random pattern of bright greens*

3. Create a transparency map in Adobe Illustrator by drawing one half of one frond, mirror-duplicating it, rotating, and copying the whole frond five times. Then create a new image in Photoshop. Place the Illustrator file (to result in a bitmap), and save it as a PICT (see figure 10.15).
4. Select the top, outermost sphere. In StrataStudio Pro's texture mapping dialog box (figure 10.16), create a new texture using the green PICT as the color map and the palm frond PICT as the transparency map.



**Figure 10.15** A palm frond transparency map



**Figure 10.16** StrataStudio Pro's Texture Mapping Dialog box



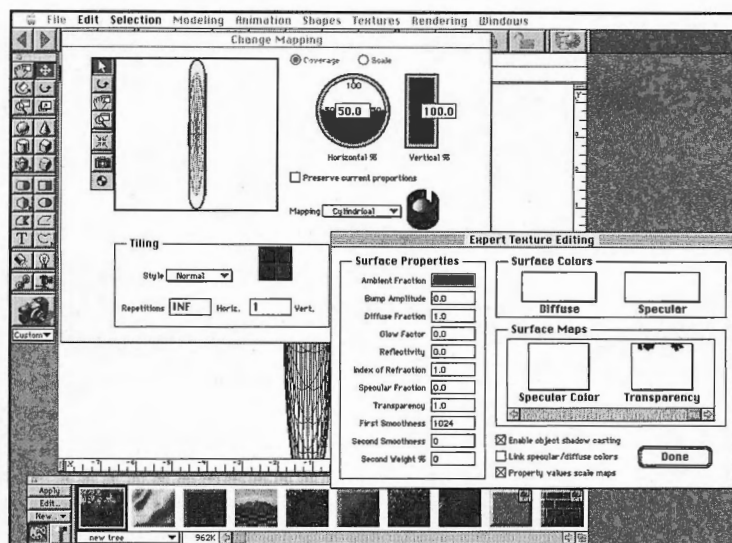
5. Steps 5 through 7 also refer to the windows shown in figure 10.16. In the expert Texture Editing window (foreground), set the Index of Refraction to 1.0. This is the refractive index of air, so the transparent spheres won't distort shapes seen through them. Also, set the reflectivity, specular, and glow fractions to 0.0 to ensure that the transparent parts of the sphere will be otherwise totally invisible.
6. Check the "Property values scale maps" checkbox to indicate that the numbers in the left-hand column will multiply the effects created by the transparency map (and any other maps present).
7. The background window enables you to change the projection, scale, and position of the texture. For the tree, set the texture to map to the top of the sphere using spherical mapping and to wrap only half-way around to the bottom. Note that the preview window shows the texture properly mapped to the sphere.
8. The transparency map for the inner three spheres is unbelievably simple; it's several blotches of vaguely leaf-shaped black spots at the very top of the PICT (see figure 10.17). The color map for this texture is the same green PICT used for the uppermost sphere.



**Figure 10.17** *The transparency map for the inner three spheres*

9. All of the settings for the inner sphere textures are otherwise the same (see figure 10.18) except that they use cylindrical mapping and the texture is scaled to be tiled twice around the circumference of the cylinder (its horizontal scale is 50 percent, but tiling is set to "Normal"). The unusual effect of this approach is that the texture is automatically pinched into the center at the top of the sphere; stretching the texture and tiling gives the fronds a long, sinewy appearance, like wilted palm tree leaves, particularly on the longest inside sphere.

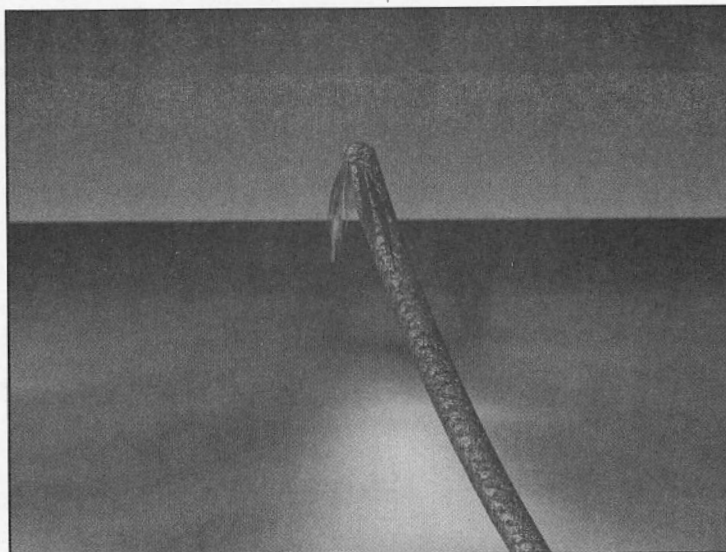




**Figure 10.18** *The inside spheres mapped with cylindrical mapping and horizontal tiling turned on*

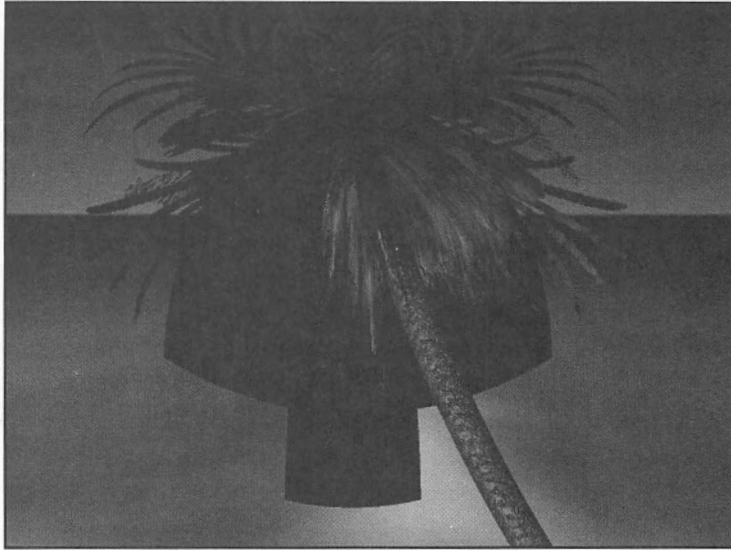
10. Move the camera into position below the palm tree, and import a background. Now, it's time to render, but the first attempt, using Phong shading, reveals a severely pruned palm tree (see figure 10.19).

A common problem is StrataStudio's inability to "see" the inside of a transparent object when Phong is rendering. You also get this effect if you render with "invisible backfaces" or "backfaces turned off." These expressions refer to the "back" side of polygons. Renderers will enable you to turn these off to save on unnecessary calculations (with the notable exception of transparent objects, and convoluted objects, backfaces usually aren't visible).

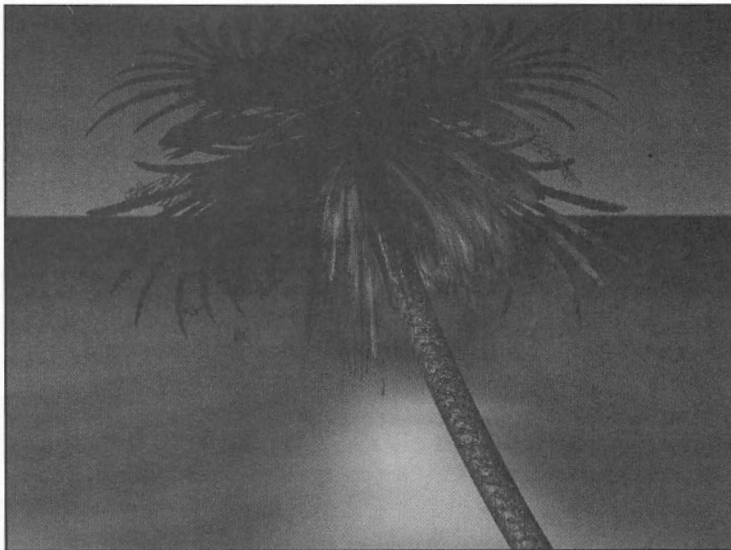


**Figure 10.19** *A first attempt at rendering using Phong shading*

11. Attempting to re-render the image with ray tracing reveals a new problem. Strange black blobs appear (see figure 10.20) where palm fronds and invisible parts of spheres should be. This is due to having too few levels of transparency recursion enabled. With both transparency and reflectivity, the maximum recursion levels determine the “strength” of each light ray. If this level is too low, the light ray dies before it reaches its destination (your eyes) and you end up with a mysterious black spot in place of transparency or reflection. With four spheres nested, as in the following image, you’ll need 8 levels of transparency (one for each side of each sphere) in order for the rays to punch all the way through. To change this value, open the Expert Settings dialog box, and set transparency recursion to 8.
12. Success! A perfectly rendered palm tree—with nary a moment wasted model building (figure 10.21).



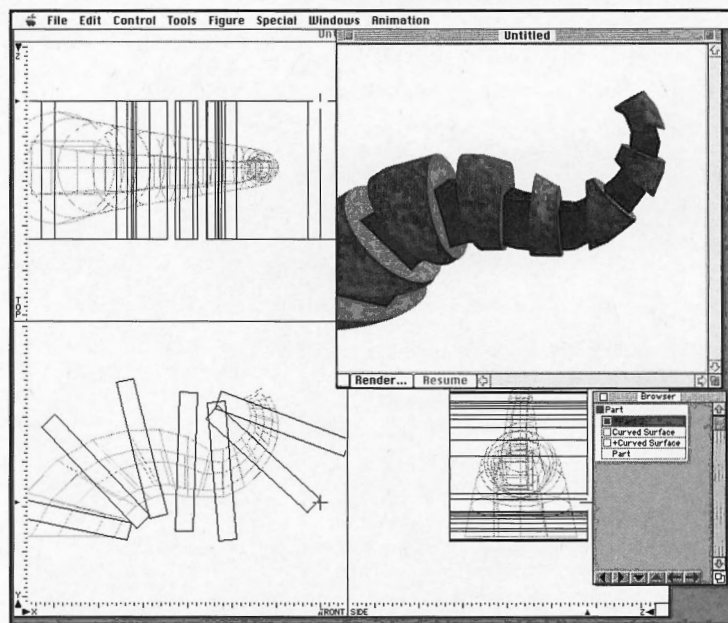
**Figure 10.20** *The effect of using too few levels of refraction*



**Figure 10.21** *The finished ray traced palm tree*

## Boolean Rendering

Shade III and RenderMan use Boolean operations during rendering. Just as you can subtract one volume from another in a solid modeler such as Zoom, Modeler Professional, Sculpt, or form•Z, you can use RenderMan or Shade III to do this at the rendering stage. In Shade III, these effects have an additional option: If you subtract a sphere from the edge of a cube, for example, you can choose to have the texture of the sphere applied to the walls of the hole left in the cube—even while the rest of the cube maintains its original texture. Of course, Boolean rendering is used for much more than simply subtracting spheres from cubes. In figure 10.22, a series of slab-shaped blocks is used to slice away the outer layer of a swept shape, leaving a new surface texture on its interior surface and leaving the inner core shape completely intact. All of this is accomplished by putting a “\*” or “+” in front of the object’s group name.



**Figure 10.22** *Shade III offers Booleans in rendering*

This saves you from having to create multiple surfaces to get multiple texture maps on a single object. It also has an unusual feature you can't get from solid modeling—the capability to animate boolean operations by animating the models involved. For example, you may create an animation that shows a drill boring down through a block, and removing material as it goes. This would be virtually impossible to do any other way.

Whether or not you can achieve similar Boolean effects with RenderMan, compatible programs depend entirely on the level of control your particular animation program allows over RenderMan parameters. Certainly, Presenter Professional and Macromedia's Three-D have the power to do this.

## Rendering Quality

Rendering quality is a subjective measure. Just what makes rendering high quality is up to the taste of the person using it. For my purposes, I define high-quality rendering as that which most resembles what we see with our own eyes in the everyday world. Of course, no computer program can ever hope to totally recreate a natural scene. But the capability to create the illusion of reality, hence the term *photorealism*, is the goal.

## Realism

A realistic image is one that accurately portrays real objects in everyday life in realistic settings. Many aspects contribute to the sense of realism:

- **Details.** These serve as clues to the authenticity of a subject. Shading and shadow, textures and reflections, and modeled and mapped details all lend to the illusion that you are seeing a real-world image.
- **Effects.** These may be as subtle as a minuscule grittiness on an object's otherwise smooth surface, or as dramatic as a bright red glow from a light source.
- **Control.** The careful placement and revelation of important parts of a scene. This may entail simply placing objects exactly where you want them, or may include one object carving into and revealing part of another object.
- **Organic appearance.** Nothing looks more computer generated than an image filled with perfectly smooth, perfectly symmetrical, perfectly geometric objects. Real-world objects are lumpy, misshapen, and malformed.

## Detail

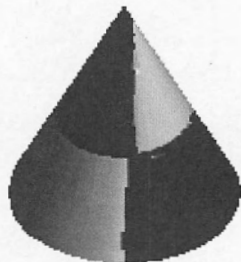
One of the distinguishing factors of better rendering systems is that they allow for a high degree of flexibility in texture mapping and detail rendering.

It is important to keep in mind that detail can be created in many ways. Modeling creates the most realistic details at the expense of time and rendering speed. Many details can be created through the judicious use of texture maps, bump maps, and carefully placed backgrounds.

## Antialiasing

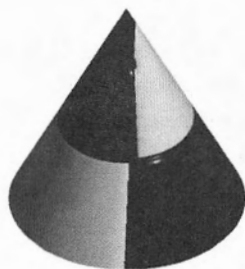
*Aliasing*, also called stair-stepping or jagged edges, is a problem of digital imaging that results from a computer's habit of working in visible square pixels as the smallest unit of an image. Aliasing occurs at the edges of contrasting curved or diagonal lines, but its effect is devastating to the best-laid renderings. An easy way to picture aliasing is to try drawing a perfect circle on a sheet of graph paper by filling in squares. Even though you may generate an approximation of a circle, its edges are very jagged. ("You can't square a circle.")

Antialiasing combats the "jaggies" by blending the pixels at the edges of objects into the background (see figures 10.23 and 10.24). A black circle on a white background, for example, becomes a black circle with a thin ring of gray pixels on the outside edge blending in to the white background. When viewed from a distance, you can't tell exactly where the white background ends and the black ring starts; they appear to fit smoothly together. Even though antialiasing essentially works by smudging the edges of objects, the visual effect is that of creating a smooth sharp line.



**Figure 10.23** Image rendered in StrataStudio Pro, with no antialiasing





**Figure 10.24** The same model with a high degree of antialiasing

There are several different methods for creating antialiased edges. Most 3-D programs use supersampling. Essentially, *supersampling* renders a high-resolution image and uses the extra rendering information to determine the blending of the edges. The result is a normal-resolution rendering, with the high-resolution information being discarded after antialiasing.

The simpler and much less time-consuming method of antialiasing is practiced by some 3-D programs, as well as by antialiasing utilities such as Jag II from RayDream. These programs can work on finished renderings (unlike supersampling, which is an integral part of the rendering process). This is sometimes called *interpolative* (or “averaging”) antialiasing. It works by interpolating between pixels and their neighbors. When the program encounters an area of high contrast or color differential, it interprets between the contrasting pixels to blend the edges. In areas of gradual color or contrast variation, pixels are left alone in order to avoid unnecessary blurring of the entire image.

## Wireframe Complexity

The wireframe complexity of an object in 3-D determines the level of actual geometric detail in a model. For example, a polygonal modeler may simulate a circle by creating a many-sided polygon. At a very low level of detail, this may be an octagon, while at a very high level of detail, it may use a 64-sided polygon that resembles a circle.

Even in spline-based systems, wireframe complexity has a direct impact on the level of detail possible in a model. To create a bump on a surface between two splines, you will need to create several new splines to use as contours for the bump.

Wireframe complexity is often distinct from rendered smoothness. That is, it is possible to create a smoothly rendered image from a very coarse wireframe through the trickery of Phong rendering, and so forth. However, more complex wireframes will lead to more accurate, more detailed images. The trade off is that the increased volume of data required for more complex wireframes results in decreased speed (and increased storage requirements) throughout the 3-D process.

Most 3-D programs have options for increasing wireframe detail, usually while creating a model. Sketch!, on the other hand, benefits from its use of splines in that you can reset the wireframe complexity of models at any time. You even can work on a model in coarse wireframe mode, then switch to finely-detailed wireframes for final detail work. Sketch! enables you to render in a different level of detail than the one in which you work. It also enables you to set different levels of detail for different models in a scene.

Presenter Professional also enables you to change the level of detail used by different models.

## Smoothness

The apparent smoothness of rendered objects is the result of a number of factors: the rendering algorithms in use, the detail of the wireframes being rendered, and the degree of abruptness in change over an object's surface.

Some rendering methods, particularly Gouraud, impose an automatic smoothing on surfaces (sometimes when it isn't desirable). This is because Gouraud shading determines the color values at the edges of polygons and blends the colors together toward the polygon's center. This is like buttering the edges of your toast and spreading the butter in towards the middle.

When you use the Smooth shading option of many renderers (usually reserved for Phong and ray traced rendering), the program changes the surface "normals" (surface vectors) of facets to make a faceted model look smooth to the renderer. This is very similar to what happens during bump mapping, only instead of tweaking the surface normals to create the illusion of bumps, you're tweaking the normals to create the illusion of smoothness. To avoid overly smooth corners, smoothing options will often enable you to specify a "breaking angle," beyond which a corner will not be smoothed during rendering.

It's also possible with some 3-D programs to break polygons with four sides or more into constituent triangles, which inherently shade more smoothly than squares and larger polygons.



## Speed

Rendering speed is an overwhelming concern of 3-D users. Fortunately, it is also a concern of hardware and software vendors. Even though 3-D modeling and rendering are becoming more and more complex and demanding more and more of rendering computers, rendering times are actually getting shorter. This is largely due to the speed of computers, but also due in part to the relentless optimization of rendering software. There are many user-controlled elements that affect the speed of rendering: the type of rendering, the complexity of the models being rendered, the detail or resolution of the rendering, the number of lights, special effects, texture maps, and so forth.

Power Macs, network rendering, workstation slave rendering, and even NuBus coprocessor rendering are greatly assisting the quest for more speed (see chapter 13, “Advanced Topics”). With the advent of Power Macs, 3-D users actually started to spend more time working on 3-D graphics than waiting for 3-D rendering; but there’s always room for improvement.

## Speed Factors

There are many factors that influence rendering speed, a few of which are subject to control by the 3-D user. While it is true that some 3-D rendering applications are much faster than others, it is more often the case that better 3-D applications provide for the creation of complex scenes in an efficient manner. For example, environment mapping is a much more time efficient technique than ray traced reflections. A program with good control over environment mapping techniques will often get the job done more efficiently than one that settles on ray tracing as the only way to create a reflection.

For users without a state-of-the-art Macintosh, it also is important to keep in mind that the factors that affect speed also are generally the ones that affect memory requirements. Cutting down on any of the more intensive tasks will lighten the load on system requirements as well as speed things up.

## Rendering Type

The type of rendering used contributes directly to the speed of rendering. Radiosity and ray tracing, respectively, are the slowest algorithms. Depending on what you are trying to achieve, RenderMan also can be extremely slow (Pixar’s 3-D scene building program, which relies on “.slo” shaders, has long been nicknamed “slo.place”). On

the other hand, RenderMan, using one of the many acceleration options, is one of the fastest rendering systems available to Macintosh users. Phong shading is the fastest realistic rendering method available in most modelers; Gouraud and flat shading are fast, but sacrifice rendering quality.

## **Scene Complexity**

The amount of time it takes to render a scene increases as you increase the number or complexity of any of these factors:

- Rendering method
- Models
- Texture and environment maps
- Lights
- Reflections and refractions
- Shadows

## **Image Size**

The size of the image you create also influences rendering times. Since a renderer calculates backward from each pixel in the image and the sum of all the lights and colors arriving at that point, doubling the number of pixels could more than double the total rendering time. Keep in mind that proportionally doubling the size of an image actually quadruples the number of pixels it contains (since you are multiplying both the horizontal and vertical dimensions). For example, a 640 by 480 pixel image will take four-times longer to render than the same image rendered at 320 by 240 pixels.

In practice, however, renderers actually spend more time calculating relationships of objects, lights, textures, and so on, so rendering times generally won't increase linearly as the number of pixels increase.

## **Mirror Reflections and Glass Refractions**

Ray traced reflections and refractions are among the most costly of rendering effects because, as far as the renderer is concerned, they require multiple renderings within a single scene. Recursive reflections and refractions (that is surfaces reflecting other reflective surfaces) can bring a renderer to its knees. This is why ray tracing programs enable you to limit the level of recursion for these effects.

## Antialiasing

Antialiasing also significantly slows down the rendering process. Even interpolative antialiasing adds a final step at the end that slows rendering down. Many 3-D applications enable you to set a level of antialiasing; the higher the level, the longer rendering will take.

## Lights

Every light source adds a new element to the calculation of all affected surfaces. The addition of lights to a scene particularly can slow down ray tracing because a single light can affect surfaces multiple times. This is why it is preferable in most cases to use a single light at full power, rather than two lights at half power. This makes it particularly bothersome when lights are constrained to a narrow range or even one or two settings; you will often need to use multiple lights where one would normally suffice.

## Atmospheric Effects

Effects that affect rendering speed include the use of shadows, fog, and depth of field.

Shadows are particularly detrimental to rendering speed. For this reason, it is often possible to turn off shadow rendering for an entire rendering, or just for individual objects and lights. In many cases it is only the key shadow (cast by the key light) on the main subject that is of significant value in an image.

## Texture and Environment Maps

Environment and texture maps are costly in two ways. One is that the mere overhead of opening and manipulating a sizable bitmapped image file imposes its own time penalties. This is an issue particularly when using a network distributed rendering system. In this case, each texture map must be copied over the network to every rendering node. The other penalty imposed by the use of maps is that the renderer must go through the task of mapping the image onto the appropriate objects. Nevertheless, this is almost always faster than ray tracing reflections.

## Batch Rendering

Most Mac 3-D programs now include a batch rendering facility. This means that you can queue up a series of renderings to complete in a bunch. While this won't improve the speed of rendering, it does relieve some of the burden of having to render many images. You can leave your Mac working and simply walk away until it's done.



**C h a p t e r**

**11**

**Working with  
Images**

**T**he uses of 3-D images are as varied as any other form of artwork. How you create them, of course, will have a lot to do with your intentions for the completed renderings.

There are many considerations, but none so important as final output. Will your still images be printed on a billboard, or used as slides in a presentation? Will your animation be used as a QuickTime movie on a CD-ROM, or will it be broadcast on network television?

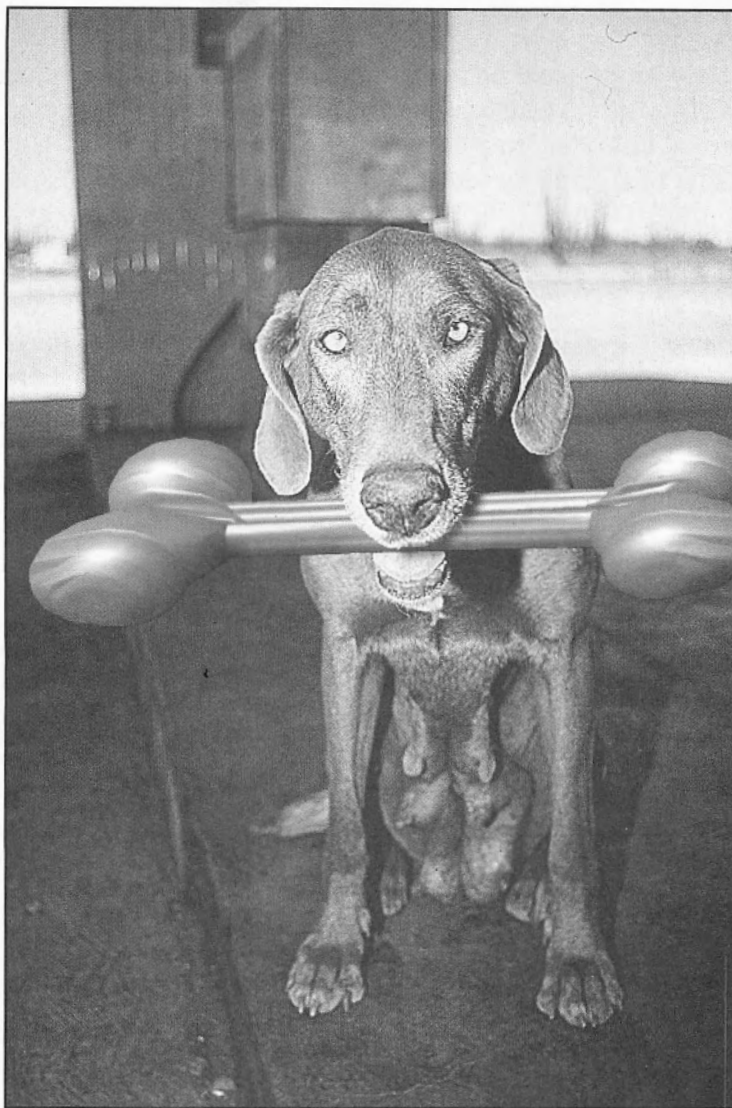
This chapter makes no claims to be an all-inclusive guide to using 3-D images; rather, it should serve as a sampling of ideas and techniques.

- For print work, you'll need to get your images into a suitable desktop publishing or color separation program.
- For multimedia, you'll want to develop animation in a format such as QuickTime, and the programs which enable you to work in that format.
- For video or film, you'll be taking your animations to a post-production service bureau for output to video, or you'll be doing it yourself with the help of a frame-accurate recording system.

The considerations go beyond simple output, however. You may want to add details to your work or special effects to your 3-D output. Or you may need to composite your images with other work so that your 3-D spacecraft flies over crowds of screaming earthlings or your architectural rendering is nestled into the landscape where it belongs.

## **Compositing**

The most common need for post processing after generating a rendering is compositing a rendered image with an imported 2-D background (see figure 11.1). Many 3-D programs enable you to import a background image before you render, but this tends to slow down rendering time significantly. The faster, more practical, and more flexible method uses the "Alpha Channel" to seamlessly composite a 3-D rendering with a 2-D background.



**Figure 11.1** This scanned photo of a dog was composited in Photoshop with a bone that was modeled and rendered in StrataStudio Pro

## The Alpha Channel

Standard photorealistic images on the Macintosh contain 24 bits of color information per pixel. This is enough to generate nearly 17 million colors. But a PICT file can actually contain 32 bits of information per pixel. In most 3-D programs, and most 2-D image processing programs, the extra 8 bits can be saved as what's commonly known as the *Alpha Channel*. This is a grayscale image that is usually used to provide transparency information. The advent of the alpha channel greatly changed the way users can work with images on the Mac.

Imagine a complex 3-D model that takes several hours to render, only to discover that you really dislike the background. Generally, this means you'll have to go back and render again with a new background. Using the alpha channel, all you have to do is open the image in a program like Photoshop and drop a new image into the background layer. Of course, it's possible to composite images without an alpha channel, assuming the part you want to dispose of is of some uniform neutral color that you can easily select with a "magic wand." However, magic wand tools can't select antialiased edges; you simply can't expect to get a cleanly composited image with this method.

Alpha channel support is even more crucial for video effects. Say, for example, that you have created a flying 3-D logo with an alpha channel. You could easily composite the logo with digitized video previews of this week's football highlights, or you can composite multiple layers of graphics, so that the word ITALY flies in over a moving 3-D map of that country. Programs such as Adobe Premiere, CoSA's After Effects, and VideoFusion's VideoFusion enable you to composite animations using alpha channels for transparency.

If your aim is to put alpha channel graphics over live videotape, you'll want to consider investing in a "32-bit" video board, such as the TrueVision NuVista Classic. This board enables you to use the alpha channel to directly composite graphics over live video.

While alpha channel support was hardly universal a year ago, virtually every 3-D package aimed at multimedia, graphics, and print work now supports alpha transparency.

## File Formats

Three-D still images and animations come in many formats destined for many kinds of output. Fortunately, the Mac excels in moving these image formats around from one program to another, so the techniques that apply to the standard formats can usually be applied to the non-standard ones as well.



Each of these formats has its own uses and its own adherents. Which formats you'll use depends largely on what you plan to do with the resulting images.

- **PICT2.** This is the standard format for exchanging 2-D image files on the Macintosh. While it supports vector graphics (scalable, drawn lines) its primary use for 3-D users is in storing and transporting 2-D bitmaps. The format supports 32-bit images (24 bits of RGB color and 8 bits of alpha), as well as all the lower color depths. PICT is the most widely supported image format on the Mac, and is the standard for use in multimedia programs, as well as many image editing programs.
- **EPS (AI).** EPS (which stands for Encapsulated PostScript) is the Adobe Illustrator format, primarily used for storing PostScript line art created in Adobe Illustrator. It is sometimes known as "editable PostScript." Adobe Dimensions and RayDream's addDepth can save rendered images in this format, while some programs, such as Presenter Professional, can save wireframes in this format. Illustrator files, which are vector based, rather than bitmapped, have the advantage that files are scalable and resolution independent. You can open an EPS (AI) file in Illustrator and ungroup and edit parts of the image without any resulting degradation of image quality. You also can save a selection of lines in an illustration as a "clipping path" which enables you to automatically wrap text around it in a desktop publishing program.
- **Scalable EPS.** This is the format normally saved by addDepth and Dimensions. Like EPS (AI) files, this format can be squashed and stretched and printed at any size without "pixelizing." However, it's not editable in Illustrator or FreeHand.
- **EPS bitmap.** Like a PICT, this stores photorealistic images in bitmapped format and can include a screen-resolution preview "header" image used for placing and cropping the image in desktop publishing and illustration programs. It will print correctly on a PostScript printer. Like any bitmap, enlarging the image results in pixelization or loss of image quality.
- **TIFF.** The TIFF (which stands for Tagged Image File Format) format is a bitmap image format often used instead of PICT. It has the advantage that it is more cross-platform compatible (many Windows packages support it) and allows for a high degree of compression. It is a popular format with desktop publishing software, because it prints quickly due to the compression which can be moved quickly over the network and decompressed by the printer. TIFF images can also store alpha channels.

- **Image.** Image files are the still image format created by Electric Image Animation System. They aren't compatible with any other program, so you'll have to convert them to PICT or QuickTime to use them outside of EIAS. The Image format is unusual in that it handles both still images and animations with alpha channel information intact.
- **JPEG.** This is an image file format that uses a standard type of compression defined by the Joint Photographic Experts Group. JPEG images are very compact, although the compression is "lossy," meaning that a certain amount of image data is thrown away when files are compressed. This is an excellent format for storing large numbers of images. QuickTime ships with a JPEG component that you'll need to open and save this format, unless you have special JPEG software (see the "Animation Formats" section for more about QuickTime). Hardware accelerators based on DSP (digital signal processor) chips make JPEG compression lightning fast, although JPEG on Power Macs is generally fast enough that hardware accelerators are no longer required. Because of the compact file sizes, JPEG is an excellent choice for storing 3-D texture libraries and archiving images. It's also a great format to use for sending out presentations on floppy disk because you can compress files to tiny sizes.



The CD-ROM that comes with this book includes the shareware program JPEGview, an excellent utility for viewing JPEG images, as well as a number of other formats.

- **Photoshop 2.5.** Adobe Photoshop version 2.5 and earlier saved a file format that retains an unlimited number of channels in addition to the alpha channel. In both versions, Photoshop calls the alpha channel "Selection" or "Channel #4," but you can save and name as many channels as you need. These extra channels work as masks or place holders that can be used to quickly and accurately select different parts of an image. When saving a Photoshop document in 32-bit image format, such as PICT or TIFF, only the primary alpha channel ("#4") is retained.
- **Photoshop 3.0.** Adobe Photoshop 3.0 introduced a new file format that supports multiple image layers, in addition to multiple channels. This is a great format for keeping lots and lots of layers of images, but the resulting files can be *huge*. Photoshop 3.0 files are not compatible with earlier versions of Photoshop.

As with animations, there are many other types of Image files, especially on other platforms. Programs like DeBabelizer (described in this chapter), and to some extent, Photoshop, enable you to convert files from one format to another.

## Color Depth

Each pixel in a bitmapped image file contains a certain amount of data—1-, 8-, 16-, 24-, or 32-bits. (While it's not necessary to know it to get your work done, 2 raised to the power of the number of bits is the number of colors available in an image. For example, an 8-bit image may contain 256 colors [256 equals 2 raised to the 8th power].) One-bit graphics are black and white; 4-bit graphics support 16 levels of gray or 16 different colors; 8-bit graphics support 256 colors (or levels of gray); 16-bits gets you into the thousands of colors; and 24-bit supports over 16 million colors, while 32-bit adds 8-bits of grayscale information for the alpha channel. It's important to note that most image formats can support images in any of these color depths. Image color depth goes hand in hand with the color depth of the display system for which the images are intended. If you're using an 8-bit display, 24-bit images will look just like 8-bit graphics; that is, you won't be able to see a lot of the details. However, 3-D renderers typically render at 32 bits, regardless of the ultimate destination of the image. Sculpt recently introduced rendering at 64 bits (which is ultimately filtered down to 32 for most Mac applications). This provides a much finer level of color accuracy and shading detail than rendering at 32 bits. Each file format has its own niche, depending on the ultimate use of the image.

### 1-bit

Black and white graphics can be used for wireframe and hidden-line renderings. These files are very compact, even at large sizes. A screen-size 640 by 480 pixel image requires about 64 KB on disk. In general, I avoid using this format, because it offers no way to antialias jagged edges.

### 4-bit

Four-bit color provides for highly “banded” colors, even with dithering, since only 16 colors or levels of gray are supported. Nevertheless this format is still sometimes used by game makers because 4-bit graphics play back very quickly and realism is less important than in other applications.

### 8-bit

Eight-bit color provides for a complete range of tones in grayscale (and is a popular format for desktop publishing), but 8 bits can produce “banding” when used for full-color continuous-tone images, such as those generated by 3-D rendering. It is possible to create realistic 3-D images in 8-bit with a common technique called *dithering*, which blends any of the available 256 colors together to create the appearance of

millions of different shades of color from a relatively small 256-color palette. Since this requires a “screen” of colors, dithering reduces an image’s resolving power for showing fine details, and sometimes produces strange and unnatural colors when an image has many areas that fall outside of the palette’s range of colors supported.

Eight-bit images are popular in multimedia because they are very compact compared to 24-bit graphics, and they load and display quickly when used in animated presentations. In addition, most Macintosh systems are not equipped with expensive 24-bit color hardware, while 8-bit is now the Apple standard. A very effective and common technique for using 8-bit images is to dither using a custom palette of colors, rather than the standard Mac palette. Simply put, an image that contains many blues and greens can be dithered much more smoothly and invisibly with a palette that contains primarily green and blue colors. Some programs, such as Macromedia’s Director, can do this automatically.

Meanwhile, Equilibrium’s DeBabelizer, has refined custom palettes and dithering to an art. It will examine every image in a series, such as those you’re putting in your multimedia production, and generate a single palette that best matches them all.

QuickTime 2.0 added the very important capability of inserting a single custom palette at the beginning of a movie. This “sets” your system to use this palette while the movie plays.

An 8-bit 640 by 480 image requires about 256 KB on disk—four times the space of a 1-bit image.

## **16-bit**

Sixteen-bit color (thousands of colors) is standard on some of the newer Macs as a compromise between expensive 24-bit color and somewhat underpowered 8-bit. While not quite photorealistic, it does offer a very broad range of colors. Sixteen-bit color closely approximates the range of color supported by NTSC video, so it’s a good compromise solution for video producers who need to preview Mac-generated graphics before placing them on tape. Sixteen-bit graphics dithered to a custom palette are nearly indistinguishable from 24-bit graphics, unless the image contains a very broad range of colors and tones. Sixteen-bit is usually adequate for QuickTime movies used for CD-ROM and other multimedia applications, since it will look great on 16- and 24-bit systems and as good as possible on 8-bit monitors. A 640 by 480 pixel image at 16-bit is about 460 KB.

## **24-bit**

Twenty-four-bit graphics (“millions of colors”) are photorealistic to all but the most discerning eye and it’s the best you can get from the Macintosh, although 64-bit 3-D graphics made their first appearance in Sculpt 4.0. In fact, Macs with 24-bit graphics are used by many people as color correction and retouching stations. Twenty-four-bit color is generally required if you want to see all of the subtle shading and details generated by 3-D rendering.

While 24-bit graphics are of sufficient quality for any application, they have one very significant drawback: the files are large. A screen-resolution 640 by 480 pixel, 24-bit image takes about 900 KB of disk space. A high-resolution image suitable for printing at 150 lines per inch on an 8.5-inch-by-11-inch page requires about 24 MB of disk space (without compression).

## **32-bit**

As mentioned previously, 32-bit graphics differ from 24-bit only in their support of the 8-bit alpha channel. This is the format you’ll want to use if you plan to composite your image with others. You will sometimes see this as the “Millions of colors+” option.

## **64-bit**

Sixty-four-bit color isn’t currently supported in file formats, although it may be some time in the near future. Basically, this is an internal format (currently supported only by Sculpt 3D) that enables the program to render images with a palette much larger than that used by 32-bit programs. The resulting images are filtered to 32-bits before being saved by the program, so the effect is very subtle.

## **Animation Formats**

In terms of file size, if still images have a slight weight problem, then animation files are heavy enough to register on the Richter scale. While a screen-size 24-bit still image weighs in at about 1 MB, an uncompressed NTSC-resolution animation tips the scales at around 30 MB *per second*. Fortunately, efficient techniques exist for storing, compressing, and ultimately transporting these behemoths to their destinations.

## Numbered PICT

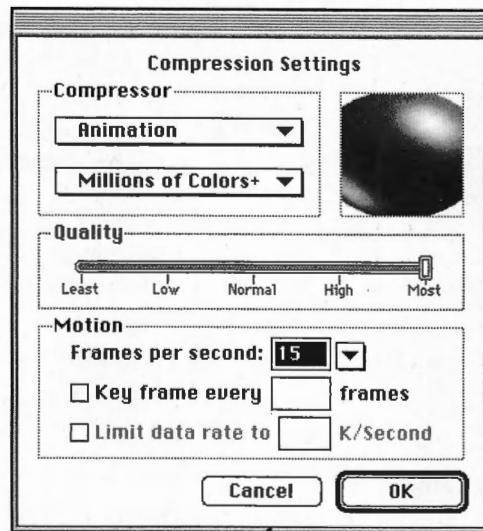
This format is no different than standard PICT, except that some programs can generate every frame of an animation as a PICT file, with each frame numbered sequentially. This is a good choice where absolute fidelity and maintaining the alpha channel is required. It is also the least efficient storage method in terms of compression and playback. It's popular with video producers, who record their images to videotape and backup the Image files to DAT tape or other high-capacity backup media. If a frame is "dropped" when recording on a professional tape deck, it's a simple matter to open the single missing PICT file and insert the image into the blank space on the tape. With numbered PICTs, it's also easy to renumber images to rearrange their order of appearance on a tape.

Service bureaus will often accept numbered PICTs for tape output. Programs such as ASDG's or Knoll Software's Abekas drivers can work with PICT files as source material.

A final consideration is that numbered PICTs can easily be opened by image editing programs for retouching or rotoscoping with a painting tool.

## QuickTime

QuickTime is an architecture that supports a range of file compression formats for 3-D animation, sounds, text, MIDI music, and still images. QuickTime uses different compression and decompression components or "codecs" (see figure 11.2), which are essentially different file translators that work within the QuickTime architecture. Most formats enable you to trade image quality for compression and speed, and to set the frame rate. QuickTime's Video, Cinepack, JPEG, and MPEG codecs use "lossy" compression; that is, they throw away some data in order to achieve greater compression efficiency. The result is some loss of image quality.



**Figure 11.2** *The QuickTime Compression Settings dialog box*

The animation and image compressors, meanwhile, enable you to save images and animations without data loss at the expense of efficiency. You must have the QuickTime extension in the Extensions folder (in your System folder) in order to use its features. (As I write this, the most current version of QuickTime is 2.0.) Most Mac 3-D animation programs now have an option to save movies in QuickTime format. If you don't have this option, you can use a variety of software to convert other types of animation files to QuickTime. For 3-D users, the most important compressors include:

- **Animation—Millions of Colors +.** This compressor is the most commonly used for storing 3-D animations. It saves the first image, and then saves only subsequent changes from one frame to the next, resulting in a complete, compact file with the alpha channel intact. Because animation backgrounds tend to remain fairly static, with only foreground objects in motion, this is an effective technique enabling rapid compositing of files.

On the other hand, if you're creating animations with moving, or video backgrounds, or you're trying to achieve the best possible playback from CD-ROM, the "Video" or "Cinepack" compressors are a better choice. For animations like flying logos over a black background, Animation-compressed files play extremely quickly and cleanly. The Animation codec uses lossless compression at the highest settings and lossy compression at the low-quality settings.

One potential problem with this format is that you need to have the whole clip in order to use the frames at the end of the clip; this is because it bases its compression on a sequence of frames over time. This becomes a problem when you're trying to edit a clip with a program that doesn't refer back to source files, such as simple movie players. This generally isn't a problem with programs like Premiere.

- **Animation—Millions of Colors.** This compressor (without the "+") leaves out the alpha channel, resulting in increased speed and smaller files. It's not, however, the format to use if you plan to composite your animations with other files. This is the format to use if you plan to play back your animations without compositing them, or if you'll be playing them directly from disk.
- **Video.** This is the best format to use when you need high-quality clips that play in real time on faster Macs. This codec and the Compact Video codec are appropriate when using rotoscoped (moving) backgrounds, since the whole image changes from frame to frame. The alpha channel is not supported.
- **Apple Cinepak.** This format is as much as ten times slower to compress than the Video format, but it generally plays back about twice as fast. It's a very lossy format, but appropriate when you need animations to playback from CD-ROM at high frame rates, at large sizes, or from slower Macs. With Power Macs, Cinepak can achieve full-frame 640 by 480 video at 30 fps off a hard disk (using QuickTime 2.0). One notable feature of this compressor is that you can set the maximum data rate, so that users with slow Macs can still view your movie at an acceptable frame rate.
- **MPEG.** A relatively new video compression format that's popular in the DOS world. A new generation of very inexpensive MPEG playback boards is on the way, so this compressor should see a lot of use. Pundits claim MPEG will put 70 minutes of full-motion, full-screen video on a CD-ROM.
- **None.** A non-compression "compressor," this is analogous to saving a series of sequential PICT files. It results in the least degradation of image quality, as well as the least benefit in terms of storage and playback. It's best use is when you want to capture video without any quality loss, such as when you're preparing textures for rotoscoping.
- **Apple Photo-JPEG.** This compressor is used primarily for compressing still images. It's an excellent choice for archiving images or storing texture libraries (see "JPEG" earlier in this chapter). JPEG accelerator boards typically come with their own compressor software (which may or may not be a



QuickTime codec) that supplements this one; “AV” Macs can take advantage of some of the available acceleration software through their on-board DSP chips. Most of the current breed of digital video boards, such as Radius’ VideoVision Studio and SuperMac’s Digital Film, rely on JPEG compression and come with their own software codecs.

## **PICS**

PICS is a format originally introduced by VIDI, specifically designed for storing animations. Macromedia’s Director has probably done the most to popularize the format, and until the introduction of QuickTime, it was a ubiquitous standard. Most 3-D animation programs can save a PICS file, and most multimedia and video production programs can open one. However, QuickTime is generally regarded as a better all-around solution.

Like QuickTime’s Animation compressor at its highest quality settings, PICS maintains all of the quality in an image and retains the alpha channel. It has the advantage of keeping every frame in a single file, although it is much more difficult to repair or replace if a frame or two is corrupted. PICS doesn’t require QuickTime to work—a significant consideration for users of older system software or those low on RAM. A recent version of PICS works like the Animation compressor in QuickTime, storing the differences between subsequent frames.

One further limitation of PICS, which makes it unsuitable for most video work, is that it has a 16 MB file size limit.

## **EIAS Image**

This is Electric Image Animation System’s format for still pictures and animations. The format has little practical use unless you’re going directly to tape from Electric Image’s Projector application. You can compress Image files into FAST files which are self-running animations that will usually play at full speed at 640 by 480. Other programs don’t support Image-format files, so you can’t use the format as a medium of exchange. You will need to store files in this format if you plan to use them as textures or backgrounds in EIAS because the program won’t accept other formats. EIAS ships with a plug-in for Photoshop that enables that program to create and acquire Image files; a similar feature is available in Equilibrium’s DeBabelizer.

## Adobe Photoshop

Adobe Systems Inc.'s Photoshop is by far the most important image processing tool available on the Macintosh. It is often compared to a Swiss Army Knife—many different tools in one—and it is as good at image compositing and color separation as it is at painting and special effects. There are other programs that perform individual functions better, but if you work with 3-D graphics on the Mac, it's pretty hard to get by without Photoshop. The following are a few examples of where the 3-D user is likely to need Photoshop:

- **Compositing images.** Because of Photoshop's robust support for alpha channels and layers (you can have unlimited layers and channels in a single document) it is the ideal platform for merging 3-D graphics with one another and other images.
- **Textures and backgrounds.** Photoshop gives you an unlimited source of material to use as textures and backgrounds. You can use it to create texture maps, glow maps, bump maps, transparency maps, and on and on. It's also the interface required to use many color scanners. Further, Photoshop plug-ins, such as Kai's Power Tools, are the cornerstone of a 3-D creator's texture supply.
- **Details.** If you need to touch up images to add subtle highlights and shadows, paint bits of grime on walls, or "strip in" an all-important fly on the nose of a statue, Photoshop does it all.
- **Generate special effects.** Photoshop plug-in filters such as Paint Alchemy and Gallery Effects enable you to add unique special effects to renderings. You can filter a photorealistic image to look as though it were painted with watercolor or sketched with charcoal, for example.
- **Image retouching and color processing.** You can balance the colors in an image and paint out unwanted glitches. You also can "legalize" palettes for use in NTSC video or convert images to CMYK for color printing.

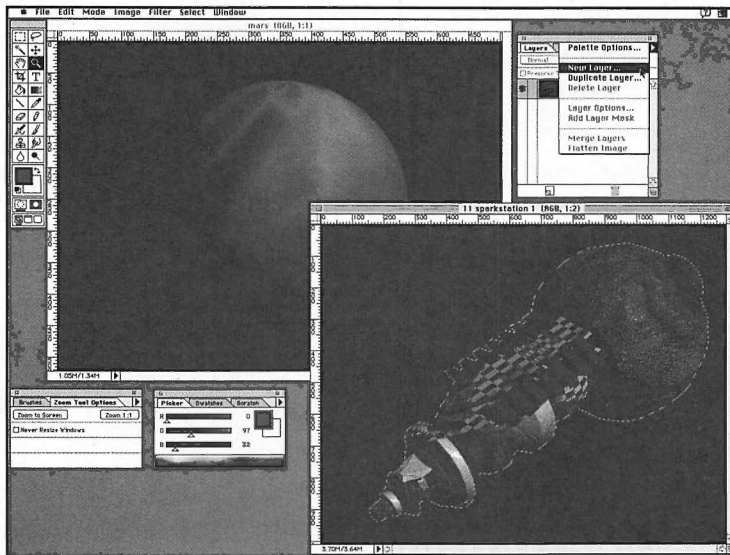


A "Try Me" version of Photoshop is included on the CD-ROM.

## Compositing Images

Photoshop 3.0's multi-layer format makes it painless to composite images and the program enables you to use the alpha channel in a variety of ways. The most common is compositing a 3-D rendering with a photograph or other image, in this case a rendered version of Mars.

1. Begin by creating a background image and a separate 3-D rendering in a program that supports an alpha channel, in this case StrataStudio Pro. Open both of them in Photoshop 3.0. With the spaceship image active, in the Select menu, choose Load Selection to select only the outline of the spaceship. Note that even parts of the ship in dark shadow against the dark background are selected without trouble (see figure 11.3). Choose Copy to move the selection to the clipboard. Then create a new layer in the Layers palette, select it, and Paste the ship in from the clipboard. This makes a duplicate of the selected area in the new layer with a transparent background.



**Figure 11.3** Opening a background image and a 3-D image, rendered with an alpha channel, in Photoshop

2. Drag the new layer from the spaceship image onto the background layer in the Mars image (see figure 11.4).
3. The original spaceship image is a higher resolution than the Mars image, so the spaceship is disproportionately large. From the Image menu, choose Scale, and drag the bounding box in from the corners to resize the ship (see figure 11.5).

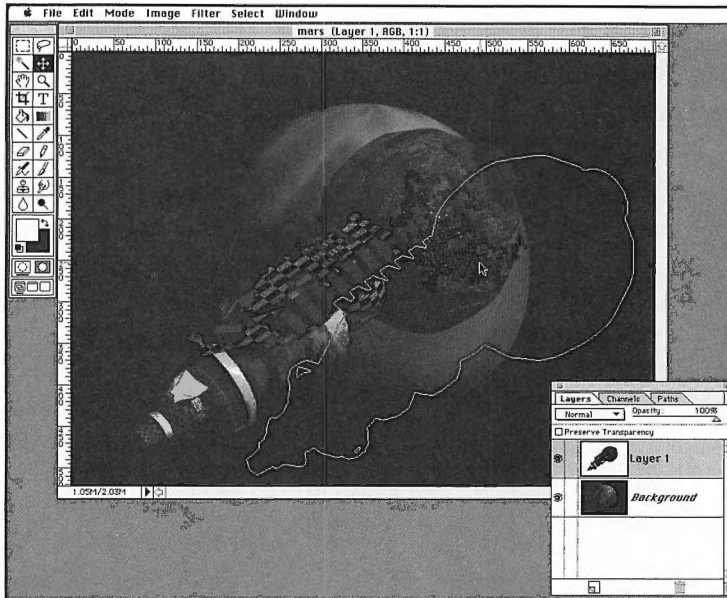


**Figure 11.4** Dragging the spaceship layer onto the Mars background



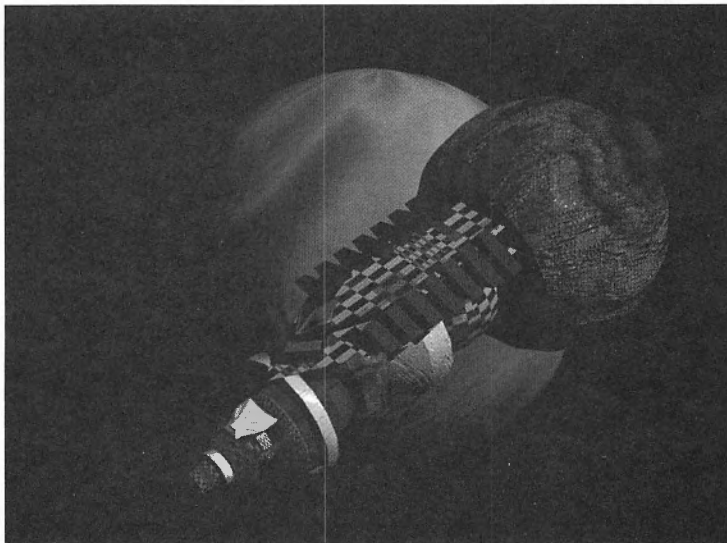
**Figure 11.5** Resizing the ship layer to fit into the Mars image

4. You can now easily drag the spaceship layer around in the image and position it by simply dragging on it with the mouse as in figure 11.6. (In earlier versions of Photoshop, doing this would have cut a "hole" in the image.)



**Figure 11.6** *Dragging the spaceship around to reposition it*

5. Once the layers are in their final positions, you can save the Photoshop document with layers intact, or you can “flatten” the document (which merges all of the layers together), to save it as a PICT, TIFF, or other standard image format. Note that the antialiasing around the edges of the ship is still as perfect as when the image was first rendered (see figure 11.7).



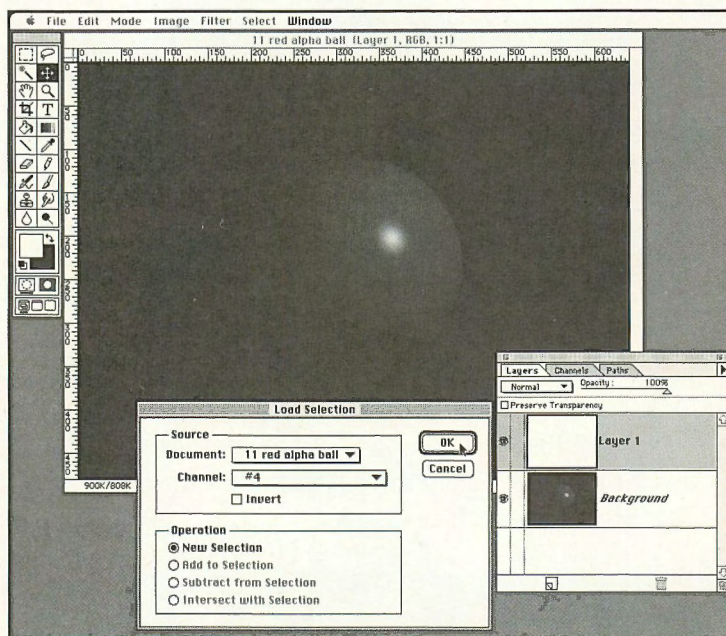
**Figure 11.7** *The final image is still perfectly antialiased*



## Faking a Shadow

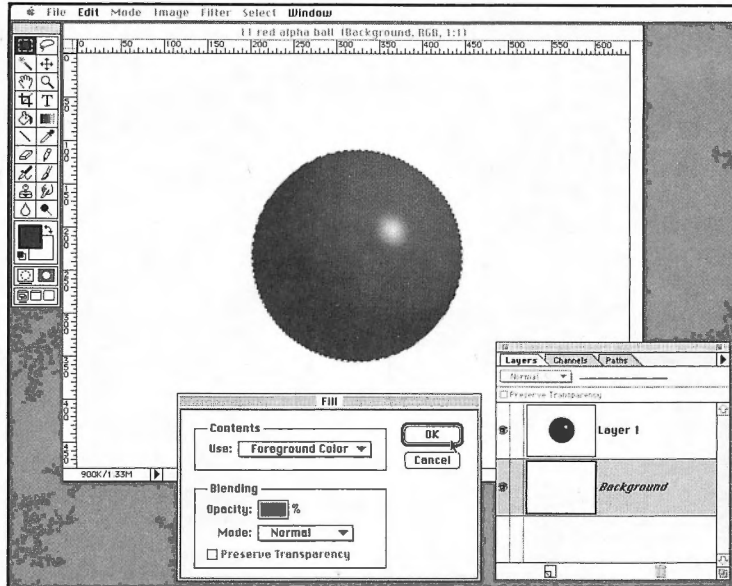
The following technique is an example of how you can use the alpha channel and Photoshop in combination to create unusual effects, in this case, faking a soft shadow behind a rendered sphere.

1. Begin by opening a 3-D rendering with an alpha channel in Photoshop (see figure 11.8). Choose Load Selection from the Selection menu to put a marquis around the sphere and choose Cut. Create a New Layer in the Layers palette.

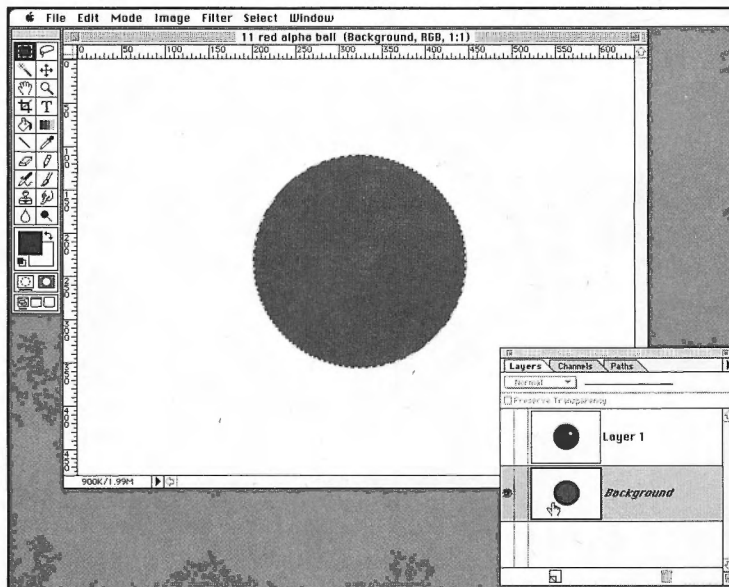


**Figure 11.8** A rendering of a sphere, opened in Photoshop

2. Activate the background layer, select the entire image, and fill it with white. Then set the fill color to black, choose Load Selection again, and fill the selection with 80 percent black. You won't see the effect yet because it's on the background layer behind the red sphere (see figure 11.9).
3. Click the hide/show (eyeball) button next to the red sphere's layer to hide it and see the new shadow shape (see figure 11.10).

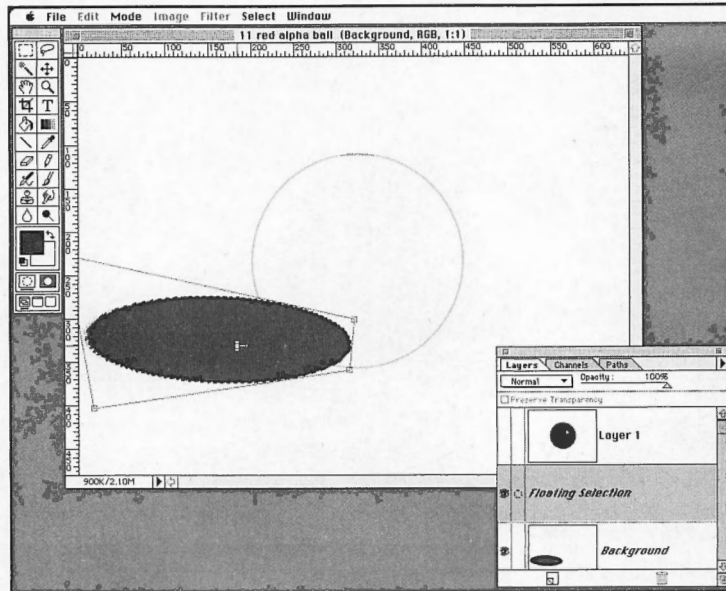


**Figure 11.9** Filling the selection in the background with 80 percent black



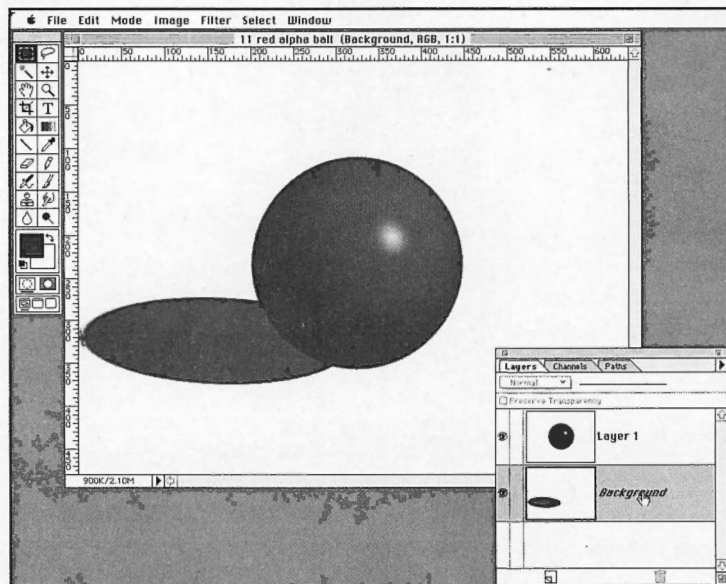
**Figure 11.10** Hiding the foreground layer

4. Use the Skew and Perspective tools in the Image/Effects menu to stretch the shadow over to the side and put it into perspective (see figure 11.11).



**Figure 11.11** *Skewing the shadow layer to one side*

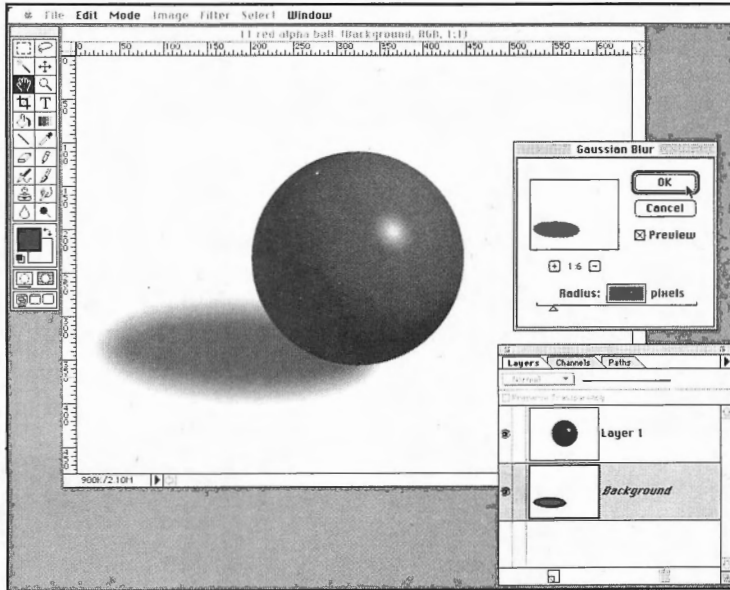
5. Unhide the foreground layer to see the results. Note that you can move either of the layers around to position them the way you want. You also can make any editing changes to either layer (see figure 11.12).



**Figure 11.12** *Viewing all of the layers*

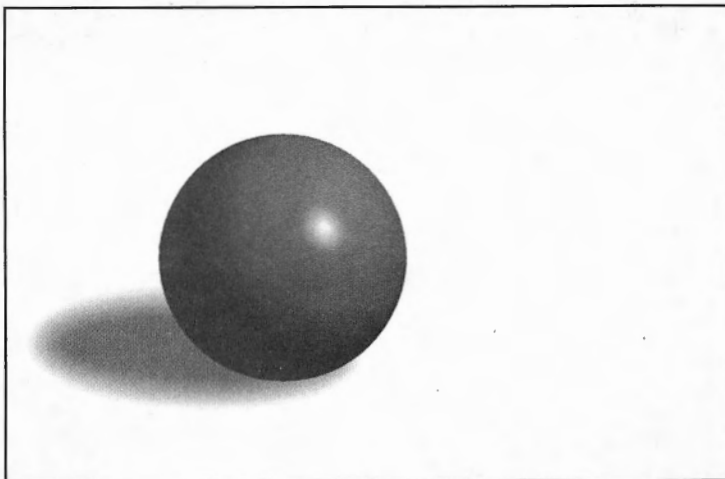


6. Select the background layer and choose Blur/Gaussian Blur from the Filters menu. Apply a large amount of blur to create a realistic looking shadow (see figure 11.13).



**Figure 11.13** *Blurring the shadow*

7. Shadows created in this way can be more realistic than the hard-edged shadows produced by many 3-D renderers (see figure 11.14).

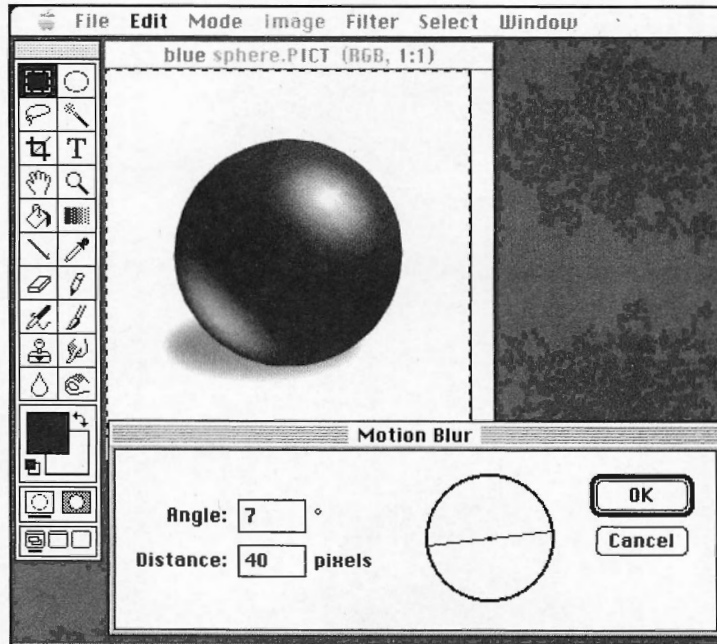


**Figure 11.14** *The finished image with a realistic soft shadow*

## Motion Blur

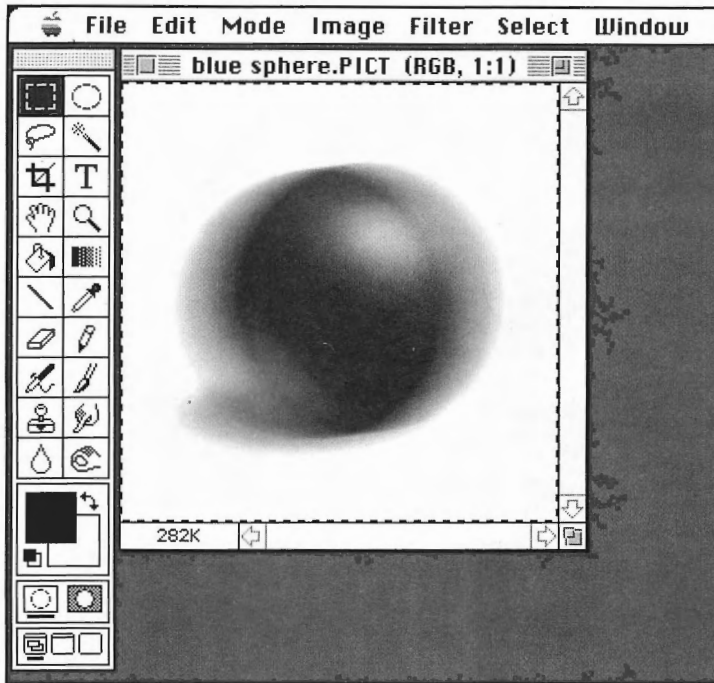
Animations and still images gain a great deal of realism from the addition of *motion blur*. With the Motion Blur filter (under the Blur submenu in the Filters menu) you can easily create this effect in Photoshop (this example uses Photoshop 2.5, but it's equally easy in version 3.0). While the Motion Blur filter is effective by itself, you can create really interesting images by pasting a sharp version of the object at the end of the blur trail, then blurring this slightly. This creates the effect of a photograph exposed with a flash at the end of the exposure.

1. Begin by opening a rendering with an alpha channel.
2. Choose Load selection #4 from the Selection menu and Copy the image.
3. Choose Select All and apply the Blur/Motion blur from the Filter menu. Set a direction by dragging the line in the wheel, and set the amount to 40 pixels (see figure 11.15). The results are shown in figure 11.16.

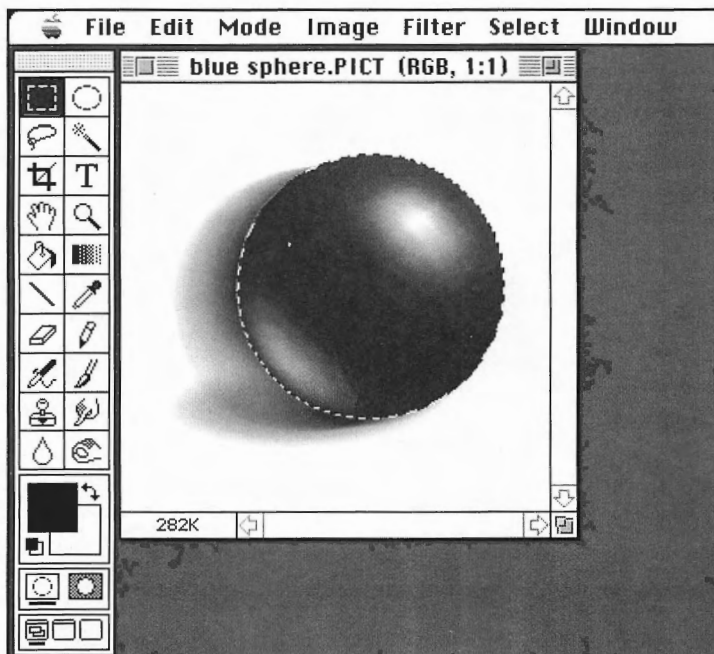


**Figure 11.15** The Motion Blur dialog box

4. Choose Paste and drag the pasted object to one end of the blur trail (see figure 11.17). Click on the background to lock the selection into place.

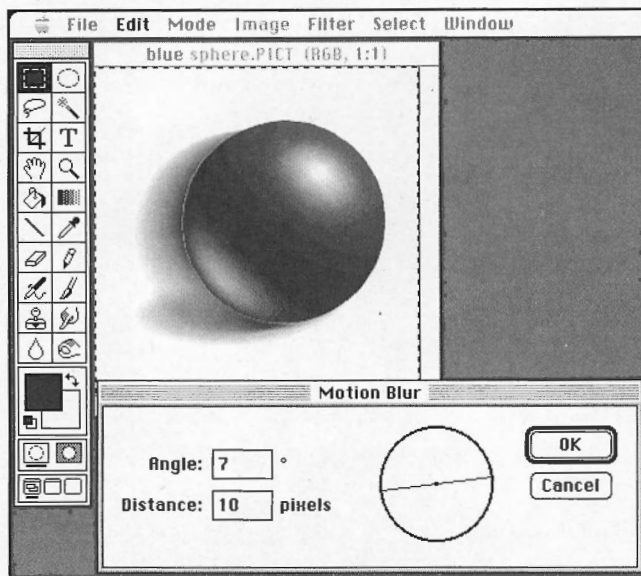


*Figure 11.16 The results of the motion blur*



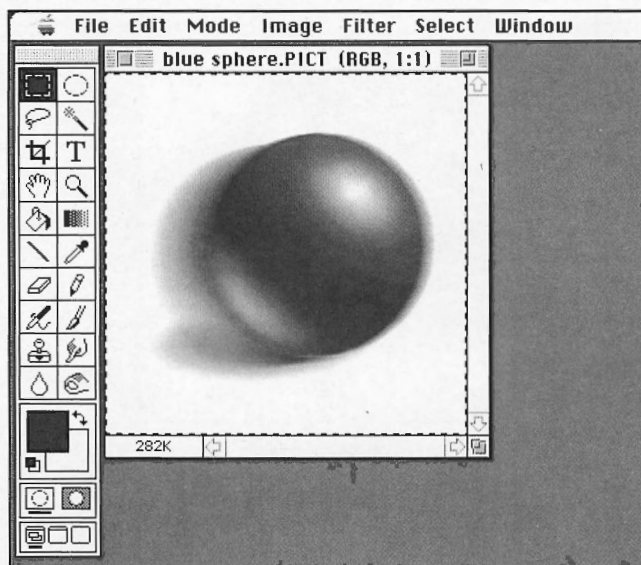
*Figure 11.17 Pasting the object on one end of the blur*

5. Select All and choose Motion Blur again. This time set the amount to 10 pixels (see figure 11.18).



**Figure 11.18** Applying the motion blur a second time

6. The result is shown in figure 11.19. Note that the first blur creates a streaking tail; the second blur gives the foreground object's surface a more subtle movement.



**Figure 11.19** The completed motion blur

## ColorIt!



For 3-D users who want access to Kai's Power Tools and other must-have texture creation and image processing filters, as well as a good selection of painting and image editing tools, ColorIt! from MicroFrontier, is a great low-priced solution. It's so good, in fact, that many users who only need an image editing program for occasional texture making may never think twice about buying Photoshop. I've included a demo version of it on the CD-ROM so you can compare for yourself. You can use just about any Photoshop-compatible plug-in with the program, so it can even drive your scanner. The current version of ColorIt! doesn't support alpha channels, but MicroFrontier has promised that a future version will.

## Painter

Fractal Design's Painter is every bit as unique as Photoshop in its own ways. It does support alpha channels, so you can easily do image compositing in the program, and it supports third-party plug-ins, like Kai's Power Tools, so it's an outstanding program for creating textures and backgrounds. What really sets Painter apart, however, is its "natural media" painting capability (see figure 11.20). Painter offers a wealth of painting tools that closely mimic real-world art supplies. You can paint with oils, watercolors, and acrylics; you can draw with charcoals, pastels, crayons, felt pens, ink pens, and pencils, to name a few. Another unique capability of Painter is its Cloning feature that enables you to paint over an existing image, pulling colors from the original, but creating strokes with the brushes in hand. You also can get colors from a regular palette. You can even automatically adopt the painting style of masters. This makes it audaciously easy to turn a rendered 3-D haystack into an "automatic Monet."

While it's possible to create some really strange fakes—what if Van Gogh had painted spaceships?—the possibilities for elegant architectural renderings and presentation illustrations, for example, are almost endless.

Painter 3.0 introduced two features that are a boon to 3-D users—animators in particular. The first is the "Image Hose." This tool enables you to create a library of floating selections and then paint randomly using the selections as "stamps." Since the tool is very fast to use, it's easy to "paint" amazing textures using any manner of bitmapped objects. For example, you could create a library of feathers, then paint feathers onto a surface. As long as your strokes are left-to-right, the feathers can be laid down left-to-right, for a very realistic effect. By "recording" the painter strokes as you work, you could then substitute a different library, say, grayscale feathers, to create a bump map to go with your color map.



**Figure 11.20** Image rendered in Sketch! and “cloned” in Painter with pastel brushes

The second feature is the capability to paint over frames of QuickTime movies. You can use QuickTime frames any way you’d normally work with images in Painter, so you can easily create “painterly” versions of your animations, or you can paint animated textures for rotoscoping in 3-D.

## DeBabelizer

If Photoshop is the Swiss Army Knife for still images, then Equilibrium’s DeBabelizer is the Cuisinart for images and animations (see figure 11.21). While it lacks painting tools, and many of Photoshop’s pre-press tools, DeBabelizer has a wealth of indispensable power for the 3-D user. Among other things, it supports third-party Photoshop plug-ins, so filters and special-effects tools designed to enhance Photoshop also work in DeBabelizer’s powerful, complex image translation and management environment.

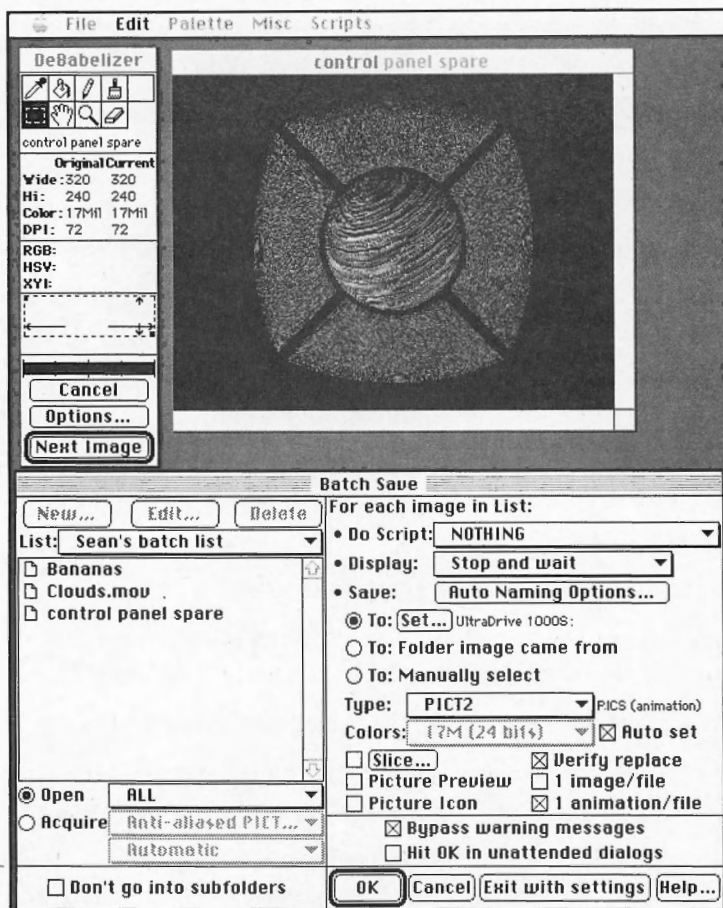
DeBabelizer enables you to convert most image file formats to most other formats. It enables you to adjust the color palettes of images, or to composite one group of images with another. DeBabelizer also has full support for alpha channels (as well as for other kinds of masking).

But, it's very difficult to summarize precisely what makes DeBabelizer good. The following is a sample of how a resourceful 3-D user might put it to work:

- Composite a series of images against a single background or composite one animation over another.
- Convert a folder full of 640 by 480 pixel 24-bit PICS files into a single 320 by 240 pixel QuickTime movie using the Animation compressor.
- Generate a single color "Super palette" based on the dominant colors in all of the images of a 24-bit animation. Dither the animation using this palette to play as well as possible on an 8-bit Mac.
- Legalize the palette of an animation so colors will display correctly on NTSC video.
- Convert a QuickTime movie to an Electric Image Animation format for use as an animated texture map.
- Convert a flying logo to Abekas format for overlaying onto video.
- Use a procedural blend to merge a Kai's Power Tools texture over everything that's a specific shade of green or purple in a series of numbered PICTs.
- Convert Mac QuickTime movies to PC FLI movies.
- Reduce a disk full of differently-sized PICTs, TIFFs, and EPS bitmaps to the same size and resolution and store them in one folder on another disk.
- Replace the white highlights in every frame in a PICS animation with yellow, so Macromedia Director won't turn them transparent during playback.
- Prepare a batch of images as textures along with corresponding bump and transparency maps for use in a 3-D program.

Anything you can do in DeBabelizer can be saved as a script, and scripts can call other scripts, so it's possible to create very complex image processing procedures that are totally automatic. A script can be repeated automatically for every image in a folder or animation, for example. DeBabelizer is AppleScript compatible, so you can create a drop folder, for example, that will automatically process anything dropped into it. You might, for instance, want a script that automatically makes a grayscale copy of every image you throw into a certain folder—in a size and resolution that's compatible with John Knoll's CyberMesh; you could also create a script that automatically composites successive images in folder "A" with successive images in folder "B."





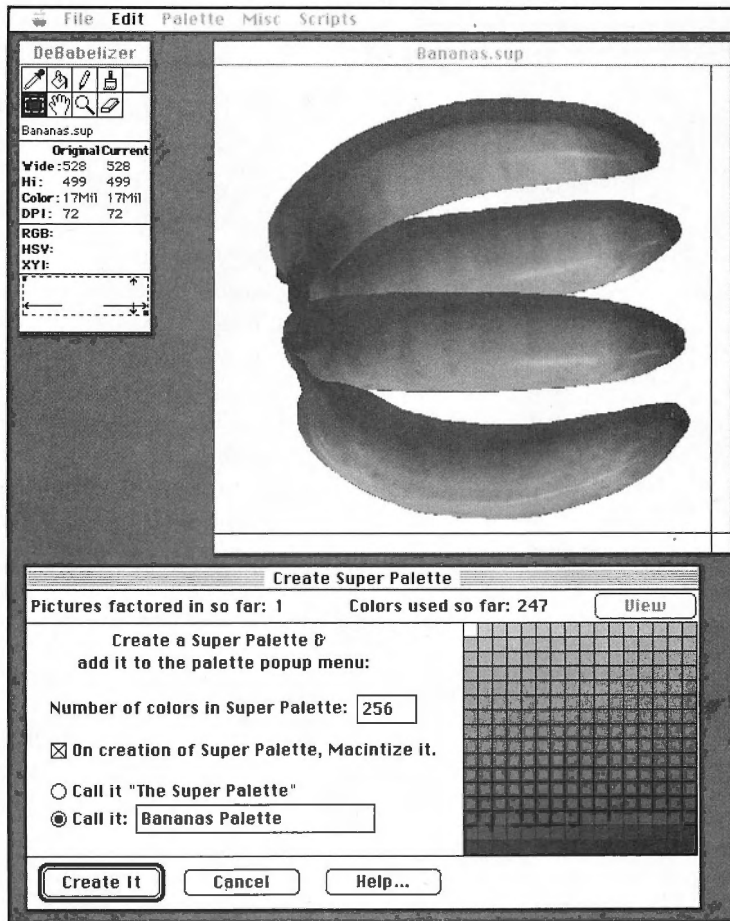
**Figure 11.21** DeBabelizer has powerful batch processing tools capable of doing just about anything to batches of images or animations

Version 1.6 of DeBabelizer introduced animatable filters that enable you to tween the effects of filters over time. For example, you can apply a twirl filter to generate a whirlpool effect in an animation.

There's nothing else that will manage color palettes as effectively, making DeBabelizer particularly invaluable for multimedia developers (see figure 11.22).

In my book, DeBabelizer is a must-have for the 3-D power user. In fact, every one of the 500 or so images in this book began as a color file in any one of a dozen formats and 50 sizes. Each was scaled down to a manageable size, reduced to grayscale, and saved in JPEG format before being shipped off to Hayden Books. All of it was done with a single DeBabelizer script, usually a whole chapter's worth of images at a time.





**Figure 11.22** One of DeBabelizer's most important features is the fast and beautifully-handled batch reduction of color palettes to single palettes

## Desktop Publishing

There are some considerations that will make your life easier when generating images for use in programs like QuarkXPress and Aldus PageMaker. The first consideration is which rendering format you will use. Dimensions and addDepth are naturals because they automatically save in EPS format. This makes it easy to incorporate images into desktop pages—with all of the advantages of PostScript printing.

When rendering bitmap images, however, you'll need to determine what file type is best suited to the work you're doing. EPS bitmap and TIFF are standards.

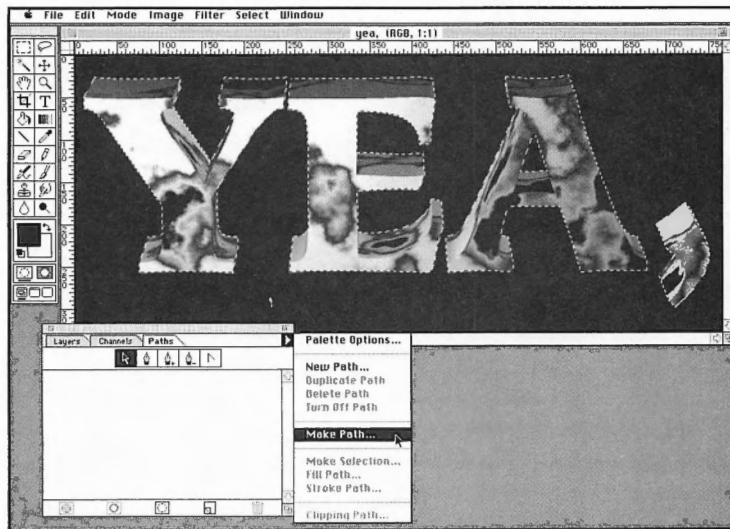
You'll also need to figure out in advance what resolution you'll be printing at; 80 dpi is a common screen resolution for black-and-white work, while color printing often calls for 150 dpi. The standard rule of thumb, when creating halftones for black-and-white or color printing, is that you need to create originals that are twice the resolution of the halftone screen the printer will use. Thus, for a 80 dpi (sometimes called lines per inch or "lpi") halftone, you'll need an original image that's 160 dpi. Similarly, for a 150 dpi screen, you'll need to render a 300 dpi image.

Some programs enable you to specify the resolution of the image to render, so it's easy to specify an 8-inch by 10-inch image at 300 dpi. Many programs, however, only enable you to specify a total number of horizontal and vertical pixels. So for example, you'll have to render a 2,400- by 3,000-pixel image to get 300 dpi at 8 by 10. Keep in mind that doubling the resolution roughly quadruples the required rendering time.

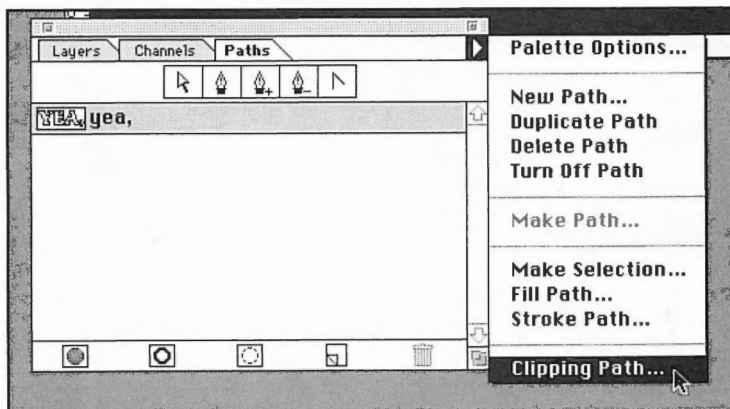
## **Making and Using a Clipping Path**

A clipping path is a feature of EPS images that enables you to save a mask along with an image. This is analogous to the alpha channel, and is used by desktop publishing programs to automatically flow text around graphics. You can use the alpha channel in Photoshop to create a clipping path for a 3-D image. While Photoshop 2.5 did this very easily, Photoshop 3.0 makes it even simpler:

1. Open a 3-D rendering created with an alpha channel. Choose Load Selection to select the 3-D object and select the Paths/Make Path... option in the Layers palette (see figure 11.23). This traces the pixel selection with a pen-drawn path.
2. Again, in the layers palette, choose Clipping Path.... This will prompt you for details in creating a clipping path (see figure 11.24).
3. Save the image as EPS with an 8-bit preview.

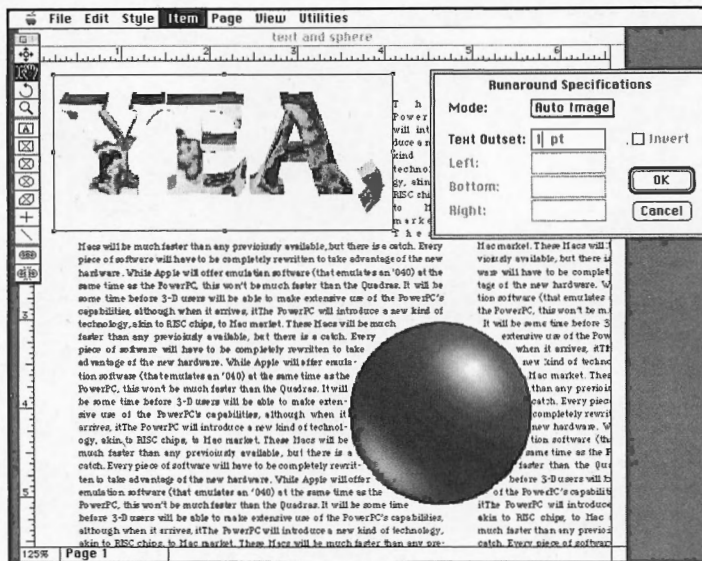


**Figure 11.23** Making a path out of the selection



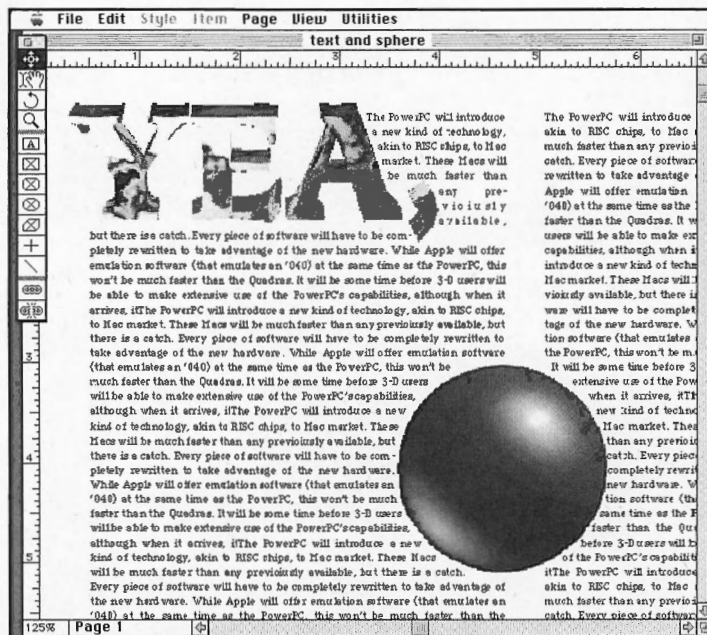
**Figure 11.24** Making the path a clipping path

4. In QuarkXPress, drag a new picture box over an existing block of type and import the EPS image through the File menu. The text is automatically displaced to wrap around the image box (see figure 11.25).



**Figure 11.25** The text automatically wraps around the image box

5. With the image box still selected, choose Runaround from the Item menu. In the pop-up menu choose Auto Image and set the amount to 1 point. The results are shown in figure 11.26.



**Figure 11.26** Text wraps to the shape of the clipping path

## Color Management

If you're producing color separations, consider using the EFIColor System from Electronics For Imaging Inc. It uses sophisticated color profiles for different printers and types of printing, in order to produce the best possible separations. There is a stand-alone version called Cachet, a set of Photoshop calibration tables, or a version included with QuarkXPress.

If you have the right profiles for your particular color proofing device, such as a Tektronix dye sublimation printer, the EFI Color system will print the closest possible approximation to what your final web-offset or other final printing is going to look like.

## Animations

The uses of 3-D animation are endless. Once you have determined the size, resolution, frame rate, and file format, and have actually rendered an animation, there are many applications that can help you put it to use.

## QuickTime

There are now several capable editing programs designed for creating complete movies from QuickTime clips. In addition to the Apple MoviePlayer, which can serve as a rudimentary editor for QuickTime, there are a host of more creatively flexible applications. All of the commercial applications for QuickTime editing and effects support alpha channel compositing with still or moving images. VideoFusion, Premiere, and After Effects also support field rendering for extra-smooth video graphics (described later in this chapter).

- **Adobe Premiere** is the most flexible of the QuickTime editing applications currently available. It offers many styles of transitions and filter effects, as well as titling and animation of video windows. With the proper hardware compression and decompression, such as Radius' VideoVision Studio, it supports real-time full-screen, full-motion playback of video at near-broadcast quality. It's also a very capable program for compositing animations, although it's own animation features are minimal.
- **VideoFusion** is a special effects program aimed at compositing and adding effects to QuickTime animations and video. Its most interesting feature is the pan-zoom-rotate tool, which can spin video around any or all axes simultaneously. This is another viable choice for basic animation compositing. One cool feature is that all of VideoFusion's filters are time-linked, so you can animate them over time.

- **CoSA's After Effects** is very much aimed at the video graphics professional. It provides high-quality sub-pixel animation of video clips and still images as well as unlimited layering using alpha channels. Like the best 3-D animation programs, animation is all event based, so you can animate clips and images along their own timelines and paths. It has incredibly cool special effects to boot. CoSA also offers a low-end version of the product, called AfterImage, which is basically the same, but removes many of the professional, broadcast video tools.

## Video

Video editing is a very specialized discipline requiring lots of expensive hardware. There are, however, easy and inexpensive ways to output video in a manner that can be easily transferred to video by a service bureau.

When creating animations for video, there are some important things you should know. The first is that NTSC video has a very narrow color gamut (the standard was created before cable and other TV breakthroughs). This means that if your animations are using saturated colors, particularly blues, yellows, and reds, you're going to get runny, bleeding, gooey colors all over your screen. The best way to avoid this is with a waveform monitor, but most animators settle for having a TV or NTSC monitor hooked up to an encoder in their Mac, so they can see what they're going to get before making the costly transition to tape.

Once again, DeBabelizer comes to the rescue. It offers a NTSC filter that you can run your animations through to "strip" out-of-gamut colors. Photoshop also can do this, albeit, one image at a time. Either way, the process is almost sickening; it's hard to believe how much better graphics look on a 24-bit Mac monitor.

Another really crucial concept is the "safe" area. While the actual size of NTSC video is 720 by 486 pixels, you never get to see them all. The video that actually gets sent to your screen is only 640 by 480 pixels, but even these are usually not all onscreen. This is partly due to the curve of TV screens at the edges and partly due to the frames that hold the monitor in. Just because your TV doesn't obscure pixels, don't assume that your neighbor's is as cool. Most animators consider that the safe area for action is 580 high by 430 pixels wide (about 20 to 30 pixels in from each edge). That means you should consider anything outside this area off the screen for most viewers. For titles, which are even more critical, the safe area is 520 wide by 380 pixels high; that's about 60 pixels off the top and bottom edges and 50 off of each side. On a standard 72 dpi monitor, it's roughly an inch from each edge, give

or take a few pixels. While this seems like a lot of real estate to surrender, your chances of a disappointing surprise when you get your tape back from the service bureau are significantly reduced if you follow these guidelines.

### Tip

To create a NTSC safe zone for your 3-D application, you can model a three-ring rectangular frame with the unsafe and safe proportions as described above (640 by 480, 580 by 430, and 520 by 380) and group it with your camera so that the 640 by 480 frame is right at the edges of your camera window. This will constantly display what's inside the safe zone and what's not when you're building your scene. This will work even as you animate the camera (but you'll have to move the frame if you zoom the lens). When it comes time to render, hide the frame. This isn't a simple or perfectly accurate solution, particularly if you're using a wide-angle lens (which will screw up all the proportions). But if you really need to know exactly what's safe, and what's not, it works. A far better solution is to have a TV or NTSC monitor hooked to your Mac so that you can tape off the out-of-bounds areas with semi-transparent tape. This way you'll always know exactly what's outside the safe area.

## Frame-by-Frame Recording

Frame-by-frame recording is a multi-step process requiring a frame-accurate video deck that can be controlled via a serial link or a custom connection, a video encoder and software and sometimes hardware, used to control the display and recording of graphics. Essentially, the process works by opening one frame of animation at a time, displaying it (through the encoder) on the video system, and recording subsequent frames on the videotape.

Since the video hardware alone is extremely costly, this work is often done by service bureaus. However, some advancements have come along to make this type of video animation reasonable for small businesses.

## Software Animation Controllers

There are several software-only animation controllers that work in conjunction with a variety of video hardware systems. Popular software programs include:

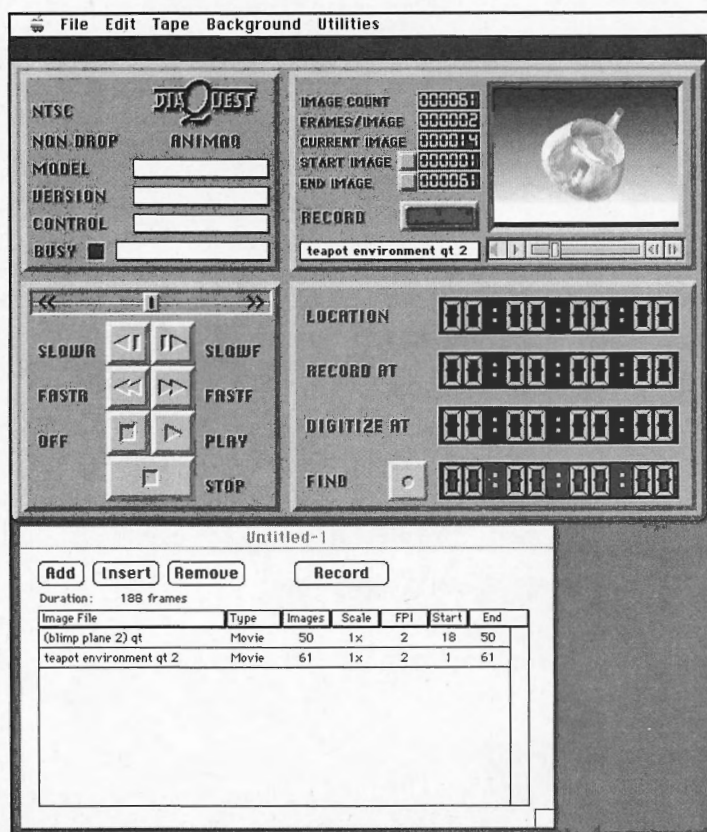
- Auto-PICT QT



- DQ-MAC232
- MacAnimator Plus
- MacVAC

### DQ-MAC232 and EVO-9650

One example of a highly effective, relatively low-cost solution for putting animations on tape is Diaquest Inc's DQ-MAC232 software (see figure 11.27), paired with a low-cost animation deck such as the Sony EVO-9650 hi-8 deck. The video deck is controlled via the Mac's serial port, so no NuBus hardware is required. There are many different video encoders available on the market, ranging in price from a few hundred to a few thousand dollars. Examples include the TruVision NuVista+ and the Radius VideoVision. All of this software and hardware combined is equivalent to the price of a well-loaded graphics Mac.



**Figure 11.27** The DQ-MAC232 interface, showing a QuickTime movie to be recorded; the cue for animations on video is the window at lower left



Using this setup, you load a “play list” in the DQ-Mac software, specifying the frames to record and the starting and stopping points of both the animations and the videotape. When you click the Record button, the software takes over, running the tape deck forward and back, loading and displaying the subsequent frames of video, and hitting the tape deck’s “record” button at the appropriate moments. The software also will automatically composite images with alpha channels, so you can combine a foreground and background animation, for example.

### **DQ-Animaq**

By far the most popular hardware-based animation controller is the DQ-Animaq from Diaquest. This board controls two decks at a time, frame-grabbing from one, compositing those images with an existing animation, and outputting the result to the other. The board and associated software are about the price of a high-end Mac, making this an economical solution for sites willing to spring for the video decks. Optional software provides a capability called QuickPass. This enables the board to grab and record frames of video at full quality and very high speed. It does this, for example, by grabbing every seventh frame starting with frame one. When it reaches the end, it rewinds and grabs every seventh frame starting at frame two, and so-on. This enables the deck to play at full speed and eliminates the delays caused by the deck’s having to “pre-roll” before every frame. It’s also much easier on the expensive hardware.

The software for outputting video to tape is almost indistinguishable from the DQ-MAC232 software.

### **Electric Image Animation System**

Electric Image Animation System (EIAS) has long been able to control video devices through the Diaquest board. But it also offers another option: Saving animations on Exabyte 8-mm backup drives in Abekas backup format.

The Abekas video system is D-1 quality digital video, providing high fidelity without the intricacies or pitfalls of recording to videotape (such as tape stretch). It uses specialized recording and playback hardware and costs about the same as a nice home. Most cities have at least one service bureau with an Abekas system, and all of them are equipped with Exabyte backup drives. Once you’ve saved your animation in this format, you simply carry the Exabyte tape to the service bureau, and for a very reasonable fee, transfer the animation directly into the Abekas system. It’s a completely digital solution, and the result is top quality.

## Exabyte Drivers

Just as EIAS can record its animations to Exabyte tape in Abekas backup format, two general animation packages are available to provide the same functionality to users of other Mac animation software.

- ASDG, a company known for special effects on SGI workstations and other platforms, offers Abekas drivers for the Macintosh and other platforms.
- Knoll Software (also the creator of CyberMesh) offers a driver package called Missing Link that can save images and their alpha channels in a variety of configurations.

Both write Abekas backup format files to Exabyte tape, but they offer a variety of features for compositing graphics, as well.

Top-of-the-line Exabyte drives retail for about \$2,000, although they can be found for much less, and the tapes (identical to 8 mm Hi-8 camcorder cassettes) cost about as much as a premium VHS tape. Exabyte drives are much faster than DAT, so your time is spent rendering, not recording. This is a vastly more economical solution than investing in video hardware, unless you own a service bureau or are outputting animations full time. And it probably makes a lot more sense to pay the small fee every time you lay an animation on tape than it does to go out and buy a professional animation deck. In addition, many animators are swearing off the “old-fashioned” method of outputting animation a frame-at-a-time to video decks. The process is extremely hard on videotapes, resulting in degraded video quality, and it’s also extremely hard on the professional video equipment which can easily run up to \$30,000 per deck.

## Field Rendering

So what is field rendering, anyway? NTSC video is made up of two fields. Each of the two fields is made up of alternating horizontal lines a single pixel thick. If you hold your open hands in front of your face, fingers interlocked, you’ll get a good idea of what fields look like—one hand is the odd field, the other hand is even. Video actually displays these fields in sequence, first the even one, then the odd (depending on whether you’re using NTSC, the U.S. standard, or PAL, the European standard). The result—Oh, foolish eye!—is that we see smoothly flowing motion at a rate of 30 frames per second. We never notice that every other line on the screen goes blank every 30th of a second. We do, however, notice that single-pixel horizontal lines in graphics flicker distractingly, and that the top and bottom edges of shapes seem to jump up and down in a little dance. This is due to this *interlacing* of fields.

**Tip**

When creating graphics for video avoid single-pixel horizontal lines. Also try to avoid very contrasty hard edges at the tops and bottoms of shapes as these will also flicker annoyingly. Keep in mind that diagonal lines, particularly in fine geometric textures and details, will look better than horizontal ones.

Since a frame of video is actually comprised of two fields (for convenience, you can think of these as frame “1a” and frame “1b”), many animation systems are cheating when they record a single image to a frame of video—1a and 1b both get the same image. The visual impact of this is very subtle: graphics are slightly jerky as they move across the screen. This is because we’re actually used to half the frame changing every 60th of a second.

When field rendering, on the other hand, every field receives a new image that has changed subtly since the last. In terms of the smoothness of motion, this effectively increases the frame rate from 30 frames per second to 60. Field rendering has two immediate effects: the first is that you’ll have to render twice as many frames. You would think this means rendering takes twice as long, but if your 3-D software supports field rendering, you’re only rendering half of the lines in each frame. (Since most 3-D programs still don’t support field rendering, however, you will have to render twice as much material to get this effect.) The second effect, of course, is that you’ll need hardware, or a service bureau, that can record to fields. Most service bureaus are equipped to do this, particularly with the Abekas transfer method described previously.

## **Multimedia**

Interactive media, non-linear media, multimedia—call it what you will—the concept is really fairly simple: enabling people to exercise personal control over what they see and hear.

Multimedia enables users to see things they haven’t seen before, and that’s where 3-D comes in. It’s one thing to see a tea kettle whistling frantically on top of the stove. It’s quite another to see it explode, blowing away the walls and windows but leaving the kitchen otherwise unscathed.

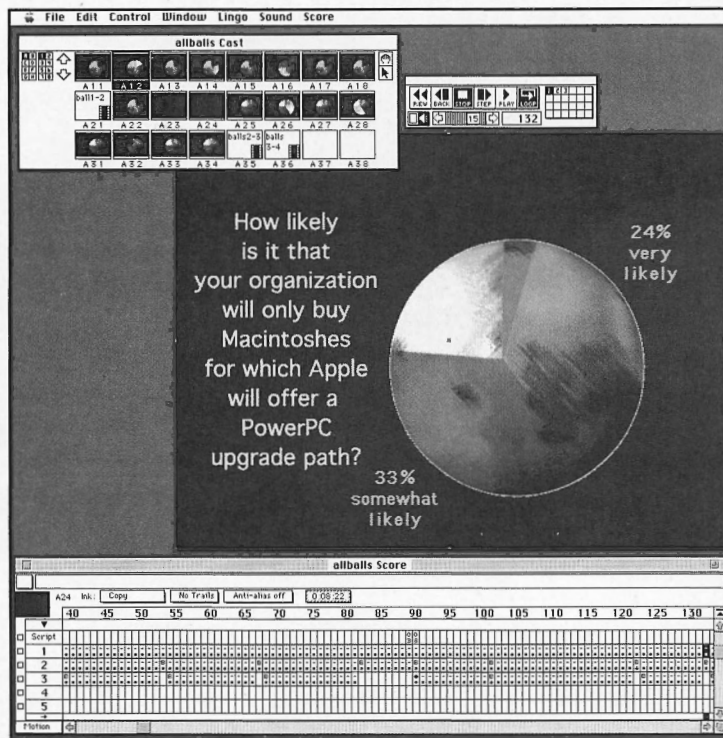
Such scenes are possible using the Mac and 3-D software. The software also makes it possible to create new and unusual interfaces and strange vehicles to transport passengers of your media through unfamiliar worlds.

There are as many ways to incorporate 3-D into media as there are kinds of media. Some of the most common uses today include:

- Interface elements
- Scenery and props.
- Presentations
- Animations

## Authoring Tools

Macromedia Director, despite a programming interface that is hard to love (and even harder to work with) is nevertheless the standard tool of choice for creating interactive media on the Mac (see figure 11.28). However, there are several very good products giving Director a run for the money.



**Figure 11.28** Interactive presentation created in Macromedia Director

For those looking to try something different, there are three really outstanding authoring tools available on the Mac: Passport Producer Professional from Passport Designs Inc., Authorware Professional also from Macromedia, and the Apple Media Tool.

These programs occupy different ends of the spectrum for creating interactive media. Producer Pro offers buttons with a modest amount of interactive functionality, as well as a wonderful time-based approach to playing animation. It costs about as much as a mid-level 3-D package. Producer Pro is specifically created for delivering synchronized media from the Mac. It uses QuickTime and SMPTE time code, so you can, for example, set an animation and a MIDI music file to play on the same beat.

Authorware is the high end of the price spectrum. What you get for the money, however, is an authoring package that goes far beyond Director in terms of user interactivity and tracking of user responses. You also get a programming system that even a 3-D artist can understand. Authorware costs more than the most expensive Mac, but if interactive development is what you do, this program is worth the money. It's a great bargain if you can get it through Macromedia's educational discount program.

Apple has introduced a multimedia authoring tool called the Apple Media Tool that can be extremely powerful in the right hands. It's a fairly simple authoring system for combining media elements, but it exports applications to Script X, which is poised to become the ultimate multimedia programming system. Anything you can produce in the Media Tool can be customized and configured in Script X. The two biggest problems with this approach is that Script X still wasn't shipping when I wrote this, even though it was long overdue, and applications written with Script X are rumored to be voracious RAM users.

## **Color Palettes**

While 3-D software is capable of producing photorealistic animations and graphics, most Macs aren't capable of displaying them. The standard display system is now 8-bit color—paltry compared to the 17 million colors most 3-D software can generate. There is, however, a way to display images and animations on these Macs without the color posterization and banding you'll get if you simply open a 24-bit image on an 8-bit system. The trick is to “dither” the image with a custom color palette.

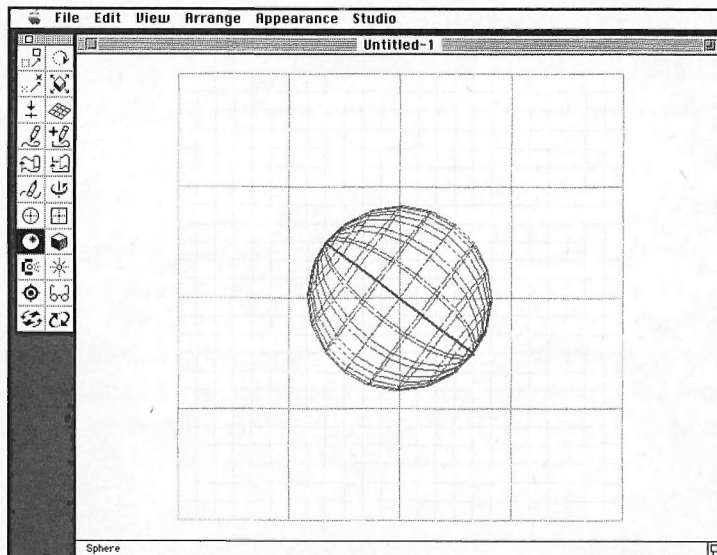
Photoshop enables you to do this through its Index Color command. When you save an image with an Adaptive palette, Photoshop adjusts the palette to reflect the colors in the image.

DeBabelizer has the best approach currently available. It can survey every image in a batch (all of the images you plan to use in a Director presentation, for example), and create a single palette which it uses to dither every image. This means an entire presentation, animation, or whatever, can use the same 256-color palette. This results in great performance gains during playback and eliminates alarming flashes of bizarre color on the screen as the Mac readjusts its system palette to the current image. It also will let you set aside certain colors, such as those used in the interface, as protected, so they'll automatically be part of the palette. (See the "DeBabelizer" section earlier in this chapter.)

## Creating a Console

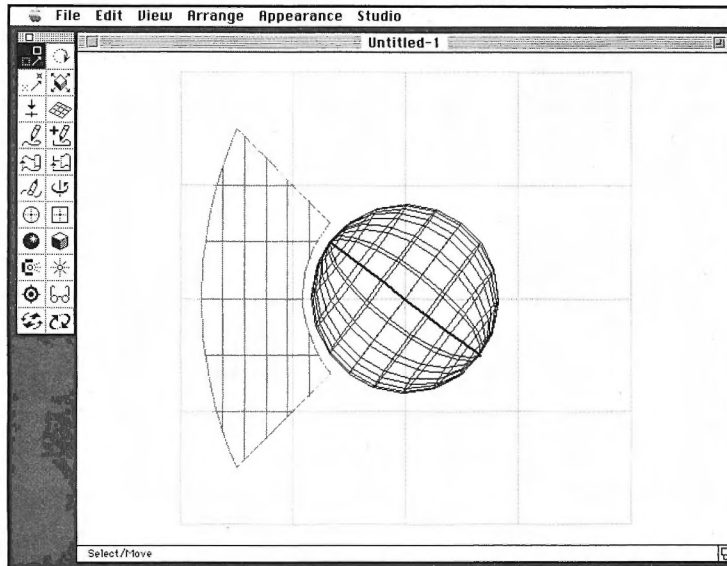
Three-D software makes it very easy to generate buttons and control panels for use in multimedia applications. Buttons are generally very simple shapes, so it's easy to create any button you need with just about any 3-D software. The following example uses Alias Sketch! to create a 3-D trackball.

1. In the Top view, create a sphere at the center of the working plane (see figure 11.29). In the Materials dialog box, give the sphere a texture and bump map.



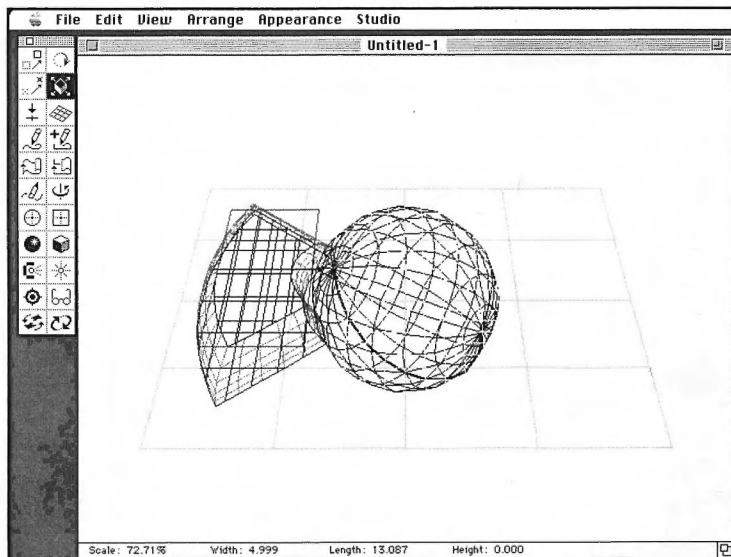
**Figure 11.29** Creating the sphere in Sketch!

2. Draw a 2-D shape that takes up slightly less than one quarter of the space around the sphere (see figure 11.30).



**Figure 11.30** Adding a 2-D shape

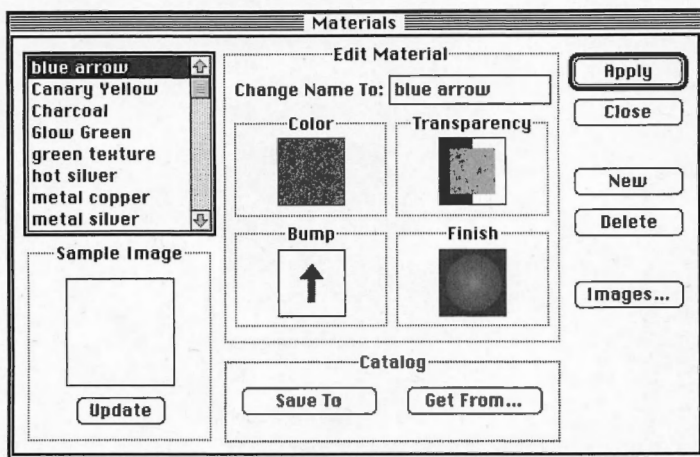
3. Use the Bevel Extrude menu command to give this shape depth (see figure 11.31).



**Figure 11.31** Adding depth to the 2-D shape

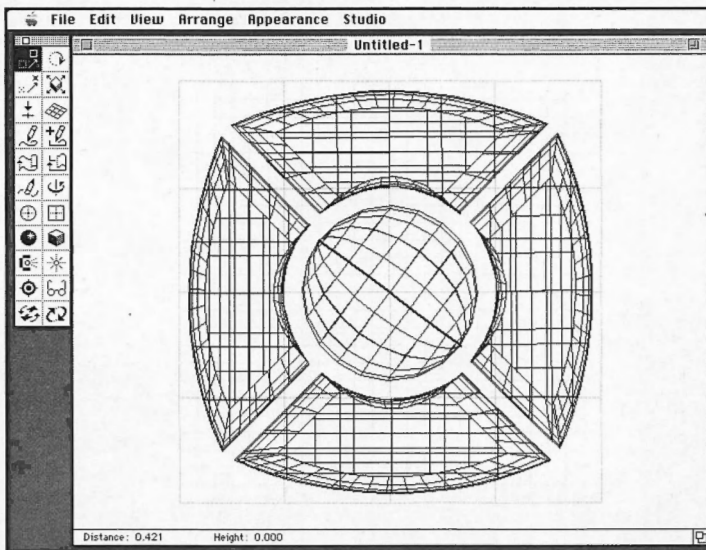


4. In the Materials dialog box, apply a texture and a bump map (see figure 11.32). This bump map uses an arrow. Use the Position Materials command to rotate the bump map into position. Do a test render of just the button to make sure it's positioned properly.



**Figure 11.32** Applying a texture and bump map through the Materials dialog box

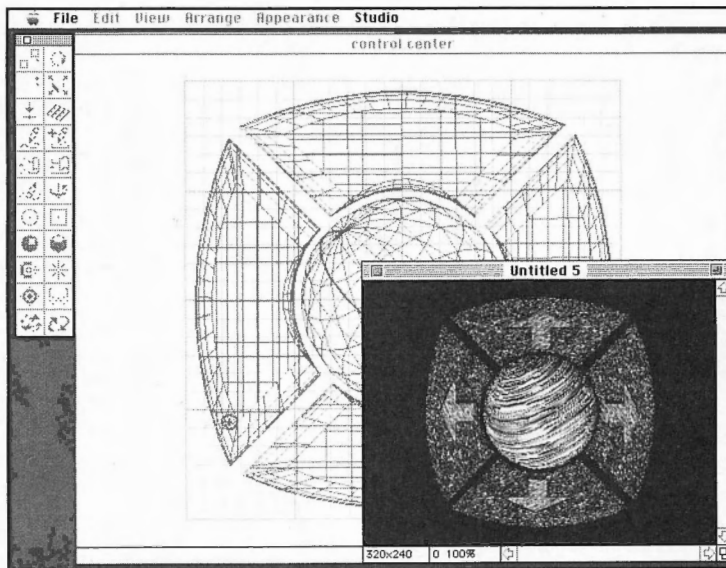
5. Select the button shape you've created and choose Duplicate from the Edit menu. Choose the Rotate tool from the Tool palette and rotate the button object 90 degrees around the center of the sphere. Use the Duplicate command (⌘-D) to repeat the duplication twice more (see figure 11.33).



**Figure 11.33** Duplicating the first button to create the other three buttons



6. Set two lights on opposite sides of the model, far away, with one light slightly weaker than the other. Set the background to black and render the finished control panel (see figure 11.34).
7. For use in multimedia projects, you may find it advantageous to hide all but one piece of the model and render it separately; then repeat this procedure for each piece, in turn. This will enable you to import the separate pieces as distinct objects which will be easy to make into live objects in your authoring program. For Director, and programs that do similar compositing, you may be better off rendering with antialiasing turned off. This makes it possible to perfectly matte the object in Director, and to then antialias it with Director's composite controls. At least render with a black background so you avoid an antialiased "halo" around the rendered object.

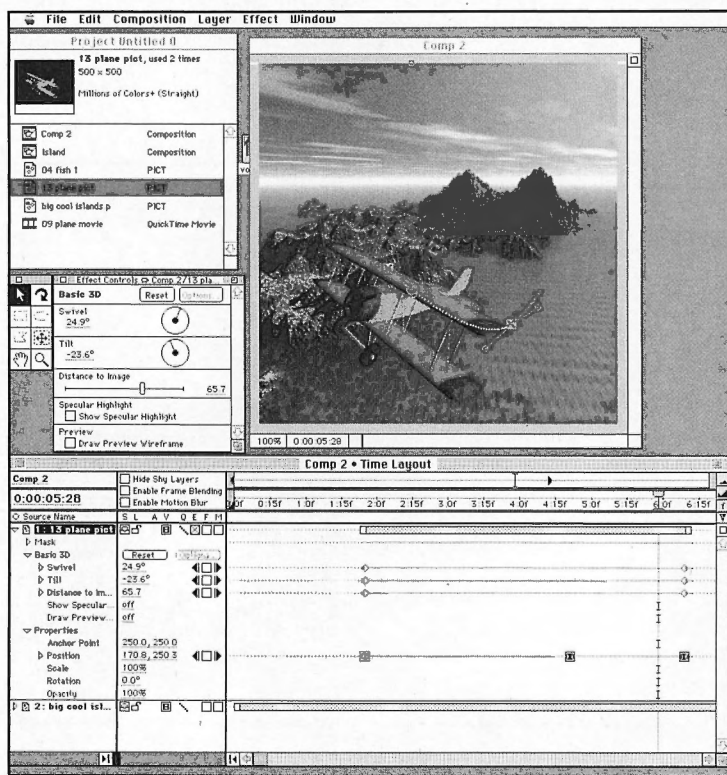


**Figure 11.34** *The final rendered image*

## Animating Still Images

CoSA's After Effects is a remarkable package designed specifically for producing broadcast-quality video animation. You can use it to make the most out of your 3-D renderings. For instance, it can help you by simulating some 3-D animation effects with a single frame of 3-D rendering, like making an object appear to fly out of the background scene. After Effects enables you to perform varieties of pans, zooms, squashes, and more esoteric effects (even 3-D rotation) to any image or QuickTime file (see figure 11.35). You also can apply a wide range of special effects filters to the various layers of an animation.

After Effects provides stunning automatic compositing of images with alpha channels. The program does “sub-pixel” animation and field rendering. *Sub-pixel animation* means that the program works at higher resolution than your screen, so images animate very smoothly. The most recent version of After Effects does event-based spline animation, which means it’s very easy to control the appearance and disappearance of objects, and their motion across your screen. It is definitely one of the best 2-D compositing and animation programs on any personal computer platform, and is quickly becoming a broadcast industry standard.



**Figure 11.35** After Effects can be used to composite multiple layers of graphics and animation almost effortlessly; this airplane rendering, done in StrataStudio Pro, is composited over a KPT Bryce backdrop, using several key events to animate it

**c h a p t e r**

**12**

**RenderMan**

**R**enderMan is unquestionably the most ambitious of all 3-D rendering systems. Not only is it extremely powerful on the Macintosh, but it is available on many other computer systems. RenderMan was conceived as a universal 3-D language. The original idea was to create a 3-D rendering engine that could run on any computer, so that the program could grow at the same pace as computer technology. To some extent, that's exactly what has happened. While many products take advantage of it, RenderMan is not a Macintosh product in and of itself—although a number of Macintosh products are built around it, and many support it in some way.

## The 3-D “Printer”

RenderMan is developed by Pixar (a company created by the alchemists of Lucasfilm) for the depiction and rendering of realistic 3-D images. RenderMan takes input in the form of geometry, lighting descriptions, and “shaders,” and generates rendered images. Unlike other 3-D systems, RenderMan is application-independent, as well as platform-independent. In other words, it acts very much like a PostScript printer—you send it a RIB (RenderMan Interface Bytestream) file, and it processes a picture. Just as PostScript is a language for defining 2-D images, RenderMan is a language that defines 3-D objects and surfaces.

The beauty of the system is that any machine that runs RenderMan can render a RIB file created on another machine. Another unique aspect of RenderMan is that Shaders—the algorithms that define surfaces and how they are rendered—are separate from RenderMan itself. As long as a shader obeys the very flexible rules set down by RenderMan, the renderer should be able to use it. This means that RenderMan has almost open-ended power: it can create images with the realism of ray tracing or radiosity, but is not limited to using one method for an entire scene. Essentially, each object in a scene can have its own rendering system attached in the form of a Shader.

The reason that Industrial Light and Magic (ILM) and other companies that use 3-D extensively are able to put RenderMan to such good use is because it's possible to create an infinite variety of very realistic surfaces and effects by programming custom RenderMan shaders. In addition, RenderMan's device independence means the images can be rendered at extremely high speeds using networks of high-powered computers.

Anyone with a Mac and software that can generate RIB files can theoretically generate the same type of images that Industrial Light and Magic creates for blockbuster movies. Be aware, however, that the computing power at the disposal

of ILM's RenderMan makes the fastest Mac look like a gnat on the knee of a *Brachiosaurus*. While ILM uses RenderMan's core technology, much of the special effects and animation done with RenderMan is created by teams of graphics-wizard programmers.

ILM remains the greatest master of this sorcery. The company has used RenderMan technology to create the world's most recognizable liquid-metal cyborgs, rippling submarine monsters, and *Tyrannosaurus* dinosaurs, among other fantasia.

It's very common for 3-D users to model and build scenes on a Macintosh or other personal computer and to send these scenes to a powerful workstation for rendering. Macintosh users can even use add-on acceleration hardware, such as YARC Systems Inc.'s Zuma or MacRageous boards, which are essentially high-speed RISC computers that plug into your Mac (see chapter 13, "Advanced Topics"). A special version of RenderMan, in this case YARCRenderMan, accepts RIB files created on the Mac and generates renderings at very high speed. Alternatively, you can network the Mac to one or more UNIX machines from Silicon Graphics Inc. or Sun Microsystems (running their own versions of RenderMan) for high-speed rendering.

## **Choosing MacRenderMan**

The version of RenderMan that runs on the Macintosh is called MacRenderMan. It comes with several related applications:

- MacRenderMan is the primary rendering engine—this software takes RIB files and actually generates a picture
- Render Monitor is very much like Apple's Print Monitor—it's a small application that manages the queuing of multiple RIB files in and sends them to be rendered in the background
- RenderApp is a utility that enables you to view and make minor changes to images created by RenderMan
- NetRenderMan is Pixar's optional distributed rendering application, which may or may not be integrated into the version of MacRenderMan that you're using

MacRenderMan is supplied with Pixar's ShowPlace, but in the past has been available as a separate stand-alone product. When you install MacRenderMan, it appears on the Mac as a Chooser device. This means you select RenderMan in the Chooser; then select from among the various flavors of RenderMan that you have available.

These include “VectorRMan,” which renders a wireframe view of the scene, and “PhotoRMan,” the renderer of choice for photorealistic rendering.

When you select a RenderMan renderer, it enables you to choose the default rendering “printer.” This may be the Local version of RenderMan (running on the same machine you’re working on), YARCRenderMan (if you have one or more YARC boards installed), or a network RenderMan server accessed through NetRenderMan.

The system also comes with a program called RenderApp, which works very much like the Print Monitor application supplied with every Mac (through System 7.5). Anything you send out for rendering is processed by the RenderMan “device” and returned to RenderApp as a finished image.

NetRenderMan’s distributed processing means that it can divvy up a single rendering job among several copies of RenderMan located on the network. Each of these rendering slaves renders its portion of the image and sends it back to the rendering Mac for assembly into a single image. This is a classic case of division of labor, and it works great, particularly if you have a high-speed network, like Ethernet. (Distributed rendering is also available for RayDream’s Designer, Strata’s products, Infini-D, Sketch!, and to some extent, Shade III, Sculpt, and Presenter Professional; see chapter 13, “Advanced Topics,” for further explanation.)

## RenderMan Shaders

Shaders are RenderMan procedures used to compute details of a surface to be rendered. For the purposes of most Mac users, shaders (usually denoted by a “.slo” appended to their names) are similar to “textures” in other 3-D programs. Some types of shaders are used behind the scenes, such as light source shaders (which are supplied with MacRenderMan and modified by your scene-building application). The shaders of most interest to end users typically are surface shaders and, sometimes, volume shaders.

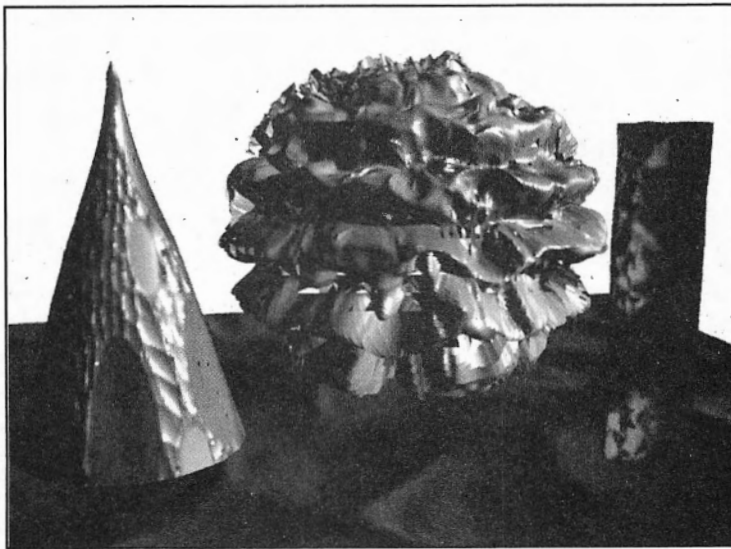
Technically speaking, shaders are program files which specify some of the parameters for a whole range of possible variables that make up a surface quality. Skilled RenderMan users can open a shader, modify it using a word processor, and wield magic on a 3-D scene. Fortunately, most users are insulated from the details of shaders by 3-D software which offers some kind of friendly interface to RenderMan. Usually, you can apply RenderMan textures without understanding the underlying technology.

## Surface Shaders

Surface shaders are the most common shaders you'll have to deal with because they define the color and texture of a surface, much the way procedural shaders (often used for wood and marble) do in other rendering systems. Commercially available shaders from Pixar and the Valis Group can be as simple as "paint," or as complex as "metal bubbles" or "brushed aluminum." In general, these shaders won't look much different than the surface and procedural textures you'll find in other 3-D programs.

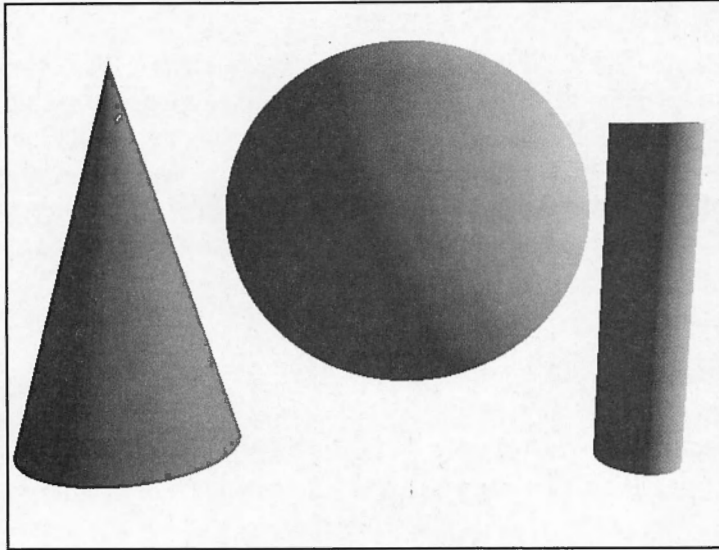
## Displacement Shaders

Surface shaders can include displacement, which handles the deformation of an object's surface and the creation of irregularities. When applied to a surface, you can create realistic pits, ridges, threads, or other displacements of a surface's volume. In essence, these shaders are actually distorting the geometry of a surface as it's being rendered. Figure 12.1 shows a rendering generated with displacement shaders in RenderMan, while figure 12.2 is the same model ray traced with only simple colors applied to the parts of the model. Note that displacement shaders can radically modify a surface's shape, unlike bump maps which simply tweak the model's surface normals prior to rendering.



**Figure 12.1** RenderMan rendering generated in Macromedia 3-D using a variety of displacement shaders, including the Valis Group's Alien Brain and Dragon Skin





**Figure 12.2** *Phong rendering of the same model seen in figure 12.1*

Displacement shaders can include materials such as glass, wood, metal, or exotic substances such as alien brains. For example, a brick displacement may actually modify the surface of an object so that it rises at the brick's surface and falls inward at the mortar between bricks.

## Volume Shaders

Volume Shaders actually occupy a volume in space. Things like clouds, mist, smoke, and glowing lights can all be created with this type of shader. One of the variables of a volume shader is density, so you can specify the thickness and lumpiness of your volume. Imagine how this works with clouds that can be either thin and wispy, white and puffy, or dark and threatening. Another important effect of volume shaders is that they can be used for boolean operations. As with displacement shaders, which actually modify the surface shape of an object, volume shaders can be subtracted from one another—to create very accurate holes, for example. Currently, only Typestry supports RenderMan's smoke- and cloud-type shader parameters.

## Looks

Shaders are an extremely powerful format, and a skilled programmer can create all kinds of unusual surfaces. However, RenderMan shaders have traditionally been mysterious to casual or beginning users. Their power lies in the capability to configure the many parameters that determine an object's surface appearance, but most



3-D users lack the sophistication, time, or patience to program shaders. Some programs enable you to set these parameters without manually typing in program code, but programs such as Presenter Professional and MacroMedia Three-D still require you to set values for characteristics such as “X(float),” “UpVector,” and “AtVector” to get the most out of RenderMan shaders. Needless to say, most designers don’t immediately perceive the value of “floats” or the ups and downs of AtVectors.

Fortunately for novice users, Pixar offers an easy-to-use shader format called Looks that comes with both Typestry and ShowPlace. In addition, all of the Valis Group’s shaders have been updated to this format. (Pixar sells a “locked” CD-ROM containing the total Pixar and Valis library of Looks.)

The Looks format makes it much easier for a 3-D user to use textures. Unlike other forms of RenderMan shaders, Looks includes a visual preview of each texture. This means that you can see a sample of the material you’re applying to an object before you render the scene. This is not as helpful as a real-time preview of an object with the texture applied, but it’s better than pure guesswork. In addition, most of the useful parameters in Looks are adjustable via sliders (so you don’t have to know what numbers to type in). Macromedia’s new 3-D application will probably be the first to support Looks, although Pixar has said it will offer a development kit for other vendors to do this as well. This means that you’ll be able to preview surfaces and other RenderMan effects, at least to some extent, from within your scene-building application.

Currently, however, the non-Pixar applications that support RenderMan (such as Presenter Professional, Macromedia Three-D, and Sketch!) do not have any support for the Looks format, so you’ll have to take the well-traveled, but more confusing, “.slo” road.

## **Looks vs. Shaders**

The fact that RenderMan is supposed to be a standardized rendering language, and that Shaders are supposed to be the “words” one uses to speak its dialect, makes Looks something of a sore subject. To date, only Pixar’s own software products support Looks. This is primarily because Pixar has never opened up the Looks format to other developers through an application programming interface (API). It’s easy to conjecture that Pixar will take its Typestry animation tools into its more complete, but motionless Showplace, and therefore doesn’t want to share the Looks technology with “competitors.” On the other hand, there are a number of really good 3-D animation programs that currently support RenderMan Shaders, that would greatly enhance the popularity of MacRenderMan if only an easy way into its surfaces (i.e., Looks) existed.

Late summer *MacWEEK* reports said that Macromedia's "Extreme 3D" would support Looks directly, and postings on America Online bear out the deduction that Pixar will, in fact, offer an API for 3-D software vendors. I guess only time will tell where to look for "Looks" support.

While Looks are the tool of choice for current versions of Typestry and Showplace, other 3-D programs support RenderMan through Shaders. Presenter Professional and Macromedia Three-D have long vied to have the best RenderMan shaders interface on the Mac.

### **Animated Shaders**

Presenter Pro 3.0 enables you to animate the parameters of shaders over time, so that you can create moving, undulating, lifelike surfaces. It's possible, for example, to make a monster's brain "breathe," or to create a sea anemone that appears to "wave" in underwater currents. As with any other animation settings, you do this by applying a shader at one event mark, then changing its values at subsequent events. With its very detailed level of animation support, it would be quite surprising if Macromedia's next 3-D offering didn't enable you to animate shaders over time, as well.

#### **Tip**

The following Mac programs work with RenderMan with various levels of integration.

Render directly to RenderMan and support Looks shader format:

- Typestry
- Showplace
- Macromedia "Extreme 3D" (rumored)

Render directly to RenderMan, using ".slo" shaders:

- Presenter Professional
- Macromedia Three-D

- PixelPutty Solo
- ZOOM

Output RIB files for rendering with Showplace:

- Rend•X (a plug-in for Strata's products)
- Sketch!
- form•Z
- Shade III

## Glimpse

Looks are actually divided into two parts in the current version: Looks Masters and Looks Instances. A master contains all of the default settings for a particular Look. Since you'd probably be unhappy if you edited a Look, didn't like what you got, and couldn't figure out how to edit it back again, Pixar has made the Looks Masters uneditable. However, the Glimpse utility, which comes with Showplace and Typestry, enables you to knock off as many different Instances of the Master as you like. For example, you may want to change the grain colors in a wood shader or modify the degree of refraction in a glass texture. Glimpse enables you to do this, as well as to make many other modifications to existing Looks.

Glimpse essentially enables you to modify individual parameters of a shader through a graphical interface. This format protects the user from having to program shaders or truly understand the workings of all of a shader's parameters. You are free to experiment to see how changes will affect your designs. Glimpse also enables a designer to simply make selections such as choosing a color from a color picker or specifying an amount of lumpiness with a slider bar (rather than specifying a color number or typing in a displacement value). Once you have defined an Instance, it is saved and can be applied to an object from within Typestry or Showplace.

## Programming Shaders

For those who want to go beyond what's available in commercial and online libraries of RenderMan Shaders and Looks, it's possible to write your own shaders. This technical and complex undertaking requires detailed knowledge of

the RenderMan interface, an understanding of surface qualities, and a practical grasp of computer programming in a language such as C. In short, it's beyond the scope of this book, except to say that with some effort, shader programming can yield wonderful results. It's possible to create shaders that will generate almost anything you can see.

What is possible and what is practical don't always coincide, however. The single-purpose renderers that come with most 3-D applications (at least on the Mac) have the advantage of being complete and easy to use. RenderMan, being a rendering language rather than a specific application, is relatively unlimited, but mastering the language is not for the casual renderer.

**Tip**

The author encourages anyone interested in programming Shaders—or just learning a lot more about RenderMan—to read the *RenderMan Companion* (Steve Upstill, Addison Wesley, 1990). No author could hope to come close to documenting this system better than this authoritative textbook. And, for those who can handle the frequent intrusion of mathematical formulae and program code, it's a very interesting read, as well.

## The Valis Group

The Valis Group is a company that has had on-again, off-again ties with Pixar since its inception. While it started out doing RenderMan shaders, the Valis Group also now offers a really great 2-D morphing and deformation package, “Flo” and it's big brother “MetaFlo,” as well as a spline modeler called PixelPutty (both “Solo” and Showplace plug-in versions). Valis provides the best commercially available shaders for Mac users of RenderMan—they are generally far more creative and interesting than the ones to come out of Pixar. Fortunately, users of Pixar's products Showplace and Typestry can also purchase Valis' shaders in Looks format. (Pixar sells them on its Looks CD-ROM.)

## Shader Toolkit

Valis offers a Shader toolkit that is very similar in style and function to Glimpse. It actually lets you go deeper under the hood than Glimpse does with Looks, but has the disadvantage of not providing the same level of feedback or quite the user-friendly interface (which was the whole point of Looks/Glimpse). On the other hand, if you're working with programs that work with Shaders, it's an indispensable part of your own toolkit.

**C h a p t e r**

**13**

**Advanced  
Topics**

**W**hile some people may consider the very act of creating a 3-D image an “advanced topic,” it is nevertheless sometimes necessary to go beyond the day-to-day basic requirements of modeling and rendering. This chapter attempts to shed light on some of the most important techniques used to take 3-D to a higher level: translating and transporting models between applications; smoothing the facets of polygonal models; coping with the many file formats endemic to 3-D; and accelerating rendering through special hardware and software and distributed processing. For those looking into the future, I’ve outlined where Apple seems to be leading us with 3-D technology.

## Transporting Models

Because no 3-D tool is perfect, many users take advantage of multiple applications. Architects, for example, often work in a full-featured CAD application when designing structures, but export their models to a program such as StrataVision 3d for photorealistic ray tracing. Animators, who rely on programs like Electric Image Animation System and Macromedia Three-D, must use a separate application to build their models. Moving models between applications is by no means as simple as it is in the 2-D world, where file formats like EPSF, QuickTime, and PICT guarantee that graphics files are compatible from one application to the next. The list of available file formats for 3-D is almost as long as the list of applications themselves. And some programs greatly complicate matters by supporting some, but not all, of a file format’s features.

With recent releases of 3-D software, some measure of help has arrived. MiniCAD+5, for example, exports native StrataVision 3d-format files, so rendering a MiniCAD model is as simple as saving in one program and opening in the other. Modeler Professional and form•Z both export Electric Image’s FACT format directly. MacroModel supports RIB output, as does Sketch!, but you will need Pixar’s ShowPlace to apply lights and shaders. Working Model, a 2-D physics simulation package, exports its inverse kinematics animations directly to Macromedia Three-D.

## 3-D File Formats

Often things are not so rosy with most applications. In general, most applications rely on a few key file formats, including DXF, RIB, and IGES. (Electric Image Animation System’s Transporter lists about 20 file formats in its import menu.) Usually, the support for these standards is somewhat less than “standard.” Often a subset of the possible formats is supported. Conversely, a program may import a particular file format, but not all of the included objects will make it in translation. It is

common, for example, to import a complex object from another program, only to find that it renders as “inside-out.” Other problems include holes in surfaces that disappear when moved to another application, or groups of objects that are combined into single inseparable objects when imported. There are many tricks for dealing with these problems, but there are no standard techniques. In cases where a file does not convert accurately, you may need to modify it in the original application to make it work when exported. Many programs offer a variety of export settings that affect your ability to export a model.

## DXF

The Drawing eXchange Format (DXF) is the AutoCAD standard file format developed by Autodesk, Inc. Originally a 2-D format (2D DXF), it has been upgraded to include the 3D DXF specification (which is the only one that really matters for 3-D applications). Often, the 3D DXF format is referred to as simply “DXF;” however, it comes in a variety of “flavors” that correspond to releases of AutoCAD. DXF 12 is the most current version. If your program imports DXF 12, it should handle all the earlier versions without too much trouble, if it exports DXF 12, programs that only read earlier versions may have trouble with it.

DXF is the most widely supported file format by far. Its strengths are in defining and exchanging polygonal (faceted) geometry. This makes it a natural for exchanging architectural drawings and other polygonal primitives, but it is weak at defining spline-based surfaces and other curves (except for circles and arcs). Most animation programs are currently polygon-based (including Electric Image, Macromedia Three-D, Infini-D, and Strata StudioPro); therefore DXF files don’t pose a problem.

DXF has become a de-facto standard for exchanging geometry from one program to another, but the way programs support the format is erratic, to say the least. There are many possible options for saving and reading DXF, such as the number of facets in polygons (3, 4, or more) and whether lines should be exported as lines or as the edges of faces.

This is particularly difficult when you’re dealing with a spline-based program like Sketch!. DXF is poor at communicating curves, so in Sketch!, for example, you can manually convert your models to polygonal shapes before you export DXF.

DXF files tend to be much larger than equivalent spline-based models. DXF is a more-or-less reliable format, however, so it is widely supported and used. DXF can carry color and layer information that you can use to transport object groupings intact. It does not, however, support textures, lights, animation, linking, or other elements of a 3-D scene.



## IGES

The Initial Graphic Exchange Specifications (IGES) standard defines far more types of objects than most programs support. That makes it a good format for importing documents (if you can read all of the possible features in an IGES file, you can import just about anything). But exporting a file to IGES is by no means a guarantee that another 3-D application can read it.

Unlike DXF, which is widely supported in both directions, IGES is widely supported as an export format, but few programs on the Mac offer the option of importing an IGES file. In order to import IGES, a program would have to support a vast array of polygonal and NuRBS data—which is more than most developers are willing to commit to.

## RIB

Interface Bytestream (RIB) files are the standard medium of exchange for RenderMan geometry, lights, and shader information. One big advantage of this format is that it has integral support for splines. Because RenderMan is available on most platforms, it is a natural for distributed rendering and general file exchange for users of RenderMan-compatible products. On the other hand, because the file format includes the specifications for shaders, it is difficult and therefore unpopular to implement in applications that are not aimed at using RenderMan. Many programs support RIB output, but only Showplace currently imports a RIB file intact.

## Holding Your N-Gons

Polygonal 3-D systems and polygonal file formats deal with a pretty limited variety of surfaces. Next to the polyline (a connected sequence of straight lines in space), the most basic of these surfaces is the polygon. Some programs will only deal with triangles (stereolithography and early versions of Sculpt, for example). Triangles are the only polygons that are always flat planes in space. Meanwhile, many 3-D programs enable you to have 4-sided (or more-sided) polygons, which aren't necessarily flat planes. These are more efficient than triangles, because, for example, you can identify a square surface with a single polygon, instead of the two polygons required to describe the surface with triangles. If you're going from a program that supports 4- or more-sided polygons to one that demands triangles, you'll have to find a way to tessellate the surfaces either at export or import. One advantage of triangles is that



they provide the smoothest rendering in polygonal systems. In fact, Strata's products tessellate 4-sided or larger polygons to triangles when you apply the Smooth command. Sketch! enables you to convert surfaces to triangles in the Convert to Polygons dialog and form•Z offers very fine control over polygonal export. Programs that are frequently called upon to export or import DXF usually have some level of control over the shape of polygons.

## **Converting Polygons to Splines**

There are actually two programs on the Mac that enable you to convert polygonal models to splines. The first is Animation Master. When you import a DXF file, it places a series of spline control points to correspond to the vertices of the original model. This means you have to use very small, non-detailed DXF models, unless you want to end up with a massive, unusable shape.

Sculpt 4.0, on the other hand, uses intelligent curve fitting to approximate the shape of a polygonal model with splines. This is a very efficient and elegant approach, and it results in the ultimate "smoothing" of polygonal geometry.

## **Kandu's Cadmover**

Cadmover, from Kandu Software Corp., is a commercial software application specifically created to translate a 3-D file to another format. Because this is its specialty, Cadmover tends to be very good at what it does. There are many possible scenarios for moving files from one program to another that simply aren't handled by the software's regular import/export formats and capabilities. Cadmover can generally overcome these obstacles by acting as an intermediary. Furthermore, Cadmover sometimes makes it possible to use a program's "best" file format for export, then use the receiving program's "best" file format for import. It's often possible to avoid throwing out valuable geometry this way.

Because Cadmover is such an important product to so many users, I've decided to include the complete list of its import and export formats. Keep in mind that many programs not specifically on this list are actually supported by Cadmover through the formats they can read and write. In fact, I couldn't find a Mac 3-D program (among the general modelers or CAD packages) that isn't supported through Cadmover. Virtually every vector-based 2-D program on the Mac also is supported. The only exception, as far as I can tell, is RIB, which is not handled by Cadmover.



## Acceleration

Power Macintoshes completely changed the perception of Macintosh 3-D graphics; nevertheless, modeling, rendering, and all the other phases of 3-D still leave a lot of room for acceleration. The fact that Mac users still have to work with wireframe previews of models and bounding boxes for animations is evidence of this. On the very fastest workstations, users have the luxury of working in real-time shaded previews, which makes work much simpler and faster. The choices for acceleration on the Mac are pretty simple: faster hardware or faster software, or both. There are a number of options on both fronts, but the user will have to balance speed, cost, and convenience to come up with an equation that works.

### Faster Hardware

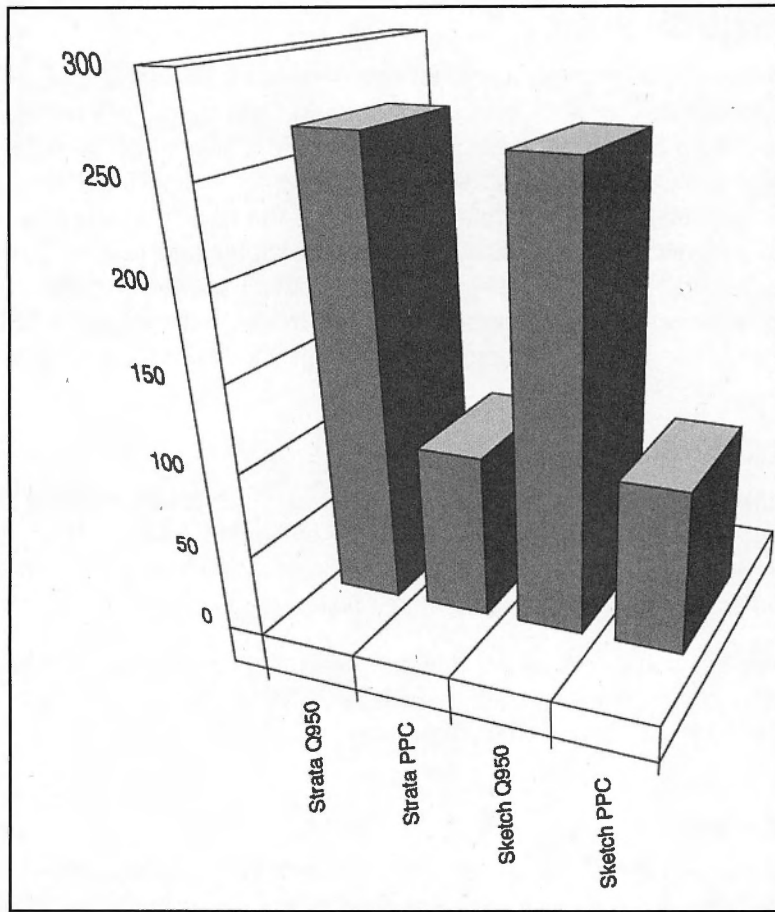
Power Macintoshes are currently the first and best investment you can make in speeding up 3-D work. For the equivalent price of Apple's fastest Mac a year ago (the Quadra 840 AV), you can now buy a machine that's as much as ten times faster at rendering (and three to four times faster at everything else).

In fact, Power Macs are very close, in some regards, to the fastest CPUs running on the highest-powered workstations available. This is due to the implementation of RISC technology in Apple's new computers.

### Power Macintosh

Reduced Instruction Set Computing (RISC) technology, as implemented by Apple in the Power Macs, is not much different than that used in high-end graphics workstations by Silicon Graphics. The raw processor speed is several times faster than that of the 68040 chips used in the Macintosh Quadra, and floating point math performance (which is used heavily in rendering) is as much as ten times faster.

The following normalized graph (see figure 13.2) shows typical rendering times for ray tracings completed in Sketch! and StrataStudio Pro. The slower times were rendered on a Quadra 950 with 64 MB of RAM; the faster times (shorter bars) were the same files rendered on the same Mac, with Apple's Power Mac PDS card installed. This configuration is about 30 percent slower than the current fastest Power Mac, the 8100/88.



**Figure 13.2** *Rendering comparison of ray tracing performance on a Quadra 950 and the same Quadra with a Power Mac PDS upgrade*

New versions of the Power Macs may be running at twice their current speeds by the time this book hits the street, and Power PC 604 and 620 chips are yet to come. Note that while Sketch! and StrataStudio Pro are roughly three times faster with the Power Mac PDS card running, some programs will put heavier demands on the floating point performance, and thus generate even greater improvements with the Power Mac.

What is amazing about all this speed improvement is that it has actually coincided with reduced Mac prices. In fact, you now can get the very top-of-the-line Mac for about half the price of the top-of-the-line Mac a couple of years ago.

## **Power Mac “Native”**

In order for your software to take advantage of the Power Mac’s performance, you’ll need versions of the software that run in “native mode.” This means that the software is specifically written to work on the Power PC chip. Older software is written to support the 68000 series of processors, which use very different core technology than the RISC-based Power Macs.

By putting emulation software in the PowerMac’s operating system, Apple has done a remarkably good job at maintaining backward compatibility with the 680x0 Macs. The downside to using software that works in emulation mode is that it’s actually slower than running the same software on a Quadra (typically about Mac IIci speed, although this may improve).

For users who already have an investment in 68040-based Macs, such as the Quadras, Apple offers a very low-cost PDS card that simply replaces the existing 040 in your Mac with a Power Mac CPU. This is a great solution, especially if you need to switch back and forth between the two processors for compatibility reasons. A control panel lets you switch off the Power PC card so that after you reboot your Mac, you’re back in full-speed 68040 mode. Electric Image, for example, offered highly optimized 68040 software, but seemed a long way off from actually delivering Power Mac native software as this was being written. For Electric Image users, standard 68040 Macs are still (as of late August 1994) the best option. Fortunately, the vast majority of Mac 3-D software is now running native.

## **Daystar PowerPro**

Currently, the only other Power Mac option for owners of Quadras, and possibly other Macs, is the Power Pro line of Power PC boards from DayStar, Inc. Like Apple’s PDS Power Mac upgrade cards, these override the normal 040 processor in your Mac and substitute the board’s Power PC. The Power Pro boards have the advantage of running at somewhat higher speeds than the Apple PDS card. They also enable you to install large quantities of RAM and a cache directly on the board, which results in a significant performance boost. The major drawback to these cards is their cost—about three to four times that of the Apple card.

## **YARC Boards**

Long before the Power Mac, YARC Systems, Inc. offered the NuBUS YARC board. This is a RISC CPU (not too different from the PowerPC) running on a board within your Mac.

The advantage to YARC's approach is that you can use the YARC as a rendering engine that runs independently of your Mac's CPU. This means you can send a rendering to the YARC board and get right back to work while it renders in the background. YARC's boards are very fast: the Zuma board, YARC's fastest, is based on a Power PC chip, and is roughly three times faster than a Power Mac, largely due to the on-board static RAM. Another advantage of the YARC boards is that distributed processing software—such as YARCRenderMan, BackBurner, or Shade III—can access multiple YARC boards in a single Mac, or even on multiple Macs on the network. If you put 12 Zuma boards together, you can have the equivalent of 36 or so PowerPCs all plugging away simultaneously on the same rendering.

The downside to the YARC approach is that RISC processors can be used only for rendering (or other applications for which custom software has been written, such as OCR). Because the Apple operating system still runs on the Mac's motherboard, modeling and other functions are not accelerated. However, rendering is by far the slowest part of any 3-D project, so the YARC boards are exceptionally good values for the serious 3-D user.

YARCRenderMan (the YARC version of RenderMan) enables RenderMan users to see the YARC board as just another RenderMan Chooser extension. When you go to render, your job is shipped across the NuBus to the YARC board and the rendering is displayed on your screen as it is completed.

## YARC Support

The following programs support YARC's boards for rendering:

- **VIDI's Presenter Professional** (YARC version) accelerates ray tracing, Phong shading, and RenderMan
- **Sculpt RISC** accelerates ray tracing and Phong shading
- **Shade III** accelerates ray tracing and Phong shading
- **BackBurner** (Infini-D's distributed renderer) accelerates ray tracing and Phong shading
- **MacRenderMan** (and all programs that support it, including Showplace, Typestry, PixelPutty Solo)

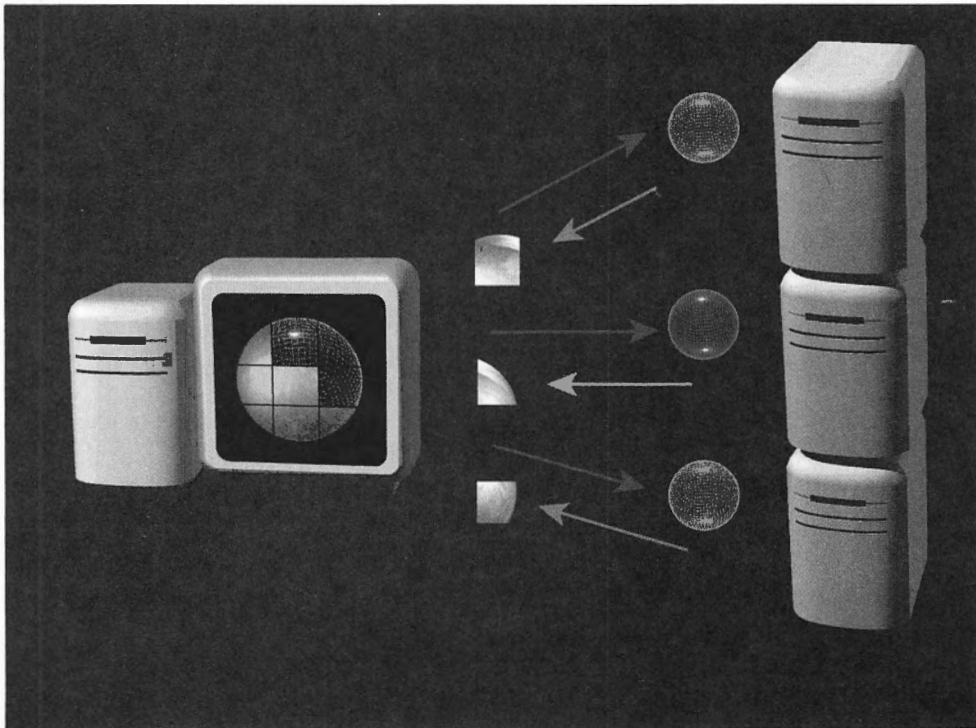
Strata has indicated that it may support YARC boards in the near future.

## Distributed Processing

As with hardware acceleration, software vendors are struggling to make the heavy demands of their renderers reasonable in a production environment, without requiring users to make huge additional hardware investments. One of the ways they can do this is with distributed processing. Also called *distributed rendering*, this technique harnesses the power of several computers to perform the work of one in much less time. Users of UNIX workstations have long had this capability, so it is with belated enthusiasm that Mac users are approaching this technique.

### How Distributed Rendering Works

Distributed rendering is simple in concept, but generally not so easy to implement. Essentially, it operates on the division of labor principle (see figure 13.3).



**Figure 13.3** *Distributed rendering*

1. A *master* computer (often the computer used for modeling and scene building) is connected to one or more *slaves* via a network. This reverses the traditional client-server relationship, in which many clients command the resources of one server.

2. The master sends the entire model and scene description, complete with texture maps, over the network to every slave machine. Each of these slaves then has everything it needs to render the entire scene, if necessary.
3. The master makes rendering assignments, instructing each slave to work on a different slice of the total scene. The slaves render their slices as quickly as possible and send the results to the master.
4. The master machine receives each slice of the final rendering and assembles it like a jigsaw puzzle. As each slave finishes a slice, the master assigns it a new missing piece of the rendering to help complete the puzzle.

Distributed rendering can be implemented in a variety of schemes. There are currently five software-only distributed rendering systems available on the Mac: Strata's RenderPro, Specular's BackBurner, RayDream's DreamNet, Alias' SketchNet!, and Pixar's NetRenderMan.

Essentially, each of these packages works the same way: you install a slave application on each machine on the network and you configure it to enable access to unused processor time. RenderPro's slaves include a calendar to schedule when the Mac will be available, so that rendering does not interfere with your regular work. Sketch!Net works much the same way. Specular's AutoBurner works like a screen saver, only enabling access to a Mac when the keyboard has been idle a specified amount of time.

When a rendering job is requested by the user at the master Macintosh, the master checks the network for available slaves and, if it finds them, initiates a network rendering.

The advantage of distributed rendering is that you can take advantage of existing networked Macs and have them all collaborate on a single project. The disadvantage is that rendering generally must take over the slave Mac to do any good. If the owners of the slave Macs are working on their machines, CPU cycles are withheld from the distributed rendering process and it does not do much good. It also annoys the owner of the slave Mac, whose work gets slowed down significantly. Essentially, this means you will need to reserve distributed rendering for after hours—a common practice, anyway, since rendering usually puts an end to modeling and other tasks for quite a while.

The other significant limitation of distributed rendering is the speed of the network. For users with a slow LocalTalk network, the time it takes to send models, textures, and environments over the wire tends to negate the advantage of having



many machines working on the same image or animation. Because there is a certain loss of efficiency for every machine added, the performance curve tends to level off after you add a dozen machines or so. With Ethernet, distributed rendering is much more viable, since the files are piped over the network much more quickly and network performance doesn't interfere with the process.

The final problem with distributed rendering software is that, like any rendering program, the slaves are memory and CPU hungry. Even if you manage to network a roomful of Mac Classics and install distributed rendering software on each one, you are not likely to see much of a performance gain over rendering on your master Power Mac 8100. The performance gained is directly related to the speed and system configuration of the Macs used as slaves. Since each slave essentially has to be prepared to render the whole image, minimum system requirements for the slave Macs are roughly the same as for the master machine.

One of the requirements of a distributed rendering system is that the master must be able to keep track of slaves as they are turned on or off, crash during a rendering, or disappear from the network for some other reason. All of the available distributed rendering systems manage the ins and outs of distributed rendering without much trouble, although they do it with widely varying degrees of efficiency.

While distributed rendering got fairly short shrift in its early incarnations, Power Macs should significantly bolster its popularity (assuming the distributed software isn't deemed unnecessary due to Power Mac's high speeds). While you used to need six or seven machines to significantly change the way you worked, adding two inexpensive Power Mac 6100's (with 80 MHz accelerated clocks) as dedicated renderers is like putting eight *extra* Quadra 840 AVs to work on your rendering (for the price of a single 840 AV!). Gone are the complications of setting up many rendering machines around an office, and gone is most of the network overhead that slows the process down. Renderings that used to take an entire eight-hour day now should require an hour or less—and the cost is miniscule compared to other comparable solutions. This means artists can now be free to work on designs and render them several times in an average work day.

## NetRenderMan

NetRenderMan is unique among distributed rendering software because it enables the master Mac to access any version of RenderMan available on the network. Thus, you may have a Mac calling on the services of a Quadra, a Mac II with a YARC board, and a Sun or SGI workstation—all at the same time. NetRenderMan is bundled with MacRenderMan and you get three rendering slave nodes for free.

### **Render Pro**

Strata's distributed rendering software, Render Pro, is extremely efficient at accessing network resources, so it closely approximates a linear performance gain as you add Macs to the chain. As mentioned, it includes a calendar to restrict access, so owners of slave machines can lock out the rendering access, except at specified times.

### **BackBurner**

Specular's distributed rendering software, BackBurner, includes a screen saver that automatically enables access to the slave when its owner has been idle for a specified amount of time. A YARC rendering engine is available for BackBurner, which means anyone rendering with BackBurner can access a YARC board on the network.

### **DreamNet**

Like the other distributed rendering systems, DreamNet offers access control by the owner of the slave Mac. RayDream also has plans for the DreamNet system software in the form of products from other vendors.

### **Sketch!Net**

Alias' distributed rendering software for the Mac uses a slightly different approach than the others. Instead of sending an entire model to the slave machines, it sends only 16-scanline chunks of tessellated data. The advantage is that Macs with less memory can contribute to rendering. The disadvantage is that there's a lot of network overhead and the master machine, usually the fastest renderer in the group, is reduced to running interference for all of the background processing. Sketch!Net, at least in its first version, wasn't very effective.

## **Mac vs. Other Platforms**

Of all the questions new 3-D users ask me, the most common is, "How does the Mac compare to DOS PCs and SGI?" It's a difficult question to answer, particularly as new developments emerge from Silicon Graphics, Microsoft, and Apple.

Silicon Graphics workstations, such as the Indy and Indigo, are high-speed RISC computers based on MIPS processors. They offer a number of real advantages over the current Macs: speed, graphics acceleration, and software. Meanwhile, PCs based

on Intel's 486 and Pentium chips have established phenomenal popularity due to their low prices over the last several years.

While the low end of the SGI line uses processors roughly equivalent to the PowerPC chips, the higher-end workstations offer extremely fast (150 MHz+) CPUs that sometimes work in tandem with other processors in the same machine.

All of the 3-D SGIs can be equipped with high-speed graphics engines such as the Elan or Extreme cards. These are accelerator boards that specifically speed up IrisGL system calls. IrisGL is SGI's graphics language for describing 3-D geometry in system software. This means that any software that supports it can display 3-D graphics with the help of these real-time rendering engines. SGI's can display smoothly-shaded animations in real time—a far cry from what you can currently get on your Mac's screen.

The final ingredient in SGI's favor is the breadth of fantastic software that's available. While Mac users have a lot of cool software to choose from, SGI users can create incredibly realistic inverse kinematics, physics-based animation, particle effects, morphing, and other effects that just don't exist on any other platform in such a refined state. Power Animator, SoftImage, and Wavefront are all Hollywood-class software tools that run, at least for now, only on SGIs. The vast majority of 3-D special effects created for TV commercials and videos are currently done on SGIs.

Microsoft (the heart-less soul of the PC world) has staked its claim to a 3-D future with two interesting business maneuvers. The first is that the company bought SoftImage, the famous SGI 3-D software developer, lock, stock, and barrel; the second is that it licensed the OpenGL system software (SGI's evolved version of IrisGL) for use in its own 32-bit operating system.

Neither of these developments has yielded any fruit for Windows users as yet; however, there are a number of ways that they may come into play. First of all, OpenGL support means that software developers who create tools for Windows NT can easily use the OpenGL toolbox to write 3-D display code—one of the hardest parts of developing 3-D software. With the help of low-cost PCI display accelerators tuned for OpenGL, Windows users could see graphics performance comparable to that of the SGIs for a fraction of the cost. Many of the display cards will be based on the Glint chip, which can provide up to 300,000 shaded, z-buffered polygons per second, according to early market hype.

The SoftImage purchase is a little hard to explain, but it's a safe bet that Microsoft is incorporating some of that company's technology into its own products.

Three-D Software for PCs is an odd mixed bag. 3D Studio, the de-facto 3-D standard for PC users, is a modest, polygonal 3-D package that supports IPAS plug-ins. Herein lies its power. Many developers have written IPAS plug-in extensions for 3D Studio. Things like spline modeling, booleans, particle effects, deformations, inverse kinematics, and other special effects are all accessible as add-on products. The problem with 3D Studio, which is common to all of the PC 3-D software, is that the interfaces are very non-standard, and usually unfriendly as well. Anyone who's comfortable with the Mac will be utterly horrified by their first experience with a PC 3-D package. The exception is probably MacroModel, which shares the same clean interface on both platforms.

Lots of other 3-D software is available to PC users, including Animation Master (which is identical on the Mac), Envisage 3D, and TrueSpace. Many of these programs offer features that still aren't available on the Mac.

One significant problem with the Microsoft entry into 3-D is that its newest operating system, dubbed "Windows '95" (which will be fully 32-bit and support Open GL), requires a huge remastering of programs that still run under DOS (like 3D Studio). Without this effort, OpenGL will be useless and it's unlikely that Windows NT will become a mainstream interface due to its high overhead.

While Windows users may revel in their software features, they definitely face a much tougher learning curve and less friendly day-to-day working conditions than their Mac counterparts.

## **Apple's Advantages**

The Macintosh computer has many advantages over PCs and SGIs, both as general graphics computers, and as 3-D workstations. To begin with, the Mac has an extremely elegant operating system which remains one of the best examples of modern research and development. Virtually every part of the system, from interface to networking and Inter-Application Communications, is transparently easy to use. This is crucial for graphics people who constantly move files between programs such as Photoshop and 3-D rendering software. Because of Apple's IAC engine, things like distributed rendering are extremely simple to set up and use. Anyone who has worked on or even seen a SGI operator in action, knows that UNIX doesn't work this way (no matter how pretty a "shell" you put around it). It's common for Unix graphics users to have separate program utilities (many costing thousands of dollars) for every type of file translation they need to make—and that generally means from one 3-D program to the next or even from one 2-D program to another.

The same goes for PCs. There are no widely accepted file formats for graphics, and networking and communication between applications are extremely cumbersome and difficult to set up. Systems integration (between hardware-and-software, as well as hardware-and-hardware) is also a mess. If you've ever tried installing a video board in your PC and getting it to work with all of your applications, you'll understand what I'm saying. (The last time I tried this, I was working on a top-of-the-line Compaq Pentium with top-of-the-line 24-bit video display board. After hours of frustration, and failed attempts to get my 3-D software to run, it turned out that the video board suffered an undocumented incompatibility with Crystal TOPAS Professional. This kind of miserable experience is common on PCs, but virtually unheard of on Macs.)

While Macintosh software is generally not as feature laden as the programs on other platforms, it often makes up for it in elegance of interface, ease of use, and simplicity of communications with other programs. If you need to create a texture in Photoshop, for example, you can simply launch the program, build the texture, save it as a PICT, and open the texture in your 3-D program directly. (The only exceptions to this kind of fluidity are Electric Image, which makes ammends with its very high-powered but simple interface, and Animation Master, which shamelessly adheres to DOS file naming conventions even on the Mac.) Key commands are standard on the Mac, so you always know which keys to press for Cut, Copy, and Paste, for example. And many 2-D graphics conventions, such as Adobe Illustrator's drawing methods, have become de-facto standards for Mac programs. Many professional 3-D artists who have access to both Macs and SGIs spend the majority of their time working on Macs, because in general, they're more productive.

Performance for the money is Apple's real ace in the hole. While Apple has historically been dinged on this count, the tables have turned dramatically with Power Macs. Apple's operating system is far more efficient than the coming Daytona (Windows NT) implementation (which needs 200 MB of hard disk and 8 MB of RAM, just for the minimal OS). The Power Mac is much less expensive to get off the ground. The prices of the top-end Pentium machines are actually higher now than a comparably equipped top-end Power Macintosh, but the Power PC, which is only in its first generation, is faster than the Pentium, which is nearly at the end of the line. As Microsoft works to put NT on RISC-based machines, Apple will already be long strides ahead as far as features, performance, integration, and ease of use in RISC operating systems.

SGI has made a lot of noise about its Indy computers. These are designed to be "desktop" machines at roughly the \$5,000 price point. However, the Indys at this

price lack the hardware graphics acceleration which is the primary reason for using a SGI in the first place. By the time you add the expensive fast RAM that an Indy needs, and a hardware graphics accelerator for “GL” applications, the price of an Indy is more like \$15,000. “Show specials” have promoted Indys with Alias Animator for \$10,000, but by the time you get the system fully configured and upgrade your Animator to “Power Animator,” you’ll likely have spent well over \$25,000. Fully-configured SGI Indigos, with the 3-D graphics boards, RAM, and so on, are much better suited to serious 3-D work. They start at around \$35,000. The average SGI software package is around \$10,000-\$15,000, but keep in mind that few computer artists are content to work with only one piece of software.

An important consideration when comparing SGIs to Macs is the pricey service contract you’ll need to keep a SGI operational. This is basically insurance on the operating system, so that when something goes wrong, someone who understands how will come and fix it. These contracts run in the neighborhood of \$150/hour. The Mac, in comparison, is almost childishly simple to maintain yourself. Even hardware problems usually are easily corrected. In addition, most Mac software comes with unlimited technical support for the price of the software itself, and this is also true of the Mac operating system.

## Tapping into SGIs

If you *do* work in an environment where SGIs are available, there are several products that will help you network into them from your Mac. This can be invaluable in terms of saving you from having to have SGI versions of applications like Photoshop (which may very well run better on Macs, anyway). It also gives you a way to get geometry and textures in and out of your Mac, as well as to use the SGI for distributed rendering with things like RenderMan.

While cross-platform networking is slightly beyond the scope of this book, there are several products available for networking to a Unix machine. These tools basically enable you to open a “folder” on the SGI to upload or download files. If you’re networked via MacTCP, you may also be able to use the SGI for RenderMan or Torqueware/Flashtacer rendering.

- **MacTCP.** This is the Mac system software that enables the Mac to communicate with a Unix workstation
- **KA Share.** Mac-to-Unix file sharing that has a reputation for being very “Mac-like” and reliable
- **Fetch.** An extremely popular shareware graphical front end to MacTCP that provides an interface for connecting to the Unix workstation

- **Telnet.** A public-domain terminal emulation program for “talking” to a Unix machine from a Mac
- **NFS share.** Another Mac-to-Unix file sharing utility

## **RenderMan Unix**

This version of RenderMan is required if you want RenderMan running on your Unix workstation (there are actually varieties for several different Unix platforms). You can install RenderMan on a whole network of Unix machines and use distributed rendering for extremely high-speed operation.

## **FlashTracer and TorqueWare**

The TorqueWare/Flashtracer bundle consists of a ray tracing engine for Strata’s products that runs on a SGI, and the communications software that manages the back-and-forth of 3-D files and rendered images between the Mac and the workstation. This is currently the fastest way you can ray trace a StrataStudio Pro or StrataVision 3d image.

## **IDT CZAR**

Integrated Device Technologies’ CZAR board is another RISC chip on a NuBUS board. This one, however, emulates a Silicon Graphics workstation, and supports the TorqueWare/VideoBits rendering solution for StrataVision 3d and StrataStudio Pro.

You can use this board (inside your Mac) exactly the same way you’d use the SGI with the Torqueware/Flashtracer bundle.

## **Apple’s 3-D Future**

Apple is just now entering the race to become the 3-D platform of choice. The release of Power Macs was the first step, but there are several missing pieces the company needs to produce to become a serious competitor to SGI, and to fend off the competition from Microsoft. Whether or not these plans will become reality is a source of conjecture. While none of Apple’s plans can be put down as “fact,” since the company refused to confirm any of them, the ideas I present here are the result of many conversations with people who have a very detailed understanding of the 3-D market, some of whom also have an inside line on Apple’s strategies and plans.

### 3-D QuickDraw (“Escher”)

Apple has privately demonstrated 3-D display technology code-named “Escher,” and sometimes referred to as 3-D QuickDraw, but the company has never formally announced it as a product. Basically, 3-D QuickDraw would be a system-level API, like the current QuickDraw. However, it would specifically provide automatic support for standard 3-D functions, like real-time polygon shading. This is very much how IrisGL works on the SGI and OpenGL will work in Windows NT. The catch is that IrisGL is roughly ten years old, and leaves much room for improvement. Apple, being loathe to adopt anyone else’s technology in the first place, is almost certainly “rolling its own” 3-D display system.

However, given the fact that OpenGL is a very popular standard with many vendors, and that those vendors are likely to incorporate it into their software to ensure optimum Windows compatibility, OpenGL seems to pose an attractive option for Apple. Since Apple will soon standardize on the PCI expansion board standard, and the Glint-based OpenGL accelerators will also be PCI based, there’s further reason to suspect that Apple won’t close off this path to high-speed 3-D graphics display.

On the other hand, because OpenGL admittedly isn’t the best possible solution to modern graphics performance (for example, real-time texture mapping is only supported at the highest end), it’s hard to imagine Apple chucking the opportunity to do things better—and gain market share in the process. The obvious solution would be to create a 3-D system software architecture that would enable OpenGL to “plug-in,” much as compression/ decompression codecs can plug-in to Apple’s QuickTime. At the same time, this would enable Apple, in cahoots with other hardware and software vendors, to come up with a state-of-the art graphics API to go well beyond the capabilities of OpenGL.

If Apple were extremely energetic, this set of commands would include support for many kinds of geometry, including NuRBS, and polygons, solids, and surfaces. Lights, textures, and even animation commands such as “pitch,” “roll,” and “yaw” could be supported. A robust set of commands would satisfy the developers of conceptual modelers, CAD products, and animators alike. Software developers would be able to write to these toolbox commands, instead of laboriously building their own.

Apple wouldn’t be alone in development of this technology. Canon, for instance, recently demonstrated a 3-D display technology to software developers which renders 20,000 or more polygons per second in software.



Apple's 3-D operating system could go much further than 3-D graphics. Instead of having a desktop with folder icons, Apple could offer a virtual office. Imagine how easy it would be to learn the Mac's interface if accessing a file meant pulling open a file drawer. Applications could become studios or workshops into which you could pick up and carry a file. Instead of storing graphics in folders, you could hang them in virtual galleries.

3-D QuickDraw would have important implications for business, as well. For example, you might be able to roll a 3-D graph around and zoom into important details from any angle. The opportunities provided by 3-D system software are hardly even hinted at by current products.

### **3-D Hardware (SuperMac 3D)**

Currently, the only hardware devoted to 3-D work on the Mac is in the form of dedicated rendering accelerators, such as YARC Systems' Zuma board. What's missing, however, is an accelerator for real-time 3-D display and a 3-D input device. As mentioned previously, cheap 3-D display accelerators for OpenGL will flood onto the market, probably in 1995. For users animating complex models, or for real-time 3-D visualization, such as virtual reality, hardware acceleration could take 3-D QuickDraw directly into the realm occupied by SGI's high-end systems. Apple would have to support display vendors and provide the software tools needed to develop such hardware. Eventually, as with DSPs and accelerated QuickDraw, accelerated 3-D circuits could end up on Apple's motherboards (this actually could happen soon, since developing the software is most of the battle). It may be that Apple will ship "garden-variety" 3-D acceleration on its motherboards. The highly-accelerated cards for high-end users will likely come from companies like SuperMac and Radius, whose 32-bit QuickDraw accelerators are currently indispensable tools for 2-D graphics users.

On the user interface level, the mouse may be OK for 2-D input (though not as good as a pen), but the two-dimensional operation of a mouse doesn't translate well at all into three dimensions. Every software company that creates 3-D software grapples with this problem and arrives at a different solution (which users then grapple with on a daily basis). Apple's ingenuity at solving this problem will say much more about its industrial design prowess than the stubby feet of the Quadra 800.

Apple also needs to deliver on its promise of a multiprocessing workstation based on the 620 PowerPC chip. This is Apple's only hope of catching SGI in high-end processing performance. A new machine supposedly shipping early in 1995 (tentatively called "TNT") is supposed to have five or six slots, no on-board video

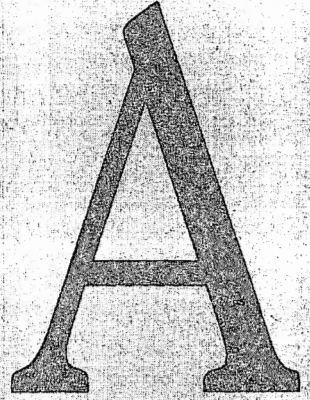
(since you'll be adding your own 3-D video card!), and top-of the line Power PC performance. At least one configuration of this Mac will be specifically tuned for 3-D performance.

### **3-D Metafile (3D PICT)**

For the several dozen different 3-D applications currently available on the Macintosh, there are an equal number of disparate 3-D file formats. This is a disaster for the Mac, where graphics files have always enjoyed a commonality that has made it easy to move work from one program to another. In the 2-D graphics world, it's one of the few features that really sets the Mac apart from other platforms. The only things that come close to a 3-D standard are ACIS, DXF, IGES, and RIB. However, 3-D can contain many kinds of information, including polygonal and spline geometry, textures, lights, and animation. None of these existing formats satisfactorily supports them all. It's impossible to save a complete 3-D scene with one program and open it, completely intact, in another. Even simple 3-D geometry is painful to move around between programs.

Ultimately, Apple will have to decide which features will please most of its customers and settle on these for a 3-D graphics file standard. Many of the vendors, such as Alias, have lobbied for acceptance of their own formats. Few companies, if any, are working together for a solution. Apple needs to introduce a flexible, extensible format that supports as many features as possible and entices software vendors to support it in their own products through ease of implementation and robust development tools. Apple would do well to follow its own examples in PICT2 and QuickTime. While Apple isn't in danger from competition at this point since nobody offers a good solution to this problem, a solid, functional metafile is mandatory to bring real Mac-like ease of use to the 3-D discipline.

**A p p e n d i x**



# **Macintosh 3-D Applications**

**W**hile the following information may have left out the odd product, this reference is intended to provide a basic run-down of all the products discussed in this book, and even some that I've not covered until now. Current Macintosh 3-D applications (and indispensable related products) as well as those soon-to-be available are described. Products in the body of the chapter are listed alphabetically by vendor.

The following table shows all of the products listed by category.

<b>2-D graphics</b>	Adobe Systems Inc.	Illustrator
	Adobe Systems Inc.	Photoshop
	Aldus Corp.	Aldus Fetch
	Fractal Designs Inc.	Painter
	MircoFrontier	ColorIt!
	Specular International Ltd.	Collage
<b>3D CAM</b>	Gibbs & Associates	Virtual Gibbs
<b>3D PostScript</b>	Adobe Systems Inc.	Dimensions
	Ray Dream Inc.	addDepth
<b>Animation utility</b>	ASDG	ASDG Exabyte Driver
	Diaquest	Mac-232
	Equilibrium Technologies	DeBabelizer
	Equilibrium Technologies	DeBabelizer Lite
	Knoll Software	The Missing Link
	Macromedia Inc.	LifeForms
	McQ Productions/ Software Systems	MacAnimator Pro

<b>Animation and rendering</b>	Electric Image Inc.	Electric Image Animation System
	Macromedia Inc.	Macromedia Three-D
<b>Architectural design</b>	Artifice Inc.	Design Workshop
	Dynaware USA Inc.	DynaPerspective
	Macromedia Inc.	ModelShop II
	SketchTech	UpFront
<b>3-D CAD</b>	Ashlar Inc.	Vellum 3D
	Autodesk Inc.	AutoCAD r 12
	DesignCAD Inc.	DesignCAD 2D/3D
	Graphisoft Inc.	ArchiCAD
	Graphsoft Inc.	MiniCAD+
	Intergraph Corp.	MicroStation Mac
	Unic Inc.	Architron II
<b>Format translation</b>	Kandu Software	CADMover
	Strata Inc.	Rend•X
<b>Hardware</b>	DayStar	PowerPro 601
	Diaquest Inc.	DQ-Animaq
	Immersion Corp.	Personal Digitizer
	Integrated Device Technology Inc.	CZAR
	Mira Imaging	Hyperspace
	Silicon Graphics Inc.	Graphics workstations
	Wacom	ArtZ
	YARC Systems Corp.	MacRageous II
	YARC Systems Corp.	NuSprint
	YARC Systems Corp.	Zuma

**Integrated 3-D  
modeling,  
rendering,  
animation**

Byte by Byte Corp.	Sculpt 4D
Byte by Byte Corp.	Sculpt 4D RISC
CrystalGraphics Inc.	Crystal TOPAS Mac
ElectroGig Inc.	3DGIG
Hash, Inc.	Animation Master
Hash, Inc.	Playmation
Macromedia	Swivel 3D Professional
Macromedia Inc.	Extreme 3D
Pixar	Typestry
Specular International Ltd.	Infini-D
Specular International Ltd.	LogoMotion
Strata Inc.	StrataStudio Pro
Strata Inc.	StrataVision 3d
Visual Information Development Inc. (VIDI)	Presenter Pro
YARC Systems Corp	Shade III

**Integrated  
still modeling  
and rendering**

Aldus Corp.	Super 3D
Alias Research Inc.	Sketch!
Ashlar Inc.	Design Reality
Byte by Byte Corp.	Sculpt 3D
Byte by Byte Corp.	Sculpt 3D RISC
HSC Software Inc.	KPT Bryce
Pixar	Showplace
Ray Dream Inc.	Ray Dream Designer
Strata Inc.	StrataType

<b>Inverse kinematics and simulation</b>	Knowledge Revolution	Working Model
<b>Model libraries</b>	Acuris	ClipModel Libraries
	Specular International	Replicas
	Strata Inc.	Master Series
	Strata Inc.	Strata Clip
	Viewpoint DataLabs	Viewpoint Datasets
<b>Modeling utilities</b>	Art of Fact	Terrainman
	Knoll Software	CyberMesh
	Onyx Computing	Tree Professional
	PaXar Technologies Inc.	Touch-3D
	Strata Inc.	Fractal Terrain
	Valis Group, The	PixelPutty
	View by View	AMAP
<b>Online services</b>	Alpha Channel	Alpha Channel BBS
	America Online	3DSIG
	CompuServe	Multimedia+/ Modeling & Animation
<b>Presentation graphics</b>	DeltaPoint Inc.	DeltaGraph
	Macromedia	Director
	Passport Designs	Producer Pro
<b>Renderers</b>	Graphisoft Inc.	Atlantis
	Pixar	MacRenderMan
	Video Bits	Flash Tracer
	YARC	YARC RenderMan

**Rendering utilities**

Pixar

NetRenderMan

Ray Dream Inc.

DreamNet

Specular International  
Ltd.

Backburner/YARC Burner

Strata Inc.

Render Pro

Torque Systems Inc.

TorqueWare

**Solid modelers**

Autodesk Inc.

form•Z

Bentley Systems

Solids modeler

Graphisoft Inc.

Zoom

View by View

Turbo 3D

**Surface modelers**

Byte by Byte Corp.

Sculptor

Macromedia Inc.

MacroModel

Valis Group, The

PixelPutty Solo

Visual Information  
Development Inc. (VIDI)

Modeler Pro

**Texture libraries**

Artbeats

Artbeats textures

Form and Function

Wraptures

Pixar

128

Pixar

Looks CD

Strata Inc.

StrataTextures

Valis Group, The

RenderMan Shaders

Visual Information  
Development Inc. (VIDI)

Sports Shaders

Xaos

Fresco

**Texture utilities**

Aldus/CoSA

Gallery Effects

HSC Software Inc.

Kai's Power Tools



	Specular International Ltd.	Texturescape
	Valis Group, The	Shader Detective
	Xaos	Paint Alchemy
	Xaos	Terazzo
<b>Video effects</b>	Adobe Systems Inc.	Premiere
	Aldus/CoSA	After Effects
	Aldus/CoSA	After Image
	VideoFusion Inc.	VideoFusion
<b>Virtual reality</b>	Strata Inc.	Strata Virtual
	The Harvii	Freeze Tag Power Mac
	Virtus Inc.	Virtus vr
	Virtus Inc.	Virtus WalkThrough
	Virtus Inc.	Virtus WalkThrough Pro

**Acuris Inc.**

931 Hamilton Ave. Menlo Park, CA 94025  
 Phone: (415) 329-1920 Fax: (415) 329-1928

**Clip Model Libraries**

Acuris creates high-quality 3-D geometry files for use in 3-D applications. The files are available in several formats with prices according to the complexity of the model.

**Adobe Systems Inc.**

1585 Charleston Road Mountain View, CA 94039  
 Phone: (415) 961-4400 Fax: (415) 961-3769

## Dimensions

Purpose: 3-D PostScript



A demo version of Dimensions is included on the CD-ROM.

Requires: 68020 or greater Mac (Power Mac native) System 7.0 or later, hard disk, 3 MB free RAM.

Dimensions began as an add-on product to Adobe Illustrator, but it has evolved into a sophisticated stand-alone program. You can extrude and lathe objects drawn with Dimensions' drawing tools, and you can draw labels which can be mapped onto the surfaces of objects. You can also import Illustrator artwork for use as starting outlines for creating 3-D shapes or for use as texture maps. Dimensions saves rendered 3-D artwork in editable Adobe Illustrator format. Highlights and shadows are really PostScript blends derived from an object's shape and colors. Dimensions is a very good program on its own, but really shines when you combine it with Adobe Illustrator. It exports Illustrator 1.1, 3.0, and 5.0 file formats and it can import Swivel 3D geometry files.

## Illustrator

Purpose: 2-D PostScript



A demo version of Illustrator is included on the CD-ROM.

Requires: System 6.05 or later, 2 MB RAM, hard disk. Power Mac native.

Illustrator is one of the all-time-great Macintosh graphics applications. In 3-D it's used for creating labels, such as those you'd put on a wine bottle, as well as many other kinds of textures. It's often used for creating line art which is then lathed or extruded in 3-D. If you also have Photoshop, you can "render" PostScript line art created in Illustrator as bitmapped art for use in your 3-D program. While it's often possible to work with type directly within 3-D applications, Illustrator is perhaps the best design tool available. You can save your finely kerned and adjusted type as outlines, then extrude it in your 3-D program, resulting in much finer control than you're likely to find in your 3-D program alone. Version 5.0 of Illustrator introduced powerful 2-D Boolean operations (under the Pathfinder filter menu option) and it has many, many features for quickly and easily creating complex artwork.

## **Photoshop**

**Purpose:** 2-D image processing and compositing



A demo version of Photoshop is included on the CD-ROM. The demo version of Photoshop 3.0 will be on the Alpha Channel BBS as soon as it's available.

**Requires:** Color Mac, hard disk, 10 MB free RAM. Power Mac native.

Photoshop is probably even more popular than Illustrator, and more useful for 3-D work. While it's a 2-D program, it's uses in 3-D are almost endless. Photoshop is used for scanning and creating textures and backgrounds, color correction and manipulation, filtering images using a wide range of third-party special effects plug-ins, creating motion blur, compositing multiple layers of images, retouching fine details in images, and on and on. It's safe to say that if you're a serious 3-D user, eventually you'll end up owning this program. Version 3.0 of Photoshop introduced sophisticated layering and floating selections, so you can now very easily move parts of objects around without affecting other layers.

## **Premiere**

**Purpose:** Digital video editing and effects



A demo version of Premiere is included on the CD-ROM. Power Mac native.

**Requires:** Color Mac, 4 MB application RAM, 80 MB hard disk. 16 MB RAM, 500 MB hard disk recommended.

Premiere is rapidly becoming the mainstream digital video equivalent of Photoshop. It's a powerful video editing application that's extremely well-suited to editing and compositing layers of 3-D animation. This is a great place to add transitions and sound to your animations, as well. Premiere is based on QuickTime, so it's highly accelerated by QuickTime-compatible JPEG hardware. Premiere has many professionally minded features, so if you're planning to take your animations out to video, you may want to consider using Premiere as a "post-production studio."

## **Aldus Corp.**

411 First Ave. South Seattle, WA 98104

Phone: (206) 628-4526 Fax: (206) 343-4240

### **Fetch**

Purpose: Image cataloging

Requires: Mac LC or greater, System 6.07, 4 MB RAM, hard disk. Mac SE/30 or greater, 5 MB RAM, System 7, and QuickTime recommended.

Fetch is an image catalog application that's quite useful for maintaining a library of textures and backgrounds. You can use it to archive images on your hard disk, the network, or on CD-ROM. The program has very good support for addition of key words and other information, so that it's easy to search for particular images.

### **After Effects**

Purpose: 2-D animation and post processing

Requires: Macintosh II or greater, 8 MB RAM, 80 MB hard disk, System 7.0 or later, QuickTime 1.6.1 or later. Power Mac native.

After Effects from Aldus' CoSA division, is the premiere 2-D animation and post processing application on the Mac—perhaps on any platform. It's invaluable for compositing animations and still images created in 3-D applications. After Effects has many advanced capabilities, like sub-pixel rendering (for extremely smooth motion) and a high degree of control over antialiasing and time-based filters. Many users create 3-D animations using alpha channel transparency, and save the background compositing for After Effects. If you need effects like multiple semi-transparent background layers flying in from all directions, while your main animation plays in the foreground, After Effects is definitely the power user's choice.

### **After Image**

Purpose: 3-D animation and post processing

Requires: System 7.0 or later, QuickTime 1.5 or greater, Mac II or greater, hard disk, 4 MB RAM or greater.

For budget-minded animators, After Image is actually the first version of After Effects. While it lacks that program's event-based timeline and spline motion paths, it's a very

capable program, and probably well suited to most 3-D animator's needs. After all, you'll probably create most of your high-level animation in your 3-D program.

## **Gallery Effects**

Purpose: Textures and image filtering

Requires: Photoshop or Photoshop plug-in-compatible program. Power Mac native.

Gallery Effects is an amazing collection of 2-D image "filters" for use in Photoshop (or other programs that support the Photoshop plug-in standard). Gallery Effects filters can instantly transform images into pencil sketches, watercolor swashes, or electrified auras. As well as working in 2-D, you can use Gallery Effects in programs like DeBabelizer or Premiere, to apply a filter to a series of animation frames over time. One really cool trick is to apply Gallery Effects filters to a single red, green, or blue channel of a Photoshop or ColorIt! image. This provides interesting effects that are more subtle than zapping an entire image.

## **Alias Research Inc.**

110 Richmond St. East Toronto, Ontario M5C1P1

Phone: (416) 362-9181 or (800) 267-8697 Fax: (416) 362-0630

## **Sketch!**

Purpose: Integrated modeling and rendering



A demo version of Sketch! 2.0 is included on the CD-ROM.

Requires: Mac IIci or greater with FPU, System 7 or later, Adobe Type Manager (to create outline text), 12 MB RAM, 10 MB disk space. Power Mac native.

Sketch! is a free-form 3-D modeler that uses unique navigational tools and a powerful NURBS-based modeling system with "putty" surfaces. It's designed as a 3-D illustration tool, rather than a high-end industrial design or CAD-accurate program, so it offers a fairly small selection of generalized features, rather than a huge toolbox of specialized features.

You can build just about any type of model in Sketch! with the exception that it doesn't offer true Booleans. Trim surfaces, however, are supported so you

can simulate holes in objects with the “treat as hole” feature. Although it’s a spline-based modeler, Sketch! has exceptionally good polygonal support. You can break models into any level of polygonal resolution, and save them as triangles or rectangular polygons. Sketch! lacks complex numeric inputs, so you can’t create highly accurate measured forms. However, it has very good snapping tools, which make it easy to create things that are proportionate to one another.

Sketch! ships with a batch renderer, Render!Q, and a distributed renderer, Sketch!Net, which make it easy to get rendering done in hurry—or at least out of your way. Its ray trace and Phong rendering are exceptionally good.

## The Alpha Channel BBS

Phone: (415) 759-6563 (data line)



A First Class connection client for the Alpha Channel BBS is included on the CD-ROM.

Purpose: Macintosh 3-D Bulletin Board Service

The Alpha Channel BBS is the online support forum for the *3-D Starter Kit for Macintosh*. Initial access to the BBS and comments and questions for Sean Wagstaff, the author of the *3-D Starter Kit for Macintosh*, are free.

A vast amount of material, cut from this book due to budget considerations, is available on the Alpha Channel. Look for in depth comparisons for various 3-D programs and hardware platforms, as well as informed speculation and scuttlebutt regarding future developments in Mac 3-D.

This is not only a support forum for the book, it’s also the only Macintosh 3-D “magazine” in existence. The BBS is devoted to bringing you up-to-date news about Macintosh 3-D graphics and providing unusual 3-D support services. The BBS includes the latest in Mac 3-D graphics news and gossip, demos of the most recent versions of 3-D applications, special offers and 3-D software discounts, cool shareware, and special-edition CD-ROMs. Look here for killer texture libraries, unusual 3-D models, and helpful expert tips. Unlike other online services, the Alpha Channel is operated like a periodical. It offers regular features, editorials, and news, in addition to an open forum.

One of the services provided by the Alpha Channel is an up-to-date database with the most current versions and prices of every 3-D and related program on the Mac.

Look here first for demos and reviews of Macromedia's Extreme 3D, 3DGIG, Presenter Professional 3.0, form•Z 3.0, Sculpt 3D 4.0 and Bentley Systems' solid modeler for Microstation Mac. In case you're intrigued by Apple's promises of 3-D QuickDraw and a 3-D metafile, The Alpha Channel BBS will offer demo versions and reviews of these applications and technologies absolutely as soon as possible— sooner than any print magazine.

## America Online

### 3DSIG

Purpose: Online service

America Online's 3-D forum (GO 3DSIG) is an invaluable service for 3-D users who want to keep up with what's going on in the Mac 3-D community. Within the forum are many friendly and helpful users who spend their time and energy answering questions and discussing cool ideas and solutions for 3-D problems. In fact, many of the ideas in this book are a direct result of conversations I had with people on this forum over the last year.

## Art of Fact

### Terrainman

Purpose: 3-D terrain maker



Terrainman, a great shareware program, is included on the CD-ROM.

Requires: Mac II or greater with a math coprocessor and 5 MB of application RAM. Power Mac native.

Terrainman is the pet hobby of Joe Ashear, an accomplished industrial designer in Brooklyn, N.Y., whose business, Art of Fact, is devoted to 3-D modeling and rendering.

## **Artbeats**

Purpose: Textures and backgrounds

Phone: (800) 444-9392 or (503) 863-4429.

Artbeats makes great textures and backgrounds for use in 3-D. One of the most popular collections is the Marble & Granite set which includes 120 PICT backgrounds and over 100 seamless tiles. The collections are available on CD-ROM.

## **Artifice Inc.**

PO Box 1588 Eugene, OR 97440

Phone: (503) 345-7421 Fax: (503) 346-0782

## **Design Workshop**

Purpose: Architectural design and 3-D visualization



A demo version of Design Workshop is included on the CD-ROM.

Requires: Mac II or greater with math coprocessor, hard disk, 5 MB of RAM, and System 7.01 or later. Power Mac native.

Design Workshop is an excellent, fast tool for 3-D architectural design and visualization. It's useful for other disciplines as well, but the program is really tuned for the architect. It's very easy, for example, to create walls, roofs, windows, and stairs.

The program features very fast tools for prototyping architectural space and it features unlimited layering and Boolean effects for windows and doors. Among other nice features, it calculates and renders architectural models for the path of the sun on a given day of the year.

## **ASDG Inc.**

925 Stewart Street Madison, WI 53713

Phone: (608) 273-6585



### **Abekas Driver**

Purpose: Video animation utility

Requires: Mac, 8 MB RAM, System 7.01 or later, 24-bit color recommended. Exabyte tape drive.

The Abekas Driver is an animation utility for transporting Mac animations on Exabyte tape to service bureaus for transfer to Abekas video systems—and subsequent perfect D1-quality dubbing to video.

### **Ashlar Inc.**

1290 Oakmead Parkway, Suite 218 Sunnyvale, CA 94086  
Phone: (408) 746-1800 or (800) 877-2745 Fax: (408) 746-0749

### **Vellum 3D**

Purpose: CAD and drafting

Requires: SE/30 or faster Macintosh, 4 MB RAM, 5 MB hard disk space, System 6.05 or later. Power Macintosh native.

Vellum 3D is a complex drafting and CAD environment for creating complex precision shapes. One of its greatest strengths is parametric modeling, the capability to relate the dimensions of one object to another. This means, for example, that you can build a part without regard to its dimensions. You can then modify one dimension, such as the length of a single side, and Vellum 3D will automatically scale the other sides accordingly. You could create a triangle, for example, and specify the angle defined by one intersection, and the program will automatically adjust the angles of the other corners.

Vellum 3D deals in surfaces, not solids, so many Boolean operations have to be simulated with surface operations. In addition, Vellum lacks many of the rapid-design features found in most free form modelers. If you use it, you may want to supplement it with a design-oriented program, such as Sculpt or Presenter Pro.

### **Design Reality**

Purpose: Modeling and rendering

Requires: Mac II or greater (with 32-bit addressing), math coprocessor, 256 colors or greater, 5 MB hard disk space, 16 MB RAM, System 7.01 or later.

Design Reality, when it first shipped, was a shock to the system of any Macintosh user. An arcane user interface ported from Unix Motif, along with the rest of the program, made it almost unusable for anyone accustomed to working on the Mac. The sample files and documentation shipped with the product made it clear that this was Ashlar's futile first attempt to offer a check-list item to potential corporate customers of Vellum 3D, namely, a surface modeler oriented towards design and visualization.

## **Autodesk Inc.**

2320 Marinship Way Sausalito, CA 94965  
Phone: (415) 332-2344 or (800) 964-6432 Fax: (415) 331-8093

### **AutoCAD Release 12**

Purpose: CAD and drafting

AutoCAD is an industry-standard, high-end CAD program, with a limitless range and depth of powerful, customizable, modeling tools and a programmable work environment. The optional extension to AutoCAD, AME, provides solid modeling functionality. Add-on modules from other companies provide capabilities such as RenderMan support and ray tracing.

Light types include: spot, point, sun, and ambient. File export formats include: DXF, IGES, EPS, and PICT. Formats supported by AutoCAD can be supplemented with add-on export tools. Render type is Gouraud.

## **Autodessys**

2011 Riverside Drive Columbus, OH 43221  
Phone: (614) 488-9777 Fax: (614) 488-0848

## **form•Z**

Purpose: Solid modeler

Requires: Macintosh II or greater, 3 MB RAM, math coprocessor, 5 MB hard disk space, System 6.05 or later. Power Mac native.



A demo version of form•Z 2.6 is included on the CD-ROM.

form•Z is one of the most powerful modelers currently available on the Mac. It's a solid modeler with a vast array of tools and features. It combines NURBS and polygonal geometry for the best of both worlds. Form•Z is one of two free-form solid modelers currently available on the Mac and it has extensive Boolean support, enabling you to easily carve one object with one another or to fuse objects together. The program has a high degree of integration with its 2-D tools for exporting models as plans and importing plans and turning them into models.

form•Z is used because of its high degree of control over surfaces and solids. Because of the numeric control, it's a popular program with industrial and mechanical designers and architects.

form•Z requires you to work in a single camera view, but you can save multiple views. Rendering is limited to flat shading and scanline, but form•Z's export support is unparalleled. The program saves in PICT, DXF, Electric Image Animation System (FACT), stereolithography (STL), 3DGF, RIB, EPS, and IGES formats, so it's easy to bring models into more capable rendering and animation programs.

Because of its fast, fluid modeling and its excellent polygonal model export features, form•Z is a popular modeling tool for animators. But the program also has a high degree of control at the engineering level, and will, for example, export files ready to be turned into physical models through stereolithography.

### **Byte by Byte Corp.**

8920 Business Park Drive, Suite 330, Austin, TX 78759  
Phone: (512) 795-0150 Fax: (512) 795-0021

### **Sculpt 3D**

Purpose: Integrated modeling and rendering

While a demo version of Sculpt 4.0 was not available in time for this book, it will be available on the Alpha Channel BBS as soon as it's ready.

Byte by Byte completely overhauled Sculpt 3D for version 4.0. Sculpt is a fast and very powerful modeler that includes surface-trimming booleans, NURBs, and vertex-level editing. Users value the capability to create just about any shape with the product. One valuable characteristic of the program is that no matter where a model originated, if you can import it in Sculpt, you can edit it with any of Sculpt's tools. Many other 3-D programs treat imported geometry as a frozen block; you can move it around, but you can't really change it.

Version 4.0 of Sculpt introduced the capability of mixing and matching splines and polygons, so for example, you can have a surface that's composed of both smoothly curved and faceted surfaces. A Promote feature turns polygonal objects, such as imported DXF files, into spline meshes.

Sculpt is renowned for the capability of placing multiple textures on a single surface; you also can apply textures to any face on a surface, such as bevels on type. Sculpt's ray-traced rendering is as good as any on the Mac. A not-so-subtle feature of the ray tracer is its speed—as much as four times as fast as other ray tracers by some accounts.

### **Sculptor**

Purpose: Modeler

Sculptor is simply Sculpt 3D without the rendering. A good choice for animators, architects, or users who otherwise need to build models for use in other programs.

### **Sculpt 4D**

Purpose: Integrated modeling, rendering, and animation

Byte by Byte hadn't finished developing the interface for the animation component of Sculpt, but a demo version will be available on the Alpha Channel BBS when it ships. The company expects to develop an event-based animation interface with considerably more functionality than in earlier versions.

## **CompuServe**

### **Multimedia+/Modeling & Animation**

Purpose: Online service

The Modeling & Animation section in the Multimedia+ forum on CompuServe is not nearly as Mac-centric as the America Online forum (in fact, you'll find its users tend to be somewhat PC arrogant and Mac hostile) but there are still a lot of useful topics and threads to be found here.

### **CrystalGraphics Inc.**

3110 Patrick Henry Drive Santa Clara, CA 95054  
Phone: (408) 496-6175 Fax: (408) 496-6998

### **Crystal TOPAS Mac**

Purpose: Integrated modeling, rendering, and animation

Ported from the powerful PC version, Crystal TOPAS Macintosh still has a long way to go to catch up with its PC counterpart (and it hasn't significantly changed in over a year). As late as August 1994, the Mac version was essentially unsupported and was not being actively developed by Crystal Graphics. Until this situation changes, there will be many better options available for modeling, rendering, and animation on the Mac.

### **DayStar Inc.**

5556 Atlanta Highway Flowery Branch, GA 30542

Phone: (800) 962-2077 or (404) 967-2077 Fax: (404) 967-3018

### **PowerPro 601**

Purpose: Power PC accelerators

System requirements: Quadra or another '040-based Mac. (DayStar plans to offer PowerPro boards for other Macs by the end of 1994.)

DayStar's accelerator boards turn Macs into high-speed Power Macs. The advantage over Apple's upgrade board is the high-speed on-board RAM access and cutting-edge speed. The disadvantage is the high price.

### **DeltaPoint Inc.**

2 Harris Court, Suite B-1 Monterey, CA 93940

Phone: (800) 446-6955 or (408) 648-4000 Fax: (408) 648-4020

### **DeltaGraph Professional**

Purpose: 2-D and 3-D presentation graphing



A demo version of DeltaGraph 3.5 is on the CD-ROM.

Requires: System 6.02 or later, 3 MB application RAM. Mac Plus or greater, System 7 and 5 MB RAM recommended, especially for 3-D charting.

DeltaGraph is probably the best way to quickly and reliably make graphs on the Mac. It's not a 3-D program as such, but it does support a wide variety of 3-D graphs which you can manipulate in a simple 3-D fashion. The benchmark graphs in this book were created using DeltaGraph.

DeltaGraph has its own presentation capabilities built in, so you may not need an additional presentation software product.

## **Diaquest Inc.**

1440 San Pablo Ave. Berkeley, CA 94702  
Phone: (510) 526-7167 Fax: (510) 526-7073

## **DQ-Animaq**

Purpose: Animation control hardware

Diaquest's Animaq boards are the industry standard hardware for controlling frame-accurate video decks and managing the layout of animations onto video. The QuickPass option enables you to capture video in near-real-time, while maintaining the high level of image quality that results from frame-at-a-time grabbing and recording.

If you do a great deal of professional level frame capture or recording to tape, the Diaquest boards are an excellent solution to part of the process. To use an Animaq board, you'll need a broadcast video editing deck with time code and a video frame grabber board—the faster the better.

## **Mac-232**

Purpose: Animation software

This software enables you to put animations directly onto videotape if you have a frame-accurate animation video deck. The software works just like Diaquest's hardware boards, except that some of the high-speed features of the boards, such as QuickPass, are not supported.

## **Dynaware USA Inc.**

950 Tower Lane, Suite 1150 Foster City, CA 94404  
Phone: (415) 349-5700 or (800) 445-3962 Fax: (415) 349-5879

### **DynaPerspective**

Purpose: Architectural design and 3-D visualization

Requires: Macintosh II or later, 2 MB RAM, 8-bit color or greater, hard disk.

DynaPerspective is a 3-D design and visualization package with lots of tools specifically for architectural design. It offers a polygonal modeler with architecture-specific features. Like most of the programs in this category, it offers simple animation using key frames, primarily for architectural fly-bys. Its export formats include: DXF, as well as flat-shaded PICT, Wireframe, and Hidden line renderings.

## **Electric Image Inc.**

117 E. Colorado Blvd., Suite 300 Pasadena, CA 91105  
Phone: (818) 577-1627 Fax: (818) 577-2426

### **Electric Image Animation System**

Purpose: Animation and rendering

Electric Image Animation System has the most complete set of animation tools available on the Macintosh. Features include spline-based motion paths, parameter curves, and detailed control of individual motion parameters.

EIAS competes directly with SGI-based animation software. It's very expensive (even more than some SGI programs), but for the professional video and multimedia animator, it justifies its price with power (as this book went to press, EIAS was not available in a native Power Mac version, but the company had promised that it soon would be.)

EIAS includes complete animation device control so you can render directly to video. The program also directly controls various animation hardware and can record directly to Abekas video recorders or to Exabyte tape drives in Abekas format. Version 2.0 introduced simple sound synchronization, as well as very cool particle animation and explosions. The most interesting feature of the newest version is deformations. You can bend, bulge, twist, and distort models in all kinds of ways. This is a must-have feature for character animators.

## **ElectroGig Inc.**

30 Huron Plaza, Suite 3807 Chicago, IL 60611

Phone: (312)573-1515 Fax: (312) 573-1512

## **3DGIG**

Purpose: Integrated modeling, animation, and rendering

3DGIG was only a twinkle in the eye of Mac animators when this book went to press, but according to the creator of the high-powered SGI-based application, it will be much more than that by spring of 1995. The program is expected to come to the Power Mac with most of its UNIX features: solid modeling, event-based animation with particle effects, and high-quality, ray-traced rendering. Stay tuned on the Alpha Channel BBS for news of this product's development.

## **Equilibrium Technologies**

475 Gate Five Road, Suite 225 Sausalito, CA 94965

Phone: (800) 524-8651 or (415) 332-4343 Fax: (415) 332-4433

## **DeBabelizer**

Purpose: Batch image processing and palette optimization



A demo version of DeBabelizer is included on the CD-ROM.

Requires: Macintosh with System 6.07 or later, 700K RAM. Non-Color-QuickDraw Macs can be used with DeBabelizer for image processing. Power Mac native.

DeBabelizer is a must-have for the professional Macintosh animator and multimedia developer. It's an image processing program that can perform a wide range of standard operations on large batches of files. You can write scripts to perform often-repeated functions, such as image resizing or palette reduction, then set the program running and walk away. DeBabelizer will convert almost any image format you can get onto your Mac's hard disk to almost any other format. File formats from many different platforms and programs are supported quite seamlessly. For example, it's simple to convert a Windows .FLI file to a QuickTime movie.

A new special-effects feature was released in version 1.6 that enables you to animate Photoshop-type filters over time when processing animations. This means you can



apply blur, for example, that decreases toward the end of an animation as a logo flies to a stop.

Most image and animation and still image formats from the following computer platforms are supported: Macintosh, MS DOS/Windows, General (such as Abekas, GIF, and Photo CD), Apple II, Atari, Commodore 64, Commodore Amiga, SGI, Sun, X Windows, and Philips CD-I.

### **DeBabelizer Lite**

Purpose: Batch image processing and palette optimization

DeBabelizer Lite is a greatly scaled-down version of DeBabelizer that will appeal more to those who create still images, than to serious animators (who will find the full version invaluable with all its bells and whistles).

### **Form and Function**

1595 Seventeenth Ave. San Francisco, CA 94122  
Phone: (415) 664-4010 Fax: (415) 664-4030

### **Wraptures and Wrapture Reels**

Purpose: Seamless still and animated backgrounds and textures

Wraptures are great collections of seamless textures, animated backgrounds, and still images that can be used to add all kinds of effects to 3-D images. They also are quite reasonably priced.

### **Fractal Designs Inc.**

335 Spreckels Drive, Suite F  
Aptos, CA 95003  
Phone: (408) 688-8800

### **Painter**

Purpose: 2-D painting, image processing, and animation

Painter is a "natural media" paint program that supports pressure-sensitive drawing tablets (such as those from Wacom). Its paint brushes and paints look and act like

natural artist's tools. For anyone who wants to create natural-looking computer images, this program is not only useful, but it's also quite fun—sort of like crayons for the Mac.

You can use Painter to create textures and backgrounds. It's also a good choice for compositing images. In addition, you can “clone” 3-D renderings using Painter's brushes to get realistic-looking “hand painted” 3-D images.

In version 3.0, Fractal introduced the capability to paint over frames of a QuickTime animation, so you can now do rotoscoping of animation frames without having to first save each of them as a separate image.

## **Gibbs & Associates**

5400 Tech Circle Moorpark, CA 93021

Phone: (800) 645-9399 (outside CA) or (805) 523-0004

### **Virtual Gibbs**

Purpose: Computer aided manufacturing (CAM)

Virtual Gibbs is a commercially available CAM system that controls milling and other manufacturing machinery. It does animated previews of a part being made so you can see the actual process in action before you commit it to metal. In some ways it's the ultimate expression of the 3-D process.

## **Graphisoft**

400 Oyster Point Blvd., Suite 429 South San Francisco, CA 94080

Phone: (415) 737-8665 or (800) 344-3468 Fax: (415) 871-5481

### **ArchiCAD**

Purpose: Architectural CAD



A demo version of ArchiCAD 4.5 is on the CD-ROM.

ArchiCAD is perhaps the premiere 3-D architectural CAD package on the Mac, particularly for small to mid-sized projects. It has a highly-refined interface and many tools especially suited to architectural design and drafting. Like most extensible CAD systems, it's extensible and features a built-in database and materials cost tracking.

Also, like other high-end CAD systems, it offers parametric modeling, which enables you to change individual dimensions of an object and have other related objects updated automatically to match the change.

Phong shading is ArchiCAD's highest-quality rendering (with shadows using a single light source); it also generates complete plans and other working drawings.

File import support includes DXF, GDL, ZOOM/Focus, and Swivel 3D. 3-D export support includes DXF, Lines, PICT, RIB, StrataVision (with an external module), topCAD, and GDL.

### Atlantis

Purpose: Rendering

Atlantis is intended for ray tracing ArchiCAD and Zoom models—in fact, it requires one of these programs to be useful (according to Graphisoft's marketing materials). It's from Abvent, the same French company that created ZOOM.

### Zoom

Purpose: Solid modeling (and RenderMan rendering)



A demo version of Zoom is on the CD-ROM.

Zoom is a full-featured solids modeler with complete Booleans and many of the features found in form•Z. It's not, however, as well developed as form•Z (nor does it seem to be advancing at nearly as rapid a rate). On the other hand, it offers some unique advantages, not the least of which is its direct support for RenderMan. Directional, spot, and point lights are supported, and you can create your own RenderMan shaders within the program. (You'll also need to buy MacRenderMan to use this feature.)

Graphisoft boasts an extremely high level of accuracy for Zoom, and all of the permutations of Boolean operations most users will ever need.

Zoom also has an great fly-through capability that enables you to create QuickTime animations.

## **Graphsoft**

8370 Court Ave. Ellicott City, CA 21043  
Phone: (410) 290-5114 Fax: (410) 461-9345

### **MiniCAD+**

Purpose: General CAD



A demo version of MiniCAD+ 5.0 is on the CD-ROM.

Requires: MiniCAD+ will run on virtually any Macintosh, though it's best suited to faster Macs with larger monitors. 2.5 MB free RAM (5 or more recommended), System 6.05 or later (System 7 recommended). More resources are required for the native Power Mac version.

MiniCAD+ is a phenomenal value for users who need general 3-D CAD features in an extensible product that's also easy to use and reasonably priced. While the program is very functional in every respect, and competes quite well with most Mac CAD programs, it costs about a fourth as much as most of the higher-end CAD systems.

It fully supports both 2-D and 3-D geometry, and you can work in both modes interchangeably (which admittedly takes getting used to). Like all CAD programs, MiniCAD+ will produce final working drawings, but it also will generate very complete 3-D models that you can export to other programs.

For visualizing 3-D scenes as you work, the program supports flat shading and hidden-line rendering. You easily can drag the view camera around to change your point of view.

MiniCAD+ has direct support for Strata's file format, so you can do photorealistic rendering in StrataVision 3d or StrataStudio Pro. Other export formats include RIB, DXF, EPSF, Worksheet, and Database. Import formats include DXF and EPS format. The DXF format is particularly useful for importing objects you've modeled in a freeform program as 3-D symbols. These can include things like ornate doors and windows, which you can simply "drop" into walls.

## **Hash Inc.**

72800 E. Evergreen Blvd., Vancouver, WA 98661  
Phone: (206)750-0042 Fax: (206)750-0451

### Animation Master

Purpose: Integrated 3-D animator

Requires: Mac II or greater with a math coprocessor and 5 MB of application RAM.  
Power Mac native.

Animation Master is one of the most unusual Mac 3-D products. It actually exists on several platforms, and it shows its DOS roots strongly in the Mac version. However, it's the only Mac 3-D product that currently offers inverse kinematics, and skin and skeletal movement. This makes it the only Mac program that really excels at character animation.

If it has a real down side, it's that the entire file naming structure and approach of the program in general, is directly ported from the DOS version of the program. This is not a warm, fuzzy Mac product, and you'll need a great deal more time to learn it than you would with most Mac programs. The program is broken into six modules—separate programs that provide an important part of the total animation project.

1. **Sculpture.** This is where you build the actual models.
2. **Decal.** This is the texture mapping phase.
3. **Character.** This mode is where you add parts of a model together and apply constraints, such as linking of limbs.
4. **Action.** While it's possible to describe a character's motion in the Direction phase, you can use this mode to apply motions, such as walking or talking, to a model.
5. **Direction.** This is where you set up actual animations, complete with lighting and other scene characteristics.
6. **Render.** Animation Master does ray tracing and only ray tracing.

### HSC Software Inc.

1661 Lincoln Blvd., Suite 101 Santa Monica, CA 90404  
Phone: (310) 392-8441 Fax: (310) 392-6015

### Kai's Power Tools 2.0

Purpose: Texture creation and image manipulation

Requires: Photoshop or another program that supports Photoshop-compatible plug-ins.

Kai's Power Tools is an indispensable Photoshop plug-in that provides an endless variety of seamless textures and other special effects. Since its introduction, it has become virtually impossible to conceive 3-D graphics without it. Ray Dream's Designer even enables you to use KPT right within the program. PICT (or JPEG or TIFF) files created with KPT and whatever the program in question (even DeBabelizer!) can be saved and imported into 3-D programs. Perhaps the three most useful tools in the package are Kai's Texture Explorer, which creates an infinite variety of procedural textures; Kai's Gradient Designer, which is invaluable for creating realistic backdrops, environment maps, and other gradients; and Kai's Fractal Explorer, which enables you to generate just about every conceivable type of fractal image. Any one of these tools is worth the price of the package, and there are lots and lots of other features in the KPT set.

### **KPT Bryce**

Purpose: 3-D scenery creation

Requires: Mac II with FPU or Power Mac, 5 MB free RAM. Power Mac native.

What Kai's Power Tools is to 3-D imagery, KPT Bryce attempts to be for 3-D scenery. It's an extremely powerful terrain generator—that works in 12-bits, instead of 8, like most terrain makers—and its ray-traced renderings and procedural textures are also otherworldly. The first 680x version of Bryce was extremely fun to use, with the caveat that renderings were quite slow—but the Power Mac version is much more useable. KPT has hinted about the possibility of a Power User's version: This may include DXF import and export as well as a deeper level of control over parameters. Stay tuned on the Alpha Channel BBS for news of this development.

### **Immersion Corp.**

P.O. Box 8669 Palo Alto, CA 94309-8669  
Phone: (415) 960-6882 Fax: (415) 960-6977

### **Personal Digitizer**

Purpose: 3-D digitizing hardware

Requires: HyperSpace or other Mac-compatible digitizing software.

The Personal Digitizer is the first low-cost 3-D digitizer available for the Mac. Working in conjunction with digitizing software, in particular, Mira Imaging's HyperSpace, it enables you to input 3-D geometry from real 3-D objects. Considering its low cost (there are actually two models available) it's highly accurate, and really quite easy to use.

Be forewarned, however, that digitizing objects is not as straightforward as you might hope. You have to carefully trace over a surface and press a key for every point where you want to add a data point to the model. As you create "patches" of polygons you have to add these into your model, and you also have to be aware of the direction the polygons are facing (this depends on the direction you move the digitizing stylus).

While the tablet I tried has a range of about 17" on a side, a slightly larger model is available. More importantly, it's possible to digitize much larger models by identifying three reference points on a part of the model and using these to "stitch" parts together.

Despite the hassles, I found the Personal Digitizer to be worth the trouble (if slightly fragile and fussy) if you need to put real 3-D objects into virtual 3-D space.

### **Integrated Device Technologies Inc.**

2975 Stender Way Santa Clara, CA 95054  
Phone: (800) 345-7015 Fax: (408) 499-4048

### **CZAR**

Purpose: Hardware acceleration

Requires: a Macintosh II or greater (with NuBUS) Torque Systems TorqueWare software.

This is a RISC-technology NuBUS board that supports VideoBits' FlashTracer rendering software for StrataVision 3d and StrataStudio Pro. Essentially, it's like having a SGI workstation on a board available for the acceleration of rendering.

### **Intergraph Corp.**

289 Dunlop Blvd. Huntsville, AL 35894  
Phone: (205) 730-2700 or (800) 345-4856 Fax: (205) 730-9491

## **MicroStation Mac**

Purpose: General CAD

MicroStation Mac is a high-end, extensible, programmable 3-D CAD system, with a built-in database, associative dimensioning, and complete cross-platform compatibility. It is soon due to take over as the favored high-end system on the Mac (partly because of Autodesk's waffling over Power Mac—or any Mac—support). Since MicroStation is highly extensible, it's open to a variety of software add-ons. Bentley Systems (the actual creators of MicroStation—Integrgraph markets it) plans to offer a solid modeling extension by the end of 1994. The release of this product will probably coincide with the release of the native Power Mac version of MicroStation.

## **Kandu Software Corp.**

131 Great Falls Street Church, VA 22046  
Phone: (703) 532-0213 Fax: (703) 533-0291

## **Cadmover**

Purpose: 3-D file translation utility

Cadmover is an amazing utility program that will find its way into the heart of any user that has to move a lot of 3-D information from one program to the next. It supports a far wider range of 3-D file formats than any other program (as well as 2-D vector graphics, such as vector PICT and EPS).

While many programs default to using DXF as a file translation medium, Cadmover will strive to maintain as much information as possible in translation. Spline formats, for example, benefit from the use of IGES, rather than DXF, because spline data is retained, rather than converted into polylines and polygonal surfaces.

Unlike other 3-D programs that generally have a narrow range of configurability when importing or exporting 3-D files, Cadmover is extremely intelligent in how it handles geometry. Where applicable, you are offered a fine level of control over import and export settings to ensure the best possible translations. Because the program is scriptable, you can repeat specific translations very easily once you've worked out the bugs on a single file.

Cadmover has matured greatly since earlier versions. All of the supported file formats are now bundled with the program and copy protection has been upgraded to a hardware dongle.



Cadmover can pay for itself in a single use if you come up against a translation that your 3-D packages just can't manage.

### Knoll Software

PO Box 6887 San Rafael, CA 94903-0887  
Phone: (415) 453-2471 Fax: (415) 499-9322

### CyberMesh

Purpose: 3-D modeling and terrain utility



A demo version of CyberMesh is included on the CD-ROM.

Requires: Photoshop.

CyberMesh is a really cool 3-D plug-in for Photoshop that converts grayscale images to rectangular, cylindrical, or polar-mapped 3-D objects. It's a great value for \$49.

You begin with a grayscale image, which CyberMesh converts to 3-D coordinate data. You can save this model as DXF, Cyberware (a high-end 3-D digitizing system), or FACT (Electric Image) format.

### Missing Link

Purpose: Video animation utility

Missing Link is a utility that enables you to turn Mac animation files into Abekas format files on Exabyte tape. It's a must have for animators who need to get their animations onto the highest-quality video possible.

### Knowledge Revolution

15 Brush Place San Francisco, CA 94103  
Phone: (415) 553-8153 or (800) 766-6615 Fax: (415) 553-8012

### Working Model

Purpose: Physics simulation and animation



A demo version of Working Model is included on the CD-ROM.

Working Model is a mathematically accurate physics simulation system that enables you to build 2-D models and set them in motion using physical laws. Forces and mechanical influences can be used to create machines and other simulations that respond to inverse kinematics. Animations created in Working Model can be exported directly, as key frames, to Macromedia Three-D; or motion values can be exported in tab-delimited format, and translated into position, rotation, and time values in programs such as Electric Image Animation System; then 3-D models can be substituted for their 2-D place holders.

Knowledge Revolution may be working on a 3-D version—or on physics-based plug-ins for upcoming 3-D packages. However, this is only my own speculation.

## **Macromedia Inc.**

600 Townsend St. San Francisco, CA 94103

Phone: (415) 252-2000 or (800) 945-4601 Fax: (415) 442-0190

## **Swivel 3D Professional/SwivelMan**

Purpose: Integrated 3-D animation



A demo version of Swivel 3D Professional is included on the CD-ROM.

Requires: System 6.07 or greater, 5 MB RAM, 5 MB disk space, 8-bit or greater color.

Swivel 3D is an outdated modeler that still enjoys a certain amount of popularity. Because it's a profile-based polygonal modeler, its files are very easy to convert to other formats, so it's widely supported as a file format. One of Swivel's hallmarks is its outstanding object linking for creating moveable joints. Swivel's rendering is accomplished through RenderMan. You can assign and configure RenderMan shaders and apply sun and ambient lights. It's even possible to animate shaders and render.

The demo is provided more for historical perspective than anything else. Swivel is no longer actively marketed or developed by Macromedia.

## **LifeForms**

Purpose: Human-form animation



A demo version of LifeForms is included on the CD-ROM.

Also available from:

Kinetic Effects Inc., Suite 310 1319 Dexter Ave. Seattle, WA 98109

Phone: (206) 283-6961

Life Forms uses key frames to animate accurately linked human figures. You can export animations directly to Macromedia Three-D or Swivel 3D or save them as wireframe QuickTime movies.

## **Macromedia Three-D**

Purpose: Scene building and animation

Macromedia Three-D is a very powerful animation program with a fussy, stiff interface. It enables you to animate 3-D objects in a single world view window, and to set cameras for new points of view.

Animation is event-based with a multiple-timeline frame-by-frame score. This is quite similar to Macromedia Director's score. You can control event parameters at a minute level of detail and set spline motion paths for objects. The program supports ease-in and ease-out, but no object tracking or banking.

A small library of primitives enables you to drop in simple shapes, but you need a separate modeler to really make the program useful. Bump maps are supported as are rotoscoped backgrounds. Shape morphing also is supported. Light types include point and sun. The program extrudes outline fonts. Three-D's best native rendering type is Phong shading, but the vast majority of users prefer its very powerful RenderMan rendering. It can animate shaders and renders directly from within the program.

## MacroModel

Purpose: Modeling



A demo version of MacroModel is included on the CD-ROM.

Requires: System 6.07 or later, 8 MB RAM, 8-bit or greater graphics, 6 MB hard disk space.

MacroModel is a highly accurate spline based free-form modeler. It uses a single active drawing window (whose size is fixed by your rendering settings) and offers extremely quick real-time smooth shaded previews. You can push, pull, twist, nudge, and bend surfaces easily. MacroModel has very good support for vertex-level editing; you can simply choose a point and bend it around just as you would with an Adobe Illustrator drawing.

Many users find MacroModel's interface unbeatably easy to use. In fact the tool palette is smaller than for any other powerful 3-D program on the Mac. On the other hand, the result is less of a sense of control over modeling operations than is offered by many other 3-D programs. MacroModel uses a unique View Browser and Object Browser, which can be opened simultaneously. These are for controlling views and object grouping, and they're quite powerful. Both offer virtual trackballs for spinning your view or for spinning the active model around any axis.

MacroModel's texture formats are limited to solid colors and internal procedural shaders and surface qualities. Lighting options include point, sun, and ambient. Rendering is limited to Gouraud. MacroModel exports geometry, lights, and textures in RIB format for RenderMan rendering. The program automatically extrudes outline type, with bevels, and it exports DXF, Macromedia Three-D, and Swivel 3D formats.

## ModelShop II

Purpose: Architectural design and visualization.

Simple polygonal modeler for architectural design and visualization. This is another of MacroMedia's outdated 3-D products that came to the company through its many corporate mergers. This one's due to disappear sooner than later.

## Extreme 3D

Purpose: Modeling, animation, and rendering

As this book went to press, *MacWEEK* broke a story about Extreme 3D, the heir apparent to both MacroModel and Macromedia Three-D. The story said Extreme 3D (the working code name for the product) would have an integrated modeling animation and rendering environment as well as support for Pixar's Looks texture format.

If you look at Macromedia's development efforts over the last two years, they've been exclusively focused on MacroModel—all of the other products have gone to seed, including Macromedia Three-D. So it's very easy to imagine Macromedia extending the capabilities of MacroModel by gutting its other outdated 3-D products to bring their functionality into MacroModel. The most logical of these products to roll into MacroModel would be Three-D and its animation score. Since Three-D boasts powerful parameter-level animation, the new program would be likely to contain this feature as well. And since vertex-level editing of splines is an integral part of MacroModel and spline based animation is an important feature of Three-D, it's nice to imagine that these features would merge in the new product. Because MacroModel has such "liquid" surfaces, it would be cool to be able to animate them.

### McQ

PO Box 1676 San Mateo, CA 94401-0908  
Phone: (800) 659-4755 Fax: (415) 342-0199

### MacAnimator Pro

Purpose: Video animation utility

Requires: NuBUS Mac, video tape recorder with a time code reader and a RS-422 port; or a Panasonic LQ-4000 laser video disc recorder; or a Sony LVR-series laser video disc recorder or a Sony EVO-9650. Video output and/or capture hardware, such as a NuVista board, a VideoVision board, a Video Explorer, and so on, and a serial cable for the Mac-video connection.



A demo version of MacAnimator Pro is on the CD-ROM.

MacAnimator Pro is similar in many ways to Diaquest's DQ-Animaq product. The difference is that MacAnimator is a software-only product that also does "FastPass" recording, so you can capture or record frames one at a time at optimal speed. Unlike the Diaquest boards, you can work with several common industrial and hi-8 video decks, which is an affordable alternative to buying a professional video deck. The drawback is that the software-only method isn't as fast as using dedicated hardware—also, MacAnimator Pro isn't cheap.

**Mira Imaging Inc.**

2257 South 1100 East, Suite 1A Salt Lake City, UT 84106  
Phone: (800) 950-6472 or (801) 466-4641 Fax: (801) 466-4699

**HyperSpace**

Purpose: 3-D digitizers and software

Mira Imaging's HyperSpace software is available separately, or with Mira's sophisticated (and fairly expensive) 3-D digitizing systems that come in about a dozen configurations. Combined, they enable you to digitize real-world models (such as clay sculptures or other physical models) in 3-D. These files can be saved in standard 3-D formats, such as DXF, for use in scene building, animation, and rendering applications.

HyperSpace also is the software used with Immersion's Personal Digitizer, and it's available for use with other hardware, as well.

**MicroFrontier**

3401 101st. St., Suite E Des Moines, IA 50322  
Phone: (515) 270-8109 Fax: 515-278-6828

**ColorIt!**

Purpose: 2-D paint software

Requires: Mac II or greater, 2 MB free RAM, hard disk.

ColorIt! is a full-featured, low-cost paint program that supports Photoshop-compatible plug-in filters and texture tools. It's a great 24-bit paint program for the money, and it will be especially good when it supports alpha channels in the next version. If all you need is a way to build textures for 3-D and do simple image editing, it's a low-cost solution.

**Onyx Computing**

10 Avon Street Cambridge, MA 02138  
Phone: (617) 876-3876 Fax: (617) 868-8033

## **Tree Professional**

Purpose: Tree modeling utility

Requires: Macintosh with FPU, 3 MB RAM, 13" color display.

Tree Professional generates very realistic 3-D tree models that you can use in other programs (trees are generated as DXF files). It offers a virtually limitless variety of broadleaf and pine trees, and you can generate the models with very fine control over the level of detail in a model, which is important for reducing model sizes.

Version 2.0, which was to ship in August 1994, is supposed to add palm trees, improve color manipulation, and provide new export formats.

## **PaXar Technologies Inc.**

3443 Fulton Road Victoria B.C. V9C 3N2

Phone: (604) 474-8975 Fax: (604) 474-8976

## **Touch 3-D**

Purpose: Hands-on modeling utility



A demo version of Touch 3D is included on the CD-ROM.

Requires: Macintosh with FPU, 3 MB RAM, 13" color display.

Touch 3D is a remarkably tactile product in an otherwise ethereal 3-D world. Its sole purpose is to “unfold” 3-D models, so that you can print out templates for building real models.

Touch 3-D indicates cut and fold lines and will print an unfolded model on a plotter—on foam board, for example—so you can generate very large physical models from a 3-D model. For computer modelers with an urge to take up Origami, it’s a fantastic product; many users will find it a great way to keep their kids occupied, as well. For professional 3-D users, it offers a very low-cost alternative to stereolithography. You can create a presentation-ready model in a fairly short time.

One cool way to put the program to work is with a color ink jet printer. Once you’ve printed out the color-mapped, unfolded model, you laminate the print out with adhesive overhead transparency film—then cut and fold the new model. The result is a high-gloss model with a very durable finish.

While it's probably easier to get from the Canadian distributor, the program is created by a Swedish company, Lundstrom Design:

Lundstrom Design  
Ekhagsvagen 7 104 05 Stockholm, Sweden  
Phone: +46-(0)8-15 46 63, 15 47 77 Fax: +(46) (0)8-15-82-85

## **Pixar**

1001 West Cutting Richmond, CA 94804  
Phone: (510) 236-4000 Fax: (510) 236-0388

## **128**

Purpose: Texture collection

128 is an excellent collection of seamless background images and bitmapped textures for use in multimedia and print projects.

## **Looks CD**

Purpose: RenderMan "Looks" shader library

The Looks CD contains the entire library of Pixar shaders and Valis Groups shaders in Looks format. The Looks are "locked" on the disc; you unlock them by purchasing the key code from Pixar (about \$50 per set).

## **MacRenderMan**

Purpose: Rendering engine

Requires: Macintosh II family computer with a floating-point coprocessor, 8 MB of RAM (12 MB or greater is strongly recommended).

MacRenderMan is the Macintosh version of RenderMan. You'll need this application to use RenderMan rendering with programs other than Showplace (Showplace ships with RenderMan). Examples of RenderMan rendering programs include Macromedia Three-D, PixelPutty Solo, VIDI's Presenter Professional, and ZOOM—and RenderMan is available bundled with some of these products.



## **NetRenderMan**

Purpose: Distributed rendering engine

Requires: Macintosh II family computer with a floating-point coprocessor, 8 MB of RAM (12 MB or better is strongly recommended).

NetRenderMan is the distributed rendering engine for RenderMan users. It enables you to access any machine on the network running its own copy of RenderMan. NetRenderMan is a separate purchase from RenderMan; it's sold by the number of machines you plan to run it on. A separate NetRenderMan UNIX license is available for running RenderMan as a rendering slave on UNIX workstations.

While NetRenderMan is capable of using most RenderMan nodes available to it on the network, it does not support YARCRenderMan—a letdown for YARC board owners.

## **Showplace**

Purpose: Scene building, rendering, and limited modeling

Requires: a Macintosh with a math coprocessor, System 6.07 or later, 8 MB of RAM, 13-21 MB hard disk, 8-bit color or greater.

Showplace is Pixar's direct interface to RenderMan. It enables you to build simple models, construct elaborate scenes, apply Looks-format shaders, and render final images using the included RenderMan. It is also an important intermediary between modelers that can export RIB files and RenderMan itself. Showplace can be used to open a RIB file, apply lights and shaders, and send the rendering job to RenderMan. Simple modeling is provided via plug-in extensions. The program ships with lathe, terrain, and window shade modelers. Showplace enables you to save multiple cameras. You can apply one texture per object. The program offers spot, point, sun, and ambient lights, and a type tool enables you to extrude fonts.

## **Typestry**

Purpose: Animation, rendering, and limited modeling

Requires: a Macintosh II or greater, System 6.07 or later (Multifinder under 6.07), 8 MB RAM, 5 MB hard disk, 8-bit color or greater. Power Mac native.

Typestry is a simple-to-use tool for creating animated 3-D artwork—primarily out of fonts. It enables you to extrude and bevel font characters, arrange fonts for a logo in 3-D space, add lights and Looks textures, and render the scene using the internal

version of RenderMan or MacRenderMan. Typestry has a full event-based animation interface, although parameter level animation is fairly simplistic. It does, however, support some cool special effects, including particle emission, fog, and motion blur.

Lights are limited to front-and-back directional (sun) lights. You can apply separate Looks to bevels and faces of objects. Typestry ships with the Glimpse editor for Looks, so you can modify shaders to meet your needs. Export formats include DXF, EPS, PICT, and RIB.

### **RayDream Inc.**

1804 N. Shoreline Blvd. Mountain View, CA 94043  
Phone: (415) 960-0768 Fax: (415) 960-1198

### **addDepth**

Purpose: 3-D PostScript

addDepth is a PostScript 3-D illustration program. It imports Illustrator or FreeHand line art and extrudes outlines, or text, with bevels. The program enables you to apply different textures to faces, sides, and bevels. As with Adobe Dimensions, rendering is provided as a PostScript line-art image that can be exported to Illustrator or FreeHand, or used directly in a desktop publishing package.

addDepth enables you to draw Bezier curves with its own drawing tools and to extrude anything you draw. Unfortunately, it doesn't offer a lathing tool, so round objects are not possible. You can apply multiple PostScript textures per object (one each to sides, bevels, and faces). As with Dimensions, lighting is limited to white directional lights. Export formats include PICT, FreeHand, editable PostScript, and EPS.

### **DreamNet**

Purpose: Distributed rendering utility

Requires: System 7 program linking, networked Macintoshes.

Dream Net is Ray Dream's distributed rendering software for use with Ray Dream Designer. The program is sold in multi-user packs, so you have to buy enough licenses for the number of machines you plan to use.

**RayDream Designer**

Purpose: Integrated still 3-D

Requires: Color-capable Macintosh, system 6.05 or later and 5 MB RAM (8 MB RAM for use with System 7), 32-bit QuickDraw 1.2 or higher (built in to System 7). Power Mac native.

Designer is the lowest-cost, full-featured 3-D program available on the Mac. Its primary market is illustrators who want to branch into creating 3-D images. It has many high-end features, such as spline modeling and ray-traced rendering, but the interface has been simplified to the point where almost anyone can successfully create 3-D illustrations.

Designer has an amazing array of features for a product at this low price and its Power Mac performance is quite respectable.

The spline modeler enables you to build extremely complex free-form and organic shapes. The program supports extensive texture and surface quality mapping, as well as direct drawing on model surfaces (the only program that does this). An open plug-in architecture allows for effects plug-ins as well as additional features in the future—for example, you can use Kai's Power Tools directly within the program. The program works in a single world-view window, so it's easy to stay oriented. You can use cameras to create multiple views.

Designer is very much geared towards the print or multimedia designer, and its production tools, such as the capability to set an image size and resolution, reflect that orientation.

**Silicon Graphics Inc.**

2011 N. Shoreline Blvd. Mountain View, CA 94039-7311

Phone: (415) 960-1980 Fax: (415) 549-0595

Purpose: Graphics workstations

The SGI line of products range in price from about \$5,000 for the lowest Indy machine to over \$600,000 for the Onyx and Challenge computers. Keep in mind that what makes the SGI's killer 3-D machines are the real-time graphics engines. The low-priced Indys all lack this feature, except for the R4600sc XZ.

## **SketchTech**

43 Main Street SE, Suite 410, Minneapolis, MN 55414

Phone: (612) 379-1435 or (800) 379-DRAW Fax: (612) 331-4962

## **UpFront**

Purpose: Architectural design and visualization

Requires: Color Classic or greater Macintosh, 4 MB RAM, System 6.05 or later. Recommended: 8 MB of RAM, math coprocessor, color monitor.

UpFront is designed as a basic 3-D package for visualization and illustration. As with Design Workshop, it's primarily geared at architectural space design and visualization, although it can be used for more general purposes.

UpFront offers animated walkthroughs and fly-by animations, and the sun light source can be positioned and automatically animated for time of day. UpFront is a polygonal modeler with simple tools for cutting out doors and windows. You can save unlimited views of a scene and animate the camera using basic key frames.

UpFront's export formats include: DXF, PICT, EPS, RIB, Alias PIX, TIFF, Draw, PICT, Architrion, Swivel CMD, Comma Text, and Tab Text. Rendering types include: flat shading, hidden line rendering, Editable PostScript, and RIB export.

## **Specular International**

479 West Street Amherst, MA 01002

Phone: (413) 253-3100 or (800) 433-7732 Fax: (413) 253-0540

## **Backburner/YARCBurner**

Purpose: Distributed rendering for Infini-D

Accelerates Infini-D rendering by distributing tasks over multiple Macs. BackBurner automatically accesses any YARC boards that are present on the network as well. BackBurner is sold on a per-slave basis; multi-user packs are available.

## **Collage**

Purpose: Still image compositing and manipulation

Collage is a multi-layer image compositing and manipulation program. It's great for use as a post-processing tool for compositing 3-D rendered images with backgrounds and so on.

## **Infini-D**

Purpose: Integrated modeling, animation, and rendering



A demo version of Infini-D is included on the CD-ROM.

Requires: Macintosh II or greater, 8 MB RAM, System 7 or later, 8-bit graphics or greater. Power Mac native.

Infini-D is a full-featured application that covers all the bases for the multimedia animator and graphic artist. It includes a polygonal, profile-based modeler that uses "workshops" for creating shapes.

Work is done in a standard four-window environment and you can place unlimited cameras to provide different views of a scene. Event-based animation offers multiple timelines but not minute parameter level animation controls.

Infini-D's Animation Assistants provide control over parameters such as ease-in and ease-out, banking, and aligning to motion. It's texture mapping is remarkably powerful: you can place multiple textures per object, as well as bump, glow, transparency, and other effects maps.

Shape and surface morphing are supported. Light types include spot, point, and ambient. Lights are limited to 1x and 5x power, although you can fake lights of different intensities by using lights with varying gray values. Infini-D automatically extrudes fonts with bevels.

## **LogoMotion**

Purpose: Integrated modeling, rendering, and animation

Requires: Macintosh II or greater, 8 MB RAM, System 7 or later, 8-bit graphics or greater. Power Mac native.

LogoMotion is a scaled-down version of Infini-D, whose modeling and rendering features are significantly less developed. On the other hand, it has a number of tools that actually make it easier for beginning 3-D users to complete 3-D projects.

## **Replicas**

Purpose: Textures and models

Specular's libraries of textures and models in Infini-D format.

## **TextureScape**

Purpose: Texture creation tool

Requires: 68020 or greater Mac, System 7.0 or later, 4 MB application RAM, hard disk. (68030 or greater, 8 MB RAM, Color Monitor, QuickTime recommended.)

TextureScape is a great 2-D texture generator that makes seamless textures out of any Illustrator-format outlines. You supply your own colors, transparency, and layering values, as well as the relative sizes and distribution of texture elements.

TextureScape will even animate textures to QuickTime movie format; you can set key frames for transparency, color, or any other parameters for any elements.

## **Strata Inc.**

2 West St. George Blvd. Ancestor Square, Suite 2100

St. George, UT 84770

Phone: (801) 628-5218 Fax: (801) 628-9756

## **Fractal Terrain Modeler**

Purpose: Fractal terrain generator plug-in

Requires: StrataVision 3d or StrataStudio Pro.

Fractal Terrain is a 3-D terrain generator that uses 2-D fractal geometry to generate 3-D landscapes.

## **Rend•X**

Purpose: RenderMan file rendering plug-in

Requires: StrataVision 3d or StrataStudio Pro, MacRenderMan.

Rend•X provides RIB output from StrataStudio Pro or StrataVision 3d. It supports lights, shaders, and geometry, so all you have to do is move the resulting RIB files into Showplace for rendering. Rend•X will even translate existing texture maps to RenderMan format.

## **Render Pro**

Purpose: Distributed rendering utility

Requires: System 7, AppleTalk network, program linking StrataVision 3d or StrataStudio Pro.

RenderPro is Strata's highly-efficient distributed rendering system that operates on multiple Macs on a network. It works as a plug-in to StrataVision 3d or StrataStudio Pro. You can get near-linear rendering speed increases by hooking up several fast Macs on a fast Ethernet network. The native Power Mac version should be particularly effective if and when it ships.

## **StrataType**

Purpose: Simple modeling and rendering

StrataType is a simple application for quickly generating realistic extruded type with bevels. It's an ideal solution for desktop publishing where you need very realistic rendered images. StrataType supports only one texture per object, although you can apply procedural textures that map differently to bevels. Light types include distant and ambient. Export formats include DXF, EPS, PICT, and StrataVision 3d files. The program's best rendering method is Phong shading, but shadows are not supported.

## **Strata Clip 3d**

Purpose: Model and texture libraries

Strata offers a wide selection of prebuilt models and textures in library form. The textures libraries include procedural as well as bitmapped textures.

## **StrataVision 3d**

Purpose: Integrated modeling and rendering

Requires: a Macintosh II or greater with math coprocessor, 5 MB of RAM, hard disk, System 6.07 or later (System 7 recommended), 8-bit color or greater. Power Mac native.

StrataVision 3d is a scaled-down version of StudioPro. It offers virtually all of StudioPro's features, except for event-based animation, animation plug-in support, and spline modeling.

As with StudioPro, StrataVision 3d has outstanding texture mapping. Only one texture per object is supported, but multiple effects maps can be used to significantly alter a surface's appearance. These include: bump maps, glow maps, transparency maps, and others. Light types include spot, point, ambient, and suns. The program automatically extrudes fonts with bevels. It also will extrude bitmapped images based on the bitmap's gray values.

StrataVision 3d does support limited key-frame animation—good for fly throughs and fly-bys, but not really suited to complex animated scenes.

StrataVision 3d supports the following file formats: DXF, IGES, EPS, PICT, MiniCAD, PICS, QuickTime, Super 3d, Swivel 3D, and TIFF. StrataVision's rendering options include RenderMan (with the optional Rend•X plug-in), radiosity, ray tracing, Phong, Gouraud, flat shading, hidden line removal, and wireframe. In addition to RenderPro, acceleration is available from Torque and Video Bits in the form of the TorqueWare/Flash Tracer tandem.

Because of its low price, excellent polygonal file support, easy-to-use texture mapping, and excellent ray tracing, StrataVision 3d is one of the most popular programs for architectural rendering and general ray tracing of models created in stand-alone modeling packages.

## Studio Pro

Purpose: Integrated modeling, rendering, and animation



A demo version of StudioPro is on the CD-ROM.

Requires: Macintosh II or greater with math coprocessor, 5 MB of RAM, hard disk, System 6.07 or later (System 7 recommended), 8-bit color or greater. Power Mac native.

StrataStudio Pro was the first integrated 3-D modeling, animation, and rendering package on the Macintosh to have a high degree of power and control in all areas. In many ways, it resembles the earlier version of StrataVision 3d. However, a number of important features set it apart: a Bézier-based 3-D modeling mode (which sits on top a polygonal modeling environment) and extensive high-level animation features such as event-based key framing, particle animation, and animation special effects.



This is the best all-around integrated package on the Macintosh for beginning and intermediate users of 3-D software interested in animation. It gets particularly high marks for its ease of use. Texture mapping is extremely powerful, yet simple to use—so you can precisely apply and manipulate textures and get very specific texture effects.

The event-based time score does not offer a very high degree of control at the parameter level, although Strata has hinted that this may be available in future versions. It does offer spline motion paths, animation plug-ins that provide auto ease-in and ease-out, banking, and particle animation features. Animation extensions are continually being introduced to extend the capabilities of the product. For example, shape morphing and particle morphing are both supported. Lights include point, sun, spot, and ambient; lights can have shadow-casting gels, among other cool features.

### **Strata Virtual**

Purpose: Virtual reality (real-time shaded animation)

Strata Virtual is both a rendering extension for StrataStudio Pro or StrataVision 3d and an application.

To use it, you build a model, apply textures, and “render” the scene using the Virtual renderer. This saves the file in a special format that is handled by Strata Virtual. You then open the file in StrataVirtual—at which point it uses real-time shading—and enables you to navigate around and to move the model around to see it from any angle. It’s very fast and much more realistic than early versions of the Virtus Walkthrough programs, but the navigation tools don’t give you nearly the same sense of interactivity. Nevertheless, it’s a good product; if for example, you want to be able to show a 3-D object from any angle on demand.

### **Torque Systems Inc.**

835 Emerson St. Palo Alto, CA 94301  
Phone: (415) 321-1200 Fax: (415) 321-1298

### **TorqueWare**

Purpose: Acceleration

Requires: Flash Tracer rendering software (sold in bundle), SGI workstation or IDT RISC hardware, Macintosh II or greater.

TorqueWare, in conjunction with the FlashTracer software, is used to accelerate StrataVision 3d and StrataStudio Pro rendering on a SGI workstation or IDT NuBUS RISC board. TorqueWare is the interface software between the Macintosh and the RISC system; FlashTracer (from Video Bits) is a rendering module that accelerates rendering.

## **The Valis Group**

2270 Paradise Drive Tiburon, CA 94920  
Phone: (415) 435-5404 Fax: (415) 435-9862

### **PixelPutty**

Purpose: Showplace modeling extension

A spline mesh surface modeler, PixelPutty is available as an option for Showplace. It enables you to create very organic “spliney” models by pushing and pulling on control handles in 3-D.

### **PixelPutty Solo**

Purpose: Modeling and rendering

PixelPutty Solo is a far different product than its Showplace plug-in sibling. It features very powerful spline editing tools (nine types of NURBS are supported!). It also does spline deformation Booleans—instead of “cutting” a spline surface, the program “dents” a model to create a hole (this feature was still in development, but due to ship around the same time as this book).

PixelPutty Solo enables you to place shaders and set lights, and it renders directly to RenderMan. The program comes with the Valis Group’s Shader Maker utility and a batch of shaders.

Despite its high-powered appearance, PixelPutty has one outstanding feature not boasted by similar products—a very low price.

## **RenderMan shaders**

Valis offers a wide variety of textures in Looks and Shader format for use in programs that support RenderMan.

## **Video Bits**

612 Fremont Ave., Suite 3 South Pasadena, CA 91030

Phone: (818) 403-0151 Fax: (818) 403-0148

## **Flash Tracer**

Purpose: Accelerated RISC rendering

Requires: TorqueWare software (sold in bundle), SGI workstation or IDT RISC hardware, a Macintosh II or greater.

Flash tracer is a SGI- or IDT CZAR-based renderer for StrataVision 3d and StrataStudio Pro. TorqueWare is the interface software between the Macintosh and the RISC system; FlashTracer is a rendering module that accelerates rendering. The program is specifically designed to speed up ray tracing, although with the delivery of the native Power Mac versions of Strata's products, the advantages of this approach are probably dubious for most users.

## **VideoFusion Ltd.**

1722 Indian Wood Circle, Suite H Maumee, OH 43537

Phone: (800) 638-5253 or (419) 891-1090 Fax: (419) 891-9673.

## **VideoFusion**

Purpose: QuickTime special effects

Requires: Mac II or greater, hard disk, color display, 8 MB RAM, System 7, QuickTime 1.6 or later. Power Mac native.

VideoFusion is a great QuickTime special effects program with many professional features. Version 1.6 introduced YARC and DSP support for high-speed rendering and image effects, full-screen previews, NTSC-safe filter, field rendering, and native Power Mac support.

It has many unusual features including Pan-Zoom-Rotate to spin video in 3-D, warp, and morph. The program has excellent alpha channel support and enables you to layer movies of different sizes.

VideoFusion has a complete environment for overlays and animated compositing and renders animations with very high quality. All of its filter effects are time-based so you can animate effects.

## **View by View**

1203 Union St. San Francisco, CA 94109  
Phone: (415) 775-6926 Fax: (415) 923-1205

## **AMAP**

Purpose: 3-D plant generator (modeling utility)

Requires: math coprocessor and System 7 or later.

AMAP is a highly functional modeling utility that generates realistic plants and trees. It supports Wavefront, DXF, and Turbo 3D export file formats.

While the variety of tree and plant types supported by AMAP is much larger than Tree Pro's, the level of control over individual plants and trees—and over the resulting files—is not nearly as great.

## **Turbo 3D**

Purpose: CAD-accurate solid modeling

Requires: Macintosh with math coprocessor, 5 MB RAM, and System 6.05 or later. Eight-bit or greater color monitor recommended. Power Mac native.

View by View is an architectural rendering firm that also distributes the program in the U.S. The company has done extremely high-profile renderings for prestigious clients (including the White House) and it uses Turbo 3D as its primary modeling software.

Turbo 3D is a CAD-accurate, highly-complex solid-modeling system for design and visualization of large architectural projects. It is for serious 3-D users and demands a very dedicated student. The program is capable of building highly-complex solid polygonal models. Animation is for basic visualization and includes key frames and time of day lighting. Rendering used is flat shading, wireframe, and hidden line removal.

## **Viewpoint DataLabs**

870 West Center Orem, UT 84057

Phone: (801) 224-2222 Fax: (801) 224-2272

### **3-D datasets**

Purpose: 3-D datasets

Viewpoint does extremely high-resolution 3-D models in a variety of popular 3-D formats. The company provides free low-resolution models for browsing; you can order the high resolution versions as needed.

## **Virtus Inc.**

117 Edinburgh S., Suite 204 Cary, NC 27511

Phone: (919) 460-4530 or (800) 847-8887 Fax: (919) 460-4530

### **Virtus vr**

Purpose: Real-time 3-D visualization

Requires: Mac SE or greater, System 6.04 or later, 4 MB RAM. Recommended configuration is a Mac LC III or greater, System 7.1 or later, and 6 MB RAM. Power Mac native.

Virtus vr is a real-time 3-D visualization tool that has the same navigational commands as Virtus' other applications. It does not have a modeler built in, although you can create additive models by putting together primitives from its fairly extensive library.

### **Virtus WalkThrough**

Purpose: Modeling and real-time 3-D visualization



A demo version of WalkThrough is included on the CD-ROM.

Requires: a Macintosh II, with 2 MB free RAM, a hard disk, 8-bit color or greater. Recommended: Centris or Quadra or greater, System 7.1 or later, 8 MB or more RAM.

WalkThrough is a simple polygonal modeler combined with a sophisticated, real-time navigational interface that enables users to "walk through" a model and look at it from many perspectives. The models are flat shaded, but the real-time navigation

supplies surprisingly realistic effects. The program can import a variety of models, so you can walk through designs created in other applications (although you'll want to restrain the level of complexity). The whole package is particularly well-suited to architectural designs, since walkthroughs are such a natural part of this genre, though the program also works well with other types of models.

Animation is user-controllable in real time: it's the closest thing to "virtual reality" available on the Macintosh to date. The program can export a PICS or QuickTime movie, as well as export models in the following formats: DXF (2D/3D), Claris CAD, and MacDraw II. Rendering types include flat shading and wireframe.

Because of the performance limitations imposed by real-time rendering, it's necessary to keep models fairly simple for the sake of smooth animated playback when navigating through scenes.

### **Virtus WalkThrough Pro**

Purpose: Modeling and real-time 3-D visualization with texture mapping

Requires: a Macintosh II, with 2 MB free RAM, a hard disk, 8-bit color or greater. Recommended: Centris or Quadra or greater, System 7.1 or later, 8 MB or more RAM.

WalkThrough Pro differs from WalkThrough in one very significant way—you can apply textures to your models which are rendered on the fly as you navigate through a scene. While the 680X0 version was very cool, I can't wait to see the Power Mac release. This will be the first "realistic" virtual reality program on the Mac.

### **Visual Information Development Inc. (VIDI)**

136 W. Olive Ave. Monrovia, CA 91016  
Phone: (818) 358-3936 Fax: (818) 358-4766

### **Presenter Professional**

Purpose: Integrated modeling, animation, and rendering

While a demo version of Modeler Pro was available on the CD-ROM with the first edition of the *Macintosh 3-D Workshop*, 3.0 was not quite ready in time for this edition. Stay tuned on the Alpha Channel BBS for a demo version as soon as one is ready.

Requires: a Macintosh II or greater with FPU, 8 MB RAM, 80 MB hard disk, System 7 or later. A Mac IIx or Quadra or 16 MB RAM and a 160 MB hard disk is a recommended minimum. Power Mac native.

Presenter Pro is an outstanding combination of two applications: Modeler, where models are built and grouped, and Presenter, a scene builder and animator. Modeler is a spline-based surface modeler with a full range of Boolean surface trimming tools. It has very good, highly-accurate modeling controls all around, and is a favorite industrial design-oriented product with many users. It's generally regarded as the most full-featured, fully spline-based modeler on the Mac. Modeler's output formats include: VersaCAD, WaveFront, Dimensions, EPS, IGES, DXF, PICT, RIB, and FACT (Electric Image).

The Presenter portion of the package enables you to place and aim cameras and lights, assign textures, and set up renderings. The animation offers very deep parameter-level control, drag-and-drop animation attributes, and outstanding RenderMan support. In version 3.0, VIDi completely revamped the rendering to provide high-powered ray tracing.

The animation interface features a unique option not found in any other Mac animation product—true 3-D sound. You can animate microphones and sounds; the sounds can be set so they are only picked up by mics that are in range. Unfortunately, inverse kinematics, a feature VIDi had hoped to introduce in version 3.0, has been put off for a future release.

### **Xaos Tools Inc.**

600 Townsend Street, Suite 270 East San Francisco, CA 94103  
Phone: (415) 487-7000 Fax: (415) 558-9886

### **Fresco**

Purpose: Backgrounds

Fresco is Xaos' first collection of background images for use in graphic arts. All of them were created by "artist in residence" George Lawson. They make excellent texture maps or backgrounds, particularly for use in print and video. Both high resolution and NTSC resolution images are supplied on the Fresco CD-ROM.

## **Terrazzo**

Purpose: Texture creation plug-in



A demo version of Terrazzo is included on the CD-ROM.

Requires: Photoshop or Photoshop plug-in-compatible application.

Terrazzo is a creative Photoshop-compatible plug-in that enables you to create kaleidoscopic textures out of existing images. The effect is fantastic when you want to create a tiled surface out of a fractal image, for example. It's very easy to create rich tiled patterns with the program, although it would be nice if there were more flexibility in the way you select kaleidoscope styles (they're currently selected from a fixed palette of options). On the other hand, if you need tiles for floors, walls, bathrooms, textiles, and so on, this plug-in makes fast work of them.

## **Paint Alchemy**

Purpose: Image effects filters

Requires: Photoshop or Photoshop plug-in-compatible application.

Paint Alchemy is a series of algorithmic filters that use user-selected "brushes" to filter an image. Similar in style and intent to Gallery Effects, Alchemy filters have the advantage that they're "image sensitive." Brush size, direction, and other parameters vary according to the content of an image. This makes them far more pleasing aesthetically when used with animations, because the brushes seem to actually relate to the image being filtered, rather than appearing like a textural projection screen.

## **YARC Systems Corp.**

975 Business Center Circle Newbury Park, CA 91320

Phone: (805) 499-9444 Fax: (805) 499-4048

## **NuSprint/MacRageous II/Zuma**

Purpose: Accelerators

Requires: a NuBUS-equipped Mac and specially adapted rendering or effects software.



YARC boards are extremely high-speed RISC NuBUS boards that provide super-fast rendering acceleration for the following 3-D products: Sculpt RISC and Sculpt 4D RISC; VIDI Presenter Pro YARC version (ray tracing and RenderMan); Infini-D (using BackBurner extension); all programs that render with MacRenderMan (using YARCRenderMan); Shade III, and Animation Master. Video Fusion also supports YARC rendering of 2-D effects.

Using a YARC board is as simple as plugging it in, installing a YARC application, and rendering.

The Zuma board is the new leader in the YARC line; it's based on a PowerPC processor. You can install your own RAM in the board—so it's far cheaper than any earlier YARC boards have been. At the same time, it's considerably faster. YARC says Shade III is as much as 3 times faster rendering on a Zuma board than on a Power Mac.

## **Shade III**

Purpose: Integrated 3-D modeling, rendering, and animation



A demo version of Shade III is included on the CD-ROM.

Shade III is a high-powered, high-priced integrated modeler, animator, and renderer. It's a popular (and extremely expensive) product in Japan and has been available there for a long time. The program is fully spline based and offers many unique features.

A ray trace renderer provides Boolean operations in rendering (this also works with RenderMan, which the program also supports quite well). Shade III relies on very powerful and easily configured procedural shaders, although you can also use bitmaps as texture maps.

Due to its robust spline support, Shade III's real forte is creating organic shapes. The Booleans in rendering mean you can combine these shapes in unusual ways to come up with very complex objects.

While the program doesn't have the depth of tools offered by other industrial-design-oriented applications, its rather few tools enable for fast and efficient work. The animation has an awkward interface, but is capable of providing complex nested motions. One of Shade III's greatest achievements is its very fast ray tracing—especially when teamed up with one or more YARC boards.



**A p p e n d i x**

**B**

**3-D Starter  
Kit for  
Macintosh  
CD-ROM**

**T**he CD-ROM included with this book has everything you need to get started with 3-D. Try out the working versions of software to see how it looks and feels to work in 3-D. An interactive tutorial, developed with Mac 3-D tools, guides you through many of the essential 3-D techniques. Images from the book are reproduced in color with explanations of how they were created.

Image galleries from a number of vendors provide a look at some of the amazing images created with Macintosh 3-D software. A library of sample models (in several standard formats) is a toy box full of surprises. A huge library of textures, backgrounds, bump maps, and effects maps supplies lots of material to work with.

**Tip**

To use the CD, you'll need a CD-ROM drive that works with your Mac. Most of the applications require that you copy them to your hard disk before launching. While requirements vary for each application, a Mac LC II or greater, 13-inch color monitor, and 5 MB of free RAM are recommended. If you don't have a CD-ROM drive, you also can obtain all of this material through the Alpha Channel BBS (see information below).

## Macintosh 3-D Workshop Interactive

Included on the disc is a full-color interactive guide to lighting, surface qualities, texture and environment mapping, rendering, and animation. Images and examples from the book are reproduced in color, along with many not found anywhere else—3-D illustrations and interactive lessons about key 3-D concepts. See how multimedia and 3-D really work together. This software is available only on this disc! The interactive guide requires a 13-inch color monitor and 4 MB free RAM. To install it, simply drag the whole “3-D Workshop Interactive” folder to your hard disk.

## Alpha Channel BBS

This connection client will automatically log you onto the Alpha Channel BBS, the online support forum for the *3-D Starter Kit for Macintosh*. If you'd like to e-mail the author, or find out the very latest news on Mac 3-D, the Alpha Channel's the place to do it. If you're looking for something on this disc, such as the latest demo of a

particular 3-D application, but you can't find it, the Alpha Channel may well have it ready to download. If not, you can certainly e-mail the author and request that it be made available.

The Alpha Channel BBS includes discussions of advanced topics not covered in the *3-D Starter Kit for Macintosh*, and features extended "author's cuts" of magazine reviews, tips from expert users, special offers for CD-ROM model and texture collections, and up-to-the-minute news of developments in 3-D. Logging on and sending e-mail on the Alpha Channel is free. A small fee is charged for downloading privileges to offset the cost of running the BBS.

If you don't have a CD-ROM drive, but would like access to any of the material on the CD, you can dial into the BBS using standard terminal emulation and download the regular connection client. You can also use a standard First Class connection client. The BBS phone number is (415) 759-6563.

The connection client works with most modems.

## Absynthe's Power Tools

This is the complete library of Infini-D tips written and illustrated by Chris Bernardi (a.k.a., "Absynthe," on AOL). They are by far the most informative collection of tips assembled for Infini-D users, but many of the ideas are directly applicable to other 3-D programs. Since the Power Tools include embedded illustrations, you'll need Microsoft Word to read them.

## Commercial 3-D Software

You'll find 24 working 3-D programs from different vendors. These are "try before you buy" versions that enable you to do everything but save your work. The try-out version of Infini-D even enables you to save small renderings and QuickTime movies.

The try-out programs on the CD-ROM have specific system requirements. In all cases, you'll get much better performance if you copy the folder containing the software to your hard disk before launching the application; in some cases, this is required.

- ArchiCAD
- CyberMesh
- Design Workshop
- Modeler Professional
- Shade III
- Sketch!

- Dimensions
- form•Z
- Infini-D
- Life Forms
- MacroModel
- MiniCad+
- ZOOM
- StrataStudio Pro
- Swivel 3D Professional
- Terrazzo
- Touch 3D
- Virtus vr
- Vistapro

## Other Try-out Software

Because no 3-D program is an island, this disc includes demo versions of important software that supplements and enhances the capabilities of any 3-D application.

- Photoshop
- Premiere
- Working Model
- Color It!
- MacAnimator Pro

## Shareware & Freeware

There is a giant collection of shareware and freeware on the disc, including a number of fully-functional 3-D modelers and renderers, as well as a host of supporting utilities, such as terrain generators and fractal pattern makers. The collection includes:

- 3-D Illusions
- DXF Made Easy
- epsConverter
- FFT
- Fractal! 1.2
- GraphicConverter
- MacConcept
- Mandella
- Mirage
- MovieMaker
- Points 2 DXF
- PoV-RAY

- JPEGView (Power Mac)
- RenderCAD-Pro
- Sculptor
- Terrainman
- Texture Map

## **Image Galleries**

Sample images and animations created by many different users in many Mac 3-D applications can be found in the same folders as the software from those vendors.

These include images created in the following software:

- Animation Master
- Sculpt 3D
- Presenter Professional
- Ray Dream Designer
- Typestry
- Showplace
- Photoshop
- form•Z

## **Sample Models**

Models used to create many of the illustrations in this book are included, as are impressive samples of digitized 3-D objects from third parties. You can open these models in most 3-D applications.

Commercial samples also are provided by:

- Mira Imaging
- Illustrative Impressions

Many sample models come with the try-out applications listed under “Commercial Demos.”

## **Backgrounds and Textures**

Every 3-D user needs more seamless textures, bump maps, and effects maps. Powerful 3-D images also call for environment maps and high-quality backgrounds. These textures and effects maps can be used with any 3-D application that supports bitmapped textures. The disc contains two collections:

### **Wagstaff's Favorites**

Folders full of the author's favorite home-brewed textures. These include lots of variations on several themes, with matching bump maps and effects maps for creating highly-realistic and unusual surfaces. There's also a collection of interesting and fanciful backgrounds and environment maps.

### **The Alpha Channel**

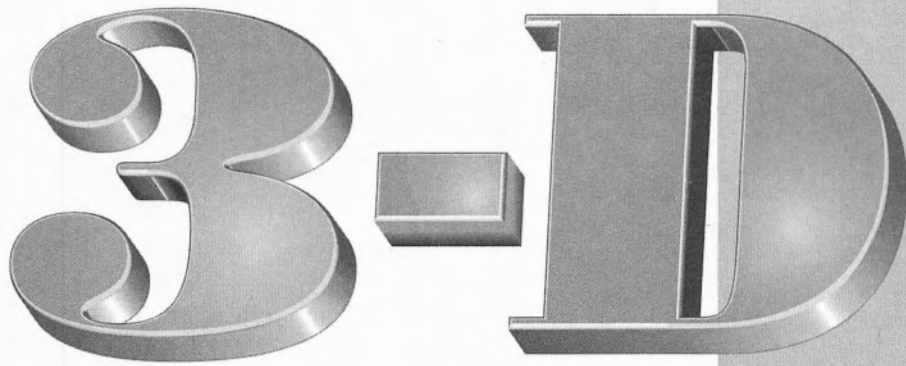
The disc contains two complete sets from the Alpha Channel's Expert Collection 3D. Visually and thematically matching backgrounds, textures, and bump maps make it easy to create attractive, color-coordinated 3-D graphics for multimedia and video.

## **HyperCard Player**

The latest version of the HyperCard Player enables you to use the Interactive Macintosh 3-D Workshop. You also can use it to run other HyperCard-based applications.



l n d e x



**STARTER KIT  
FOR  
MACINTOSH**

---

## **Symbols**

---

1-bit (black and white) format, 345  
2D graphics vendors, 418  
4-bit format, 345  
8-bit format, 345-346  
64-bit format, 347  
128 (Pixar), 454  
16-bit format, 346  
24-bit format, 347  
3-plane view, 56-60  
32-bit format, 347  
3D CAD vendors, 419  
3D CAD (computer-aided design)  
    modeling, 162-164  
3D CAM vendor, 418  
3D datasets (Viewpoint DataLabs),  
    467  
3D QuickDraw (Escher), 414-415  
3DGIG (ElectroGig), 438  
3DSIG (America Online), 429

---

## **A**

---

Abekas Driver (ASDG), 431  
absolute position, 56-59  
Absynthe's Power Tools (CD-ROM),  
    475  
acceleration  
    Daystar PowerPro, 403  
    Power Macintosh, 401-402  
    YARC boards, 403-404  
accelerators, 34  
Acuris' Clip Model Libraries, 423  
Add Floor command (Backgrounds  
    menu), 93  
addDepth (RayDream), 86-89, 456  
additive modeling, 26, 102-104  
Adobe Systems Inc., 423-425

Dimensions, 78-86, 424  
Illustrator, 424  
Photoshop, 352-362, 425  
Premiere, 371, 425  
After Effects  
    Aldus, 426  
    CoSA, 372  
After Image (Aldus), 426-427  
Aldus Corp., 426-427  
    After Effects, 426  
    After Image, 426-427  
    Fetch, 426  
    Gallery Effects, 246, 427  
Alias Research's Sketch!, 427-428  
alignments in three-plane view, 59  
alpha channel (compositing images),  
    342  
Alpha Channel BBS, 21, 428-429  
    CD-ROM, 474-475, 478  
AMAP (View by View), 466, 471  
    modeling, 158-159  
ambient light, 255-256  
ambient reflection (surface quality),  
    204-205  
America Online, 429  
America Online's 3DSIG, 429  
animals, 157  
animation, 5, 11-13, 30-31, 269-308,  
    371-372  
    After Effects (CoSA), 372  
    architectural key-framing, 274  
    bevels, 150  
    cameras, 49, 286-289  
    CD-ROM, 21  
    ease-in and out, 291  
    ELIAS (Electric Image Animation  
        System), 283-284, 351  
    event-based, 274  
    film, 300-307  
    formats, 347-351  
    frame rate, 304

- groups, 141
- hardware, 43-45
- key-frame animation, 31, 271-289
- long-term storage, 37
- morphing, 296-299
- MPEG video, 306
- multimedia, 306
- numbered PICT format, 348
- particles, 302-308
- PICS format, 351
- Premiere (Adobe), 371
- Presenter Professional 3.0 (VIDI), 284-285
- QuickTime, 348-351, 371-372
- resolution, 304
- rotoscoping, 300
- shaders, 392-393
- smoothing motion, 291-296
- sound, 289-296
- still images, 383-384
- timelines, 275-285
- type, 100
- vendors, 418
- video, 305
- VideoFusion, 371
- virtual reality, 274, 307-308
- animation and rendering vendors, 419
- animation controllers, 44, 373
  - DQ-Animaq, 375
  - DQ-MAC232, 374-375
  - Electric Image Animation System (EIAS), 375
  - EVO-9650, 374-375
  - Exabyte drivers, 376
- animation decks, 44
- Animation Master (Hash), 443
- Animation-Millions of Colors compressor, 349-350
- antialiasing, 332
  - interpolative antialiasing, 333
  - rendering speed, 337
- Apple, 413-416
  - Cinepak compressor, 350
  - Photo-JPEG compressor, 350
- applications, 417-471
- ArchiCAD (Graphisoft), 440-441
- architecture, 14-16
  - vendors, 419
  - key-frame animators, 274
- Art of Fact, 429
- Artbeats, 430
- Artifice's Design Workshop, 430
- Artwork menu commands
  - Make Guide, 78
  - Map Artwork, 82
- ASDG Inc., 430-431
- Ashlar Inc., 431-432
- Atlantis (Graphisoft), 441
- atmosphere, 30
  - rendering, 337
  - scene building, 191-193
- attenuation (lighting), 259
- Attributes window in Presenter Professional 3.0, 285
- audio animation, 289-296
- authoring tools (multimedia), 378-379
- AutoCAD Release 12 (Autodesk), 432
- Autodesk Inc., 432
- Autodesk's AutoCAD Release 12, 432
- Autodessys, 432-433
- Autodessys' form•Z, 432-433
- axis of rotation (lathing), 119-122

---

**B**

---

back light, 258  
Backburner (Specular), 408, 458  
backgrounds  
    bitmaps to models, 176-184  
    CD-ROM, 478  
    CyberMesh, 178-179  
    imported backgrounds, 184  
    Infini-D, 182  
    modeling, 161-162  
    rendered backgrounds, 175-176  
    scene building, 174-191  
    shadow casting on (scene building), 190-191  
    Showplace, 184  
    StrataStudio Pro, 179-182  
    Terrainman, 177-178  
    transparent backgrounds, 188-189  
banking with StrataStudio Pro, 292-295  
basic extrusions, 106-107  
batch rendering, 337  
BBSs (bulletin board systems)  
    Alpha Channel BBS, 21, 428-429  
beveled type, 100  
    animation, 150  
    modeling, 149-150  
bitmaps to models (backgrounds), 176-184  
black and white (1-bit) format, 345  
blocks  
    extrusion, 107  
    primitives, 143  
Blur/Motion Blur command (Filter menu), 360  
bonding of objects (modeling), 102-104  
Boolean modeling, 26  
Boolean rendering, 162, 330-331

bounding boxes, 66-68  
Bryce (scene generator), 186-188  
bump maps, 28, 221-227  
    CD-ROM, 224  
    modeling, 160  
    texture creation, 236-239  
bump-mapped type, 100  
bumps (textures), 196  
Byte by Byte Corp., 433-434

---

**C**

---

CAD (computer-aided design), 14, 16, 102  
    modeling, 162-164  
    vendors, 419  
Cadmover (Kandu Software), 399-400, 446-447  
CAID (computer-aided industrial design), 10  
CAM (computer aided manufacturing), 10  
Camera Info dialog box, 286  
cameras, 30  
    animation, 49, 286-289  
    depth of field, 191  
    flying camera logos, 286-289  
    fully animated, 51  
    lens length compared to moving the camera, 52-56  
    lenses, 52  
    movable still image cameras, 51  
    navigation, 49-56  
    orientation, 191  
    panning, 51  
    point of view, 51  
    scene building, 191  
    single still cameras, 51  
    single-perspective view, 50

- Sketch!, 49
- switchable fixed-positon cameras, 51
- tilting, 51
- views, 73
- virtual real-time cameras, 51
- zooming, 51
- capped surfaces (extrusion), 104
- caps (extrusion), 109
- Cartesian coordinates, 56-59
- CD-ROMs, 21-22
  - Absynthe's Power Tools, 475
  - Alpha Channel, 478
  - Alpha Channel BBS, 474-475
  - animation, 21
  - backgrounds, 478
  - bump maps, 224
  - color, 21
  - commercial software, 475-476
  - drives, 39-40
  - freeware, 476-477
  - HyperCard Player, 478
  - image galleries, 477
  - interactive guide, 474
  - multimedia animation, 306
  - Photo CD, 40
  - sample models, 477
  - shareware, 476-477
  - textures, 478
  - try-out software, 476
  - Wagstaff's Favorites, 478
- Change Mapping dialog box, 225
- child groups (scene building), 171
- Clip Model Libraries (Acuris), 423
- clip models, 76
- clipping paths, 368-370
- Collage (Specular), 458-459
- color
  - 4-bit format, 345
  - 8-bit format, 345-346
  - 16-bit format, 346
  - 24-bit format, 347
  - 32-bit format, 347
  - 64-bit format, 347
  - CD-ROM, 21
  - depth, 345-347
  - desktop publishing, 371
  - legalized colors, 305
  - lighting, 259
  - mapping, 209
  - monitors, 35, 46
  - NTSC video limitations, 372
  - surface qualities, 199-200
  - textures, 196
- color palettes, 379-380
- Color Style dialog box, 89
- ColorIt! (MicroFrontier), 363, 452
- commands
  - Animation menu, 298
  - Artwork menu
    - Make Guide, 78
    - Map Artwork, 82
  - Backgrounds menu, 93
  - Edit menu, 382
  - File menu, 214
  - Filter menu
    - Blur/Motion Blur, 360
    - KPT Texture Explorer, 240
  - Image menu, 353
  - Item menu, 370
  - Operations menu, 85
  - Selection menu, 356
  - View Mode menu, 84
- commercial software (CD-ROM), 475-476
- compositing images, 340-342
  - alpha channel, 342
  - Photoshop, 352-355
- compression in QuickTime format, 348-351

## compressors

- Animation-Millions of Colors, 350
- Animation-Millions of Colors+, 349
- Apple Cinepack, 350
- Apple Photo-JPEG, 350
- MPEG, 350
- None, 350
- Video, 350

## CompuServe, 434

- computer aided manufacturing (CAM), 10

- computer-aided design (CAD), 14, 16, 102

- computer-aided industrial design (CAID), 10

- cones (primitives), 145-146

- console creation (multimedia), 380-383

- constraining in scene building, 170-172

- constructive modeling, 26

  - extrusion, 104-106

  - lathing, 116-129

- coordinate system, 56-59

- CoSA's After Effects, 372

- Crystal TOPAS Mac

  - (CrystalGraphics), 435

- CrystalGraphics' Crystal TOPAS Mac, 435

- CrystalGraphics Inc., 434-435

- curved, hollow tubes (lathing), 129

- Custom Perspective palette (View menu), 81

- CyberMesh (Knoll Software), 447

- CyberMesh (scene building), 178-179

- cycling in event-based key framing, 273

## cylinders

- lathing, 122-124

- primitives, 143

- cylindrical mapping (texture mapping), 217-218

- CZAR (Integrated Device Technologies), 413, 445

---

## D

---

- DAT (Digital Audio Tape), 37

- daylight (lighting), 250

- DayStar Inc., 403, 435

- DeBabelizer (Equilibrium Technologies), 364-367, 438-439

- DeBabelizer Lite (Equilibrium Technologies), 439

- Decal mapping, 225

- decay (lighting), 259

- deformation (modeling), 26, 135-137

- DeltaGraph Professional

  - (DeltaPoint), 435-436

- DeltaPoint Inc., 435-436

- deltas, 72

- depth, 48

- depth of field (cameras), 191

- Design Reality (Ashlar), 431-432

- Design Workshop (Artifice), 430

- Designer (RayDream), 457

- Designer's painting, 220

- designing

  - packaging, 10-11

  - products, 10-11

- desktop publishing

  - clipping paths, 368-370

  - color, 371

- detail in rendering, 332

- dialog boxes

  - Camera Info, 286

  - Change Mapping, 225

  - Color Style, 89

  - Floor Setup, 93

Get Picture, 215  
 Make Movie, 282  
 Materials, 380  
 Motion Blur, 360  
 Position Material, 239  
 Diaquest Inc., 436  
 Diaquest's DQ-Animaq, 436  
 Diaquest's Mac-232, 436  
 diffuse reflection (surface quality),  
     203-204  
 Digital Audio Tape (DAT), 37  
 digital video hardware, 45  
 digitizing, 27, 43, 142  
 digitizing tablets, 40-43  
 Dimensions (Adobe), 78-86, 424  
 direction (texture mapping), 212  
 displacement, 196  
     mapping, 162  
     shaders, 389-390  
 display cards, 36  
 distant light, 257  
 distortion (modeling), 26  
 distributed rendering, 405  
     BackBurner, 408  
     DreamNet, 408  
     NetRenderMan, 407  
     Power Macintosh, 407  
     Render Pro, 408  
     Sketch!Net, 408  
 dithering, 345-346, 379  
 Doppler effect, 290  
 DQ-Animaq (Diaquest), 436  
 DQ-Animaq animation controller, 375  
 DQ-MAC232 animation controller,  
     374-375  
 Drawing eXchange Format (DXF), 397  
 drawing tablets, 40-43  
 DreamNet (RayDream), 408, 456  
 drop-off (lighting), 259  
 Duplicate command (Edit menu), 382

DXF (Drawing eXchange Format), 397  
 DynaPerspective (Dynaware USA),  
     437  
 Dynaware USA Inc., 437  
 Dynaware USA's DynaPerspective,  
     437

---

**E**

---

ease-in and out (animation), 291  
 Edit menu commands, 382  
 EIAS (Electric Image Animation  
     System), 283-284, 351, 375  
 Electric Image Animation System  
     (EIAS), 283-284, 375, 437  
     groups, 141  
     lighting, 250, 267  
     splines, 283  
     Velocity Graph window, 284  
 Electric Image Inc., 437  
 ElectroGig Inc., 438  
 ElectroGig's 3DGIG, 438  
 embossed type, 100  
 engraved type, 98-99  
 environment maps, 337  
 environmental reflection maps, 28  
 EPS (Encapsulated PostScript)  
     bitmaps, 78  
     format, 312, 343  
 Equilibrium Technologies, 364-367,  
     438-439  
 Escher (3D QuickDraw), 414-415  
 event-based animation, 274-285  
 event-based key framing  
     compared to scene-based key  
         framing, 272-274  
     cycling, 273  
 EVO-9650 animation controller,  
     374-375

- Exabyte drivers (animation controllers), 376
- extensions, turning off (hardware optimization), 46
- extent lines, 69
- Extreme 3D (Macromedia), 450-451
- Extrude command (Operations menu), 85
- extrusion (modeling), 104-106
  - basic extrusions, 106-107
  - blocks, 107
  - capped surfaces, 104
  - caps, 109
  - Illustrator outlines, 109-113
  - path extrusion, 114-115
  - slabs, 107
  - star-shaped tubes, 108-109

---

## F

---

- fall-off (lighting), 259
- Fetch (Aldus), 412, 426
- field rendering (video), 305, 376-377
- file formats, 342-344, 396-399
  - 1-bit, 345
  - 4-bit, 345
  - 8-bit, 345-346
  - 16-bit, 346
  - 24-bit, 347
  - 3D metafile, 416
  - 32-bit, 347
  - 64-bit, 347
  - DXF (Drawing eXchange Format), 397
  - ELAS (Electric Image Animation System), 351
  - EPS (Encapsulated PostScript), 343
  - IGES (Initial Graphic Exchange Specification), 398

- Image, 344
- JPEG (Joint Photographic Experts Group), 344
- numbered PICT, 348
- Photoshop, 344
- PICS, 351
- PICT2, 343
- polygons, 398-399
- QuickTime, 348-351
- RIB (Interface ByteStream), 398
- Scalable EPS, 343
- TIFF (Tagged Image File Format), 343
- File menu commands, 214
- file sharing
  - Fetch, 412
  - KA Share, 412
  - MacTCP, 412
  - NFS share, 413
  - Telnet, 413
- file translation (Cadmover), 399-400
- film
  - animation, 306-307
  - loops
    - animation, 300-304
    - morphing, 302
- Filter menu commands
  - Blur/Motion Blur, 360
  - KPT Texture Explorer, 240
- Flash Tracer (Video Bits), 413, 465
- flat shading (rendering), 32, 314
- flatbed scanners, 38
- flicker effect (video), 376
- Floor Setup dialog box, 93
- floppy disks, 37
- floptical drives, 37
- flying camera logos, 286-289
- flying logos, 277-283
- fog in scene building, 192-193
- foliage (modeling), 158-160



fonts, 75-100  
 Form and Function, 439  
 form•Z (Autodessys), 432-433  
 format translation vendors, 419  
 four-plane view (modeling), 26  
 Fractal Designs Inc., 439-440  
     Painter, 363-364, 439-440  
 Fractal Terrain Modeler (Strata), 460  
 fractals, 182  
 frame rate, 304  
 frame-by-frame recording (video),  
     373-376  
 freeware (CD-ROM), 476-477  
 Fresco (Xaos Tools), 469  
 fully animated cameras, 51

---

**G**

---

G-buffer (RayDream), 220-221  
 Gallery Effects (Aldus), 246, 427  
 gels  
     shadows, 264-267  
     StrataStudio Pro, 265-267  
 geometries (primitives), 76  
 Get Picture dialog box, 215  
 ghosting (hiding objects), 138  
 Gibbs & Associates, 440  
 glass refractions (rendering), 336  
 Glimpse utility, 393  
 glow (surface quality), 207-208  
     maps, 228-232  
     objects (lighting), 261-263  
 gobos (shadows), 264-267  
 Gouraud shading (rendering), 32, 315  
 grains (textures), 196  
 graphic arts, 8-9  
 Graphisoft  
     ArchiCAD, 440-441  
     Atlantis, 441  
     Zoom, 441

Graphsoft, 442  
 grids, 69, 170  
 groups  
     animation, 141  
     Electric Image Animation System,  
         141  
     modeling, 138-142  
     shapes, 142

---

**H**

---

hand-held scanners, 39  
 hard disks, 36, 38  
 hardware, 32-43  
     accelerators, 34  
     animation controllers, 44  
     animation video decks, 44  
     CD-ROM drives, 39-40  
     digital video hardware, 45  
     digitizers, 43  
     display cards, 36  
     drawing tablets, 40-43  
     hard disks, 36, 38  
     long-term storage, 36-37  
     monitors, 35  
     Power Macintosh, 33-34  
     scanners, 38-39  
     SuperMac 3D, 415-416  
     vendors, 419  
     video animation, 43-45  
     video encoders, 45  
 hardware optimization, 46  
 Hash Inc., 442-443  
 headlines, creating with addDepth,  
     87-89  
 hidden-line rendering, 312-313  
 hiding objects (ghosting), 138  
 highlights (spotlights), 255  
 HSC Software Inc., 443-444

humans, modeling, 157  
HyperCard Player (CD-ROM), 478  
HyperSpace (Mira Imaging), 452

---

I

---

IGES (Initial Graphic Exchange Specifications) format, 398  
igneous rock logo (StrataType 3d), 95-98  
ILC (Industrial Light and Magic), 386  
illustration, 8-9  
Illustrator (Adobe), 109-113, 424  
Image format, 344  
image galleries (CD-ROM), 477  
Image Hose (Painter), 363  
Image menu commands, 353  
image size (rendering), 336  
images, 339-384  
    animation, 371-372  
        formats, 347-351  
    color depth, 345-347  
    ColorIt! (MicroFrontier), 363  
    compositing, 340-342  
    DeBabelizer (Equilibrium), 364-367  
    desktop publishing, 367-371  
    file formats, 342-344  
    Painter (Fractal Design), 363-364  
    Photoshop (Adobe), 352-362  
    video, 372-377  
Immersion Corp., 444-445  
imported backgrounds, 184  
index of refraction, 206  
industrial design, 10-11  
Industrial Light and Magic (ILM), 386  
Infini-D  
    morphing, 297-299  
    scene building, 182  
    Specular, 459  
inserting textures, 235-236

Integrated Device Technologies Inc., 413, 445  
integrated modeling, rendering,  
    animation vendors, 420  
integrated still modeling and rendering vendors, 420  
intensity of lighting, 259  
interactive guide (CD-ROM), 474  
Interface Bytestream (RIB) format, 398  
interfaces, 30  
Intergraph Corp., 445-446  
interpolative antialiasing, 333  
inverse kinematics vendor, 421  
Item menu commands, 370

---

J-K

---

joining objects (modeling), 102-104  
joints (scene building), 172-173  
JPEG (Joint Photographic Experts Group) format, 344  
KA Share, 412  
Kai's Power Tools (KPT), 239-243, 443-444  
Kandu Software Corp., 399-400, 446-447  
key-frame animation, 31, 271-289  
    smoothing motion, 291-296  
Knoll Software, 447  
Knowledge Revolution, 447-448  
Knowledge Revolution's Working Model, 447-448  
KPT (Kai's Power Tools) texture creation, 239-243  
KPT Bryce (HSC Software), 186-188, 444  
KPT Texture Explorer command (Filter menu), 240

---

**L**


---

label maps, 28  
 labels, 232-233  
 lamps (lighting), 261-263  
 lathing (modeling), 116-129  
   axis of rotation, 119-122  
   common objects, 119  
   curved, hollow tubes, 129  
   cylinders, 122-124  
   partial lathes, 119, 124-126  
   pulleys, 128  
   techniques, 119-129  
   vases, 127  
 layering textures, 232-233  
 legalized colors, 305  
 lenses (cameras), 52-56  
 libraries of textures, 236  
 LifeForms (Macromedia), 449  
 lighting, 28, 247-268  
   ambient light, 255-256  
   back light, 258  
   color, 259  
   daylight, 250  
   decay, 259  
   distant light, 257  
   drop-off, 259  
   Electric Image Animation System,  
     250, 267  
   glowing objects, 261-263  
   intensity, 259  
   lamps, 261-263  
   night lights, 252  
   pan light, 257-258  
   point lights, 253-254  
   rendering, 337  
   setting lights, 258-260  
   shadows, 263-267  
   Sketch!, 259-260  
   spotlights, 254-255

  studio lighting, 251-252  
   time-of-day lighting, 268  
   visible lights, 261-263  
 Lighting Palette (Appearance menu),  
   81  
 linking in scene building, 170-172  
 living forms (modeling), 157-160  
 Load Selection command (Selection  
   menu), 356  
 loading the system, 46  
 lofting (modeling), 130-135  
 LogoMotion (Specular International),  
   98, 459  
 long-term storage, 36  
   animation, 37  
   DAT (Digital Audio Tape), 37  
   floppy disks, 37  
   floptical drives, 37  
 Looks (RenderMan), 390-391  
   compared to shaders, 391-392  
   Glimpse utility, 393  
 Looks CD (Pixar), 454  
 loops  
   animation, 300-304  
   morphing, 302

---

**M**


---

Mac-232 (Diaquest), 436  
 MacAnimator Pro (McQ), 451  
 Macintosh  
   compared to other platforms,  
     408-413  
   Power Macintosh, 3, 5, 13, 33-34  
 MacRageous II (YARC), 470-471  
 MacRenderMan  
   NetRenderMan, 387, 388  
   Render Monitor, 387  
   RenderApp, 387-388

MacRenderMan (Pixar), 32, 454

Macromedia Inc.

- Three-D, 449

- Extreme 3D, 450-451

- LifeForms, 449

- MacroModel, 450

- ModelShop II, 450

- Swivel 3D Professional, 448

- SwivelMan, 448

MacroModel (Macromedia), 450

MacTCP, 412

Make Guide command (Artwork menu), 78

Make Movie dialog box, 282

manually smoothing motion, 296

Map Artwork command (Artwork menu), 82

mapping

- bumps, 28, 221-227

- colors, 209

- environmental reflection maps, 28

- label maps, 28

- surfaces, 227-232

- textures, 27, 208-212

materials, 196-246

- bump maps, 221-227

- labels, 232-233

- procedural shading, 233-235

- projections, 212-221

- surface

  - maps, 227-232

  - quality, 197-208

- texture

  - creating, 236-246

  - inserting textures, 235-236

  - maps, 208-212

Materials dialog box, 380

Matte Object (Sketch!), 190-191

McQ, 451

metafiles (3D), 416

MicroFrontier's ColorIt!, 363, 452

microphones (Presenter Professional 3.0), 289-290

MicroStation Mac (Intergraph), 446

MiniCAD+ (Graphsoft), 442

Mira Imaging's HyperSpace, 452

mirror reflection

- ray tracing, 319

- surface quality, 200-201

- optics, 198

- rendering, 336

Missing Link (Knoll Software), 447

modelers

- CAD (computer-aided design), 164

- lofting, 130

modeling, 4, 25-27, 102-167

- additive modeling, 26, 102-104

- AMAP, 158-159

- animals, 157

- backgrounds, 161-162

- bevels, 149-150

- blocks, 143

- Boolean rendering, 162

- bump maps, 160

- CAD (computer-aided design), 162-164

- cones, 145-146

- constructive modeling, 26, 104-115

- cylinders, 143

- deformation, 135-137

- digitizing, 27

- displacement mapping, 162

- extrusion, 104-106

- four-plane view, 26

- groups, 138-142

- humans, 157

- lathing, 116-129

- libraries vendors, 421

- living forms, 157-160

- lofting, 130-133

molding, 26, 135-137  
 objects, joining, 102-104  
 ovoids, 144-145  
 plants and trees, 158-160  
 Playmation, 157-158  
 polygons compared to splines,  
     164-167  
 primitives, 143-149  
 pyramids, 147-149  
 scale, 77  
 single-perspective view, 25  
 solid modeling, 104, 151  
 spheres, 143-144  
 splines, 25, 164-167  
 surface modeling, 104  
 sweeping, 133-135  
 textures, 160-161  
 tori, 146-147  
 transparency, 161, 323-329  
 Tree Professional (Onyx), 159-160  
 utilities vendors, 421  
 models  
     placement (scene building), 170  
     samples (CD-ROM), 477  
     transporting, 396  
 ModelShop II (Macromedia), 450  
 Moiré patterns, 242  
 molding (modeling), 26, 135  
     PixelPutty Solo, 136  
     program support, 137  
     splines, 136  
 monitors, 35, 35, 46  
 morphing  
     animation, 296-299  
     film loops, 302  
     Infini-D, 297-299  
     tools, 302-304  
 motion blur, 360-362  
 Motion Blur dialog box, 360  
 motion pictures, 306-307

movable still image cameras, 51  
 moving  
     models, 396  
     objects, 66-70  
 MPEG compressor, 350  
 MPEG video, 306  
 multimedia, 13-14, 377-384  
     animation, 306  
     authoring tools, 378-379  
     color palettes, 379-380  
     console creation, 380-383  
     still image animation, 383-384  
 Multimedia+/Modeling & Animation  
     (CompuServe), 434  
 multiple computers (hardware  
     optimization), 46

---

**N**

---

native mode (Power Macintosh), 403  
 native software, 33  
 natural forms, 157-160  
 navigation, 48-73  
     absolute position, 56-59  
     cameras, 49-56  
     objects, moving and positioning,  
         66-70  
     orientation (pitch, yaw, roll), 71-72  
     relative position, 70  
     saving views, 73  
     three-plane view, 59-60  
 NetRenderMan  
     distributed rendering, 407  
     MacRenderMan, 387-388  
     Pixar, 455  
 NFS share, 413  
 night lights (lighting), 252  
 non-uniform rational b-splines  
     (NuRBS), 137

None compressor, 350  
NTSC safe zones, 372-373  
NTSC video  
    color limitations, 372  
    field rendering, 376-377  
numbered PICT format, 348  
NuRBS (non-uniform rational  
    b-splines), 137  
NuSprint (YARC), 470-471

---

## O

---

objects  
    bounding boxes, 66-68  
    grids, 69  
    hiding (ghosting), 138  
    joining for modeling, 102-104  
    moving and positioning, 66-70  
    point of view, 68  
    reference point, 70  
    showing, 138  
    smoothing, 400  
    snaps, 69  
online services  
    America Online, 429  
    CompuServe, 434  
    vendors, 421  
Onyx Computing, 452-453  
Onyx's Tree Professional, 453  
    modeling, 159-160  
Operations menu commands, 85  
optics, 198-199  
organic forms, 157-160  
orientation of cameras, 191  
orientation (pitch, yaw, roll), 71-72  
origin (universal reference point), 56  
orthogonal views, 56, 59  
ovoids (primitives), 144-145

---

## P

---

package design, 10-11  
Paint Alchemy (Xaos Tools), 246, 470  
Painter (Fractal Design), 363-364,  
    439-440  
painting, 219-221  
    Designer, 220  
    G-buffer (RayDream), 220-221  
    over QuickTime frames (Painter),  
        364  
    RayPainting, 219  
palettes, color, 379-380  
pan light, 257-258  
panning, 51  
parametric mapping (texture map-  
    ping), 217  
parent-child relationships (scene  
    building), 171  
partial lathes, 119, 124-126  
particle animation, 302-308  
path extrusion, 114-115  
PaXar Technologies Inc., 453-454  
PaXar's Touch 3D, 453-454  
PCs compared to Macintoshes,  
    408-413  
people modeling, 157  
Personal Digitizer (Immersion),  
    444-445  
perspective matching (Sketch!), 189  
perspective matching (scene build-  
    ing), 189-190  
perspective views, 56  
Phong rendering, 32, 316-318  
Photo CD (CD-ROM), 40  
Photoshop (Adobe), 425  
    compositing images, 352-355  
    file format, 344  
    motion blur, 360-362  
    seamless textures, 237-239  
    shadows, 356-359

- physics-based animation particles, 302-308
- PICS format, 351
- PICT format, 348
- PICT2 format, 343
- pitch (orientation), 71-72
- Pixar
  - 128, 454
  - Looks CD, 454
  - MacRenderMan, 32, 454
  - Movie Tool, 282
  - NetRenderMan, 455
  - Showplace, 455
  - Typestry, 455-457
  - Typestry, 90-95
- PixelPutty (Valis), 464
- PixelPutty Solo, 136, 464
- Place command (File menu), 214
- placing models (scene building), 170
- planar mapping (texture mapping), 213-216, 225
- plants (modeling), 158-160
- Playmation modeling, 157-158
- point lights, 253-254
- point of view
  - cameras, 51
  - objects, 68
- polygonal file formats, 398-399
- polygons
  - compared to splines (modeling), 164-167
  - converting to splines, 399
- position (texture mapping), 210
- Position Material dialog box, 239
- positioning objects, 66-70
- PostScript, 77-89
  - blending (rendering), 32, 311
- Power Macintosh, 3, 5, 13, 33-34
  - distributed rendering, 407
  - native mode, 403
  - real-time rendering, 34
  - rendering, 33
  - speed, 401-402
- Power Tools (CD-ROM), 475
- PowerPC chip, 24
- PowerPro 601 (DayStar), 435
- Premiere (Adobe), 371, 425
- presentations, 13-14, 421
- Presenter Professional, 134
  - VIDI, 284-285, 468-469
    - Attributes window, 285
    - microphones, 289-290
    - Script (timeline) window, 285
- primitives, 102, 103
  - blocks, 143
  - cones, 145-146
  - cylinders, 143
  - geometries, 76
  - modeling, 143-149
  - ovoids, 144-145
  - pyramids, 147-149
  - spheres, 143-144
  - tori, 146-147
- printing, 11
  - color, 371
- procedural shading, 90-91, 233-235
- product design, 10-11
- programming shaders, 393-394
- projections
  - cylindrical mapping, 217-218
  - parametric mapping, 217
  - planar mapping, 213-216
  - spherical mapping, 218
  - surface projections, 196
  - texture mapping, 212-221
- proportional mapping, 217
- pulleys, lathing, 128
- pyramids (primitives), 147-149

---

**Q**

---

QuickDraw (3D), 414-415  
QuickTime, 346  
    animation, 371-372  
    painting over frames with Painter, 364  
    sound in animation, 289  
QuickTime format, 348-351

---

**R**

---

radiosity (rendering), 32, 321-322  
ray tracing, 32, 318-321  
    mirror reflection, 319  
    recursion, 319-320  
    refraction, 320-321  
    transparency, 320  
RayDream Inc., 456-457  
    addDepth, 86-89, 456  
    G-buffer, 220-221  
    Designer, 457  
    DreamNet, 456  
RayPainting (rendering), 219, 322  
real-time rendering with Power Macintosh, 34  
realism in rendering, 331-333  
recursion (ray tracing), 319-320  
Reduced Instruction Set Computing (RISC), 401  
reference points, 70, 172  
reflection  
    ambient, 204-205  
    diffuse, 203-204  
    mirror, 200-201  
    optics, 198-199  
    Phong rendering, 318  
    specular, 202-203

reflectivity (mirror reflection), 200-201  
refraction (ray tracing), 320-321  
    surface quality, 206-207  
relative position, 70  
removable hard disks, 38  
Rend-X (Strata), 460  
Render Monitor (MacRenderMan), 387  
Render Pro (Strata), 408, 461  
RenderApp (MacRenderMan), 387, 388  
rendered backgrounds, 175-176  
rendering, 5, 14-16, 31-32, 309-337  
    antialiasing, 332-333, 337  
    atmospheric effects, 337  
    batch rendering, 337  
    Boolean rendering (modeling), 162, 330-331  
    detail, 332  
    distributed rendering, 405-408  
    environment maps, 337  
    field rendering, 376-377  
    flat shading, 32, 314  
    glass refractions, 336  
    Gouraud shading, 32, 315  
    hidden-line rendering, 312-313  
    image size, 336  
    lighting, 337  
    MacRenderMan (Pixar), 32  
    mirror reflections, 336  
    modeling with transparency, 323-329  
    Phong, 32, 316-318  
    PostScript blending, 32, 311  
    Power Macintosh, 33  
    radiosity, 32, 321-322  
    ray tracing, 32, 318-321  
    RayPainting, 322  
    real-time rendering, 34



realism, 331-333  
 RenderMan, 385-394  
   scene complexity, 336  
   smoothness, 334  
   speed, 335-337  
   texture maps, 337  
   utilities vendors, 422  
   vendors, 421  
   wireframe complexity, 333-334  
   wireframe rendering, 311-312  
 rendering and animation vendors, 419  
 rendering interface, 30  
 RenderMan, 385-394  
   Boolean rendering, 330-331  
   Looks, 390-391  
   MacRenderMan, 387-388  
   shaders, 386, 388-394  
   Unix, 413  
   Valis Group, 394, 465  
 RenderQ! (Sketch!), 272-274, 273-274  
 Replicas (Specular), 460  
 resolution  
   animation, 304  
   textures, 235-236  
 RIB (Interface Bytestream) format, 398  
 RISC (Reduced Instruction Set Computing), 401  
 roll (orientation), 71-72  
 rotation (scene building), 171-172  
 rotoscoping, 45  
   animation, 300  
   program support, 300  
 Runaround command (Item menu), 370

---

**S**


---

safe zones (video), 372-373  
 sample models (CD-ROM), 477  
 saving views, 73  
 Scalable EPS format, 343  
 scale, 77  
   texture mapping, 210-211  
 Scale command (Image menu), 353  
 scanners, 38-39  
 scene building, 4, 27-30, 169-193  
   atmosphere, 30, 191-193  
   backgrounds  
     imported, 184  
     rendered, 175-176  
     transparent, 188-189  
   bitmaps to models (backgrounds), 176-184  
   bump maps, 28  
   cameras, 30, 191  
   constraints, 170-172  
   CyberMesh, 178-179  
   fog, 192-193  
   grids, 170  
   Infini-D, 182  
   joints, 172-173  
   KPT Bryce, 186-188  
   label maps, 28  
   lighting, 28  
   linking, 170-172  
   model placement, 170  
   parent-child relationships, 171  
   perspective matching, 189-190  
   rendering interfaces, 30  
   rotation, 171-172  
   scenery generators, 184-188  
   shadow casting on backgrounds, 190-191  
   Showplace, 184  
   Sketch!, 189

- snaps, 170
- StrataStudio Pro, 179-182
- stretching and squashing, 174
- Terrainman, 177-178
- texture mapping, 27
- Vista Pro, 185-186
- scene complexity in rendering, 336
- scene-based key framing, 272-274
- scenery generators
  - KPT Bryce, 186-188
  - Vista Pro, 185-186
- scenery generators (scene building), 184-188
- Script (timeline) window, 285
- scripts (DeBabelizer), 365
- Sculpt 3D (Byte by Byte), 433-434
- Sculpt 4D (Byte by Byte), 434
- Sculptor (Byte by Byte), 434
- seamless textures, 209-210, 237-239
- Selection menu commands, 356
- sequencers (animation), 275-285
- setting lights, 258-260
- SGI (Silicon Graphics Inc.), 408-413, 457
- Shade III (YARC), 471
- Shaded command (View Mode menu), 84
- shading
  - animation, 392-393
  - compared to Looks, 391-392
  - displacement shaders, 389-390
  - Glimpse utility, 393
  - procedural shaders, 90-91
  - procedural shading, 233-235
  - programming, 393-394
  - RenderMan, 386, 388-394
  - surface shaders, 389
  - Valis Group, 394
  - volume shaders, 390
- shadows
  - backgrounds (scene building), 190-191
  - gels and gobos, 264-267
  - lighting, 263-267
  - Photoshop, 356-359
- shapes, 75-100
  - groups, 142
- shareware (CD-ROM), 476-477
- sharing files
  - Fetch, 412
  - KA Share, 412
  - MacTCP, 412
  - NFS share, 413
  - Telnet, 413
- showing objects, 138
- Showplace (Pixar), 455
- Showplace (scene building), 184
- Silicon Graphics Inc., 457
- single still cameras, 51
- single-perspective view, 25, 58-59, 61-65
  - cameras, 50
  - extent lines, 69
  - virtual trackball, 63-65
- Sketch! (Alias Research), 427-428
  - cameras, 49
  - Matte Object, 190-191
  - perspective matching, 189
  - RenderQ!, 272, 273
  - seamless textures, 237-239
  - setting lights, 259-260
- Sketch!Net, 408
- SketchTech, 458
- SketchTech's UpFront, 458
- skinning (modeling), 130-133
- slabs (extrusion), 107
- smoothing objects, 400

- smoothing motion
  - animation, 291-296
  - manually, 296
  - rendering, 334
- snaps, 69, 170
- Snapshot command (Animation menu), 298
- software, 6-7
  - commercial software (CD-ROM), 475-476
  - native software, 33
  - try-out software (CD-ROM), 476
  - walk-through software, 15
- solid modeler vendors, 422
- solid modeling, 104, 151
- solid shading, 233-235
- solid textures, 196
- sound
  - animation, 289-296
  - Presenter Professional 3.0's microphones, 289-290
  - synchronizing, 290-292
- Specular International, 458-460
  - Backburner, 458
  - Collage, 458-459
  - Infini-D, 459
  - LogoMotion, 98, 459
  - Replicas, 460
  - TextureScape, 243-245, 460
  - YARCBurner, 458
- specular reflection (surface quality), 202-203
- speed, 401-404
  - Daystar PowerPro, 403
  - Power Macintosh, 401-402
  - rendering, 335-337
  - YARC boards, 403-404
- spheres (primitives), 143-144
- spherical mapping (texture mapping), 218
- splines, 25
  - compared to polygons (modeling), 164-167
  - converting polygons to, 399
  - Electric Image Animation System, 283
  - molding, 136
  - spotlights (lighting), 254-255
  - squashing (scene building), 174
  - star-shaped extrusions, 108-109
  - stereolithography (3D printing), 11
  - stereoscopic vision, 48
  - still image animation, 383-384
  - still modeling and rendering (integrated) vendors, 420
- storage, 36-37
- Strata Inc., 460-463
  - Clip 3d, 461
  - Virtual, 463
  - Fractal Terrain Modeler, 460
  - Rend-X, 460
  - Render Pro, 461
  - Studio Pro, 462-463
- StrataStudio Pro (scene building), 179-182
  - bank turns, 292-295
  - gels, 265-267
  - sweeping, 134
- StrataType, 461
- StrataType 3d, 95-98
- StrataVision 3d, 461-462
- stretching (scene building), 174
- studio lighting, 251-252
- Studio Pro (Strata), 462-463
- SuperMac 3D, 415-416
- surface maps, 227-232
- surface modeler vendors, 422
- surface modeling, 104
- surface projection, 196
- Surface Properties palette (Appearance menu), 80

- surface qualities
  - ambient reflection, 204-205
  - color, 199-200
  - diffuse reflection, 203-204
  - glow, 207-208
  - mirror reflection, 200-201
  - refraction, 206-207
  - specular reflection, 202-203
  - transparency, 205-206
- surface quality, 197-208
- surface shaders, 389
- surface trimming (modeling), 26
- sweeping (lofting), 133-135
  - common shapes, 134-135
  - Presenter Professional, 134
  - StrataStudio Pro, 134
- switchable fixed-position cameras, 51
- Swivel 3D Professional (Macromedia), 448
- SwivelMan (Macromedia), 448
- synchronizing sound, 290-292

---

## T

---

- Tagged Image File Format (TIFF), 343
- Telnet, 413
- terrain modeling, 176-184
  - Cybermesh, 178-179
  - Infini-D, 182
  - Showplace, 184
  - StrataStudio Pro, 179-182
  - Terrainman, 177-178
- Terrainman (Art of Fact), 429
- Terrainman (scene building), 177-178
- Terrazzo (Xaos Tools), 245-246, 470
- Texture Explorer (Kai's Power Tools), 240-242
- texture libraries, 236

- texture mapping, 27
  - 3D painting, 219-221
  - color mapping, 209
  - cylindrical mapping, 217-218
  - direction, 212
  - parametric mapping, 217
  - planar mapping, 213-216
  - position, 210
  - projections, 212-221
  - rendering, 323-329
  - scale, 210-211
  - spherical mapping, 218
  - tiling, 211
- texture maps, rendering, 337
- textures
  - bump maps, 221-227, 236-239
  - bumps, 196
  - CD-ROM, 478
  - color, 196
  - creating, 236-246
  - displacement, 196
  - Gallery Effects (Aldus), 246
  - grains, 196
  - inserting, 235-236
  - Kai's Power Tools (KPT), 239-243
  - labels, 232-233
  - layering, 232-233
  - library vendors, 422
  - mapping, 208-212
  - modeling, 160-161
  - Paint Alchemy (Xaos Tools), 246
  - procedural shading, 233-235
  - resolution, 235-236
  - seamless textures, 209-210, 237-239
  - solid textures, 196
  - surface maps, 227-232
  - surface projection, 196
  - surface quality, 196-208
  - Terrazzo (Xaos Tools), 245-246
  - utilities vendors, 422-423

TextureScape (Specular), 243-245, 460  
 three-plane view, 59-60  
 TIFF (Tagged Image File Format), 343  
 tiling (texture mapping), 211  
 tilting, 51  
 time-of-day lighting, 268  
 timelines (animation), 275-285  
 tori (primitives), 146-147  
 Torque Systems Inc., 463-464  
 TorqueWare, 413, 463-464  
 Touch 3D (PaXar), 453-454  
 translating files with Cadmover, 399-400  
 translations, 72  
 transparency modeling, 161, 205-206, 320  
 transparent backgrounds, 188-189  
 transparent modeling (rendering), 323-329  
 transporting models, 396  
 Tree Professional, 159-160  
 Tree Professional (Onyx), 453  
 trees (modeling), 158-160  
 try-out software (CD-ROM), 476  
 Turbo 3D (View by View), 466, 471  
 types  
   animated type, 100  
   beveled type, 100  
   bump-mapped type, 100  
   embossed type, 100  
   engraved type, 98-99  
 Typestry  
   flying logos, creating  
 Typestry (Pixar), 90-95, 277-283, 455-457

---

**U-V**

---

Unix RenderMan, 413  
 UpFront (SketchTech), 458  
 Valis Group  
   PixelPutty, 464  
   PixelPutty Solo, 464  
   RenderMan shaders, 394, 464-465  
 vegetation (modeling), 158-160  
 Vellum 3D (Ashlar), 431  
 Velocity Graph window (Electric Image Animation System), 284  
 vendors  
   2D graphics, 418  
   3D CAD, 419  
   3D CAM, 418  
   3D PostScript, 418  
   animation, 418  
   animation and rendering, 419  
   architectural design, 419  
   format translation, 419  
   hardware, 419  
   integrated 3D modeling, rendering, animation, 420  
   integrated still modeling and rendering, 420  
   inverse kinematics, 421  
   model libraries, 421  
   modeling utilities, 421  
   online services, 421  
   presentation graphics, 421  
   rendering, 421  
   rendering utilities, 422  
   solid modelers, 422  
   surface modelers, 422  
   texture libraries, 422  
   texture utilities, 422-423  
   video effects, 423  
   virtual reality, 423

- video, 11-13, 372-377
  - alpha channel support, 342
  - animation, 305
  - animation controllers, 373-376
  - digital video hardware, 45
  - field rendering, 305, 376-377
  - flicker effect, 376
  - frame-by-frame recording, 373-376
  - hardware, 43-45
  - legalized colors, 305
  - NTSC video, 372
  - safe zones, 372-373
  - vendors, 423
- Video Bits, 465
- Video compressor, 350
- video decks, 44
- video encoders, 45
- VideoFusion, 371, 465-466
- VideoFusion Ltd., 465-466
- VIDI (Visual Information Development Inc.), 468
- VIDI's Presenter Professional, 284-285, 468-469
- View by View, 466
- View by View's AMAP, 466, 471
- View by View's Turbo 3D, 466
- View Mode menu commands Shaded, 84
- Viewpoint DataLabs, 467
- Viewpoint DataLabs' 3D datasets, 467
- views, 73
- Virtual Gibbs (Gibbs & Associates), 440
- virtual real-time cameras, 51
- virtual reality, 15, 17
  - animation, 307-308
  - animators, 274
  - vendors, 423
- virtual trackball, 63-65

- Virtus Inc.
  - vr, 467
  - Walkthrough, 307-308, 467-468
- visible lights, 261-263
- Vista Pro (scenery generator), 185-186
- Visual Information Development Inc., 468
- volume shaders, 390
- volume shading, 233-235
- vr (Virtus), 467

---

## W

---

- Wacom drawing tablets, 41
- Wagstaff's Favorites (CD-ROM), 478
- walk-through software, 15
- Walkthrough (Virtus), 307-308, 467-468
- Walkthrough Pro (Virtus), 307-308
- walkthroughs, 17
- wicked water (Typestry), 91-95
- Will Vinton's Playmation modeling, 157-158
- wireframe complexity (rendering), 333-334
- wireframe rendering, 311-312
- Working Model (Knowledge Revolution), 447-448
- Wrapture Reels (Form and Function), 439
- Wraptures (Form and Function), 439

---

**X**

---

x-axis (absolute position), 56-59  
Xaos Tools Inc., 469-470  
    Fresco, 469  
    Paint Alchemy, 246, 470  
    Terrazzo, 245-246, 470

---

**Y**

---

y-axis (absolute position), 56-59  
YARC boards, 403-404  
YARCBurner (Specular), 458  
YARC Systems Corp.  
    MacRageous II, 470-471  
    NuSprint, 470-471  
    Shade III, 471  
    Zuma, 470-471  
yaw (orientation), 71-72

---

**Z**

---

z-axis (absolute position), 56-59  
Zoom (Graphisoft), 441  
zooming, 51  
Zuma (YARC), 470-471



# About the CD

The CD-ROM included with this book has everything you need to get started in 3-D!

---

## Interactive 3-D Guide

Included on the disc is a full-color interactive guide to lighting, surface qualities, texture and environment mapping, rendering, and animation. Images and examples from the book are reproduced in color, along with many not found anywhere else.

## Alpha Channel BBS

This connection client will automatically log you onto the Alpha Channel BBS, the online support forum for the *3-D Starter Kit for Macintosh*.

## Absynthe's Power Tools

This is the complete library of Infini-D tips written and illustrated by Chris Bernardi (a.k.a., "Absynthe," on AOL). Since the Power Tools include embedded illustrations, you'll need Microsoft Word to read them.

## Commercial 3-D Software

You'll find 24 working 3-D programs from different vendors. These are "try before you buy" versions that enable you to do everything but save your work.

## Other Try-out Software

Because no 3-D program is an island, this disc includes demo versions of important software that supplements and enhances the capabilities of any 3-D application.

## Shareware & Freeware

There is a giant collection of shareware and freeware on the disc, including a number of fully-functional 3-D modelers and renderers, as well as a host of supporting utilities, such as terrain generators and fractal pattern makers.

## Image Galleries

Many sample images and animations are also included on the disc.

## Sample Models

Models used to create many of the illustrations in this book are included; you can open these models in most 3-D applications.

## Backgrounds and Textures

Every 3-D user needs more seamless textures, bump maps, and effects maps. These textures and effects maps can be used with any 3-D application that supports bitmapped textures. The disc contains two collections: Wagstaff's Favorites and the Alpha Channel.

## HyperCard Player

The HyperCard Player enables you to use the Interactive Macintosh 3-D Workshop.





# Everything you need to design and animate in 3-D!

The *3-D Starter Kit* gets you going fast with easy steps to the basics—plus all the software you need to take advantage of the best Macintosh® 3-D techniques!

- Build, render, and animate 3-D models
- Get up-to-date information on more than 100 Macintosh 3-D programs, hardware products, and utilities
- Discover advanced 3-D techniques from an expert
- Learn specific tips on 3-D cameras, modeling techniques, scene building, materials, lighting, and more
- Find out the latest 3-D news on Power Macintosh

You get the information you need to create your own 3-D projects. Sean Wagstaff, former Associate Editor of *MacWEEK*, knows the challenges you face and the solutions you need. Valuable notes, tips, cautions, and shortcuts guarantee the best 3-D results in the least time.

## Here's what people said about the previous edition, *Macintosh 3-D Workshop*!

"... a detailed look at 3-D on the Mac. Wagstaff, a veteran *MacWEEK* writer, covers the basics of 3-D, plus the hardware and software required."

**Computer Currents**

"... provides the information needed to get the most out of creating 3-D graphics on the Macintosh."

**Computer Graphics World**

# Other books just tell you about 3-D. This kit lets you do it yourself!

## Get great software you need to work in 3-D!



# LIGHT

- Macintosh 3-D Workshop Interactive—The exclusive guide to 3-D techniques!
- Over 15 working demos for creating stunning 3-D projects
- Try-out versions of StrataStudio Pro, formoz, Infini-D, Dimensions, and many more
- Shareware for working with terrains, sculpting, and texture design
- Galleries of sample images and animations
- 3-D models you can use in your own projects
- And much more!

System Requirements:

Macintosh or Power Macintosh  
System 7.0 or higher  
5M of RAM  
CD-ROM Drive



Hayden  
Books

INFORMATION  
CENTER:

Category: Graphics  
User Level: All Users

\$40.00 USA / \$54.95 CAN

ISBN



**SAL** Soft 85 Cover Book  
SIZE: NF \$1.67  
G

0 33931 04997 8