



Digital Compositing for Film and Video

Production Workflows and Techniques

Fourth Edition

Featuring new content:

Powerful New Keying Techniques

ACES and OpenColorIO

Working in Linear

VR Stitching

A **Focal Press** Book

Steve Wright



Digital Compositing for Film and Video

Written by senior compositor, technical director and master trainer Steve Wright, this book condenses years of production experience into an easy-to-read and highly informative guide suitable both for working and aspiring visual effects artists.

This expanded and updated edition of *Digital Compositing for Film and Video* addresses the problems and difficult choices that professional compositors face on a daily basis with an elegant blend of theory, practical production techniques and workflows. It is written to be software-agnostic, so it is applicable to any brand of software. This edition features many step-by-step workflows, powerful new keying techniques and updates on the latest tech in the visual effects industry.

Workflow examples for:

- Grain Management
- Lens Distortion Management
- Merging CGI Render Passes
- Blending Multiple Keys
- Photorealistic Color Correction
- Rotoscoping

Production techniques for:

- Keying Difficult Greenscreens
- Replicating Optical Lens Effects
- Advanced Spill Suppression
- Fixing Discolored Edges
- Adding Interactive Lighting
- Managing Motion Blur

With brand new information on:

- Working in Linear
- ACES Color Management
- Light Field Cinematography
- Planar Tracking
- Creating Color Difference Keys
- Premultiply vs. Unpremultiply
- Deep Compositing
- VR Stitching
- 3D Compositing from 2D Images
- How Color Correction Ops Affect Images
- Color Spaces
- Retiming Clips
- Working with Digital Cinema Images
- OpenColorIO

A companion website offers project files based on the examples discussed in the book, allowing readers to experiment with the material first-hand.

Steve Wright is a visual effects pioneer and a 20-year veteran of visual effects compositing on over 70 feature films and many broadcast television commercials. With extensive production experience and a knack for the math and science of visual effects he is a world-recognized expert on visual effects compositing. Since 2005 he has been a master trainer in compositing visual effects, providing staff training to over 25 visual effects studios around the world including Pixar Animation Studios, Disney Feature Animation, Troublemaker Studios, New Deal Studios, and Reliance MediaWorks, along with many others. He has also trained over 1,000 artists in compositing.

Visit Steve's training website at www.fxacademy.com



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Digital Compositing for Film and Video

Production Workflows and Techniques

FOURTH EDITION

Steve Wright

Fourth edition published 2018
by Routledge
711 Third Avenue, New York, NY 10017

and by Routledge
2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN

Routledge is an imprint of the Taylor & Francis Group, an informa business

© 2018 Taylor & Francis

The right of Steve Wright to be identified as the author of this work has been asserted by him in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

First edition published by Focal Press/Butterworth-Heinemann 2002
Third edition published by Focal Press 2010

Library of Congress Cataloging in Publication Data
A catalog record for this title has been requested

ISBN: 978-1-138-24036-0 (hbk)
ISBN: 978-1-138-24037-7 (pbk)
ISBN: 978-1-315-28401-9 (ebk)

Typeset in Helvetica Light
by Alex Lazarou

Visit the companion website: www.routledge.com/cw/wright



About the Authorxix
Acknowledgements.....xxi
Preface xxiii

Chapter 1 – Getting Started **1**

- 1.1 How this Book is Organized.....2
- 1.2 Web Content5
- 1.3 What’s New in the 4th Edition..... 5
- 1.4 Gold Mines.....9
- 1.5 Tool Conventions.....10
 - 1.5.1 The Slice Tool11
 - 1.5.2 Flowgraphs13
 - 1.5.3 Color Lookup Tables (LUTs).....14
 - 1.5.4 Nuke.....16
- 1.6 Data Conventions.....17
 - 1.6.1 Floating Point Data.....17
 - 1.6.1.1 Banding18
 - 1.6.1.2 Clipping.....19
 - 1.6.2 Linear Light Space.....19
 - 1.6.3 HDR Images.....20
 - 1.6.4 Stops.....20

PART I
MAKING A GREAT COMPOSITE..... **21**

Chapter 2 – Pulling Keys **23**

- 2.1 Luma Keys24
 - 2.1.1 How Luma Keys Work.....25
 - 2.1.2 Making Your Own Luminance Image27
 - 2.1.2.1 Variations on the Luminance Equations.....27

2.1.2.2	Non-luminance Monochrome Images.....	29
2.1.3	Making Your Own Luma Keyer.....	29
2.2	Chroma Keys	32
2.2.1	How Chroma Keys Work.....	32
2.2.2	Making Your Own Chroma Keyer.....	33
2.2.3	Making a 3D Chroma Keyer.....	34
2.3	Difference Mattes.....	36
2.3.1	How Difference Mattes Work.....	37
2.3.2	Making Your Own Difference Matte.....	39
2.3.2.1	Making the Difference Image.....	39
2.3.2.2	Making the Difference Matte.....	40
2.4	Bump Mattes.....	41
2.5	Color Difference Keys	43
2.6	The “Blur and Grow” Technique.....	46
2.7	Rotoscoping.....	47
2.7.1	Control Point Coherency	48
2.7.2	Shape Breakdown.....	48
2.7.2.1	Hierarchical Articulation.....	49
2.7.2.2	Organization.....	50
2.7.3	Bezier or B-spline?.....	51
2.7.4	Keyframe Strategies.....	52
2.7.4.1	On 2s	52
2.7.4.2	Binary Multiples	52
2.7.4.3	Bifurcation	53
2.7.4.4	Motion Extremes	54
2.7.5	Motion Blur	55
2.7.5.1	Spline Placement	55
2.7.5.2	Edge Decontamination	57
2.7.6	Inspection	60
Chapter 3	– Working with Keyers	63
3.1	Keyers.....	63
3.2	How Keyers Work.....	64
3.2.1	Calculating the Color Difference Matte	65
3.2.1.1	The Theory	65
3.2.1.2	Pulling the Raw Matte	66
3.2.1.3	A Simplified Example	66
3.2.1.4	A Slightly More Realistic Case.....	68
3.2.1.5	And Now, the Real World.....	70
3.2.1.6	Matte Edge Penetration	70
3.2.2	Scaling the Raw Matte	71
3.3	The After Effects Keyer	74

3.3.1	Step-by-step Procedure	74
3.3.2	Flowgraph of the After Effects Keyer	76
3.4	Typical Greenscreen Problems	78
3.4.1	Overexposed.....	78
3.4.2	Underexposed	79
3.4.3	Impure Greenscreens	81
3.4.4	Uneven Lighting.....	82
3.5	Preprocessing the Greenscreen	84
3.5.1	Denoise and Degrain	85
3.5.2	Screen Leveling	87
3.5.3	Local Suppression	89
3.5.4	Channel Clamping	91
3.5.5	Channel Shifting.....	93
3.5.6	Screen Correction	94
3.5.6.1	Step-by-step Procedure.....	95
3.5.6.2	Pictographic Flow Chart	98
3.5.6.3	Flowgraph of the Screen Correction Procedure	98
3.5.6.4	How to Create a Clean Greenscreen	99
Chapter 4 – Refining Mattes		103
4.1	Gamma Slamming	104
4.2	Garbage Mattes	105
4.2.1	Pre-matting.....	106
4.2.2	Post-matting	107
4.3	Filtering the Matte	109
4.3.1	Noise Suppression with a Median Filter.....	109
4.3.2	Softer Edges	111
4.3.3	Controlling the Blur Operation.....	112
4.3.3.1	The Blur Radius.....	112
4.3.3.2	The Blur Percentage	113
4.3.3.3	Masking the Blur	113
4.4	Adjusting the Matte Size	113
4.4.1	Eroding a Matte with Blur and Scale.....	114
4.4.2	Dilating a Matte with Blur and Scale	116
4.4.3	Blurring Out.....	116
4.4.4	Sculpting Edges.....	117
4.5	Edge Masks	119
Chapter 5 – Spill Suppression.....		121
5.1	Sources of Spill	122
5.2	The Despill Operation	123
5.3	Despill Algorithms	124

5.3.1	Green Limited by Red	125
5.3.1.1	Implementing the Algorithm.....	127
5.3.1.2	The Spillmap	128
5.3.2	Green Limited by the Average of Red and Blue	129
5.3.3	An Adjustable Despill.....	130
5.3.4	What About Blue Spill?.....	130
5.3.5	Refining the Despill	130
5.3.5.1	Channel Shifting.....	131
5.3.5.2	Spillmap Scaling	131
5.3.5.3	Mixing Despills.....	131
5.3.5.4	Matting Despills Together	131
5.4	The Unspill Operation	131
5.4.1	How to Set It Up	132
5.4.2	Grading to the Backing Color	133
5.5	Despill Artifacts.....	134
5.5.1	Finding the Artifacts	134
5.5.2	Hue Shifts	135
5.5.3	Dark Edges	136
5.5.4	Fixing Despill Artifacts	136
5.6	Edge Grading	138
5.7	Edge Extension.....	140
Chapter 6 – The Composite		143
6.1	Premultiply vs. Unpremultiply.....	144
6.1.1	Premultiply.....	145
6.1.2	Unpremultiply	146
6.1.3	The Double Premultiply	147
6.2	The Composite.....	148
6.2.1	The Over Composite	148
6.2.2	The KeyMix Composite	149
6.2.3	The AddMix Composite.....	149
6.2.3.1	How It Works	150
6.2.3.2	How to Build It.....	151
6.2.3.3	How to Use It.....	151
6.2.4	The Processed Foreground Method	153
6.2.4.1	The Workflow.....	154
6.2.4.2	What to Watch Out For	155
6.3	Compositing With a Keyer	156
6.3.1	Soft Comp/Hard Comp.....	156
6.3.2	“Cut and Paste” Keyer Compositing	158
6.4	Compositing Outside the Keyer.....	159
6.4.1	The Single Key	160

6.4.2	The Uberkey.....	161
6.4.3	Soft Key/Hard Key.....	162
6.4.4	The Additive Keyer.....	164
6.5	Stereo Compositing	165
6.5.1	Anaglyph.....	166
6.5.2	Stereopsis	167
6.5.3	Stereoscopy	168
6.5.4	The Stereo Conversion Process.....	169
6.5.5	Depth Grading	171
6.5.5.1	Scene Transition.....	171
6.5.5.2	The Dashboard Effect	172
6.5.5.3	Window Violation.....	172
6.5.5.4	Miniaturization.....	173
6.5.5.5	Divergence.....	173
6.5.6	Stereo Compositing.....	173
6.5.6.1	Dual View Display	174
6.5.6.2	Split and Join Views.....	175
6.5.6.3	Disparity Maps.....	175
Chapter 7 – Compositing CGI.....		177
7.1	Multi-pass CGI Compositing.....	178
7.1.1	Process Verification for Your Renderer.....	179
7.1.2	Render Passes.....	180
7.1.3	Lighting Passes.....	181
7.1.3.1	Render Passes Workflow	182
7.1.3.2	Beauty Pass Workflow.....	184
7.1.4	AOVs.....	185
7.1.5	ID Passes	187
7.1.6	Normals Relighting.....	189
7.2	EXR File Format	191
7.2.1	Film Scans.....	191
7.2.2	Linear Lightspace	192
7.2.3	Arbitrary Image Channels	192
7.3	HDR Images.....	193
7.4	Deep Compositing	195
7.4.1	Deep Images	196
7.4.2	The Layering Complexity Problem.....	197
7.4.3	The Depth Compositing Edge Problem	198
7.4.4	The Re-rendering Problem.....	199
7.4.5	Deep Compositing with Live Action.....	200

Chapter 8 – 3D Compositing	203
8.1 A Short Course in 3D	204
8.1.1 The 3D Coordinate System	204
8.1.2 Vertices	205
8.1.3 Meshes.....	206
8.1.4 Surface Normals	206
8.1.5 UV Coordinates.....	208
8.1.6 Map Projection	208
8.1.7 UV Projection.....	209
8.1.8 3D Geometry.....	211
8.1.9 Geometric Transformations	212
8.1.10 Geometric Deformations	213
8.1.10.1 Image Displacement.....	213
8.1.10.2 Noise Displacement.....	214
8.1.10.3 Deformation Lattice.....	215
8.1.11 Point Clouds.....	215
8.1.12 Lights	217
8.1.13 Shaders.....	217
8.1.14 Reflection Mapping.....	219
8.1.15 Ray Tracing	220
8.1.16 Image-based Lighting.....	221
8.1.17 Cameras.....	222
8.2 3D Compositing	223
8.2.1 3D Compositing from 2D Images.....	223
8.2.2 Pan and Tile	224
8.2.3 Camera Projection.....	226
8.2.4 Multiplane Shots.....	227
8.2.5 Set Extension	229
8.2.6 3D Backgrounds	230
8.3 Alembic Geometry	232
8.3.1 The Simple Case.....	233
8.3.2 Scenegraphs.....	233
8.3.3 Advantages Over FBX	234
8.4 Camera Tracking.....	235
8.4.1 Step 1 – Feature Tracking	236
8.4.2 Step 2 – The Solve	237
8.4.3 Step 3 – Build the Scene.....	238
8.4.4 Placing the Geometry	239
8.4.5 A Large Outdoor Scene	240

PART II	
THE QUEST FOR REALISM	243
Chapter 9 – Color Correction	245
9.1 The Behavior of Light	246
9.1.1 The Inverse Square Law	246
9.1.2 Diffuse Reflections	247
9.1.3 Specular Reflections	248
9.1.4 Bounce Light	249
9.1.5 Scattering	249
9.2 Gamma	251
9.2.1 The Math	251
9.2.2 Why Do We Need Gamma?	252
9.3 The Affect of Color Operations	254
9.3.1 Lift	255
9.3.2 Gamma	256
9.3.3 Gain	257
9.3.4 Offset	258
9.3.5 Saturation	259
9.3.6 Color Grading vs. Color Correcting	260
9.3.7 Increasing Contrast with the “S” Curve	260
9.3.8 Histograms	261
9.3.9 Channel Swapping	263
9.3.10 Premultiply vs. Unpremultiply – Again	264
9.4 Matching the Light Space	265
9.4.1 Brightness and Contrast	266
9.4.1.1 Matching the Black and White Points	267
9.4.1.2 Matching the Midtones	270
9.4.1.3 Gamma Slamming	271
9.4.2 Matching Color	272
9.4.2.1 Grayscale Balancing	272
9.4.2.2 Flesh Tones	273
9.4.2.3 The “Constant Green” Method of Color Correction	273
9.4.2.4 Daylight	274
9.4.2.5 Specular Highlights	274
9.4.3 Lighting Direction	275
9.4.4 Quality of Light Sources	275
9.4.4.1 Creating Softer Lighting	276
9.4.4.2 Creating Harsher Lighting	276
9.4.5 Non-linear Gradients for Color Correction	276
9.4.6 The DI Process	278
9.4.7 A Checklist	279

Chapter 10 – Sweetening the Comp	281
10.1 Layer Integration	282
10.2 Interactive Lighting	283
10.3 Edge Blending	285
10.4 Light Wrap.....	287
10.5 Creating Shadows.....	289
10.5.1 Edge Characteristics	289
10.5.2 Density	290
10.5.3 Color.....	291
10.5.4 Faux Shadows.....	292
10.5.5 Shadow Warping.....	293
10.5.6 Contact Shadows.....	294
10.6 Atmospheric Haze	296
10.7 Adding a Glow	299
10.8 Grain Management	300
10.8.1 Grain Characteristics	300
10.8.2 Regraining Techniques	301
10.8.2.1 Regrain Tool	301
10.8.2.2 Lifted Grain	302
10.8.2.3 Grain Rescue	302
10.8.3 Grain Management Workflows	304
10.8.3.1 Live Over Live	304
10.8.3.2 Live Over CGI	305
10.8.3.3 CGI Over Live	305
10.8.3.4 CGI Over CGI.....	305
10.8.3.5 Still Photos.....	305
10.9 Managing Clipping	305
 Chapter 11 – Camera Effects	 309
11.1 Lens Effects	310
11.1.1 Lens Distortion	311
11.1.2 Depth of Field.....	312
11.1.3 Vignetting	313
11.1.4 Lens Defects	314
11.1.4.1 Spherical Aberration	314
11.1.4.2 Astigmatism	315
11.1.4.3 Chromatic Aberration.....	316
11.1.5 Glows and Flares	316
11.1.5.1 Lens Flare	317
11.1.5.2 Lens Filter Flare.....	318
11.1.5.3 Diffraction Glows.....	320
11.1.5.4 Veiling Glare.....	321

11.1.6	Grain	322
11.2	Lens Distortion Workflows.....	322
11.2.1	CGI Over Live Action	323
11.2.2	Live Action Over CGI	325
11.2.3	CGI Over CGI.....	325
11.2.4	Live Action Over Live Action.....	326
11.3	Matching the Focus	327
11.3.1	Using a Blur for Defocus.....	327
11.3.2	How to Simulate a Defocus.....	329
11.3.3	Sharpening.....	330
11.3.3.1	Sharpening Operations.....	330
11.3.3.2	Unsharp Masks.....	331
11.3.3.3	Making Your Own Unsharp Mask	332
11.4	Rolling Shutter.....	333

PART III**THINGS YOU SHOULD KNOW..... 337****Chapter 12 – Digital Color..... 339**

12.1	Color Spaces	340
12.1.1	Primary Chromaticities	340
12.1.2	Units of Measure	341
12.1.3	Transfer Function.....	341
12.1.4	Gamut	341
12.1.5	HSV and HSL	342
12.1.6	Log and Linear.....	342
12.2	Working in Linear	343
12.2.1	What Exactly Is Linear?.....	344
12.2.2	Color Operations.....	345
12.2.3	Transformations and Filtering.....	346
12.2.4	CGI.....	347
12.3	Metadata	349
12.4	OpenColorIO.....	350
12.5	ACES Color Management.....	351
12.5.1	The ACES Workflow	351
12.5.2	The ACES Gamut	353
12.5.3	What About Video Productions?	353

Chapter 13 – Image Blending	355
13.1 Image Blending in Linear Light Space	356
13.1.1 Image-blending Operations	356
13.1.2 Compositing Operations	358
13.1.3 Matching the Look of sRGB in Linear	359
13.1.3.1 All sRGB Color Space	359
13.1.3.2 sRGB Within Linear	360
13.2 Alpha Compositing Operations	361
13.3 Image-blending Operations	362
13.3.1 The Screen Operation	362
13.3.1.1 Adjusting the Appearance	364
13.3.2 The Weighted Screen Operation	365
13.3.3 Multiply	366
13.3.3.1 Adjusting the Appearance	368
13.3.4 Maximum	369
13.3.5 Minimum	370
13.3.6 Absolute Difference	371
13.4 Adobe Photoshop Blending Modes	372
13.4.1 Simple Blending Modes	372
13.4.2 Complex Blending Modes	373
13.5 Slot Gags	376
13.6 Retiming Clips	377
13.6.1 Constant Speed Changes	377
13.6.2 Variable Speed Changes	378
13.6.3 Interpolation Methods	380
13.6.3.1 Nearest Neighbor	381
13.6.3.2 Frame Average	381
13.6.3.3 Motion Estimation	382
13.7 VR Stitching	384
13.7.1 Workflow Overview	386
13.7.2 Removing Lens Distortion	388
13.7.3 Building a Matching Computer Rig	389
13.7.4 Projecting Onto the Panosphere	390
13.7.5 The Stitching Process	392
13.7.6 Coping with Parallax	393
13.7.7 Exposure Correction	395
13.7.8 Visual Effects	396

Chapter 14 – Transforms and Tracking	401
14.1 Geometric Transforms.....	402
14.1.1 2D Transforms.....	402
14.1.1.1 Translation.....	402
14.1.1.2 Rotation.....	404
14.1.1.3 Resize vs. Scale.....	406
14.1.1.4 Skew	407
14.1.1.5 Corner Pinning	408
14.1.2 Managing Motion Blur.....	410
14.1.2.1 Transform Motion Blur.....	410
14.1.2.2 Motion UV Motion Blur	411
14.1.2.3 Speed Changes.....	412
14.1.3 3D Transforms.....	413
14.1.4 Filtering	414
14.1.4.1 The Effects of Filtering	414
14.1.4.2 Twinkling Starfields	415
14.1.4.3 Choosing a Filter.....	416
14.1.5 Lining Up Images.....	417
14.1.5.1 Offset Mask Lineup Display.....	417
14.1.5.2 Edge-detection Lineup Display	418
14.1.5.3 The Pivot Point Lineup Procedure.....	420
14.2 Image Displacement.....	422
14.3 Warps and Morphs.....	423
14.3.1 Mesh Warps	423
14.3.2 Spline Warps	424
14.3.3 Morphs	425
14.3.4 Tips, Tricks, and Techniques.....	427
14.4 Point Tracking	428
14.4.1 The Tracking Operation.....	429
14.4.1.1 Selecting Good Tracking Targets.....	430
14.4.1.2 Bad Tracking Targets	431
14.4.1.3 Tracker Enable/Disable.....	432
14.4.1.4 Offset Tracking.....	432
14.4.1.5 Keep Shape and Follow Shape	433
14.4.1.6 Pre-processing the Clip	435
14.4.1.7 Coping with Grain.....	436
14.4.1.8 Tracking Workflow.....	437
14.4.1.9 Cleaning up Tracking Data	438
14.4.1.10 The Stability Test	439
14.4.1.11 Reasons for Failure	439
14.4.2 Match Move	441
14.4.2.1 2D Transforms.....	442

14.4.2.2	Corner Pinning	442
14.4.3	Stabilizing.....	444
14.4.3.1	The Repo Problem	444
14.4.3.2	Motion Smoothing	446
14.4.3.3	Stabilizing For Rotoscoping	446
14.5	Planar Tracking	448
14.5.1	The Planar Grid	450
14.5.2	Drift Correction	450
14.5.3	Exporting Data	451
14.5.4	Roto Assist	451
Chapter 15 – Digital Images.....		453
15.1	HD Video.....	454
15.1.1	Frame Formats	455
15.1.2	Anamorphic Video.....	456
15.1.3	Scan Modes	457
15.1.4	Working with Interlaced Video	458
15.1.4.1	De-interlacing.....	459
15.1.4.2	Scan Line Interpolation	460
15.1.4.3	Field Averaging.....	460
15.1.5	Color Subsampling.....	461
15.1.6	Keying with 4:2:2 Video.....	463
15.1.7	Frame Rates	465
15.1.7.1	24, 25, 30, 60fps	465
15.1.7.2	23.98, 29.97, 59.94fps	466
15.1.8	Timecodes	466
15.1.9	Video File Formats	467
15.1.10	Telecine	467
15.1.10.1	The 3:2 Pull-down	468
15.1.10.2	The 3:2 Pull-up	469
15.2	Digital Cinema Images.....	470
15.2.1	Digital Camera Advantages	471
15.2.2	The Bayer Array	472
15.2.3	Sensor Crop	473
15.2.4	HFR – High Frame Rate	473
15.2.5	The DCI	474
15.3	Film Scans.....	474
15.3.1	Grain	475
15.3.2	The “Safe-to” Window	476
15.3.3	Apertures	476
15.3.4	Aspect Ratios.....	477
15.3.5	Film Formats.....	478

15.3.5.1	Full Aperture.....	478
15.3.5.2	Academy Aperture.....	479
15.3.5.3	Super 35 Formats	479
15.3.5.4	Cinemascope.....	480
15.3.5.5	Working with Cscope	481
15.3.5.6	“3-perf” Film	482
15.3.5.7	VistaVision.....	483
15.3.5.8	65mm/70mm	483
15.3.5.9	IMAX	484
15.4	Log Images	485
15.4.1	What Are Log Images?.....	485
15.4.2	Why We Need Log Images	487
15.4.2.1	Human Vision	487
15.4.2.2	Data Compression	488
15.4.2.3	Working with Log Images	488
15.5	Light Field Cinematography.....	492
15.5.1	How It Works	493
15.5.2	The Impact on Visual Effects.....	494
15.5.2.1	Deep Images	494
15.5.2.2	Arbitrary Depth of Field.....	495
15.5.2.3	Depth Maps	495
15.5.2.4	Stereo Through a Single Lens.....	496
15.5.2.5	Volumetric Optical Flow	496
15.5.2.6	Position Pass.....	497
15.5.2.7	Point Clouds.....	497
15.5.2.8	Mattes	497
15.5.2.9	Normals and Normal Relighting.....	498
15.5.2.10	Camera Tracking.....	498
15.5.3	When, and How Much?.....	499
	<i>Glossary</i>	501
	<i>Index</i>	537



Taylor & Francis

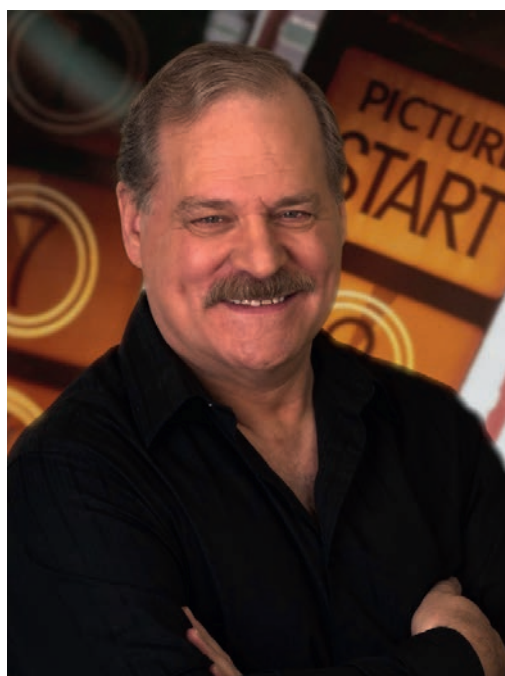
Taylor & Francis Group

<http://taylorandfrancis.com>

About the Author

Steve Wright is a visual effects pioneer and a 20-year veteran of visual effects compositing on over 70 feature films as well as 70 broadcast television commercials. With extensive production experience and a knack for the math and science of visual effects he is a world-recognized expert on visual effects compositing.

Since 2005 he has been a master trainer in compositing visual effects providing staff training to over 25 visual effects studios around the world including Pixar Animation Studios, Disney Feature Animation, Troublemaker Studios, New Deal Studios, Reliance MediaWorks, and many others. He has hundreds of video tutorials on Lynda.com, the world's leading online education company, has won 2 Telly awards for his tutorials, a SMPTE (Society of Motion Picture and Television Engineers) Education award, and has created training materials for companies like the Foundry, Red Giant, Imagineer, and YUVSoft. He has students all around the world and to date has trained over 1,000 artists in compositing, and published two books on compositing available on Amazon.com.



Visit Steve's training website at www.fxecademy.com

Awards:

2014 – SMPTE Kodak Educational Award for outstanding VFX training

2010 – Telly award for outstanding tutorial videos – Lynda.com

2010 – Telly award for outstanding tutorial videos – CG Society

2008 – Telly award for best VFX for a fantasy short film DORME

2006 – President's Volunteer Service Award for VFX training in Macedonia

1993 – PROMAX Gold Medallion Award for ACTION 7 News, Florida, campaign

1992 – Regional Emmy for CGI in MASTERS OF ILLUSION for The National Academy of Television Arts and Sciences.

1991 – Dentsu Award (Japanese CLIO) for outstanding CGI commercial REGAL SHOE



Acknowledgements

One of the great challenges to writing a book on compositing visual effects is that you need footage – lots of footage. You need greenscreens, background plates, digital matte paintings, and CGI renders. Fortunately there are many people out there that are willing to share their work so that all of us in the visual effects industry may benefit. To these talented and generous filmmakers I owe many thanks.

At the top of my list is Jon Karafin of LYTRO, Inc. Jon and I became fast friends starting in 2009 when he was the Manager of Production Operations for In-Three, one of the founding companies of stereo conversion. A brilliant man, he holds multiple patents in the field and cheerfully indulged my fascination for the math and science of stereo filmmaking that appears in this book. I turned to Jon again to add a section on light field cinematography and by this time he was the Head of Light Field Video at LYTRO, Inc. As the inter-galactic expert in this dazzling new technology, I could not have had a better mentor and he generously provided the one-of-a-kind light field images that grace the pages of this book.

I must also thank Marcin Biegunaajtys, the energetic owner of Manmade Media Studio in Denver. He generously fired up his great and powerful RED Dragon for an all-day green-screen shoot that provided several pictures used in this book. I hope to return the favor one day by learning how to correctly pronounce his last name.

Nick Fredin is the very talented 3D animator that designed and modeled the 3D Mantis character that graces the cover of this book and appears in both wireframe and render form within its pages. His Mantis character appears in a CG corridor designed and rendered by Pascal Bruguière, one of my most-talented students, currently working at Motion Picture Company. Many thanks to Pascal for his superb designs and stellar renders.

It is surprisingly difficult to find that perfect photographic example to illustrate a teaching point, so a vast library of photographs is essential. Several generous and talented shutterbugs have donated their photographs to my library so I would like to thank Pavel Dvorak, Bronwen Aker, and my own talented wife Diane Wright for their photographic contributions to this book. I would also like to thank Diane for her tremendous support and assistance in the writing of this, the 4th edition.

While color science has been an interest of mine for years, when I needed an expert to check my writings on OpenColorIO and the ACES color management system I turned to my long-time friend Doug Walker, a leading color scientist and the Color Technology Lead for Autodesk.

Similarly, when I wanted to add an important new section on VR stitching I needed industry experts to review my writings so I received the generous assistance of Sean Jamieson, the VR Editor at Digital Domain, Grant Anderson, the Head of Studio and Executive Producer at Jaunt, Daniele Pugni, the VFX Supervisor for Sky TV's 3D projects, and Steve Cooper of Lytro, Inc.

And finally to my over 1,000 students around the world that I have had the honor to teach visual effects compositing, this book is dedicated to them. Their inquisitive minds and need to know how things work have inspired the contents of this book and helped me to really understand what students need to know to make it in the industry today.



This book is about hands-on visual effects compositing techniques, in the quest for photorealism for feature film and HDTV visual effects. This, the fourth edition, is not just an update but a major rewrite with three major thrusts – step-by-step workflow examples, the addition of several powerful new techniques for keying, and the latest VFX industry tech. I have 20 years “in the chair” doing visual effects compositing for over 70 feature films so I have brought a wealth of real-world production experience to this book. Due in large part to the first edition of this book, in 2005 I turned to teaching and training visual effects compositing and developed a unique understanding of what compositing artists really need to know to achieve employment in the visual effects industry.

Of course, artistic training is an essential ingredient to achieving photorealism since it is in the art class that you learn how things are supposed to look. But the other requirements are a mastery of your tools and of technique. Knowing what the picture should look like will do little good if you can't bend the images to your vision. Reading the owner's manual will teach you how to operate your compositing software, but it will not teach you how to pull a good key from a badly lit bluescreen, or what to do when banding erupts in a digital matte painting. It is the difference between the owner's manual for a car and driving school. The first teaches you where the knobs are and the second how to use them to actually get somewhere.

While very suitable for beginners, this is not an introductory book. It assumes that the reader already knows how to operate the compositing software, what a pixel is, and what RGB means. It is intended for the digital artist sitting in front of a workstation with a picture on the monitor wondering why there are edge artifacts in a bluescreen composite and how to get rid of them. The attempt was to present the topics with sufficient background detail that they will be of value to the beginner, while including advanced concepts and production techniques that will also be of value to the more-experienced compositor.

This is also a software-agnostic book. It carefully avoids the specifics of any particular brand of software and uses procedures and techniques such as rotoscoping, tracking and color correcting that are common to all software products. You will be able to perform all of the procedures in this book regardless of the brand of software that you are using, including Adobe Photoshop. Photoshop has become an important tool for pre-visualizing visual effects shots, so it would be very useful to know how to use it to pull a color difference matte and perform the despill for a bluescreen shot. Better yet, we will see how to take the art director's "look dev" where he previewed a shot for the nice client in Photoshop and match it perfectly in your compositing software even though it works in a different color space and may not even have some of Photoshop's advanced blend modes. No problem.

Each topic has two thrusts. The first is understanding how things work. It is much more difficult to solve a problem when you don't know what is causing it, and a large part of time spent in comping a shot is problem solving. There is a great deal of information on different ways to pull a key, how despill works, and what goes on inside of a compositing node. The purpose of all of this information is to provide sufficient understanding of the issues that conspire to spoil your fine work, so that when you encounter a production problem you will be able to stab your finger in the air and declare "I know what's wrong here!", then turn around and fix it straight away instead of fumbling around for hours, or worse, introducing compensating errors in a vain attempt to make the problem go away.

The second thrust is production technique. How to sweeten a bluescreen so your keyer will pull a better key, how to do photorealistic color correction, how to manage lens distortion between the different layers of a comp, to cite just a few examples. There are dozens of step-by-step procedures to walk you through complex workflow issues like grain management for a multilayer comp, or the proper math ops to use for compositing multi-pass CGI renders. There is an entire section on the various types of artifacts introduced by lenses with the procedures for how to replicate them in your software so that you may add an additional level of photorealism to your comps.

The third thrust of this book is to bring you up to speed on all of the latest technology being used for compositing visual effects today such as AOVs for CGI compositing, 3D compositing, the use of Alembic geometry, OpenColorIO, and ACES, the awesome color management system created by the Academy of Motion Pictures Arts and Sciences specifically for feature-film visual effects. There is even a sneak preview of the near future of visual effects for when light field cinematography comes on line in the next few years. This astonishing technology will revolutionize visual effects. Be ready.

While the information and techniques described herein apply equally to film and video work, the focus of the book's presentation is feature film, simply because it is the most demanding application of visual effects. The much greater resolution and dynamic range

of filmmaking make it more challenging to achieve excellent results so it is the “gold standard”. HD video is, of course, extremely important, so it gets its own special section detailing its special issues such as limited dynamic range, interlaced images, chroma sub-sampling, and other technical delights.

By virtue of gaining a deeper understanding of how things work my hopes for the reader of this book are threefold. That you will finish your comps faster, that they will look more photorealistic, and that you will have more fun in the process. Let's face it. What could be more rewarding than putting together a beautiful shot to be admired by millions of viewers? Enjoy.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>



The Composer's Creed:

The ultimate artistic objective of compositing is to take images from a variety of different sources and combine them in such a way that they appear to have been shot at the same time under the under the same lighting conditions with the same camera.

Steve Wright, 2001

To do all of this well it is important to have an understanding of the technology of compositing because many of the obstacles that you will encounter are, in fact, technical problems. They stem from underlying technical issues that are not at all obvious to the casual observer but create problems in the shot. One example would be a double premultiply introducing dark edges around an element in the comp.

The designers of most visual effects compositing software have tried mightily to create a software tool that hides the technology so that it can be used by artists, and to a large degree they have succeeded. However, no amount of artistic training will help you to pull a good matte from a bad bluescreen or smooth motion track jitter caused by grain content. Problems of this sort require an understanding of the underlying principles of the digital operations being used to create the shot plus a library of techniques with multiple approaches to a problem. To that end, this 4th edition of this book contains a great many step-by-step workflow examples for keying and visual effects compositing.

It takes three distinct bodies of knowledge to be a good visual effects artist – art, tools, and technique. The artistic knowledge is what allows you to know what it should look like in the first place in order to achieve photorealism. The knowledge of your tools is simply knowing how to operate your particular compositing software package. The third body of knowledge, technique, comes with experience. Eventually you become a seasoned veteran where you are seeing problems for the second or third time and know exactly what to do about them. The beginner, however, is continually confronting new problems and it takes time to run through all the bad solutions to get to the good ones. This book contains over 20 years of production experience to help you get to the good ones faster.

While digital artists are invariably smart people, being artists they undoubtedly paid more attention in art class than in math class. But math is occasionally an indispensable part of understanding what is going on behind the screen. What I did in this book was, first, avoid the math wherever possible. Then, in those situations where the math is utterly unavoidable, it is presented as clearly as I know how, with lots of visuals to smooth the path for artists that are, ultimately, visual thinkers. I hope that as a result you will find the light smattering of math relatively painless.

1.1 How this Book is Organized

This book is organized in a task-oriented way rather than a technology-oriented way. With a technology-oriented approach all topics relating to the blur operation, for example, might be clustered into one chapter on **convolution kernels**. But blurs are used in a variety of work situations – refining mattes, motion blur, and defocus operations, to name a few. Each of these tasks requires a blur, but trying to put all of the blur information in one location is counter to a task-oriented approach when the task is refining a matte. Of course, scattering the blur information across several chapters introduces the problem of trying to find all of the information on blurs. For this issue a robust index comes to the rescue.

Part I – Making a Great Composite

The first part of this book is organized in the workflow order for the core tasks of pulling a key, performing the despill, and compositing the layers for both greenscreen and CGI shots.

Chapter 2: Pulling Keys – a million ways to pull a key. All about luma keys, chroma keys, difference mattes, bump mattes, color differences mattes plus an extensive section on professional rotoscoping techniques.

Chapter 3: Working with Keyers – pulling a key with a keyer. The internal workings of keyers, how to build your own After Effects keyer, coping with problem greenscreens and extensive techniques on preprocessing the greenscreen including the awesome screen-correction process.

Chapter 4: Refining Mattes – refining matte cores and edges. Using viewer “gamma slamming” to reveal matte defects, procedural techniques for creating garbage mattes, use of the media filter for noisy mattes, expanding, contracting and sculpting the matte edge falloff.

Chapter 5: Spill Suppression – multiple spill suppression techniques. Sources of spill and how despill works. Different spill suppression algorithms, my unspill operation, detecting and preventing despill artifacts, dialing in the edge color for the comp, and the edge-extension technique for fixing discolored edges.

Chapter 6: The Composite – a survey of methods for setting up a greenscreen composite. A look at the Over, AddMix and KeyMix composites, the Processed Foreground method, compositing inside the keyer, compositing outside the keyer, creating the Uber-key, and stereo compositing.

Chapter 7: Compositing CGI – proper technique for combining multi-pass CGI renders. Working with AOVs, ID passes, and normals relighting. All about EXR files, High Dynamic Range (HDR) images, and a large section about deep compositing.

Chapter 8: 3D Compositing – all about 3D compositing. Includes a short course in 3D explaining 3D terms and concepts for compositors. Camera tracking and how it is used, classic 3D compositing setups, an explanation of Alembic geometry and how to work with it.

Part II – The Quest for Realism

After we have a technically excellent composite we turn our attention to color correcting the layers to make them look like they are in the same lightspace, then adding several subtle effects such as light wrap that help to visually integrate the layers, and finally further blending the layers with camera effects such as vignetting.

Chapter 9: Color Correction – techniques for great color correction. First an understanding of light and light propagation, all about gamma, then a detailed explanation of the color operations (lift, gamma, gain, etc.) with their affects on appearance, on the code values, and when to use which one. Then a step-by-step process for color correcting a comp.

Chapter 10: Sweetening the Comp – a dozen things to do to make your comps more photorealistic. Adding interactive lighting, edge blending, light wrap, layer integration and creating realistic shadows. Extensive section on grain management, plus how to manage clipping.

Chapter 11: Camera Effects – recreating multiple lens effects. Lens distortion and depth of field plus explanations of and the steps for recreating vignetting, spherical aberration, chromatic aberration, diffraction glows, and many other lens effects. Complete workflow for managing lens distortion on a job, plus an explanation of rolling shutter.

Part III – Things You Should Know

At this juncture we have completed the basic composite. The topics now turn to a broad range of issues beyond the specifics of keying and compositing, which affect the quality of the overall results.

Chapter 12: Digital Color – color space and color management for visual effects. A brief description of what a color space is and what it is made of, plus a discussion of working in linear space. What is OpenColorIO and how do we work with it, plus a detailed description of the ACES color management system.

Chapter 13: Image Blending – blending images with and without an alpha channel. How image blending is affected by linear light space, a review of the alpha compositing operations, binary image blending math and visual results. How to duplicate Adobe Photoshop special blending modes, the many uses of slot gags, and a detailed discussion of retiming. There is also a major section on VR stitching using generic software, as it is a challenging image blending process for today's compositors.

Chapter 14: Transforms and Tracking – transforms, warping, morphing, point tracking and planar tracking. Understanding 2D transforms with the affects of pivot points and filter choices. Spline and Mesh warps used for morphs. Point tracking used for match move and stabilizing, plus how planar tracker work.

Chapter 15: Digital Images – a survey of today’s digital images. A study of HD video, digital cinema images, film scans, and what are log images and why we need them. There is also an extensive section on the awesome new light field cinematography just around the corner and its implications for you and your job.

1.2 Web Content

WWW To download the web content for this chapter go to the website www.routledge.com/cw/wright , select this book, then click on Chapter 1.

This message greets you at the top of each chapter, alerting you that you may download all of the support media for that chapter from the publisher’s website. It is recommended that you download the respective web content before reading each chapter so the images are on hand when you encounter them in the text. The web content is provided to encourage you to try the techniques in the book as you read about them, and they consist of images, clips, some informative QuickTime movies, and the odd pdf file. If you download Chapter 1’s web content now you will be set up for the next paragraph.

WWW Macbeth Chart.dpx – this is your own copy of the classic Macbeth chart used for calibrating color print film stock.

You will see messages like the one above at points in the text where an element can be used to try out what you have just been reading about. If you have downloaded the content for Chapter 1 you will find this Macbeth chart. It’s not used in this chapter but is offered here as a standard color reference that you can use for your work.

1.3 What’s New in the 4th Edition

The truth is, practically everything. This was a major rewrite to place the emphasis on good compositing technique, as well as adding new technology in the field of visual effects. For these two reasons there are a great many brand new topics in this edition that were not in the previous ones. The following is a list of the new topics in this edition

so that those of you who have already the previous three editions can jump to the juicy new stuff right away:

“Blur and Grow” Technique – an amazing method for expanding edge pixels to fill an interior void such as for marker removal, plus many other applications. *Chapter 2, Section 2.6.*

Roto Shape Breakdown – strategies for breaking a roto target into multiple shapes for the most efficient rotoscoping. *Chapter 2, Section 2.7.2.*

Roto Edge Decontamination – remove the background pixels mixed in the motion blur of a roto'd edge. *Chapter 2, Section 2.7.5.2.*

Screen Correction – an astonishing procedure for replacing a non-uniform greenscreen backing with a perfectly uniform color that every keyer will love. *Chapter 3, Section 3.5.6.*

Blurring Out – blurring the edge of a matte without shrinking the matte core. *Chapter 4, Section 4.4.3.*

Sculpting Edges – using a color LUT to literally sculpt the falloff of a key to sweeten the edges of a comp. *Chapter 4, Section 4.4.4.*

Edge Masks – creating masks that only affect the edges of a comp. *Chapter 4, Section 4.5.*

Despill Edge Grading – apply color correction to only the despillied regions of a keyed element in order to fix discoloration. *Chapter 5, Section 5.6.*

Despill Edge Extension – trim away ugly discolored edge pixels and “grow” new edge pixels that are the perfect color. *Chapter 5, Section 5.7.*

Premultiply vs. Unpremultiply – a detailed explanation of this confounding topic, with workflow examples. *Chapter 6, Section 6.1.*

Compositing with Multiple Keyers – combining the keyed outputs of more than one keyer. *Chapter 6, Section 6.3.*

Compositing Outside the Keyer – workflows for combining multiples keys from different keyers into a single “Uberkey”. *Chapter 6, Section 6.4.2.*

Lighting Passes – workflows for combining CGI lighting passes. *Chapter 7, Section 7.1.3.*

AOVs – major section on working with AOVs (Arbitrary Output Variables). *Chapter 7, Section 7.1.4.*

Deep Compositing – a detailed exploration of deep images and deep compositing. *Chapter 7, Section 7.4.*

3D Compositing from 2D Images – using a position pass to build a 3D point cloud from a CG image to use for 3D lineup. *Chapter 8, Section 8.2.1.*

Alembic Geometry – a powerful new file interchange format that allows a 3D scene to be imported into a compositing program for 3D compositing. *Chapter 8, Section 8.3.*

Camera Tracking – an exploration of camera tracking, with workflow recommendations. *Chapter 8, Section 8.4.*

The Effect of Color Operations – a detailed breakdown of the effect of color ops such as Lift, Gamma and Gain on both the image's code values and the look of the image. *Chapter 9, Section 9.3.*

Regraining Techniques – various workflows for creating, “borrowing”, and rescuing grain. *Chapter 10, Section 10.8.2.*

Grain Management – detailed workflow examples for how to preserve grain attributes throughout the comp. *Chapter 10, Section 10.8.3.*

Managing Clipping – how to handle clipping both to prevent, or to correct, during the comp. *Chapter 10, Section 10.9.*

Lens Effects – extensive section on how to replicate key lens effects such as chromatic aberration, astigmatism, diffraction glows, and many more. *Chapter 11, Section 11.1.*

Lens Distortion Workflows – detailed workflow for proper management of the lens distortion for live action and CGI shots. *Chapter 11, Section 11.2.*

Rolling Shutter – most digital cameras can introduce rolling shutter so we will see what a causes it and how to fix it. *Chapter 11, Section 11.4.*

Color Spaces – a major informational section on color space. *Chapter 12, Section 12.1.*

Working in Linear – what exactly is linear light space and why it is critical to high dynamic range feature work. *Chapter 12, Section 12.2.*

OpenColorIO – what it is and how is it used in the modern visual effect pipeline. *Chapter 12, Section 12.4.*

ACES Color Management – the powerful color management system developed by The Academy of Motion Picture Arts and Sciences and being adopted throughout the visual effects world. *Chapter 12, Section 12.5.*

Image Blending in Linear – how working in linear affects classic image blending ops such as Screen and Over. *Chapter 13, Section 13.1.*

Matching the sRGB Look in Linear – workflows for matching the sRGB composites from others when working in linear. *Chapter 13, Section 13.1.3.*

Adjusting a Multiply – the multiply operation normally darkens but this technique fixes that when working in linear. *Chapter 13, Section 13.3.3.1.*

VR Stitching – a major new section stitching 360 video into a VR LatLong map using generic software. *Chapter 13, Section 13.7.*

Managing Motion Blur – managing motion blur for transformations, motion UV channels, and speed changes. *Chapter 14, Section 14.1.2.*

Image Displacement – using a displacement map to distort a second image. *Chapter 14, Section 14.2.*

Point Tracking Tips – tips for improving point tracking results. *Chapter 14, Section 14.4.1.*

Planar Tracking – how it works, how to use it as a roto assist, plus workflow tips for best tracking results. *Chapter 14, Section 14.5.*

Digital Cinema Images – all about the awesome images produced by the high-end digital cinema cameras being used for filmmaking today. *Chapter 15, Section 15.2.*

Log Images – digital cinema cameras routinely produce log images of their own design, but what are they and why do we need them? *Chapter 15, Section 15.4.*

Light Field Cinematography – an astonishing new technology for filmmaking that will revolutionize visual effects by capturing the scene's "light field". *Chapter 15, Section 15.5.*

1.4 Gold Mines

One of the big additions to this edition is a great number of step-by-step production workflow examples for essential tasks like keying, spill suppression, tracking, and many others. The workflow examples are extremely valuable so are listed here as a sort of highly specialized index that you can use for quick reference after reading the book.

Difference Mattes – how to create your own difference mattes even if your software doesn't support them. *Chapter 2, Section 2.3.2.*

Color Difference Keys – a novel method for creating a key for an arbitrary object in an arbitrary background. *Chapter 2, Section 2.5.*

“Blur and Grow” Technique – how to expand the RGB pixels at the edge of a key much better than a dilate operation. *Chapter 2, Section 2.6.*

Rotoscoping – an extensive section on rotoscoping techniques with many productivity and quality tips. *Chapter 2, Section 2.7.*

Edge Decontamination – how to remove the background pixel contamination in a motion blurred roto or key. *Chapter 2, Section 2.7.5.2.*

The After Effects Keyer – how to build this elegant little keyer and dial it in. *Chapter 3, Section 3.3.*

Local Suppression – how to create an isolation mask to suppress problem areas for a keyer. *Chapter 3, Section 3.5.3.*

Screen Correction – how to use screen correction to even up bad greenscreens to get a great key from your keyer. *Chapter 3, Section 3.5.6.*

How to Create a Clean Greenscreen – advanced technique for combining all the good parts of a greenscreen into one clean plate of green backing. *Chapter 3, Section 3.5.6.4.*

Edge Extension – how to remove the bad pixels of an ugly edge and grow new pixels of the right color. *Chapter 5, Section 5.7.*

Premultiply vs. Unpremultiply – at last a comprehensive explanation of these important but confusing operations. *Chapter 6, Section 6.1.*

The Processed Foreground Method – an amazing compositing technique that produces the best edges in the known universe. *Chapter 6, Section 6.2.4.*

Soft Comp/Hard Comp – laying down a soft comp on the background for nice edges then on top of that a hard comp for a solid core. *Chapter 6, Section 6.3.1.*

“Cut and Paste” Keyer Compositing – cutting and pasting the output of different keyers into a single hero key. *Chapter 6, Section 6.3.2.*

Camera Tracking – how camera tracking works and detailed workflow tips for getting better results. *Chapter 8, Section 8.4.*

Matching the Lightspace – starting with a luminance version set the black and white points, then gamma, then turn on color and set the hue. *Chapter 9, Section 9.4.*

Light Wrap – creating and applying a light wrap effect to a composite. *Chapter 10, Section 10.4.*

Lens Filter Flare – lens filters introduce a type of flaring to the image so learn how to replicate this affect. *Chapter 11, Section 11.1.5.2.*

Lens Distortion Workflows – detailed workflow steps for managing the lens distortion when compositing live action or CGI elements. *Chapter 11, Section 11.2.*

The Pivot Point Lineup Procedure – line up two images faster and more accurately with this methodical procedure. *Chapter 14, Section 14.1.5.3.*

Stabilizing For Rotoscoping – get your rotos done faster and cleaner by stabilizing the target first. *Chapter 14, Section 14.4.3.3.*

1.5 Tool Conventions

The **slice tool**, **flowgraphs**, and **color curves** (color LUTs) are used extensively throughout this book to analyze images, show procedures, and modify **pixel** values. They are essential to the telling of the story, so this section describes them in order to develop a common vocabulary and overcome any terminology differences with your software. Your software may not have a slice tool so it may be a new concept to some, but I'm sure you will quickly appreciate why it is such an important tool for revealing what's happening inside an image. Who knows, you may even want write to your software developer and request it as a new feature.

While the flowgraph is becoming the standard interface for compositing software, terminology differences, and its absence in a few compositing packages, warrant a quick review of how it is used in this book. While all software packages have color LUTs, we will be putting them to some new and interesting uses, so there is a quick review of their operation here.

1.5.1 The Slice Tool

The slice tool is a very important image analysis tool used extensively throughout this book to illustrate pixel processing. Only certain software packages have such a cool tool, so an introduction to its operation and why it is so helpful is in order. Use of the slice tool starts with drawing a single straight line across the region of interest of a picture, shown in **Figure 1.1** as a diagonal white line. The second step is to plot the pixel values that are under the slice line into a graph, shown in **Figure 1.2**. We are starting with a **grayscale image** because it is easier to follow, then we will see it in action on a three **channel RGB** image.

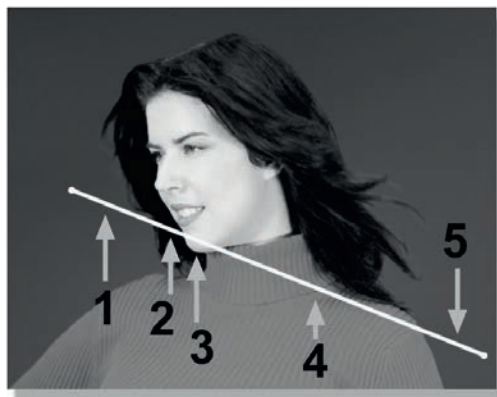


Figure 1.1 Slice line on image

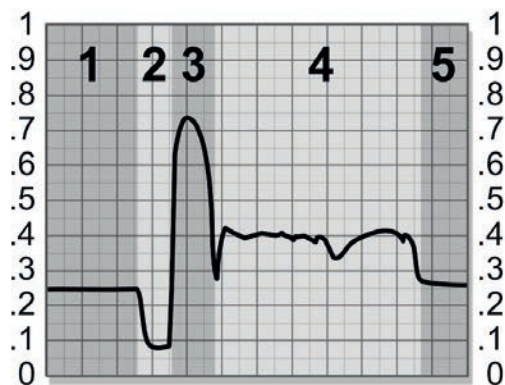


Figure 1.2 Slice graph of image

Starting at the left end of the slice line in **Figure 1.1**, the graph in **Figure 1.2** plots the pixel values under the line from left to right. In this introductory example each region of the image that the slice line crosses is numbered to help correlate it to the graph. Region 1 crosses the **bluescreen** backing region, which results in a pixel **brightness** of around 0.25 in the graph. Region 2 crosses the black hair so the graph dips down to below 0.1 there. Region 3 has a bright spike where it crosses the chin, and region 4 is the long section of the medium gray sweater. Finally, region 5 ends up back on the bluescreen. Interestingly, even though regions 1 and 5 are both on the bluescreen, the graph is a bit higher in region 5 suggesting that the bluescreen is a bit brighter on the right side of the picture.



Figure 1.3 Slice line across an RGB image

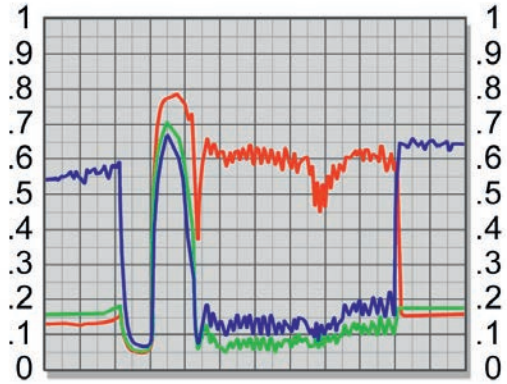


Figure 1.4 Graph of RGB values under slice line

Now that the operation of the slice tool has been established for a one channel grayscale image we can compare it to an example of the same image in full color in **Figure 1.3**. Here the pixel values of all three channels are plotted in the colored slice graph in **Figure 1.4**. Over the bluescreen backing regions the blue record can be seen to be much greater than either the red or green, as we would expect in a well-exposed bluescreen. The brightness levels and color proportions of the hair and skin tones are also easy to see. The red sweater can even be seen to have a slight blue bias. These graphs of the absolute level of each color record across the image also reveal their levels relative to each other, which provides invaluable insights for pulling **mattes**, **despill** operations, and a myriad other digital effects tasks.

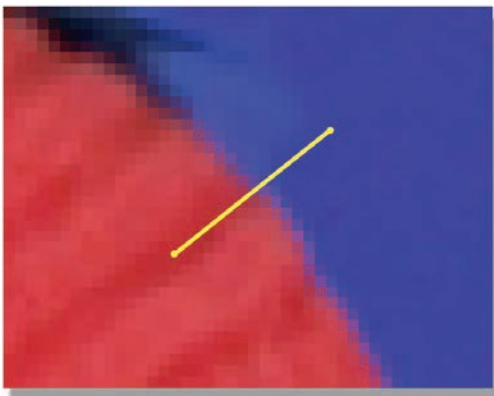


Figure 1.5 Slice line across edge transition

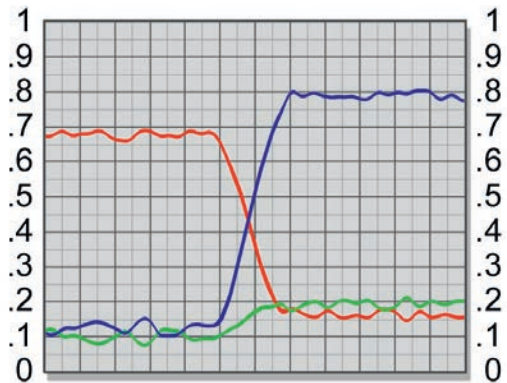


Figure 1.6 Graph of RGB values across edge transition

Another very important application of the slice tool is to graph edge transitions as shown with the close-up in **Figure 1.5** and **Figure 1.6**. Here a short slice line starts on the red sweater and crosses over to the bluescreen. The actual transition pixel values between

the two regions can now be seen and studied. By striking a short line on the image that spans relatively few pixels, the resulting graph in effect “zooms in” to the image to show more detail.

The slice graph is much more revealing of what’s happening in your **color space** than a point sample tool because it displays the pixel values across entire regions of the picture and reveals the relationship between the channels, which is often a key item of interest. With a slice line across an entire bluescreen, for example, the graph will reveal the uneven lighting, which way it rolls off, and by how much. Appropriate correction strategies can then be developed to compensate. Being able to characterize the pixel values across broad regions of a regular image such as the sky can also be very helpful. You will also undoubtedly find it more intuitively helpful to see the RGB values plotted on a graph than just the RGB numbers from a pixel sample tool.

1.5.2 Flowgraphs

Flowgraphs are used extensively throughout this book to illustrate the sequence of operations needed to achieve a particular effect. Most of the major compositing software packages have settled on a flowgraph **GUI** (Graphical User Interface) as the logical presentation of the compositing data flow to the user. The flowgraph is even useful to users that do not have a flowgraph GUI, such as Adobe Photoshop, because they clearly show the sequence of operations in a way that can be translated to any software package, including those with a command line interface. Even for someone that has never seen a flowgraph before it is intuitively clear not only what operations are to be done, but their specific order, including any branching and combining.

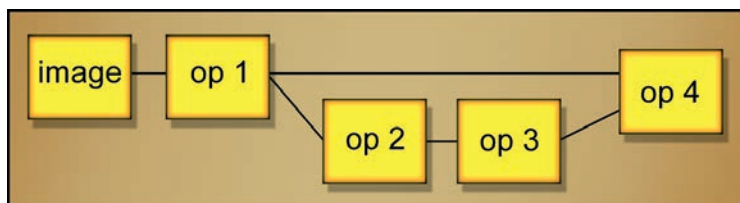


Figure 1.7 A generic flowgraph example

Figure 1.7 shows an example of a generic flowgraph, which is read from left to right. Each box is called a “**node**” and represents some **image-processing** operation such as a blur or **color correction**. Reading the flowgraph in **Figure 1.7**, the first node on the left, labeled “image”, can represent the initial input operation where the image file is read from disk or simply the initial state of an image to which the operations that follow are to be attached. The node labeled “op 1” (operation 1) is the first operation done to the image, then the processing forks to the nodes labeled “op 2” and “op 3”. The

output of “op 3” is then combined with “op 1” to form the two inputs to the last node labeled “op 4”.

Digital compositing is in actuality an example of what is referred to as “data flow programming”, a computer science discipline. The reason that the flowgraph works so well is that it is a straightforward graphical representation of the “flow” of data (the images) from operation to operation, which results in a “program” (the compositing script) that produces the desired results – an outstanding **composite**, in this case. The clarity and intuitiveness of the flowgraph has earned it a permanent place in the man–machine interface for digital compositing.

1.5.3 Color Lookup Tables (LUTs)

The Color **Lookup Table (LUT)**, also known as a color curve, is another important convention used throughout this book to adjust RGB and grayscale color values. Beyond its original intent as a color correction tool it has a myriad of other uses such as scaling mattes, altering edge characteristics, clamping, and more – all of which will be utilized in this book. By whatever name it is known in your software, this operation is present in virtually every compositing package, as well as Adobe Photoshop. The convention in this book is to label the pixel values in **floating point** format from zero to 1.0, which is also common practice in most software packages.

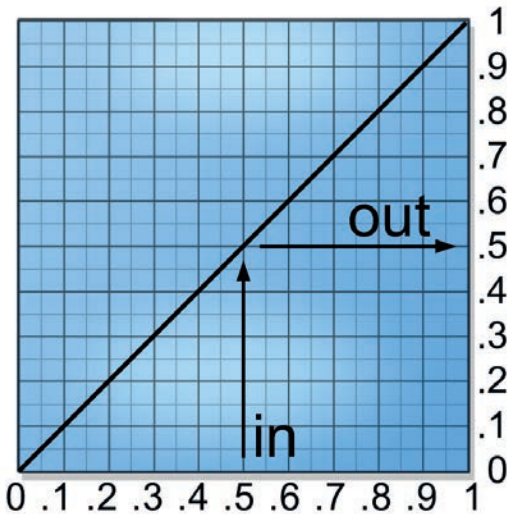


Figure 1.8 An “identity” curve

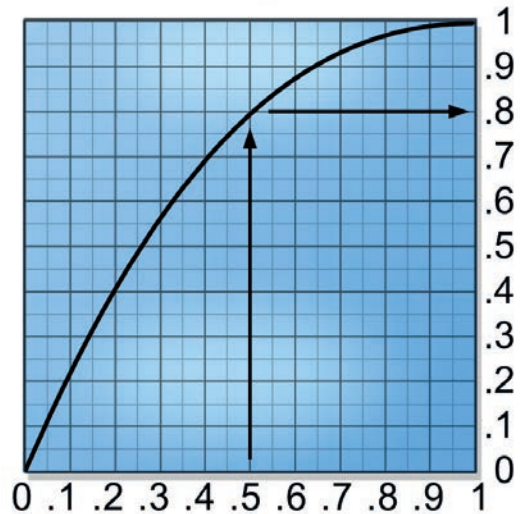


Figure 1.9 Mapping an input value to a new output value

The short form of the color LUT story is that it simply re-maps the pixel values of the input image to a new set of pixel values for the output image based on the color curve. The input image pixel values are indicated along the bottom edge of the graph marked “in” on **Figure 1.8**, and the new output values are read along the right edge marked “out”. The graph in **Figure 1.8** is an identity LUT, meaning that it gives the same value out as the value in, making no change to the image. The arrows point to an input value of 0.5, mapping to an output value of 0.5.

The color LUT can duplicate many color ops like **lift**, **gain**, and **offset**. The key new feature a LUT brings to the table is the ability to do a **non-linear** operation where the LUT is “bent” in the middle like in **Figure 1.9**, where an input value of 0.5 results in an output value of 0.8. This would brighten the darks and **midtones** of any image that it was connected to. The color LUT can be drawn in virtually any desired shape, and from this comes its great flexibility and multiple applications.

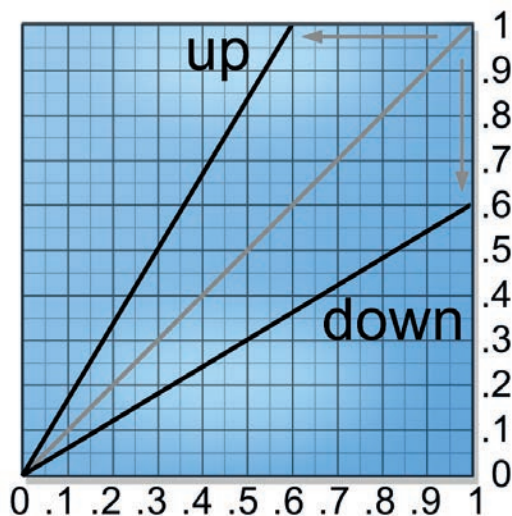


Figure 1.10 Scaling values up and down

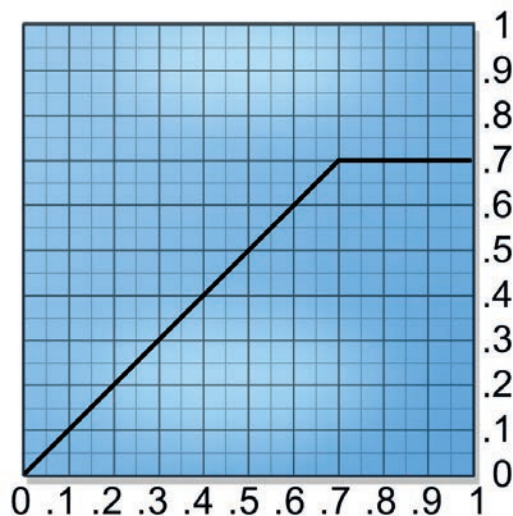


Figure 1.11 Clamping

Here are just a few of those multiple applications. In **Figure 1.10** two color curves are shown: one that **scales** the pixel values up, and one that scales them down. For the “scale up” curve an input pixel value of 0.6 becomes an output pixel value of 1.0, which is a scale factor of $(1.0 \div 0.6 =) 1.67$ on all pixel values. In other words, it is performing the same operation that a “scale RGB” or Gain op set to 1.67 would do. The “scale down” curve maps an input pixel value of 1.0 to an output pixel value of 0.6, which scales all pixel values by, well, 0.6. These scale operations assume that the curve is a straight line, of course. **Figure 1.11** illustrates the color curve used to clamp an image to limit its brightness values. The color curve is an identity curve up to an input value of 0.7, so

all pixels between 0 and 0.7 are unchanged. All input pixel values above 0.7, however, become clamped to the value 0.7 upon output.

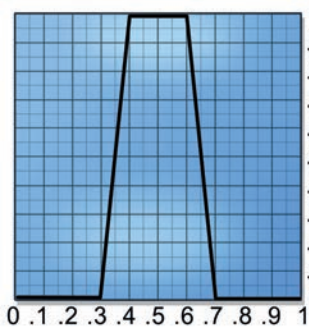


Figure 1.12 Hicon matte from grayscale image

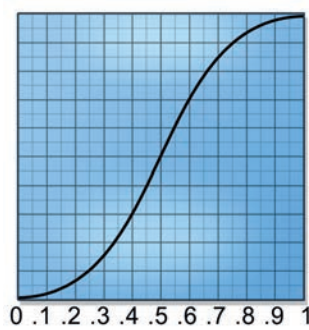


Figure 1.13 Contrast “S” curve

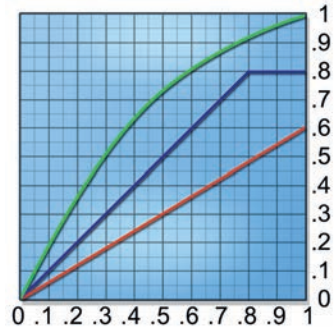


Figure 1.14 RGB color curve

Moving to more elaborate configurations, **Figure 1.12** represents using a color LUT to raise a middle gray to 100% white for a **luma key**. In this example code values between 0.4 and 0.6 would be pulled up to 1.0 while everything less than 0.3 and greater than 0.7 would go to black. The sloped edges make for a nice **falloff** for soft edges on the key. **Figure 1.13** shows an “**S**” curve for increasing **contrast**. This would increase the contrast of an image without **clipping** it in the blacks or whites like a typical contrast operation would.

Of course, the color LUT works on 3-channel RGB images as shown in **Figure 1.14**. In this example the red channel is scaled down by 0.6, the green channel has the darks and midtones brightened, and the blue channel is clamped to 0.7. We will see later that this ability to tweak each of the three channels individually comes in very handy in a number of situations.

1.5.4 Nuke

This book is software agnostic so you can use the techniques herein with any compositing program. My compositing app is Nuke, the industry standard in virtually every visual effects studio around the world, so I do use a few screenshots from Nuke to illustrate certain points. I also teach Nuke to individual students and do staff training for visual effects studios around the world from my training website www.fxacademy.com.

1.6 Data Conventions

The data conventions in this book assume that you are using floating point data, not integer (8 or 16 **bit**), and working in linear **light space** or “linear” instead of **sRGB** or **rec709**. All of the code values for pixels are in floating point and we assume a high **dynamic range** system that does not clip the data above code value 1.0 or below zero. If you are using this book with an integer based system then you will have to make the necessary allowances and in some cases modify the techniques a bit. It was simply not practical to provide dual workflows, one for float and one for integer, and the great majority of professional visual effects studios work in float and linear these days.

1.6.1 Floating Point Data

All pixel code values in this book are expressed in floating point values such as 0.4, 0.542, or 2.197, because most professional compositing software today works in float. A floating point system is able to make very large to very small numbers, as well as go negative, all with very great precision. For example, 9,348.3 for an insanely bright pixel is no problem for float, as well as incredibly tiny code values like 0.0003281. And of course, these can also go negative with the same precision.

There are several reasons why we need float for visual effects. First is that we work with high dynamic range images which can have code values far in excess of 1.0 and they must not be clipped. Second, we often do heavy processing on our pixels, scaling them way up or way down and do not want to add round-off errors to our computations so the pixel values need to be represented with great precision, meaning many decimal places of accuracy.

There are some apps (application programs) such as Photoshop, After Effects and Shake that can also work in 8 bit (0–255) or 16 bit (0–65,535) integer. For these programs the floating point values in this book will need to be converted to integer by the reader. Note that even if your app is working in 8 bit integer many internal operations are done in float. For example, you will not get the right answer if you multiply 128×128 , but you will get the right answer if you multiply $0.5 \times 0.5 = 0.25$ then convert that back to its integer value of 64.

1.6.1.1 Banding

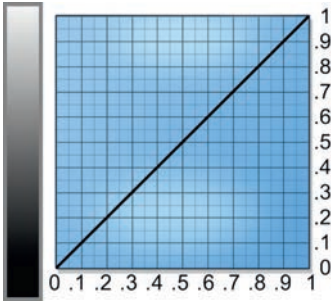


Figure 1.15 256 steps

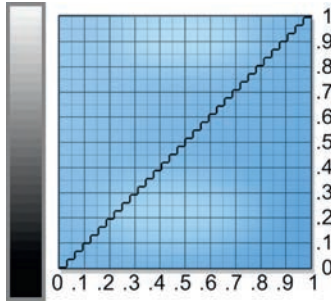


Figure 1.16 32 steps

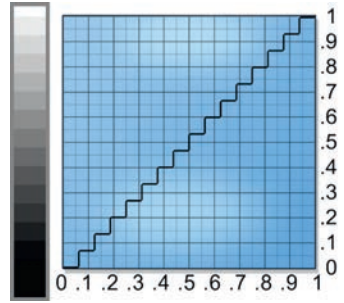


Figure 1.17 16 steps

Working in integer data introduces two hideous problems for compositing visual effects – **banding** and clipping. We see banding in an image if the steps between the pixel brightness' are too great. **Figure 1.15** illustrates the smooth appearing gradient on the left when using 8 bits to get 256 brightness steps from 0 to 255. The jump in brightness from step to step is small enough that the eye does not see it. However, if the number of steps is reduced to 32 like in **Figure 1.16** we start to see banding. And if the number of steps is reduced further to 16, like in **Figure 1.17** the banding becomes very obvious and objectionable.



Figure 1.18 Grayscale image

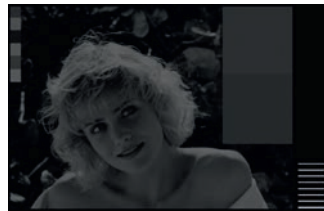


Figure 1.19 RGB scaled by 0.1



Figure 1.20 RGB scaled by 10

Why banding occurs during image processing can be seen starting with the 8 bit grayscale image in **Figure 1.18**. Using a grayscale image just makes the banding easier to see. To the right of the picture is what I call a “bit comb”, a little graphic that represents the number of code values available to represent the entire image. Integer code values are like the teeth of a comb because they have only, and exactly, 256 code values ranging from 0 to 255. If any teeth are missing it starts to introduce banding.

If the image is heavily darkened by scaling down the RGB values by 0.1 for example, the image grows darker like **Figure 1.19** as we expect, but the code values are also reduced to between 0 and 25. At this point the image now only has 26 possible code values as reflected in the adjacent much shorter bit comb. If that image is then scaled up by 10 to restore the original brightness like **Figure 1.20** the 26 available code values are now spread out from 0 to 255 leaving gaps between the teeth of the “bit comb”

which we see as banding in the image. The gaps occur because we have only integers to work with, no fractions, so we cannot make in-between code values when we stretch 0–25 up to 0–255.

1.6.1.2 Clipping

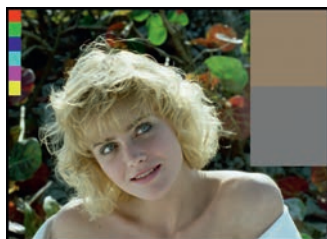


Figure 1.21 Original image

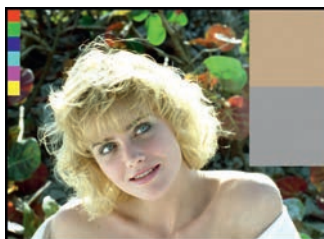


Figure 1.22 Brightness $\times 2$

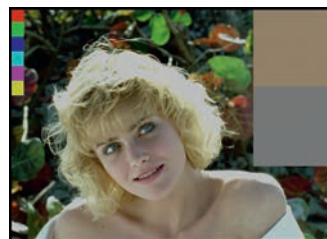


Figure 1.23 Brightness $\times 0.5$ showing clipping in hair

The other hideous problem I promised you was clipping. Starting with the original image in **Figure 1.21** the brightness is scaled by 2 in **Figure 1.22**. The original image's brightest pixel was 255 but after being scaled by 2 it should become 510. You can see how the hair has become one bright slab of white. Tragically, in an 8-bit system 255 is the largest number possible so what was supposed to be code value 510 is clipped to 255. In fact all code values above 255 are clipped to 255. When the image brightness is scaled by 0.5 to return it to normal in **Figure 1.23** all of the clipped pixels drop back from 255 down to 128, producing a flat slab of gray. The hair has become hideously clipped.

If these same brightness-scaling operations were done in float the original image's brightest pixel would be 1.0. After scaling it would become 2.0 – but it would not be clipped. When the brightened image is returned to normal, code value 2.0 would be scaled back down to 1.0, but again it would not be clipped.

1.6.2 Linear Light Space

Another premise of this book is that we are working in linear light space – or simply “linear”. **Linear data** simply means that if you double the code value of pixel it doubles its brightness. Many think they are working in linear when working with sRGB or rec709 (HDTV) images but these are in fact **gamma-corrected** linear and not truly linear. Linear is assumed in this book for one simple reason – it is the correct data format for visual effects image processing. A detailed explanation of linear workflows and its affects on color ops, image transformations and **CGI** compositing are discussed in *Chapter 12: Digital Color*.

1.6.3 HDR Images

HDR (High Dynamic Range) images are also assumed throughout the book. HDR images are defined as those images that have code values that exceed 1.0. They may also go negative, containing code values below zero. HDR images must be treated with special care to avoid clipping them either in the highlights or the blacks, so this book addresses the care and feeding of HDR images in the visual effects pipeline with appropriate do's, don'ts and workflow cautions.

1.6.4 Stops

The term “**stop**” is a photography term used throughout this book that refers to a doubling or halving of an image's **exposure**. One stop up doubles the exposure, one stop down cuts it in half, and two stops up quadruples the exposure (doubling it twice). If the exposure is doubled a pixel's code value of 0.4 will double to 0.8. However, the perceived brightness to the human eye will only increase by about 18% because the human eye is a non-linear device. More on this in *Chapter 12, Section 12.2: Working in Linear*.

Enough talk. Let's jump into the wonderful world of keying, starting in the next chapter: all about pulling keys – all kinds of keys – luma keys, chroma keys, difference mattes, and more. These are the keying techniques we need for isolated items of interest that are not on a nice bluescreen or **greenscreen**, but we need to key them anyway. They are often needed to aid greenscreen **keyers**, which are covered in the chapter after that.

Part I

Making a Great Composite



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>



WWW

To download the web content for this chapter go to the website www.routledge.com/cw/wright, select this book then click on Chapter 2.

Pulling keys is a core skill for artists compositing visual effects. For bluescreen and greenscreen shots we have excellent keyers such as **Keylight** and **Primatte**, but we start here with a variety of ways of pulling keys in general – and save the keyers for the following chapter. There are two reasons for this. First, as you will see in the next chapter on keyers, we often have to pull “auxiliary keys” to assist the keyer, and we should have that information before tackling the keyers. Second, we frequently have to pull keys for non-greenscreen tasks. For both of these missions we will need an array of techniques for pulling keys on arbitrary targets in arbitrary scenes. So this chapter is about a million ways to pull a key.

The Terminology Turmoil – the language of visual effects has accumulated terminology over the years from a variety of different disciplines such as visual effects, video and CG animation. As a result we have four different words for the same thing here – key, matte, mask, and alpha – all of which do essentially the same thing, namely use a one-channel grayscale image to mark regions that are “inside” and “outside”. What is inside or outside simply depends on how you use it. Here is a micro-glossary of our “key” terms (sorry about that) as seen from the **perspective** of their historical origins, so you can see why there is so much confusion:

- **Matte** – (film opticals) a **one-channel image** that isolates a target object from its background for compositing.
- **Key** – (video) a one-channel image that isolates a target object from its background to isolate it for special treatment like color correction or blurring.
- **Mask** – (visual effects) a one-channel image that isolates a target object for special treatment like color correction or blurring.
- **Alpha** – (CG animation) a one-channel image created by the CG renderer to isolate the **foreground** image from the **background** image for compositing.

Note that you could, by the above definitions, take an **alpha channel** from a CG **render** and use it as a mask for a color correction operation. Or draw a mask for a color correction operation then turn around and use it as a key in a composite. In this book I switch terminology apropos the topic (i.e. “alpha” for CG, “key” for keyers, etc.), but hopefully this explanation will help you understand that they are not fundamentally different things.

This chapter on keying contains a major section on rotoscoping because that is yet another way to create a key that does not use a keyer. It also fits well in a chapter about a million ways to pull a key because I teach my students to “key what you can, then **roto** the rest”. The roto information provided herein is designed to make your roto work faster and better. There are tips on organization, keyframing strategies, how to roto special problems like **motion blurred** objects, and several inspection techniques so that you find the problems, and not your boss.

2.1 Luma Keys

First up on the keying hit parade is the ever-popular luma key. Luma keys get their name from the world of video where the video signal is naturally separated into **luminance** (brightness) and **chrominance** (color). A luma keyer isolates some range of the luminance values of an image to create the key. This key can then be used in any number of ways to isolate an item of interest for selected manipulation like color correction or blurring.

Luma keys are simple to create and flexible in their use because the target is often darker or lighter than the rest of the picture so this lightness or darkness can be used as the criteria of the key. Of course, they are usually mixed with other objects of the same luminance so we need techniques for isolating the one we want. This section describes how luma keyers work, their strengths and weaknesses, and how to make your own customized versions to solve special problems.

2.1.1 How Luma Keys Work

A luma keyer takes in an RGB image and first creates a luminance version of it, which is a **monochrome** (one-channel grayscale) image. A threshold value is set, and all pixel values greater than or equal to the threshold are set to 100% white, while all pixel values less than the threshold are set to black. Of course, this will produce a hard-edged key that is practically useless in the real world, so luma keyers typically have two settings: one for the black threshold and one for the white threshold. More on this shortly.

My favorite visualization of a luminance key is to imagine a **luminance image** laid down flat on a table and the pixels pulled up into mountains. The height of each “mountain” is based on how bright the pixels are. Bright pixels make tall mountain peaks, medium-gray makes low rolling hills, and dark pixels make valleys like those shown in **Figure 2.1**.

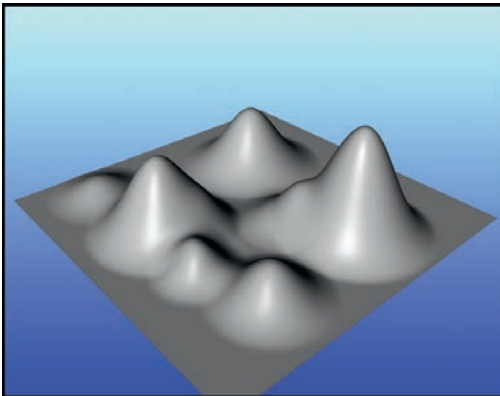


Figure 2.1 Monochrome image visualized as brightness “mountains”

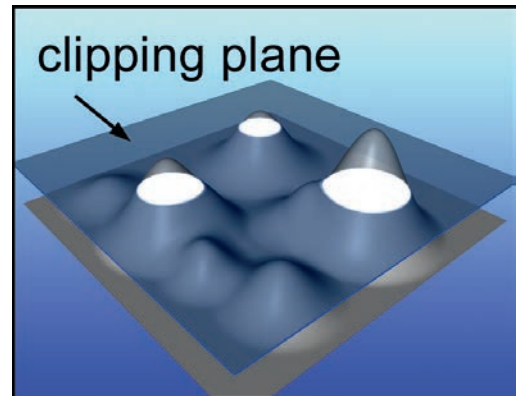


Figure 2.2 Clipping plane clips the tallest (brightest) mountain peaks

You can now imagine the threshold point of the luma key as a **clipping plane** that slices off the peaks of the mountains like **Figure 2.2**. The white areas are the intersection of the clipping plane with the mountain peaks and form the resulting luma key.

This metaphor provides a number of interesting insights to the luma key. The first and most obvious issue is how there are several mountain peaks sliced off by the clipping

plane, not just the one we may be interested in. This means that the luminance key will have “snared” unwanted regions of the picture and this is one big problem with luma keys – they are not very discriminating. Some method will have to be devised to isolate just the item we are interested in. Another point is how the clipped regions will get larger if the threshold point (clipping plane) is lowered and smaller if it is raised, which expands and contracts the size of the key. Yet another problem with luma keys is that your target object might have a lot of variation in its luminance, which results in the luma key snaring most of the picture and is not at all helpful. We may need to switch to a more discriminating type of key such as a **chroma key**.

There are two approaches to the problem of snaring more regions than you want. The first is to **garbage matte** the area of interest. Crude, but effective, if the garbage mattes do not require too much time to make. The second approach entails altering the heights of the “mountain peaks” in the luminance image so that the one you are interested in is the tallest. This is covered in the following *Section 2.1.2: Making Your Own Luminance Image*.

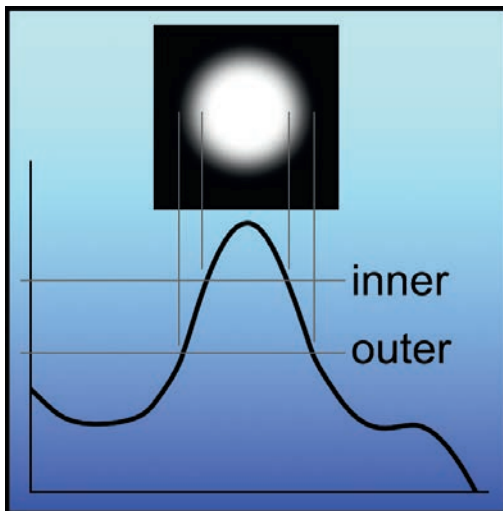


Figure 2.3 Inner and outer thresholds for soft edge key

Simply binarizing the image (separating it into just black and white pixel values) at a single threshold value creates a very hard edged key, but most uses require a soft edge. The soft edge issue is addressed by having two threshold values in the luma keyer settings. One setting is the inner, 100% density edge, and the other is the outer 0% density, with a gradient between them. Switching to a cross section view of one of the “mountain peaks” in **Figure 2.3** it can be seen how these inner and outer settings create a soft edged key. Everything greater than the inner threshold value is pulled up to 100% white. Everything below the outer threshold value is pulled down to a **zero black**. The pixels in-between these two values take on various shades of gray, which creates the soft-edged key shown in the inset above the mountain peak.

2.1.2 Making Your Own Luminance Image

There are two important things to be aware of about the luminance image that is generated internally by the luma keyer. The first is that there are a variety of ways to calculate this luminance image, and some of these variations might isolate your target better than the default. The second thing to be aware of is that it does not have to be a luminance image at all. Luma keyers will accept monochrome (one channel) images in addition to RGB images as inputs. We can take serious advantage of this fact by feeding it a customized monochrome image that we have created that better isolates the target.

2.1.2.1 Variations on the Luminance Equations

The normal idea of a luminance image is to convert a three-channel color image to a monochrome image, such that the apparent brightness of the monochrome image matches the apparent brightness of the color image to the human eye. As usual, this is more complicated than it first seems. The obvious approach of simply taking 1/3 of each of the three channels and summing them together doesn't work because the eye's sensitivity to each of the three primary colors is different. The eye is most sensitive to green, so it appears much brighter to the eye than equal values of red or blue.

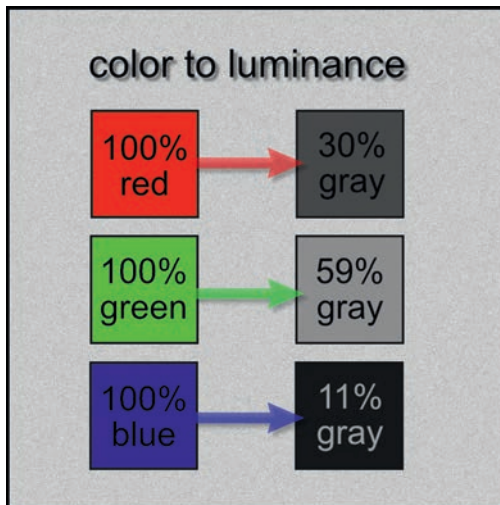


Figure 2.4 Color converted to luminance

Figure 2.4 shows how the eye responds differently to the three primary colors as luminance by taking a set of three color chips made up of primary colors. While each chip is 100% of its pure color, when converted to luminance they each produce a different gray. The 100% red chip produces a 30% gray, the green chip a 59% gray, and the poor blue chip a paltry 11% gray.

Here's the point: if you do not create the luminance image with the proper proportions of the red, green and blue values that correctly correspond to the eye's sensitivities, the results will look wrong. A simple 1/3 averaging of all three colors would

result, for example, in a blue sky looking too bright because the blue would be over-represented. A green forest would appear too dark because the green would be under-represented. One standard equation that mixes the right proportions of RGB values to create a luminance image for a color monitors is:

Equation 2.1 **Luminance = 0.30 R + 0.59 G + 0.11 B**

This means each luminance pixel is the sum of 30% of the red plus 59% of the green plus 11% of the blue. Be advised that the exact proportions differ slightly depending on what colorspace you are working in, so they may not be the exact values used in your luma keyer. And, of course, these percentage mixes can be different for media other than color monitors. If your software has a channel math operation then you can use it to create your own luminance images using **Equation 2.1**.

But making a luminance image that looks right to the eye is not the objective here. What we really want is to make a luminance image that best separates the target from the surrounding picture. Armed with this information you can now think about “rolling your own” luminance image that better separates your target by juggling the relative values of the red, green, and blue channels. Some luma keyers allow you to juggle the RGB ratios within their settings. Try different settings to make your target item stand out from the background better. If your luma keyer does not permit you to tweak the RGB values then feed it a luminance image you created externally with some other method such as a monochrome node that does allow you to adjust the RGB mix ratios.

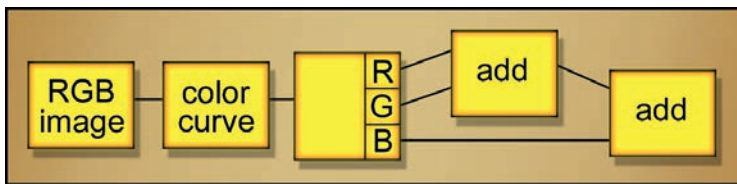


Figure 2.5 Flowgraph for generating a custom luminance image

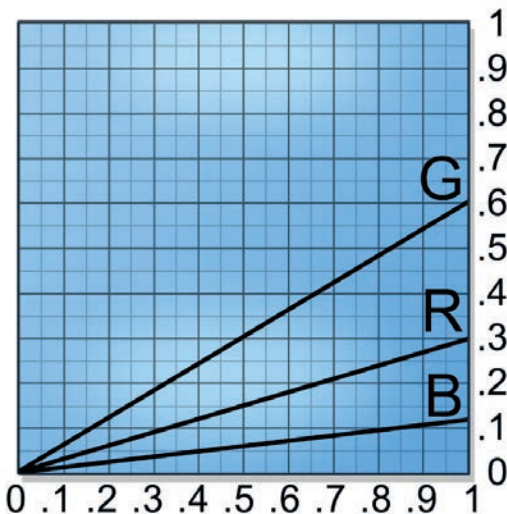


Figure 2.6 RGB scaling in the color curve

If your monochrome operation does not permit you to change the RGB mix proportions then another way you can “roll your own” luminance image is with a color curve or other operation in your software that allows you to scale the RGB values individually, then sum the RGB channels together. The flowgraph in **Figure 2.5** shows the sequence of operations. The RGB image goes into the color curve, where each **color channel** is scaled to the desired percentage mix value as shown in **Figure 2.6**. The 3-channel output goes to a channel split operation so the R, G, and B channels can be separated then summed together to create a custom luminance image with the attributes you need. This luminance image is then connected to the luma keyer to pull the actual key.

2.1.2.2 Non-luminance Monochrome Images

Another approach is to not use a luminance image at all. The luma key process is simply based on the brightness values in a monochrome image. This one channel image is usually created by making a luminance version of the color image, but it is not against the laws of man or nature to make the one-channel image in other ways. For example, just separate the green channel and pipe it into the luma keyer. Perhaps the blue channel has a better separation of your target from the rest of the picture, or maybe a mix of 50% of the red channel plus 50% of the green channel will work best. It totally depends on the color content of your target and the surrounding pixels. Unfortunately, the RGB colorspace of an image is so complex that you cannot simply analyze it to determine the best approach. Experience and a lot of trial and error are invariably required. Fortunately, the iterations can be very quick and a good approach can usually be discovered in a few minutes.

2.1.3 Making Your Own Luma Keyer

Perhaps you are using some primitive software that does not have a luma keyer, or you don't like your luma keyer for some reason. Maybe you just want to demonstrate your pixel prowess. Fine. You can make your own luma keyer using a color curve node, which every software package has. The first step is to make your best monochrome (one-channel) image that separates the target from the background using one of the techniques described above, then pipe it to a color curve node to scale it to a "hicon" (high contrast) key.

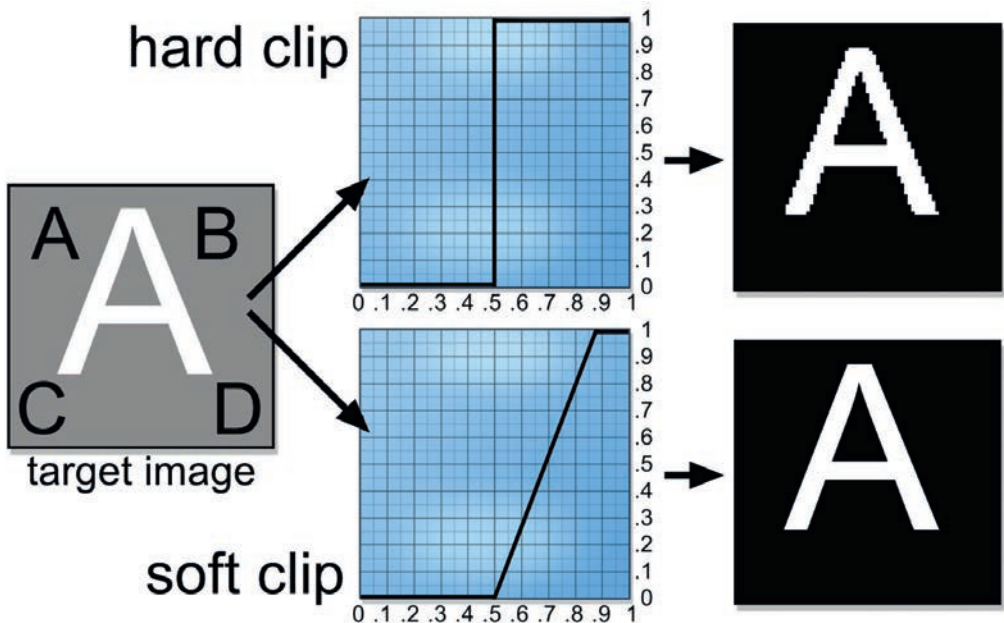


Figure 2.7 Using a color curve to make a luma key

So, how do we set up the color curve demonstrated in **Figure 2.7**? The target is the white letter “A” in the target image on the left, a deliberately easy target for demo purposes. The color curve labeled “**hard clip**” uses a single threshold value, but this results in a hard key with jagged edges (the effect is exaggerated for demo purposes). The color curve labeled “**soft clip**” shows a slope introduced to the color curve to create a soft edge for the key. The gentler the slope of the line, the softer the edge. Of course, nothing says that the target object will be the white object. A black object could be the target, so in that event the color curve would be set to pull all pixels lighter than the black target pixels up to 100% white. This produces a black key on a white background, which can be inverted later if needed. Alternately, the color curve could pull the black pixels up to 100% white and all others down to black to produce a white key on a black background directly.

WWW

Fig 2-7 luma key target.tiff – use this image to test your skill at a soft-edged luma key for an easy white target.

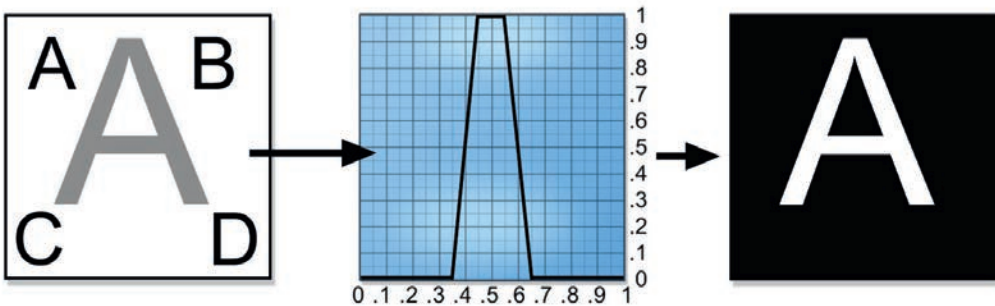


Figure 2.8 Luma key for a middle gray target

Very often the target object is a middle-gray, like the big gray “A” in **Figure 2.8**. The pixel values of the target object are pulled up to 100% white with the color curve, and all pixel values greater and less than the target are both pulled down to black. The sloped sides ensure a nice soft edge to the key. Occasionally the luma key is pulled on the darkest or lightest element in the frame so the simple soft clip shown in **Figure 2.7** can be used, but most often it will be some middle gray like this example where you have to suppress pixels that are both lighter and darker than the target.

WWW

Fig 2-8 luma key target.tiff – use this image to test your skill at a soft-edged luma key for this less easy middle-gray target.

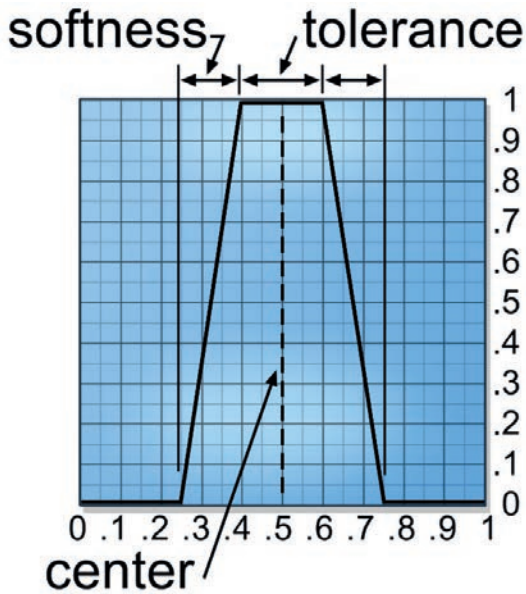


Figure 2.9 Details of a luma key scaling operation

Figure 2.9 shows the details of shaping the luma key with a curve in the color curve node. The “center” represents the average luminance value of the target. The “tolerance” is how wide a range of luminance will end up in the 100% white region of the resulting key. The “softness” reflects the steepness of the slope between black and white, and the steeper this slope the harder the key edges will be. Only four **control points** are needed to adjust the tolerance and softness. Measuring a few pixel values will give you a starting point, but you will probably end up adjusting the four control points visually while watching the resulting key in order to see the effects of the changes to the tolerance and softness.

Rolling your own luma key with a color curve like this does have one distinct advantage over the off-the-shelf luma keyer, and that is the ability to adjust the curvature of the curves – they don’t have to be the straight lines of a regular luma keyer. This gives you one more layer of control over the edges of your key, and in compositing control is the name of the game.

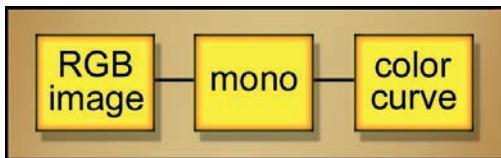


Figure 2.10 Flowgraph of luma key operation

To summarize the homemade luma key process, **Figure 2.10** shows a flowgraph of the operations described above. The RGB image is piped into a “monochrome” node, which makes a luminance version of color images. Hopefully yours has internal settings that can be tweaked for best results. The mono image is then piped to a color curve to scale it up to the final key. Alternately, instead of the mono node, a single channel of the RGB image might be used, or perhaps some mix of two channels as described above for making your own luma keys.