

SECOND EDITION

# VIDEO COMPRESSION HANDBOOK

Andy **BEACH**  
Aaron **OWEN**

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



SECOND EDITION

# VIDEO COMPRESSION HANDBOOK

Andy **BEACH**  
Aaron **OWEN**

## **Video Compression Handbook, Second Edition**

Andy Beach and Aaron Owen

Peachpit Press

[www.peachpit.com](http://www.peachpit.com)

Copyright © 2019 Andy Beach and Aaron Owen. All Rights Reserved.

Peachpit Press is an imprint of Pearson Education, Inc.

To report errors, please send a note to [errata@peachpit.com](mailto:errata@peachpit.com)

### **Notice of Rights**

This publication is protected by copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit [www.pearsoned.com/permissions/](http://www.pearsoned.com/permissions/).

### **Notice of Liability**

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

### **Trademarks**

Unless otherwise indicated herein, any third party trademarks that may appear in this work are the property of their respective owners and any references to third party trademarks, logos or other trade dress are for demonstrative or descriptive purposes only. Such references are not intended to imply any sponsorship, endorsement, authorization, or promotion of Pearson Education, Inc. products by the owners of such marks, or any relationship between the owner and Pearson Education, Inc., or its affiliates, authors, licensees or distributors.

Credit: Table 3.2 Data from: White Paper: Blu-ray Disc Read-Only Format: 2.B Audio Visual Application Format Specifications for BD-ROM Version 2.5. Blu-ray Disc Association. July 2011; and “White Paper: Blu-ray Disc Read-Only Format (Ultra HD Blu-ray): Audio Visual Application Format Specifications for BD-ROM Version 3.0.” Blu-ray Disc Association. July 2015.

Executive Editor: Laura Norman

Editor: Karyn Johnson

Copyeditor: Kim Wimpsett

Proofreader: Patricia Pane

Technical Editor, Chapter 9: Steve Llamb

Senior Production Editor: Tracey Croom

Compositor: David Van Ness

Indexer: Jack Lewis

Cover Design: Chuti Prasertsith

Interior Design: David Van Ness

ISBN-13: 978-0-13-486621-5

ISBN-10: 0-13-486621-5

Andy:

*This book is dedicated to my wife, Lisa, and daughter, Davy.*

Aaron:

*For Katie, the love of my life.*



# Contents at a Glance

Introduction .....	xi
Chapter One: Understanding Video and Audio .....	1
Chapter Two: Understanding the Language of Compression .....	19
Chapter Three: Planning and Preparation .....	43
Interview with a Compressionist: Derrick Freeman .....	73
Chapter Four: Preprocessing Video and Audio .....	77
Interview with a Compressionist: Bryce Castillo .....	97
Chapter Five: Compression Tools .....	101
Interview with a Compressionist: Alex Zambelli .....	129
Chapter Six: Compression for Video-on-Demand Delivery .....	133
Interview with a Compressionist: Doug Daulton .....	161
Chapter Seven: Compression for Live Delivery .....	165
Chapter Eight: Compression for Optical Discs .....	183
Interview with a Compressionist: Aaron Owen .....	207
Chapter Nine: Compression for Digital Cinema and Broadcast .....	211
Chapter Ten: Compression for Virtual Reality and 360 Video .....	261
Index .....	273

# Table of Contents

<b>Introduction</b> .....	<b>xi</b>
<b>Chapter One: Understanding Video and Audio</b> .....	<b>1</b>
Elements of Video .....	2
Frames and Fields .....	3
Progressive Scan Video .....	4
Resolutions .....	5
Aspect Ratio .....	7
How Compression Works .....	9
Lossless and Lossy Compression .....	11
Spatial (DCT) and Wavelet Compression .....	11
Quantization .....	13
Interframe and Intraframe Compression .....	13
High Dynamic Range .....	14
Audio Compression .....	15
Online Video Delivery .....	16
Conclusion .....	18
<b>Chapter Two: Understanding the Language of Compression</b> .....	<b>19</b>
Compression Parameters .....	20
Data Rates (or Bit Rates) .....	20
VBR and CBR .....	21
Frame Rate .....	21
Frame Types .....	22
Aspect Ratios and Letterboxing .....	24
Square and Nonsquare Pixels .....	24
Containers .....	25
Containers for Audio .....	25
Containers for Various Media Types .....	25
Codecs .....	26
A Time and a Place for Everything .....	26
Acquisition Codecs .....	28
Mezzanine Codecs .....	29
Delivery Codecs .....	30
Players .....	33
Desktop Players .....	33
Built-in Players .....	34

Third-Party Players .....	36
HTML5 Players .....	38
Conclusion .....	42
<b>Chapter Three: Planning and Preparation .....</b>	<b>43</b>
What Are Your Compression Needs? .....	44
Final Delivery or Work in Progress? .....	44
Are You the Master of Your Destiny? .....	44
Long-Form or Short-Form Content? .....	44
What Is Your Final Delivery Target? .....	45
How Is Your Content Delivered to You? .....	45
How Much Content Are You Compressing? .....	46
Is Compression All You're Doing? .....	47
Quality or Timeliness? .....	48
How Knowledgeable Are Those with Whom You Work? .....	48
Equipment and Workflows for Different Scenarios .....	49
Encoding as an Editor .....	49
Encoding as a Compressionist .....	52
Automating the Encoding Process .....	52
Enterprise Systems: The Big Leagues .....	53
Essential Encoding Equipment .....	54
Hardware .....	55
Software .....	60
Productivity Tips .....	66
Minimizing Quality Loss Without Overdoing It .....	66
Stay Organized .....	67
Filenaming Conventions .....	68
Experimenting Can Be a Good Thing .....	70
Mezzanine, or In-Between, Files .....	71
Archiving and Transcoding .....	72
Conclusion .....	72
<b>Interview with a Compressionist: Derrick Freeman .....</b>	<b>73</b>
<b>Chapter Four: Preprocessing Video and Audio .....</b>	<b>77</b>
Spatial/Geometric Preprocessing .....	78
Cropping: Title and Action Safe Zones .....	78
Scaling .....	80

Temporal Preprocessing . . . . .	82
Easy Frame Rate Conversions . . . . .	82
Telecine and Inverse Telecine . . . . .	83
The Difficult Conversion . . . . .	84
Color and Image Preprocessing . . . . .	85
Luma Adjustments . . . . .	85
Gamma Adjustments . . . . .	87
Brightness and Contrast . . . . .	87
Chroma Adjustments . . . . .	87
Saturation . . . . .	88
Hue . . . . .	88
Noise Reduction . . . . .	88
Legacy Video Issues . . . . .	89
Deinterlacing Video . . . . .	90
Image Aspect Ratio Correction . . . . .	92
Pixel Aspect Ratio Correction . . . . .	93
Audio Preprocessing . . . . .	94
Adjusting Volume . . . . .	94
Noise Reduction . . . . .	95
Stereo Mixdowns of 5.1 Surround Mixes . . . . .	95
Conclusion . . . . .	96
<b>Interview with a Compressionist: Bryce Castillo . . . . .</b>	<b>97</b>
<b>Chapter Five: Compression Tools . . . . .</b>	<b>101</b>
Free and Low-Cost Tools . . . . .	102
HandBrake . . . . .	102
MPEG Streamclip . . . . .	107
Middle-of-the-Road Tools . . . . .	111
Adobe Media Encoder CC . . . . .	111
Apple Compressor . . . . .	115
Enterprise-Grade Tools . . . . .	122
On-Premise: Telestream Vantage . . . . .	123
On-Premise: AWS Elemental Server . . . . .	124
Cloud-Based Systems . . . . .	126
Hybrid Cloud Systems . . . . .	127
Conclusion . . . . .	128
<b>Interview with a Compressionist: Alex Zambelli . . . . .</b>	<b>129</b>

<b>Chapter Six: Compression for Video-on-Demand Delivery</b>	<b>133</b>
The State of Online Video	133
Video-Hosting Websites	135
Publishing for Yourself	135
Publishing as a Business	135
Web Application Video Ecosystem	136
Recipes for Encoding Video for the Web	137
Encoding for YouTube	137
Encoding for HTML5 Video	146
Encoding an ABR File for Video-on-Demand Delivery	151
Conclusion	160
<b>Interview with a Compressionist: Doug Daulton</b>	<b>161</b>
<b>Chapter Seven: Compression for Live Delivery</b>	<b>165</b>
Basic Live Streaming Equipment	166
Camera	166
Capture Device	166
Encoder	167
Video Streaming Host	167
Cloud-Based Transcoding	167
Live Streaming Considerations	168
Push vs. Pull	168
Content Delivery Networks	169
Challenges to Live Streaming	170
Social Media and Live Streaming	170
Popular Live Streaming Platforms	171
Recipes for Live Streaming Video	173
Streaming to Facebook Live via OBS	173
Streaming to YouTube Live Using XSplit	177
Conclusion	181
<b>Chapter Eight: Compression for Optical Discs</b>	<b>183</b>
What's a DVD?	184
Producing SD DVDs	184
DVD Disc Types	185
SD Codecs	187
What the Heck Is a VOB?	188
What's a Blu-ray Disc?	189

Producing Blu-ray Discs . . . . .	191
Blu-ray Overview . . . . .	191
Recording Blu-ray Discs . . . . .	193
That Whole Bit Budget Thing . . . . .	194
Recipe for Producing Blu-ray Discs . . . . .	196
Working in Premiere Pro . . . . .	197
Producing an H.264 Blu-ray-Compatible File in the Adobe Media Encoder . . . . .	198
Creating a Blu-ray Disc in Encore . . . . .	200
Recipe for Producing SD DVDs . . . . .	203
Conclusion . . . . .	206
<b>Interview with a Compressionist: Aaron Owen . . . . .</b>	<b>207</b>
<b>Chapter Nine: Compression for Digital Cinema and Broadcast . . . . .</b>	<b>211</b>
A Brief History of Package Formats . . . . .	211
Dissecting a DCP . . . . .	213
What Comprises a DCP? . . . . .	214
Digital Cinema Naming Convention . . . . .	216
More About CPLs . . . . .	218
Encryption . . . . .	218
Version Files . . . . .	219
Creating a DCP . . . . .	220
Workflow for Creating a DCP . . . . .	220
Encoding Picture Assets . . . . .	220
Preparing the Audio . . . . .	224
Creating Subtitles and Captions . . . . .	226
Wrapping Assets into a Package . . . . .	228
Recipes for Creating a DCP . . . . .	230
Example DCP: Recipe 1 . . . . .	230
Example DCP: Recipe 2 . . . . .	245
Closing Thoughts on DCP . . . . .	252
Understanding the Interoperable Master Format . . . . .	254
What IMF Is Not . . . . .	254
IMF's Heritage . . . . .	255
IMF Applications . . . . .	255
Anatomy of an IMF and IMP . . . . .	256
Supplemental IMPs . . . . .	258
QC and IMP . . . . .	259
Conclusion . . . . .	260



**Chapter Ten: Compression for Virtual Reality and 360 Video . . . . . 261**

    VR Concepts . . . . . 261

        What Is Virtual Reality? . . . . . 261

        What Is Augmented Reality? . . . . . 262

        What Is Mixed Reality and Extended Reality? . . . . . 263

    VR Hardware . . . . . 264

        Basic Handheld Kits. . . . . 264

        Premium Smartphone VR . . . . . 265

        Tethered Headsets . . . . . 265

    360 Video Overview . . . . . 267

        Monoscopic vs. Stereoscopic . . . . . 268

        Video Resolution vs. Perceived Resolution . . . . . 268

        Equirectangular vs. Cube Map vs. Pyramid Projection . . . . . 268

        Codecs . . . . . 269

        Streaming vs. Downloading. . . . . 270

        Device-Specific Targeting. . . . . 270

    Optimizing Your 360 Video . . . . . 270

    VRecorder . . . . . 271

    Conclusion . . . . . 272

**Index . . . . . 273**

# Preface

Ten years ago, when the previous edition of this book was written, it was already established that video was a major part of our lives. Nothing has stopped this from being true, and in fact we are continually finding ways to make it an integral part of our daily lives. Digital video, compressed or otherwise, is all about bits. It seems like it should be easy to move those bits from one place to another, and yet it isn't. A myriad of little settings, tweaks, steps, and reasons make the process of moving video from one medium to another a frustration. If you mess it up, the results are unwatchable at best and unplayable at worst. With so many possibilities as to what can go wrong, troubleshooting becomes a prolonged guessing game, punctuated by panic-inducing deadline checks.

Sorry, we didn't mean to make you hyperventilate.

It's truly amazing how much video compression occurs now. The phones in our pockets are basically portable film studios, allowing us to capture, edit, compress, and publish video while we're on the move. Traditional broadcasters have capitulated and started making their shows available online, and IP video has become *the* infrastructure for modern studios. Oh, and did we mention Twitch and its army of broadcasters delivering game content to millions of viewers daily?

Now we're the ones hyperventilating.

## What This Book Is Not

This book is decidedly not a technical tome on the nature of video compression and streaming media. We will not get into the science of encoding (well, maybe just a tiny bit), and we won't explain how committees of smart people

came up with the standards we use to make this all work. Yes, we will discuss some fairly complex topics, but believe us, this is just the beginning. There is a world of math and science associated with this technology, but for the most part it doesn't mean a lick if you're in the last hour of your production, trying to figure out why your encodes are failing.

Though we'll discuss some aspects of video production, this book is not about shooting the best video possible. We have been on our fair share of shoots, but don't feel qualified to try to tell you the best way to shoot video. Plenty of other books do that. Also, we will not explain how to do visual effects, how to edit video better, or how to troubleshoot hardware and software.

## What Is This Book About, Then?

This book focuses on the essential information you need to get your video online, on disc, in theaters, or onto some other device. Sometimes discussions in compression get bogged down in all the options available. What we'll try to do in this book is present you with some fundamentals and some best practices.

We'll start by covering the fundamentals of video and video compression, giving you the most essential background information you'll need to know. Next, we'll explore the current applications and web services you'll most likely run across while encoding your video. We'll discuss their merits and explain the most common workflows for each. Finally, we'll discuss the most common delivery media for your compressed video and use those same applications to create some specific content for various delivery media.

Our hope is that by combining the fundamental information, practical knowledge about the tools of the trade, and insight into the delivery platforms, we'll leave you well armed to tackle the variety of content that's thrown your way.

But we didn't want you to hear this just from us, so we asked a variety of friends and co-workers to contribute their own insight into the world of video compression. Their profiles are intermingled with the chapters, and although they all are involved in video compression, their professions are incredibly varied.

## Who Should Read This Book

If you have ever suddenly had to add video compression to your regular work responsibilities, willingly or otherwise, then this book is for you. But if you have also been doing this for a few years and are suddenly learning about new topics such as HDR or DASH, then this book is also for you.

Video compression, encoding, and transcoding (they are all more or less the same) used to be the exclusive province of engineers who specialized in these tasks and were worried about meeting transmission specs or addressing other technical issues. Today, pretty much anyone involved in the Web, traditional film and video production, and other interactive media realms will need to know at least a little bit about compression. In addition, the Web has created a new wave of individual publishers who stream their gameplay, publish video podcasts, or have created whole shows on YouTube.

So, if you were ever disappointed (or perhaps even shocked) by the results of your video compression, then read on, and we hope to provide some help.

That's not to say this book is just for newbies. Although this book lays out the fundamentals of video and video encoding, it contains a variety of workflow suggestions and deliverable-specific information that can help even the most seasoned compressionists improve their games. If there is one thing we have learned in this profession, it's that you will always be learning new things because the industry is ever evolving.

## About the Authors

Andy Beach was a video geek who got into this business after starting in the postproduction world. He quickly realized that making video look good was hard and was worth spending time to learn to do better and to pass that knowledge on to others. His career took him to the startup world, where he built encoders and live streaming tools for a living, and eventually to his role at Microsoft, where he works with a large-scale commercial media infrastructure.

Aaron Owen got his start in live television and then continued his career in commercial postproduction, playing a key role in transforming Phoenix Editorial (now Bonfire Labs) from a three-suite boutique editorial company

into a full-fledged facility. Leveraging his vast advertising and high-end corporate experience, he founded Cinematiq in 2012 with the goal of bringing high-end postproduction techniques to filmmakers everywhere.

## Acknowledgments

We've had the privilege of meeting and working with a number of talented people in both the production and software development worlds, including each other!

In the combined 20 or so years of work in this industry, we have met and worked with a lot of very smart (often scary-smart) people.

In addition to the people profiled in the book, a variety of other real-world compressionists we know deserve heartfelt thanks. Thanks to Jem Scholfield, of Buttons Production and theC47, for mulling over new announcements and prognosticating their meaning (and for not being afraid to split a bucket of mussels and a few pints of beer). Thanks to Colleen Henry at Facebook, who is never afraid to offer an opinion or tell you why you are wrong about live streaming. Steve LLamb at Deluxe kept us honest when it comes to all things digital cinema. John Simmons of Microsoft is working to ensure there are standards for the video world so that the new technologies we create work everywhere we want them to (and best of all, he'll take our calls anytime we ask).

We also sorely need to thank the nice folks at Pearson who have patiently waited for us to complete this book. Karyn Johnson, who helped make the first edition a reality, patiently listened to all of our excuses and then held our feet to the fire to get things done. Likewise, Laura Norman helped keep us on track throughout the process. Production editor Tracey Croom worked with us on the layout and turned the text and images into a well-crafted book. Kim Wimpsett did an amazing job of copyediting and often turned around feedback to us so quickly it made our heads turn. Last, but certainly not least, Patricia Pane continually removed all the poor grammar, odd hyphens, and other elements we kept putting in.

Andy would also like to thank his work colleagues, who listened to him talk about the process and complain about changes to the industry and the late nights involved in writing this book. Every time he yawned in a meeting,

they knew it was because he'd been up working on this book. For that, thank you, Greg Oliver, Jit Ghosh, LaBrina Loving, Luis Guerrero Guirado, Robert Eichenseer, Tiago Andrade e Silva, and Tobias Weisserth.

And, of course, none of this would have been possible at all without the love and support of our families. For Andy, his wife, Lisa Weisman, and Davy Beach couldn't care less about an inverse telecine error but are always happy when he can figure out why Hulu stopped working on the television. For Aaron, his soon-to-be wife, Dr. Katie Bean, and business partner, Paul Scolieri, deserve great thanks for allowing him the time to put the headphones in and the head down.



*This page intentionally left blank*

# Preprocessing Video and Audio

# 4

You have shot and edited your video, and now it's time to compress, encode, and deliver it. But one more step is left before final encoding can begin. This step is called *preprocessing*, and it consists of a variety of optimizations you need to perform on the video and audio before you can hand them off to the encoder. These optimizations can include deinterlacing, inverse telecining, cropping, scaling, aspect ratio adjustments, noise reduction, brightness and color corrections, and audio adjustments.

Preprocessing of some sort is almost always necessary to get your video to look its best. The goals of preprocessing are to clean up any noise in the video and to optimize its basic elements for playback on the devices you are targeting (TVs, mobile phones, computers, and so on). Preprocessing is sometimes referred to as the “magic” part of compression because it takes a lot of practice to achieve the desired results. This is the most artistic and frequently misused part of the compression process. It's easy to go overboard with the changes until you get a feel for how much you really need to do to your video and audio to optimize it for encoding.

Fortunately, preprocessing is a craft that can be learned, and practice will only make you better at it. Understanding why you preprocess video and audio and how the various types of optimization that occur at this stage affect your final product will help you make your preprocessing choices.

## When and What to Preprocess

To some degree, every piece of video will need some preprocessing, but the amount is wholly dependent on both the source video and the format you are creating. There are basic things such as scaling and changing the frame rate that will often be required, but color and luma changes should be used judiciously to avoid taking away from the intent of the content. Often, new compressionists will want to pull out all the bells and whistles when preprocessing, and the results may be negligible at best. Add to that the fact that any additional processing you are asking the machine to do to the content will lengthen the overall encoding process.

Remember, the goal of preprocessing is to improve the quality of the video and audio and to maintain the original creative intent. So, it's best to use a light hand most of the time.

## Spatial/Geometric Preprocessing

Whether it's simply changing the frame size and aspect ratio to match the destination or reframing the video entirely, spatial preprocessing is one of the most common types of preprocessing you'll encounter as a compressionist.

### Cropping: Title and Action Safe Zones

*Cropping* is a way of identifying a specific region of the video to use in the compression, excluding the other areas of the source frame. Cropping can be used to change the aspect ratio of a wide-screen 16 × 9 video to the older 4 × 3 format, or it can be used to crop out unwanted picture areas at the edge of the frame.

Most TVs do not display the entire image that is transmitted. Instead, they are *overscanned*, meaning slightly larger than the viewable area of a consumer-grade television. This is done for several reasons, most of which culminate in needing to hide irregularities that exist in the edges of the video frames. Production people are aware of this and have created three

regions of a video image that affect how they frame shots and incorporate graphic overlays: overscan, action safe, and title safe. **Figure 4.1** will be familiar to anyone who has looked through a video camera viewfinder. It depicts the action- and title-safe areas that a camera operator needs to be conscious of when framing a shot.



**Figure 4.1** Compression-ists need to be aware of the action- and title-safe regions of the frame and use them as a general guideline for cropping nonbroadcast content.

The outermost region of the video is known as the *overscan* region. This area may not appear on standard consumer TV screens, and it often contains the edge of the set or cables and other equipment. Professional-grade monitors have a mode that allows this overscan area to be viewed, known as *underscan* mode. These monitors may also include action-safe and title-safe indicators.

The *action-safe* area is the larger rectangle within the image area. This area displays approximately 90 percent of the video image and is where camera operators will make sure to keep the primary action framed to keep it viewable on TVs.

The smaller rectangle in the image is the *title-safe* area (comprising about 80 percent of the visible image). It is far enough in from the four edges of a standard TV that text or graphics should show neatly without being cut off or distorted. The title-safe area started out as a guide for keeping text from being distorted by the rounded corners on old cathode ray tube (CRT) TVs. Most modern TVs display a lot more of the area outside of the title-safe zone

than their CRT counterparts. However, as a rule of thumb, text and titles should still be contained within the title-safe area.

Cropping can be a useful tool to ensure that the video displays the same regardless of delivery and playback format. While TVs routinely overscan, mobile devices and online video players do not. If you want the video to appear to display the same on all devices, it may be necessary to crop the overscan area on the formats that do not overscan.

## Scaling

*Scaling* is another key part of the preprocessing process, and it simply means either enlarging or shrinking the frame size. A *proportionate scale* means that the same scaling factor is applied to both the horizontal and vertical axes. Many times you'll crop an image and then scale it up so that the frame size stays the same as it did when you started. Other times, it may mean shrinking a video from the original size to a size more appropriate for delivery (Figure 4.2).

**Figure 4.2** These images demonstrate how a  $720 \times 480$  source clip (left) can be scaled down to  $320 \times 240$  (right) for web delivery with no loss in quality.



Scaling video up is called an *upconvert*. Going from a small frame size to a higher frame size (as shown in Figure 4.3) is not recommended but can be necessary in certain situations. In these situations, the different scaling algorithms in different tools can produce dramatically different results. Some tools will simply scale pixels, while others will *interpolate*, or create new pixels. There are lots of options for tools in this area ranging from lower-cost

plug-ins for After Effects all the way up to dedicated hardware for real-time SDI conversions. As always, your mileage may vary, and it's always best to test your content against multiple tools to determine the best tool for the job.



**Figure 4.3** *Upscaling, unlike downscaling, is a bad idea—you can't add pixels that weren't in the image to begin with without compromising quality.*

A key question when scaling is what size should you scale to? The answer will be determined by your destination and the frame sizes supported. As bandwidth has increased and technologies such as adaptive bitrate streaming are invented, the answers to these questions have gotten both simpler and more complex.

While there are all sorts of possibilities, there are two general facts: the larger the frame size, the bigger the output file will need to be to maintain good quality, and the slower it will play on some machines. Of course, small frame sizes could also be low quality, so you need to find a good balance that works for your projects. One thing to keep in mind, however, is that it is always best to scale down rather than scale up. That means if you have a video finished at  $1920 \times 1080$ , creating an upconverted 4k/UHD version won't mean higher quality.

Here are some general guidelines, regardless of aspect ratio:

- **Height of 2160 pixels:** At double the size of HD video, this is a great high-quality encode if you don't need to worry about file size or playback speed (some older computers will have difficulty playing back files this large without dropping frames).
- **Height of 1080 pixels:** This is probably the most common delivery size today. Most platforms and players support some version of 1080p.



- **Height of 720 pixels:** This is a good choice if you need to prioritize file size over spatial resolution. It can also be appropriate if your target machine is older and slower.
- **Height of 480 pixels and smaller:** What used to be called *standard def*, these frame sizes are pretty small. These are more suited to mobile video, given the screen resolution and bandwidth.

For more details on general guidelines for frame sizes, see **Table 4.1**. These ratios assume that the pixel aspect ratio is 1:1 (square).

**Table 4.1** *Common frame sizes*

Output	Full-Screen (4:3)	Wide-Screen (16:9)
Ultra-high definition	2880 × 2160	3840 × 2160
Extra-large broadband	1920 × 1440	2560 × 1440
Large broadband/high definition	1440 × 1080	1920 × 1080
Small broadband and large mobile	960 × 720	1280 × 720
Large mobile	640 × 480	854 × 480
Medium mobile	480 × 360	640 × 360
Small mobile	320 × 240	426 × 240

# Temporal Preprocessing

Whatever the reason, sometimes you'll need to adjust the frame rate of your video. Most of the time this is done to better match the source to the destination. Whatever the reason, temporal processing can be one of the most difficult steps to get right.

## Easy Frame Rate Conversions

Certain frame rate conversions aren't too hard and usually yield good-quality results. For example, when the source frame rate is a fractional frame rate, say 29.97 fps or 23.976 fps, converting to its corresponding integer frame rate—30 fps or 24 fps in this case— isn't too hard. You'll just change the amount of time each frame appears on the screen slightly. Audio will need to be adjusted to stay in sync, but most modern tools can do this compensation for you. Another simple conversion is when you want to go from a higher

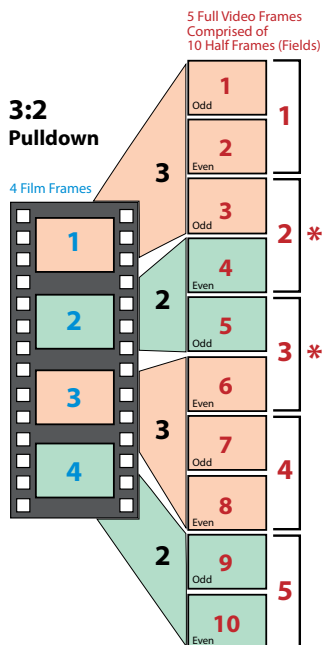
frame rate to one that is half as fast (or the other way around), such as 59.94 fps to 29.97 fps. In this conversion, you'll simply drop (or duplicate) every other frame to get from the source rate to destination rate. Most modern nonlinear editing (NLE) systems can do these conversions in the timeline.

## Telecine and Inverse Telecine

*Telecine* is the process of transferring motion-picture film into electronic form. The term is also used to describe the machine used in this process. Traditional motion-picture film runs at exactly 24 fps progressive scan, which doesn't easily convert to 29.97 fps for NTSC.

Converting from film to PAL is easy. The video is sped up 4 percent to 25 fps and converted into progressive PAL.

The telecine process for NTSC is more complex: the film is first slowed down 0.1 percent to 23.976 fps and is then converted to 29.97 fps by using a process called *3:2 pulldown* (**Figure 4.4**). In this process, the first frame of film becomes three fields of video, the next frame becomes two fields of video, the next becomes three fields, and so on, resulting in two out of five frames having fields that come from different source frames.



**Figure 4.4** In the 3:2 pulldown process inherent to NTSC telecine, four frames of film become ten fields/five frames of video by using a three-then-two field pattern to maintain the smooth motion of the film in the newly created video.

To correctly compress video that has been telecined, it is crucial to be able to identify these duplicate frames and then remove them. *Inverse telecine* basically reconstructs the four frames from every five to turn the source back into progressive video.

## The Difficult Conversion

There's one particularly tricky conversion to be aware of. It occurs when your source is 30 fps (or 29.97) and your destination is 24 fps (or 23.976). Now when we say "tricky" or "difficult," we mean that it's difficult to get it to look good. Let's dive in to see why.

When you want to get from 24 fps to 30 fps, you can add interlaced 3:2 pulldown to a clip to achieve the desired result with minimal unwanted motion artifacts. This is because you're using fields created from a progressive image to create additional frames in a pattern that gives six additional frames per second that are distributed in an even cadence. You can reverse this process to get from 29.97i to 23.976p.

The problem arises when you start with a progressive source sequence and need to remove frames from it to lower the frame rate. To keep the dropped frames in an even cadence, you need to drop every fourth frame. The problem is that if you have smooth motion in your shot, like a car driving through the frame from left to right. The change in the car's position between each frame is consistent, which means when you remove one of those frames, the car appears to move further between the third and fifth frames, which makes it look like it jumps. When this is repeated six times per second, the result is a really jumpy-looking video that is unpleasant to watch.

Just like with the upconversion process, the way to try to deal with this motion artifact is through interpolation. While there are a lot of standards converters on the market, there are a couple to take a look at for this specific conversion: Cinnafilm's Tachyon and BlackMagic Design's Teranex. But be warned, these tools can be fairly expensive and sometimes often yield subpar or unacceptable results depending on the content. Many times, the best way to deal with this situation is to manage expectations or try to find a solution that doesn't involve this conversion.

# Color and Image Preprocessing

When working with finished content that has already been color corrected, most of the time you won't need to make further adjustments. However, certain situations where adjustments are necessary do arise from time to time. For example, if you have a show that's being encoded for broadcast and mobile, you might need to adjust the mobile version to match the look of the broadcast version since the two formats use different color spaces. Understanding what adjustments are available will allow you to evaluate your outputs and determine whether any adjustments are necessary.

## Luma Adjustments

Image adjustment is the process of using controls, similar to those in image-editing applications, to change aspects such as brightness and contrast. Digital video is described in either RGB or in Y'CbCr (aka YUV) color space. A *color space* (or *model*) is a way of describing and specifying a color. RGB and the Y'CbCr color space formulas contain three variables, also known as *components* or *channels*. RGB's variables are red, green, and blue, while Y'CbCr breaks down as follows: Y is luma (or black and white or lightness), and CbCr is chroma or color (Cb is blue minus luma, and Cr is red minus luma).

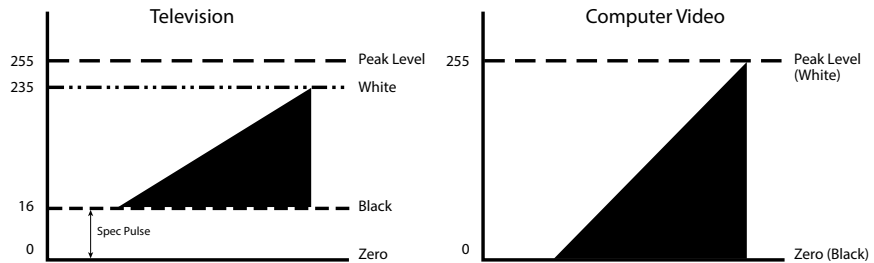
This representation addresses the human eye's predisposition to green-light sensitivity, which is why most of the information about the proportion of green is in the luma (Y), and only the deviations for the red and blue portions need to be represented. The Y values have twice the resolution of the other two values, Cb and Cr, in most practical applications, such as on DVDs.

Because different color spaces are used by different codecs and video playback devices, image adjustment may be required as part of preprocessing.

## Luma Range Expansion

The RGB color space breaks the steps from black to white into 256 even steps (0 to 255, with 0 being black and 255 being white). Standard TV has only 220 steps from black to white, using black as 16 and white as 235 within the same scale (**Figure 4.5** on the next page). Generally, modern postproduction applications automatically keep black and white consistent. However, if blacks and whites appear either crushed (too black) or faded (too white), the luma range may need to be remapped.

**Figure 4.5** *Television and computer-based video use different luma ranges to depict the range between black and white. When this is not compensated for in preprocessing, the distortion of the luma range can leave video either washed out or too dark.*



## The Digital Color Meter

Apple users have some built-in help when figuring out luma and chroma values. In the Utilities folder (Application > Utilities) is the Digital Color Meter (Figure 4.6). This little application launches a small magnifier and a table that shows the specific RGB value of the samples part of the screen. This application is a handy way of checking values, rather than just trusting that your display is calibrated correctly. You can learn more about it here: <https://support.apple.com/guide/digital-color-meter/welcome/mac>.



**Figure 4.6** *Apple's Digital Color Meter makes it easy to check luma and chroma values.*

**TIP** Using noise reduction in addition to adjusting the luma may make achieving proper levels easier.

## Gamma Adjustments

*Gamma* is a measurement of how the luma range turns into brightness when it is displayed. Although black and white will look the same under different gamma values, the middle of the range will look different. Gamma correction is a complex mathematical concept that works to best encode images according to how human image perception works. Different computer platforms and TVs all have different default gammas, with the majority of gamma values falling between 2.2 and 2.4. Before OS X 10.6 Snow Leopard, Apple used a default gamma value of 1.8. CRT television sets used a gamma of 2.2, and Windows systems used gamma values ranging from 2.2 to 2.5. Modern LCD televisions use a gamma value of 2.4. If the value used to encode the video is not the same as the display/output gamma value, the image will appear either slightly darker, which will lose detail in the shadows, or slightly brighter, which will make colors appear washed out and “milky.”

If you are encoding for a mixed environment or are unsure of what playback device the end user has, it's best to target an encode value of 2.2.

## Brightness and Contrast

*Brightness* filters adjust the overall intensity of each pixel in a frame by a fixed amount. *Contrast* settings, on the other hand, increase or decrease each pixel's value by an amount proportional to how far away from the middle value it is (the farther away, the bigger the change). These values are often used together, rather than separately, because just brightening an image without adjusting the contrast can leave the black levels too muddy gray, rather than the desired black.

## Chroma Adjustments

Chroma adjustments affect the color (*chrominance*) of a video image, similar to the way luma affects the brightness. Changes to the color of the video aren't commonly required. In fact, some compression tools do not even include these filters.





**NOTE** Digital video has a narrower saturation range than RGB, so slightly increasing the saturation will make video look more vibrant on computer screens.

## Saturation

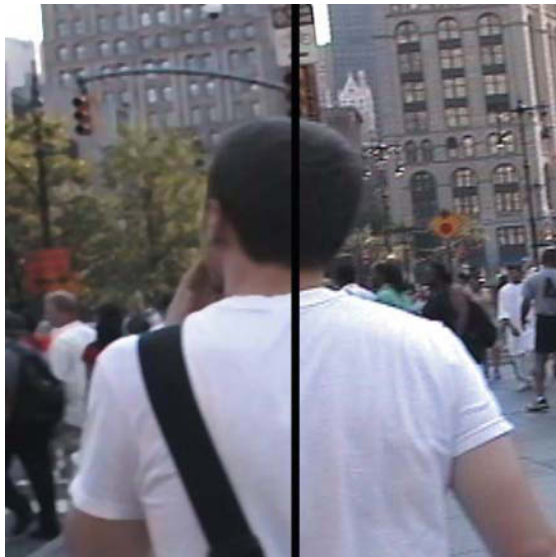
*Saturation* controls the intensity of a color. Increasing saturation will brighten a color artificially, while decreasing it (or desaturating) will remove the color, all without affecting the luminosity of the image (turning it to black and white).

## Hue

*Hue* is the way of describing a color (for example, red or blue). The hue adjustment in compression tools shifts the color spectrum of the image. It is used to do simple color correction, most often when white levels or flesh tones do not appear correctly. This is a pretty coarse adjustment and should be used with care or not at all.

## Noise Reduction

Noisy video is a big headache for video compressionists. Random noise causes pixels to change rapidly and constantly, making it difficult for the codec to encode the clip correctly.



**Figure 4.7** Here are two versions of a DV-NTSC clip. On the right it has had a median filter applied. Note the increased softness in the image on the left but the smoother quality of the image also.

Noise reduction encompasses a variety of techniques to remove that noise from an image, making it easier to compress. Simple noise reduction algorithms are just blurs that hide grain but make the image softer, such as when a median filter is applied (**Figure 4.7**). More advanced algorithms try to blur only the parts of the image that have grain and may take advantage of differences among frames.

### Noise Reduction Before Compression

If the damage from video noise is significant enough, such as dust or scratches in film, no amount of blurring will help remove this damage completely from the image during compression. In this case, you may want to use a stand-alone application, such as Adobe After Effects or Photoshop, to digitally remove the scratches either manually (with a clone tool) or with an automatic scratch removal filter. Once you've treated the source footage to remove the damage, you can then compress the footage as you normally would.

Noise reduction is not always required. If you have a clean video source to work with, simply skip the noise reduction step when preprocessing your work. If you do have damaged footage, it may take several experiments to find the right balance of cleaning up the source without degrading the finished video image.


## Legacy Video Issues

There has been a lot of change in the realm of digital video. We've moved from interlaced to progressive formats. The number of standard aspect ratios has exploded. So-called high-definition resolutions have gone from being in the minority to the majority. Even higher resolutions like ultra-high definition, 4k, and even 8k are now coming into play. But as a compressionist, you'll be asked to work with all sorts of video from various periods in time. That means there's a whole host of issues that come with older video that you'll need to understand and be comfortable with fixing.

## Deinterlacing Video

In the past, most digital video was interlaced because it was assumed that at some point it was meant for playback on a standard CRT television, which had an interlaced display. We've now moved to a world where most displays are progressive scan, and they are not always televisions. Because interlaced video is unacceptable for web and mobile delivery, deinterlacing video for playback on the Web or other progressive displays is a fundamental and necessary step.

If the source video is left with interlacing lines intact, the output will appear as jagged lines sometimes referred to as *combing*. The lines do not make for a good viewing experience and are also difficult to encode for a couple of reasons. Moving objects will keep merging and reassembling (with the two fields moving out of step), and motion estimation will become difficult, thus making the encode inefficient. The interlaced image will also have more detail than is necessary to display the image, so additional bits are wasted when relaying this redundant detail.

 **TIP** It is rare for a compression application to offer all the deinterlacing options listed here, and each tool may have its own naming scheme for them. Experiment with the options to find the ones that work for your content.

You can perform a deinterlace in several ways. Each is designed to optimize different types of images and different types of motion within the video. All in all, there are approximately eight ways to perform a deinterlace, though those eight have several different names by which they are recognized, depending on the tools you are working with.

### Blend

The first common deinterlace method is referred to as *blending*, also known as *averaging* and *combining* fields. This method involves both fields being overlaid. This method gives you good results when there's no movement, but it results in unnatural, low-quality movements. The advantage of this approach is that it's a fast way to deinterlace and is good for low-motion scenes, such as interviews, but you will get ghosting every time an image moves.

### Weave

Another commonly used method is *weaving*, which shows both fields in each frame. This method basically doesn't do anything to the frame, leaving you with jagged edges but with the full resolution, which can be good.

## Area-Based Deinterlacing

*Area-based* deinterlacing blends nothing but the jagged edges. You do this by comparing frames over time or by space/position. It gives you good results in quiet scenes with little movement because in those circumstances there is nothing to blur.

## Motion Blur

The *motion blur* method blurs the jagged edges where needed, instead of mixing (that is, blending) them with the other field. This way, you get a more filmlike look. You may have to apply motion blur with a program such as Apple Final Cut Pro or Adobe After Effects before using a compression application.

## Discard

With the *discarding* method, you throw away every second line (leaving the movie at half the original height) and then resize the picture during playback. Because this is the same as skipping Field 2, Field 4, Field 6, and so on, you could also call this *even fields only* or *odd fields only*. Although you won't get artifacts from trying to blend or merge the images, you'll lose half the resolution, and motion will become less smooth.

## Bob

The *bob* approach displays every field (so you don't lose any information) one after the other (that is, without interlacing) but with double the frames per second. Thus, each interlaced frame is split into two frames (that is, the two former fields) at half the height. Sometimes bobbing is also called *progressive scanning*. However, since the bob approach doesn't analyze areas or the differences between fields, the two approaches are not really the same (see the next section).

## Progressive Scan

*Progressive scanning* analyzes the two fields and deinterlaces only the parts that need to be deinterlaced. The main difference between progressive scanning and area-based deinterlacing is that progressive scanning gives you a

movie with twice the frames per second instead of the standard 25 fps or 30 fps movie, thus leaving you with perfect fluidity of motion. To say it more academically, it has high temporal and vertical resolution.

This method is also variously called *motion adaptive*, *bob and weave*, and *intelligent motion adaptive*.

## Motion Compensation



**NOTE** If you control the video production process for your content and know you don't plan to distribute via a traditional broadcast, then try to keep your whole project in progressive mode to avoid issues with deinterlacing.

The *motion compensation* method analyzes the movement of objects in a scene when the scene consists of a lot of frames. In other words, it involves tracking each object that moves around in the scene, thus effectively analyzing a group of consecutive frames instead of just single frames.

Although effective for horizontal motion, some software for this technique does not handle vertical motion at all and may fall back on selective blending or other techniques when it is unable to resolve the motion vectors.

## Image Aspect Ratio Correction

The film world has dealt with the difficulty of matching the source aspect ratio to the screen ever since movies were invented. For the video world, this is somewhat of a new challenge. Much like Henry Ford's Model T that came in any color as long as it was black, if you worked with video, the aspect ratio was 4:3. This was a much simpler time because most of the video displays were also 4:3. We're now living in a multiscreen/multiformat world, and it's become a lot more complicated. Most of the time, it's advisable to keep the aspect ratio the same as the source and allow the playback system to account for any difference in aspect ratio. An example would be if your source file is a movie trailer that was delivered to you at  $2048 \times 858$ . If you divide the width by the height, you get the result of 2.3869, which is generally rounded up and written as a ratio to one like so: 2.39:1. This is one of the standard aspect ratios for cinema. Let's say you want to upload this to YouTube and have it look correct. You could reformat the source video to  $16 \times 9$  by adding a letterbox, which is simply black bars that take up the remaining space between the original 2.39:1 aspect ratio and the destination 1.78:1 player aspect ratio. The problem with this approach is that if the end user plays the YouTube video full-screen on a computer or tablet that doesn't have a  $16 \times 9$  display, the black bars are "burned in," and the player may not format the video optimally for that display.

The better choice is to resize the video to a standard/optimal width and keep the original aspect ratio. In the previous example, this would mean keeping the 2.39:1 aspect while resizing to 1920 wide. The end result would be a file that is 1920 × 804. The YouTube player will automatically letterbox the video with the appropriate amount of black for the final output display.

Some facilities will also ship finished content in the *anamorphic* format—that is, wide-screen content that is horizontally squished to fit in 4:3. During playback, specialized hardware is used to restore the wide-screen aspect ratio. When working with content that is either letterboxed or anamorphic, it's important to correct the image appropriately. In the case of letterboxing, this probably means cropping out the black bars, and in the case of anamorphic, it means changing the aspect ratio from 4:3 to 16:9 to correct the image.

Take care when dealing with aspect ratio conversions to ensure that you aren't losing content by accident or distorting the picture in a way not intended by the creator.

## Pixel Aspect Ratio Correction

An important element of scaling is pixel aspect ratio correction. What is pixel aspect ratio? You already know that *aspect ratio* just means the relationship between the height and width of the image. *Pixel aspect ratio* is the same concept, but instead of being applied to the image as a whole, it describes the actual shape of the pixels.

Three common pixel aspect ratios are in use today. The first is taller than wide; the second is the same height and width, which makes a perfect square; and the last is wider than tall. Let's take a closer look at where you'll most likely run into each of these now.

You'll find pixels that are taller than they are wide when working with legacy standard-definition content. These pixels are referred to as *nonsquare* pixels or 0.9 pixels. Pixels on the old CRT televisions had this shape.

Computer displays and modern HD televisions use pixels that are square. Square pixels are the most common pixel aspect ratio today and by far the easiest to work with.

A few codecs store the image in an anamorphic (squeezed) way but use a special pixel aspect ratio to stretch that squeezed image back out upon display. These codecs use pixels that are wider than tall to do this. Depending on if you're in a standard-definition environment or an HD environment, the codec will stretch by slightly different amounts: 1.21 for SD and 1.33 for HD.

Make sure you get the aspect ratio correction right when you convert to square-pixel sources from nonsquare-pixel sources by verifying that the output frame size matches the source aspect ratio. So, if you use a 4:3 source, a 4:3 frame size such as  $1440 \times 1080$ ,  $960 \times 720$ ,  $640 \times 480$ , and  $320 \times 240$  are all acceptable choices, even if the source frame size is  $720 \times 480$ ,  $720 \times 486$ ,  $640 \times 480$ , or  $352 \times 480$  (all nonsquare pixels).

## Audio Preprocessing

Audio is a huge part of any production and can make or break the viewing experience. There isn't typically a lot of preprocessing that needs to be done with well-produced audio, however. Raw captured audio may need some preprocessing (sometimes referred to as *sweetening*) to clean it up, but this is typically done during the edit process, rather than during preprocessing for compression. When it comes to audio, the job of the compressionist is to make sure that the program's audio is appropriate for the distribution channel.

## Adjusting Volume

Volume is one of the elements easily adjusted in preprocessing, and you can do this in a few different ways. The quickest way is to raise or lower the volume, either by an absolute decibel (dB) amount or by a relative percentage amount. This is another fairly coarse adjustment, akin to turning up or down the volume on your radio. For those needing a little more finesse, there are other possibilities, such as normalization and compression.

### Normalization

*Normalization* is the act of adjusting the audio levels in the content and then raising or lowering the volume of the entire clip so that the loudest sound

matches the level you have specified. This is a global adjustment, affecting the volume of the entire track the same way, rather than affecting the relative levels.

## Audio Compression

This is a totally different type of compression than what we have discussed so far in this book; with regard to audio, *compression* refers to a specific type of audio filter known as a *compressor*. Loud noises in a digital audio track can cause distortion, and likewise, quiet sounds, such as whispering, can be lost. An audio compressor can smooth out these issues in an audio track by removing dynamic range. By pulling down large spikes and lifting up those quiet parts, compression will ensure that the average loudness is fairly constant.

## Noise Reduction

Just as with video, there are also noise reduction filters for audio, although these are more often found in professional audio-editing tools, not in compression tools. Unwanted noise in audio tracks is just as bad for compression because noisy video (meaning bits) will be wasted, and the end result will be lower quality than desired. Some compression tools have simple hum-removal filters that will help clean up the audio during encoding, but truly bad audio may need to be preprocessed separately in a professional application such as Avid Pro Tools or Adobe Audition.

## Stereo Mixdowns of 5.1 Surround Mixes

Depending on the type of content you normally work with, you may be given source material that has a 5.1 surround sound track and asked to encode it for a destination that supports only stereo audio. Nine times out of ten, if you ask for a stereo mix, you'll be able to get it because a typical surround mix session produces both the 5.1 and stereo mixdown. It's almost always preferred to have an audio professional produce mixdowns, but in the rare case where you have only the 5.1 source and you're the only resource available, you'll need to know how to properly convert the six channels into two.



The most straightforward stereo mixdown is referred to as Lo/Ro, which stands for “left only and right only.” Sometimes referred to as an *ITU downmix*, the formula to create this two-channel mix from a 5.1 source is as follows:

1. Discard the LFE (subwoofer) channel. This channel is of little use to small speakers that will most likely be the destination and will overdrive them.
2. Leave the left and right channels alone, each continuing to be mapped to Ch1/Left and Ch2/Right, respectively.
3. Map/pan the center channel to both Ch1 and Ch2 equally while lowering its volume -3 dB.
4. Map the left surround channel to Ch1 while lowering its volume -3 dB.
5. Map the right surround channel to Ch2 while lowering its volume -3 dB.

If you are asked to deliver an Lt/Rt (which stands for “left total/right total”), this is a different type of downmix, often referred to as *matrix encoding*. It is much more complicated to create and is best left to an audio professional.

## Conclusion

Being aware of the preprocessing techniques covered in this chapter will no doubt make you a better compressionist. However, the application you use determines how well you will succeed with your content almost as much as the skill set you’re developing. Each application, through supporting multiple formats, typically has one or two specific workflows or techniques it does better than most, so knowing how to use several applications can be crucial to successful compression.

# Index

## Numbers

1:85:1 (flat), cinema aspect ratios, 221  
 2K, cinema resolution, 221  
 2.39:1  
     cinema aspect ratios, 221  
     image aspect ratios, 92–93  
 2.6 gamma, DCP, 223–224  
 4:3 aspect ratio  
     common aspect ratios, 24  
     correcting image aspect ratios, 92–93  
     letterboxing wide-screen formats for, 7–8  
     overview of, 7  
 4K  
     cinema resolution, 221  
     encoding for YouTube, 137  
 5.1 surround mixes  
     preprocessing audio, 95–96  
     working with in Compressor, 121  
 16:9 aspect ratio standard, 8, 24  
 23.976 fps  
     converting to 24 fps, 234–235  
     preparing 24 fps QuickTime files, 245–246  
 24 fps  
     converting to 23.976 to, 234–235  
     DCP frame rate, 223  
     preparing QuickTime files, 245–246  
     syncing QuickTime audio, 230–231  
 30 fps, DCP frame rate, 223  
 35mm film, DCP compared with, 214  
 360-degree video  
     codecs, 269  
     device-specific, 270  
     encoding with VREncoder, 271  
     monoscopic vs. stereoscopic, 268  
     optimizing, 270–271  
     overview of, 267  
     projection options, 268–269  
     resolution, 268  
     streaming vs. downloading, 270  
 480 pixels, guidelines for video aspect ratios, 82  
 720 pixels, guidelines for video aspect ratios, 82

1080 pixels  
     encoding for YouTube, 137  
     guidelines for video aspect ratios, 81  
 2160 pixels, guidelines for video aspect ratios, 81

## A

AAC  
     audio settings using HandBrake, 105  
     audio wrappers/containers, 25  
 ABR (adaptive bitrate streaming)  
     encoding uploaded content, 156–157  
     monitoring progress of encoding job, 157  
     for online video delivery, 18  
     output to HLS, 117–120  
     overview of, 151–152  
     playing content, 159  
     publishing content, 158  
     starting streaming endpoint, 152–153  
     uploading source video content, 154–156  
 ACES color system, 256  
 Acquisition codes, codecs, 28  
 Action-safe area, cropping and, 79  
 Adaptive bitrate streaming. *See* ABR (adaptive bitrate streaming)  
 Adobe After Effects  
     creating intermediate image sequence for DCP, 230–233  
     noise reduction before compression, 89  
 Adobe Creative Cloud suite, 111, 196  
 Adobe Encore  
     bit budgeting, 194  
     Blu-ray support, 191  
     compatibility with Blu-ray burners, 197  
     installing, 196  
     producing Blu-ray Discs, 200–203  
     producing SD DVDs, 203  
 Adobe Flash Player, 41  
 Adobe Media Encoder CC (AME)  
     advantages/disadvantages, 114–115  
     encoding equipment, 54

- Adobe Media Encoder CC (*continued*)
    - how it works, 112–114
    - installing, 196
    - overview of, 111
    - producing H.264 Blu-ray compatible file
      - using, 198–199
    - user interface, 111–112
  - Adobe Photoshop, 89
  - Adobe Premiere Pro
    - bit budgeting, 194
    - encoding for YouTube, 143–145
    - as NLE system, 104
    - preparing 24 fps QuickTime files, 245–246
    - producing Blu-Ray Discs, 197–198
  - Advanced Video Coding (AVC), 31
  - After Effects. *See* Adobe After Effects
  - AIFF, audio wrappers/containers, 25
  - AJA DataCalc, video storage and data rate calculators, 61
  - AME. *See* Adobe Media Encoder CC (AME)
  - Analysis, video compression, 64–65
  - Andrew Hedges Aspect Ratio Converter, 64
  - Apple
    - Compressor. *See* Compressor (Apple)
    - Digital Color Meter, 86
    - macOS. *See* macOS
    - ProRes format. *See* ProRes (Apple)
    - QuickTime. *See* QuickTime (Apple/macOS)
  - Applications, IMF (Interoperable Master Format), 255–256
  - AR (augmented reality), 262
  - Archiving, preparing for video compression, 72
  - Area-based deinterlacing, 91
  - Aspect ratio
    - calculators, 63–64
    - compression parameters, 24
    - cropping and, 78
    - DCP requirements, 221–222
    - defined, 7
    - image aspect ratio correction, 92–93
    - pixel aspect ratio correction, 93–94
    - as video element, 7–8
    - YouTube requirements, 139–140
  - Aspect Ratio Calculator, 64
  - ASSETMAP file
    - DCP files, 214
    - IMF files, 257
  - Assets, DCP
    - assigning UUID, 220
    - encoding picture assets, 220
    - wrapping into a package, 228–230, 247–252
  - Audio
    - audio track files in IMF, 256
    - Azure Media Services codecs supported, 156
    - choosing audio format when creating Blu-ray Disc, 200
    - containers (wrappers/formats) for, 25
    - create final audio tracks for DCP, 234–236
    - main sound essence in DCP, 215
    - online video delivery and, 16–17
    - preparing for DCP, 224–225
    - preprocessing. *See* Preprocessing audio
    - SD codecs, 188
    - YouTube bitrate requirements, 141
  - Audio compression
    - overview of, 15–16
    - preprocessing audio, 95
  - Audio pull-up, 225, 235
  - Augmented reality (AR), 262
  - Aurora (Tektronix), analysis and quality control tools, 65
  - Authoring formats, 65
  - AVC (Advanced Video Coding), 31
  - AVI
    - compressionist on, 75
    - delivery codes, 31
    - media wrappers/containers, 25
  - Avid DNX, mezzanine codecs, 29
  - Avid ProTools, create final audio tracks for DCP, 234–236
  - AWS Elemental Server, 124–126
  - Azure Media Services
    - delivering ABR content, 152
    - encoding uploaded content, 156–157
    - monitoring progress of encoding job, 157
    - playing content, 159
    - publishing content, 158
    - starting streaming endpoint, 152–153
    - uploading source video content, 154–156
- B**
- b (bits), 20
  - B (bytes), 20
  - B-frames (bidirectional frames), 22–23

- Backups, hardware requirements for video compression, 58
- Bandwidth, challenges in live streaming, 170
- Batch encoding, 63, 105
- BD. *See* Blu-ray Discs (BD)
- BDAV (Blu-ray Disc Audio/Visual), 192–193
- BDMV (Blu-ray Disc Movie), 192–193
- Bidirectional frames (B-frames), 22–23
- Bit budgeting, DVDs and Blu-ray, 194–196
- Bit (data) rates
  - calculators, 64
  - compression parameters, 20–21
- Bitdash, HTML5 players, 40
- Bitrate. *See also* ABR (adaptive bitrate streaming)
  - CBR (constant bitrate), 21
  - DCP, 224
  - YouTube requirements, 140–141
- Bits (b), 20
- BlackMagic Design Teranex, 84
- Blackmagic Resolve Studio, 253
- Blanking interval, 6
- Blend, deinterlacing video, 90
- Blu-ray Disc Audio/Visual (BDAV), 192–193
- Blu-ray Disc Movie (BDMV), 192–193
- Blu-ray Discs (BD)
  - bit budgeting, 194–196
  - disc rippers, 61–62
  - encoding as an editor, 50
  - exporting MP4 from, 104
  - features, 189–192
  - overview of, 191–192
  - producing, 196–197
  - producing H.264 Blu-ray compatible file, 198–199
  - producing in Encore, 200–203
  - recording, 193
  - software, 192
  - working with in Premiere Pro, 197–198
- Bluetooth, game controllers, 265
- Bob and weave, deinterlacing video, 92
- Bob, deinterlacing video, 91
- Brightcove
  - transcoding several video at once using Zencoder, 148–151
  - video streaming host, 167
- Brightness, color/image preprocessing, 87
- Bytes (B), 20

## C

- Camera, for live streaming, 166
- Canon XF100 camera, 166
- Captions, creating for DCP, 226–228
- Capture device, for live streaming, 166–167
- Carbon Coder (ProMedia), analysis and quality control tools, 65
- Cardboard (Google), VR handheld kits, 264
- Cathode-ray tubes. *See* CRT (cathode-ray tube) monitors
- Cb (chroma with blue minus luma), in Y'CbCr color space, 85
- CBR (constant bitrate), encoding options, 21
- CCAP (closed captions), 226
- CDNs (content distribution networks), 169
- Channels, color space, 85
- Chroma
  - color/image preprocessing, 87
  - in Y'CbCr color space, 85
- CineForm (GoPro), as mezzanine format, 29, 71
- Cinema Mezzanine, IMF, 256
- Cisco Visual Networking Index, 134
- CLI (command-line interface), HandBrake, 106
- Clock speeds, CPUs, 59
- Closed captions (CCAP), 226
- Cloud
  - cloud-based transcoding, 167–168
  - compression tools, 126–127
  - encoding on cloud vs. on-premises, 54
- Codecs
  - 360-degree video, 269
  - acquisition codes, 28
  - advantages/disadvantages of various types, 26–27
  - Azure Media Services supported, 155–156
  - compression process and, 9
  - DCP, 224
  - delivery codes, 30–33
  - lossy and lossless video, 11
  - mezzanine codecs, 29
  - overview of, 26
  - pixel aspect ratio correction, 94
  - SD codecs, 187–188
  - spatial and wavelet-based, 11–12
  - YouTube requirements, 141, 143
- Color/image preprocessing
  - brightness and contrast, 87
  - chroma, 87

- Color/image preprocessing (*continued*)
  - gamma, 87
  - hue, 88
  - luma, 85–86
  - noise reduction, 88–89
  - overview of, 85
  - saturation, 88
- Color meters, 86
- Color space
  - DCP, 223–224
  - models, 85
- Combing, deinterlacing video and, 90
- Command-line interface (CLI), HandBrake, 106
- compatibility, optical media, 186, 197–199
- Components, color space, 85
- Composition playlist. *See* CPL (composition playlist)
- Compression tools
  - Adobe Media Encoder. *See* Adobe Media Encoder CC (AME)
  - Apple Compressor. *See* Compressor (Apple)
  - AWS Elemental Server, 124–126
  - cloud-based systems, 126–127
  - enterprise-grade tools, 122–123
  - free/low cost, 102
  - HandBrake. *See* HandBrake
  - hybrid cloud systems, 127–128
  - list of low cost tools, 110
  - list of mid-level tools, 122
  - middle-of-the-road tools, 111
  - MPEG Streamclip. *See* MPEG Streamclip
  - overview of, 101–102
  - software requirements for video
    - compression, 62–63
  - Telestream Vantage, 123–124
- Compressionists
  - interview with Aaron Owen, 207–210
  - interview with Alex Zambelli, 129–131
  - interview with Bryce Castillo, 97–99
  - interview with Derrick Freeman, 73–75
  - interview with Doug Daulton, 161–163
  - workflow for, 52
- Compressor (Apple)
  - advantages/disadvantages, 121–122
  - encoding equipment, 54
  - how it works, 117–121
  - improvements in, 205
  - overview of, 115
    - producing SD DVDs, 203–206
    - user interface, 116–117
- Compressors, audio filtering with, 95
- Computer processing unit (CPU)
  - clock speeds, 59
  - hardware requirements for video
    - compression, 55
- Computers, luma adjustments, 86
- Consumer Technology Association (CTA), 136
- Containers (wrappers/formats)
  - for audio, 25
  - DCP, 221–222
  - overview of, 25
  - for various media types, 25–26
- Content distribution networks (CDNs), 169
- Content, evaluating compression needs
  - how it is received, 45–46
  - how much is to be compressed, 46–47
  - long-form vs. short form, 44–45
- Contrast, color/image preprocessing, 87
- Conversion, in temporal preprocessing
  - difficulties, 84
  - frame rate, 82–83
- Costs, challenges in live streaming, 170
- CPL (composition playlist)
  - DCP files, 214
  - IMF files, 256
  - overview of, 218
  - version files, 219–220
- CPU (computer processing unit)
  - clock speeds, 59
  - hardware requirements for video
    - compression, 55
- Cr (chroma with red minus luma), in Y'CbCr
  - color space, 85
- Creative Cloud suite (Adobe), 111, 196
- Cropping, spatial/geometric preprocessing, 78–80
- CRT (cathode-ray tube) monitors
  - gamma and, 87
  - interlaced scan video used for, 3–4
  - interlaced video and, 90
  - modern video technologies vs., 4–5
  - nonsquare pixels, 93
- CRU Dataport DX115 drive sled, for delivery of
  - DCPs, 228
- CTA (Consumer Technology Association), 136
- Cube map, 360 video projection, 268–269
- CyberLink PowerProducer, 193

**D**

- Data (bit) rates
  - calculators, 64
  - compression parameters, 20–21
- DataCalc (AJA), video storage and data rate
  - calculators, 61
- dBs (decibels), preprocessing audio, 94
- DCI (Digital Cinema Initiative), 211–212, 224–225
- DCP (Digital Cinema Package)
  - aspect ratio and container size, 221–222
  - codecs, 224
  - color space and gamma, 223–224
  - CPL (composition playlist), 218
  - creating subtitles and captions, 226–228
  - encoding picture assets, 220
  - encryption, 218
  - files in, 214–215
  - frame rate, 223
  - history of DCP formats, 211–213
  - Interoperable Master Format compared
    - with, 255
  - naming conventions, 216–217
  - overview of, 213–214
  - preparing audio, 224–225
  - steps in creating, 220
  - tools for use with, 253
  - version files, 219–220
  - wrapping assets into a package, 228–230
- DCP-O-Matic, 253
- DCP recipe 1
  - checking source files, 237–239
  - create final audio tracks using Avid ProTools, 234–236
  - create intermediate image sequence using
    - After Effects, 230–233
  - creating delivery drive, 243–244
  - encoding intermediate TIFF files to JPEG
    - 2000, 236–237
  - generating package, 240–241
  - playing, 242
- DCP recipe 2
  - creating delivery drive, 252
  - preparing 24 fps QuickTime files, 245–246
  - wrapping assets using easyDCP Publisher, 247–252
- DCP Transfer, creating delivery drive, 243–244
- DCT (spatial) video compression, 11–13
- Decibels (dBs), preprocessing audio, 94
- Deinterlacing video, 90–92
- Delivery
  - creating delivery drive, 243–244, 252
  - delivery codes, 30–33
  - delivery date in evaluating compression
    - needs, 45
  - online video, 16–17
- Desktop players, 33–34
- Desktop, staying organized, 67
- Digital Cinema and Broadcasting
  - DCP. *See* DCP (Digital Cinema Package)
  - IMF. *See* IMF (Interoperable Master Format)
- Digital Cinema Initiative (DCI), 211–212, 224–225
- Digital Cinema Package. *See* DCP (Digital Cinema Package)
- Digital Color Meter (Apple), 86
- Digital Source Master (DSM), QuickTime, 230–231
- Disc rippers, 61–62
- Discard, deinterlacing video, 91
- DNX (Avid), mezzanine codecs, 29
- DOF (degrees of freedom), VR hardware, 264
- Dolby Digital, 188
- Downloading, 360-degree video, 270
- DSM (Digital Source Master), QuickTime, 230–231
- Duration, YouTube requirements, 139
- DVD Architect (Vegas), 203
- DVD-R, 185–186
- DVD-RW, 185–186
- DVD Studio Pro, 203
- DVD+R, 185–186
- DVDs
  - authoring, 50
  - bit budgeting, 194–196
  - codecs, 187–188
  - compatibility risks, 186
  - demise of HD DVDs, 190
  - disc rippers, 61–62
  - disc types, 185–187
  - overview of, 184
  - producing SD DVDs, 203–206
  - software, 192
  - .vob extensions, 188–189
- DVDxDV, disc rippers, 61
- Dynamic packaging, Media Services, 152

**E**

## easyDCP

- checking DCP source files, 237–239
- easyDCP Resolve Plug-in, 253
- encoding intermediate TIFF files to JPEG 2000, 236–237
- generating package, 240–241
- playing DCP, 242
- wrapping assets using easyDCP Publisher, 247–252

## Editors, workflow for, 49–51

## El Gato, capture device for live streaming, 167

## EME (Encrypted Media Extensions), HTML5 players, 38

## Encoders

- Adobe Media Encoder. *See* Adobe Media Encoder CC (AME)
- equipment required for live streaming, 167
- for streaming video, 167
- VRencoder, 271
- Zencoder (Brightcove), 148–151

## Encoding

- analysis and quality control, 64–65
- aspect ratio calculators, 63–64
- audio. *See* Audio compression
- automating, 52–53
- automating on enterprise systems, 53–54
- on cloud vs. on-premises, 54
- compression tools, 62–63
- disc rippers, 61–62
- hardware requirements, 55–56
- input/output options, 59–60
- players, 60
- software requirements, 60
- storage options, 56–58
- tools. *See* Compression tools
- video. *See* Video compression
- video storage and data rate calculators, 64
- workflow for compressionists, 52
- workflow for editors, 49–51

Encore. *See* Adobe Encore

## Encrypted Media Extensions (EME), HTML5 players, 38

## Encryption/decryption, DCP, 218

## Enterprise systems, automating encoding process, 53–54

## Equirectangular, 360 video projection, 268–269

## ER (extended reality), 263

**F**

## Facebook

- OBS support, 173
- Oculus Go, 265–266
- social media and live streaming, 170
- streaming to Facebook Live via OBS, 173–177
- uses of online video, 133
- video streaming hosts, 167

## Facebook Live

- live streaming platforms, 171
- streaming guidelines, 176
- streaming to via OBS, 173–177

## Farnsworth, Philo T., 2

## FFMPEG

- encoding equipment, 54
- low cost compression tools built on, 110
- VRencoder built on, 271

## Fields, as video elements, 3–4

## File extensions, naming conventions and, 68

## Files

- Azure Media Services supported file formats, 155–156
- checking DCP source files, 237–239
- DCP, 214–215
- encoding intermediate TIFF files to JPEG 2000, 236–237
- file size and codecs, 27
- IMF files, 257–258
- naming conventions, 68–69
- preparing 24 fps QuickTime files, 245–246
- supplemental IMPs, 258–259
- YouTube file size requirements, 139

## Filters, brightness and contrast, 87

## Final Cut Pro X, 121

## Final delivery, evaluating compression needs, 44

## Flash Player (Adobe), 41

## Flash Video, HTML5 players, 39

## Flash, web browser plug-ins for video playback, 146

## Flat (1:85:1), cinema aspect ratios, 221

## FLV, media wrappers/containers, 25

## Folders, staying organized, 67–68

Formats. *See also* Containers (wrappers/formats)

- adaptive bitrate streaming, 18
- audio, 25, 200
- Azure Media Services supported file formats, 155–156
- DCP, 211–213, 221–222, 229

- media types, 25–26
  - mezzanine format, 71
  - production vs. authoring, 25–26
  - wide-screen, 7–8
  - YouTube supported video formats, 139
  - FOV (field of view), VR headsets, 262
  - fps (frames per second). *See also* Frame rate, 3
  - Frame frequency. *See* Frame rate
  - Frame rate
    - compression parameters, 21
    - conversion in temporal preprocessing, 82–83
    - converting to 23/976 to 24 fps, 234–235, 245–246
    - DCP, 223
    - defined, 3
    - syncing QuickTime audio, 230–231
    - YouTube requirements, 140
  - Frames
    - common sizes, 82
    - compression parameters, 22–23
    - compression process, 9–10
    - DCI-complaint sizes, 221
    - interframe vs. intraframe video
      - compression, 13–14
    - as video element, 3–4
    - working with HandBrake, 105
  - Frames per second (fps). *See also* Frame rate, 3
- G**
- Gamma
    - color/image preprocessing, 87
    - DCP, 223–224
  - Gear VR (Samsung), 265
  - Geometric preprocessing. *See* Spatial/geometric preprocessing
  - Google Cardboard, VR handheld kits, 264
  - GoPro CineForm, as mezzanine format, 29, 71
  - GPU (graphics processing unit), 55
  - Grid encoding, 53
- H**
- H.264
    - delivery codes, 31
    - encoding for YouTube, 137
    - optimizing 360 video, 270–271
    - output options in HandBrake, 106
    - preparing 24 fps QuickTime files, 246
    - producing H.264 Blu-ray compatible file
      - using AME, 198–199
    - VR codecs, 269
    - YouTube codecs, 141
  - H.265
    - delivery codes, 30
    - optimizing 360 video, 271
    - output options in HandBrake, 106
  - HandBrake
    - advantages/disadvantages, 105–106
    - disc rippers, 61
    - how it works, 104–105
    - overview of, 102
    - user interface, 103–104
  - Hardware
    - input/output options, 59–60
    - preparing for video compression, 55–56
    - storage options, 56–58
    - VR (virtual reality), 264
  - HD
    - demise of HD DVDs, 190
    - YouTube supported resolutions, 139
  - HDMI, live streaming and, 166
  - HDMV (Movie Mode), Blu-ray and, 191
  - HDR (high dynamic range) video
    - IMF support, 255
    - overview of, 14–15
    - YouTube bitrate requirements, 141
  - HDTV
    - 16:9 aspect ratio standard for, 8
    - interlaced and progressive scanning in, 3
    - square pixels, 93
  - Headsets, VR
    - handheld kits, 264–265
    - overview of, 262
    - tethered, 265–266
  - Hertz (Hz). *See* Frame rate
  - HEVC
    - delivery codes, 30
    - optimizing 360 video, 270
    - VR delivery codecs, 269
  - High dynamic range video. *See* HDR (high dynamic range) video
  - HLS (HTTP Live Streaming), 18, 117–120
  - HTML5 video
    - encoding for, 146
    - markup examples, 146–148



HTML5 video (*continued*)  
 players, 38–40  
 transcoding several video at once using  
   Zencoder, 148–151  
 HTTP Live Streaming (HLS), 18, 117–120  
 Hue, color/image preprocessing, 88  
 Hybrid cloud systems, compression tools,  
   127–128  
 Hz (Hertz). *See* Frame rate

## I

I-frames. *See* Intra frames  
 I/O (input/output), hardware requirements for  
   video compression, 59–60  
 Images  
   aspect ratio correction, 92–93  
   create intermediate image sequence for DCP,  
     230–233  
   encoding DCP picture assets, 220  
   image quality comparing codecs, 27  
   image track files in IMF, 256  
   preprocessing, 85–89  
 IMF (Interoperable Master Format)  
   anatomy of, 256–258  
   applications, 255–256  
   comparing with DCP, 255  
   overview of, 254  
   quality control, 259  
   SMPTE in creation of, 212  
   supplemental IMPs, 258–259  
   what it is not, 254  
 Immersion, characteristics of VR, 261  
 Immersive video. *See* 360-degree video  
 IMP (Interoperable Master Package)  
   anatomy of IMF, 256–257  
   quality control, 259  
   supplemental IMPs, 258–259  
 Input/output (I/O), hardware requirements for  
   video compression, 59–60  
 Intelligent motion adaptive, deinterlacing  
   video, 92  
 Inter-Society Digital Cinema Forum.  
   *See* ISDCF (Inter-Society Digital  
   Cinema Forum)  
 Interframe video compression, vs.  
   intraframe, 13–14

Interlaced video  
   deinterlacing video, 90  
   HDTV and, 3  
   overview of, 3–4  
   progressive scanning vs., 4  
 Internet  
   online video delivery, 16–17  
   uses of online video, 134  
 InterOp (IOP) format, DCP, 213  
 Interoperable Master Format. *See* IMF  
   (Interoperable Master Format)  
 Interoperable Master Package. *See* IMP  
   (Interoperable Master Package)  
 Interpolate, creating pixels in scaling process, 80  
 Interviews with compressionists  
   Aaron Owen, 207–210  
   Alex Zambelli, 129–131  
   Bryce Castillo, 97–99  
   Derrick Freeman, 73–75  
   Doug Daulton, 161–163  
 Intra frames  
   characteristics of, 22  
   intraframe video compression vs.  
     interframe, 13–14  
 Inverse telecine, temporal preprocessing, 84  
 IOP (InterOp) format, DCP, 213  
 iOS (Macs). *See also* macOS, 61  
 ISDCF (Inter-Society Digital Cinema Forum)  
   compliance, 243  
   DCP audio specs, 224–225  
   Digital cinema naming convention, 216–217  
 ITU downmix, preprocessing audio, 96  
 iTunes (macOS), 35

## J

JPEG 2000  
   codecs for DCP, 212  
   DCP spec, 224  
   encoding intermediate TIFF files to, 236–237  
   encoding picture assets into JPEG 2000  
     frames, 220–221  
   file types in DCP and, 212  
   IMF spec, 255  
   wavelet video codec, 12  
 JW Player, HTML5 players, 41

**K**

Kbps (kilobits per second), data rates, 20  
 KDM (key delivery message), encryption of  
     DCP, 218  
 Key frames. *See* Intra frames  
 Kilobits per second (Kbps), data rates, 20  
 Knowledge/training, 48

**L**

Labeling, DCP naming conventions, 226  
 Latency, challenges in live streaming, 170  
 LCD monitors, 87  
 LED (Light-emitting diode) screens, 5  
 Legacy codecs  
     acquisition codes, 28  
     delivery codes, 30, 32–33  
     mezzanine codecs, 29  
 Legacy video issues, 89  
 Letterboxing  
     aspect ratios and, 24  
     wide-screen formats for standard TVs  
         using, 7–8  
 Library management system (LMS), 228  
 Light-emitting diode (LED) screens, 5  
 Linux OS  
     HandBrake support, 106  
     making delivery drives, 243  
     playback servers running on, 229  
     third-party players, 36  
 Live linear, vs. live streaming, 169  
 Live streaming. *See* Streaming video  
 LMS (library management system), 228  
 Lookup table (LUT), color conversion  
     and, 224  
 Lossless compression  
     audio, 16  
     video, 11–12  
 Lossy compression  
     audio, 16  
     video, 11–12  
 Luma (black and white or lightness)  
     gamma adjustments, 87  
     range expansion, 85–86  
     in Y'CbCr color space, 85  
 LUT (lookup table), color conversion and, 224

**M**

M4A, audio wrappers/containers, 25  
 M4V, converting to ProRes, 109  
 macOS  
     AIFF container for audio, 25  
     AME support, 111  
     backup systems, 58  
     built-in players, 35–36  
     Compressor support, 115  
     disc rippers, 61  
     gamma and, 87  
     HandBrake support, 106  
     MPEG-4 support, 31  
     Switch support, 38  
     video storage and data rate calculators, 64  
 Macroblock grids, compression process and, 9  
 Main picture essence, file types in DCP, 215  
 Main sound essence, file types in DCP, 215  
 Markup, HTML5 examples, 146–148  
 Matrix encoding, preprocessing audio, 96  
 Matroska (MKV), media wrappers/containers, 25  
 Mbps (megabits per second), data rates, 20  
 MBR partition table, DCP, 229  
 Media Services. *See* Azure Media Services  
 Media Source Extensions (MSE), HTML5  
     players, 38  
 Media types, containers, 25–26  
 Megabits per second (Mbps), data rates, 20  
 Metadata, 227, 257  
 Mezzanine codecs, 29  
 Mezzanine (in-between) files, 71  
 Microsoft  
     AVI standard, 25  
     cloud and, 54, 127  
     encoding for video-on-demand delivery, 152  
     HoloLens, 263  
     legacy delivery codes, 32–33  
     Mixer, 172  
     Movies & TV app, 34–35  
     third-party players, 36  
     VR headsets, 266  
     Windows Media Services, 17  
 Mixed reality (MR), 263  
 Mixer, live streaming platforms, 172  
 Monitoring, progress of encoding job, 157  
 Monoscopic, 360-degree video, 268  
 Motion adaptive, deinterlacing video, 92  
 Motion blur, deinterlacing video, 91

Motion compensation, deinterlacing video, 92  
 MOV, media wrappers/containers, 25  
 Movies & TV (Microsoft), 34–35  
 MOX, media wrappers/containers, 25  
 MP3 (MPEG-1 Layer 3), 25  
 MP4  
     exporting from Blu-ray Disc, 104  
     media wrappers/containers, 25  
     output file options in HandBrake, 106  
 MPEG-1, 188  
 MPEG-1 Layer 3 (MP3), 25  
 MPEG-2  
     as mezzanine format, 71  
     SD codecs, 187  
     storing video data, 61  
     .vob extensions and, 188–189  
 MPEG-2 transport stream (MPEG-TS), 25  
 MPEG-4, 255  
 MPEG-4 Part 10, 31  
 MPEG-DASH, 18  
 MPEG Streamclip  
     advantages/disadvantages, 109–110  
     how it works, 108  
     overview of, 107  
     user interface, 107–108  
 MPEG-TS (MPEG-2 transport stream), 25  
 MR (mixed reality), 263  
 MSE (Media Source Extensions), HTML5  
     players, 38  
 Music, audio compression and, 16  
 MXF files  
     DCP files, 215  
     encoding picture assets into JPEG 2000  
         frames, 220–221  
     IMF files, 256  
     media wrappers/containers, 25

## N

Naming conventions, DCP, 216–217  
 NAS (network-attached storage), 56–57  
 National Television Standards Committee. *See*  
     NTSC (National Television Standards  
     Committee)  
 Network-attached storage (NAS), 56–57  
 NLE (nonlinear editing)  
     encoding as an editor, 49  
     encoding equipment, 54

    frame rate conversions, 83  
     Premier Pro for, 104  
 Noise reduction  
     combining with luma adjustments, 86  
     preprocessing audio, 95  
     preprocessing images, 88–89  
 Nonsquare pixels, 24, 93  
 Normalization, volume adjustments, 94–95  
 NTSC (National Television Standards Committee)  
     choosing television standard when producing  
         Blu-ray Disc, 201  
     frame rate standard, 3  
     transferring motion-picture film into  
         electronic form, 83  
     vertical-line resolution, 6

## O

OBS (Open Broadcaster Software)  
     guide to, 176  
     setting up, 175–177  
     streaming to Facebook Live via OBS, 173  
 OCAP (open captions), 226  
 Oculus Rift, tethered headsets, 265  
 OGM (Ogg Media), 25  
 OLED (Organic light-emitting diode) screens, 5  
 On-premise enterprise-grade tools  
     AWS Elemental Server, 124–126  
     Telestream Vantage, 123–124  
 Online video delivery. *See also* VOD (video-on-  
     demand), 16–18  
 Online video platforms (OVP), 135–136, 167  
 Ooyala, video streaming hosts, 167  
 Open Broadcaster Software. *See* OBS (Open  
     Broadcaster Software)  
 OpenDCP, 253  
 Operating systems (OSs), choosing. *See also* by  
     individual types, 60  
 OPL (output profile list), IMF files, 258  
 Optical media  
     bit budgeting, 194–196  
     Blu-ray Discs, 189–192  
     compatibility risks, 186  
     DVD discs, 185–187  
     overview of, 183–184  
     producing Blu-ray Disc in Encore, 200–203  
     producing Blu-ray Discs, 196–197  
     producing H.264 Blu-ray compatible file  
         using AME, 198–199

- producing SD DVDs, 203–206
- recording Blu-ray Discs, 193
- SD codecs, 187–188
- .vob extensions, 188–189
- working with in Premiere Pro, 197–198
- Organic light-emitting diode (OLED) screens, 5
- OSs (operating systems), choosing. *See also* by individual types, 60
- Output profile list (OPL), IMF files, 258
- Output, to HTTP Live Streaming (HLS), 117–120
- Overscan region, cropping and, 79–80
- Overscanning, reasons for, 78
- OVP (online video platforms), 135–136, 167

## **P**

- P-frames (predicted frames), 22–23
- Packages. *See also* DCP (Digital Cinema Package); IMF (Interoperable Master Format)
  - generating, 240–241
  - history of DCP formats, 211–213
  - wrapping DCP assets into, 228–230
- PAL (Phase-Alternating Line)
  - frame rate standard, 3
  - PAL video, 83
  - vertical-line resolution for, 6
- Pan and scan process, 7
- Partitions, DCP, 229
- PCM
  - choosing audio format when creating Blu-ray Disc, 200
  - DCP audio specs, 224
  - file types in DCP and, 215
  - SD audio codecs, 188
- PCs, disc rippers, 61
- Periscope, live streaming platforms, 171
- Persistence of vision, 3
- Phase-Alternating Line. *See* PAL (Phase-Alternating Line)
- Pillar bars, letterboxing, 24
- Pixels
  - aspect ratio correction, 93–94
  - compression parameters, 22
  - scaling and, 80–82
  - YouTube requirements, 139–140
- PKL (packing list)
  - DCP files, 215
  - IMF files, 257

- Platforms, for live streaming, 171–172

## **Players**

- built-in players, 34–36
- desktop players, 33–34
- HTML5 players, 38–40
- overview of, 33
- software requirements for video
  - compression, 60
- third-party players, 36–38

## **Playing**

- content for video-on-demand delivery, 159
- DCP package, 242

- PlayStation, 266

- Policies, storage policy, 68

- PowerProducer (CyberLink), 193

- Predicted frames (P-frames), 22–23

- Premiere Pro. *See* Adobe Premiere Pro

## **Preparation, for video compression**

- analysis and quality control, 64–65
- archiving and transcoding, 72
- aspect ratio calculators, 63–64
- automating encoding process, 52–53
- automation on enterprise systems, 53–54
- compression tools, 62–63
- determining compression needs, 44–48
- disc rippers, 61–62
- essential equipment, 54
- experimentation, 70
- filename conventions, 68–69
- hardware requirements, 55–56
- input/output options, 59–60
- mezzanine (in-between) files, 71
- minimizing quality loss, 66–67
- overview of, 43
- players, 60
- productivity tips, 66
- software requirements, 60
- staying organized, 67–68
- storage options, 56–58
- video storage and data rate calculators, 64
- workflow for compressionist, 52
- workflow for editors, 49–51

- Preprocessing, 72, 77–78

## **Preprocessing audio**

- noise reduction, 95
- overview of, 94
- stereo mixdowns of 5.1 surround mixes, 95–96
- volume adjustments, 94–95

Preprocessing video

- brightness and contrast, 87
- chroma, 87
- color and images, 85
- conversion difficulties, 84
- cropping, 78–80
- deinterlacing video, 90–92
- frame rate conversions, 82–83
- gamma, 87
- hue, 88
- image aspect ratio correction, 92–93
- legacy video issues, 89
- luma range expansion, 85–86
- noise reduction, 88–89
- pixel aspect ratio correction, 93–94
- saturation, 88
- scaling, 80–82
- spatial/geometric, 78
- telecine/inverse telecine, 83–84
- temporal, 82

Processors

- comparing codecs, 27
- hardware requirements for video compression, 55

Production formats, 65

Productivity tips, 66

Progressive downloads, 16–17

Progressive scan

- deinterlacing video, 91–92
- HDTV and, 3
- overview of, 4–5
- YouTube requirements, 140

Projection, 360-degree video, 268–269

Projekktor, HTML5 players, 40

ProMedia Carbon Coder, analysis and quality control tools, 65

Proportionate scale, 80

ProRes (Apple)

- converting M4V files to ProRes QuickTime file, 109
- IMF support, 255
- intraframe video compression for, 13
- mezzanine codecs, 29
- as mezzanine format, 71
- preparing 24 fps QuickTime files, 246
- YouTube codecs, 141

Prosumer, 50

ProTools (Avid), 234–236

Publishing video

- for business use, 135–136
- content for video-on-demand delivery, 158

Pull, vs. push in live streaming, 168

Pulldown, telecine process, 83

Purple Pill (VR Studio), 271

Push, vs. pull in live streaming, 168

Pyramid, 360 video projection, 268–269

## Q

Quality

- timeliness weighed against, 48
- video compression and, 47

Quality control (QC)

- DCP and, 242
- IMP and, 259
- minimizing quality loss, 66–67
- player selection and, 60
- software requirements for video compression, 64–65

Quantization, video compression and, 13

QuickTime (Apple/macOS)

- built-in players, 35–36
- converting M4V files to ProRes, 109
- DSM (Digital Source Master), 230–231
- encoding as an editor, 50
- encoding equipment, 54
- preparing 24 fps files, 245–246
- web browser plug-ins for video playback, 146

## R

RAID (Redundant Array of Independent Devices)

- clock speeds, 59
- hardware requirements for video compression, 56–57

RAM

- hardware requirements for video compression, 55
- input/output options, 59

RealPlayer, web browser plug-ins for video playback, 146

REDCODE, acquisition codes, 28

Redundant Array of Independent Devices (RAID)

- clock speeds, 59
- hardware requirements for video compression, 56–57

Reels, in DCP, 215

Resolution

- 360-degree video, 268, 270–271
- cinema, 221
- modern video, 6
- video quality and, 5–6
- YouTube requirements, 139–140

Resolve Studio (Blackmagic), 253

Restream.IO, mirroring live stream to multiple sites, 181

RGB (red, green, blue)

- color space/color model, 85
- luma range expansion, 85–86
- saturation range, 88

Rippers, disc rippers, 61–62

Rotation movement, VR degrees of freedom, 264

## **S**

Samsung Gear VR, 265

SAN (storage area network), 56–57

Saturation, color/image preprocessing, 88

Scaling, spatial/geometric preprocessing, 80–82

Scope (2:39:1)

- cinema aspect ratios, 221
- wide-screen, 7–8

Screen Aspect Ratio & Dimension Calculator, 64

SD (standard definition)

- codecs, 187–188
- producing SD DVDs, 184, 203–206
- recording BDMV-formatted disc to SD DVD, 193
- YouTube supported resolutions, 139

SDR (standard dynamic range) video

- HDR (high dynamic range) vs., 14–15
- YouTube bitrate requirements, 141

SECAM (Sequentiel Couleur Avec Memoire), frame rate standard, 3

Smartphones, VR (virtual reality) and, 265

SMPTE (Society of Motion Picture and Television Engineers)

- data rates, 20
- DCP frame rate, 223
- DCP naming conventions, 217
- DCP types, 213–214
- IMF, 212

Social media

- live streaming and, 170–171
- uses of online video, 133

Software

- Blu-ray, 192
- DVD, 192

Software requirements

- analysis and quality control, 64–65
- aspect ratio calculators, 63–64
- compression tools, 62–63
- disc rippers, 61–62
- players, 60
- preparing for video compression, 60
- video storage and data rate calculators, 64

Sorenson Squeeze, encoding equipment, 54

Source files, checking DCP source files, 237–239

SOX, 253

Spatial (DCT) video compression, 11–13

Spatial/geometric preprocessing

- cropping, 78–80
- overview of, 78
- scaling, 80–82

Spatial resolution, 7–8

Spherical video. *See* 360-degree video

SplitmediaLabs XSplit Broadcaster/XSplit Gamecaster, 177–180

Square pixels

- compression parameters, 24
- pixel aspect ratio correction, 93
- YouTube supported resolutions, 139–140

Squeeze (Sorensen), encoding equipment, 54

SSD, input/output options, 59

Standard definition. *See* SD (standard definition)

Standard dynamic range video. *See* SDR (standard dynamic range) video

Standards

- aspect ratio, 8, 24
- motion pictures and television. *See* SMPTE (Society of Motion Picture and Television Engineers)
- television. *See* NTSC (National Television Standards Committee)
- video frame rate, 3

Stereo mixdowns, 95–96

Stereoscopic video, 360-degree, 268

Storage  
     hardware requirements for video  
         compression, 56–58  
         policies, 68  
 Storage area network (SAN), 56–57  
 Streaming. *See also* ABR (adaptive bitrate streaming)  
     compressionist on, 75  
     formats, 18  
     starting streaming endpoint, 152–153  
     traditional, 17  
 Streaming video  
     360-degree video, 270  
     camera, 166  
     capture device, 166–167  
     CDNs (content distribution networks), 169  
     challenges, 170  
     cloud-based transcoding, 167–168  
     encoder for, 167  
     equipment, 166  
     to Facebook Live via OBS, 173–177  
     mirroring live stream to multiple sites, 181  
     platforms for live streaming, 171–172  
     push vs. pull, 168  
     social media and, 170–171  
     video-streaming host, 167  
     to YouTube Live using XSplit, 177–180  
 Subtitles, creating for DCP, 226–228  
 Sweetening. *See* Preprocessing audio  
 Switch (Telestream), third-party players, 37–38

## T

Tachyon (Cinnafilm), in temporal preprocessing  
     conversion, 84  
 Tektronix Aurora, analysis and quality control  
     tools, 65  
 Telecine, temporal preprocessing, 83–84  
 Telestream Switch, third-party players, 37–38  
 Telestream Vantage, 123–124  
 Telestream VidChecker, 65  
 Temporal preprocessing  
     conversion difficulties, 84  
     frame rate conversions, 82–83  
     overview of, 82  
     telecine/inverse telecine, 83–84  
 Teranex (BlackMagic Design), in temporal  
     preprocessing conversion, 84

Tethered headsets, VR (virtual reality) and,  
     265–266  
 Theater management system (TMS), 228  
 Third-party players, 36–38  
 TIFF files  
     encoding intermediate TIFF files to JPEG  
         2000, 236–237  
     rendering image sequence of, 230–233  
 Time Machine, backup systems, 58  
 Timeliness, quality weighed against, 48  
 Title-safe area, cropping and, 79–80  
 TMS (theater management system), 228  
 Transcoders/transcoding. *See also* Encoders  
     cloud-based, 167–168  
     easyDCP Standalone JPEG 2000 Transcoder,  
         236–237  
     equipment required for live streaming, 167  
     preparing for video compression, 72  
     producing Blu-ray Disc and, 201  
     several video at once using Zencoder, 148–151  
 Translation movement, VR degrees of  
     freedom, 264  
 Trial and error, value of experimentation, 70–71  
 TV (Television)  
     choosing television standard when producing  
         Blu-ray Disc, 201  
     luma adjustments, 86  
     uses of online video, 134  
 Twitch  
     live streaming platforms, 172  
     OBS support, 173  
 Twitter, 170, 172

## U

UHD, 255  
 UI (user interface)  
     Adobe Media Encoder CC, 111–112  
     Compressor (Apple), 116–117  
     HandBrake, 103–104  
     MPEG Streamclip, 107–108  
 Underscan mode, 79  
 Unique user ID (UUID), assigning to DCP  
     assets, 220  
 Upconvert, scaling video up, 80  
 User interface. *See* UI (user interface)  
 UUID (unique user ID), assigning to DCP  
     assets, 220

**V**

- Vantage (Telestream), 123–124
- VBR (variable bit rate), 21
- Vegas DVD Architect, 203
- vertical-line resolution, for NTSC and PAL, 6
- VF (version files), DCP, 219–220
- VidChecker (Telestream), 65
- Video
  - 360-degree video. *See* 360-degree video
  - actual resolution and perceived resolution, 268
  - aspect ratio, 7–8
  - Azure Media Services codecs supported, 155
  - on-demand delivery. *See* VOD (video-on-demand)
  - frames and fields, 3–4
  - live delivery. *See* Video, live delivery
  - main picture essence in DCP, 215
  - overview of, 2–3
  - preprocessing. *See* Preprocessing video
  - progressive scan, 4–5
  - resolutions, 5–6
  - SD codecs, 187
  - storage calculators, 64
- Video compression
  - correct aspect ratio in, 7
  - determining compression needs, 44–48
  - encoding real time video. *See* Video, live delivery
  - encoding video-on-demand. *See* VOD (video-on-demand)
  - high dynamic range (HDR) video, 14–15
  - how it works, 9–10
  - interframe and intraframe compression and, 13–14
  - lossless and lossy compression, 11
  - overview of, 1–2
  - parameters, 20
  - preparing for. *See* Preparation, for video compression
  - quantization and, 13
  - spatial (DCT) and wavelet compression, 11–12
- Video-gaming, 261
- Video, live delivery
  - camera, 166
  - capture device, 166–167
  - CDNs (content distribution networks), 169
  - challenges in live streaming, 170
  - cloud-based transcoding, 167–168
  - encoder for, 167
  - equipment for live streaming, 166
  - mirroring live stream to multiple sites, 181
  - overview of, 165
  - platforms for live streaming, 171–172
  - push vs. pull, 168
  - social media and, 170–171
  - streaming to Facebook Live via OBS, 173–177
  - streaming to YouTube Live using XSplit, 177–180
  - video-streaming host, 167
- Video-on-demand. *See* VOD (video-on-demand)
- Video-streaming host, 167
- Video.js, HTML5 players, 39
- VideoSpace, video storage and data rate calculators, 64
- Vimeo, video hosting websites, 135
- Virtual reality. *See* VR (virtual reality)
- Visual Networking Index (Cisco), 134
- VLC, third-party players, 36–37
- VOBs (video objects)
  - storing video data, 61
  - .vob extensions, 188–189
- VOD (video-on-demand)
  - encoding ABR file for. *See* ABR (adaptive bitrate streaming)
  - encoding for HTML5 video. *See* HTML5 video
  - encoding for YouTube. *See* YouTube
  - online delivery, 16–17
  - overview of, 133
  - publishing for business use, 135–136
  - publishing for personal use, 135
  - state of online video, 133–135
  - video hosting websites, 135
  - WAVE (Web Application Video Ecosystem), 136–137
- VOLINDEX file, in DCP, 214
- Volume adjustments, preprocessing audio, 94–95
- VP9
  - delivery codes, 30
  - optimizing 360 video, 270
- VR Studio Purple Pill, 271
- VR (virtual reality)
  - augmented reality, 262
  - hardware, 264
  - mixed reality and extended reality, 263
  - overview of, 261



VR (*continued*)

- smartphones, 265
- tethered headsets, 265–266
- what it is, 261–262

VREncoder, encoding 360-degree video, 271

**W**

- Watch folder, automation features, 53
- WAV, audio wrappers/containers, 25
- WAVE (Web Application Video Ecosystem), 136–137
- Wavelet-based video compression, 11–13
- Weave, technique for deinterlacing video, 90
- Web Application Video Ecosystem (WAVE), 136–137
- Web browsers, HTML5 support, 146
- Webcam, equipment required for live streaming, 166
- Websites, video hosting, 135
- Wide-screen
  - 16:9 aspect ratio for TV, 8
  - “scope” formats, 7–8
- Windows Media Player
  - encoding as an editor, 50
  - web browser plug-ins for video playback, 146
- Windows OS
  - AME support, 111
  - backup systems, 58
  - built-in players, 34–35
  - gamma and, 87
  - HandBrake support, 106
  - MPEG-4 support, 31
  - Switch support, 38
  - video storage and data rate calculators, 64
- WMA, audio wrappers/containers, 25
- WMV, media wrappers/containers, 25
- Work in progress, evaluating compression needs, 44
- Workflow
  - for compressionist, 52
  - for editors, 49–51
- Wrappers. *See* Containers (wrappers/formats)
- WraptorDCP, 253

**X**

- XAVC, acquisition codes, 28
- XDCAM, acquisition codes, 28
- XF-AVC, acquisition codes, 28
- XML
  - CPL (composition playlist), 218
  - data and metadata track files in IMF, 257
- XSplit Broadcaster/XSplit Gamecaster (SplitmediaLabs), 177–180
- X’Y’Z’ color space, 223–224

**Y**

- Y (luma), in Y’CbCr color space. *See also* Luma (black and white or lightness), 85
- Y’CbCr (YUV) color space, 85
- YouTube
  - bitrate provisions, 140–141
  - codecs, 141, 143
  - encoding for, 137
  - encoding using Adobe Premiere Pro, 143–145
  - facts regarding, 138
  - file size and duration limits, 139
  - frame rates, 140
  - OBS support, 173
  - progressive scanning required, 140
  - recommendations summarized, 142
  - resolution and pixel aspect ratio recommendations, 139–140
  - streaming to YouTube Live using XSplit, 177–180
  - video formats supported, 139
  - video hosting websites, 135
  - video streaming hosts, 167
- YouTube Live
  - live streaming platforms, 171
  - restrictions, 180
  - streaming to using XSplit, 177–180
- YUV (Y’CbCr) color space, 85

**Z**

- Zencoder (Brightcove), transcoding several video at once using, 148–151