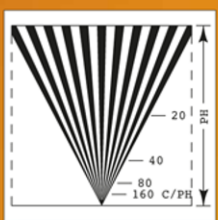
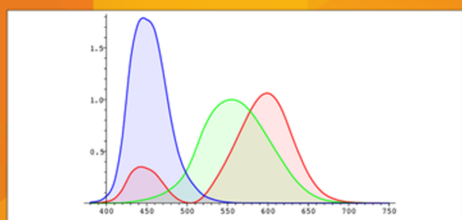
SECOND EDITION

DIGITAL VIDEO AND HD

ALGORITHMS AND INTERFACES





CHARLES POYNTON

MORGAN KAUFMANN

SECOND EDITION

DIGITAL VIDEO AND HD

ALGORITHMS AND INTERFACES

This page intentionally left blank

SECOND EDITION

DIGITAL VIDEO AND HD

ALGORITHMS AND INTERFACES

CHARLES POYNTON

ELSEVIER

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

Morgan Kaufmann is an imprint of Elsevier

MK MORGAN KAUFMANN

Acquiring editor: Laura Lewin
Development Editor: Graham Smith
Project Manager: Sarah W. Binns
Design, illustration, and composition: Charles Poynton
Copy editor/proofreader: Charles Roumeliotis
Cover Design: Joanne Blank

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

Library of Congress Cataloging-in-Publication Data
Application submitted.

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

ISBN: 978-0-12-391926-7

For information on all Morgan Kaufmann publications, visit our Web site at www.mkp.com or www.elsevierdirect.com

Printed in the United States of America on acid-free paper.

2012 2013 2014 2015 2016 2017 2018 9 8 7 6 5 4 3 2 1

Dedicated to
my dear friend
James C. (Jamie) Avis
1949–2011

This page intentionally left blank

Contents

This page intentionally left blank

Figures

Tables

Preface

Video technology continues to advance since the publication, in early 2003, of the first edition of this book. Further "convergence" – Jim Blinn might say collision – between video and computer graphics has occurred. Television is losing; computing and internet transport are winning. Even the acronym "TV" is questionable today: Owing to its usage over the last half century, *TV* implies broadcast, but much of today's video – from the Apple iTunes store, Hulu, NetFlix, YouTube – is not broadcast in the conventional sense. In this edition, I have replaced *SDTV* with *SD* and *HDTV* with *HD.*

Digital video is now ubiquitous; analog scanning, as depicted by Figure P.1 below, is archaic. In this edition, I promote the pixel array to first-class status. The first edition described scan lines; I have retrained myself to speak of image rows and image columns instead. I expunge microseconds in favour of sample counts; I expunge millivolts in favour of pixel values. Phrases in the previous edition such as "immense data capacity" have been replaced by "fairly large" or even "modest data capacity."

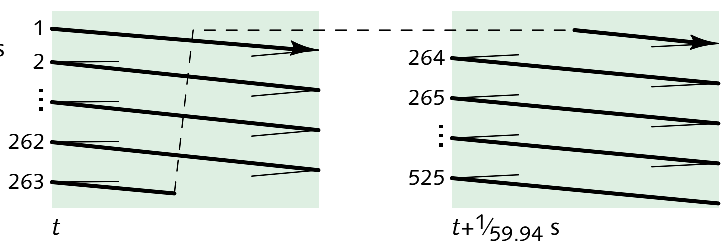


Figure P.1 **Scanning a raster,** as suggested by this sketch, is obsolete. In modern video and HD, the image exists in a pixel array. Any book that describes image acquisition or display using a drawing such as this doesn't accurately portray digital video.

In my first book *Technical Introduction to Digital Video,* published in 1996, and in the first edition of the present book, I described encoding and then decoding. That order made sense to me from an engineering perspective. However, I now think it buries a deep philosophical flaw. Once program material is prepared, decoded, and viewed on a reference display in the studio, and mastered – that is, downstream of final approval – only the decoding and display matters. It is convenient for image data to be captured and encoded in a manner that displays a realistic image at review, but decoding and presentation of the image data at mastering is preeminent. If creative staff warp the colours at encoding in order to achieve an æsthetic effect – say they lift the black levels by 0.15, or rotate hue by 123° – the classic encoding equations no longer apply, but those image data modifications are not evidence of faulty encoding, they are consequence of the exercise of creative intent. The traditional explanation is presented in a manner that suggests that the encoding is fixed; however, what really matters is that *decoding* is fixed. The principle that I'm advocating is much like the principle of MPEG, where the *decoder* is defined precisely but the *encoder* is permitted to do anything that produces a legal bitstream. In this edition I emphasize decoding. A new chapter – Chapter 2, on page 19 – outlines this philosophy.

Many chapters here end with a *Further reading* section if extensive, authoritative information on the chapter's topic is available elsewhere.

Video technology is a broad area. There are entire books that cover subject matter for which this book provides only a chapter. My expertise centres on image coding aspects, particularly the relationship between vision science, colour science, image science, signal processing, and video technology. Chapters of this book on those topics – mainly, the topics of the *Theory* part of this book – have (as far as I know) no textbook counterpart.

Legacy technology

SMPTE is the source of many standards, both legacy and modern. During the interval between the first and second editions of this book, SMPTE abolished the *M* suffix of many historical standards, and prepended *ST* to standards (paralleling *EG* for Engineering Guideline and

RP for Recommended Practice). I cite recent SMPTE standards according to the new nomenclature.

As of 2012, we can safely say that analog television technology and composite (NTSC/PAL) television technology are obsolete. When writing the first edition of this book, I concentrated my efforts on the things that I didn't expect to change rapidly; nonetheless, perhaps 15 or 20 percent of the material in the first edition represents technology – mainly analog and composite and NTSC and PAL – that we would now classify as "legacy." It is difficult for an author to abandon work that he or she has written that represents hundreds or thousands of hours of work; nonetheless, I have removed this material and placed it in a self-contained book entitled *Composite NTSC and PAL: Legacy Video Systems* that is freely available on the web.

www.poynton.com/CNPLVS/

Layout and typography

I designed this book with wide margins. I write notes here, and I encourage you to do the same!

Many years ago, when my daughter Quinn was proofreading a draft chapter of the first edition of this book, she circled, in red, two lines at the top of a page that were followed by a new section. She drew an arrow indicating that the two lines should be moved to the bottom of the previous page. She didn't immediately realize that the lines had wrapped to the top of the page because there was no room for them earlier. She marked them nonetheless, and explained to me that they needed to be moved because that section should start at the top of the page. Quinn intuitively understood the awkward page break – and she was only twelve years old! I have spent a lot of time executing the illustration, layout, and typesetting for this book, based upon my belief that this story is told not only through the words but also through pictures and layout.

In designing and typesetting, I continue to be inspired by the work of Robert Bringhurst, Jan Tschichold, and Edward R. Tufte; their books are cited in the margin.

Bringhurst, Robert (2008), *The Elements of Typographic Style*, version/edition 3.1, (Vancouver, B.C.: Hartley & Marks).

Tschichold, Jan (1991), *The Form of the Book* (London: Lund Humphries). [Originally published in German in 1975.]

Tufte, Edward R. (1990), *Envisioning Information* (Cheshire, Conn.: Graphic Press).

Formulas

It is said that every formula in a book cuts the potential readership in half. I hope readers of this book can compute that after a mere ten formulas my readership would drop to 2^{-10}! I decided to retain formulas, but they are not generally necessary to achieve an understanding of the concepts. If you are intimidated by a formula, just skip it and come back later if you wish. I hope that you will treat the mathematics the way that Bringhurst recommends that you treat his mathematical description of the principles of page composition. In Chapter 8 of his classic book, *Elements of Typographic Style*, Bringhurst says,

> "The mathematics are not here to impose drudgery upon anyone. On the contrary, they are here entirely for pleasure. They are here for the pleasure of those who like to examine what they are doing, or what they might do or have already done, perhaps in the hope of doing it still better. Those who prefer to act directly at all times, and leave the analysis to others, may be content in this chapter to study the pictures and skim the text."

Spelling

At the urging of my wife Barbara and my two daughters, I have resumed spelling colour with a *u.* However, *colorimetric* and *colorimetry* are without. *Greyscale* is now spelled with an *e* (for English), not with an *a* (for American). The world is getting smaller, and Google's reach is worldwide; however, cultural diversity shouldn't suffer.

www.poynton.com/DVAI2/

I tried carefully to avoid errors while preparing this book. Nonetheless, despite my efforts and the efforts of my reviewers, a few errors may have crept in. As with my previous book, I will compile errata for this book and make the corrections available at the URL indicated in the margin. Please report any error that you discover, and I will endeavour to repair it and attribute the correction to you!

Charles Poynton
Toronto, Jan. 2012

Acknowledgments

My introduction to digital video was writing microcode many years ago for hardware conceived by John Lowry and engineered by Richard Kupnicki. John Ross, founder of Ross Video, continued my education in video. I thank all three.

I spent many hours at CJOH-TV in Ottawa, Canada, testing my theories at the invitation of CJOH's Vice-President of Engineering, Austin Reeve. I thank him for his confidence, good humor, and patience.

I owe a debt of gratitude to four people who have been not only colleagues but close personal friends for a few decades: C.R. Caillouet, Pierre Deguire, Charlie Pantuso, and Mark Schubin.

I thank the colleagues who encouraged me in this project, all of whom reviewed the manuscript at various stages: David Bancroft, George Joblove, Peter Symes and especially Dave LeHoty – who suggested the title, long ago!

Portions of the manuscript were reviewed, and error reports were provided by the following people, who I thank: Don Craig, Joseph Goldstone, and Adam Wilt. My apologies to them, also, because not all of their suggestions could be included this time around.

I thank the netizens that contributed error reports to the first edition, especially Ken Greenebaum, Dragan Matković, and Andrew Murray; thanks especially to Jay Zipnick for contributing several suggestions regarding colour calculations.

While writing this book, I was working in a Ph.D. program under the supervision of Brian Funt. I thank him for his contributions to that effort, many of which have found their way into this book.

Diane Cerra was my patient and thoughtful editor for my first book, and for the first edition of this one. Her ideas and hard work shaped this second edition. Sara Binns, Laura Lewin, Sara Scott, and Graham Smith comprised my thoroughly professional editorial staff at Morgan Kaufmann; I thank them. I also thank my superb copy editor, Charles Roumeliotis.

Thanks to my family, for their love and encouragement: Peg, Al, Kim, Brenna, Alana, Dad, and Jean. Thanks also to Jeri, Corot, Ben, and Paige.

Thanks to Barbara, the love of my life, and our daughters Quinn and Georgia.

Part 1

Introduction

This page intentionally left blank

Raster images 1

A *vector* image comprises data describing a set of geometric primitives, each of which is associated with grey or colour values. A process of interpretation – *rasterizing,* or *raster image processing,* or *ripping* – is necessary to convert a vector image to a raster. *Vector* suggests a straight line but paradoxically, "vector" images commonly contain primitives describing curves.

A digital image is represented by a rectangular array (matrix) of *picture elements* (*pels,* or *pixels*). Pixel arrays of several image standards are sketched in Figure 1.1. In a greyscale system each pixel comprises a single component whose value is related to what is loosely called brightness. In a colour system each pixel comprises several components – usually three – whose values are closely related to human colour perception.

Historically, a video image was acquired at the camera, conveyed through the channel, and displayed using analog scanning; there was no explicit pixel array. Modern cameras and modern displays directly represent the discrete elements of an image array having fixed structure. Signal processing at the camera, in the pipeline, or at the display may perform spatial and/or temporal resampling to adapt to different formats.

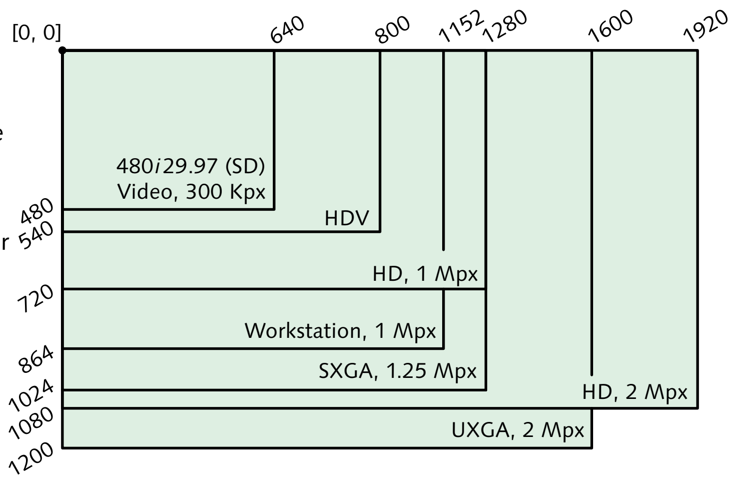


Figure 1.1 **Pixel arrays** of several imaging standards are shown, with their counts of image columns and rows. The 640×480 square sampled structure common in computing is included; however, studio and consumer 480*i* standards are sampled 704×480 or 720×480 with nonsquare sampling.

In art, the frame surrounds the picture; in video, the frame *is* the picture.

The pixel array is for one image is a *frame.* In video, digital memory used to store one image is called a *framestore;* in computing, it's a *framebuffer*. The total pixel count in an image is the number of image columns N_C (or in video, *samples per active line*, S_{AL}) times the number of image rows N_R (or *active lines*, L_A). The total pixel count is usually expressed in megapixels (Mpx).

In video and in computing, a pixel comprises the set of *all* components necessary to represent colour (typically red, green, and blue). In the mosaic sensors typical of digital still cameras (DSCs) a pixel is *any* colour component individually; the process of *demosaicking* interpolates the missing components to create a fully populated image array. In digital cinema cameras the DSC interpretation of *pixel* is used; however, in a digital cinema projector, a pixel is a triad.

A computer enthusiast refers to the image column and row counts (*width* × *height*) as *resolution.* An image engineer reserves the term *resolution* for the image detail that is acquired, conveyed, and/or delivered. Pixel count imposes an upper limit to the image detail; however, many other factors are involved.

The value of each pixel component represents brightness and colour in a small region surrounding the corresponding point in the sampling lattice.

Pixel component values are quantized, typically to an integer value that occupies between 1 and 16 bits – and often 8 or 10 bits – of digital storage. The number of bits per component, or per pixel, is called the *bit depth.* (We use *bit depth* instead of *width* to avoid confusion: The term *width* refers to the entire picture.)

Aspect ratio

Aspect ratio is simply the ratio of an image's width to its height. Standard aspect ratios for film and video are sketched, to scale, in Figure 1.2. What I call simply *aspect ratio* is sometimes called *display aspect ratio*

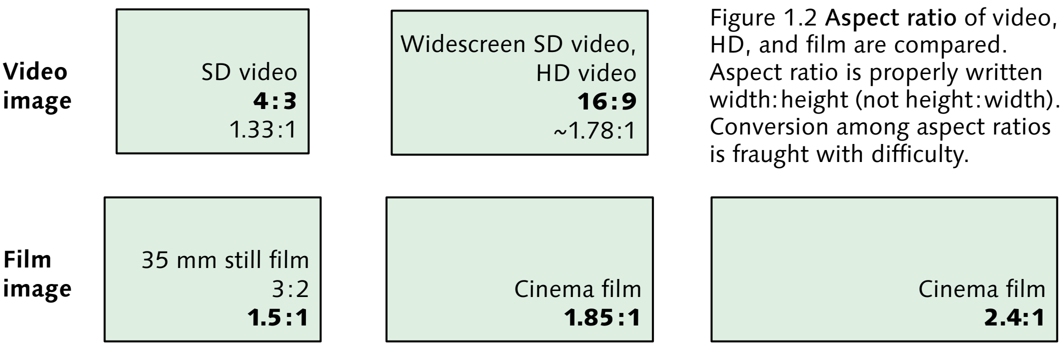


Figure 1.2 **Aspect ratio** of video, HD, and film are compared. Aspect ratio is properly written width:height (not height:width). Conversion among aspect ratios is fraught with difficulty.

$$\frac{width}{height} = \frac{AR}{SAR} \qquad \text{Eq 1.1}$$

$$N_C = \sqrt{n \cdot AR}; \quad N_R = \sqrt{\frac{n}{AR}} \qquad \text{Eq 1.2}$$

(DAR) or *picture aspect ratio* (PAR). *Standard-definition* (SD) television has an aspect ratio of 4:3.

Equation 1.1 relates picture and sample aspect ratios. To assign *n* square-sampled pixels to a picture having aspect ratio *AR*, choose image column and image row counts (*c* and *r*, respectively) according to Equation 1.2.

Cinema film commonly uses 1.85:1 (which for historical reasons is called either *flat* or *spherical*), or 2.4:1 ("CinemaScope," or colloquially, '*scope*). Many films are 1.85:1, but "blockbusters" are usually 2.4:1. Film at 2.4:1 aspect ratio was historically acquired using an aspherical lens that squeezes the horizontal dimension of the image by a factor of two. The projector is equipped with a similar lens, to restore the horizontal dimension of the projected image. The lens and the technique are called *anamorphic.* In principle, an anamorphic lens can have any ratio; in practice, a ratio of exactly two is ubiquitous in cinema.

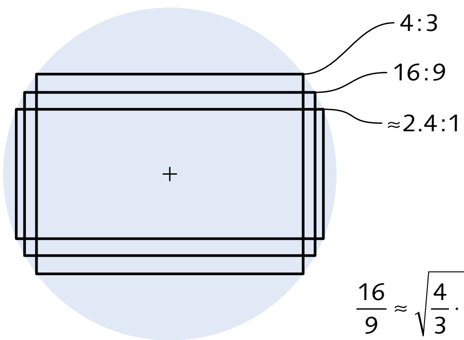
In Europe and Asia, 1.66:1 was the historical standard for cinema, though 1.85 is increasingly used owing to the worldwide market for entertainment imagery.

Widescreen refers to an aspect ratio wider than 4:3. *High-definition* (HD) television is standardized with an aspect ratio of 16:9. In video, the term *anamorphic* usually refers to a 16:9 widescreen variant of a base video standard, where the horizontal dimension of the 16:9 image occupies the same width as the 4:3 aspect ratio standard. Consumer electronic equipment rarely recovers the correct aspect ratio of such conversions (as we will explore later in the chapter.)

FHA: Full-height anamorphic

HD is standardized with an aspect ratio of 16:9 (about 1.78:1), fairly close to the 1.85:1 ordinary movie aspect ratio. Figure 1.3 below illustrates the origin of the 16:9 aspect ratio. Through a numerological coincidence apparently first revealed by Kerns Powers, the

Schubin, Mark (1996), "Searching for the perfect aspect ratio," in *SMPTE Journal* **105** (8): 460–478 (Aug.).



$$\frac{16}{9} \approx \sqrt{\frac{4}{3} \cdot 2.4}$$

Figure 1.3 **The choice of 16:9 aspect ratio** for HD came about because 16:9 is very close to the geometric mean of the 4:3 picture aspect ratio of conventional television and the 2:4:1 picture aspect ratio of CinemaScope movies.

geometric mean of 4:3 (the standard aspect ratio of conventional television) and 2.4 (the aspect ratio of a CinemaScope movie) is very close – within a fraction of a percent – to 16:9. (The calculation is shown in the lower right corner of the figure.) A choice of 16:9 for HD meant that SD, HD, and CinemaScope shared the same "image circle": 16:9 was a compromise between the vertical cropping required for SD and the horizontal cropping required for CinemaScope.

Geometry

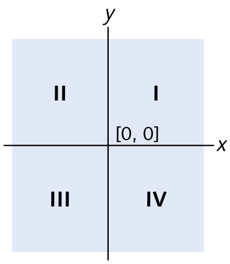


Figure 1.4 **Cartesian coordinates** [*x*, *y*] define four quadrants. *Quadrant I* contains points having positive *x* and *y* values. Coordinates in quadrant I are used in some imaging systems. *Quadrant IV* contains points having positive *x* and negative *y*. Raster image coordinates are ordinarily represented with image row numbers increasing down the height of the image – that is, in quadrant IV, but omitting the negative sign on the *y* values.

In mathematics, coordinate values of the (two-dimensional) plane range both positive and negative. The plane is thereby divided into four quadrants (see Figure 1.4). Quadrants are denoted by Roman numerals in the counterclockwise direction. In the continuous image plane, locations are described using Cartesian coordinates [*x*, *y*] – the first coordinate is associated with the horizontal direction, the second with the vertical. When both *x* and *y* are positive, the location is in the *first quadrant* (quadrant I). In image science, the image lies in this quadrant. (Adobe's Postscript system uses first-quadrant coordinates.)

In matrix indexing, axis ordering is reversed from Cartesian coordinates: A matrix is indexed by *row* then *column*. The top row of a matrix has the smallest index, so matrix indices lie in quadrant IV. In mathematics, matrix elements are ordinarily identified using 1-origin indexing. Some image processing software packages use 1-origin indexing – in particular, MATLAB and Mathematica, both of which have deep roots in mathematics. The scan line order of conventional video and image processing usually adheres to the matrix convention, but with zero-origin indexing: Rows and columns are usually numbered [*r*, *c*] from [0, 0] at the top left. In other words, the image is in quadrant IV (but eliding the negative sign on the *y*-coordinate), but ordinarily using zero-origin indexing.

Digital image sampling structures are denoted *width*×*height*. For example, a 1920×1080 system has columns numbered 0 through 1919 and rows (historically, "picture lines") numbered 0 through 1079.

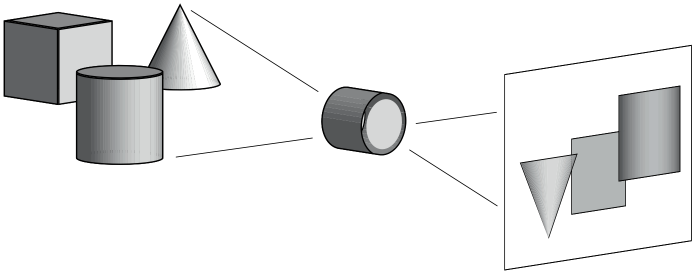


Figure 1.5 **Scene, lens, image plane**

Image capture

In human vision, the three-dimensional world is imaged by the lens of the eye onto the retina, which is populated with photoreceptor cells that respond to light having wavelengths ranging from about 400 nm to 700 nm. In video and in film, we build a camera having a lens and a photosensitive device, to mimic how the world is perceived by vision. Although the shape of the retina is roughly a section of a sphere, it is topologically two dimensional. In a camera, for practical reasons, we employ a flat *image plane,* sketched in Figure 1.5 above, instead of a section of a sphere. Image science involves analyzing the continuous distribution of optical power that is incident on the image plane.

Digitization

Signals captured from the physical world are translated into digital form by *digitization,* which involves two processes: *sampling* (in time or space) and *quantization* (in amplitude), sketched in Figure 1.6 below. The operations may take place in either order, though sampling usually precedes quantization.

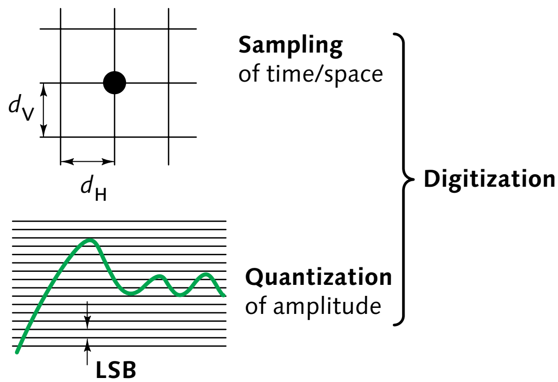


Figure 1.6 **Digitization** comprises *sampling* and *quantization*, in either order. Sampling density, expressed in units such as pixels per inch (ppi), relates to resolution. Quantization relates to the number of bits per pixel (bpp) or bits per component/channel (bpc). Total data rate or data capacity depends upon the product of these two factors.

Quantization

Quantization assigns an integer to signal amplitude at an instant of time or a point in space, as I will explain in *Quantization,* on page 37. Virtually all image exchange standards – TIFF, JPEG, SD, HD, MPEG, H.264 – involve pixel values that are *not* proportional to light power in the scene or at the display: With respect to light power, pixel values in these systems are nonlinearly quantized.

1-D sampling

A continuous one-dimensional function of time, such as audio sound pressure level, is sampled through forming a series of discrete values, each of which is a function of the distribution of a physical quantity (such as intensity) across a small interval of time. *Uniform sampling,* where the time intervals are of equal duration, is nearly always used. (Details will be presented in *Filtering and sampling,* on page 191.)

2-D sampling

A continuous two-dimensional function of space is sampled by assigning, to each element of the image matrix, a value that is a function of the distribution of intensity over a small region of space. In digital video and in conventional image processing, the samples lie on a regular, rectangular grid.

Analog video was not sampled horizontally; however, it was sampled vertically by scanning and sampled temporally at the frame rate. Historically, samples were not necessarily digital: CCD and CMOS image sensors are inherently sampled, but they are not inherently quantized. (On-chip analog-to-digital conversion is now common in CMOS sensors.) In practice, though, sampling and quantization generally go together.

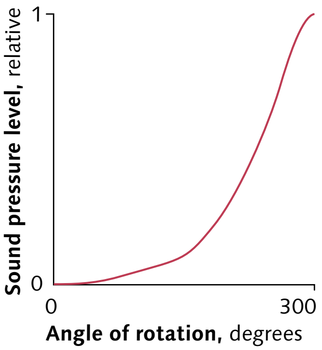


Figure 1.7 **Audio taper** imposes perceptual uniformity on the adjustment of volume. I use the term perceptual uniformity instead of perceptual linearity: Because we can't attach an oscilloscope probe to the brain, we can't ascribe to perception a mathematical property as strong as linearity. This graph is redrawn from Bourns, Inc. (2005), *General Application Note – Panel Controls – Taper.*

Perceptual uniformity

A perceptual quantity is encoded in a *perceptually uniform* manner if a small perturbation to the coded value is approximately equally perceptible across the range of that value. Consider the volume control on your radio. If it were physically linear, the roughly logarithmic nature of loudness perception would place most of the perceptual "action" of the control at the bottom of its range. Instead, the control is designed to be perceptually uniform. Figure 1.7 shows the transfer function of a potentiometer with standard *audio taper:* Angle of rotation is mapped to sound pressure level such that rotating the knob 10 degrees produces

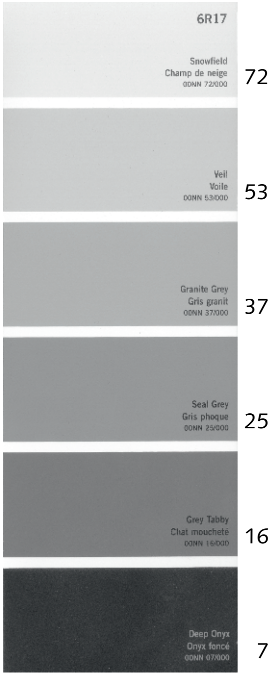


Figure 1.8 **Grey paint samples** exhibit perceptual uniformity: The goal of the manufacturer is to cover a reasonably wide range of reflectance values such that the samples are uniformly spaced as judged by human vision. The manufacturer's code for each chip typically includes an approximate *L** value. In image coding, we use a similar scheme, but with code (pixel) value *V* instead of *L**, and a hundred or a thousand codes instead of six.

a similar perceptual increment in volume across the range of the control. This is one of many examples of perceptual considerations built into the engineering of electronic systems. (For another example, see Figure 1.8.)

CIE: *Commission Internationale de L'Éclairage.* See Chapter 25, on page 265.

$0.495 \approx L^*(0.18)$

$0.487 \approx 0.18^{0.42}$

Compared to linear-light encoding, a dramatic improvement in signal-to-noise performance can be obtained by using nonlinear image coding that mimics human lightness perception. Ideally, coding for distribution should be arranged such that the step between pixel component values is proportional to a *just noticeable difference* (JND) in physical light power. The CIE standardized the *L** function in 1976 as its best estimate of the lightness sensitivity of human vision. Although the *L** equation incorporates a cube root, *L** is effectively a power function having an exponent of about 0.42; 18% "mid grey" in relative luminance corresponds to about 50 on the *L** scale from 0 to 100. The inverse of the *L** function is approximately a 2.4-power function. Most commercial imaging systems incorporate a mapping from digital code value to linear-light luminance that approximates the inverse of *L**.

EOCF: *Electro-optical conversion function*. See Chapter 27, *Gamma*, on page 315.

Different EOCFs have been standardized in different industries:

- In digital cinema, DCI/SMPTE standardizes the reference (approval) projector; that standard is closely approximated in commercial cinemas. The standard digital cinema reference projector has an EOCF that is a pure 2.6-power function.

- In SD and HD, EOCF was historically poorly standardized or not standardized at all. Consistency has been achieved only through use of *de facto* industry-standard CRT studio reference displays having EOCFs well approximated by a 2.4-power function. In 2011, BT.1886 was adopted formalizing the 2.4-power, but reference white luminance and viewing conditions are not [yet] standardized.
- In high-end graphics arts, the Adobe RGB 1998 industry standard is used. That standard establishes a reference display and its viewing conditions. Its EOCF is a pure 2.2-power function.
- In commodity desktop computing and low-end graphics arts, the sRGB standard is used. The sRGB standard establishes a reference display and its viewing conditions. Its EOCF is a pure 2.2-power function.

Colour

Vision when only the rod cells are active is termed *scotopic.* When light levels are sufficiently high that the rod cells are inactive, vision is *photopic.* In the *mesopic* realm, both rods and cones are active.

To be useful for colour imaging, pixel components represent quantities closely related to human colour vision. There are three types of photoreceptor *cone* cells in the retina, so human vision is trichromatic: Three components are necessary and sufficient to represent colour for a normal human observer. Rod cells constitute a fourth photoreceptor type, responsible for what can loosely be called night vision. When you see colour, cone cells are responding. Rod (scotopic) vision is disregarded in the design of virtually all colour imaging systems.

Colour images are generally best captured with sensors having spectral responsivities that peak at about 630, 540, and 450 nm – loosely, red, green, and blue – and having spectral bandwidths of about 50, 40, and 30 nm respectively. Details will be presented in Chapters 25 and 26.

The term *multispectral* refers to cameras and scanners, or to their data representations. Display systems using more than three primaries are called *multiprimary.*

In *multispectral* and *hyperspectral* imaging, each pixel has 4 or more components each representing power from different wavelength bands. Hyperspectral refers to a device having more than a handful of spectral components. There is currently no widely accepted definition of how many components constitute multispectral or hyperspectral. I define a multispectral system as having between 4 and 10 spectral components, and a hyperspectral system as having 11 or more. Hyperspectral systems may be described as having colour, but they are usually designed for purposes of science, not vision: A set of pixel component values in a hyperspectral system usually has no close relationship to colour perception. Apart from highly specialized applications such as satellite imaging and the preservation or reproduction of fine art, multispectral and hyperspectral techniques are not used in commercial imaging.

Luma and colour difference components

Some digital video equipment uses *R'G'B'* components directly. However, human vision has considerably less ability to sense detail in colour information than in lightness. Provided achromatic detail is maintained, colour detail can be reduced by *subsampling,* which is a form of spatial filtering (or averaging).

A colour scientist might implement subsampling by forming relative luminance as a weighted sum of linear *RGB* tristimulus values, then imposing a nonlinear transfer function approximating CIE lightness (L^*). In video, we depart from the theory of colour science, and implement an engineering approximation that I will describe in *Constant luminance,* on page 107. Briefly, component video systems convey image data as a luma component, Y', approximating lightness and coding the achromatic component, and two colour difference components – in the historical analog domain, P_B and P_R, and in digital systems, C_B and C_R – that represent colour disregarding lightness. The colour difference components are subsampled (horizontally, or both horizontally and vertically) to reduce their data rate. $Y'C_BC_R$ and $Y'P_BP_R$ components are explained in *Introduction to luma and chroma,* on page 121.

Digital image representation

Many different file, memory, and stream formats are used to convey still digital images and motion sequences. Most formats have three components per pixel (representing additive red, green, and blue colour components). In consumer electronics and commodity computing, most formats have 8 bits per component. In professional applications such as studio video and digital cinema, 10, 12, or more bits per component are typically used.

Imaging systems are commonly optimized for other aspects of human perception; for example, the JPEG and MPEG compression systems exploit the spatial frequency characteristics of vision. Such optimizations can also be referred to as perceptual coding.

Virtually all commercial imaging systems use perceptual coding, whereby pixel values are disposed along a scale that approximates the capability of human vision to distinguish greyscale shades. In colour science, capital letter symbols *R*, *G*, and *B* are used to denote *tristimulus values* that are proportional to light power in three wavelength bands. Tristimulus values are not perceptually uniform. It is explicit or implicit in nearly all commercial digital imaging systems that pixel component values are coded as the desired display *RGB* tristimuli raised to a power between about $^1/_{2.2}$ (that is, about 0.45) and $^1/_{2.6}$ (that is, about 0.38). Pixel values so constructed are denoted with primes: *R'G'B'* (though the primes are often omitted, causing confusion).

In order for image data to be exchanged and interpreted reasonably faithfully, digital image standards define pixel values for reference black and reference white. Digital image standards typically specify a target luminance for reference white. Most digital image standards offer no specific reflectance or relative luminance for reference black; it is implicit that the display system will make reference black as dark as possible.

It is a mistake to place a linear segment at the bottom of the sRGB EOCF. (A linear segment *is* called for in the OECF defined in sRGB, but that's a different matter.)

In computing, 8-bit digital image data ranges from reference black at code 0 to reference white at code 255. The sRGB standard calls for an exponent ("gamma") of 2.2 at the display. Reference white is supposed to display luminance of 80 $cd \cdot m^{-2}$, though in practice values up to about 320 $cd \cdot m^{-2}$ are common.

In consumer digital video, image data is coded in 8-bit components ranging from reference black at code 16 to reference white at code 235. (In the studio, 10-bit coding is common.) Studio standards call for an exponent ("gamma") of about 2.4 at the display. Common practice places reference white at luminance of about 100 $cd \cdot m^{-2}$, with a dark viewing environment (about 1 lx) and a dark surround (1%). Faithful display is obtained at the consumers' premises when these conventions are followed. It is common for consumer displays to be brighter than 100 $cd \cdot m^{-2}$; consumers' viewing environments are often fairly bright (often around 100 lx) and the surround is often dim (5%). In cases where the consumer display and viewing conditions differ from those at the studio, preserving picture appearance requires display adjustment.

The DVI computer display interface is defined to carry *R'G'B'* values having reference range 0 through 255. The HDMI and DisplayPort interfaces also accommodate that coding, but in addition they accommodate *R'G'B'* values having reference range 16 through 235, they accommodate bit depths deeper than 8 bits, and they accommodate $Y'C_BC_R$ coding. The SDI and HD-SDI interfaces common in studio video accommodate 10-bit *R'G'B'* or $Y'C_BC_R$ values having reference range 64 through 940 (that is, 16·4 and 219·4).

Image data for offset printing is typically represented as 8-bit code values for the relative amounts of cyan, magenta, yellow, and black (*CMYK*) inks required at a pixel location, where [0, 0, 0, 0] defines no ink (that is, producing the colour of the printing substrate, typically white) and [0, 0, 0, 255] defines black. Perceptual uniformity is imposed by what a printer calls *dot gain.* User interfaces typically present the 0 to 255 range as 0 to 100 – that is, the interface expresses percentage.

In some circles, dot gain is now called *tone value increase* (TVI). The *increase* makes things darker: Printers are more concerned with ink than with light.

SD and HD

Until about 1995, the term *television* referred to either 480*i* (historically denoted 525/59.94, or "NTSC") or 576*i* (625/50, or "PAL"). The emergence of widescreen television, high-definition television (HDTV), and other new systems introduced ambiguity into the unqualified term *television*. What we used to call "television" is now *standard-definition television* (SDTV). Video technology was and is widely used to distribute entertainment programming – historically, "television," TV – but applications are now so diverse that I omit the letters *TV* and refer simply to *SD* and *HD.*

Surprisingly, there are no broadly agreed definitions of SD and HD. I classify as SD any video system having a frame rate of 23.976 Hz or more whose digital image has fewer than 1152 image columns or fewer than 648 image rows – that is, fewer than about ¾ million total pixels. I classify as HD any video system with a native aspect ratio of 16:9 and frame rate of 23.976 Hz or more, whose digital image has 1152 or more image columns and 648 or more image rows – that is, a digital image comprising 729 Kpixels (about ¾-million pixels) or more.

My definition excludes 480*p*, 540*p*, and 576*p* as HD, but admits 720*p*. (My threshold of image rows is the geometric mean of 540 and 720.)

A 16:9 image having area of 1 m^2 has dimensions $^4/_3$ m by $^3/_4$ m (about 1.33 m by 0.75 m) and a diagonal of about 1.53 m (60 in). Today, this size represents the largest practical consumer direct-view display.

A 16:9 image having width of 1 m has height of 0.5625 m, and area of 0.5625 m^2. Its diagonal is about 1.15 m, or 45 inches; this is approximately the median size of HD receivers purchased today. At 1920×1080, pixel pitch is $^1/_{1920}$ m, or about 520 µm; each RGB "subpixel" has dimensions roughly 174 µm by 520 µm.

Square sampling

The term *square* refers to the sample arrangement: *Square* does not mean that image information is uniformly distributed throughout a square area associated with each pixel! Some 1080*i* HD compression systems resample to $^4/_3$ or $^3/_2$ pixel aspect ratio.

In modern digital imaging, including computing, digital photography, graphics arts, and HD, samples (pixels) are ordinarily equally spaced vertically and horizontally – they have equal horizontal and vertical sample density. These systems have *square sampling* ("square pixels"); they have *sample aspect ratio* (*SAR*) of unity. With square sampling, the count of image columns is simply the aspect ratio times the count of image rows.

If you use the term *pitch,* it isn't clear whether you refer to the dimension of an element or to the number of elements per unit distance. I prefer the term *density.*

Legacy imaging and video systems including digital SD systems had unequal horizontal and vertical sample pitch (nonsquare sampling). That situation was sometimes misleadingly referred to as "rectangular sampling," but a square is also a rectangle! A heated debate in the early 1990s led to the adoption of square sampling for HD. In 1995 the New York Times wrote,

New York Times (1995, Sep. 11), "Patents; The debate over high-definition TV formats is resolved with a system that provides both"

> HDTV signals can be sent in a variety of formats that rely not only on progressive or interlaced signals, but on features like square versus round pixels …

A technical person finds humour in that statement; surprisingly, though – and unintentionally – it contains a technical grain of truth: In sampled continuous-tone imagery, the image information associated with each sample is spread out over a neighbourhood which, ideally, has circular symmetry.

Comparison of aspect ratios

Concerning the origin of the 16:9 aspect ratio, see page 5.

In the mid-1990s, when HD was undergoing standardization, SD and HD were compared using various measures, depicted in Figure 1.9 at the top of the facing page, based upon the difference in aspect ratio between 4:3 and 16:9. Comparisons were made of equal height,

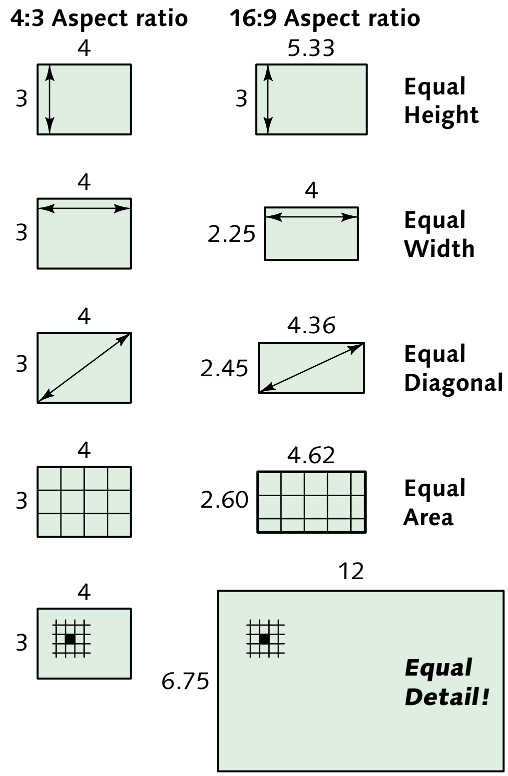


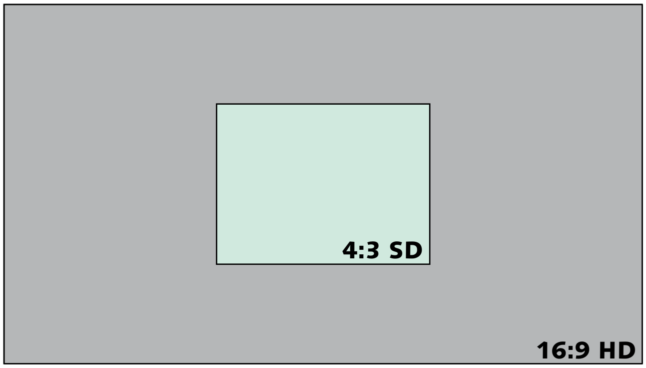
Figure 1.9 **Comparison of aspect ratios** between conventional television (now SD) and HD was attempted using various measures: equal height, equal width, equal diagonal, and equal area. All of these comparisons overlooked the fundamental improvement of HD: its increased pixel count. The correct comparison is based upon equal picture detail. It is the angular subtense of a pixel that should be preserved.

equal width, equal diagonal, and equal area. All of those measures overlooked the fundamental improvement of HD: Its "high definition" (or "resolution") does not squeeze six times the number of pixels into the same visual angle! Instead, the angular subtense of a single pixel should be maintained, and the entire image can now occupy a much larger area of the viewer's visual field. HD allows a greatly increased picture angle. The correct comparison between conventional television and HD is not based upon picture aspect ratio; it is based upon picture detail.

Aspect ratio

With the advent of HD consumer television receivers, it became necessary to display 4:3 (SD) material on 16:9 (HD) displays and 16:9 material on 4:3 displays. During the standardization of HD, I proposed – not entirely facetiously – that SD content at 4:3 should be "pixel-mapped" into the HD frame as sketched in Figure 1.10, preserving aspect ratio and equal detail. I anticipated

Figure 1.10 **SD to HD pixel mapping** is one way to convert 4:3 material to 16:9. The angular subtense of SD pixels is preserved. If CE vendors had adopted this approach at the introduction of HD, today's aspect ratio chaos would have been avoided.



that provisions would be made for the consumer to enlarge the SD image – but the consumer would have been aware of two qualitatively different image sources. (My idea wasn't adopted!)

Widescreen 16:9 material can be adapted to 4:3 by cropping the image width; however, picture content is lost, and creative intent is liable to be compromised. Figures Figure 1.11 and 1.12 below show the result of centre-cropping 16:9 material. The plot might suffer!

Pan-and-scan, sketched in Figure 1.13 at the top of the facing page, refers to choosing on a scene-by-scene basis the 4:3 region to be maintained, to mitigate the creative loss that might otherwise result from cropping.

Figure 1.11 **Aspect ratio changes** can compromise creative intent. Consider this frame at 1.78:1 aspect ratio. The two figures survey the water prior to embarking on an adventure.

Figure 1.12 **When centre-cut** to 4:3 aspect ratio, one character is deleted; the story has changed. Much drama and much comedy depends upon action at the edges of the frame.

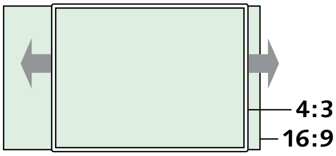


Figure 1.13 **Pan-and-scan** crops the width of widescreen material – here, 16:9 – for a 4:3 aspect ratio display.

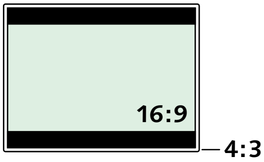


Figure 1.14 **Letterbox format** fits widescreen material – here, 16:9 – to the width of a 4:3 display.

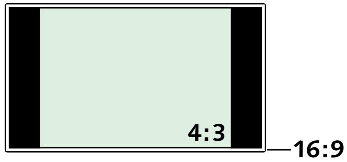


Figure 1.15 **Pillarbox format** (sometimes called *sidebar*) fits narrow-aspect-ratio material to the height of a 16:9 display.

Some consumer HD receivers have nonlinear stretching where the horizontal expansion ratio is a function of position. The intended image geometry is distorted; horizontal panning looks wonky.

Many directors and producers refuse to allow their films to be altered by cropping; consequently, many movies on DVD are released in *letterbox* format, sketched in Figure 1.14 below. In letterbox format, the entirety of the widescreen image is maintained, and the top and bottom of the 4:3 frame are unused. (Typically, either grey or black is displayed.)

Conventional 4:3 material can be adapted to 16:9 in *pillarbox* format, shown in Figure 1.15. The full height of the display is used; the left and right of the widescreen frame are blanked. However, consumer electronics (CE) manufacturers were concerned about consumers complaining about unused screen area after upconversion of SD. So, CE vendors devised schemes to stretch the image horizontally to eliminate the side panels.

The centre panel below, Figure 1.17, shows an image with correct geometry. To its left (Figure 1.16), the image is squeezed horizontally to 75%; to its right (Figure 1.18), it is stretched horizontally to 133.3%. The distortion is so blatant that you may suspect that I have

Figure 1.16 **Squeeze to $^3/_4$** is necessary if 16:9 material is crudely resized to fit a 4:3 frame.

Figure 1.17 **A normal image** of Barbara Morris is shown here for comparison.

Figure 1.18 **Stretch to $^4/_3$** is necessary if 4:3 material is crudely resized to fit 16:9.

exaggerated the effect – but the images here are distorted by exactly the amounts that would be used for SD-to-HD and HD-to-SD conversion to fit the frame width. Such shrinking and stretching is disastrous to picture integrity – but it has been commonplace since the introduction of HD to consumer television in North America. Failure of content distributors and consumer electronics manufacturers to properly respect picture aspect ratio has been, in my opinion, the most serious engineering error made in the introduction of HD systems to North America.

Frame rates

Details concerning frame rates and interlace are found in *Flicker, refresh rate, and frame rate,* on page 83.

SD broadcast television historically used interlaced scanning. In 480*i* ("NTSC") systems, a frame rate of $^{30}/_{1.001}$ Hz ("29.97 Hz") is standard; in 576*i* ("PAL") systems, a frame rate of 25 Hz is standard. The frame rates of composite NTSC and PAL video are rigid. Component video systems potentially have flexibility in the choice of frame rate. However, production and distribution infrastructure is generally locked-in to one of two frame rates, 25 Hz or 29.97 Hz. For international distribution of programming, frame-rate conversion is necessary either in the distribution infrastructure or in consumer equipment.

Frame rates have historically been chosen on a regional basis to match the prevailing AC power line frequency. Efforts were made in the 1990s to establish a single worldwide frame rate for HD; these efforts were unsuccessful. Origination and broadcasting of HD typically takes place at the prevailing power-line frequency, 50 Hz or (nominally) 60 Hz. Certain lighting units used for acquisition flash at twice the AC power line frequency (though well above the perceptual flicker sensitivity). If a camera operates at a frame rate different from the AC line frequency, such flashing is liable to "beat" with the frame rate of the camera to produce an objectionable low-frequency strobing.

With distribution of video across commodity IP networks to consumer PCs, decoding recovers the native frame rate of the program, but generally no attempt is made to synchronize the display system. Poor motion portrayal often results.

Image acquisition and presentation 2

The basic proposition of digital imaging is summarized in Figure 2.1. Image data is captured, processed, and/or recorded, then presented to a viewer. As outlined in the caption, and detailed later, appearance depends upon display and viewing conditions. Viewing ordinarily takes place in conditions different from those in effect at the time of capture of a scene. If those conditions differ, a nontrivial mapping of the captured image data – *picture rendering* – must be imposed in order to achieve faithful portrayal, to the ultimate viewer, of the appearance of the scene (as opposed to its physical stimulus).

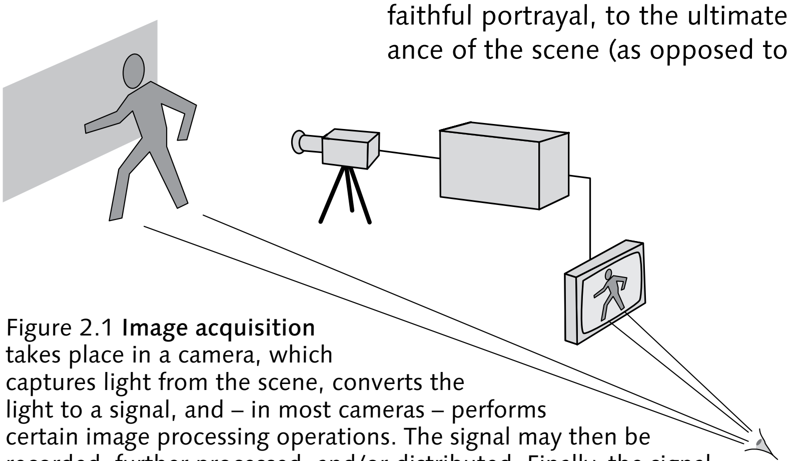


Figure 2.1 **Image acquisition** takes place in a camera, which captures light from the scene, converts the light to a signal, and – in most cameras – performs certain image processing operations. The signal may then be recorded, further processed, and/or distributed. Finally, the signal is converted to light at a display device. The appearance of the displayed image depends upon display conditions (such as peak luminance); upon viewing conditions (such as the surroundings of the display surface); and upon conditions dependent upon both the display and its environment (such as contrast ratio). It is common for the scene to be much brighter than the displayed image: The scene may be captured in daylight, with white at 30,000 $cd \cdot m^{-2}$, but a studio display produces white of just 100 $cd \cdot m^{-2}$. The usual goal of imaging is not to match the *physical* stimulus associated with the scene (say, at daylight luminance levels), but to match the viewers' expectation of the *appearance* of the scene. Producing an appearance match requires imposing a nontrivial mapping – termed *picture rendering* – that maps scene luminance to display luminance.

Examine the flowers in a garden at noon on a bright, sunny day. Look at the same garden half an hour after sunset. Physically, the spectra of the flowers have not changed, except by scaling to lower luminance levels. However, the flowers are markedly less colourful after sunset: Colourfulness decreases as luminance decreases. This is the *Hunt effect,* named after the famous colour scientist R.W.G. Hunt. Images are usually viewed at a small fraction, perhaps $^1/_{100}$ or $^1/_{1000}$, of the luminance at which they were captured. If the image is presented with luminance proportional to the scene luminance, the presented image would appear less colourful, and lower in contrast, than the original scene.

Giorgianni, Edward J., and Thomas E. Madden (2008), *Digital Color Management: Encoding Solutions,* Second Edition (Chichester, U.K.: Wiley).

To present contrast and colourfulness comparable to the original scene, we *must* alter the characteristics of the image. An engineer or physicist might strive to achieve mathematical linearity in an imaging system; however, the required alterations cause the displayed relative luminance to depart from proportionality with scene luminance. The dilemma is this: We can achieve mathematical linearity, or we can achieve correct appearance, but we cannot simultaneously achieve both! Successful commercial imaging systems sacrifice mathematics to achieve the correct perceptual result.

Image state

In many professional imaging applications, imagery is reviewed and/or approved prior to distribution. Even if the image data originated with a colorimetric link from the scene, any technical or creative decision that results in alteration of the image data will break that link. Consider the movie *Pleasantville.* Colour is used as a storytelling device. The story hinges upon characters depicted in greyscale and characters depicted in colour. (See Figure 2.2.) The *R'G'B'* values of the final movie do not accurately represent what was in front of the camera! This example is from the entertainment industry, however, examples abound whereever colour is adjusted for æsthetic purposes.

Picture rendering is ordinarily a nonlinear operation, not easily described in a simple equation or even a set of equations. Once picture rendering is imposed, its parameters aren't usually preserved. In many applications of imaging, image data is manipulated to achieve

Figure 2.2 **Colour as a dramatic device.** This image mimics the visual style of the 1998 New Line Cinema movie, *Pleasantville.* When the scene was captured, the characters in the background weren't grey; they were rotoscoped in post-production. Image data has been altered to achieve an artistic goal.

an artistic effect – for example, colours in a wedding photograph may be selectively altered by the photographer. In such cases, data concerning picture rendering is potentially as complex as the whole original image!

The design of an imaging system determines where picture rendering is imposed:

- In consumer digital photography and in video production, picture rendering is typically imposed in the camera.
- In movie making, picture rendering is typically imposed in the processing chain.

ISO 22028-1 (2004), *Photography and graphic technology– Extended colour encodings for digital image storage, manipulation and interchange.*

If an imaging system has a direct, deterministic link from luminance in the scene to image code values, in colour management terminology the image data is said to have an *image state* that is *scene referred.* If there is a direct, deterministic linkage from image code values to the luminance intended to be produced by a display, then image data is said to be *display referred*.

Video standards such as BT.709 and SMPTE ST 274 (both to be detailed later) are at best unclear and at worst wrong concerning image state. Consequently, video engineers often mistakenly believe that video data is linked colorimetrically to the scene. Users of digital still cameras may believe that their cameras capture "science"; however, when capturing TIFF or JPEG images, camera algorithms perform rendering, so the colorimetric link to the scene is broken. What is important in these applications is not the OECF that once mapped light from the scene to image data values, but rather the EOCF that is expected to map image data values to light presented to the viewer.

High-end D-SLR cameras have provisions to capture "raw" data that has *not* been subject to picture rendering operations. These cameras are capable of capturing "science."

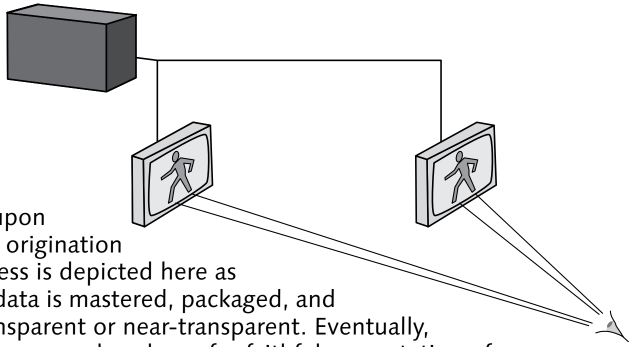


Figure 2.3 **Image approval** is based upon the display at the culmination of the origination process. (The entire origination process is depicted here as a black box.) Upon approval, image data is mastered, packaged, and distributed; these operations are transparent or near-transparent. Eventually, imagery is presented to the viewer. Image makers hope for faithful presentation of what was reviewed and approved. There is not necessarily any reference to the original scene (if indeed there was a physical scene). In principle, the viewer should be able to compare the presented image to that which was approved.

EOCF standards

In imaging systems where imagery is subject to review or approval at origination, faithful presentation requires consistent mapping from image data to light – and in entertainment applications, from audio signal to sound – between the origination environment and the ultimate viewing environment.

Figure 2.3 depicts the basic chain of origination, approval, distribution, and presentation. Origination is depicted as a "black box." The mapping from image data to displayed light involves an *electro-optical conversion function* (EOCF). It is clear from the sketch that faithful presentation requires matching EOCFs at the approval display and the presentation display. EOCF is thereby incorporated – explicitly or implicitly – in any image interchange standard. Faithful presentation also requires agreement – again, implicit or explicit – upon reference viewing conditions.

To make the most effective use of limited capacity in the "channel," the EOCFs common in commercial imaging incorporate perceptual uniformity, a topic to which we now turn.

Entertainment programming

Entertainment represents an economically important application of imaging, so it deserves special mention here. Digital video, HD, and digital cinema all involve acquisition, recording, processing, distribution, and

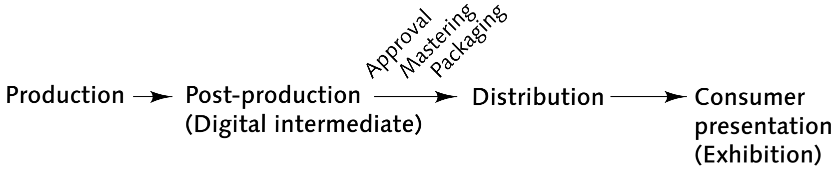


Figure 2.4 **Stages of production** are depicted. In video, the final stage is *presentation;* in cinema, it's called *exhibition.*

presentation of programs. I'll use the generic word "program" as shorthand for a movie, a television show, or a short piece such as a commercial. Figure 2.4 above presents a sketch of the entire chain.

If a movie is "in production," then principal photography is not yet complete.

Production refers to acquisition, recording, and processing. In a live action movie, the term *production* may be limited to just the acquisition of imagery (on set or on location); processes that follow are then postproduction ("post"). In the case of a movie whose visual elements are all represented digitally, post production is referred to as the *digital intermediate* process, or DI.

Production culminates with display and approval of a program on a studio reference display – or, in the case of digital cinema, approval on a cinema reference projector in a review theatre. (If distribution involves compression, then approval properly includes review of compression at the studio and decompression by a reference decompressor.) Following approval, the program is mastered, packaged, and distributed.

The word *reproduction,* taken literally, suggests production again! I propose *presentation.*

Professional content creators rarely seek to present, at the viewer's premises, an accurate representation of the scene in front of the camera. Apart from makers of documentaries, movie makers often make creative choices that alter that reality. They hope that when the program completes its journey through the distribution chain, the ultimate consumer will be presented with a faithful approximation *not* of the original scene, but rather of what the director saw on his or her studio display when he or she approved the final product of postproduction. In colour management terms, movie and video image data is display-referred.

Acquisition

A person using a camera to acquire image data from a scene expects that when the acquired material is displayed it will approximately match the appearance of the scene. Luminance of white in an outdoor scene might reach 30,000 $cd \cdot m^{-2}$, but it is rare to find an electronic display whose luminance exceeds 450 $cd \cdot m^{-2}$, and professional HD content mastering and approval is performed with a reference white around 100 $cd \cdot m^{-2}$. Linear transfer of the scene luminance to the display – in effect, scaling absolute luminance by a factor of 0.015 or 0.01 – won't present the same appearance as the outdoor scene. The person using the camera expects an approximate apppearance match upon eventual display; consequently, picture rendering must be imposed. In HD, and in consumer still photography, rendering is imposed at the camera; in digital cinema and in professional ("raw") still photography, rendering is imposed in postproduction.

Some people use the phrase "scene-to-screen" to describe the goal of delivering an accurate representation of scene luminance and colour to the display. Unless proper account is taken of appearance phenomena – that is, unless picture rendering is imposed – this effort is doomed to failure.

Consumer origination

Figure 2.5 summarizes consumer-originated video. Consumers may exercise creative control through signal processing after acquisition; however, picture rendering is imposed by algorithms in the camera. The camera is engineered to encode signals for presentation in the consumers' living room. Studio origination is built on an assumption of viewing at 100 $cd \cdot m^{-2}$ in a dark surround (today, around 1%). Consumer camcorders incorporate picture rendering based upon comparable parameters.

Consumer electronics (CE) display

In the consumer electronics domain, there is a diversity of display devices (having different contrast ratios, different peak luminance values, and different colour gamuts), and there is a diversity of viewing environ-

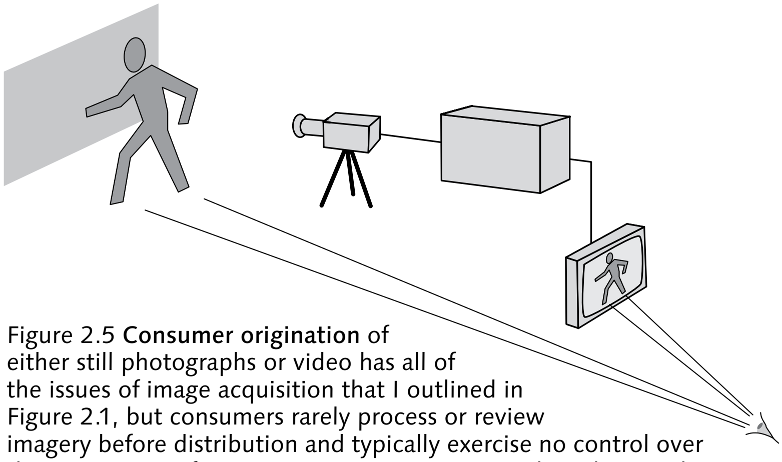


Figure 2.5 **Consumer origination** of either still photographs or video has all of the issues of image acquisition that I outlined in Figure 2.1, but consumers rarely process or review imagery before distribution and typically exercise no control over the parameters of image capture or processing. Algorithms in the camera impose picture rendering and incorporate the rendering into the image data. Those operations assume a certain display and viewing environment. That reference viewing environment is thereby incorporated (explicitly or implictly) into the image exchange standard.

ments (some bright, some dark; some having bright surround, some dim, and some dark).

Different consumer display devices have different default EOCFs. The EOCF for a particular product is preset at the factory in a manner suitable for the viewing conditions expected for that product. Traditional domestic television receivers had EOCFs approximating the 2.4-power function used in the studio; however, modern consumer receivers are considerably brighter than 100 $cd \cdot m^{-2}$ (up to 350 $cd \cdot m^{-2}$ today) and the higher brightness necessitates a somewhat lower value of gamma, today typically between 2.1 and 2.3. A home theatre projector used in a rather dark environment will have characteristics comparable to those of a studio reference display (see *Reference display and viewing conditions*, on page 427), and will typically have “gamma” of about 2.4. A PC has a default sRGB EOCF, with gamma of 2.2.

Consumer television receiver vendors commonly impose signal processing claimed to “improve” the image – often described by adjectives such as “naturalness” or “vividness.” However, the director may have thoughtful reasons for wanting the picture to look unnatural, pale, or noisy! Creative control is properly

Digital Reality Creation (DRC) is a trademark of Sony Electronics Inc.

DNIe is a trademark of Samsung Electronics Co., Ltd.

exercised at production, not at presentation. Creative staff generally despise consumer processing that goes by such names as Digital Reality Creation (DRC) or Digital Natural Image engine (DNIe).

Linear-light and perceptual uniformity 3

Each pixel value in a greyscale image represents what is loosely called *brightness*. However, brightness is defined formally as *the attribute of* a *visual sensation according to which an area appears to emit more or less light*. This definition is obviously subjective: *Brightness* can't be measured, so is an inappropriate metric for image data. Also, according to colour appearance theory, brightness has no top end: Brightness is not relative to anything.

See Appendix B, *Introduction to radiometry and photometry,* on page 601. Sound intensity is conceptually very different from light intensity.

Intensity is radiant power in a particular direction, that is, power per unit solid angle [W·sr^{-2}]; *radiance* is intensity per unit projected area. These terms disregard wavelength composition, but in colour imaging, wavelength is important! Neither of these quantities is a suitable metric for colour image data.

Old texts use "brightness," symbolized *B,* where today we would use (absolute) *luminance*, symbolized *L.* In image reproduction, we are usually concerned not with (absolute) luminance, but with *relative luminance*, symbolized *Y,* as I will describe on page 206. The term *luminance* is often carelessly and incorrectly used by video engineers to refer to *luma,* as I will describe.

Luminance is radiance weighted by the spectral sensitivity associated with the lightness sensation of vision. Luminance is proportional to intensity; in the SI system, it carries units of candelas per meter squared [cd·m^{-2}], commonly called *nits* [nt]. Imaging systems rarely use pixel values proportional to luminance; values nonlinearly related to luminance are usually used.

Illuminance describes light falling on an object; technically, it is luminance integrated over a half-sphere.

Lightness is defined by the CIE as *the brightness of an area judged relative to the brightness of* a *similarly illuminated area that appears to be white or highly transmitting*. A purist may claim this definition to be subjective; however, an objective quantity L^* is defined as the standard estimate of the perceptual response to relative luminance. It is computed by subjecting relative luminance to a nonlinear transfer function that mimics the response of vision at a certain state of adaptation. A few greyscale imaging systems have pixel values proportional to L^*.

Value refers to measures of lightness roughly equivalent to CIE L^* (but typically scaled 0 to 10, not 100). In image science, *value* is rarely – if ever – used in any sense consistent with accurate colour. (Several different value scales are graphed in Figure 20.2 on page 208.)

Regrettably, many practitioners of digital image processing and computer graphics have a cavalier attitude toward the terms described here. In the *HSB*, *HSI*, *HSL*, and *HSV* systems, *B* allegedly stands for brightness, *I* for intensity, *L* for lightness, and *V* for value; however, none of these systems is associated with brightness, intensity, luminance, or value according to any definition that is recognized in colour science!

Colour images are sensed and reproduced based upon *tristimulus values* (*tristimuli*), whose amplitudes are proportional to intensity but whose spectral compositions are carefully chosen according to the principles of colour science. Relative luminance can be regarded as a distinguished tristimulus value useful on its own; apart from that, tristimulus values come in sets of three.

The image sensor of a digital camera produces values, proportional to radiance, that approximate red, green, and blue (*RGB*) tristimulus values. I call these values *linear-light*. However, in most imaging systems, *RGB* tristimulus values are coded using a nonlinear transfer function – *gamma correction* – that mimics the perceptual response of human vision. Most image coding systems use *R'G'B'* values that are *not* proportional to intensity; the primes in the notation denote imposition of a perceptually motivated nonlinearity.

Luma is comparable to lightness; it is often carelessly and incorrectly called *luminance* by video engineers. See Appendix A, *YUV and luminance considered harmful,* on page 567.

Luma (*Y'*) is formed as a suitably weighted sum of *R'G'B'*; it is the basis of luma/colour difference coding in video, MPEG, JPEG, and similar image coding systems. In video, the nonlinearity implicit in gamma correction that forms *R'G'B'* components is subsequently incorporated into the luma and chroma ($Y'C_BC_R$) components.

Contrast

The term *contrast,* as used in imaging, is heavily overloaded. "Contrast" can relate to physical properties of light in a scene, physical properties of light in an image, properties of an imaging system's mapping between the two, physical properties of a display system (independent of any image), or various attributes of perception.

The histogram computed and displayed by a D-SLR camera typically has luma as the independent axis, not luminance or log luminance.

Photographers speak of scenes having "high contrast" or "low contrast." A scene's contrast can be characterized by the coefficient of variation of its luminance – that is, by the standard deviation of its luminance divided by its mean luminance. Scene contrast is closely related to the width of the scene's luminance histogram with respect to its "centre of gravity." A dark scene and a light scene may have identical scene contrast. Scene contrast is unaffected by noise.

The term *key* used by photographers is unrelated to *key* used in video to refer to matting.

Photographers also speak of "high key" and "low key" scenes. Loosely speaking, these are predominantly light or dark (respectively). From a cumulative histogram of log luminance, key can be quantified as the fraction of the interval along the *x*-axis between the 10^{th} and 90^{th} percentile where the 50^{th} percentile falls.

Contrast ratio

A basic property of a display is its *contrast ratio,* the ratio between specified light and dark luminances – typically the luminance associated with reference (or peak) white and the luminance associated with reference black. *Inter-image* (or *on-off,* or *sequential*) contrast ratio is measured between fully-white and fully-black images presented individually. Intra-image contrast ratio is measured from white and black regions of a single test image such as that specified by ANSI IT7.228-1997 (now withdrawn, for projectors) or ITU-R BT.815. Visual performance of a display system is best characterized by intra-image contrast ratio.

Simultaneous contrast ratio is sometimes shortened to *simultaneous contrast*, which unfortunately has a second (unrelated) meaning. See *Surround effect,* on page 116. Owing to this ambiguity, I avoid the term *simultaneous contrast ratio* and use the term *intra-image contrast ratio* instead. Contrast ratio without qualification should be taken as intra-image.

BT.815 specifies a test image, having 16:9 aspect ratio, useful for measuring contrast ratio ;see Figure 3.1. Contrast ratio is the luminance of the white square divided by the average luminance of the black squares.

In practical imaging systems, many factors conspire to increase the luminance of black, thereby lessening contrast ratio and impairing picture quality. On an electronic display or in a projected image, intra-image contrast ratio is typically less than 800:1 owing to flare in the display system and/or spill light (stray light) in the ambient environment. Typical contrast ratios are shown in Table 3.1. Contrast ratio is a major determi-

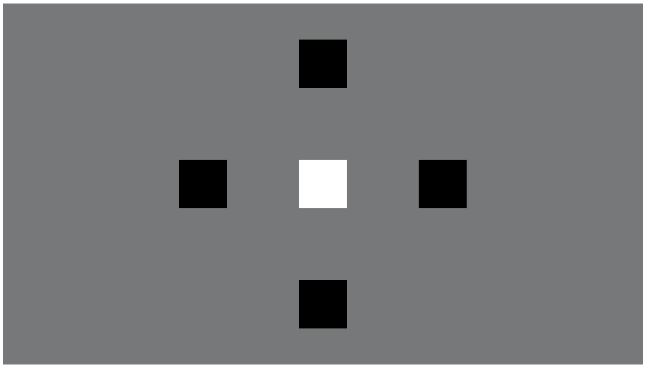


Figure 3.1 **The ITU-R BT.815 pattern** is used to characterize intra-image contrast ratio. The grey background is at 8-bit interface code 126. A small square of reference white lies at the centre of the image; it occupies $^2/_{15}$ of the image height. Four identical small squares of reference black are disposed around the centre, offset by $^1/_3$ of the image height, at 12 o'clock, 3 o'clock, 6 o'clock, and 9 o'clock.

Application	*Max. (Ref.) luminance* [cd·m^{-2}]	*Typ. PDR luminance* [cd·m^{-2}]	*Typical inter-image contrast ratio*	*Typical BT.815 intra-image contrast ratio*	*Minimum L**
Cinema (projection)	48 (48)	24	1 000:1	100:1	9
HD, studio mastering	120 (100)	90	10 000:1	1000:1	0.9
HD, living room (typ.)	200	200	1 000:1	400:1	2.3
Office (sRGB, typ.)	320	320	100:1	100:1	16

Table 3.1 **Typical contrast ratios** are summarized. PDR refers to perfect diffuse reflector.

nant of subjective image quality, so much so that an image reproduced with a high simultaneous contrast ratio may be judged sharper than another image that has higher measured spatial frequency content.

Because contrast ratio involves measurement of just two quantities – reference white and reference black – it is independent of transfer function. Contrast ratio (and its first cousin, dynamic range) say nothing about bit depth: A bilevel image contains both reference white and reference black, and suffices to produce the optical stimulus necessary to measure contrast ratio.

Perceptual uniformity

I introduced perceptual uniformity on page 8. Coding of a perceptual quantity is *perceptually uniform* if a small perturbation to the coded value is approximately equally perceptible across the range of that value. Vision cannot distinguish two luminance levels if the ratio between them is less than about 1.01 – in other words, the visual threshold for luminance difference is about 1%. This *contrast sensitivity* threshold is established by experiments using the test pattern such as the

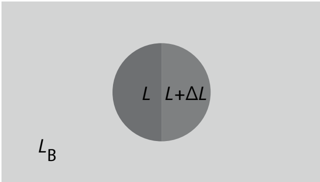


Figure 3.2 **A contrast sensitivity test pattern** reveals that a difference in luminance will be observed in certain conditions when Δ*L* exceeds about 1% of *L*. This threshold is called a *just-noticeable difference* (JND).

one sketched in Figure 3.2 in the margin; details will be presented in *Contrast sensitivity,* on page 249. Ideally, pixel values are placed at these *just noticeable difference* (JND) increments along the scale from reference black to reference white.

The "code 100" problem and nonlinear image coding

Consider 8-bit pixel values proportional to luminance, where code zero represents black, and the maximum code value of 255 represents white, as in Figure 3.3 below. Code 100 lies at the point on the scale where the ratio between adjacent luminance values is 1%: Owing to the approximate 1% contrast threshold of vision, the boundary between a region of code-100 samples and a region of code-101 samples is liable to be visible.

As pixel value decreases below 100, the difference in luminance between adjacent codes becomes increasingly perceptible: At code 20, the ratio between adjacent luminance values is 5%. In a large area of smoothly

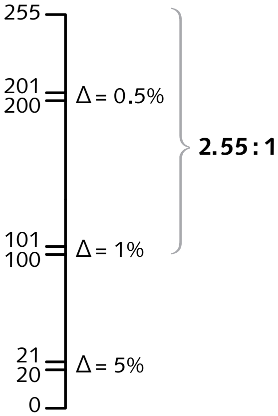


Figure 3.3 **The "code 100" problem** with linear-light coding is that for code levels below 100 the steps between code values have ratios larger than the visual threshold: With just 256 steps, some steps are liable to be visible as banding.

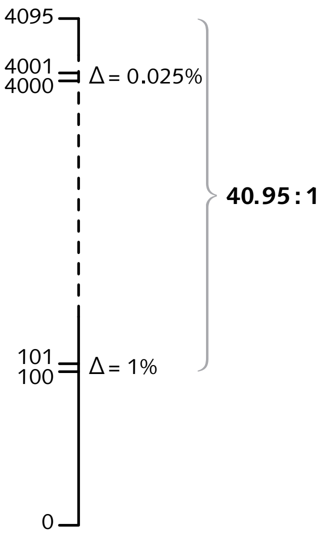


Figure 3.4 **The "code 100" problem is mitigated** by using more than 8 bits to represent luminance. Here, 12 bits are used, placing the top end of the scale at 4095. However, the majority of these 4096 codes cannot be distinguished visually.

varying shades of grey, these luminance differences are likely to be visible or even objectionable. Visible jumps in luminance produce *contouring* or *banding* artifacts.

Linear-light codes above 100 suffer no banding artifacts. However, as code value increases toward white, the codes have decreasing perceptual utility: At code 200, the luminance ratio between adjacent codes is just 0.5%, below the threshold of visibility. Codes 200 and 201 are visually indistinguishable; code 201 could be discarded without its absence being noticed.

High-quality image display requires a contrast ratio of at least 30 to 1 between the luminance of white and the luminance of black. In 8-bit linear-light coding, the ratio between the brightest luminance (code 255) and the darkest luminance that can be reproduced without banding (code 100) is only 2.55:1. Linear-light coding in 8 bits is therefore unsuitable for high-quality images.

One way to mitigate the "code 100" problem is to place the top end of the scale at a code value much higher than 100, as sketched in Figure 3.4. If luminance is represented in 12 bits, with white at code 4095, the luminance ratio between code 100 and white reaches 40.95:1. However, the vast majority of those 4096 code values cannot be distinguished visually; for example, codes 4000 through 4039 are visually indistinguishable. Rather than coding luminance linearly with a large number of bits, we can use many fewer code values, and assign them on a perceptual scale that is nonlinear with respect to light power. Such nonlinear-light – or *perceptually uniform* – coding is pervasive in digital imaging; it works so beautifully that many practitioners don't even know it's happening.

In video (including motion-JPEG, MPEG, and H.264/AVC), and in digital photography (including JPEG/JFIF/Exif), *R'G'B'* components are coded in a perceptually uniform manner. A video display has a nonlinear transfer function from code value to luminance. That function is comparable to the function graphed in Figure 1.7 on page 8, and approximates the inverse of the lightness sensitivity of human vision. Perceptual uniformity is effected by applying a nonlinear transfer function – *gamma correction* – to each tristimulus estimate sensed from the scene. Gamma correction parameters are chosen – by default, manually, or automatically – such that the intended image appearance is obtained on the reference display in the reference viewing environment. Gamma correction thereby incorporates both considerations of perceptual uniformity and considerations of picture rendering.

An example of 8-bit quantization as commonly used in computing is shown in the right-hand sketch of Figure 4.1, on page 37.

A truecolour image in computing is usually represented in *R'G'B'* components of 8 bits each, as I will explain further on page 36. Each component ranges from 0 (black) through 255 (white). Greyscale and truecolour data in computing is usually coded so as to exhibit approximate perceptual uniformity: The steps are not proportional to intensity, but are instead uniformly spaced perceptually. The number of steps required depends upon properties of the display system, of the viewing environment, and of perception.

Reference display and viewing parameters for studio video have historically been either poorly standardized or not standardized at all. In *Reference display and viewing conditions,* on page 427, I summarize a set of realistic parameters.

Video standards such as BT.709 and SMPTE ST 274 have historically established standard *opto-electronic conversion functions* (OECFs), as if video had scene-referred image state and as if cameras had no adjustments! In fact, video is effectively *output* (display) referred, and what matters most is not the OECF but the *electro-optical conversion function,* EOCF! Nonlinear coding is the central topic of Chapter 27, *Gamma,* on page 315; that chapter discusses OECF and EOCF.

$$\frac{\log 100}{\log 1.01} \approx 462; \quad 1.01^{462} \approx 100$$

If the threshold of vision behaved strictly according to the 1% relationship across the whole tone scale, then luminance could be coded logarithmically. For a contrast ratio of 100:1, about 462 code values would be required, corresponding to about 9 bits.

Conversely, display *R'G'B'* values are proportional to displayed luminance raised to approximately the 0.42-power.-See *Gamma,* on page 315.

$$\frac{\left(\frac{\mathbf{255}}{255}\right)^{2.4}}{\left(\frac{\mathbf{254}}{255}\right)^{2.4}} \approx 1.01$$

For reasons to be explained in *Luminance and lightness,* on page 255, instead of modelling the lightness sensitivity of vision as a logarithmic function, in most digital imaging systems we model it as a power function. The luminance of the red, green, or blue primary light produced by a display is made proportional to code value raised to approximately the 2.4-power. The equation in the margin computes the ratio of linear-light value between the highest 8-bit code value (255) and the next-lowest value (254): The 1.01-requirement of human vision is met. As code values decrease, the ratio gets larger; however, as luminance decreases the luminance ratio required to maintain the "JND" of vision gets larger. The 2.4-power function turns out to be a very good match to the perceptual requirement.

A related claim is that 8-bit imaging has an optical density range of about 2.4, where 2.4 is the base-10 log of $^1/_{255}$. This claim similarly rests upon linear-light coding.

$$\left(\frac{1}{255}\right)^{2.4} \approx 0.0000016$$

Eight-bit imaging systems are often claimed to have "dynamic range" of 255:1 or 256:1. Such claims arise from the assumption that image data codes are linearly related to light. However, most 8-bit image data is coded perceptually, like sRGB, assuming a 2.2- or 2.4-power function at the display: For an ideal display, the dynamic range associated with code 1 would be close to a million to one. In practice, physical parameters of the display and its environment limit the dynamic range.

The cathode ray tube (CRT) was, for many decades, the dominant display device for television receivers and for desktop computers. Amazingly, a CRT exhibits an EOCF that is very nearly the inverse of vision's lightness sensitivity! The nonlinear lightness response of vision and the power function intrinsic to a CRT combine to cause the display code value (historically, voltage) to exhibit perceptual uniformity, as demonstrated in Figures 3.5 and 3.6 (opposite). The CRT's characteristics were the basis for perceptual uniformity for the first half century of electronic imaging. Most modern display devices – such as LCDs, PDPs, and DLPs – do not have a native, physical power function like that of a CRT; however, signal processing circuits impose whatever transfer function is necessary to make the device behave as if it had a 2.2- or 2.4-power function from signal value to displayed tristimulus value.

See *Bit depth requirements*, on page 326.

In video, this perceptually uniform relationship is exploited by *gamma correction* circuitry incorporated into every video camera. The *R'G'B'* values that result from gamma correction – the values that are processed, recorded, and distributed in video – are roughly proportional to the 0.42-power of the intended display luminance: *R'G'B'* values are nearly perceptually uniform. Perceptual uniformity allows as few as 8 bits to be used for each *R'G'B'* component. Without perceptual uniformity, each component would need 11 bits or more. Image coding standards for digital still cameras – for example, JPEG/Exif – adopt a similar approach.

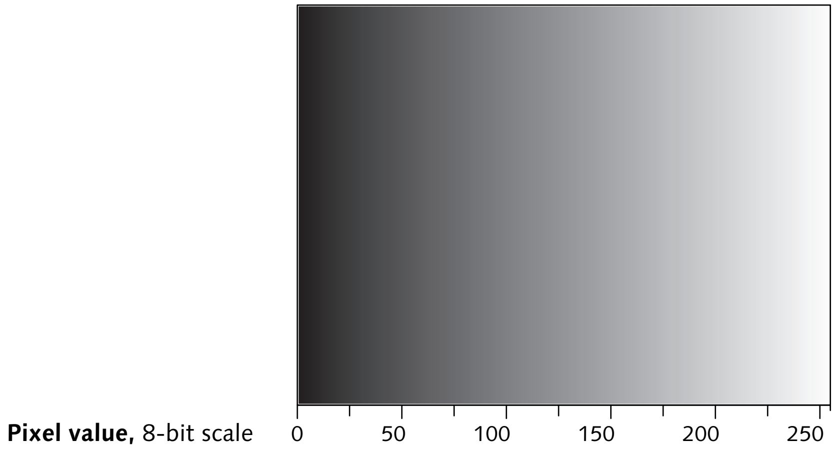


Figure 3.5 **A greyscale ramp** on a CRT display is generated by writing successive integer values 0 through 255 into the columns of a framebuffer. When processed by a digital-to-analog converter (DAC), and presented to a CRT display, a perceptually uniform sweep of lightness results. A naive experimenter might conclude – mistakenly! – that code values are proportional to intensity.

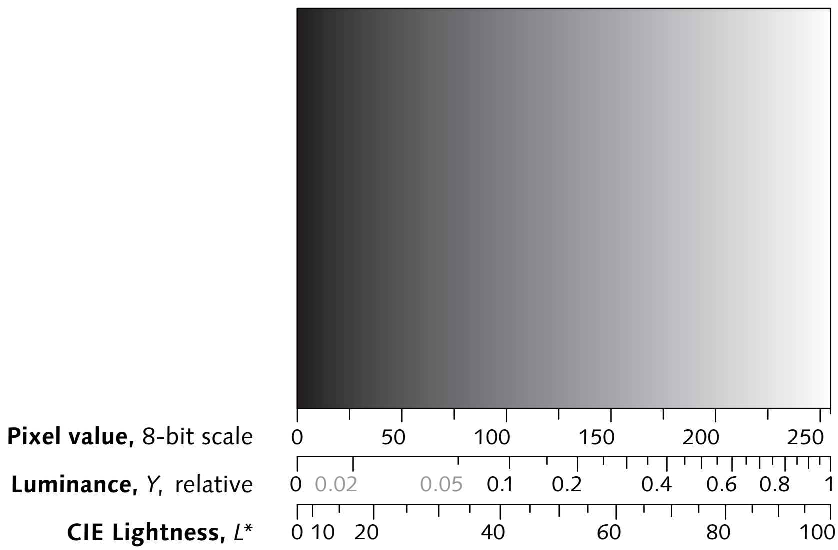


Figure 3.6 **A greyscale ramp, augmented** with CIE relative luminance (*Y*, proportional to intensity, on the middle scale), and CIE lightness (*L**, on the bottom scale). The point midway across the screen has lightness value midway between black and white. There is a near-linear relationship between code value and lightness. However, luminance at the midway point is only about 18% of white! Luminance produced by a CRT is approximately proportional to the 2.4-power of code value. Lightness is approximately the 0.42-power of luminance. Amazingly, these relationships are near inverses. Their near-perfect cancellation has led many workers in video, computer graphics, and digital image processing to misinterpret the term *intensity*, and to underestimate – or remain ignorant of – the importance of nonlinear transfer functions.

Linear and nonlinear

Modern image sensor devices such as CCD and CMOS sensors effectively convert photons to electrons: They produce signals whose amplitude is proportional to physical intensity; they are said to be linear.

Video signals were historically processed through analog circuits that had linear response to voltage. Today's digital video systems are often linear with respect to the arithmetic performed on the pixel values. Video systems are said to be linear.

Linear is an adjective, not a noun!

However, linearity in one domain cannot be carried across to another domain if a nonlinear function separates the two. In video, scene luminance is in a linear optical domain, and the video signal is subject to linear operations in the electrical signal domain. However, OECF and EOCF functions are imposed between the domains. Luminance and pixel value are therefore *not* linearly related. When you ask an optical engineer if her system is linear, she will say, "Of course!" – referring to intensity, radiance, or luminance. When you ask a video engineer if his system is linear, he will say, "Of course!" – referring to arithmetic on pixel values. However, a nonlinear transform lies between the two systems: A linear operation performed in one domain is not linear in the other.

If your computation involves perception, nonlinear representation may be required. If you perform a discrete cosine transform (DCT) on image data as part of image compression, as in JPEG, you should use coding that exhibits perceptual uniformity, in order to minimize the perceptibility of the errors that will be introduced by the coding process.

Quantization 4

Resolution properly refers to spatial phenomena; see page 65. In my view, it is a mistake to refer to a sample as having "8-bit resolution": Say *quantization* or *precision* instead. To make a 100-foot-long fence with fence posts every 10 feet, you need 11 posts, not 10! Take care to distinguish *levels* (in the left-hand portion of Figure 4.1, 11) from *steps* or *risers* (here, 10).

A signal whose amplitude takes a range of continuous values is *quantized* by assigning to each of several (or several hundred or several thousand) intervals of amplitude a discrete, numbered level. In *uniform quantization*, the *steps* between levels have equal amplitude. Quantization discards signal information lying between quantizer levels. Quantizer performance is characterized by the extent of this loss. Figure 4.1 shows, on the left, the transfer function of a uniform quantizer.

Linearity

Electronic systems are often expected to satisfy the *principle of superposition;* in other words, they are expected to exhibit *linearity*. A system g is linear *if and only if* (iff) it satisfies both of these conditions:

Eq 4.1

$$g(a \cdot x) \equiv a \cdot g(x) \qquad \text{[for scalar } a\text{]}$$

$$g(x + y) \equiv g(x) + g(y)$$

The function g can encompass an entire system: A system is linear iff the sum of the individual responses of the system to any two signals is identical to its response to the sum of the two. Linearity can pertain to

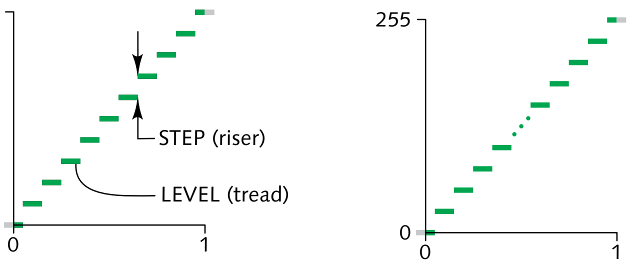


Figure 4.1 **A Quantizer transfer function** is shown on the left. The usual 0 to 255 range of quantized *R′G′B′* components in computing is sketched on the right.

steady-state response, or to the system's temporal response to a changing signal.

Linearity is a very important property in mathematics, signal processing, and video. Many electronic systems operate in the linear intensity domain, and use signals that directly represent physical quantities. One example is compact audio disc (CD) coding: *Sound pressure level* (SPL), proportional to physical intensity, is quantized linearly into 16-bit samples.

Human perception, though, is nonlinear, and in applications where perceptual quantities are being encoded or transmitted, the perceptual nonlinearity can be exploited to achieve coding more efficient than coding the raw physical quantity. For example, audio for digital telephony is nonlinearly coded using just 8 bits per sample. (Two coding laws are in use, A-law and μ-law; both of these involve decoder transfer functions that are comparable to bipolar versions of Figure 4.1.) Image signals that are captured, recorded, processed, or transmitted can similarly be coded in a nonlinear, perceptually uniform manner in order to optimize perceptual performance.

Bellamy, John C. (2000), *Digital Telephony*, Third Edition (New York: Wiley): 98–111 and 472–476.

Decibels

In the following sections, I will describe signal amplitude, noise amplitude, and the ratio between these – the *signal to noise ratio* (SNR). In engineering, ratios such as SNR are usually expressed in logarithmic units. A power ratio of 10:1 is defined as a *bel* (B), in honour of Alexander Graham Bell. A more practical measure is one-tenth of a bel – a *decibel* (dB), which represents a power ratio of $10^{0.1}$, or about 1.259. The ratio (expressed in decibels) of a power P_1 to a power P_2 is given by Equation 4.2. Signal power is often given with respect to a reference power P_{REF}, which must either be specified (often as a letter following dB) or implied by the context; the computation is expressed in Equation 4.3. An increase of 3 dB in power represents very nearly a doubling of power ($10^{0.3}$ = 1.995). An increase of +10 dB multiplies power exactly tenfold; a change of –10 dB reduces power to a tenth.

Eq 4.2 Power ratio, in decibels:

$$m = 10 \log_{10} \frac{P_1}{P_2} \quad \text{(dB)}$$

Eq 4.3 Power ratio, with respect to a reference power:

$$m = 10 \log_{10} \frac{P}{P_{REF}} \quad \text{(dB)}$$

Consider a cable conveying a 100 MHz radio frequency signal. After 100 m of cable, power has diminished to some fraction, perhaps ⅛, of its original

Eq 4.4 Power ratio, in decibels, as a function of voltage:

$$m = 20 \log_{10} \frac{V_1}{V_2} \quad \text{(dB)}$$

Voltage ratio	*Decibels*
10	20 dB
2	6 dB
1.112	1 dB
1.0116	0.1 dB
1	0 dB
0.5	–6 dB
0.1	–20 dB
0.01	–40 dB
0.001	–60 dB

Table 4.1 **Decibel examples**

value. After another 100 m, power will be reduced by the same fraction again. Rather than expressing this cable attenuation as a unitless fraction 0.125 per 100 m, we express it as 9 dB per 100 m; power at the end of 1 km of cable is -90 dB referenced to the source power.

The decibel is defined as a power ratio. If a voltage source is applied to a constant impedance, and the voltage is doubled, current doubles as well, so power increases by a factor of four. More generally, if voltage (or current) into a constant impedance changes by a ratio r, power changes by the ratio r^2. (The log of r^2 is 2 log r.) To compute decibels from a voltage ratio, use Equation 4.4. In digital signal processing (DSP), digital code levels are treated equivalently to voltage; the decibel in DSP is based upon voltage ratios. In historical analog systems it was common to use a reference of 1 mV (dBmV); in digital systems, the reference is usually the "full scale" range from reference black to reference white (dB_{FS}), equivalent to 219 codes at 8-bit interface levels. Beware: Historical 8-bit computer graphics processed 8-bit signals with no footroom and no headroom, and that practice found its way into PSNR calculations in the MPEG community, where it is common to have full scale interpreted as 0–255 instead of 0–219.

Table 4.1 gives numerical examples of decibels used for voltage ratios.

A 2:1 ratio of frequencies is an *octave,* referring to the eight whole tones in music, *do, re, me, fa, sol, la, ti, do,* that cover a 2:1 range of frequency. When voltage halves with each doubling in frequency, an electronics engineer refers to this as a loss of *6 dB per octave*. If voltage halves with each doubling, then it is reduced to one-tenth at ten times the frequency; a 10:1 ratio of quantities is a *decade*, so 6 dB/octave is equivalent to 20 dB/decade. (The base-2 log of 10 is very nearly $^{20}/_{6}$.)

In photography, a stop is taken to be a ratio of 2. For scientific and engineering purposes it is more convenient to define a stop as exactly three tenths of a density unit, that is, $10^{0.3}$, or about 1.995.

A *stop* in photography is a 2:1 ratio of light power. As mentioned above, a decibel is a power ratio of $10^{0.1}$, or about 1.259. Sensor and camera engineers prefer to use units that are equivalent between the optical and electrical domains: They treat digital code level as signal (like voltage), and they describe an optical power of 2 as 6 dB. It is a numerological coincidence that $10^{0.3}$ is very nearly equal to 2; so 6 dB corresponds to one stop, and 2 dB corresponds to $^1/_3$ stop.

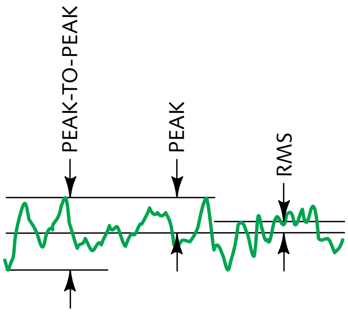


Figure 4.2 **Peak-to-peak, peak, and RMS values** are measured as the total excursion, half the total excursion, and the square root of the average of squared values, respectively. Here, a noise component is shown.

Noise, signal, sensitivity

Analog electronic systems are inevitably subject to noise introduced from thermal and other sources. Thermal noise is unrelated to the signal being processed. A system may also be subject to external sources of interference. As signal amplitude decreases, noise and interference make a larger relative contribution.

Processing, recording, and transmission may introduce noise that is uncorrelated to the signal. In addition, *distortion* that is correlated to the signal may be introduced. As it pertains to objective measurement of the performance of a system, distortion is treated like noise; however, a given amount of distortion may be more or less perceptible than the same amount of noise. Distortion that can be attributed to a particular process is known as an *artifact*, particularly if it has a distinctive perceptual effect.

In video, *signal-to-noise ratio* (SNR) is the ratio of the peak-to-peak amplitude of a specified signal, often the reference amplitude or the largest amplitude that can be carried by a system, to the root mean square (RMS) magnitude of undesired components including noise and distortion. (It is sometimes called PSNR, to emphasize *peak* signal; see Figure 4.2.) SNR is expressed in units of decibels. In many fields, such as audio, SNR is specified or measured in a physical (intensity) domain. In video, SNR usually applies to gamma-corrected components *R'*, *G'*, *B'*, or *Y'* that are in the perceptual domain; so, SNR correlates with perceptual performance.

Sensitivity refers to the minimum source power that achieves acceptable (or specified) SNR performance.

Quantization error

A quantized signal takes only discrete, predetermined levels: Compared to the original continuous signal, *quantization error* has been introduced. This error is correlated with the signal, and is properly called *distortion*. However, classical signal theory deals with the addition of noise to signals. Providing each quantizer step is small compared to signal amplitude, we can consider the loss of signal in a quantizer as addition of an equivalent amount of noise instead: Quantization

Eq 4.5 Theoretical SNR limit for a *k*-step quantizer:

$$20 \log_{10}\left(k\sqrt{12}\right)$$

The factor of root-12, about 11 dB, accounts for the ratio between peak-to-peak and RMS; for details, see Schreiber (cited below).

Some people use the word *dither* to refer to the technique of adding noise prior to quantization; other people restrict the term *dither to* schemes that involve spatial distribution of the noise. The technique was first described by Roberts, Lawrence G. (1962), "Picture coding using pseudorandom noise," *IRE Trans.* **IT-8** (2): 145–154. Roberts' work is summarized in Schreiber, William F. (1993), *Fundamentals of Electronic Imaging Systems,* Third Edition (Berlin: Springer-Verlag).

diminishes signal-to-noise ratio. The theoretical SNR limit of a *k*-step quantizer is given by Equation 4.5. Eight-bit quantization, common in consumer video, has a theoretical SNR limit (peak-to-peak signal to RMS noise) of about 56 dB.

If an analog signal has very little noise, then its quantized value can be nearly exact when near a step, but can exhibit an error of nearly ±½ of a step when the analog signal is midway between quantized levels. In video, this situation can cause the reproduced image to exhibit *noise modulation*. It is beneficial to introduce, prior to quantization, roughly ±½ of a quantizer step's worth of high-frequency random or pseudorandom noise to avoid this effect. This introduces a little noise into the picture, but this noise is less visible than low-frequency "patterning" of the quantization that would be liable to result without it. SNR is slightly degraded, but subjective picture quality is improved. Historically, video digitizers implicitly assumed that the input signal itself arrived with sufficient analog noise to perform this function; nowadays, analog noise levels are lower, and the noise should be added explicitly at the digitizer.

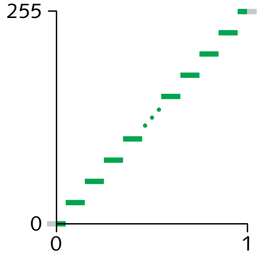
The degree to which noise in a video signal is visible – or objectionable – depends upon the properties of vision. To minimize noise visibility, we digitize a signal that is a carefully chosen nonlinear function of luminance (or tristimulus values). The function is chosen so that a given amount of noise is approximately equally perceptible across the whole tone scale from black to white. This concept was outlined in *Nonlinear image coding,* on page 12; in the sections to follow, linearity and perceptual uniformity are elaborated.

Full-swing

Excursion (or colloquially, *swing*) refers to the range of a signal – the difference between its maximum and minimum levels. In video, reference excursion is the range between standardized *reference white* and *reference black* signal levels.

Computer graphics image data ordinarily has *full-swing* (or *full-range*), where reference black is assigned to the lowest code level and reference white is assigned to the highest code level. In desktop graphics, *R'G'B'*

Figure 4.3 **Full-swing 8-bit quantization** is depicted in this sketch. A nominally continuous input signal, here represented by values ranging 0 to 1 on the *x*-axis, is quantized to 8 bits – that is, 256 steps – for storage or transmission. The coding provides no footroom or headroom.



components typically have 8 bits each and range 0 through 255. Figure 4.3 depicts 8-bit full-swing coding.

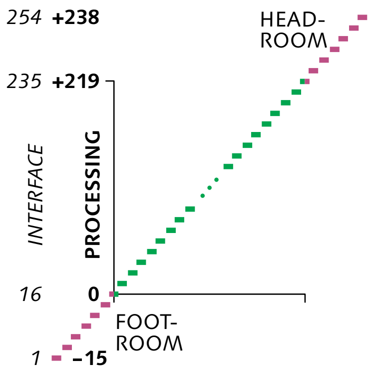
Studio-swing (footroom and headroom)

In high-quality video, it is necessary to preserve transient signal excursions below black and above white that are liable to result from processing by digital and analog filters. Studio video standards therefore provide footroom below reference black, and headroom above reference white. Since headroom allows code values that exceed reference white, *reference* white and *peak* white refer to different signal levels.

When the BT.601 standard was established in 1984, an engineering error was made. Luma should have been assigned interface codes from 16 to 240 to match the chroma excursion. Instead, the range 16 to 235 was chosen, for no good reason.

I have described signals in the abstract range 0 to 1. When *R′G′B′* or *Y′* components are interfaced in 8 bits, the 0 to 1 values are scaled by 219 and offset by +16. Eight-bit studio standards thus have 219 steps between reference black and reference white. Interface codes below 16 and above 235 are used for footroom and headroom. Unfortunately, footroom and headroom are not symmetrical. Figure 4.4 below shows the standard coding range for 8-bit *R′*, *G′*, or *B′*, or luma. Codes

Figure 4.4 **Footroom and headroom** is provided in digital video standards. For processing, reference black is considered to lie at code 0; in an 8-bit system, *R′*, *G′*, *B′*, and luma (*Y′*) range 0 through reference white at code 219. At an 8-bit interface according to BT.601, an offset of +16 is added (indicated in italics). Interface codes 0 and 255 are reserved for synchronization; those codes are prohibited in video data.



having the 8 most-significant bits (MSBs) all-zero or all-one 0 and 255 are used for synchronization; these codes are prohibited within video data.

Historically, there were essentially two reasons for footroom and headroom:

- It was necessary to accommodate video whose analog levels were misadjusted.
- It was very useful to preserve filter transients (undershoot and overshoot) through as much of the system as possible (even though undershoots would eventually be clipped at the display).

As use of analog decreased, the first reason vanished. The second remained important, and three additional reasons emerged:

PLUGE: Picture line-up generator. See page 421.

- The footroom region enables conveyance of the PLUGE pattern used to set black level.
- When black is set correctly, reference black level produces luminance that is visually indistinguishable from reference black, but still nonzero. A pure 2.4-power function EOCF produces relative luminance of around 0.0003 at this point. The idealized, theoretical zero luminance occurs at a level below reference black. That level is coded in the footroom region.
- Image sensors produce noise, including noise at black. If footroom is provided, the noise averages to black. If lack of footroom clips the negative-going excursion of the noise, then the noise average rises. Provision of footroom prevents *noise modulation.*

So, the first reason for footroom is archaic, but the other four remain important.

Program creators routinely hire *quality control* (QC) contractors to vet finished programs. QC contractors have and come to believe, mistakenly, that excursions outside the 0-to-1 range are "illegal" and should be rejected. Many studios now thoughtlessly believe their QC houses instead of developing their own understanding of the issue.

Concerning headroom, reference black was originally established at a level somewhat below peak white to accommodate analog misadjustment. As digitization prevailed, that reason vanished. But reference white stands firm for digital video, and with a pure 2.4-power EOCF, peak white lies at relative luminance of $({}^{955}/_{876})^{2.4}$, or about 1.23. A studio reference display should follow the 2.4-curve all the way to peak white. The consumer electronic industry is always keen to maximize average brightness, and that goal is routinely aided by their flattening the EOCF at the high end. The working assumption in program creation is that CE equipment will reasonably reliably display levels up to reference white, then cheat.

In interfaces having k bits ($8 \le k$), reference black and white levels are multiplied by 2^{k-8}. For example, when *R'G'B'* or *Y'* components occupy 10 bits, the 0 to 1 range is scaled by 219·4 (i.e., to 0 to 876); the interface offset is +64 instead of +16.

In consumer terminology, the footroom region is often called *blacker-than-black* (BTB), and the headroom region, *whiter-than-white* (WTW).

The terms *superblack* and *superwhite* are used in the consumer arena to refer to excursions into the footroom and headroom regions respectively. In some consumer gear (e.g., PS3), *Superwhite* is an option that enables an interface to carry footroom and headroom codes; when *not* set, interface codes are clipped, and the quality of studio-standard material is liable to suffer.

I use the term *studio-swing* for the levels used in studio equipment; I use the term *full-swing* for coding that places the reference levels at the interface extremes, leaving no room for footroom or headroom. Historically, desktop graphics coding had full-swing (between 0 and 255). A realtime HD-SDI interface prohibits codes 0 and 255, or in a 10-bit system, codes 0–3 and codes 1020–1023. If full-swing coding is to be used across an HD-SDI interface, then codes 0–3 and codes 1020–1023 must be avoided.

In digital cinema distribution according to DCI/SMPTE standards (such as SMPTE ST 431), encoding is standardized without reference to prohibited code. However, standard HD-SDI interfaces are typically used and the codes 0–3 and 1020–1023 are prohibited. Consequently, a 10-bit interface clips to the range 4–1019 and the projector never sees codes outside that range. (In the 12-bit variants of HD-SDI, codes 0–16 and 4080–4095 are prohibited, and not seen by the projector.) The clipping is inconsequential in practice.

In digital cinema acquisition and postproduction, it is common to preserve the bottom 64 codes for footroom, but use all of the headroom region up to code 1019. Some digital cinema systems call this "extended" range; however, to my mind using the word "extended" runs the risk of suggesting that using that range is a good idea. (Who wouldn't want "extended?") In my view, the option should be labelled "extreme."

Interface offset

In hardware, an 8-bit interface is considered to convey values 0 through 255. In serial digital video interfaces (SDI, or its HD variant, HD-SDI), the all-zeros and all-ones 8-bit codes are prohibited from data. At an 8-bit digital video interface, an offset of +16 is added to the code values shown in boldface in Figure 4.4: Reference black is placed at interface code 16, and white at 235. I consider the offset to be added or removed at the interface, because a signed representation is necessary for many processing operations (such as changing gain). However, hardware designers often consider 8-bit digital video to have black at code 16 and white at 235.

Taking black as code 16 makes interface design easy, but makes signal arithmetic design more difficult.

Processing coding

One one-hundredth of the range from blanking to reference white was historically referred to as an *IRE* unit, for the Institute of Radio Engineers, the predecessor of today's IEEE. Now it is best to say *units.* The mapping from units to 8-bit digital video interface code is this:

$$V_{709} = 16 + 219 \cdot \frac{units}{100}$$

In signal processing, it is often convenient (and sometimes necessary) to use a coding that represents reference black at zero independent of coding range. To accommodate footroom, the number representation must allow negative numbers. In describing signal processing at an abstract level – or implementing signal processing in floating point arithmetic – it is simplest to use the range 0 to 1. The reference points 0 and 1 are taken to be reference black and reference white. (The range is also referred to as *units,* where there are 100 units from reference black to reference white.) To accommodate signals in the headroom region, the number representation must allow numbers greater than unity, and in the footroom region, less than zero.

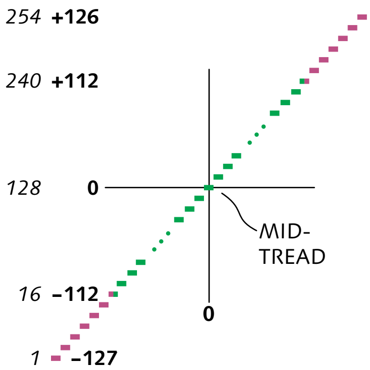
In processing hardware, a sample is ordinarily represented as a fixed-point integer with a limited number of bits. It is usually most convenient to use two's complement arithmetic. The bit depth required in processing is usually greater than that required at an interface. Black will ordinarily be coded at 0. Reference white will be coded to an appropriate value such as 219 in an 8-bit system or 876 in a 10-bit system.

In signal processing, even without the interface offset, it may be necessary to handle negative numbers. Two's complement binary representation is common.

R'G'B' or $Y'C_B C_R$ components of 8 bits each suffice for distribution of consumer video. However, if a video signal must be processed many times, say for inclusion in a multiple-layer composited image, then roundoff errors are liable to accumulate. To avoid roundoff error, studio video data typically carries 10 bits each of $Y'C_B C_R$. Ten-bit studio interfaces have the reference levels of Figures 4.4 and 4.5 multiplied by 4: The extra two bits are appended as least-significant bits to provide increased precision. Within processing equipment, intermediate results may need to be maintained to 12, 14, or even 16 bits.

Figure 4.4 showed a quantizer for a unipolar signal such as luma. C_B and C_R signals are bipolar, ranging positive and negative. For C_B and C_R it is standard to

Figure 4.5 A **Mid-tread quantizer for C_B and C_R** bipolar signals allows zero chroma to be represented exactly. (*Mid-riser* quantizers are rarely used in video.) For processing, C_B and C_R abstract values have a range of ±112. At an 8-bit studio video interface according to BT.601, an offset of +128 is added, indicated by the values in italics. Interface codes 0 and 255 are reserved for synchronization, as they are for luma.



use a *mid-tread* quantizer, such as the one graphed in Figure 4.5, so that zero chroma has an exact representation. For processing, a signed representation is necessary; at a studio video interface, it is standard to scale 8-bit colour difference components to an excursion of 224, and add an offset of +128. (Note that chroma occupies five more 8-bit codes than luma.)

Two's complement wrap-around

I use the subscript h to denote a hexadecimal (base 16) integer.

Modern computers use binary number representation. Signed integer arithmetic is implemented using two's complement representation. When the result of an arithmetic operation such as addition or subtraction overflows the fixed bit depth available, two's complement arithmetic wraps around. For example, in 16-bit two's complement arithmetic, taking the largest positive number, 32,767 (in hexadecimal, $7fff_h$) and adding one produces the smallest negative number, −32,768 (in hexadecimal, 8000_h). It is an insidious problem with computer software implementation of video algorithms that wrap-around is allowed in integer arithmetic. In video signal processing, such wrap-around must be prevented, and *saturating arithmetic* must be used.

Contrast, brightness, CONTRAST, and BRIGHTNESS 5

User-accessible controls labelled CONTRAST and BRIGHTNESS are found on nearly all electronic displays. These labels are indirectly and confusingly connected to the perceptual attributes *brightness* and *contrast*. In a CRT, adjusting BRIGHTNESS upwards from its optimum setting affects visual contrast much more than a comparable adjustment of the CONTRAST control. Adjusting CONTRAST affects visual brightness much more than a comparable adjustment of the BRIGHTNESS control. BRIGHTNESS and CONTRAST are therefore misleading labels. Today, CONTRAST and BRIGHTNESS controls are implemented in literally *billions* of pieces of equipment. Hundreds of millions of people have a poor understanding of these controls, and have had it for half a century: Imaging system designers are faced with a big problem.

This chapter describes the perceptual attributes *brightness* and *contrast.* I describe conventional CONTRAST and BRIGHTNESS controls, I explain how those controls came to do what they do, and I conclude by making some recommendations to reduce the confusion.

Perceptual attributes

HEYNDERICKX, INGRID and LANGENDIJK, ERNO (2005), "Image-quality comparison of PDPs, LCDs, CRTs and LCoS projection displays," in *SID Symposium Digest* **36** (1): 1502–1505.

According to two well respected vision and display system researchers,

> The four most important image quality attributes, at least for non-expert viewers when assessing image quality of high-end TVs, are brightness, contrast, color rendering and sharpness.

Here we address the first two image attributes, *brightness* and *contrast*, which presumably the authors consider the most important. Heynderickx and her colleague are referring to brightness and contrast as perceptual attributes. There are like-named controls on display equipment; however, I argue that the controls don't affect the perceptual attributes of a displayed image in the obvious manner. In the present chapter, including its title, we have to distinguish the names of the controls from the perceptual attributes. I typeset the names of the controls in small capitals – CONTRAST and BRIGHTNESS – and typeset normally the visual attributes brightness and contrast.

Contrast refers to a measured or visual distinction between colours or grey shades. Contrast is usually quantified by the ratio of a higher-valued luminance (or reflectance) to a lower-valued luminance (or reflectance). The ratio can be computed between widely different luminances; for example, when evaluating a display system we generally seek *contrast ratio* (the ratio of maximum to minimum luminance) of 100 or better, and perhaps up to 10,000. The ratio can be computed between similar luminances. Vision cannot distinguish two luminance levels when their contrast ratio falls below about 1.01 ("Weber's Law"), and the ratio between two luminances near the threshold of human detection is sometimes called *Weber contrast.*

History of display signal processing

Television originated with analog vacuum tube circuits; CRTs are themselves vacuum tubes. Vacuum tubes and the associated analog components (primarily resistors and capacitors) were subject to drift owing to operating temperature variation and owing to age-induced component degradation. The main effects of drift were to alter the gain and offset of the video signal; so, gain and offset controls were provided. Drift was such a serious problem that the controls were located on the front panel; consumers were expected to use them.

DRIVE historically referred to separate gain adjustments internally in the *R*, *G*, and *B* signal paths; SCREEN or BIAS referred to independent internal *R*, *G*, and *B* offset adjustments. In home theatre calibration circles these are respectively RGB-HIGH and RGB-LOW.

User-adjustable CONTRAST and BRIGHTNESS controls were implemented in vacuum tube television receivers of the early 1940s. Gain of the video amplifier circuitry was adjusted by a control that came to be called CONTRAST. Control of offset (bias) was implemented at the CRT itself, by a control called BRIGHTNESS. Gain control took effect earlier in the signal path than offset. Kallmann described a typical implementation:

KALLMANN, HEINZ.E. (1940), "The gradation of television pictures," in *Proc. IRE* **28** (4): 170–174 (Apr.).

> … the so-called contrast control … is a voltage divider controlling signal amplitude … the background-light control … adjusts bias on the cathode-ray tube.

The scheme described by Kallmann prevailed for the whole CRT era. CONTRAST and BRIGHTNESS circuitry operated in the *R'G'B'* domain – that is, operated on gamma-encoded signals. Historically, the CRT itself imposed the power function associated with display "gamma." In CRTs, gamma wasn't adjustable.

I have been unable to find any historical documents that discuss how the names CONTRAST and BRIGHTNESS came about. Some early television receivers used the label BRILLIANCE for the gain control and some used BACKGROUND for the offset control. Some early television models had concentric CONTRAST and VOLUME controls, suggesting a single place for the user to alter the magnitude of the sound and the magnitude of the picture. One model had BRIGHTNESS on the front panel between VERTICAL HOLD and FOCUS!

Video scientists, engineers, and technicians have been skeptical about the names CONTRAST and BRIGHTNESS for many, many decades. Sixty years ago, Oliver wrote:

OLIVER, B.M. (1950), "Tone rendition in television," in *Proc. IRE* **38** (11): 1288–1300 (Nov.).

> … the gain ("contrast") control certainly produces more nearly a pure brightness change than does the bias ("brightness") control, so the knobs are, in a sense, mislabeled.

The parentheses and quotes are in the original. Concerning BRIGHTNESS, Oliver stated:

> … A good name for this knob might be "blacks," or "background," or "shadows."

That these controls are misnamed was observed a few years later by the preeminent electronics engineer Donald Fink:

Fink, Donald G. (1952), Television Engineering, Second Edition (New York: McGraw-Hill)

> "Unfortunately, in television systems of the present day, ... the separate manipulation of the receiver

brightness and contrast controls (both of which are misnamed, photometrically speaking) by the nontechnical viewer may readily undo the best efforts of the system designers and the operating technicians."

In some modern television receivers, the gain control is labelled PICTURE instead of CONTRAST.

Despite researchers of the stature of Oliver and Fink complaining many decades ago, the names stuck – unfortunately, in my opinion.

Over 70 years, video signal processing technology shifted, first in about 1965 to transistors used in analog mode, then in about 1975 to analog integrated circuits, and then in about 1985 to digital integrated circuits, whose complexity has increased dramatically over the last 25 years. Around 2000, display technology started to shift from CRTs to LCD and PDP technology. With all of these shifts, the need for adjustment diminished. Nonetheless, CONTRAST and BRIGHTNESS have been carried forward (thoughtlessly, some would say) into successive generations of technology. Today, these controls are in use in around a billion CRT-based television receivers and another billion CRT displays in use with computers. The controls have been carried over (again, without much thought) into fixed-pixel displays; around a billion LCD displays are in use today, and virtually all have BRIGHTNESS and CONTRAST controls implemented in the digital signal processing path.

In video processing equipment, gain and offset controls have historically been available; they operate comparably to the display controls, but the associated controls are usually labelled GAIN and BLACK LEVEL.

LCD and plasma displays typically have CONTRAST and BRIGHTNESS controls. Despite the professional users' expectation that the controls would be implemented similarly to the like-named controls on a CRT display, and despite consumers' expectations that such controls should function in a comparable manner to CRTs, the LCD controls often have quite different effect.

CONTRAST and BRIGHTNESS controls are widespread in image applications in computers. The effect of CONTRAST and BRIGHTNESS controls in these domains is not necessarily comparable to the effect of like-named controls on display equipment. In particular, CONTRAST in Photoshop behaves very differently than CONTRAST in typical displays: Photoshop CONTRAST controls gain, but it "pivots" the gain around a certain formulation of the average pixel level instead of pivoting at zero as is usual in video equipment and in displays. For other computer imaging applications, there are no standards; it's often difficult or impossible to tell exactly how a particular application implements these controls.

Digital driving levels

The term *pixel value* is ambiguous because pixel values accessible to application software can be altered by the graphics subsystem on the way to the display – for example, they can be altered by the lookup table in a graphics adapter. DDL numbers are not necessarily those passed to the display panel column drivers and "glass": Modern display equipment ordinarily incorporates signal processing – often including lookup tables (LUTs) – to compensate for the native display panel response.

The term *digital driving level* (DDL) refers to a pixel component data value (typically produced by a PC graphics subsystem, or by a consumer electronics signal source such as a Blu-ray player) that crosses an interface (typically DVI, HDMI, or DisplayPort) and drives a display device. A DDL is interpreted as an integer value from 0 to $2^k - 1$ (where k is the bit depth at the interface, typically 8, but potentially 10 or 12).

Computer interfaces such as DVI carry 8-bit DDLs where DDL 0 is reference black and DDL 255 is reference white.

Video interface standards such as HD-SDI in the studio and HDMI in consumer equipment allow footroom below reference black and headroom above reference white. HD-SDI is standardized with 10-bit values; interface code 64 corresponds to reference black and interface code 940 corresponds to reference white. In consumer use, eight-bit HDMI is commonly used; interface code 16 corresponds to reference black and interface code 235 corresponds to reference white.

To simplify the rest of the discussion I'll refer to pixel values in terms of *normalized DDLs* (NDDLs) where reference black at the interface corresponds to NDDL 0 and reference white at the interface corresponds to NDDL 1. Pixel values in video are permitted to have modest excursions outside the reference 0 to 1 range, $-^{15}/_{219}$ to $^{235}/_{219}$ (about –0.07 to +1.09). The NDDL range 0 to 1 corresponds to what an HD engineer might call *IRE levels* 0 through 100.

Relationship between signal and lightness

The sRGB standard requires an electro-optical conversion function (EOCF) comprising a 2.2-power function.

Documentation associated with sRGB makes clear that the fundamental EOCF – the mapping from pixel value to display luminance – is supposed to be a pure 2.2-power function, followed by the addition of

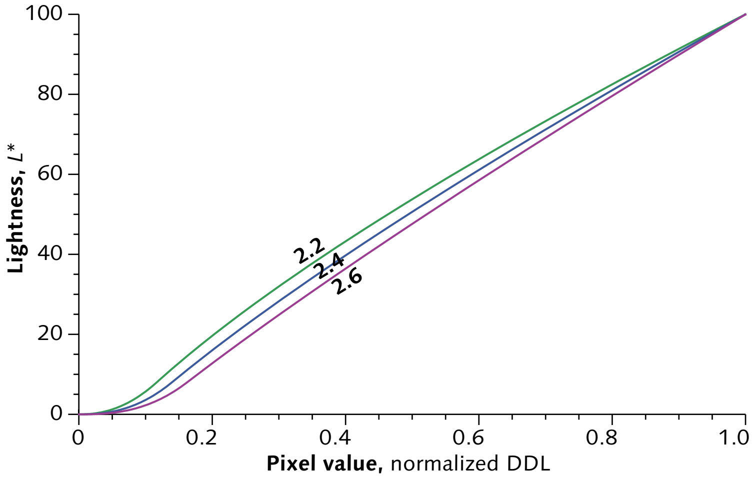


Figure 5.1 **Relationship between pixel value and L^*.** The *x*-axis plots normalized pixel (or video signal) value. The corresponding L^* values are plotted for display power functions ("gamma") of 2.2 (typical of computer imaging, e.g., sRGB), 2.4 (typical of studio video), and 2.6 (standard for digital cinema). This graph, and those to follow, apply to three channels when $R = G = B$, or to individual channels scaled appropriately (where the other two channels are zero). Adapted from Poynton's "Perceptual uniformity in digital imaging."

a veiling glare term. The sRGB standard also documents an OECF that is intended to describe a mapping from display luminance to pixel value, suitable to simulate a camera where the inverse power function's infinite slope at black would be a problem. The OECF has a linear segment near black, and has an power function segment with an exponent of $^1/_{2.4}$. The OECF should *not* be inverted for use as an EOCF; the linear slope near black is not appropriate for an EOCF.

Poynton, Charles (2009), "Perceptual uniformity in digital imaging," in *Proc. Gjøvik Color Imaging Symposium* (GCIS 2009): 102–109.

The sRGB 2.2-power function, the *de facto* 2.4-power function of today's studio reference displays, and the 2.6-power function of digital cinema all almost perfectly invert L^*, as depicted in Figure 5.1. The 2.4-curve, which typifies video and HD practice, has a highly linear relationship with L^* for NDDL 0.2 and above (that is, for 8-bit interface codes above 59). An NDDL of 0.2 yields an L^* value of 16. The line from [0.2, 16] through reference white has slope of 105; extending that line back towards the *x*-axis yields an *x*-intercept of 0.0475, and back further, a *y*-intercept of almost exactly –5.

Algorithm

The effect of conventional CONTRAST and BRIGHTNESS controls in video is approximated by the following

Figure 5.2 **Effect of gain control** ("CONTRAST") for nominal offset ("BRIGHTNESS") setting ($b = 0$). Gain values 0.8, 1.0, and 1.25 are shown; these are values of m in Equation 5.1. The y-axis is tristimulus value, a linear-light quantity that is not directly perceptually meaningful.

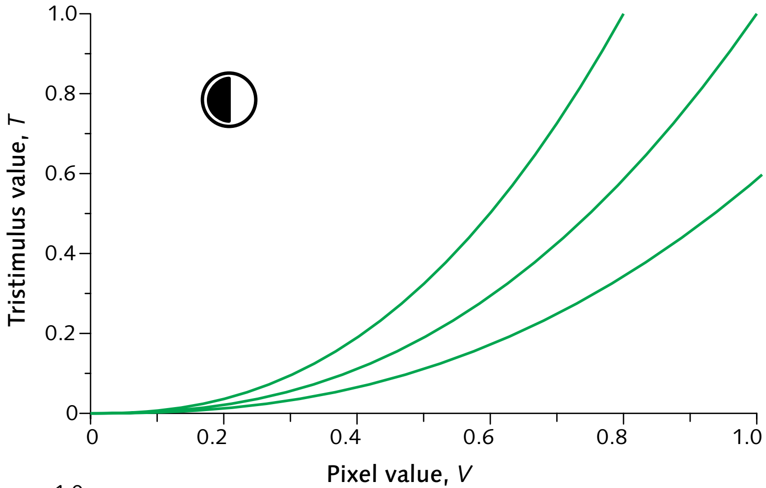
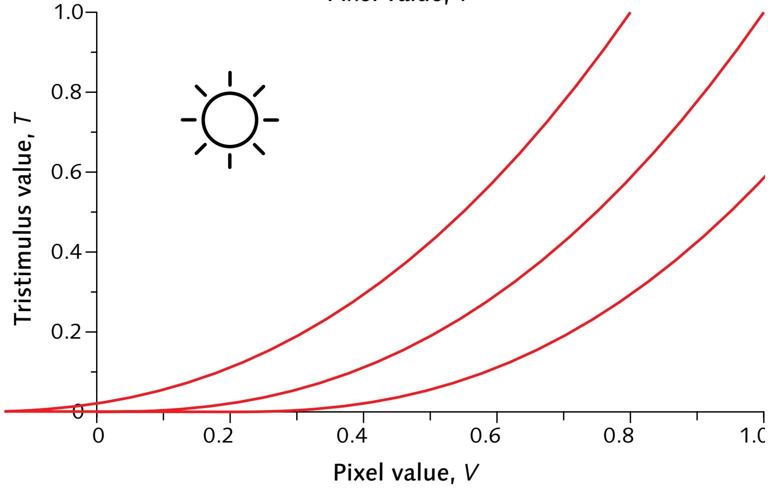


Figure 5.3 **Effect of offset control** ("BRIGHTNESS") for nominal gain ("CONTRAST") setting ($m = 1$). Offset values –0.2, 1.0, and +0.2 are shown; these are values of b in Equation 5.1.



equation, where x represents NDDL (one of R', G', or B') scaled to the range 0 to 1.

$$y = m\,x + b \qquad \text{Eq 5.1}$$

Contrast alters the m parameter – Oliver and Fink (cited on page 49) would have called it *gain* – over a range approximately 0.5 to 2. Brightness alters the b parameter – Oliver and Fink would have called it *offset* or *bias* – over a range approximately ±0.2.

Figures 5.2 and 5.3 show the effect on the display tristimuli of changing gain and offset respectively.

The result y of equation 5.1 is clipped (historically by the action of the CRT itself, or now by signal processing) so as not to fall below zero. At a sufficiently high signal value, probably not too much over 1.09, saturation is likely to set in. The user adjusting gain or offset should be careful that his or her settings don't induce clipping or saturation.

The *x* and y signals in equation 5.1 are in the gamma-corrected (*R'G'B'*) domain. The result is then raised to a modest power γ (*gamma,* ranging from about 2.0 to 2.6) to produce a displayed tristimulus value (*R*, *G*, or *B*, on the *y*-axes of Figures 5.2 and 5.3).

Historical analog gain control circuitry implemented CONTRAST as a "one-quadrant" multiplier on *R'G'B'* video signals clamped at blanking level (0). In PAL video, black and blanking levels were identically 0; consequently, adjusting CONTRAST in a PAL receiver left black of a properly coded signal where it was supposed to be (having minimal interaction with BRIGHTNESS). In NTSC encoding, +7.5-unit "setup" was inserted, causing black level of a properly encoded signal to lie at 0.075 on the 0 to 1 scale. However, clamping was still at blanking level (0). With gain "hinged" at zero, adjusting gain from 0.5 to 2 would alter black level from 0.0375 to 0.15, thereby causing CONTRAST and BRIGHTNESS to interact.

There is no standard or convention for the range of *m* and *b,* for the relationship of *m* and *b* values to the controls, or for numerical control values presented to the user. Today's video studio reference displays ("BVMs") are adjustable allowing *m* to range between about 0.5 and 2, and *b* to range about ±0.2; however in studio practice it is common for both controls to lack numerical control values. In today's consumer equipment, CONTRAST is typically presented to the user as a value (here denoted *C*) from 0 through 100 – shown for example on an on-screen display (OSD) – and BRIGHTNESS as a value (here, *B*) from 0 through 100. Suitable mappings from those control values to parameters *m* ranging 0.5 and 2 and *b* ranging ±0.2 are these:

$$m = 2^{\frac{C-50}{50}}; \quad b = \frac{B-50}{250} \qquad \text{Eq 5.2}$$

BRIGHTNESS values might alternatively be presented in the range –50 through +50, in which case the second mapping would be $b = {}^{B}/_{250}$.

Figure 5.4 **Effect of gain control** ("CONTRAST") on lightness, for nominal bias setting (b = 0), with gain control viewed as setting the display function. Gain values correspond to values of m in Equation 5.1.

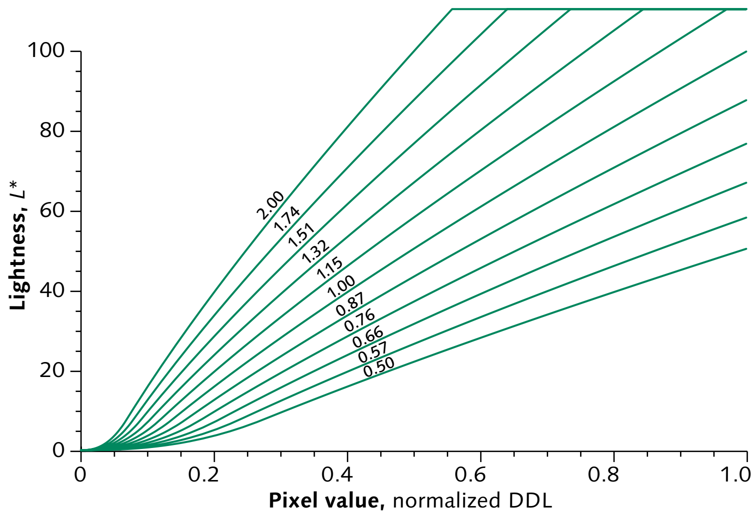


Figure 5.5 **Effect of offset control** ("BRIGHTNESS") for nominal gain setting (m = 1), with bias control viewed as setting the display function. Bias values correspond to values of b in Equation 5.1.

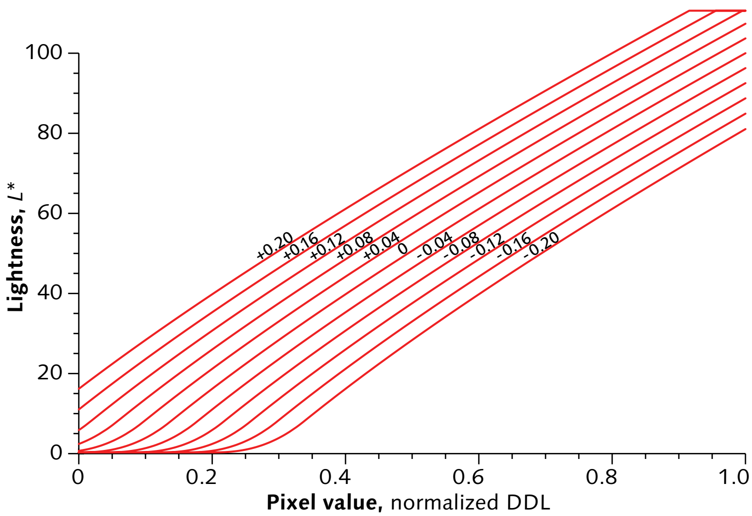


Figure 5.4 graphs the lightness produced for various settings of the gain parameter m, with the display characteristic considered a function of that parameter. The range of values in the graph corresponds to a CONTRAST range of 0 to 100 under the mapping of Equation 5.2.

Figure 5.5 graphs the lightness produced for various settings of the offset parameter b, with the display characteristic considered a function of that parameter. The range of values in the graph corresponds to a BRIGHTNESS range of 0 to 100 under the mapping of Equation 5.2.

Black level setting

PLUGE: Picture line-up generator, see page 421. ITU-R BT-814.2 standardizes a suitable test pattern. My procedure involves only the negative PLUGE bar, and is independent of its excursion. Traditional standards such as SMPTE RP 167 call for setting "so that the darker [negative] patch of the PLUGE just merges with the reference black level, but the brighter [positive] patch is clearly distinguishable." In my view the "but" phrase is not properly part of the optimization; instead, it provides a cross-check after the fact.

To set BRIGHTNESS (or BLACK LEVEL) in the studio, display a pattern containing PLUGE (levels –0.02, 0, +0.02) on a test image having average relative luminance of about 0.01 (1%). Set BLACK LEVEL high, then reduce it until the –0.02 and 0 PLUGE levels become just barely indistinguishable. You're finished.

If you have no PLUGE pattern, display a picture that is predominantly or entirely black. Set BLACK LEVEL to its minimum, then increase its level until the display barely shows a hint of dark grey, then back off a smidge.

Historically, BLACK LEVEL setting was somewhat dependent upon ambient light. However, modern displays have such low faceplate reflectance that ambient light contributes very little unwanted luminance, and the BLACK LEVEL setting is no longer very sensitive to ambient light. Modern display equipment is very stable; frequent adjustment is unnecessary.

Here I use standard digital studio video levels, not computing levels.

In the end, eight-bit codes 0 through 16 are expected to be indistinguishable. Code 16 (NDDL 0) is supposed to produce luminance that is visually indistinguishable from that of the negative-going bar of PLUGE (8-bit interface code 12, NDDL –0.02): The positive-going bar of PLUGE (8-bit interface code 20, NDDL +0.02) is expected to be visible.

See *Relative luminance,* on page 258, and *Display white reference,* on page 310. Astonishingly, no current studio standard specifies the luminance of reference white. I suggest 100 nt.

Once BLACK LEVEL is set correctly, CONTRAST can be set to whatever level is appropriate for comfortable viewing, provided that clipping is avoided. In the studio, the CONTRAST control can be used to achieve the desired luminance of reference white, typically around 100 $cd{\cdot}m^{-2}$. (Historically, Europe used a somewhat lower reference white luminance, perhaps 80 $cd{\cdot}m^{-2}$.)

Effect of CONTRAST and BRIGHTNESS on contrast and brightness

To explore the visual effect of CONTRAST and BRIGHTNESS controls, consider an ideal, properly adjusted 8-bit HD studio display.

Decreasing BRIGHTNESS leads to a darker image. Ignoring "shadow detail," a naïve viewer may find the resulting picture superior!

Decreasing BRIGHTNESS from its optimum setting causes clipping of video content lying just above reference black. Clipping doesn't impair contrast ratio *per se*, but stripping out image content "in the shadows" produces obvious artifacts, so we won't explore decreasing BRIGHTNESS.

Let's compute the effect of CONTRAST on contrast ratio. Assume a typical studio contrast ratio of 3333 (100 nt white, 0.03 nt black). Decreasing CONTRAST by 20% reduces the white video signal to 0.8, yielding a relative luminance of 0.585. Increasing CONTRAST by 20% increases the white signal to 1.25, yielding a relative luminance of 1.71. Starting with a contrast ratio of 3333, adjusting CONTRAST ±20% decreases contrast ratio to about 1950 or increases it to about 5700.

Let's compute the effect of CONTRAST on "brightness," as estimated by L^*. Adjusting CONTRAST ±20% yields L^* ranging from 81 to 118.

To compute the effect of increasing BRIGHTNESS on contrast ratio, increasing BRIGHTNESS by 20% takes the *y*-intercept of the 2.4-gamma curve of Figure 5.1 from –5 to +3. Reference black code now produces relative luminance of about 0.00332; reference white produces relative luminance of about 1.08. Increasing BRIGHTNESS thus causes contrast ratio to drop from 3333 to $^1/_{0.00332}$, that is, to 325.

Finally, increasing BRIGHTNESS by +20% causes the reference white signal to increase L^* to 103.

Increasing CONTRAST by 20% takes contrast ratio from 3333 to 5700, roughly a factor of 2. Increasing BRIGHTNESS by 20% drops contrast ratio from 3333 to 325, roughly a factor of 10. A 20% change in BRIGHTNESS has much more effect on contrast ratio than a 20% change in CONTRAST.

Increasing BRIGHTNESS by 20% takes L^* from 100 to 103, but increasing CONTRAST by 20% takes L^* from 100 to 118. The results are summarized in table 5.1:

	Contrast ratio	*Ref. black L**	*Ref. white L**
Nominal	3333	0 3	100
Decrease CONTRAST 20%	1950	0.5	81
Increase CONTRAST 20%	5700	0.2	118
Increase BRIGHTNESS 20%	325	2.8	103

Table 5.1 **Effect of adjusting CONTRAST and BRIGHTNESS**

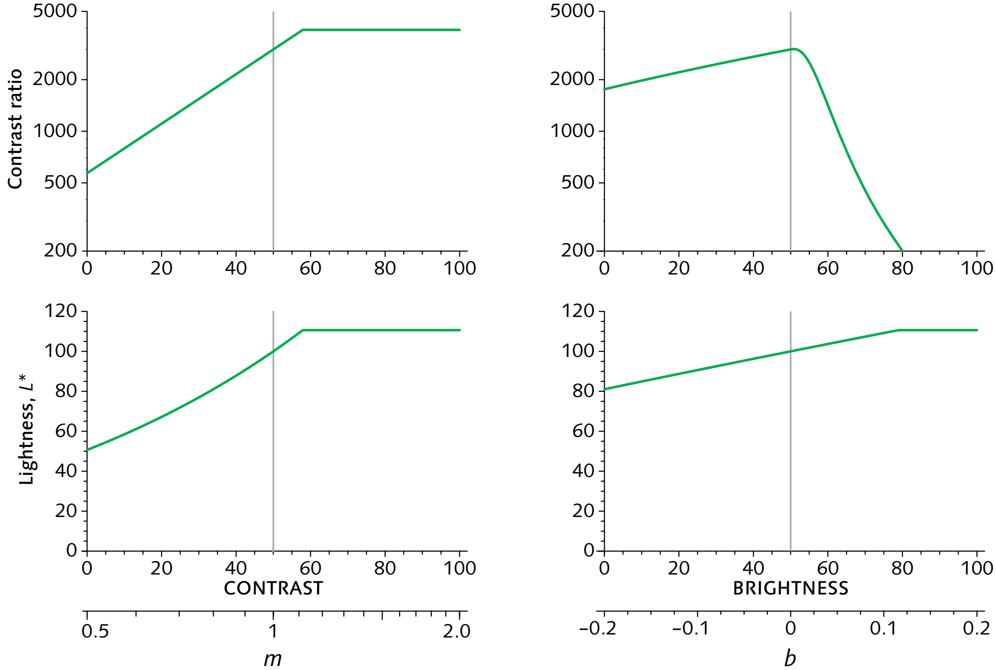


Figure 5.6 **Contrast ratio and lightness (*L**)** are graphed in the upper and lower pairs, as a function of the *m* parameter ranging from 0.5 to 2 (with the typical CONTRAST setting 0 to 100) graphed at the left, and the *b* parameter with a range of ±0.2 (with the typical BRIGHTNESS setting 0 to 100) graphed at the right. The display EOCF underlying these graphs clips at about 109% of the video signal, that is, at a relative luminance of about $1.09^{2.4}$ or 1.23. The light grey vertical lines indicate the default $m = 1$, $b = 0$ (that is, CONTRAST 50 and BRIGHTNESS 50).

$$\left(\frac{1019-64}{940-64}\right)^{2.4} \approx 1.23$$

This numerical example is elaborated by the four graphs of Figure 5.6, which show the effect on contrast ratio (at the top) and lightness (L^*, at the bottom) of adjusting CONTRAST (at the left) and BRIGHTNESS (at the right), with the CONTRAST and BRIGHTNESS scales corresponding to the mappings to *m* and *b* of Equation 5.2. The optimization of contrast ratio by choosing the appropriate BRIGHTNESS setting is clearly evident in the peak of the top-right graph. The other three graphs show saturation (clipping), which for this example is taken to set in at the studio video level of 109% of reference white, corresponding to relative luminance of about 1.23.

From the right-hand halves of the two top graphs it is evident that an adjustment to BRIGHTNESS above its optimum setting causes contrast ratio to decrease at roughly three times the rate that contrast ratio increases when CONTRAST is adjusted (in its nonclipped region):

Contrast ratio is more responsive to BRIGHTNESS than to CONTRAST. From the bottom graphs, adjusting either CONTRAST or BRIGHTNESS upwards increases the lightness of white (until clipping sets in), but the CONTRAST control is more responsive.

An alternate interpretation

In Figures 5.4 and 5.5, I interpreted the CONTRAST and BRIGHTNESS controls as changing the display's characteristics for a fixed scale of input pixel values (normalized DDLs). Let's turn that around, and consider the display characteristic to be a fixed function of display reference values ranging 0 through 1. equation 5.1 implements a linear operation on the *x*-axis of figure 5.1. Adjustment of CONTRAST and BRIGHTNESS can be interpreted as scaling and offsetting along that axis.

We can establish a parameter *B* (accessible to the user as BLACK LEVEL) to control the display reference value intended to be produced by NDDL 0, and parameter *W* (accessible to the user as WHITE LEVEL) to control the display reference value intended to be produced by NDDL 1.

Figure 5.7 overleaf shows the new interpretation. The *x*-axis in Figures 5.4 and 5.5 has been relabelled *Display reference value;* underneath that is the *Pixel value (normalized DDL)* scale. The NDDL scale is now squeezed and offset. The example of Figure 5.5 has BLACK LEVEL of 0.1 and WHITE LEVEL of 0.9. BLACK LEVEL has been elevated so that NDDL 0 produces an L^* value of about 3; WHITE LEVEL is set so that NDDL 1 produces an L^* value of about 90.

The reparameterized version of Equation 5.1 is this:

$$y = \left(W - B\right) \cdot x + B \qquad \text{Eq 5.3}$$

To implement an offset range comparable to a conventional BRIGHTNESS control, and to allow treatment of input signals that have black-level errors, settings for *B* should have a range of about ±0.2. To be comparable to the gain range of a conventional CONTRAST control, settings for *W* should extend from 0.5 to 2.0. Most displays will be expected to exhibit clipping at *W* values greater than about 1.1, and it may be desirable to limit the user setting to such a value.

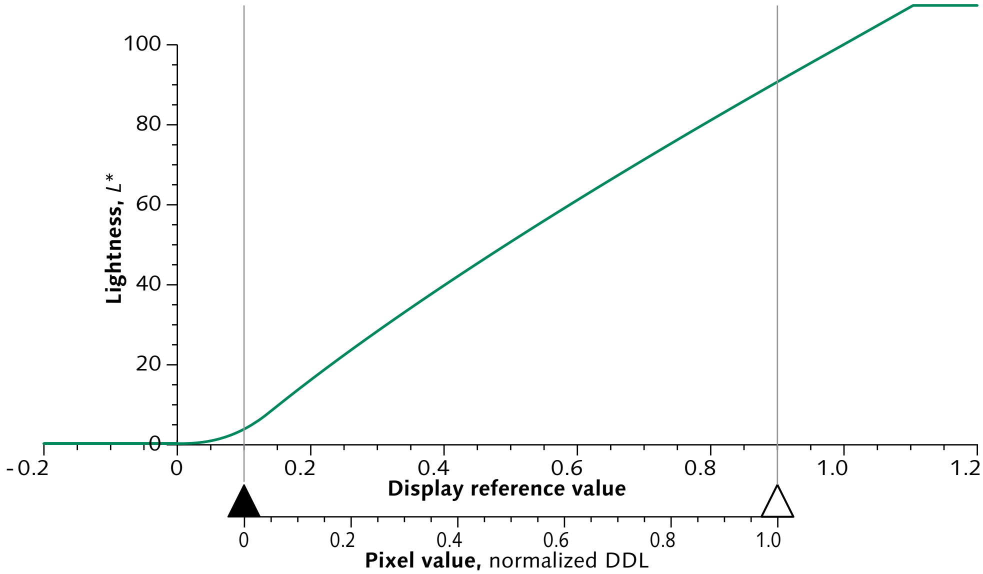


Figure 5.7 **Black level and white level controls.** The display is viewed as having a fixed conversion from display reference values (0 to slightly more than 1) to lightness. Black level and white level controls (indicated by the black and white triangles) set the display values corresponding to normalized interface pixel values 0 and 1. In this example, black level is set to 0.1 and white level to 0.9; *m* is computed as 0.8 and *b* as 0.1.

Concerning user adjustment of "poor sources," consider Poynton's Fourth Law: *Once a program is approved and packaged, errors in mastering are indistinguishable from expressions of creative intent.*

For consumer equipment, the black levels of modern source material are quite stable, and user adjustment to compensate for poor sources is no longer required. The diffuse ambient reflectance of modern displays is so low – around 0.01 – that ambient illuminance has a minor effect on contrast ratio. User adjustment to compensate for ambient light is now rarely necessary. Manufacturers should therefore consider relegating black level to an internal or service adjustment.

In a display, black level is normally used to compensate for the display, not the input signal, and so it should be effected downstream of the gain (contrast) control.

In processing equipment, it is sometimes necessary to correct errors in black level in an input signal while maintaining unity gain: A black level control should be implemented prior to the application of gain (and should not be called brightness). Figures 5.8 and 5.9 plot the transfer functions of contrast and brightness controls in the video signal path, disregarding the typical 2.4-power function of the display.

Figure 5.8 **The BRIGHTNESS (or BLACK LEVEL) control in video** applies an offset, roughly ±20% of full scale, to *R'G'B'* components. Though this function is evidently a straight line, the input and output video signals are normally in the gamma-corrected (perceptual) domain; the values are *not* proportional to intensity. At the minimum and maximum settings, I show clipping to the BT.601 footroom of $-^{15}/_{219}$ and headroom of $^{238}/_{219}$. (Light power cannot go negative, but electrical and digital signals can.)

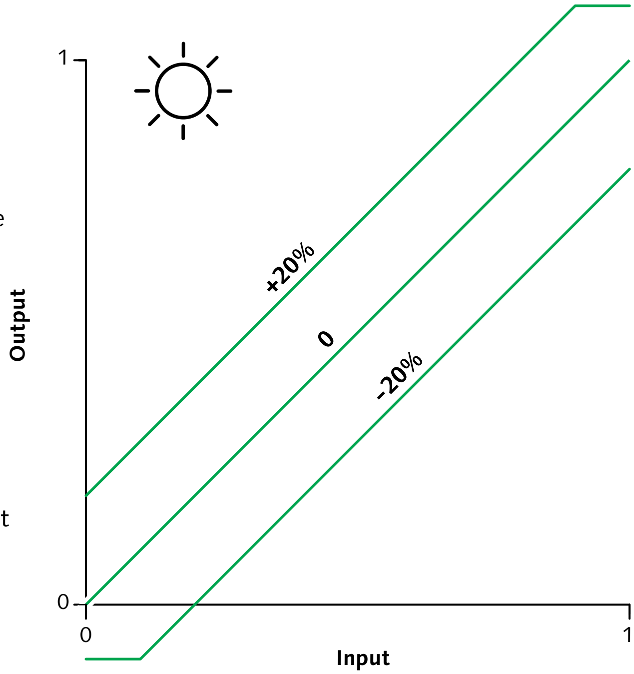
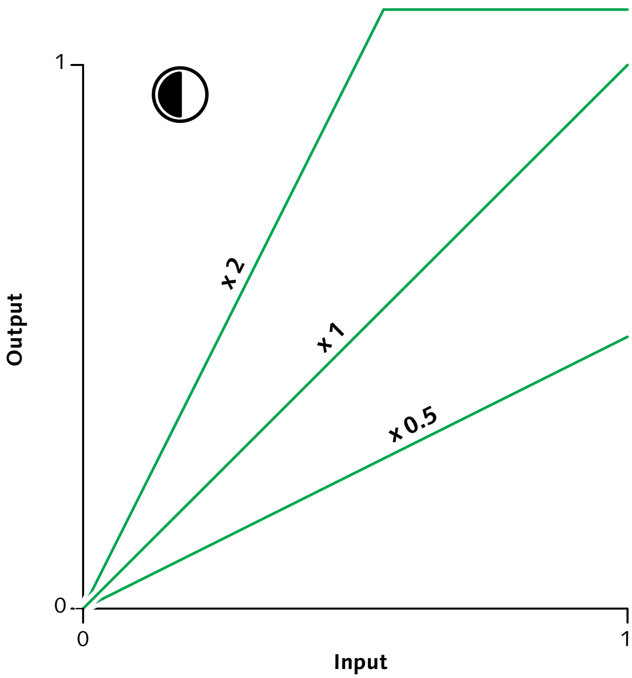


Figure 5.9 **The CONTRAST (or VIDEO LEVEL) control in video** applies a gain factor between roughly 0.5 and 2.0 to *R'G'B'* components. The output signal clips if the result would fall outside the range allowed for the coding in use. Here I show clipping to the BT.601 headroom limit.



Brightness and contrast controls in LCDs

LCD: liquid crystal display

In LCD displays, brightness typically alters the luminance of the backlight; its function is comparable to the contrast control of a CRT display. LCD displays produce luminance that is a nonlinear function of drive voltage. In early LCDs, contrast adjusted an electrical bias voltage at the panel. In modern LCDs, contrast adjusts gain in the signal path. There is no good reason for LCDs to have separate *R*, *G*, and *B* bias controls (RGB-low).

Brightness and contrast controls in PDPs

PDP: plasma display panel

In PDP displays, maximum luminance is fixed by the electronic design of the panel; brightness and contrast are implemented by digital signal processing. PDP displays produce luminance that is a linear function of drive level. DDL 0 produces the smallest possible luminance from the display, so reference black video code should produce DDL 0 – there is no good reason to have it otherwise. There is no good reason for PDPs to have separate *R*, *G*, and *B* bias controls (RGB-low).

Brightness and contrast controls in desktop graphics

This section describes Photoshop brightness and contrast controls for versions up to and including CS2, and for later versions when the "Use Legacy" option is enabled. The default brightness and contrast controls for versions CS3 and above behave differently.

Adobe's Photoshop software established the *de facto* effect of brightness and contrast controls in desktop graphics. Photoshop's brightness control is similar to the brightness control of video; however, Photoshop's contrast differs dramatically from that of video.

The transfer functions of Photoshop's controls are sketched in Figures 5.10 and 5.11 (opposite). *R'*, *G'*, and *B'* component values in Photoshop are presented to the user as values between 0 and 255. Brightness and contrast controls have sliders with a range of ±100.

Brightness effects an offset between –100 and +100 on the *R'*, *G'*, and *B'* components. Any result outside the range 0 to 255 clips to the nearest extreme value, 0 or 255. Photoshop's brightness control is comparable to that of video, but its range (roughly ±40% of full scale) is greater than the typical video range (of about ±20%).

Photoshop's (legacy) contrast control follows the application of brightness; it applies a gain factor. Instead of leaving reference black (code zero) fixed, as

Figure 5.10 **The BRIGHTNESS control in Photoshop** applies an offset of –100 to +100 to *R'G'B'* components ranging from 0 to 255. If a result falls outside the range 0 to 255, it saturates; headroom and footroom are absent. The function is evidently linear, but depending upon the image coding standard in use, the input and output values are generally nonlinearly related to luminance (or tristimulus values).

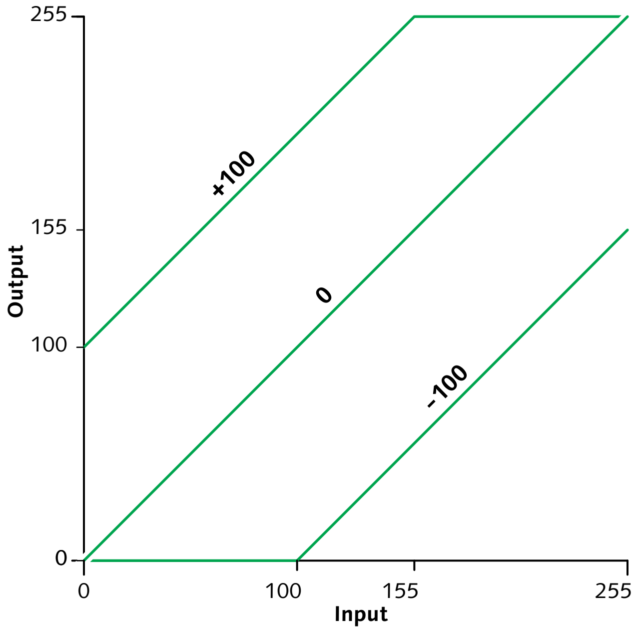
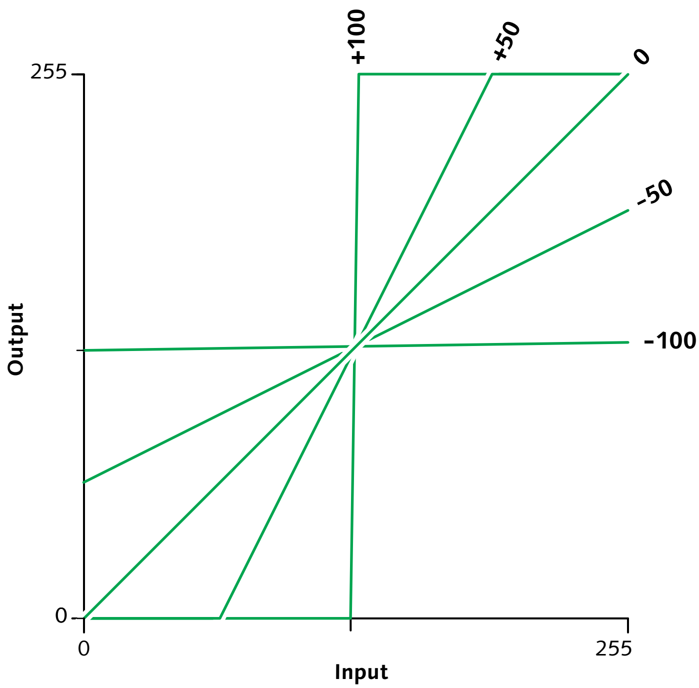


Figure 5.11 **The CONTRAST control in Photoshop** applies a gain factor between zero (for CONTRAST setting of –100) and infinity (for CONTRAST setting of +100) to image data, but "pivoted" around a weighted average pixel level (APL) of the image data, instead of "pivoting" around zero (as is the case for GAIN and CONTRAST controls in video). Each component result saturates if it falls outside the range 0 to 255.



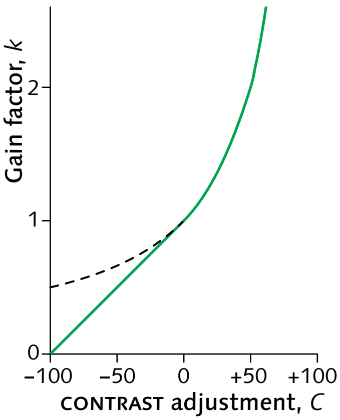


Figure 5.12 **Photoshop CONTRAST control's gain factor** depends upon the CONTRAST setting according to this function.

a video CONTRAST control does, Photoshop "pivots" the gain adjustment around a weighted average of the image data formed as 0.299 R'+0.587 G'+0.114 B'. (For image data having the "gamma correction" of video, the weighted average corresponds to BT.601 luma, or *average pixel level,* APL.) The transfer function for various settings of CONTRAST adjustment, for a weighted image average of 127.5, is graphed in Figure 5.11.

The gain available from Photoshop's CONTRAST control ranges from zero to infinity, far wider than the typical range of 0.5 to 2 of studio GAIN. The function that relates Photoshop's CONTRAST to gain is graphed in Figure 5.12. From the –100 setting to the 0 setting, gain ranges linearly from zero through unity. From the 0 setting to the +100 setting, gain ranges nonlinearly from unity to infinity, following the reciprocal curve described by Equation 5.4:

$$k = \begin{cases} 1 + \dfrac{C}{100}, & -100 \le C < 0 \\[2ex] \dfrac{1}{1 - \dfrac{C}{100}}, & 0 \le C < 100 \end{cases} \qquad \text{Eq 5.4}$$

$0.66 = \dfrac{1.45}{2.2}$

In desktop graphics applications such as Photoshop, image data is usually coded in a perceptually uniform manner, comparable to video $R'G'B'$. On a PC, $R'G'B'$ components are by default proportional to the $1/_{2.2}$-power of reproduced luminance (or tristimulus) values. On Macintosh computers prior to Mac OS X 10.6, QuickDraw $R'G'B'$ components were by default proportional to the 0.66-power of displayed luminance (or tristimulus). Modern Macintosh computers conform to the sRGB standard. However, on both PC and Macintosh computers, the user, system software, or application software can set the transfer function to nonstandard functions – perhaps so far as effecting linear-light coding – as will be described in *Gamma,* on page 315.

Raster images in computing 6

This chapter places video into the context of computing. Images in computing are represented in three forms, depicted schematically in the three rows of Figure 6.1: *symbolic image description*, *raster image*, and *compressed image*.

- A **symbolic image description** does not directly contain an image, but contains a high-level 2-D or 3-D geometric description of an image, such as its objects and their properties. A two-dimensional image in this form is sometimes called a *vector graphic*, though its primitive objects are usually much more complex than the straight-line segments suggested by the word *vector*.
- A **raster image** enumerates the greyscale or colour content of each pixel directly, in scan-line order. There are four fundamental types of raster image: *bilevel*, *pseudocolour*, *greyscale*, and *truecolour*. In Figure 6.1, the four types are arranged in columns, from low quality at the left to high quality at the right.
- A **compressed image** originates with raster image data, but the data has been processed to reduce storage and/or transmission requirements. The bottom row of Figure 6.1 indicates several compression methods. At the left are lossless (data) compression methods, generally applicable to bilevel and pseudocolour image data; at the right are lossy (image) compression methods, generally applicable to greyscale and truecolour.

The greyscale, pseudocolour, and truecolour systems used in computing involve lookup tables (LUTs) that map pixel values into display *R'G'B'* values. Most computing systems use perceptually uniform image coding; however, some systems use linear-light coding,

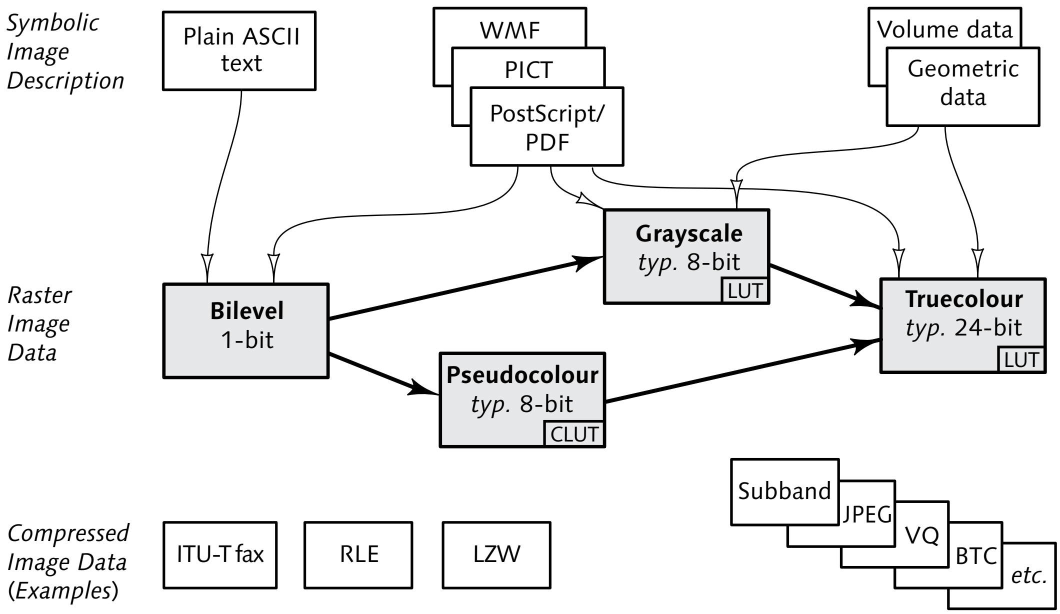


Figure 6.1 **Raster image data** may be captured directly, or may be rendered from symbolic image data. Traversal from left to right corresponds to conversions that can be accomplished without loss. Some raster image formats are associated with a lookup table (LUT) or colour lookup table (CLUT).

and some systems use other techniques. For a system to operate in a perceptually uniform manner, similar to or compatible with video, its LUTs need to be loaded with suitable transfer functions. If the LUTs are loaded with transfer functions that cause code values to be proportional to intensity, then the advantages of perceptual uniformity will be diminished or lost.

Murray, James D., and William vanRyper (1996), *Encyclopedia of Graphics File Formats*, Second Edition (Sebastopol, Calif.: O'Reilly & Associates).

Many different file formats are in use for each of these representations. Discussion of file formats is outside the scope of this book. To convey photographic-quality colour images, a file format must accommodate *at least* 24 bits per pixel. To make maximum perceptual use of a limited number of bits per component, nonlinear coding should be used, as I outlined in *Perceptual uniformity,* on page 30.

Symbolic image description

Many methods are used to describe the content of a picture at a level of abstraction higher than directly enumerating the value of each pixel. Symbolic image data is converted to a raster image by the process of *rasterizing*. Images are *rasterized* (or *imaged* or *rendered*) by interpreting symbolic data and producing raster image data. In Figure 6.1, this operation passes information from the top row to the middle row.

Geometric data describes the position, size, orientation, and other attributes of objects; 3-D geometric data may be interpreted to produce an image from a particular viewpoint. Rasterizing from geometric data is called *rendering;* truecolour images are usually produced.

Adobe's PostScript system is widely used to represent 2-D illustrations, typographic elements, and publications. PostScript is essentially a programming language specialized for imaging operations. When a PostScript file is executed by a PostScript interpreter, the image is rendered. (In PostScript, the rasterizing operation is often called raster image processing, or *RIPping*.) An encapsulated PostScript file (EPS or EPSF) is a special case of a PostScript file that describes one image; Adobe's PDF format is essentially a nonprogrammable variant of PostScript.

Once rasterized, there is no general method of converting raster image data back into a symbolic description: A raster image – in the middle row of Figure 6.1 – generally cannot be returned to its description in the top row. If your application involves rendered images, you may find it useful to retain the symbolic data even after rendering, in case the need arises to rerender the image, at a different size, perhaps, or to perform a modification such as removing an object.

Images from a fax machine, a video camera, or a greyscale or colour scanner originate in raster image form: No symbolic description is available. Optical character recognition (OCR) and raster-to-vector techniques make brave but generally unsatisfying attempts to extract text or geometric data from raster images.

Raster images

There are four distinct types of raster image data:

- *Bilevel*, by definition 1 bit per pixel;
- *Greyscale,* typically 8 bits per pixel;
- *Pseudocolour,* typically 8 bits per pixel; and
- *Truecolour,* typically 24 bits per pixel.

Greyscale and truecolour systems are capable of representing continuous tone. Video systems use only truecolour (and perhaps greyscale as a special case).

Historical *hicolour* systems assigned 5 bits to each of red, green, and blue, which – with one bit of overhead – formed a 16-bit pixel. (Some systems assigned 6 bits to green and had no overhead bit.) The hicolour scheme is obsolete.

In the following sections, I will explain bilevel, greyscale, truecolour, and pseudocolour in turn. The truecolour and pseudocolour descriptions are accompanied by block diagrams that represent the hardware at the back end of the framebuffer or graphics card. (Historically, this hardware would have included *digital-to-analog converters*, DACs; today, digital display interfaces such as DVI, HDMI, and DisplayPort are used.) Alternatively, you can consider each block diagram to represent an algorithm that converts image data to display R', G', and B' components.

Bilevel

Each pixel of a bilevel (or two-level) image comprises one bit, which represents either black or white – but nothing in between. In computing this is often called *monochrome*. (That term ought to denote shades of a single hue; however, in common usage – and particularly in video – *monochrome* denotes the black-and-white, or greyscale, component of an image.)

Since the invention of data communications, binary zero (0) has been known as *space*, and binary one (1) has been known as *mark*. A "mark" on an electronic display device emits light, so in video and in computer graphics a binary one (or the maximum code value) conventionally represents white. In printing, a "mark" deposits ink on the page, so in printing a binary one (or in greyscale, the maximum pixel value) conventionally represents black.

Greyscale

A greyscale image represents an effectively continuous range of tones, from black, through intermediate shades of grey, to white. A greyscale system with a sufficient number of bits per component, 8 bits or more, can represent a black-and-white photograph. A greyscale system may or may not have a lookup table (LUT); it may or may not be perceptually uniform.

In printing, a greyscale image is said to have *continuous tone*, or *contone* (distinguished from *line art* or *type*). When a contone image is printed, *halftoning* is ordinarily used.

Truecolour

A truecolour system has separate red, green, and blue components for each pixel. In most truecolour systems, each component is represented by a *byte* of 8

CLUTs are necessary for pseudocolour. Most truecolour systems have LUTs as by-products of their capability to handle pseudocolour.

bits (known as 8 *bits per channel,* or bpc). Each pixel has three components (channels), so this mode is often called "24-bit colour" (or "millions of colours"). The *RGB* values of each pixel can represent 2^{24}, or about 16.7 million, distinct codes. In computing, a truecolour framebuffer usually has three *lookup tables* (LUTs), one for each component. The LUTs and DACs of a 24-bit truecolour system are sketched in Figure 6.2 below.

The mapping from image code value to display voltage is determined by the content of the LUTs. Owing to the perceptually uniform nature of the display, the best perceptual use is generally made of truecolour pixel values when each LUT contains an identity function ("ramp") that maps input to output, unchanged.

In computing, the LUTs can be set to implement an arbitrary mapping from code value to tristimulus value (and so, to intensity). The total number of pixel values that represent distinguishable colours depends upon the transfer function used. If the LUT implements a power function to impose gamma correction on linear-light data, then the code-100 problem will be at its worst. With 24-bit colour and a properly chosen transfer function, photographic quality images can be displayed and geometric objects can be rendered smoothly shaded with sufficiently high quality for many applications. But if the LUTs are set for linear-light representation with 8 bits per component, contouring

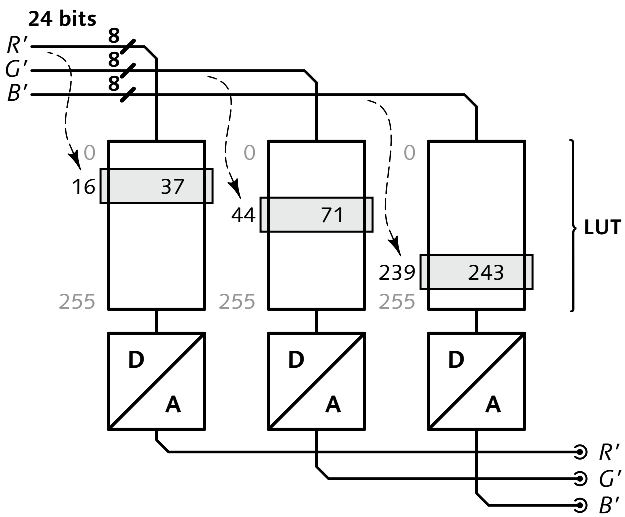


Figure 6.2 **Truecolour (24-bit) graphics** usually involves three programmable lookup tables (LUTs). The numerical values shown here are from the default Macintosh LUT (prior to Mac OS X 10.6). In video, *R'G'B'* values are transmitted to the display with no intervening lookup table. To make a truecolour computer system display video properly, the LUTs must be loaded with ramps that map input to output unchanged. Such a "unity" mapping (or "ramp") is the default in PCs.

will be evident in many images, as I mentioned on page 30. Having 24-bit truecolour is *not* a guarantee of good image quality. If a scanner claims to have 30 bits (or 36 bits) per pixel, obviously each component has 10 bits (or 12 bits). However, it makes a great deal of difference whether these values are coded physically (as linear- light luminance, loosely "intensity"), or coded perceptually (as a quantity comparable to lightness).

POYNTON, CHARLES (1998), "The rehabilitation of *gamma*," in Proc. SPIE **3299** (*Human Vision and Electronic Imaging III*, Bellingham, Wash.: SPIE).

In video, either the LUTs are absent, or each is set to the identity function. Studio video systems are effectively permanently wired in truecolour mode with perceptually uniform coding: Code values are presented directly to the DACs, without intervening lookup tables.

It is easiest to design a framebuffer memory system where each pixel has a number of bytes that is a power of two; so, a truecolour framebuffer often has four bytes per pixel – "32-bit colour." Three bytes comprise the red, green, and blue colour components; the fourth byte is used for purposes other than representing colour. The fourth byte may contain overlay information, or it may store an *alpha* (α) or *key* component representing opacity from black (0, fully transparent) to white (1, fully opaque). In computer graphics, the alpha component modulates components (usually *RGB*) that are coded in the linear-light domain. In video, the *linear key* signal modulates nonlinear (gamma-corrected) *R'G'B'* signals or luma and colour difference components such as $Y'C_BC_R$.

Concerning alpha, see page 387.

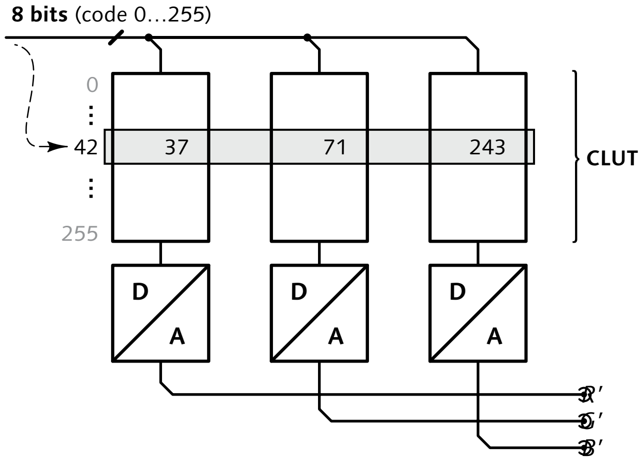
Pseudocolour

In a *pseudocolour* (or *indexed colour*, or *colour-mapped*) system, several bits – usually 8 – comprise each pixel in an image or framebuffer. This provides a moderate number of unique codes – usually 256 – for each pixel. Pseudocolour involves "painting by numbers," where the number of colours is rather small. In an 8-bit pseudocolour system, any particular image, or the content of the framebuffer at any instant in time, is limited to a selection of just 2^8 (or 256) colours from the universe of available colours.

I reserve the term *CLUT* for pseudocolour. In greyscale and truecolour systems, the LUTs store transfer functions, not colours. In Macintosh, pseudocolour CLUT values are roughly, but not optimally, perceptually coded.

Each code value is used as an index into a *colour lookup table* (CLUT, *colourmap,* or *palette*) that retrieves *R'G'B'* components; the DAC translates these linearly into voltage levels that are applied to the display. (Macintosh is an exception: Image data read from the

Figure 6.3 **Pseudocolour (8-bit) graphics** systems use a limited number of integers, usually 0 through 255, to represent colours. Each pixel value is processed through a *colour lookup table* (CLUT) to obtain red, green, and blue output values to be delivered to the display.



CLUT is in effect passed through a second LUT.) Pseudocolour CLUT values are effectively perceptually coded.

The CLUT and DACs of an 8-bit pseudocolour system are sketched in Figure 6.3 above. A typical lookup table retrieves 8-bit values for each of red, green, and blue, so each of the 256 different colours can be chosen from a universe of 2^{24}, or 16777216, colours. (A CLUT may return 4, 6, or more than 8 bits for each component.)

Pseudocolour image data is always accompanied by the associated colourmap (or *palette*). The colourmap may be fixed, independent of the image, or it may be specific to the particular image (*adaptive* or *optimized*).

The browser-safe palette forms a radix-6 number system with *RGB* digits valued 0 through 5.

$216 = 6^3$

A popular choice for a fixed CLUT is the *browser safe* palette comprising the 216 colours formed by combinations of 8-bit *R′*, *G′*, and *B′* values chosen from the set {0, 51, 102, 153, 204, 255}. This set of 216 colours fits nicely within an 8-bit pseudocolour CLUT; the colours are perceptually distributed throughout the *R′G′B′* cube.

Pseudocolour is appropriate for images such as maps, schematic diagrams, or cartoons, where each colour or combination is either completely present or completely absent at any point in the image. In a typical CLUT, adjacent pseudocolour codes are generally completely unrelated; for example, the colour assigned to code 42 has no necessary relationship to the colour assigned to code 43.

Conversion among types

In Figure 6.1, traversal from left to right corresponds to conversions that can be accomplished without loss.

Disregarding pseudocolour for the moment, data in any of the other three schemes of Figure 6.1 can be "widened" to any scheme to the right simply by assigning the codes appropriately. For example, a greyscale image can be widened to truecolour by assigning codes from black to white. Widening adds bits but not information.

A pseudocolour image can be converted to truecolour through software application of the CLUT. Conversion to truecolour can be accomplished without loss, provided that the truecolour LUTs are sensible.

Concerning conversions in the reverse direction, an image can be "narrowed" without loss only if it contains only the colours or shades available in the mode to its left in Figure 6.1; otherwise, the conversion will involve loss of shades and/or loss of colours.

Ashdown, Ian, *Color Quantization Bibliography,* available at <http://liinwww.ira.uka.de/bibliography/Graphics/cquant.html>

A truecolour image can be approximated in pseudocolour through software application of a fixed colourmap. Alternatively, a *colourmap quantization* algorithm can be used to examine a particular image (or sequence of images), and compute a colourmap that is optimized or adapted for that image or sequence.

Image files

Images in bilevel, greyscale, pseudocolour, or truecolour formats can be stored in files. A general-purpose image file format stores, in its header information, the count of columns and rows of pixels in the image.

Image width is the product of so-called resolution and the count of image columns; height is computed similarly from the count of image rows.

Many file formats – such as TIFF and EPS – store information about the intended size of the image. The intended image width and height can be directly stored, in absolute units such as inches or millimeters. Alternatively, the file can store sample density in units of *pixels per inch* (ppi), or less clearly, *dots per inch* (dpi). Sample density is often confusingly called "resolution."

In some software packages, such as Adobe Illustrator, the intended image size coded in a file is respected. In other software, such as Adobe Photoshop, viewing at 100% implies a 1:1 relationship between file pixels and display device pixels, disregarding the number of pixels per inch in the file and at the display. Image files

A *point* is a unit of distance equal to $1/72$ inch. The width of the stem of this bold letter **I** is one point, about 0.353 mm (that is, 353 μm).

without size information are often treated as having 72 pixels per inch; application software unaware of image size information often uses a default of 72 ppi.

"Resolution" in computer graphics

In computer graphics, a pixel is often regarded as an intensity distribution uniformly covering a small square area of the screen. In fixed-pixel displays such as liquid crystal displays (LCDs), plasma display panels (PDPs), and digital light processing (DLP) displays, discrete pixels such as these are constructed on the display device. When such a display is driven digitally at native pixel count, there is a one-to-one relationship between framebuffer pixels and device pixels. However, a graphic subsystem may resample by primitive means when faced with a mismatch between framebuffer pixel count and display device pixel count. If framebuffer count is higher, pixels are dropped; if lower, pixels are replicated. In both instances, image quality suffers.

CRT displays typically have a Gaussian distribution of light from each pixel, as I will discuss in the next chapter. The typical spot size is such that there is some overlap in the distributions of light from adjacent pixels. You might think that overlap between the distributions of light produced by neighboring display elements is undesirable. However, image display requires a certain degree of overlap in order to minimize the visibility of pixel structure or scan-line structure. I will discuss this issue in *Image structure,* on page 75.

Two disparate measures are referred to as *resolution* in computing:

- The count of image columns and image rows – that is, columns and rows of pixels – in a framebuffer
- The number of pixels per inch (ppi) intended for image data (often misleadingly denoted dots per inch, dpi)

An image scientist considers *resolution* to be delivered to the viewer; resolution is properly estimated from information displayed at the display surface (or screen) itself. The two measures above all limit resolution, but neither of them quantifies resolution directly. In *Resolution,* on page 97, I will describe how the term is used in image science and video.

This page intentionally left blank

Image structure 7

A naïve approach to digital imaging treats an image as a matrix of independent pixels, disregarding the spatial distribution of light power across each pixel. You might think that optimum image quality is obtained when there is no overlap between the distributions of neighboring pixels; many computer engineers hold this view. However, continuous-tone images are best reproduced with a certain degree of overlap between pixels; sharpness is reduced slightly, but pixel structure is made less visible and image quality is improved.

Don't confuse point spread function (PSF) with *progressive segmented-frame* (PsF), to be described on page 94.

The distribution of intensity across a displayed pixel is referred to as its *point spread function* (PSF). A one-dimensional slice through the center of a PSF is colloquially called a *spot profile.* A display's PSF influences the nature of the images it reproduces. The effects of a PSF can be analyzed using filter theory, discussed for one dimension in the chapter *Filtering and sampling,* on page 191, and for two dimensions in *Image digitization and reconstruction,* on page 237.

Historically, the PSFs of greyscale ("black-and-white") CRTs were roughly Gaussian in shape: Intensity distribution peaked at the center of the pixel, fell off over a small distance, and overlapped neighboring pixels to some extent. The scanning spot of colour CRTs had this shape, too; but the PSF was influenced by the shadow mask or aperture grille. The introduction of direct-view colour CRTs shifted the requirement for spatial filtering to the viewer: The assumption was introduced that the viewers were sufficiently distant from the screens that the viewers' visual systems would perform the spatial integration necessary to obscure the triad structure.

Modern direct view fixed-pixel displays (FPDs) such as LCD and PDP displays have more or less uniform light emission over most of the area corresponding to each colour component (subpixel); their modulated light has a spatial structure comparable to that of a direct-view colour CRT, and similarly depends upon the viewers being located at a sufficient distance that their visual characteristics perform the spatial integration necessary to obscure the triad structure.

A pixel whose intensity distribution uniformly covers a small square area of the screen has a point spread function referred to as a "box."

Image reconstruction

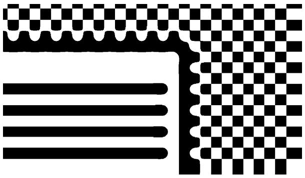


Figure 7.1 **"Box" reconstruction** of a bitmapped graphic image is shown.

Figure 7.1 reproduces a portion of an idealized bitmapped (bilevel) graphic image, part of a computer's desktop display. Each sample is either black or white. The element with horizontal "stripes" is part of a window's titlebar; the checkerboard background is intended to integrate to grey. Figure 7.1 shows reconstruction of the image with a "box" distribution. Each pixel is uniformly shaded across its extent; there is no overlap between pixels. This figure exemplifies a *raster-locked* image as displayed on an LCD. By *raster-locked,* I refer to image data having the underlying image elements aligned with the pixel array.

Figure 7.2 **Gaussian reconstruction** is shown for the same bitmapped image as Figure 7.1. I will detail the one-dimensional *AGaussian function* on page 200.

A CRT's electron gun produces an electron beam that illuminates a spot on the phosphor screen. The beam is deflected to form a raster pattern of scan lines that traces the entire screen, as I will describe in the following chapter. The beam is not perfectly focused when it is emitted from the CRT's electron gun, and is dispersed further in transit to the phosphor screen. The intensity produced for each pixel at the face of the screen has a "bell-shaped" distribution resembling a two-dimensional Gaussian function. With a typical amount of spot overlap, the checkerboard area of this example will display as a nearly uniform grey as depicted in Figure 7.2. You might think that the blur caused by overlap between pixels would diminish image quality. However, for continuous-tone ("contone") images, some degree of overlap is not only desirable but necessary, as you will see from the following examples.

Figure 7.3 **Diagonal line reconstruction.** At the left is a near-vertical line slightly more than 1 pixel wide, rendered as an array 20 pixels high that has been reconstructed using a box distribution. At the right, the line is reconstructed using a Gaussian distribution. Between the images I have placed a set of markers to indicate the vertical centers of the image rows.

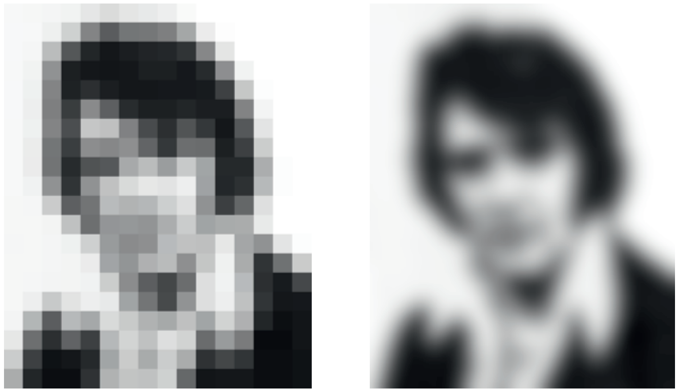


Figure 7.4 **Contone image reconstruction.** At the left is a continuous-tone image of 16×20 pixels that has been reconstructed using a box distribution. The pictured individual cannot be recognized. At the right is exactly the same image data, but reconstructed by a Gaussian function. The reconstructed image is very blurry but recognizable. Which reconstruction function do you think is best for continuous-tone imaging?

Figure 7.3 shows a 16×20-pixel image of a dark line slightly more than one pixel wide, 7.2° off the vertical. At the left, the image data is reconstructed using a box distribution; a jagged and "ropey" nature is evident. At the right, the image data is reconstructed using a Gaussian. It is blurry, but less jagged.

Figure 7.4 shows two ways to reconstruct the same 16×20 pixels (320 bytes) of continuous-tone greyscale image data. The left-hand image is reconstructed using a box function, and the right-hand image with a Gaussian. The example was constructed so that each image is 4 cm (1.6 inches) wide. At typical reading distance of 40 cm (16 inches), a pixel subtends 0.4°, where visual acuity is near its maximum. At this distance, when reconstructed with a box function, the pixel structure of each image is highly visible; visibility of the pixel structure overwhelms the perception of the image itself. The right image is reconstructed using a Gaussian distribution. It is blurry, but easily recognizable as an American

Visual acuity is detailed in *Contrast sensitivity function (CSF)*, on page 251.

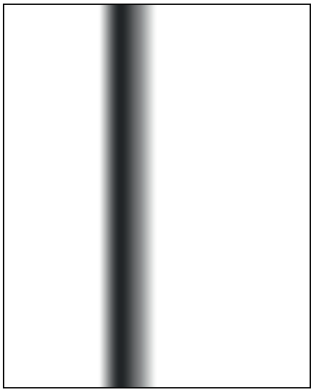


Figure 7.5 **One frame of an animated sequence,** reconstructed with a "box" filter.

cultural icon. This example shows that sharpness is not always good, and blurriness is not always bad!

Figure 7.5 in the margin shows a 16×20-pixel image comprising 20 copies of the top row of Figure 7.3 (left). Consider a sequence of 20 animated frames, where each frame is formed from successive image rows of Figure 7.3. The animation would depict a narrow vertical line drifting rightward across the screen at a rate of 1 pixel every 8 frames. If image rows of Figure 7.3 (left) were used, the width of the moving line would appear to jitter frame-to-frame, and the minimum lightness would vary. With Gaussian reconstruction, as in Figure 7.3 (right), motion portrayal is much smoother.

Sampling aperture

In a practical image sensor, each element acquires information from a finite region of the image plane; the value of each pixel is a function of the distribution of intensity over that region. The distribution of sensitivity across a pixel of an image capture device is referred to as its *sampling aperture,* sort of a PSF in reverse – you could call it a point "collection" function. The sampling aperture influences the nature of the image signal originated by a sensor. Sampling apertures used in continuous-tone imaging systems usually peak at the center of each pixel, fall off over a small distance, and overlap neighboring pixels to some extent.

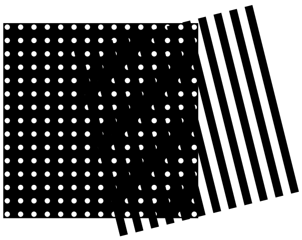


Figure 7.6 A **Moiré pattern** is a form of aliasing in two dimensions that results when a sampling pattern (here the perforated square) has a sampling density that is too low for the image content (here the dozen bars, 14° off-vertical). This figure is adapted from Fig. 3.12 of Wandell's *Foundations of Vision* (cited on page 195).

In 1915, Harry Nyquist published a landmark paper stating that a sampled analog signal cannot be reconstructed accurately unless all of its frequency components are contained strictly within half the sampling frequency. This condition subsequently became known as the *Nyquist criterion;* half the sampling rate became known as the *Nyquist rate.* Nyquist developed his theorem for one-dimensional signals, but it has been extended to two dimensions. In a digital system, it takes at least two elements – two pixels or two scanning lines – to represent a cycle. A *cycle* is equivalent to a *line pair* of film, or two "TV lines" (TVL).

In Figure 7.6, the black square punctured by a regular array of holes represents a grid of small sampling apertures. Behind the sampling grid is a set of a dozen black bars, tilted 14° off the vertical, representing image information. In the region where the image is sampled, you can see three wide dark bars tilted at 45°. Those bars represent spatial *aliases* that arise because the number of bars per inch (or mm) in the image is greater than half the number of apertures per inch (or mm) in the sampling lattice. Aliasing can be prevented – or at least minimized – by imposing a spatial filter in front of the sampling process, as I will describe for one-dimensional signals in *Filtering and sampling,* on page 191, and for two dimensions in *Image presampling filters,* on page 242.

Nyquist explained that an arbitrary signal can be reconstructed accurately only if more than two samples are taken of the highest-frequency component of the signal. Applied to an image, there must be at least twice as many samples per unit distance as there are image elements. The checkerboard pattern in Figure 7.1 (on page 76) doesn't meet this criterion in either the vertical or horizontal dimensions. Furthermore, the titlebar element doesn't meet the criterion vertically. Such elements can be represented in a bilevel image only when they are in precise registration – "locked" – to the imaging system's sampling grid. However, images captured from reality almost never have their elements precisely aligned with the grid!

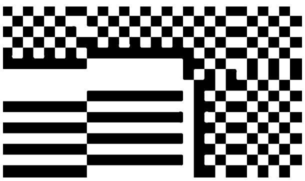


Figure 7.7 **Bitmapped graphic image, rotated.**

Point sampling refers to capture with an infinitesimal sampling aperture. This is undesirable in continuous-tone imaging. Figure 7.7 shows what would happen if a physical scene like that in Figure 7.1 were rotated 14°, captured with a point-sampled camera, and displayed with a box distribution. The alternating on-off elements are rendered with aliasing in both the checkerboard portion and the titlebar. (Aliasing would be evident even if this image were to be reconstructed with a Gaussian.) This example emphasizes that in digital imaging, we must represent arbitrary scenes, not just scenes whose elements have an intimate relationship with the sampling grid.

A suitable presampling filter would prevent (or at least minimize) the Moiré artifact of Figure 7.6, and prevent or minimize the aliasing of Figure 7.7. When image content such as the example titlebar and the desktop pattern of Figure 7.2 is presented to a presampling filter, blurring will occur. Considering only bitmapped images such as Figure 7.1, you might think the blurring to be detrimental, but to avoid spatial aliasing in capturing high-quality continuous-tone imagery, some overlap is necessary in the distribution of sensitivity across neighboring sensor elements.

Having introduced the aliasing artifact that results from poor capture PSFs, we can now return to the display and discuss reconstruction PSFs (spot profiles).

Spot profile

The designer of a display system for continuous-tone images seeks to make a display that allows viewing at a wide picture angle, with minimal intrusion of artifacts such as aliasing or visible scan-line or pixel structure. Picture size, viewing distance, spot profile, and scan-line or pixel visibility all interact. The display system designer cannot exert direct control over viewing distance; spot profile is the parameter available for optimization.

On page 77, I demonstrated the difference between a box profile and a Gaussian profile. Figures 7.3 and 7.4 showed that some overlap between neighboring distributions is desirable, even though blur is evident when the reproduced image is viewed closely.

When the images of Figure 7.3 or 7.4 are viewed from a distance of 10 m (33 feet), a pixel subtends a minute of arc ($^1/_{60}$°). At this distance, owing to the limited acuity of human vision, both pairs of images are apparently identical. Imagine placing beside these images an emissive display having an infinitesimal spot, producing the same total flux for a perfectly white pixel. At 10 m, the pixel structure of the emissive display would be somewhat visible. At a great viewing distance – say at a pixel or scan-line subtense of less than $^1/_{180}$°, corresponding to SD viewed at three times normal distance, or about 20·PH – the limited acuity of the human visual system causes all three displays to appear identical. As the viewer moves closer, different effects become apparent, depending upon spot profile. I'll discuss two cases: box distribution and Gaussian distribution.

Box distribution

A typical digital projector – such as an LCD or a PDP – has a spot profile resembling a box distribution covering nearly the entire width and nearly the entire height corresponding to the pixel pitch. There is no significant gap between image rows or image columns. Each pixel has three colour components, but the optics of the projection device are arranged to cause the distribution of light from these components to be overlaid. From a great distance, pixel structure will not be visible. However, as viewing distance decreases, aliasing ("the jaggies") will intrude. Limited performance of projection lenses mitigates aliasing somewhat; however, aliasing can be quite noticeable, as in the examples of Figures 7.3 and 7.4 on page 77.

In a typical direct-view digital display, such as an LCD or a PDP, each pixel comprises three colour components that occupy distinct regions of the area corresponding to each pixel. Ordinarily, these components are side-by-side. There is no significant gap between image rows. However, if one component (say green) is turned on and the others are off, there is a gap between columns. These systems rely upon the limited acuity of the viewer to integrate the components into a single coloured area. At a close viewing distance, the gap can be visible, and this can induce aliasing.

The viewing distance of a display using a box distribution, such as a direct-view LCD or PDP, is limited by the intrusion of aliasing.

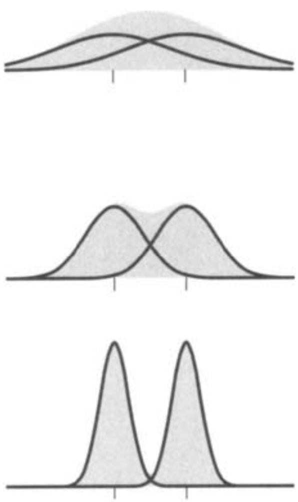


Figure 7.8 **Gaussian spot size.** Solid lines graph Gaussian distributions of intensity across two adjacent image rows, for three values of spot size. The areas under each curve are identical. The shaded areas indicate their sums. In progressive scanning, adjacent image rows correspond to consecutive scan lines. In interlaced scanning, the situation is more complex.

Gaussian distribution

As I have mentioned, a CRT display has a spot profile resembling a Gaussian. The CRT designer's choice of spot size involves a compromise illustrated by Figure 7.8.

- For a Gaussian distribution with a very small spot, say a spot width less than $1/2$ the scan-line pitch, line structure will become evident even at a fairly large viewing distance.
- For a Gaussian distribution with medium-sized spot, say a spot width approximately equal to the scan-line pitch, the onset of scan-line visibility will occur at a closer distance than with a small spot.
- As spot size is increased beyond about twice the scan-line pitch, eventually the spot becomes so large that no further improvement in line-structure visibility is achieved by making it larger. However, there is a ser-

ious disadvantage to making the spot larger than necessary: Sharpness is reduced.

You saw at the beginning of this chapter that in order to avoid visible pixel structure in image display some overlap is necessary in the distributions of light produced by neighboring display elements. Such overlap reduces sharpness, but by how much? How much overlap is necessary? I will discuss these issues in the Chapter *Resolution*, on page 97. First, though, I will introduce the fundamentals of raster scanning.

Raster scanning 8

I introduced the pixel array on page 3. This chapter outlines the basics of this process of *raster scanning*, whereby the samples of the pixel array are sequenced uniformly in time to form scan lines, which are in turn sequenced in time throughout each frame interval. In Chapter 13, *Introduction to component SD,* on page 129, I will present details on scanning in conventional "525-line" and "625-line" video. In *Introduction to composite NTSC and PAL,* on page 135, I will introduce the colour coding used in these systems. In Chapter 15, *Introduction to HD,* on page 141, I will introduce scanning in high-definition television.

Flicker, refresh rate, and frame rate

A sequence of still pictures, captured and displayed at a sufficiently high rate, can create the illusion of motion.

Flicker is sometimes redundantly called *large-area flicker*. Take care to distinguish *flicker*, described here, from *twitter*, to be described on page 89. See FUKUDA, TADAHIKO (1987), "Some Characteristics of Peripheral Vision," *NHK Tech. Monograph No.* 36 (Tokyo: NHK Science and Technical Research Laboratories).

The historical CRT display used for television emits light for a small fraction of the frame time: The display has a *short duty cycle;* it is black most of the time. If the flash rate – or *refresh* rate – is too low, flicker is perceived. The flicker sensitivity of vision is dependent upon display and viewing conditions: The brighter the environment, and the larger the angle subtended by the picture, the higher the flash rate must be to avoid flicker. Because picture angle influences flicker, flicker depends upon viewing distance.

Most modern displays – including LCDs and plasma displays – do not flash, and cannot flicker. Nonetheless, they may be subject to various motion impairments.

In a "flashing" display, the brightness of the displayed image itself influences the flicker threshold to some

Application	Display luminance	Surround	Refresh (flash) rate [Hz]	Frame rate [Hz]
Cinema	48 nt	Dark ~0%	48	24
Television	80 nt	Dim ~5%	50	25
Television	120 nt	Dim ~5%	≈60	≈30
Office	320 nt	"Average" ~20%	various, e.g., 66, 72, 76	same as refresh rate

Table 8.1 ***Refresh rate*** refers to the shortest interval over which the entire picture is updated. *Flash rate* refers to the rate at which the picture height is covered at the display. Different refresh rates and flash rates are used in different applications.

extent, so the brighter the image the higher the refresh rate must be. In a very dark environment, such as the cinema, flicker sensitivity is completely determined by the luminance of the image itself. Peripheral vision has higher temporal sensitivity than central (foveal) vision, so the flicker threshold increases to some extent with wider viewing angles. Table 8.1 summarizes refresh rates used in film, video, and computing.

The fovea has a diameter of about 1.5 mm, and subtends a visual angle of about 5°.

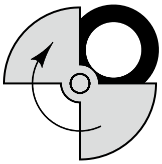


Figure 8.1 **A dual-bladed shutter** in a film projector flashes each frame twice. Rarely, three bladed shutters are used; they flash each frame thrice.

In the darkness of a cinema, a flash rate of 48 Hz is sufficient to overcome flicker. In the early days of motion pictures, 24 frames per second were found to be sufficient for good motion portrayal. So, a conventional film projector uses a dual-bladed shutter, depicted in Figure 8.1, to flash each frame twice. Higher realism can be obtained with material at 48 frames per second or higher displayed with single-bladed shutters, but such schemes are nonstandard.

Television refresh rates were originally chosen to match the local AC power line frequency. See *Frame, field, line, and sample rates,* on page 389.

In the dim viewing environment typical of television, such as a living room, a flash rate of 60 Hz suffices. The interlace technique, to be described on page 88, provides for video a function comparable to the dual-bladed shutter of a film projector: Each frame is flashed as two fields. Refresh is established by the field rate (twice the frame rate). For a given data rate, interlace doubles the apparent flash rate, and provides improved motion portrayal by doubling the temporal sampling rate. Scanning without interlace is called *progressive*.

Farrell, Joyce E., et al. (1987), "Predicting flicker thresholds for video display terminals," in *Proc. Society for Information Display* **28** (4): 449–453.

CRT computer displays used in office environments historically required refresh rates above 70 Hz to overcome flicker (see Farrell, 1987). CRTs have now been supplanted by LCD displays, which don't flicker. High refresh rates are no longer needed to avoid flicker.

Introduction to scanning

A moment ago, I outlined how refresh rate for television was chosen so as to minimize flicker. In *Viewing distance and angle,* on page 100, I will outline how spatial sampling determines the number of pixels in the image array. Video scanning represents pixels in sequential order, so as to acquire, convey, process, or display every pixel during the fixed time interval associated with each frame. In analog video, information in the image plane was scanned left to right at a uniform rate during a fixed, short interval of time – the *active line time*. Scanning established a fixed relationship between a position in the image and a time instant in the signal. Successive lines were scanned at a uniform rate from the top to the bottom of the image, so there was also a fixed relationship between vertical position and time. The stationary pattern of parallel scanning lines disposed across the image area is the *raster*.

The word *raster* is derived from the Greek word *rustum* (rake), owing to the resemblance of a raster to the pattern left on newly raked earth.

Line is a heavily overloaded term. *Lines* may refer to the total number of raster lines: Figure 8.2 shows "525-line" video, which has 525 total lines. *Line* may refer to a line containing picture, or to the total number of lines containing picture – in this example, 480. *Line* may denote the AC power line, whose frequency is very closely related to vertical scanning. Finally, *lines* is a measure of resolution, to be described in *Resolution,* on page 97.

Samples of a digital image matrix are usually conveyed in the same order that the image was historically conveyed in analog video: first the top image row (left to right), then successive rows. *Scan line* is an old-fashioned term; the term *image row* is now preferred. Successive pixels lie in *image columns.*

In cameras and displays, a certain time interval is consumed in advancing the scanning operation – historically, *horizontal retracing* – from one line to the next; several line times are consumed by vertical retrace, from the bottom of one scan to the top of the next. A CRT's electron gun had to be switched off (blanked) during these intervals, so these intervals were (and are) called *blanking intervals*. The *horizontal blanking interval* occurs between scan lines; the *vertical blanking interval* (VBI) occurs between frames (or fields). Figure 8.2 shows the blanking intervals of "525-line" video.

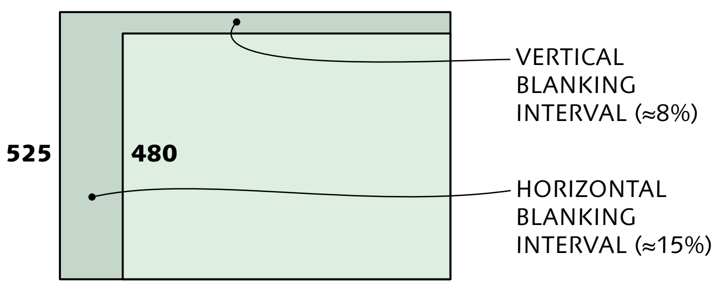


Figure 8.2 **Blanking intervals** for "525-line"video are indicated here by a dark region surrounding a light-shaded rectangle that represents the picture. The *vertical blanking interval* (VBI) consumes about 8% of each field time; horizontal blanking consumes about 15% of each line time.

The horizontal and vertical blanking intervals required for a CRT were large fractions of the line time and frame time: In SD, vertical blanking consumes roughly 8% of each frame period. In HD, that fraction is reduced to about 4%.

The count of 480 picture lines in Figure 8.2 is a recent standard; some people will quote numbers between 481 and 487. See *Picture lines,* on page 379.

In a video interface, whether analog or digital, synchronization information (*sync*) is conveyed during the blanking intervals. In principle, a digital video interface transmit just the active pixels accompanied by the minimum necessary sync information. Instead, digital video interfaces use interface rates that match the blanking intervals of historical analog equipment. What would otherwise be excess data capacity is put to good use conveying audio signals, captions, test signals, error detection or correction information, or other data or metadata.

Scanning parameters

In *progressive* (or *sequential*) *scanning*, the image rows are scanned in order, from top to bottom, at a picture rate sufficient to portray motion. Figure 8.3 at the top of the facing page indicates four basic scanning parameters:

- *Total lines* (L_T) comprises all of the scan lines, including the vertical blanking interval and the picture lines.
- The image has N_R *image rows,* containing the picture. (Historically, this was the *active line* count, L_A.)
- *Samples per total line* (S_{TL}) comprises the sample intervals in the total line, including horizontal blanking.
- The image has N_C *image columns,* containing the picture. (Historically, some number of *samples per active line* (S_{AL}) were permitted to take values different from the blanking level.

The *production aperture*, sketched in Figure 8.3, comprises the array N_C columns by N_R rows. The samples in the production aperture comprise the pixel array; they are *active*. All other sample intervals comprise blanking; they are *inactive* or *"blanked."*

VITS: Vertical interval test signal
VITC: Vertical interval timecode

The vertical blanking interval in analog signals typically carried vertical interval information such as VITS, VITC, and closed captions. Consumer display equipment must blank these lines. Both vertical and horizontal blanking intervals in a digital video interface may be used to convey ancillary (ANC) data such as audio.

Figure 8.3 **The Production aperture** comprises the image array, N_C columns by N_R rows. Blanking intervals lie outside the production aperture; here, blanking intervals are darkly shaded. The product of N_C and N_R yields the active pixel count per frame. Sampling rate (f_S) is the product of S_{TL}, L_T, and frame rate.

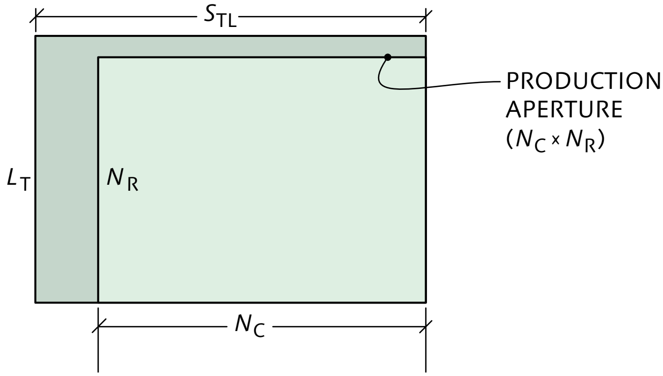
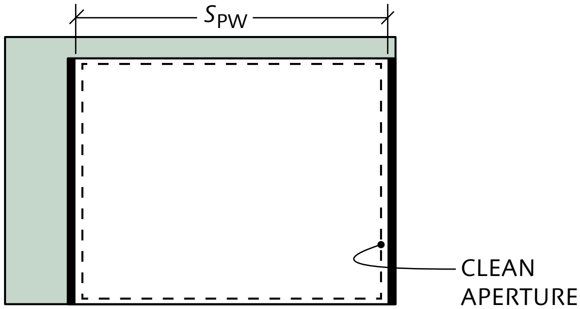


Figure 8.4 **The Clean aperture** should remain subjectively free from artifacts arising from filtering. The clean aperture excludes blanking transition samples, indicated here by black bands outside the left and right edges of the picture width, defined by the count of samples per picture width (S_{PW}).



All standard SD and HD image formats have N_C and N_R both even. The horizontal center of the picture lies midway between the central two luma samples, and the vertical center of the picture lies vertically midway between the central two image rows.

Only pixels in the image array are represented in acquisition, processing, storage, and display. However, some processing operations (such as spatial filtering) use information in a small neighbourhood surrounding the subject pixel. In the absence of any better information, we take the pixel of the image array to lie on black. At the left-hand edge of the picture, if the video signal of the leftmost pixel has a value greatly different from black, an artifact called *ringing* is liable to result when that transition is processed through an analog or digital filter. A similar circumstance arises at the right-hand picture edge. In studio video, the signal builds to full amplitude, or decays to blanking level, over several *transition* samples ideally having a raised cosine envelope.

See *Transition samples,* on page 378.

Active samples encompass not only the picture, but also the transition samples; see Figure 8.4 above. Studio equipment should maintain the widest picture possible within the production aperture, subject to appropriate blanking transitions.

I have treated the image array as an array of pixels, without regard for the spatial distribution of light

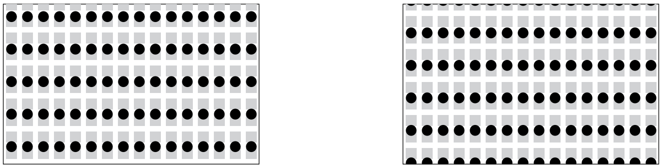


Figure 8.5 **Interlaced format** represents a complete picture – the *frame* – from two *fields,* each containing half of the total number of image rows. The second field is delayed by half the frame time from the first. This example shows 10 image rows. In analog scanning, interlace is effected by having an odd number of *total* scan lines (e.,g., 525, 625, or 1125).

power across each pixel – the pixel's *spot profile,* or more technically, *point spread function* (PSF). If the spot profile is such that there is a significant gap between the intensity distributions of adjacent image rows (scan lines), then image structure will be visible to viewers closer than a certain distance. The gap between scan lines is a function of image row (scan-line) pitch and spot profile. *Spot size* was historically characterized by spot diameter at 50% power. For a given image row pitch, a smaller spot size will force viewers to be more distant from the display if scan lines are to be rendered invisible.

Interlaced format

To refer to fields as *odd* and *even* invites confusion. Use *first field* and *second field* instead. Some people refer to scanning first the odd lines then the even; however, scan lines in interlaced video were historically numbered in *temporal* order, not spatial order: Scan lines are *not* numbered as if they were rows in the frame's image matrix. Confusion on this point among computer engineers – and confusion regarding *top* and *bottom fields* – has led to lots of improperly encoded video where the top and bottom offsets are wrong.

Interlacing is a scheme which – for given viewing distance, flicker sensitivity, and data rate – offered some increase in static spatial resolution over progressive scanning in historical CRT displays, which exhibited flicker. The full height of the image is scanned leaving gaps in the vertical direction. Then, $1/50$ or $1/60$ s later, the full image height is scanned again, but offset vertically so as to fill in the gaps. A frame thereby comprises two *fields,* denoted *first* and *second.* The scanning mechanism is depicted in Figure 8.5. Historically, the same scanning standard was used across an entire television system, so interlace was used not only for display but for the whole chain, including acquisition, recording, processing, distribution, and transmission.

Noninterlaced (*progressive*) scanning is universal in desktop computers and in computing; also, progressive scanning has been introduced for digital television and

HD. However, the interlace technique remains universal in SD, and is widely used in broadcast HD. Interlace-to-progressive (I-P) conversion, also called *deinterlacing,* is an unfortunate but necessary by-product of interlaced scanning.

CRTs are now obsolete. The dominant display technologies now used for video – LCD and plasma panels – have relatively long duty cycles, and they don't flicker. The *raison d'être* for interlace has vanished. Nonetheless, interlace remains in wide use.

Twitter

The flicker susceptibility of vision stems from a wide-area effect: In a display such as a CRT that flashes, as long as the complete height of the picture is flashed sufficiently rapidly to overcome flicker, small-scale picture detail, such as that in the alternate lines, can be transmitted at a lower rate. With progressive scanning, scan-line visibility limits the reduction of spot size. With interlaced scanning, this constraint is relaxed by a factor of two. However, interlace introduced a new constraint, that of *twitter.*

Figure 8.6 **Twitter** would result if this scene were scanned at the indicated line pitch by a camera without vertical filtering, then displayed using interlace on a short duty cycle display such as a CRT.

If an image has vertical detail at a scale comparable to the image row pitch – for example, if the fine pattern of horizontal line pairs in Figure 8.6 is scanned – then interlaced display causes the content of the first and the second fields to differ markedly. At usual field rates – 50 or 60 Hz – this causes twitter, a small-scale phenomenon that is perceived as a scintillation, or an extremely rapid up-and-down motion. If such image information occupies a large area, then flicker is perceived instead of twitter. Twitter is sometimes called *interline flicker*; however, flicker is by definition a wide-area effect, so *interline flicker* is a poor term.

Twitter is produced not only from degenerate images such as the fine black-and-white lines of Figure 8.6, but also from high-contrast vertical detail in ordinary images. High-quality video cameras include optical spatial lowpass filters to attenuate vertical detail that would otherwise be liable to produce twitter. When computer-generated imagery (CGI) is interlaced, vertical detail must be filtered in order to avoid flicker. Signal processing to accomplish this is called a *twitter filter*.

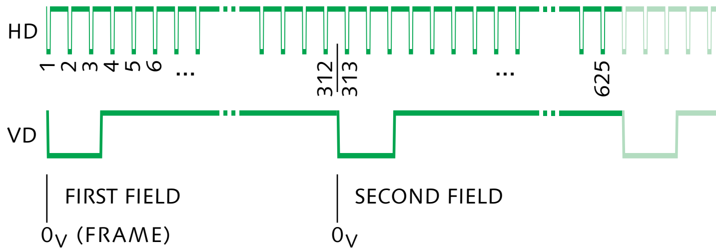


Figure 8.7 **Horizontal and vertical drive** pulses historically effected interlace in analog scanning. 0_V denotes the start of each field. The halfline offset of the second 0_V causes interlace. Here, 576*i* scanning is shown.

Interlace in analog systems

In analog video, interlace was historically achieved by scanning vertically at a constant rate, typically 50 or 60 Hz, and scanning horizontally at an odd multiple of half that rate. In SD in North America and Japan, the field rate is 59.94 Hz; the line rate (f_H) is $\frac{525}{2}$ ($262\frac{1}{2}$) times that rate. In Asia, Australia, and Europe, the field rate is 50 Hz; the line rate is $\frac{625}{2}$ ($312\frac{1}{2}$) times that rate.

Details are presented in Chapter 2, *Analog SD sync, genlock, and interface*, in *Composite NTSC and PAL: Legacy Video Systems*, and in the first edition of the present book.

Figure 8.7 shows the *horizontal drive* (HD) and *vertical drive* (VD) pulse signals that were once distributed in the studio to cause interlaced scanning in analog equipment. These signals have been superseded by a combined *sync* (or *composite sync*) signal; vertical scanning is triggered by *broad pulses* having total duration of $2\frac{1}{2}$ or 3 lines. Sync is usually imposed onto the video signal, to avoid separate distribution circuits. Analog sync is coded at a level "blacker than black."

Interlace and progressive

We'll take up resolution in interlaced systems on *Interlace revisited*, on page 105.

For a given viewing distance, sharpness is improved as spot size becomes smaller. However, if spot size is reduced beyond a certain point, depending upon the spot profile of the display, either scan lines or pixels will become visible, or aliasing will intrude. In principle, improvements in bandwidth or spot profile reduce potential viewing distance, enabling a wider picture angle. However, because consumers form expectations about viewing distance, we assume a constant viewing distance and say that *resolution* is improved instead.

A rough conceptual comparison of progressive and interlaced scanning is presented in Figure 8.8 at the top of the facing page. At first glance, an interlaced system offers twice the number of pixels – loosely, twice the

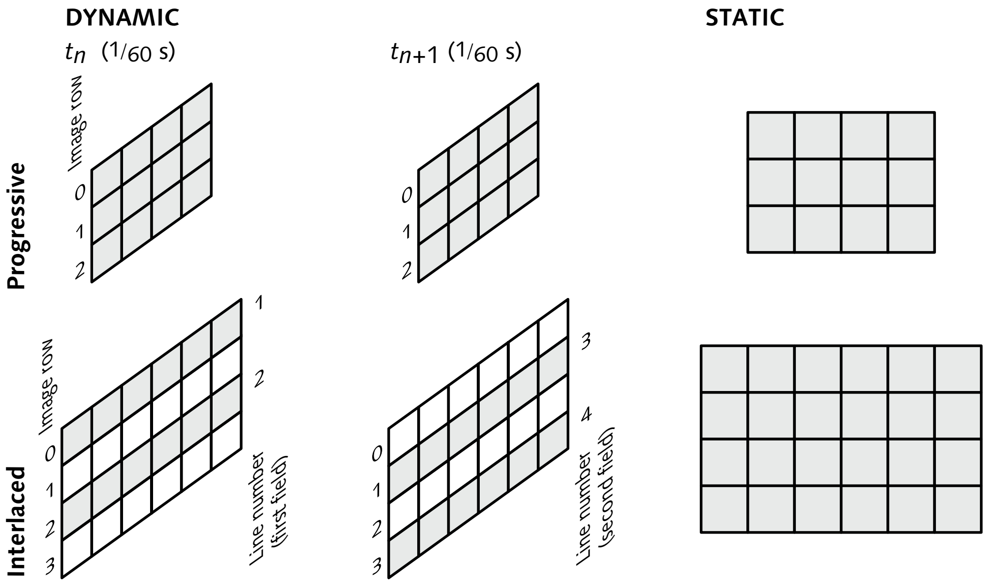


Figure 8.8 **Progressive and interlaced scanning** are compared. The top left sketch depicts an image of 4×3 pixels transmitted during an interval of 1⁄60 s. The top center sketch shows image data from the same 12 locations transmitted in the following 1⁄60 s interval. The top right sketch. shows the spatial arrangement of the 4×3 image, totalling 12 pixels; the data rate is 12 pixels per 1⁄60 s. At the bottom left, 12 pixels comprising image rows 0 and 2 of a 6×4 image array are transmitted in 1⁄60 s. At the bottom center, the 12 pixels of image rows 1 and 3 are transmitted in the following 1⁄60 s interval. At the bottom right, the spatial arrangement of the 6×4 image is shown: The 24 pixel image is transmitted in 1⁄30 s. Interlaced scanning has the same data rate as progressive, but at first glance has twice the number of pixels, and potentially twice the resolution. In practice, the improvement is a factor of about 1.4 – about 1.2 horizontally and 1.2 verticallly.

spatial resolution – as a progressive system with the same data capacity and the same frame rate. Owing to twitter, spatial resolution in a practical interlaced system is not double that of a progressive system at the same data rate. Historically, cameras have been designed to avoid producing so much vertical detail that twitter would be objectionable. However, resolution is increased by a factor large enough that interlace has historically been considered worthwhile. The improvement comes at the expense of introducing some aliasing and some vertical motion artifacts. Also, interlace makes it difficult to process motion sequences, as will be explained on page 93.

Examine the interlaced (bottom) portion of Figure 8.8, and imagine an image element moving slowly down the picture at a rate of one row of the pixel array every field time – in a 480*i*29.97 system, $\frac{1}{480}$ of the picture height in $\frac{1}{60}$ s, or one picture height in 8 seconds. Owing to interlace, half of that image's vertical information will be lost! At other rates, some portion of the vertical detail in the image will be lost. With interlaced scanning, vertical motion can cause serious motion artifacts. Techniques to avoid such artifacts will be discussed in *Deinterlacing,* on page 413.

Notation	*Pixel array*
VGA	640×480
SVGA	800×600
XGA	1024×768
WXGA	1366×768
SXGA	1280×1024
UXGA	1600×1200
WUXGA	1920×1200
QXGA	2048×1536

Table 8.2 **Scanning in computing** has no standardized notation, but these notations are widely used.

Scanning notation

In computing, a display format may be denoted by a pair of numbers: the count of pixels across the width of the image – that is, the count of image columns – and the count of pixels down the height of the image – that is, the count of image rows. Square sampling is implicit. This notation does not indicate refresh rate. (Alternatively, a display format may be denoted symbolically – VGA, SVGA, XGA, etc., as in Table 8.2. This notation is utterly opaque to the majority of computer users.)

Traditionally, video scanning was denoted by the total number of lines per *frame* (picture lines plus sync and vertical blanking overhead), a slash, and the *field* rate in hertz. (Interlace was implicit unless a slash and *1:1* was appended to indicate progressive scanning; a slash and *2:1* make interlace explicit.) For SD, 525/59.94/2:1 scanning is used in North America and Japan; 625/50/2:1 prevails in Europe, Asia, and Australia. Before the advent of HD, these were the only scanning systems used for broadcasting.

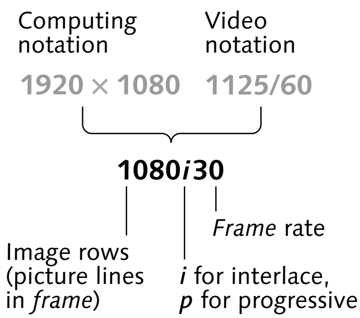


Figure 8.9 **Modern image format notation** gives the count of image rows (active/picture lines); *p* for progressive or *i* for interlace (or *psf* for progressive segmented-frame); then a rate. I write frame rate, but some people write field rate. Aspect ratio is not part of the notation.

Digital technology enabled new scanning and display standards. Conventional scanning notation could not adequately describe the new scanning systems. A new notation, depicted in Figure 8.9, emerged: Scanning is denoted by the count of image rows, followed by *p* for progressive or *i* for interlace, followed by the frame rate. I write the letter *i* in lowercase italics to avoid potential confusion with the digit 1. For consistency, I write the letter *p* in lowercase italics. Traditional video notation (such as 625/50) is inconsistent, juxtaposing lines per *frame* with *fields* per second. Some people – particularly historical proponents of interlaced scanning – seem intent upon carrying this confusion into the future, by denoting the old 525/59.94 as 480*i*59.94. I strongly prefer using frame rate.

Since all 480*i* systems have a frame rate of 29.97 Hz, I use 480*i* as shorthand for 480*i*29.97. Similarly, I use 576*i* as shorthand for 576*i*25.

Because some people write 480*p*60 when they mean 480*p*59.94, the notation *60.00* should be used to emphasize a rate of exactly 60 Hz.

In my notation, conventional 525/59.94/2:1 video is denoted 480*i*29.97; conventional 625/50/2:1 video is denoted 576*i*25. HD systems include 720*p*60 and 1080*i*30. Film-friendly versions of HD are denoted 720*p*24 and 1080*p*24. Aspect ratio is not explicit in the new notation: 720*p*, 1080*i*, and 1080*p* are implicitly 16:9 since there are no 4:3 standards for these systems, but 480*i* or 480*p* could potentially have either conventional 4:3 or widescreen 16:9 aspect ratio.

Motion portrayal

In *Flicker, refresh rate, and frame rate,* on page 83, I outlined the perceptual considerations in choosing refresh rate. In order to avoid objectionable flicker, it was historically necessary with CRT displays to flash an image at a rate higher than the rate necessary to portray its motion. Different applications have adopted different refresh rates, depending on the image quality requirements and viewing conditions. Refresh rate is generally engineered into a video system; once chosen, it cannot easily be changed.

Flicker is minimized by any display device that produces steady, unflashing light, or pulsed light lasting for the majority of the frame time. You might regard a nonflashing display to be more suitable than a device that flashes; many modern devices do not flash. However, with a display having a pixel duty cycle near 100% – that is, an *on-time* approaching the frame time – if the viewer's gaze tracks an element that moves across the image, that element will be seen as smeared. This problem becomes more severe as eye tracking velocities increase, such as with the wide viewing angle of HD. (For details of temporal characteristics of image acquisition and display, see my SMPTE paper.)

Poynton, Charles (1996), "Motion portrayal, eye tracking, and emerging display technology," in *Proc. 30th SMPTE Advanced Motion Imaging Conference* (New York: SMPTE), 192–202.

Film at 24 frames per second is transferred to interlaced video at 60 fields per second by *2-3 pulldown*. The first film frame is transferred to two video fields, then the second film frame is transferred to three video fields; the cycle repeats. The 2-3 pulldown is normally used to produce video at 59.94 Hz, not 60 Hz; the film is run 0.1% slower than 24 frames per second. The scheme is detailed in *2-3 pulldown,* on page 405. The 2-3 technique can be applied to transfer to progressive video at 59.94 or 60 frames per second.

Historically, this technique was called *3-2 pulldown*, but since the adoption of SMPTE RP 197 in 1998, it is now more accurately called *2-3 pulldown*.

Film at 24 frames per second is transferred to 576*i* video using *2-2 pulldown:* Each film frame is scanned into two video fields (or frames); the film is run 4% faster than the capture rate. The process is described more fully in Chapter 34, *2-3 pulldown,* on page 405.

Segmented-frame (24PsF)

The *progressive segmented-frame* (PsF) technique is known in consumer SD systems as *quasi-interlace*. PsF is not to be confused with *point spread function*, PSF.

Some camera sensors sample the entire scene at the same instant (*global shutter,* GS). With other sensors, sampling of the scene proceeds vertically at a uniform rate through the frame time (*rolling shutter,* RS).

A scheme called *progressive segmented-frame* is sometimes used in HD equipment that handles images at 24 frames per second. The scheme, denoted *24PsF*, uses progressive scanning: The entire image is sampled at once, and vertical filtering to reduce twitter is both unnecessary and undesirable. However, lines are rearranged to interlaced order for studio distribution and recording. The scheme offers some compatibility with interlaced processing and recording equipment.

Video system taxonomy

Insufficient channel capacity was available at the outset of television broadcasting to transmit three separate colour components. The NTSC and PAL techniques were devised to combine (*encode*) the three colour components into a single *composite* signal. Composite video remains in use in consumers' premises. However, modern video equipment – including all consumer digital video equipment, and all HD equipment – uses component video, either $Y'P_BP_R$ analog components or $Y'C_BC_R$ digital components.

A video system can be classified as component HD, component SD, or composite SD. Independently, a system can be classified as analog or digital. Table 8.3 indicates the six classifications, with the associated

Table 8.3 **Video systems are classified** as analog or digital, and component or composite (or S-video). SD may be represented in component, hybrid (S-video), or composite forms. HD is always in component form.

		Analog	*Digital*
HD	*Component*	*R'G'B'*, $^{709}Y'P_BP_R$	4:2:2 $^{709}Y'C_BC_R$
SD	*Component*	*R'G'B'*, $^{601}Y'P_BP_R$	4:2:2 $^{601}Y'C_BC_R$
	Hybrid	S-video	
	Composite	NTSC, PAL	$4f_{SC}$ (obsolete)

colour encoding schemes. Composite NTSC and PAL video encoding is used only in 480*i* and 576*i* systems; HD systems use only component video. S-video is a hybrid of component analog video and composite analog NTSC or PAL; in Table 8.3, S-video is classified in its own seventh (hybrid) category.

Conversion among systems

By NTSC, I do not mean 525/59.94 or 480*i*; by PAL, I do not mean 625/50 or 576*i*! See *Introduction to composite NTSC and PAL,* on page 135. Although SECAM is a composite technique in that luma and chroma are combined, it has little in common with NTSC and PAL. SECAM is obsolete.

In video, *encoding* traditionally referred to converting a set of $R'G'B'$ components into an NTSC or PAL composite signal. Encoding may start with $R'G'B'$, $Y'C_BC_R$, or $Y'P_BP_R$ components, or may involve matrixing from $R'G'B'$ to form luma (Y') and intermediate [U, V] components. Quadrature modulation then forms modulated chroma (C); luma and chroma are then summed. *Decoding* historically referred to converting an NTSC or PAL composite signal to $R'G'B'$. Decoding involves luma/chroma separation, quadrature demodulation to recover [U, V], then scaling to recover [C_B, C_R] or [P_B, P_R], or matrixing of luma and chroma to recover $R'G'B'$. *Encoding* and *decoding* have now become general terms that may refer to JPEG, M-JPEG, MPEG, or other encoding or decoding processes.

Transcoding refers to the technical aspects of conversion; signal modifications for creative purposes are not encompassed by the term.

Transcoding traditionally referred to conversion among different colour encoding methods having the same scanning standard. Transcoding of component video involves chroma interpolation, matrixing, and chroma subsampling. Transcoding of composite video involves decoding, then reencoding to the other standard. With the emergence of compressed storage and digital distribution, the term *transcoding* is now applied toward various methods of recoding compressed bitstreams, or decompressing then recompressing.

Scan conversion refers to conversion among scanning standards having different spatial structures, without the use of temporal processing. In consumer video and desktop video, scan conversion is commonly called *scaling.*

If the input and output frame rates differ, the operation is said to involve *frame rate conversion.* Motion portrayal is liable to be impaired.

In radio frequency (RF) technology, *upconversion* refers to conversion of a signal to a higher carrier frequency; *downconversion* refers to conversion of a signal to a lower carrier frequency.

Historically, *upconversion* referred to conversion from SD to HD; *downconversion* referred to conversion from HD to SD. Historically, these terms referred to conversion of a signal at the same frame rate as the input; nowadays, frame rate conversion might be involved. High-quality upconversion and downconversion require spatial interpolation. That, in turn, is best performed in a progressive format: If the source is interlaced, intermediate deinterlacing is required, even if the target format is interlaced.

Watkinson, John (1994), *The Engineer's Guide to Standards Conversion* (Petersfield, Hampshire, England: Snell & Wilcox, now Snell Group).

Standards conversion is the historical term denoting conversion among scanning standards having different frame rates. Historically, the term implied similar pixel count (such as conversion between 480*i* and 576*i*), but nowadays a standards converter might incorporate upconversion or downconversion. Standards conversion requires a fieldstore or framestore; to achieve high quality, it requires several fieldstores and motion-compensated interpolation. The complexity of standards conversion between different frame rates has made it difficult for broadcasters and consumers to convert European material for use in North America or Japan, or vice versa.

Resolution 9

To avoid visible pixel structure in image display, some overlap is desirable in the distributions of light produced by neighboring display elements, as I explained in *Image structure,* on page 75. Also, to avoid spatial aliasing in image capture, some overlap is necessary in the distribution of sensitivity across neighboring sensor elements. Such overlap reduces sharpness, but is beneficial to continuous-tone imagery. In this chapter, I will explain *resolution,* which is closely related to sharpness.

Resolution is an overloaded and ambiguous term that properly refers to spatial phenomena. It is confusing to refer to a sample as having 8-bit *resolution;* use *precision* or *quantization* instead. In computing, it is usual to use the term "resolution" to specify the number of columns and rows in the image matrix – that is, to express pixel count. That use disregards effects of signal processing. To preempt *resolution* for *pixel count* makes it difficult to discuss the image detail that's actually represented or delivered to the viewer. I'll present the details of resolution, but first I must introduce the concepts of *magnitude frequency response* and *bandwidth*.

Magnitude frequency response and bandwidth

To characterize the acquisition, processing, or display of smalll elements, rather than analyzing an element of certain (small) dimensions, we analyze a group of closely spaced identical elements, characterizing the spacing between the elements. This allows mathematical analysis using *transforms*, particularly the Fourier transform and the *z*-transform.

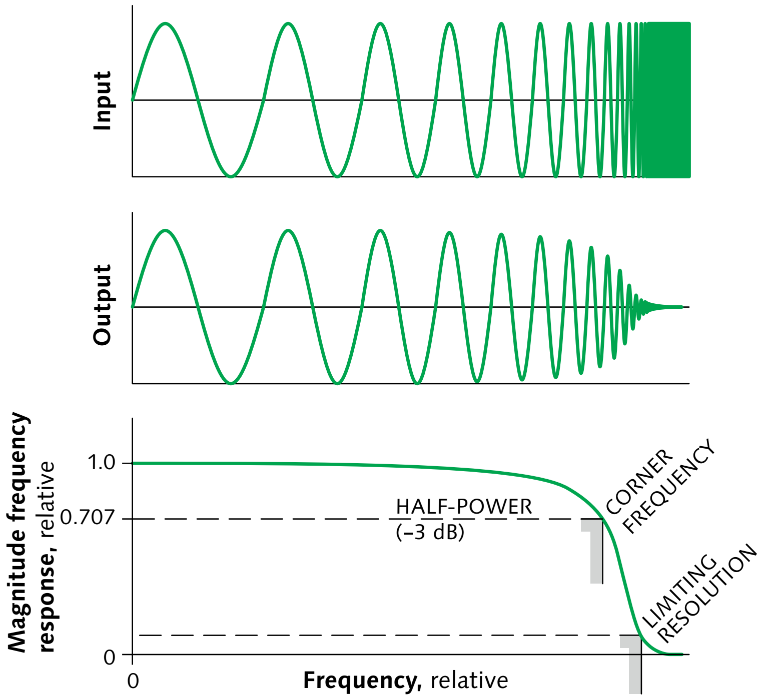


Figure 9.1 **Magnitude frequency response** of an electronic or optical system typically falls as frequency increases. Bandwidth is measured at the half-power point (–3 dB), where response has fallen to 0.707 of its value at a reference frequency (often zero frequency, or *DC*). Useful visible detail is obtained from signal power beyond the half-power bandwidth, that is, at depths of modulation less than 70.7%. I show *limiting resolution,* which might occur at about 10% response.

The top graph in Figure 9.1 shows a one-dimensional sine wave test signal "sweeping" from zero frequency up to a high frequency. (This could be a one-dimensional function of time such as an audio waveform, or the waveform of luma from one scan line of an image.) A typical optical or electronic imaging system involves temporal or spatial dispersion, which causes the response of the system to diminish at high frequency, as shown in the middle graph. The envelope of that waveform – the system's *magnitude frequency response* – is shown at the bottom. An electrical engineer may call this simply *frequency response.* The qualifier *magnitude* distinguishes it from other functions of frequency such as phase frequency response.

Bandwidth characterizes the range of frequencies that a system can capture, record, process, or transmit. *Half-power bandwidth* (also known as *3 dB bandwidth*) is specified or measured where signal magnitude has fallen 3 dB – that is, to the fraction 0.707 – from its value at a reference frequency (often zero frequency, or *DC*). Useful visual information is typically available at frequencies higher than the bandwidth. In image science, *limiting resolution* is determined visually.

There are other definitions of bandwidth than the one I present here, but this is the definition that I recommend. In *magnitude squared response*, the half-power point is at abcissa value 0.5.

The maximum rate at which an analog or digital electronic signal can change state – in an imaging system, between black and white – is limited by frequency response, and is therefore characterized by bandwidth.

Figure 9.1 shows abstract input and output signals. When bandwidth of an optical system is discussed, it is implicit that the quantities are proportional to intensity. When bandwidth of video signals is discussed, it is implicit that the input and output electrical signals are gamma-corrected.

Many digital technologists use the term *bandwidth* to refer to *data rate;* however, the terms properly refer to different concepts. *Bandwidth* refers to the frequency of signal content in an analog or digital signal. *Data rate* refers to digital transmission capacity, independent of any potential signal content. A typical studio HD luma signal has 30 MHz signal bandwidth and 74.25 MB/s data rate – the terms are obviously not interchangeable.

When digital information is processed or transmitted through analog channels, bits are coded into *symbols* that ideally remain independent. Dispersion in this context is called *intersymbol interference* (*ISI*).

Visual acuity

When an optometrist measures your visual acuity, he or she uses a chart similar to the one shown in Figure 9.2 in the margin. The results of this test depend upon viewing distance. The test is standardized for a viewing distance of 20 feet. At that distance, the strokes of the letters in the 20/20 row subtend one sixtieth of a degree ($\frac{1}{60}°$, one minute of arc). This is roughly the limit of angular discrimination of normal vision.

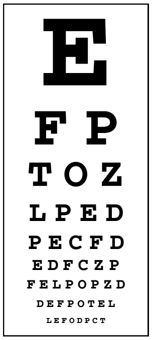


Figure 9.2
Snellen chart

Visual angles can be estimated using the astronomers' rule of thumb depicted in Figure 9.3: When held at arm's length, the joint of the thumb subtends about two degrees. The full palm subtends about ten degrees, and the nail of the little finger subtends about one degree. (The angular subtense of the full moon is about half a degree.)

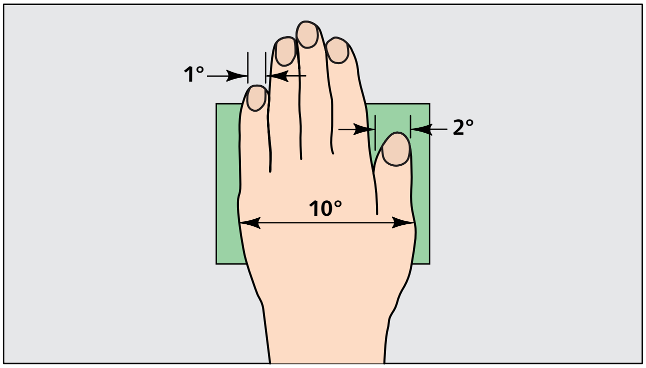


Figure 9.3 **The astronomers' rule of thumb** allows rough measurement of subtended angles. The hand is held at arm's length; the palm then subtends about 10°. Here I show the palm covering a rectangle having 4:3 aspect ratio. If that rectangle was an SD picture, the viewer would be located at roughly the optimal viewing distance.

Viewing distance and angle

If you display a white flatfield on a display with typical pixel pitch, pixel structure is likely to be visible if the viewer is located closer than the distance where adjacent image rows (or scan lines) at the display surface subtend an angle of one minute of arc ($1/60°$) or more.

To achieve viewing where pixel pitch subtends $1/60°$, viewing distance should be about 3400 times the distance *d* between image rows – that is, 3400 divided by the pixel density. For example, for pixels per inch (ppi):

$$\mathit{distance} \approx 3400 \cdot d \approx \frac{3400}{ppi}; \quad 3400 \approx \frac{1}{\sin\left(\frac{1}{60}\right)^{\circ}} \qquad \text{Eq 9.1}$$

So, at a distance of 3400 times the distance between image rows, there are about 60 pixels per degree. Viewing distance expressed numerically as a multiple of picture height is then 3400 divided by the number of image rows (N_R):

$$\mathit{distance} \approx \frac{3400}{L_A}\,\mathrm{PH} \qquad \text{Eq 9.2}$$

SD has about 480 image rows (picture lines). An image row subtends $1/60°$ at a distance of about seven times picture height (PH), as sketched in Figure 9.4 at the top of the facing page, giving roughly 600 pixels across the picture width. Picture angle is about 11°, as shown in Figure 9.5. With your hand held at arm's length, your palm ought to just cover the width of the picture. This

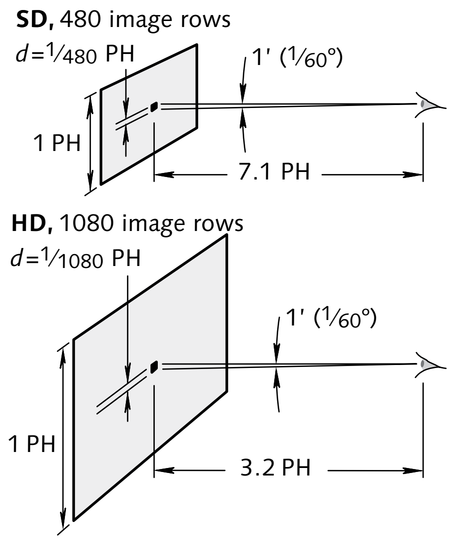


Figure 9.4 **The viewing distance** where pixels become invisible occurs approximately where the pixel pitch subtends an angle of about one minute of arc ($1/60°$) at the display surface. This is roughly the limit of angular discrimination for normal vision.

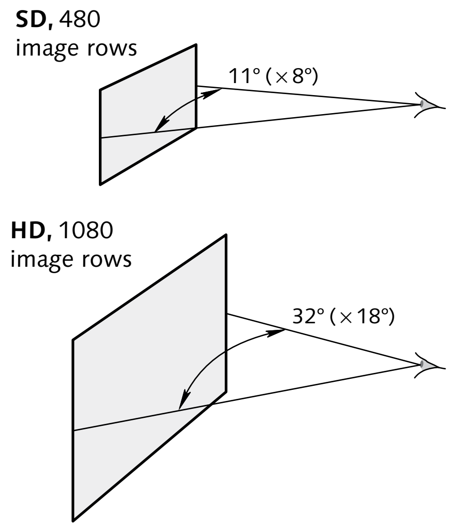


Figure 9.5 **The picture angle** of SD, sketched at the top, has a horizontal angle of about 11° and a vertical angle of about 8°, where pixel structure becomes invisible. In 1920×1080 HD, horizontal angle can increase to about 33°, and vertical angle to about 18°, preserving the pixel subtense.

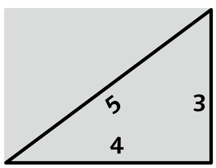


Figure 9.6 **Picture height** at an aspect ratio of 4:3 is $3/5$ of the diagonal; optimum viewing distance for conventional video is 4.25 times the diagonal. Picture height at 16:9 is about half the diagonal; optimum viewing distance for 2 Mpx HD is 1.5 times the diagonal.

distance is about 4.25 times the display diagonal, as sketched in Figure 9.6 in the margin. For HD with 1080 image rows, the viewing distance that yields the $1/60°$ pixel subtense is about 3.2 PH (see the bottom of Figure 9.4), about 1.5 times the display diagonal.

For SD, the total horizontal picture angle at that viewing distance is about 11°. Viewers tend to choose a viewing distance that renders pixel structure invisible; angular subtense of a pixel is thereby preserved. Thus, the main effect of higher pixel count is to enable viewing at a wide picture angle. For 1920×1080 HD, horizontal viewing angle is tripled to 33° compared to the 11° of SD as sketched in Figure 9.5. The “high definition” of HD does not squeeze six times the number of pixels into the same visual angle! Instead, the entire image can potentially occupy a much larger area of the viewer’s visual field. This topic is addressed further in *Viewing distance,* on page 104.

Kell effect

Lowercase *k* for *Kell factor* is unrelated to *K rating*, sometimes called *K factor*, which I will describe on page 542; neither is related to 1000 or 1024.

Early television systems failed to deliver the maximum resolution that was to be expected from Nyquist's work (which was introduced on page 78). In 1934, Kell published a paper quantifying the fraction of the maximum theoretical resolution achieved by RCA's experimental television system. He called this fraction *k;* later – apparently without Kell's consent! – it became known as the *Kell factor* (less desirably denoted *K*). Kell's first paper gives a factor of 0.64, but he failed to completely describe his experimental method. A subsequent paper (in 1940) detailed the method, and gives a factor of 0.8, under somewhat different conditions.

Kell, Ray D., Alda V. Bedford, and G. L. Fredendall (1940), "A determination of the optimum number of lines in a television system," in *RCA Review* **5**: 8–30 (July).

Kell's *k* factor was determined by subjective, not objective, criteria. If the system under test had a spot profile resembling a Gaussian, closely spaced lines on a test chart would cease to be resolved as their spacing diminished beyond a certain value. If a camera under test had an unusually small spot size, or a display had a sharp distribution (such as a box), then *k* was determined by the intrusion of objectionable artifacts as the spacing reduced – also a subjective criterion.

Hsu, Stephen C. (1986), "The Kell factor: past and present," in *SMPTE Journal* **95** (2): 206–214 (Feb.).

Kell and other authors published various theoretical derivations that justify various numerical factors; Hsu has published a comprehensive review. In my opinion, such numerical measures are so poorly defined and so unreliable that they are now useless. Hsu says:

> Kell factor is defined so ambiguously that individual researchers have justifiably used different theoretical and experimental techniques to derive widely varying values of *k*.

Today I consider it poor science to quantify a numerical Kell *factor.* However, Ray Kell made an important contribution to television science, and I think it entirely fitting that we honour him with the Kell *effect:*

> *In* a *video system – including image capture, signal processing, transmission, and display –* KELL EFFECT *refers to the loss of resolution, relative to the Nyquist limit, caused by the spatial dispersion of light power. Some dispersion is necessary to avoid aliasing upon capture; some dispersion is necessary to avoid objectionable scan line or pixel structure at* a *display.*

Twitter is introduced on page 89.

Mitsuhashi, Tetsuo (1982), "Scanning specifications and picture quality," in Fujio, Takashi, et al., *High Definition Television*, NHK Science and Technical Research Laboratories Tech. Monograph **32** (June).

Kell's 1934 paper concerned only progressive scanning. With the emergence of interlaced systems, it became clear that twitter resulted from excessive vertical detail. To reduce twitter to tolerable levels, it was necessary to reduce vertical resolution to substantially below that of a well-designed progressive system having the same spot size – for a progressive system with a given *k*, an interlaced system having the same spot size had to have lower *k*. Many people have lumped this consideration into "Kell factor," but researchers such as Mitsuhashi identify this reduction separately as an *interlace factor* or *interlace coefficient.*

Resolution

SD (at roughly 720×480), HD at 1280×720, and HD at 1920×1080 all have different pixel counts. Image quality delivered by a particular number of pixels depends upon the nature of the image data (e.g., whether the data is raster-locked or Nyquist-filtered), and upon the nature of the display device (e.g., whether it has box or Gaussian reconstruction).

In computing, unfortunately, the term *resolution* has come to refer simply to the count of vertical and horizontal pixels in the pixel array, without regard for any overlap at capture, or overlap at display that may have reduced the amount of detail in the image. A system may be described as having "resolution" of 1152×864 – this system has a total of about one million pixels (one megapixel, or 1 Mpx). Interpreted this way, "resolution" doesn't depend upon whether individual pixels can be discerned ("resolved") on the face of the display.

Resolution in a digital image system is bounded by the count of pixels across the image width and height. However, as picture detail increases in frequency, signal processing and optical effects cause response to diminish even within the bounds imposed by sampling. In video, we are concerned with resolution that is delivered to the viewer; we are also interested in limitations of frequency response ("bandwidth") that may have been imposed in capture, recording, processing, and display. In video, resolution concerns the maximum number of line pairs (or cycles) that can be resolved on the display screen. This is a subjective criterion! Resolution is related to perceived sharpness.

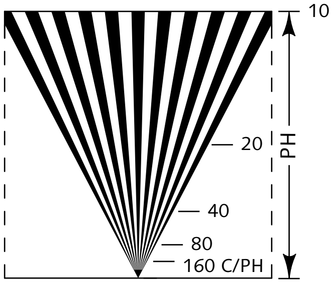


Figure 9.7 **A Resolution wedge** pattern sweeps various horizontal frequencies through an imaging system. This pattern is calibrated in terms of cycles per picture *height* (here signified PH); however, with the pattern in the orientation shown, *horizontal* resolution is measured.

Resolution is usually expressed in terms of spatial frequency, whose units are cycles per picture width (C/PW) horizontally, and cycles per picture height (C/PH) vertically, or units closely related to these. Figure 9.7 depicts a resolution test chart. In the orientation presented, it sweeps across horizontal frequencies, and can be used to estimate horizontal resolution. Turned 90°, it can be used to sweep through vertical frequencies, and thereby estimate vertical resolution.

Resolution in video

Spatial phenomena at an image sensor or at a display device may limit both vertical and horizontal resolution. Analog processing, recording, and transmission historically limits bandwidth, and thereby affects only horizontal resolution. *Resolution* in video historically refers to horizontal resolution:

> *Resolution in TVL/PH – colloquially, "TV lines" – is twice the number of vertical black and white line pairs (cycles) that can be visually discerned across* a *horizontal distance equal to the picture height.*

Vertical resampling has become common in consumer equipment; resampling potentially affects vertical resolution. In transform-based compression (such as JPEG, DV, and MPEG), dispersion comparable to overlap between pixels occurs; this affects horizontal and vertical resolution.

Viewing distance

Pixel count in SD and HD is fixed by the corresponding image format. On page 100, I explained that viewing distance is optimum where the scan-line pitch subtends an angle of about $^1/_{60}$°. If a sampled image is viewed closer than that distance, scan lines or pixels are liable to be visible. With typical displays, SD is suitable for viewing at about 7·PH; 1080*i* HD is suitable for viewing at a much closer distance of about 3·PH.

A computer user tends to position himself or herself where scan-line pitch subtends an angle greater than $^1/_{60}$° – perhaps at half that distance. However, at such a close distance, individual pixels are likely to be discernible, perhaps even objectionable, and the quality of continuous-tone images will almost certainly suffer.

MTF: Modulation transfer function

Closest viewing distance is constrained by pixel count; however, visibility of pixel or scan-line structure in an image depends upon many other factors such as camera MTF, spot profile (PSF), and frequency response. In principle, if any of these factors reduces the amount of detail in the image, the optimum viewing distance is pushed more distant. However, consumers have formed an expectation that SD is best viewed at about 7·PH; as people become familiar with HD they will form an expectation that it is best viewed at about 3·PH.

Lechner worked at RCA Labs in Princeton, New Jersey. Jackson worked at Philips Research Laboratories, Redhill, Surrey, U.K.; he is unrelated to my like-named colleague who worked at Grass Valley Group, now at AJA Video.

A countervailing argument is based upon the dimensions of consumers' living rooms. In unpublished research, Bernie Lechner found that North American viewers tend to view SD receivers at about 9 ft. In similar experiments in England, Richard Jackson found a preference for 3 m. This viewing distance is sometimes called the *Lechner distance* – or in Europe, the *Jackson distance!* These numbers are consistent with Equation 9.2, on page 100 applied to an SD display with a 27-inch (70 cm) diagonal.

Rather than saying that improvements in bandwidth or spot profile enable decreased viewing distance, and therefore wider picture angle, we assume that viewing distance is fixed, and say that resolution is improved.

Interlace revisited

We can now revisit the parameters of interlaced scanning. With the luminance and surround conditions typical of consumer television receivers, a vertical scan rate of 50 or 60 Hz is sufficient to overcome flicker. As I mentioned on page 88, at practical vertical scan rates, it is possible to flash alternate image rows in alternate vertical scans without causing flicker. This is *interlace.* The scheme is possible owing to the fact that temporal sensitivity of the visual system decreases at high spatial frequencies.

Twitter is introduced, however, by vertical detail whose scale approaches the scan-line pitch. Twitter can be reduced to tolerable levels by reducing the vertical detail somewhat, to perhaps 0.7 times. On its own, this reduction in vertical detail would push the viewing distance back to 1.4 times that of progressive scanning.

However, to maintain the same sharpness as a progressive system at a given data capacity, all else being equal, in interlaced scanning only half the picture data needs to be transmitted in each vertical scan period (field). For a given frame rate, this reduction in data per scan enables pixel count per frame to be doubled.

The pixels gained could be exploited in one of three ways: by doubling the row count, by doubling the column count, or by distributing the additional pixels proportionally to image columns and rows. Taking the third approach, the doubled pixel count could be distributed equally horizontally and vertically, increasing column count by a factor of 1.4 and row count by a factor of 1.4. Viewing distance could thereby be reduced to 0.7 that of progressive scan, winning back the lost viewing distance associated with twitter, and yielding equivalent performance to progressive scan.

Ideally, though, the additional pixels owing to interlaced scan should not be distributed equally to both dimensions. Instead, the count of image rows should be increased by about 1.4×1.2 (i.e., 1.7), and the count of image columns by about 1.2. The factor of 1.4 increase in the row count alleviates twitter. The remaining 1.2 increase in both row and column count yields a modest but significant improvement in viewing distance – and therefore picture angle – over a progressive system.

Twitter and scan-line visibility are inversely proportional to the count of image rows, a one-dimensional quantity. However, sharpness is proportional to pixel count, a two-dimensional (areal) quantity. To overcome twitter at the same picture angle, 1.4 times as many image rows are required; however, 1.2 times as many rows and 1.2 times as many columns are still available to improve picture angle.

Interlaced scanning was chosen over progressive in the early days of television, half a century ago. All other things being equal – such as data rate, frame rate, spot size, and viewing distance – various advantages have been claimed for interlace scanning.

- Neglecting the introduction of twitter, and considering just the static pixel array, interlace offers twice the static resolution for a given bandwidth and frame rate.
- If you consider an interlaced image of the same size as a progressive image and viewed at the same distance – that is, preserving the picture angle – then there is a decrease in scan-line visibility.

Constant luminance 10

Video systems convey colour image data using one component that approximates lightness, and two other components that represent colour, absent lightness. In *Colour science for video,* on page 287, I will detail how luminance can be formed as a weighted sum of linear *RGB* values each of which is proportional to optical power. A colour scientist uses the term *constant luminance* to refer to this sum being constant. Transmitting a single component from which relative luminance can be reconstructed is the *principle of constant luminance*. Preferably a nonlinear transfer function acts on that component to impose perceptually uniform coding.

Standard video systems do not strictly adhere to that principle; instead, they implement an engineering approximation. The colour scientist's weighted sum of linear *RGB* is *not* computed. Instead, a nonlinear transfer function is applied to each linear-light *RGB* component individually, then a weighted sum of the nonlinear *gamma-corrected R'G'B'* components forms what I call *luma.* (Many video engineers carelessly call this quantity *luminance*.) In standard video systems, luma is encoded using the theoretical *RGB* weighting coefficients of colour science, but in a block diagram different from the one a colour scientist would expect: In video, gamma correction is applied *before* the matrix, instead of the colour scientist's preference, after.

The term *luminance* is widely misused in video. See *Relative luminance,* on page 258, and Appendix A, *YUV and luminance considered harmful,* on page 567.

Historically, transmission of a single component representative of greyscale enabled compatibility with "black-and-white" television. Human vision has poor acuity for colour compared to luminance. Placing "black-and-white" information into one component enables *chroma subsampling* to take advantage of vision's low acuity for chroma in order to reduce data rate (historically, bandwidth) in the two other components. In colour imaging, it is sensible to code a "black-and-white" component even if "black-and-white" compatibility isn't required (for example, in JPEG).

The term "monochrome" is sometimes used instead of "greyscale." However, in classic computer graphics terminology *monochrome* refers to bilevel (1-bit) images or display systems, so I avoid that term.

I've been placing "black-and-white" in quotes. At the invention of television, the transmitted signal represented greyscale, not just black and white: Then, and now, *greyscale* would be a better term.

Historical video literature refers to the "signal representing luminance" or the "luminance signal" or the "luminance component." All of these terms were once justified; however, they are now dangerous: To use the term "luminance" suggests that relative luminance (Y) can be decoded from that component. However, without strict adherence to the principle of constant luminance, luminance *cannot* be decoded from the greyscale component alone: Two other components (typically C_B and C_R) are necessary.

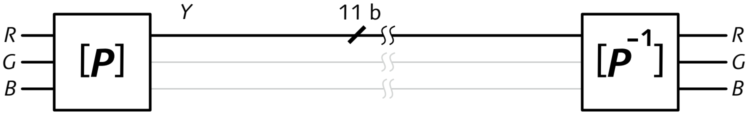
In this chapter, I will explain why and how all current video systems depart from the principle of constant luminance. If you are willing to accept this departure from theory as a fact, then you may safely skip this chapter, and proceed to *Introduction to luma and chroma,* on page 121, where I will introduce how the luma and colour difference signals are formed and subsampled.

The principle of constant luminance

Applebaum, Sidney (1952), "Gamma correction in constant luminance color television systems," in *Proc. IRE*, **40** (11): 1185–1195 (Oct.).

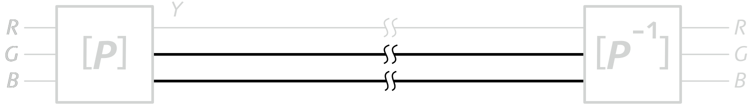
Ideally, the so-called monochrome component in colour video would mimic a greyscale system: Relative luminance would be computed as a properly weighted sum of (linear-light) R, G, and B tristimulus values, according to the principles of colour science that are explained in *Transformations between RGB and CIE XYZ,* on page 307. At the decoder, the inverse matrix would reconstruct linear R, G, and B tristimulus values:

Figure 10.1 **Formation of relative luminance**



Two colour difference (chroma) components would be computed, to enable chroma subsampling; these would be conveyed to the decoder through separate channels:

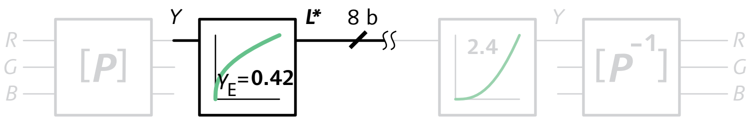
Figure 10.2 **Hypothetical chroma components (linear-light)**



Set aside the chroma components for now: No matter how they are handled, in a true constant luminance system all of the relative luminance is recoverable from the greyscale component alone.

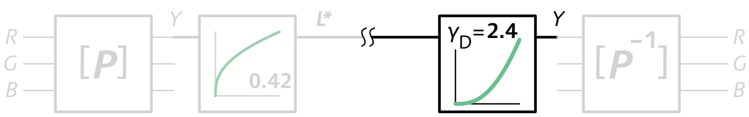
If relative luminance were conveyed directly, 11 bits or more would be necessary. Eight bits barely suffice if we use nonlinear image coding, introduced on page 31, to impose perceptual uniformity: We could subject relative luminance to a nonlinear transfer function that mimics vision's lightness sensitivity. Lightness can be approximated as CIE L^* (to be detailed on page 259); L^* is roughly the 0.42-power of relative luminance.

Figure 10.3 **Encoding nonlinearly coded relative luminance**



The decoder would apply the inverse transfer function:

Figure 10.4 **Decoding nonlinearly coded relative luminance**

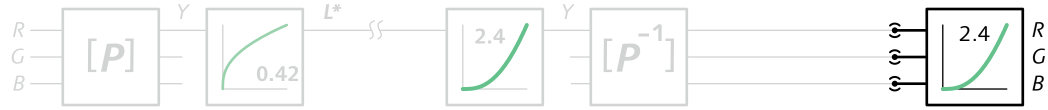


If a video system were to operate in this manner, it would conform to the principle of constant luminance: All of the relative luminance would be present in, and recoverable from, the greyscale component.

Compensating for the CRT

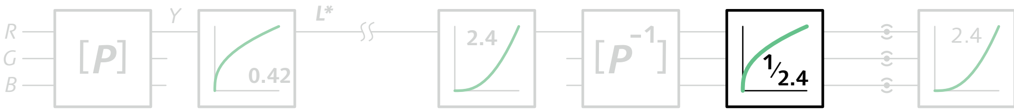
Unfortunately for the theoretical block diagram – but fortunately for video, as you will see in a moment – the electron gun of a historical CRT display introduces a power function having an exponent of about 2.4:

Figure 10.5 **The CRT transfer function**



In a constant luminance system, the decoder would have to invert the display's power function. This would require insertion of a compensating transfer function – roughly a $\frac{1}{2.4}$-power function – in front of the CRT:

Figure 10.6 **Compensating the CRT transfer function**



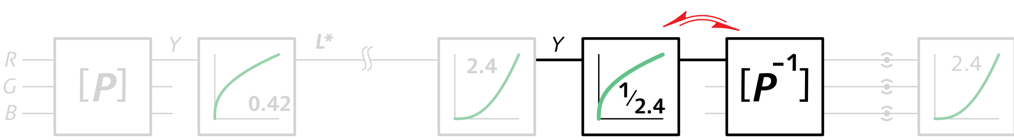
The decoder would now include two power functions: An inverse *L** function with an exponent close to 2.4 to invert the perceptually uniform coding, and a power function with an exponent of $\frac{1}{2.4}$ – that is, about 0.42 – to compensate for the CRT's nonlinearity. Figure 10.6 represents the block digram of an idealized, true constant luminance video system.

Departure from constant luminance

Having two nonlinear transfer functions at every decoder was historically expensive and impractical.

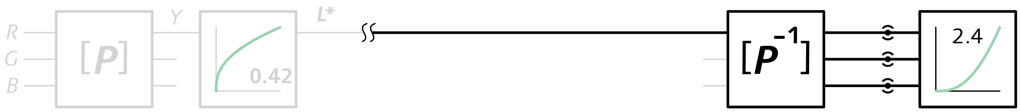
Notice that the exponents of the power functions are 2.4 and $\frac{1}{2.4}$ – the functions are inverses! To avoid the complexity of incorporating two power functions into a decoder's electronics, we begin by rearranging the block diagram, to interchange the "order of operations" of the matrix and the CRT compensation:

Figure 10.7 **Rearranged decoder**



Upon rearrangement, the two power functions are adjacent. Since the functions are effectively inverses, the combination of the two has no net effect. Both functions can be dropped from the decoder:

Figure 10.8 **Simplified decoder**



Decoder signal processing simply inverts the encoder matrix. The 2.4-power function is intrinsic to a CRT display; alternative display technologies exhibit comparable mapping from signal value to tristimulus.

Rearranging the decoder requires that the encoder also be rearranged, so as to mirror the decoder and achieve correct end-to-end reproduction of the original *RGB* tristimulus values:

Figure 10.9 **Rearranged encoder**

Figure 10.9 represents the basic signal flow for all video systems; it will be elaborated in later chapters.

Luma

Television engineers who are uneducated in colour science often mistakenly call luma (*Y'*) by the name *luminance* and denote it by the unprimed symbol *Y*. This leads to great confusion, as I explain in Appendix A, on page 567.

The rearranged flow diagram of Figure 10.9 is *not* mathematically equivalent to the arrangement of Figures 10.1 through 10.4! In Figure 10.9, the encoder's matrix does not operate on (linear-light) tristimulus signals, and relative luminance is not computed. Instead, a nonlinear quantity – denoted *luma* and symbolized *Y'* – is computed and transmitted. Luma involves an engineering approximation: The system no longer adheres strictly to the principle of constant luminance (though it is often mistakenly claimed to do so).

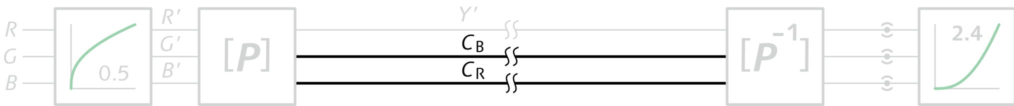
In the rearranged encoder, we no longer use CIE *L** to optimize for perceptual uniformity; instead, we use the inverse of the CRT's inherent transfer function. A 0.42-power function accomplishes approximately perceptually uniform coding, and reproduces tristimulus values proportional to those in the original scene.

The following chapter, *Picture rendering*, explains that the 0.42 value must be altered in a normal scene to about 0.5 to accommodate a perceptual effect. The alteration depends upon artistic intent, and upon display and viewing conditions. Ideally, display systems should have adjustments for picture rendering depending upon display and viewing conditions, but they rarely do!

"Leakage" of luminance into chroma

Until now, we have neglected the colour difference components. In the rearranged block diagram of Figure 10.9, colour difference components are "matrixed" from *nonlinear* (gamma-corrected) $R'G'B'$:

Figure 10.10 **Chroma components**



In a true constant luminance system, no matter how the colour difference signals are handled, all of the relative luminance is carried by the greyscale component. In the rearranged system, most of the relative luminance is conveyed through the Y' channel. However, to the extent that Y' isn't equal to Y, some relative luminance can be thought of as "leaking" into the colour difference components. If the colour difference components were not subsampled – for example, in a $Y'C_BC_R$, 4:4:4 system – this leakage would be inconsequential. However, the colour difference components are formed precisely to *enable* subsampling! So, we now turn our attention to subsampling.

Figure 10.11 below shows Figure 10.10's practical block diagram augmented with subsampling filters in the chroma paths. With conventional coding, some of

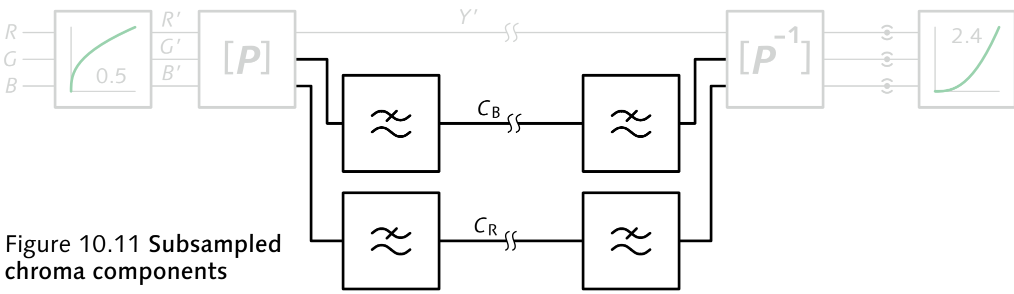


Figure 10.11 **Subsampled chroma components**

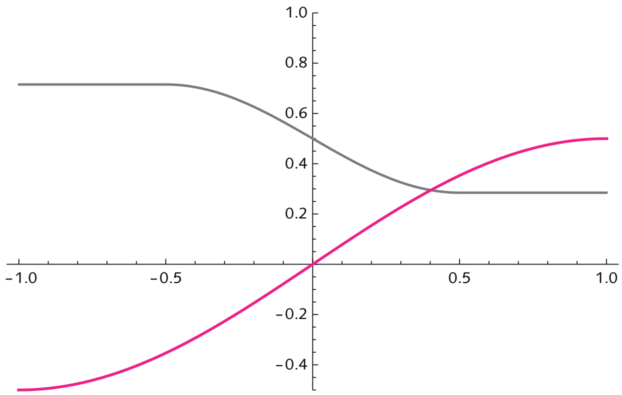


Figure 10.12 **Y' and C_B/C_R waveforms at the green-magenta transition** of SD colourbars are shown, following idealized 4:2:2 chroma subsampling. The luma waveform is plotted in grey; C_B and C_R share the same waveform, plotted in magenta. The transition rate (rise time) of the C_B and C_R components is half that of luma.

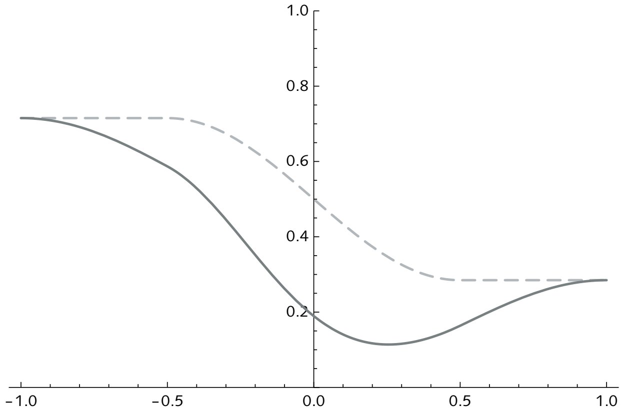


Figure 10.13 **Luminance waveform at the green-magenta transition** of colourbars is shown in the solid line. The dashed line reflects luminance in a hypothetical true constant-luminance system.

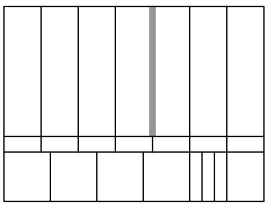


Figure 10.14 **Failure to adhere to constant luminance** is evident in the dark band in the green-magenta transition of colourbars. The dark band is found upon displaying any colourbar signal that has been subject to chroma subsampling.

the relative luminance traverses the chroma pathways. Figure 10.12 above shows the idealized $Y'C_BC_R$ waveforms at the green-magenta transition of colourbars, with 4:2:2 chroma subsampling. Figure 10.13 shows, in the solid line, the luminance that results after conventional decoding. Subsampling not only removes detail from the colour components, it removes detail from the "leaked" relative luminance. We have to ask, "What's lost?" The departure from theory is apparent in the dark band appearing between the green and magenta colour bars of the standard video test pattern, depicted in Figure 10.14 in the margin.

With conventional video coding, in areas where luminance detail is present in saturated colours, relative luminance is incorrectly reproduced: relative luminance is reproduced too dark, and saturation is reduced. This inaccurate conveyance of high-frequency

Livingston, Donald C. (1954), "Reproduction of luminance detail by NTSC color television systems," in *Proc. IRE* **42** (1): 228–234.

luminance is the price that must be paid for lack of strict adherence to the principle of constant luminance. Such "Livingston" errors are perceptible by experts, but they are very rarely noticeable – let alone objectionable – in normal imagery.

To summarize signal encoding in video systems: First, a nonlinear transfer function, *gamma correction*, comparable to a square root, is applied to each of the linear *R*, *G*, and *B* tristimulus values to form *R'*, *G'*, and *B'*. Then, a suitably weighted sum of the nonlinear components is computed to form the luma signal (*Y'*). Luma approximates the lightness response of vision. Colour difference components *blue minus luma* (*B'–Y'*) and *red minus luma* (*R'–Y'*) are formed. (Luma, *B'–Y'*, and *R'–Y'* can be computed from *R'*, *G'*, and *B'* simultaneously, through a 3×3 matrix.) The colour difference components are then subsampled (filtered), using one of several schemes – including 4:2:2, 4:1:1, and 4:2:0 – to be described starting on page 124.

This chapter has outlined how, in the development of NTSC, an engineering approximation to constant luminance was adopted rather than "true" constant luminance. This engineering decision has served spectacularly well, and has been carried into component video systems (SD and HD), and into modern compression systems such as JPEG, MPEG, and H.264.

Since about 2000, the majority of television receivers have incorporated digital signal processing that obviates the engineering argument made in 1950: The two nonlinear functions of Figure 10.6 could today be easily be implemented by lookup tables. Some purists believe that in the modern age we should abolish the approximation, and adopt the correct theoretical approach. However, the video infrastructure of SD and HD is built on Figure 10.9 (or with chroma subsampling, Figure 10.11). It seems unreasonable to change the block diagram of video, and impose a huge conversion burden, unless substantial benefit can be shown.
I appreciate the theoretical argument; however, I am unaware of any significant benefit that would result from such a change, so I argue that we should not change the block diagram of video.

Picture rendering 11

Examine the flowers in a garden at noon on a bright, sunny day. Look at the same garden half an hour after sunset. Physically, the spectra of the flowers have not changed, except by scaling to lower luminance levels. However, the flowers are markedly less colourful after sunset: Colourfulness decreases as luminance decreases. This is the *Hunt effect,* first described (in 1952) by the famous colour scientist R.W.G. Hunt. Reproduced images are usually viewed at a small fraction, perhaps $^1/_{100}$ or $^1/_{1000}$, of the luminance at which they were captured. If reproduced luminance were made proportional to scene luminance, the reproduced image would appear less colourful, and lower in contrast, than the original scene.

Giorgianni, Edward J., and Thomas E. Madden (2008), *Digital Color Management: Encoding Solutions,* Second Edition (Chichester, U.K.: Wiley).

To reproduce contrast and colourfulness comparable to the original scene, we *must* alter the characteristics of the image. An engineer or physicist might strive to achieve mathematical linearity in an imaging system; however, the required alterations cause reproduced luminance to depart from linearity. The dilemma is this: We can achieve mathematical linearity, or we can achieve correct appearance, but we cannot simultaneously do both! Successful commercial imaging systems sacrifice mathematics to achieve the correct perceptual result.

I use the term *white* to refer to diffuse white, which I will explain on page 117. A diffuse white reflector has a luminance of up to 30,000 $cd\cdot m^{-2}$ in daylight, and perhaps 100 $cd\cdot m^{-2}$ at twilight.

If "white" in the viewing environment has luminance significantly less than "white" in the environment in which it was captured, the tone scale of an image must be altered. An additional reason for correction is the surround effect, which I will now explain.

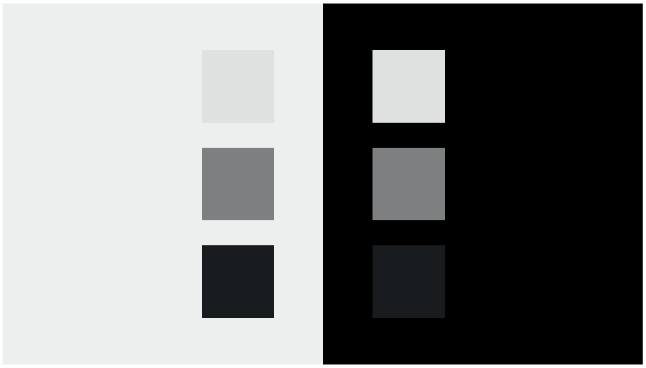


Figure 11.1 **Surround effect.** The three squares surrounded by light grey are identical to the three squares surrounded by black; however, each of the black-surround squares is apparently lighter than its counterpart. Also, the contrast of the black-surround series appears lower than that of the white-surround series.

DeMarsh, LeRoy E., and Edward J. Giorgianni (1989), "Color science for imaging systems," in *Physics Today* **42** (9): 44–52 (Sep.).

Surround effect

Human vision adapts to an extremely wide range of viewing conditions, as I will detail in *Adaptation,* on page 247. One of the mechanisms involved in adaptation increases our sensitivity to small brightness variations when the area of interest is surrounded by bright elements. Intuitively, light from a bright surround can be thought of as spilling or scattering into all areas of our vision, including the area of interest, reducing its apparent contrast. Loosely speaking, the visual system compensates for this effect by "stretching" its contrast range to increase the visibility of dark elements in the presence of a bright *surround*. Conversely, when the region of interest is surrounded by relative darkness, the contrast range of the vision system decreases: Our ability to discern dark elements in the scene decreases. The effect is demonstrated in Figure 11.1 above, from DeMarsh and Giorgianni. The surround effect stems from the perceptual phenomenon called the *simultaneous contrast effect*, also known as *lateral inhibition*.

I use the term *glare* (or *viewing glare,* or *veiling glare*) to refer to uncontrolled, unwanted light that reflects from the display surface, either diffusely or specularly. Image-related scattered light is called *flare.* Some people use these terms interchangeably, thus failing to distinguish unmodulated and modulated light.

Unfortunately, *simultaneous contrast* has another meaning, where it is a contraction of *simultaneous contrast ratio,* preferrably called *intra-image contrast ratio.* See *Contrast ratio,* on page 29.

The surround effect has implications for the display of images in dark areas, such as projection of movies in a cinema, projection of 35 mm slides, or viewing of television in your living room. If an image were reproduced with the correct relative luminance, then when viewed in a dark or dim surround, it would appear lacking in contrast.

Image reproduction is not simply concerned with physics, mathematics, chemistry, and electronics: Perceptual considerations play an essential role.

Tone scale alteration

Tone scale alteration is necessary mainly for the two reasons that I have described: The luminance of a reproduction is typically dramatically lower than the luminance of the original scene, and the surround of a reproduced image is rarely comparable to the surround of the original scene. Two additional reasons contribute to the requirement for tone scale alteration: limitation of contrast ratio, and specular highlights.

Intra-image contrast ratio is the ratio of luminances of the lightest and darkest elements of an image. For details, see *Contrast ratio,* on page 29.

A typical original scene has a ratio of luminance levels of more than 1000:1. However, contrast ratio in the captured image is limited by optical flare in the camera. Contrast ratio at a display is typically limited by physical factors and by display flare to perhaps 1000:1.

Diffuse white refers to the luminance of a diffusely reflecting white surface in a scene. Paper reflects diffusely, and white paper reflects about 90% of incident light, so a white card approximates diffuse white. However, most scenes contain shiny objects that reflect directionally. When viewed in certain directions, these objects reflect specular highlights having luminances perhaps ten times that of diffuse white. At the reproduction device, we can seldom afford to reproduce diffuse white at merely 10% of the maximum luminance of the display, solely to exactly reproduce the luminance levels of the highlights! Nor is there any need to reproduce highlights exactly: A convincing image can be formed with highlight luminance greatly reduced from its true value. To make effective use of luminance ranges that are typically available in image display systems, highlights must be compressed.

Incorporation of rendering

The correction that I have mentioned can be achieved by subjecting luminance – or, in the case of a colour system, tristimulus values – to an end-to-end power function having an exponent between about 1.1 and 1.5. The exponent depends primarily upon the ratio of scene luminance to reproduction luminance; it depends to some degree upon the display physics and the viewing environment. Nearly all image reproduction systems require some tone scale alteration.

In *Constant luminance,* on page 107, I outlined nonlinear coding in video. Continuing the sequence of sketches from Figure 10.9, on page 111, Figure 11.2 shows that correction for typical television viewing could be effected by including, in the decoder, a power function having an exponent of about 1.2:

Figure 11.2 **Imposition of picture rendering at decoder, hypothetical**

Observe that a power function is already a necessary part of the encoder. Instead of altering the decoder, we modify the encoder's power function to approximate a 0.5-power, instead of the physically correct 0.42-power:

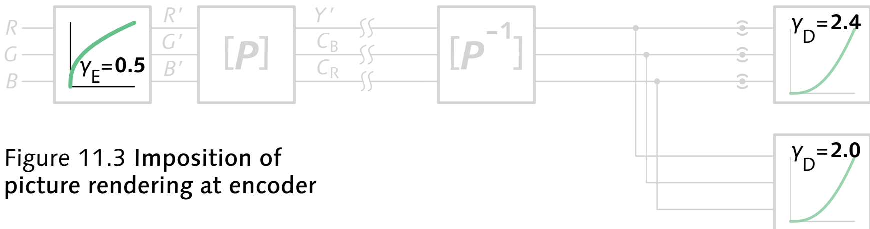


Figure 11.3 **Imposition of picture rendering at encoder**

Concatenating the 0.5-power at encoding and the 2.4-power at decoding produces the end-to-end 1.2-power required for television display in a dim surround. To recover scene tristimulus values, the encoding transfer function should simply be inverted; the decoding function then approximates a 2.0-power function, as sketched at the bottom right of Figure 11.3.

The effective power function exponent at a CRT varies depending upon the setting of the BRIGHTNESS control. In a dark viewing environment – such as a home theater – the display's BRIGHTNESS setting could be so low as to push the decoder's effective exponent up to about 2.6; the end-to-end power will be about 1.3. In a bright surround – such as a computer in a desktop environment – BRIGHTNESS will be increased; this will reduce the effective exponent to about 2.2, and thereby reduce the end-to-end exponent to about 1.1.

The encoding exponent, decoding exponent, and end-to-end power function for cinema, television, and office CRT viewing are shown in Table 11.1.

In film systems, the necessary correction was historically designed into the transfer function of the camera

FAIRCHILD, MARK D. (2005), *Color Appearance Models*, Second Edition (Chichester, U.K.: Wiley).

JAMES, THOMAS H., ed. (1977), *The Theory of the Photographic Process*, Fourth Edition (Rochester, N.Y.: Eastman Kodak). See Ch. 19 (p. 537), *Preferred tone reproduction*.

Imaging system	*Encoding exponent*	*"Advertised" exponent*	*Decoding exponent*	*Typ. surround*	*Contrast ratio*	*End-to-end exponent*
Cinema (film projection)	0.6	0.6	2.5	Dark (0%)	100:1	1.5
HD, studio mastering (BT.709/BT.1886)	0.5	0.45	2.4	Very dim (1%)	1000:1	1.2
HD, living room (typ.)	0.5	0.45	2.4	Dim (5%)	400:1	1.2
Office (sRGB, typ.)	0.45	0.42	2.2	Avg (20%)	100:1	1.1

Table 11.1 **End-to-end power functions** for several imaging systems. The encoding exponent achieves approximately perceptual coding. (The "advertised" exponent neglects the scaling and offset associated with the straight-line segment of encoding.) The decoding exponent acts at the display to approximately invert the perceptual encoding. The product of the two exponents sets the end-to-end power function that imposes the rendering. Here, contrast ratio is intra-image.

negative and print films. Projected imagery is typically intended for viewing in a dark surround; arrangements are made to have an end-to-end power function exponent considerably greater than unity – typically about 1.5 – so that the contrast range of the scene is expanded upon display. In cinema film, the correction is achieved through a combination of the transfer function ("gamma" of about 0.6) built into camera negative film and the transfer function ("gamma" of about 2.5) built into print film.

Some people suggest that NTSC should be gamma-corrected with power of $^1/_{2.2}$, and PAL with $^1/_{2.8}$. I disagree with both interpretations; see page 325.

I have described video systems as if they use a pure 0.5-power law encoding function. Practical considerations necessitate modification of the pure power function by the insertion of a linear segment near black, as I will explain in *Gamma,* on page 315. The exponent in the BT.709 standard is written ("advertised") as 0.45; however, the insertion of the linear segment, and the offsetting and scaling of the pure power function segment of the curve, cause an exponent of about 0.51 to best describe the overall curve. (To describe gamma as 0.45 in this situation is misleading.)

Rendering in desktop computing

In the desktop computer environment, the ambient condition is considerably brighter, and the surround is brighter than is typical of television viewing. An end-to-end exponent lower than the 1.2 of video is called for; a value around 1.1 is generally suitable. However, desktop computers are used in a variety of different viewing conditions. It is not practical to originate every image in several forms, optimized for several potential viewing conditions! A specific encoding function needs to be chosen. Achieving optimum reproduction in diverse viewing conditions requires selecting a suitable correction at display time. Technically, this is easy to achieve: Modern computer display subsystems have hardware lookup tables (LUTs) that can be loaded dynamically with appropriate curves. However, it is a challenge to train users to make a suitable choice. There is promise in sensors to detect ambient light, and algorithms to effect appropriate correction (largely by altering display gamma). Such schemes have been implemented commercially, but there are no standards.

In the sRGB standard, the exponent is written ("advertised") as $1/2.4$ (about 0.42). However, the insertion of the linear segment, and the offsetting and scaling of the pure power function segment of the curve, cause an exponent of about 0.45 to best describe the overall curve. See *sRGB transfer function,* on page 324.

When the sRGB standard for desktop computing was being developed, the inevitability of local, viewing-dependent correction was not appreciated. That standard promulgates decoding with a pure 2.2-power function, but the standard also described what is apparently an encoding standard with a linear segment near black and an effective exponent of about 0.45. A close reading of the sRGB standard confirms that sRGB is display referred; the video-like definition with the linear segment is a mapping from tristimulus values *at the display surface* into sRGB code values. The sRGB "encode" function is not comparable to BT.709's reference OECF. Display of sRGB material should be accomplished with the pure 2.2-power function, without any linear segment.

Video cameras, film cameras, motion picture cameras, and digital still cameras all capture images from the real world. When an image of an original scene or object is captured, it is important to introduce rendering. However, scanners used in desktop computing rarely scan original objects; they usually scan reproductions such as photographic prints or offset-printed images. When a reproduction is scanned, rendering has already been imposed by the first imaging process. It may be sensible to adjust the original rendering, but it is not sensible to introduce rendering that would be suitable for scanning a real scene or object.

Introduction to luma and chroma 12

The statement is commonly made that "the human visual system is more sensitive to luma than chroma." That statement is incorrect. It is vision's sensitivity *to information at high spatial frequency* that is diminished for chroma. Chroma subsampling is enabled by poor *acuity* for chroma, not by poor *sensitivity*.

Video systems convey image data in the form of one component that represents lightness, and two components that represent colour, disregarding lightness. This scheme exploits the reduced colour acuity of vision compared to luminance acuity: As long as lightness is conveyed with full detail, detail in the colour components can be reduced by subsampling – that is, by filtering (averaging). This chapter introduces the concepts of luma and chroma encoding; details will be presented in *Luma and colour differences*, on page 335.

Luma

A certain amount of noise is inevitable in digital imaging systems. As explained in *Perceptual uniformity*, on page 8, encoding is arranged so that noise has a perceptually similar effect across the entire tone scale from black to white. The lightness component is conveyed in a perceptually uniform manner that minimizes the amount of noise (or quantization error) introduced in processing, recording, and transmission.

Ideally, noise would be minimized by forming a signal proportional to CIE luminance, as a suitably weighted sum of linear R, G, and B tristimulus signals. Then, this signal would be subjected to a transfer function that imposes perceptual uniformity, such as the CIE L^* function of colour science that will be detailed on page 259. As explained in *Constant luminance*, on page 107, there are practical reasons in video to perform these operations in the opposite order. First, a nonlinear transfer function – *gamma correction* – is applied to each of the linear R, G, and B tristimulus signals: We impose a transfer function similar to a square root, and roughly comparable to the CIE lightness (L^*) function. Then a weighted sum of the resulting nonlinear R', G', and B' components is computed to form a *luma* signal (Y') representative of lightness. SD uses coefficients that are standardized in BT.601 (see page 131):

The prime symbols here, and in following equations, denote nonlinear components.

$$^{601}Y' = 0.299\,R' + 0.587\,G' + 0.114\,B' \qquad \text{Eq 12.1}$$

Unfortunately, luma for HD is coded differently from luma in SD! BT.709 specifies these coefficients:

$$^{709}Y' = 0.2126\,R' + 0.7152\,G' + 0.0722\,B' \qquad \text{Eq 12.2}$$

Sloppy use of the term *luminance*

CIE: Commission Internationale de l'Éclairage

The term *luminance* and the symbol Y were established 75 years ago by the CIE, the standards body for colour science. Unfortunately, in video, the term *luminance* has come to mean *the video signal representative of luminance* even though the components of the video signal have been subjected to a nonlinear transfer function. At the dawn of video, the nonlinear signal was denoted Y', where the prime symbol indicated the nonlinear treatment. But over the last 50 years the prime has not appeared consistently, and today, both the term *luminance* and the symbol Y conflict with their CIE definitions, making them ambiguous! This has led to great confusion, such as the incorrect statement commonly found in computer graphics textbooks and digital image-processing textbooks that in the *YIQ* or *YUV* colour spaces, the Y component is identical to CIE luminance!

See Appendix A, *YUV and luminance considered harmful,* on page 567.

I use the term *luminance* according to its CIE definition; I use the term *luma* to refer to the video signal; and I am careful to designate nonlinear quantities with a prime. However, many video engineers, computer graphics practitioners, and image-processing specialists use these terms carelessly. You must be careful to determine whether a linear or nonlinear interpretation is being applied to the word and the symbol.

Colour difference coding (chroma)

In component video, the three components necessary to convey colour information are transmitted separately. Rather than conveying *R'G'B'* directly, the relatively poor colour acuity of vision is exploited to reduce data capacity accorded to the colour information, while maintaining full luma detail. First, luma is formed according to Equation 12.1 (or for HD, Equation 12.2). Then, two *colour difference* signals based upon gamma-corrected *B' minus luma* and *R' minus luma*, *B'*–*Y'* and *R'*–*Y'*, are formed by "matrixing." Finally, subsampling (filtering) reduces detail in the colour difference (or *chroma*) components, as I will outline on page 127. Subsampling incurs no loss in sharpness at any reasonable viewing distance.

Luma and colour differences can be computed from *R'*, *G'*, and *B'* through a 3×3 matrix multiplication.

$Y'P_BP_R$

In component analog video, *B'*–*Y'* and *R'*–*Y'* were scaled to form colour difference signals denoted P_B and P_R, which were then analog lowpass filtered (horizontally) to about half the luma bandwidth. *Y'*, P_B, and P_R each have unity excursion (i.e., 0 to 1, ±0.5, and ±0.5). In computing, these components are commonly denoted *U* and *V*.

$Y'C_BC_R$

In component digital video (including M-JPEG, MPEG, and H.264), *B'*–*Y'* and *R'*–*Y'* are scaled to form C_B and C_R components, which can then be subsampled by digital filtering denoted 4:2:2 or 4:2:0; I will describe the subsampling in a moment.

Y'UV

In composite NTSC or PAL video, *B'*–*Y'* and *R'*–*Y'* were scaled to form *U* and *V* components. *U* and *V* were then lowpass filtered and combined into a modulated chroma component, *C*. Luma was then summed with modulated chroma to produce the composite NTSC or PAL signal. Scaling of *U* and *V* was arranged so that the excursion of the composite signal (*Y'*+*C*) was constrained to the range $-\frac{1}{3}$ to $+\frac{4}{3}$ of the unity excursion of luma. The historical *U* and *V* components are now obsolete, and today *UV* denotes $Y'P_BP_R$ (see above).

Y'IQ

Composite NTSC video was standardized in 1953 based upon *I* and *Q* components that were essentially *U* and *V* components rotated 33° and axis-exchanged. Excess detail was supposed to be removed from the *Q* component so as to improve colour quality. The scheme never achieved significant deployment; *I* and *Q* components have been completely obsolete for many decades.

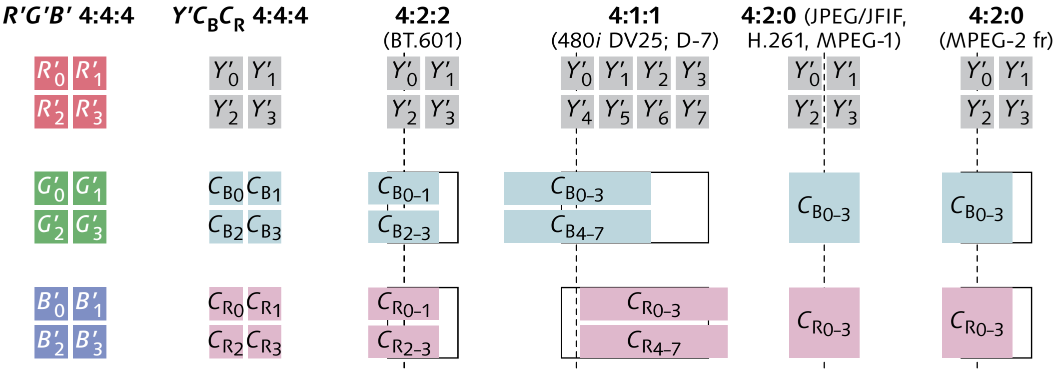


Figure 12.1 **Chroma subsampling.** A 2×2 array of *R'G'B'* pixels is matrixed into a luma component *Y'* and two colour difference components C_B and C_R. Colour detail is reduced by subsampling C_B and C_R; providing full luma detail is maintained, no degradation is perceptible. In this sketch, samples are shaded to indicate their spatial position and extent. In 4:2:2, in 4:1:1, and in 4:2:0 used in MPEG-2, C_B and C_R are cosited (positioned horizontally coincident with a luma sample). In 4:2:0 used in JPEG/JFIF, H.261, and MPEG-1, C_B and C_R are sited interstitially (midway between luma samples). In the 4:2:0 variant used in consumer 576*i* DV (not sketched here), C_B and C_R are vertically sited in line-alternate fashion in each field (starting with a C_R sample sited over the top left luma sample.)

Chroma subsampling

4:4:4

In Figure 12.1, the left-hand column sketches a 2×2 array of *R'G'B'* pixels. (Think of these 2×2 arrays as being overlaid on the display surface.) Prior to subsampling, this sampling is denoted *R'G'B'* 4:4:4. With 8 bits per sample, this 2×2 array of *R'G'B'* would consume 12 bytes. Each *R'G'B'* triplet (pixel) can be losslessly transformed ("matrixed") into $Y'C_BC_R$, as shown in the second column; this is denoted $Y'C_BC_R$ 4:4:4.

In component digital video, data capacity is reduced by subsampling C_B and C_R using one of three schemes.

4:2:2

$Y'C_BC_R$ studio digital video according to BT.601 uses 4:2:2 sampling: C_B and C_R components are each subsampled by a factor of 2 horizontally. C_B and C_R are sampled together, coincident (*cosited*) with even-numbered luma samples. The 12 bytes of *R'G'B'* are reduced to 8, effecting 1.5:1 lossy compression.

4:1:1

Certain digital video systems, such as 480*i*29.97 DV25, use 4:1:1 sampling, whereby C_B and C_R components are each subsampled by a factor of 4 horizontally, and cosited with every fourth luma sample. The 12 bytes of *R'G'B'* are reduced to 6, effecting 2:1 lossy compression.

4:2:0

This scheme is used in JPEG/JFIF, MPEG-2, and H.264. C_B and C_R are each subsampled by a factor of 2 horizontally and a factor of 2 vertically. The 12 bytes of *R'G'B'* are reduced to 6. C_B and C_R are effectively centered vertically halfway between image rows. There are three variants of 4:2:0, having different vertical and horizontal siting. In MPEG-2, C_B and C_R are cosited horizontally. In JPEG/JFIF, H.261, and MPEG-1, C_B and C_R are sited interstitially, halfway between alternate luma samples. In 4:2:0 DV, C_B and C_R alternate line by line. Figure 12.2 overleaf summarizes the various schemes.

ITU-T Rec. H.261 is a 1990s-vintage videoconferencing standard.

Subsampling effects 1.5:1 or 2:1 lossy compression. However, in studio terminology, subsampled video is referred to as *uncompressed:* The word *compression* is reserved for techniques such as JPEG, M-JPEG, MPEG, or H.264 that use transform coding (DCT or wavelets).

Chroma subsampling notation

At the outset of digital video, subsampling notation was logical; unfortunately, technology outgrew the notation. In Figure 12.3 below, I strive to clarify today's nomenclature. Despite appearances, the notation doesn't specify a ratio! The first digit originally specified luma sample rate relative to 3⅜ MHz; the leading digit is now relative to the sample rate in use. The initial digit is typically 4, since all common chroma ratios are small powers of two – 4, 2, or 1. (3:1:1 subsampling was commercialized in an HD production system – Sony's HDCAM – and *3* appeared as the leading digit. HDCAM has been superseded by HDCAM SR, which uses 4:2:2 or 4:4:4 subsampling.) Some people use 4:0:0 to denote greyscale ("monochrome").

The use of *4* as the numerical basis for subsampling notation is a historical reference to sampling at roughly four times the NTSC colour subcarrier frequency. The $4f_{SC}$ rate was already in use for composite digital video.

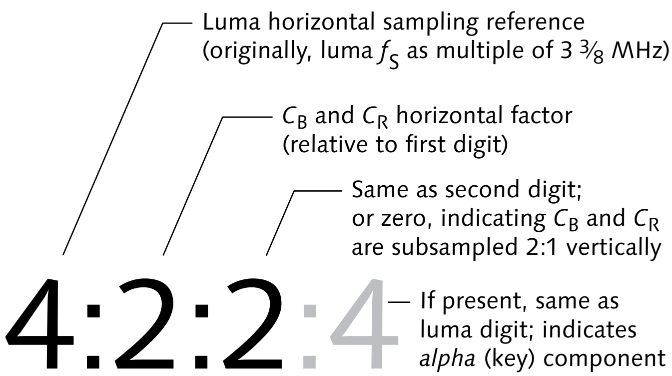
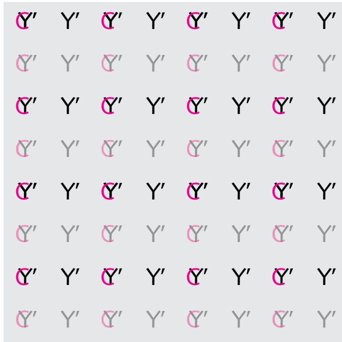
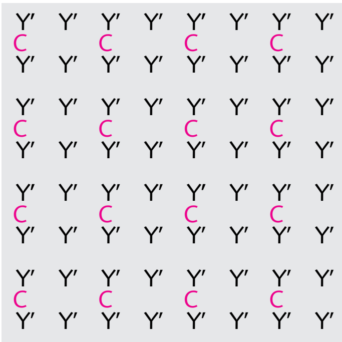


Figure 12.3 **Chroma subsampling notation** indicates, in the first digit, the luma horizontal sampling reference. The second digit specifies the horizontal subsampling of C_B and C_R with respect to luma. The third digit originally specified the horizontal subsampling of C_R. The notation developed without anticipating vertical subsampling; a third digit of zero now denotes 2:1 vertical subsampling of both C_B and C_R.

4:2:2 progressive

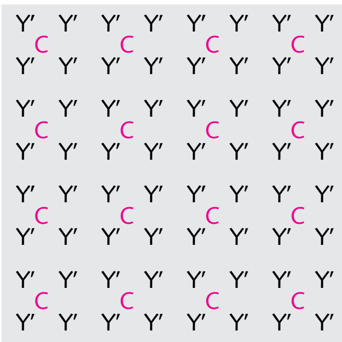


4:2:2 (BT.601) interlaced



4:2:0 MPEG-2 frame picture (progressive)

4:2:0 MPEG-2 interlaced



4:2:0 JPEG/JFIF (progressive)

4:2:0 DV interlaced

4:1:1 DV interlaced

Figure 12.2 **Subsampling schemes** are summarized here. *C* indicates a [C_B, C_R] sample pair when located at the same site; otherwise (as in the DV schemes) individual C_B and C_R notations indicate the centers of the respective chroma samples. *Y′* indicates the center of a luma sample. The schemes in the left column are progressive. The schemes in the right column are interlaced; there, solid letters indicate top field samples and shaded letters indicate bottom field samples.

Chroma subsampling filters

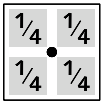


Figure 12.4 **An Interstitial chroma filter for JPEG/JFIF** averages samples over a 2×2 block. Shading represents the spatial extent of luma samples. The black dot indicates the effective subsampled chroma position, equidistant from the four luma samples. The outline represents the spatial extent of the result.

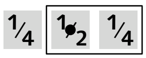


Figure 12.5 **A cosited chroma filter for BT.601, 4:2:2** causes each filtered chroma sample to be positioned coincident – *cosited* – with an even-numbered luma sample.

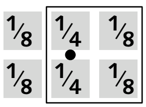


Figure 12.6 **A cosited chroma filter for MPEG-2, 4:2:0** produces a filtered result sample that is cosited horizontally, but sited interstitially in the vertical dimension.

In chroma subsampling, the encoder discards selected colour difference samples after filtering. A decoder approximates the missing samples by interpolation.

To perform 4:2:0 subsampling with minimum computation, some systems simply average C_B over a 2×2 block and average C_R over the same 2×2 block, as sketched in Figure 12.4 in the margin. To interpolate the missing chroma samples prior to conversion back to *R'G'B'*, low-end systems simply replicate the subsampled C_B and C_R values throughout the 2×2 quad. This technique is ubiquitous in JPEG/JFIF stillframes in computing, and is used in M-JPEG, H.261, and MPEG-1. This simple averaging process causes subsampled chroma to take an effective horizontal position halfway between two luma samples, what I call *interstitial* siting, not the cosited position standardized for studio video.

A simple way to perform 4:2:2 subsampling with horizontal cositing as required by BT.601 is to use weights of [$\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$], as sketched in Figure 12.5. 4:2:2 subsampling has the advantage of no interaction with interlaced scanning.

A cosited horizontal filter can be combined with [$\frac{1}{2}$, $\frac{1}{2}$] vertical averaging, as sketched in Figure 12.6, to implement 4:2:0 as used in MPEG-2.

Simple averaging filters like those of Figures 12.4, 12.5, and 12.6 have acceptable performance for stillframes, where any alias components that are generated remain stationary, or for desktop-quality video. However, in a moving image, an alias component introduced by poor filtering is liable to move at a rate different from the associated scene elements, and thereby produce a highly objectionable artifact. High-end digital video equipment uses sophisticated subsampling filters, where the subsampled C_B and C_R of a 2×1 pair in 4:2:2 (or of a 2×2 quad in 4:2:0) take contributions from several surrounding samples. The relationship of filter weights, frequency response, and filter performance will be detailed in *Filtering and sampling,* on page 191. These coefficients implement a high quality FIR filter suitable for 4:2:2 subsampling: [–1, 3, –6, 12, –24, 80, 128, 80, –24, 12, –6, 3, –1]/256.

Chroma in composite NTSC and PAL

The video literature often calls these quantities *chrominance*. That term has a specific meaning in colour science, so in video I prefer the term *modulated chroma*.

See *Introduction to composite NTSC and PAL,* on page 135. Concerning SECAM, see *SECAM*, on page 126 of *Composite NTSC and PAL: Legacy Video Systems*.

I introduced the colour difference components P_BP_R and C_BC_R, often called *chroma components*. They accompany luma in a component video system. I also introduced *UV* and *IQ* components; these are intermediate quantities in the formation of *modulated chroma*.

Historically, insufficient channel capacity was available to transmit three colour components separately. The NTSC technique was devised to combine the three colour components into a single *composite* signal; the PAL technique is both a refinement of NTSC and an adaptation of NTSC to 576*i* scanning. (In SECAM, the three colour components are also combined into one signal. SECAM is a form of composite video, but the technique has little in common with NTSC and PAL, and it is of little commercial importance today.)

NTSC and PAL encoders traditionally started with *R'G'B'* components. At the culmination of composite video, digital encoders started with $Y'C_BC_R$ components. NTSC or PAL encoding involves these steps:

- Component signals are matrixed and conditioned to form colour difference signals *U* and *V* (or *I* and *Q*).
- *U* and *V* (or *I* and *Q*) are lowpass-filtered, then *quadrature modulation* imposes the two colour difference signals onto an unmodulated colour subcarrier, to produce a *modulated chroma* signal, *C*.
- Luma and chroma are summed. In studio video, summation exploits the *frequency-interleaving* principle.

Composite NTSC and PAL signals were historically analog. During the 1990s, digital composite ($4f_{SC}$) systems were used; the $4f_{SC}$ scheme is now obsolete. As I mentioned in *Video system taxonomy,* on page 94, composite video has been supplanted by component video in consumers' premises and in industrial applications. For further information, see *Introduction to composite NTSC and PAL,* on page 135.

Introduction to component SD 13

In *Raster scanning,* on page 83, I introduced the concepts of raster scanning; in *Introduction to luma and chroma,* on page 121, I introduced the concepts of colour coding in video. This chapter combines the concepts of raster scanning and colour coding to form the basic technical parameters of the 480*i* and 576*i* SD systems. This chapter concerns modern systems that use component colour – digital $Y'C_BC_R$ (BT.601), or analog $Y'P_BP_R$. In *Introduction to composite NTSC and PAL,* on page 135, I will describe NTSC and PAL composite video encoding.

Scanning standards

The notation *CCIR* is often wrongly used to denote 576*i*25 scanning. The former CCIR (now ITU-R) standardized many scanning systems, not just 576*i*25.

Two scanning standards are in use for conventional analog television broadcasting in different parts of the world. The 480*i*29.97 system is used primarily in North America and Japan, and today accounts for roughly ¼ of all television receivers. The 576*i*25 system is used primarily in Europe, Asia, Australia, and Central America, and accounts for roughly ¾ of all television receivers. 480*i*29.97 (or 525/59.94/2:1) is colloquially referred to as *NTSC,* and 576*i*25 (or 625/50/2:1) as *PAL*; however, the terms NTSC and PAL properly apply to colour encoding and not to scanning standards. It is obvious from the scanning nomenclature that the line counts and field rates differ between the two systems: In 480*i*29.97 video, the field rate is exactly $^{60}/_{1.001}$ Hz; in 576*i*25, the field rate is exactly 50 Hz.

Several different standards for 480*i*29.97 and 576*i*25 digital video are sketched in Figure 13.1 overleaf.

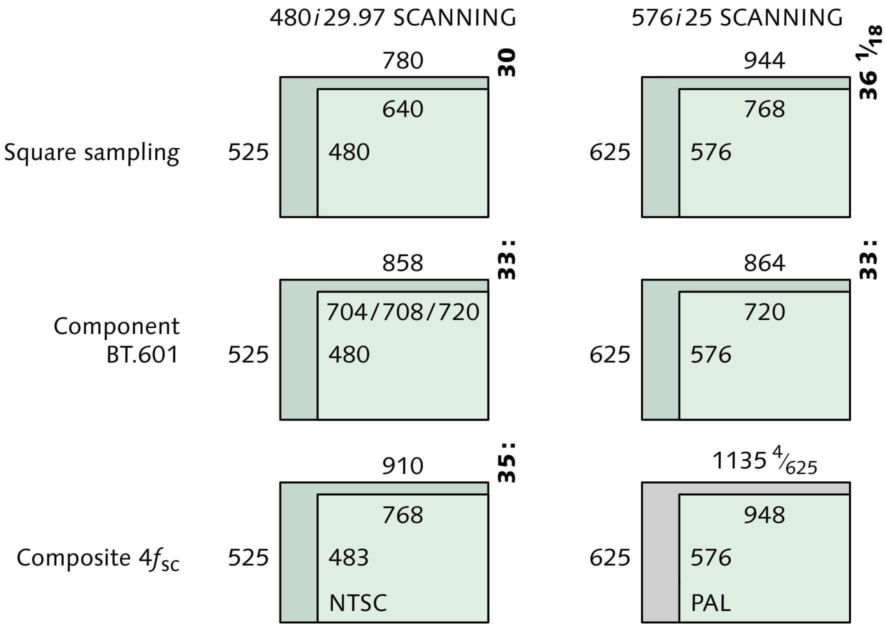


Figure 13.1 **SD digital video rasters** for 4:3 aspect ratio. 480*i*29.97 scanning is at the left, 576*i*25 at the right. The top row shows square sampling ("square pixels"). The middle row shows sampling at the BT.601 standard sampling frequency of 13.5 MHz. The bottom row shows sampling at four times the colour subcarrier frequency ($4f_{SC}$). Above each diagram is its count of samples per total line (S_{TL}); ratios among S_{TL} values are written vertically in bold numerals.

Analog broadcast of 480*i* usually uses NTSC colour coding with a colour subcarrier of about 3.58 MHz; analog broadcast of 576*i* usually uses PAL colour coding with a colour subcarrier of about 4.43 MHz. It is important to use a notation that distinguishes scanning from colour, because other combinations of scanning and colour coding are in use in large and important regions of the world. Brazil uses PAL-M, which has 480*i* scanning and PAL colour coding. Argentina uses PAL-N, which has 576*i* scanning and a 3.58 MHz colour subcarrier nearly identical to NTSC's subcarrier. In France, Russia, and other countries, SECAM is used. Production equipment is no longer manufactured for any of these obscure standards: Production in these countries is done using 480*i* or 576*i* studio equipment, either in the component domain or in 480*i* NTSC or 576*i* PAL. These studio signals are then *transcoded* prior to broadcast: The colour encoding is altered – for example, from PAL to SECAM – without altering scanning.

See *PAL-M*, *PAL-N* on page 125, and *SECAM* on page 126 of *Composite NTSC and PAL: Legacy Video Systems*. Consumer frustration with a diversity of functionally equivalent standards led to proliferation of multistandard TVs and VCRs in countries using these standards.

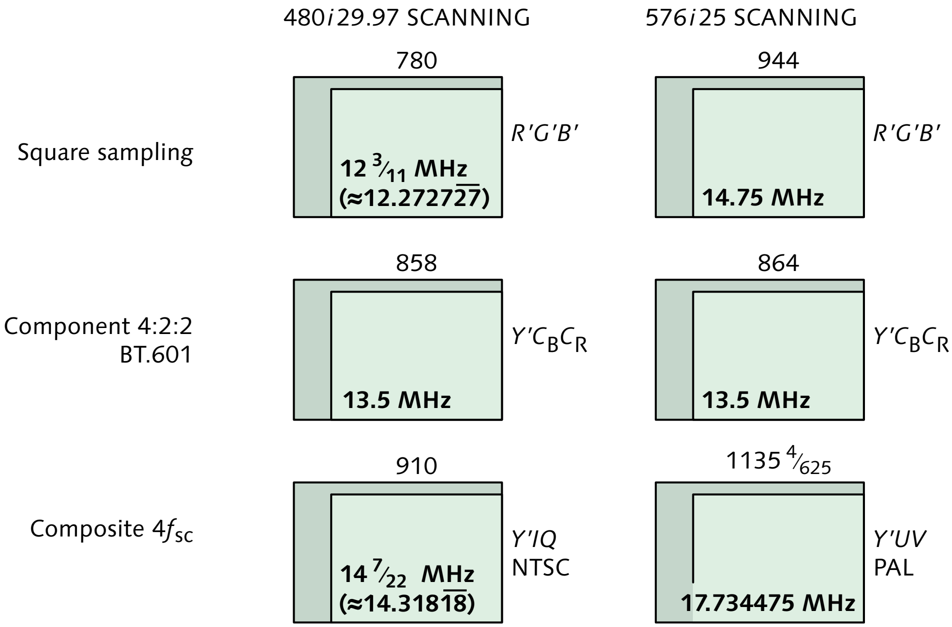


Figure 13.2 **SD sample rates** are shown for six different 4:3 standards, along with the usual colour coding for each standard. There is no realtime studio interface standard for square-sampled SD.

ITU-R Rec. BT.601-5, *Studio encoding parameters of digital television for standard 4:3 and widescreen 16:9 aspect ratios*.

Figure 13.1 indicates S_{TL} and S_{AL} for each standard. The S_{AL} values are the result of some complicated issues to be discussed in *Choice of S_{AL} and S_{PW} parameters* on page 380. For details concerning my reference to 483 active lines (L_A) in 480*i* systems, see *Picture lines,* on page 379.

Figure 13.2 above shows the standard 480*i*29.97 and 576*i*25 digital video sampling rates, and the colour coding usually associated with each of these standards. The 4:2:2, $Y'C_BC_R$ system for SD is standardized in *Recommendation BT.601* of the ITU Radiocommunication Sector (formerly CCIR). I call it *BT.601*.

With one exception, all of the sampling systems in Figure 13.2 have a whole number of samples per total line; these systems are *line-locked*. The exception is composite $4f_{SC}$ PAL sampling, which has a noninteger number ($1135\,^4\!/_{625}$) of samples per total line; this creates a huge nuisance for the system designer.

480*i* and 576*i* have gratuitous differences in many technical parameters, summarized in Table 13.1 overleaf.

† The EBU N10 component analog interface for $Y'P_BP_R$, occasionally used for 480*i*, has 7:3 picture-to-sync ratio.

‡ 480*i* video in Japan, and the EBU N10 component analog interface, have zero setup. See page 381.

System	*480i29.97*	*576i25*
Picture:sync ratio	10:4†	7:3
Setup, percent	7.5‡	0
Count of equalization, broad pulses	6	5
Line number 1, and 0_V, defined at:	First equalization pulse of field	First broad pulse of frame
Bottom picture line in:	First field	Second field

Table 13.1 **Gratuitous differences.** between 480*i* and 576*i*

Different treatment of interlace between 480*i* and 576*i* imposes different structure onto the picture data. The differences cause headaches in systems such as MPEG that are designed to accommodate both 480*i* and 576*i* images. In Figures 13.3 and 13.4 below, I show how field order, interlace nomenclature, and image structure are related. Figure 13.5 at the bottom of this page shows how MPEG-2 identifies each field as either *top* or *bottom*. In 480*i* video, the bottom field is the first field of the frame; in 576*i*, the top field is first. Figures 13.3, 13.4, and 13.5 depict just the image array (i.e., the active samples), without vertical blanking lines; MPEG makes no provision for halflines.

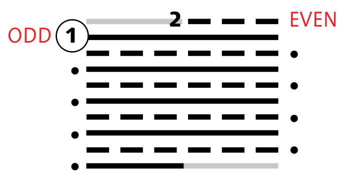


Figure 13.3 **Interlacing in 480*i*.** The first field (historically called *odd*, here denoted **1**) starts with a full picture line, and ends with a left-hand halfline containing the bottom of the picture. The second field (here dashed, historically called *even*), transmitted about $1/60$ s later, starts with a right-hand halfline containing the top of the picture; it ends with a full picture line.

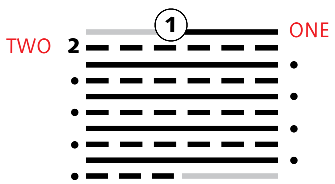


Figure 13.4 **Interlacing in 576*i*.** The first field includes a right-hand halfline containing the top line of the picture, and ends with a full picture line. The second field, transmitted $1/50$ s later, starts with a full line, and ends with a left-hand halfline that contains the bottom of the picture. (In 576*i* terminology, the terms *odd* and *even* are rarely used, and are best avoided.)

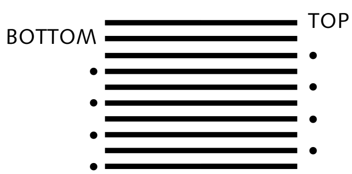


Figure 13.5 **Interlacing in MPEG-2** identifies a picture according to whether it contains the *top* or *bottom* picture line of the frame. Top and bottom fields are displayed in the order that they are coded in an MPEG-2 data stream. For frame-coded pictures, display order is determined by a one-bit flag *top field first*, typically asserted for 576*i* and negated for 480*i*.

Widescreen (16:9) SD

Programming in SD is intended for display at 4:3 aspect ratio. Prior to (and during) the development of HD, several schemes were devised to adapt SD to widescreen (16:9) material – *widescreen SD*. That term is misleading, though: Because there is no increase in pixel count, a so-called widescreen SD picture cannot be viewed with a picture angle substantially wider than regular (4:3) SD. (See page 75.) So widescreen SD does not deliver HD's major promise – that of dramatically wider viewing angle – and a more accurate term would be *wide aspect ratio SD*. The various schemes devised in the transition period are now obsolete. A discussion is found in *Widescreen (16:9) SD* on page 5 of *Composite NTSC and PAL: Legacy Video Systems*.

Square and nonsquare sampling

Computer graphics equipment now universally employs *square sampling* – that is, a sampling lattice where pixels are equally spaced horizontally and vertically. Square sampling of 480*i* and 576*i* is diagrammed in the top rows of Figures 13.1 and 13.2 on page 131.

See Table 15.1, on page 143, and the associated discussion.

Although ATSC's notorious Table 3 includes a 640×480 square-sampled image, no studio standard or realtime interface standard addresses square sampling of SD. For desktop video applications, I recommend sampling 480*i* video with exactly 780 samples per total line, for a nominal sample rate of $12\,^{3}\!/_{11}$ MHz – that is, $12.2727\overline{27}$ MHz. To accommodate full picture width in the studio, 648 samples are required; often, 640 samples are used with 480 picture lines. For square sampling of 576*i* video, I recommend using exactly 944 samples per total line, for a sample rate of exactly 14.75 MHz.

$$648 \approx 780 \cdot \left(1 - \frac{10.7\ \mu s}{63.555\overline{5}\ \mu s}\right)$$

$$767 = 944 \cdot \frac{52\ \mu s}{64\ \mu s}$$

MPEG-1, MPEG-2, DVD, and DV all conform to BT.601, which specifies nonsquare sampling. BT.601 sampling of 480*i* and 576*i* is diagrammed in the middle rows of Figures 13.1 and 13.2.

Composite digital video systems historically sampled at four times the colour subcarrier frequency ($4f_{SC}$), resulting in nonsquare sampling whose parameters are shown in the bottom rows of Figures 13.1 and 13.2. As I stated on page 128, composite $4f_{SC}$ systems are obsolete.

$$\frac{f_{S,601}}{4f_{SC,PAL\text{-}I}} = \frac{540000}{709379}$$

In 480*i*, the sampling rates for square sampling, BT.601, and $4f_{SC}$ are related by the ratio 30:33:35. The pixel aspect ratio of BT.601 480*i* is exactly $^{10}/_{11}$; the pixel aspect ratio of $4f_{SC}$ 480*i* is exactly $^{6}/_{7}$.

In 576*i*, the sampling rates for square sampling and 4:2:2 are related by the ratio 59:54, so the pixel aspect ratio of 576*i* BT.601 is precisely $^{59}/_{54}$. BT.601 and $4f_{SC}$ sample rates are related by the ratio in the margin, which is fairly impenetrable to digital hardware.

Most of this nonsquare sampling business has been put behind us: Most HD studio standards call for square sampling, and it is difficult to imagine any future studio standard being established with nonsquare sampling.

Resampling

Analog video can be digitized with square sampling simply by using an appropriate sample frequency. However, SD already digitized at a standard digital video sampling rate such as 13.5 MHz must be *resampled* – or *interpolated*, or in PC parlance, *scaled* – when entering the square-sampled desktop video domain. If video samples at 13.5 MHz are passed to a computer graphics system and then treated as if the samples are equally spaced vertically and horizontally, then picture geometry will be distorted. BT.601 480*i* video will appear horizontally stretched; BT.601 576*i* video will appear squished. In desktop video, often resampling in both axes is needed.

The ratio $^{10}/_{11}$ relates 480*i* BT.601 to square sampling: Crude resampling could be accomplished by simply dropping every eleventh sample across each scan line! Crude resampling from 576*i* BT.601 to square sampling could be accomplished by replicating 5 samples in every 54 (perhaps in the pattern 11-*R*-11-*R*-11-*R*-11-*R*-10-*R*, where *R* denotes a repeated sample). However, such sample dropping and stuffing techniques introduce aliasing. I recommend that you use a more sophisticated interpolator, of the type explained in *Filtering and sampling,* on page 191. Resampling could potentially be performed along either the vertical axis or the horizontal (transverse) axis; horizontal resampling is the easier of the two, as it processes pixels in raster order and therefore does not require any linestores.

Introduction to composite NTSC and PAL 14

NTSC stands for *National Television System Committee*. PAL stands for *Phase Alternate Line.* (Some sources say that PAL stands for *Phase Alternation* at *Line* rate, or perhaps even *Phase Alternating Line*).

SECAM is a composite technique of sorts, though it has little in common with NTSC and PAL, and it is now obsolete. See "SECAM," in Chapter 12 of *Composite NTSC and PAL: Legacy Video Systems*.

In *component* video, the three colour components are kept separate. Video can use *R'G'B'* components directly, but three signals are expensive to record, process, or transmit. Luma (*Y'*) and colour difference components based upon *B'–Y'* and *R'–Y'* can be used to enable subsampling: Luma is maintained at full data rate, and the two colour difference components are subsampled. Even after chroma subsampling, video has a fairly high information rate (data rate, or "bandwidth"). To further reduce the information rate, the composite NTSC and PAL colour coding schemes use *quadrature modulation* to combine the two colour difference components into a single *modulated chroma* signal, then use *frequency interleaving* to combine luma and modulated chroma into a *composite* signal having roughly ⅓ the data rate – or in an analog system, ⅓ the bandwidth – of *R'G'B'*.

Composite encoding was invented to address three main needs. First, there was a need to limit transmission bandwidth. Second, it was necessary to enable black-and-white receivers already deployed by 1953 to receive colour broadcasts with minimal degradation. Third, it was necessary for newly introduced colour receivers to receive the then-standard black-and-white broadcasts. Composite encoding was necessary in the early days of television, and it has proven highly effective for broadcast. NTSC and PAL are entrenched in billions of consumer electronic devices. However, component digital video has overtaken composite techniques, and composite NTSC and PAL are now "legacy" techniques.

Composite NTSC or PAL encoding has three major disadvantages compared to component video. First, encoding introduces some degree of mutual interference between luma and chroma. Once a signal has been encoded into composite form, the NTSC or PAL *footprint* is imposed: *Cross-luma* and *cross-colour* errors are irreversibly impressed on the signal. Second, it is impossible to directly perform many processing operations in the composite domain; even to reposition or resize a picture requires decoding, processing, and reencoding. Third, digital compression techniques such as JPEG and MPEG cannot be directly applied to composite signals, and the artifacts of NTSC and PAL encoding are destructive to MPEG encoding.

The bandwidth to carry separate colour components is now easily affordable, and composite encoding is now obsolete in the studio. To avoid NTSC and PAL artifacts, to facilitate image manipulation, and to enable compression, composite video has been superseded by *component video*, where three colour components $R'G'B'$, or $Y'C_BC_R$ (in digital systems), or $Y'P_BP_R$ (in analog systems), are kept separate. I hope you can manage to avoid composite NTSC and PAL, and skip this chapter!

By NTSC and PAL, I do not mean 480*i* and 576*i*, or 525/59.94 and 625/50!

The terms *NTSC* and *PAL* properly denote colour encoding. Unfortunately, they are often used incorrectly to denote scanning standards. PAL encoding has been used with both 576*i* scanning (with two different subcarrier frequencies) and 480*i* scanning (with a third subcarrier frequency); PAL alone is ambiguous.

When I use the term *PAL* in this chapter, I refer only to 576*i* PAL-B/G/H/I. Variants of PAL used for broadcasting in South America are discussed in *PAL-M*, *PAL-N*, on page 125 of *Composite NTSC and PAL: Legacy Video Systems*. PAL variants in consumer devices are discussed in *Consumer analog NTSC and PAL* in *Composite NTSC and PAL: Legacy Video Systems*.

In principle, NTSC or PAL colour coding could be used with any scanning standard. However, in practice, NTSC and PAL are used only with 480*i* and 576*i* scanning, and the parameters of NTSC and PAL encoding are optimized for those scanning systems. This chapter introduces composite encoding. Details can be found in *Composite NTSC and PAL: Legacy Video Systems*.

NTSC and PAL encoding

NTSC or PAL encoding involves these steps:

- $R'G'B'$ component signals are matrixed and filtered, or $Y'C_BC_R$ or $Y'P_BP_R$ components are scaled and filtered, to form luma (Y') and colour difference signals (U and V, or in certain NTSC systems, I and Q).

- U and V (or I and Q) colour difference signals are modulated onto a pair of intimately related continuous-wave colour subcarriers, typically at a frequency of about 3.58 MHz in 480*i*29.97 or 4.43 MHz in 576*i*25, to produce a modulated chroma signal, C. (See the left side of Figure 14.1 overleaf.)
- Luma and modulated chroma are summed to form a composite NTSC or PAL signal. (See the right side of Figure 14.1.) Summation of luma and chroma is liable to introduce a certain degree of mutual interference, called *cross-luma* and *cross-colour;* these artifacts can be minimized through *frequency interleaving*, to be described.

NTSC and PAL decoding

HUE (or TINT, or PHASE) and CHROMA (or COLOUR, or SATURATION) controls properly apply only to NTSC/PAL decoding. They have no place in modern component video systems. See *Composite decoder adjustment using colourbars*, on page 422, and *NTSC and PAL Chroma modulation* in *Composite NTSC and PAL: Legacy Video Systems*.

NTSC or PAL decoding involves these steps:

- Luma and modulated chroma are separated. Crude separation can be accomplished using a *notch filter*. Alternatively, frequency interleaving can be exploited to provide greatly improved separation; in NTSC, such a separator is a *comb filter*. (In an S-video interface, luma and modulated chroma are already separate.)
- Chroma is demodulated to produce UV, IQ, P_BP_R, or C_BC_R baseband colour difference components.
- If $R'G'B'$ components are required, the baseband colour difference components are interpolated, then luma and the colour difference components are dematrixed.

S-video interface

S-video involves NTSC or PAL chroma modulation; however, luma and modulated chroma traverse separate paths across the interface instead of being summed; the S-video interface bypasses the third step of *NTSC and PAL encoding* above. Cross-luma and cross-colour artifacts are avoided. Figure 14.2 sketches the encoder and decoder arrangement. The S-video interface is widely implemented in consumer video equipment.

Frequency interleaving

When luma and modulated chroma are summed, a certain amount of mutual interference is introduced.

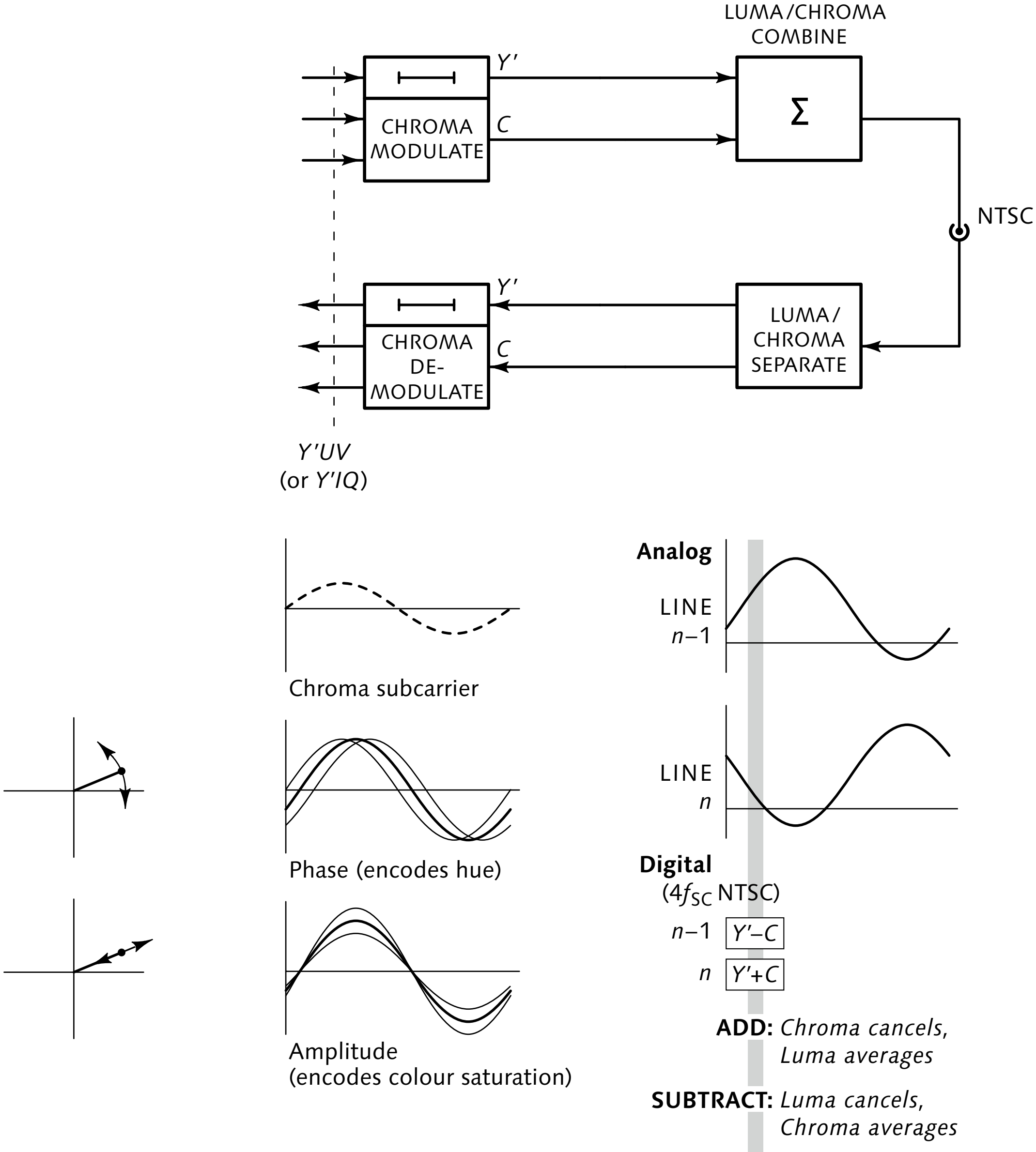
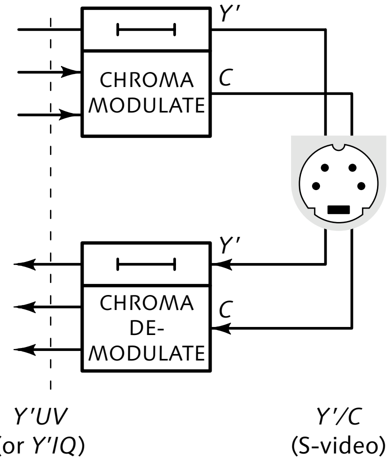


Figure 14.1 **NTSC chroma modulation and frequency interleaving** are applied, successively, to encode luma and a pair of colour difference components into NTSC composite video. First, the two colour difference signals are modulated onto a colour subcarrier. If the two colour differences are interpreted in polar coordinates, hue angle is encoded as subcarrier phase and colour saturation is encoded as subcarrier amplitude. (A sample of the unmodulated subcarrier, *burst*, is included in the composite signal.) Then, modulated chroma is summed with luma. Frequency interleaving leads to line-by-line phase inversion of the unmodulated colour subcarrier, thence to the modulated subcarrier. Summation of adjacent lines tends to cause modulated chroma to cancel, and tends to cause luma to average.

Figure 14.2 **The S-video interface** involves NTSC or PAL chroma modulation; however, luma and modulated chroma traverse separate paths across the interface instead of being summed. There are no fewer than three versions of S-video: S-video-525, S-video-525-J, and S-video-625; these are detailed in "480i NTSC composite video," in Chapter 7 of *Composite NTSC and PAL: Legacy Video Systems*.



In PAL, all but the most sophisticated comb filters separate *U* and *V*, not luma and chroma. See "NTSC and PAL Chroma modulation," Chapter 5 of *Composite NTSC and PAL: Legacy Video Systems*.

Interference is minimized by arranging for *frequency interleaving*, which is achieved when the colour subcarrier frequency and the line rate are *coherent* – that is, when the unmodulated colour subcarrier is phase-locked to a carefully chosen rational multiple of the line rate – an integer multiple of half the line rate for NTSC, and an integer multiple of $1/4$ the line rate in PAL. Coherence is achieved in the studio by deriving both the sync and colour subcarrier from a single master clock.

In NTSC, frequency interleaving enables use of a comb filter to separate luma and chroma: Adjacent lines are summed (to form vertically averaged luma) and differenced (to form vertically averaged chroma), as suggested at the bottom right of Figure 14.1.

In industrial and consumer video, the subcarrier often free-runs with respect to line rate, and the advantages of frequency interleaving are lost. Most forms of analog videotape recording introduce timebase error; left uncorrected, this also defeats frequency interleaving.

Composite analog SD

Composite analog 480*i* NTSC and 576*i* PAL is widely deployed in consumer equipment (such as television receivers and VCRs) and was used for terrestrial VHF/UHF broadcasting and cable television for many decades. Details are found in "Analog SD broadcast standards," Chapter 12 of *Composite NTSC and PAL: Legacy Video Systems*.

This page intentionally left blank

Introduction to HD 15

FUJIO, TAKASHI, J. ISHIDA, T. KOMOTO, and TAIJI NISHIZAWA (1980), *High definition television system – signal standards and transmission*, *SMPTE Journal*, **89** (8): 579–584 (Aug.).

FUJIO, TAKASHI (1981), "High definition television systems: desirable standards, signal forms, and transmission systems," in *IEEE Tr. Comm.* **29** (12): 1882–1891 (Dec.).

FUJIO, TAKASHI, et al. (1982), *High Definition television*, NHK Science and Technical Research Laboratories Technical Monograph 32 (June).

Developmental HD systems had 1125/60.00/2:1 scanning, an aspect ratio of 5:3, and 1035 active lines. The alternate 59.94 Hz field rate was added later. Aspect ratio was changed to 16:9 to achieve international agreement upon standards. A count of 1080 image rows was eventually agreed upon to provide square sampling. The developmental 1035*i* (1125/60) system, standardized in SMPTE 240M, was discussed in the first edition of this book.

This chapter outlines the 1280×720 and 1920×1080 image formats for high-definition (HD) television, and introduces the scanning parameters of the associated video systems such as 720*p*60 and 1080*i*30.

Today's HD systems stem from research directed by Dr. Fujio at NHK (Nippon Hoso Kyokai, the Japan Broadcasting Corporation). HD has about twice the vertical and twice the horizontal resolution of conventional television, a picture aspect ratio of 16:9, at least two channels of CD-quality audio, and a frame rate of 23.976 Hz or higher. By my definition, HD has ¾-million pixels or more. NHK conceived HD to have interlaced scanning; however, progressive HD systems have since emerged.

Studio HD has a sampling rate of 74.25 MHz, 5.5 times that of the BT.601 standard for SD. HD has a pixel rate of about 60 megapixels per second. Apart from a few annoying exceptions, parameters of *R'G'B'* coding are similar those of SD standards – in fact, several parameters adopted for HD in BT.709 have essentially been retrofitted into SD. Details concerning scanning, sample rates, and interface levels of HD will be presented in *1280×720 HD* on page 467 and *1920×1080 HD* on page 473. Unfortunately, the parameters for $Y'C_BC_R$ colour coding for HD differ from the parameters for SD! Details will be provided in *Component video colour coding for HD,* on page 369.

HD scanning

A great debate took place in the 1980s and 1990s concerning whether HD should have interlaced or

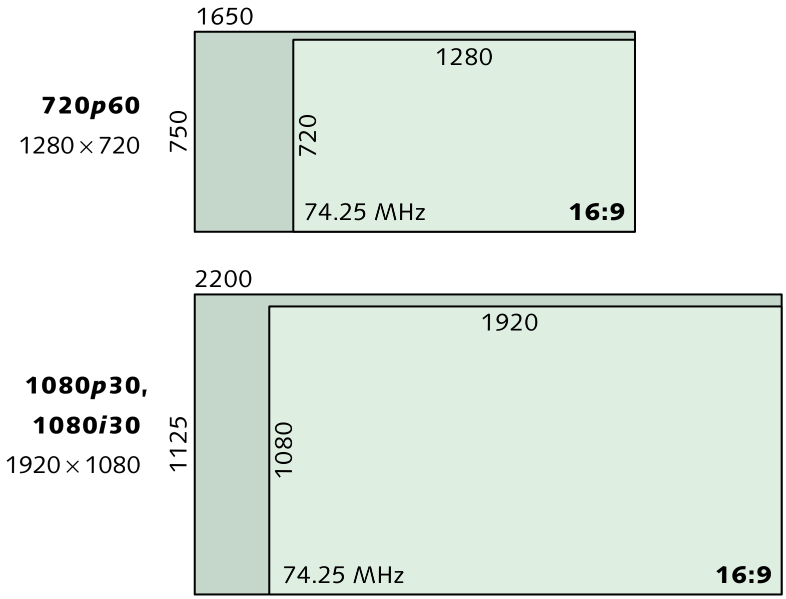


Figure 15.1 **HD rasters at 30 and 60 frames per second** are standardized in two formats, 1280×720 (1 Mpx, always progressive), and 1920×1080 (2 Mpx, interlaced or progressive). The latter is often denoted 1080*i*, but the standards accommodate progressive scan. These sketches are scaled to match Figures 13.1 and 13.2; pixels in these sketches have identical area.

progressive scanning. At given flicker and data rates, interlace offers some increase in static spatial resolution, as suggested by Figure 8.8 on page 91. Broadcasters have historically accepted the motion artifacts and spatial aliasing that accompany interlace, in order to gain some static spatial resolution. In the HD debate, the computer industry and the creative film community were resolutely set against interlace. Eventually, both camps compromised to some degree and HD standards were established to accommodate both interlaced and progressive image formats. To be commercially viable a receiver must decode both formats, though there is no "legal" requirement to do so.

In *Numerology of HD scanning,* on page 395, I explain the origin of the numbers in Figure 15.1.

Figure 15.1 above sketches the rasters of the 1 Mpx progressive system (1280×720, 720*p*60) and the 2 Mpx interlaced system (1920×1080, 1080*i*30) that were agreed upon. 1280×720 is very simply related to 1920×1080: 1280 is two thirds of 1920, and 720 is two thirds of 1080.

Image format	Progressive/interlace	‡Frame rate [Hz]	Image aspect ratio	Sampling
640×480	*p*	24, 30, 60	4:3	Square
	i	30		
704×480	*p*	24, 30, 60	4:3	Nonsquare
	i	30		
	p	24, 30, 60	16:9	Nonsquare
	i	30		
1280×720	*p*	24, 30, 60	16:9	Square
1920×1080	*p*	24, 30	16:9	Square
	i	30		

‡ Frame rates modified by the ratio $^{1000}/_{1001}$ – that is, frame rates of 23.976 Hz, 29.97 Hz, and 59.94 Hz – are permitted.

Table 15.1 **ATSC A/53 Table 3** defines the so-called 18 formats – including 12 SD formats – for digital television in the U.S. I find the layout of ATSC's Table 3 to be hopelessly contorted, so I rearranged it. ATSC specifies 704 S_{AL} for several SD formats, instead of BT.601's 720 S_{AL}; see page 380. ATSC standard A/53 doesn't accommodate 25 Hz and 50 Hz frame rates, but A/63 does.

ATSC A/53, *Digital Television Standard*.

In addition to the 1 Mpx (progressive) and 2 Mpx (interlaced) systems, several SD scanning systems and several additional frame rates were included in the ultimate ATSC standards for U.S. digital television (DTV). Table 15.1 summarizes the "18 formats" that are found in Table 3 of the ATSC's A/53 standard.

The 1920×1080 system was conceived as interlaced-only (1080*i*30), but was adapted to 24 and 30 Hz progressive scan (1080*p*24, 1080*p*30) using the standard 74.25 MHz sample rate. The adaptation to 24 Hz was seminal to digital cinema. Figure 15.2 overleaf sketches raster structures for 24 Hz and 25 Hz systems; Table 15.2 summarizes the scanning parameters.

In Sony's legacy HDCAM system, the 1920×1080 image was downsampled to 1440×1080, and colour difference signals were subsampled 3:1:1, prior to compression. This was an internal representation only; there was no corresponding uncompressed external interface standard. The current Sony HDCAM SR format represents 1920×1080 image data directly at 4:2:2 (or in some variations 4:4:4), and alleviates the need for such downsampling.

SMPTE ST 274 provides for carriage of a 1920×1080 image at a frame rate of 25 Hz: 1125 total lines are

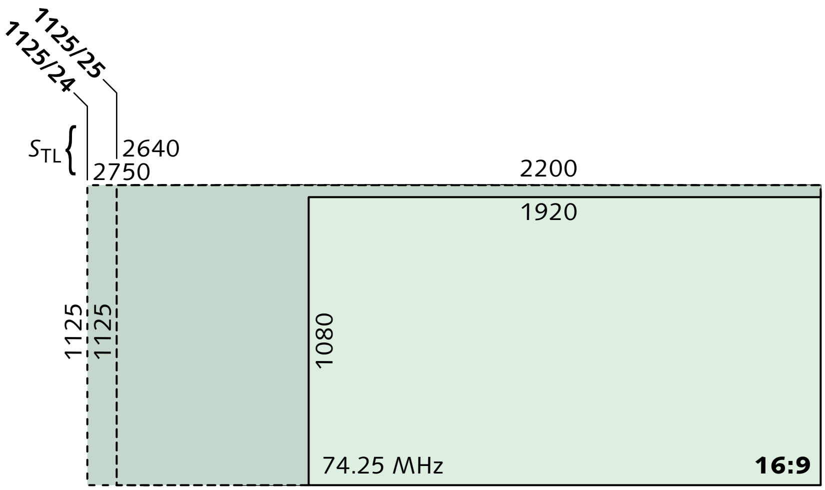


Figure 15.2 **HD rasters at 24 Hz and 25 Hz** carry an array of 1920×1080 active samples, using a 74.25 MHz sampling rate at the interface. For 24 Hz (1080*p*24), the 1920×1080 array is carried in an 1125/24 raster. For 25 Hz, the array is carried in an 1125/25 raster.

retained, and S_{TL} is increased to 2640. This yields the 1080*p*25 format, using an 1125/25 raster. Scanning can be either progressive or interlaced; with progressive scanning, the signal is usually interfaced using the progressive segmented frame (PsF) scheme that I introduced on page 94.

For 24 Hz, 1125 total lines are retained, and S_{TL} is increased to 2750 to achieve the 24 Hz frame rate. This yields the 1080*p*24 format, in an 1125/24 raster. This system is used in digital cinema (D-cinema). A variant at 23.976 Hz is accommodated.

$$\frac{24}{1.001} \approx 23.976$$

Table 15.2 summarizes the scanning parameters for 720*p*, 1080*i*, and 1080*p* systems. Studio interfaces for HD will be introduced in *SDI and HD-SDI interfaces,* on page 429.

Colour coding for BT.709 HD

The conventional view of BT.709 “encoding” involves the OECF in the standard. In *Image acquisition and presentation,* on page 19, the argument is made that what matters is the *display* process. Faithful presentation of completed program material requires that the display EOCF be standardized. A standard camera OECF (such as that in BT.709) is useful for engineering

System	Scanning	SMPTE standard	S_{TL}	L_T	N_C (S_{AL})	N_R (L_A)
720*p*60	750/60/1:1	SMPTE ST 296	1650	750	1280	720
1080*i*30	1125/60/2:1	SMPTE ST 274	2200	1125	1920	1080
1080*p*60¶	1125/60/1:1	SMPTE ST 274	2200	1125	1920	1080
1080*p*30	1125/30/1:1	SMPTE ST 274	2200	1125	1920	1080
1080*i*25	1125/25/2:1	SMPTE ST 274	2640	1125	1920	1080
1080*p*25	1125/25/1:1	SMPTE ST 274	2640	1125	1920	1080
1080*p*24	1125/24/1:1	SMPTE ST 274	2750	1125	1920	1080

Table 15.2 **HD scanning parameters.** are summarized. SMPTE ST 274 includes a progressive 2 Mpx, 1080*p*60 system with 1125/60/1:1 scanning, flagged with ¶ above; this system is not permitted for ATSC broadcasting. Each of the 24, 30, and 60 Hz systems above has an associated system at $^{1000}/_{1001}$ of that rate.

purposes, but has no impact upon faithful presentation. In practice, so-called BT.709 program material has *R'G'B'* data values established so that the inrtended image appearance is obtained when those *R'G'B'* values are displayed through a reference EOCF (for example, that of BT.1886) into a known set of viewing conditions (unfortunately, not yet standardized).

BT.709 defines $Y'C_BC_R$ colour coding. Unfortunately, the luma coefficients standardized in BT.709 – and the C_BC_R scale factors derived from them – differ from those of SD. $Y'C_BC_R$ coding now comes in two flavors: coding for small (SD) pictures, and coding for large (HD) pictures. I will present details concerning this troublesome issue in *SD and HD luma chaos,* on page 350.

This page intentionally left blank

Introduction to video compression 16

Directly storing or transmitting digital video requires fairly high data capacity – about 20 megabytes per second for SD, or about 120 megabytes per second for HD. Here is a rule of thumb that relates storage capacity and data rate: Eight, 2000-ft reels of motion picture print film can carry a $133\,^1/_3$ minute movie; there are 8 bits in a byte and 60 seconds in a minute; and $^{60}/_8 \cdot 133\,^1/_3$ is 1000. So one megabit per second equals one gigabyte per movie – whether compressed or not! Similarly, one gigabit per second equals one terabyte per movie.

A rule of thumb that relates data rate to storage capacity:

$$^{\mathrm{Mb}}/_{\mathrm{s}} = {}^{\mathrm{GB}}/_{\mathrm{movie}}$$

$$^{\mathrm{Gb}}/_{\mathrm{s}} = {}^{\mathrm{TB}}/_{\mathrm{movie}}$$

Economical storage or transmission requires compression. This chapter introduces the JPEG, M-JPEG, MPEG, and H.264 compression techniques.

In previous chapters, we have discussed representation of image data in a rather small number of colour components (say, three); a rather small number of bits per component (say 8 or 10); perceptual coding by way of a nonlinear EOCF; and chroma subsampling yielding a data rate reduction of around 50%. In video terminology, all of these techniques are termed – paradoxically, perhaps – to be *uncompressed* video. *Compression* involves transform techniques such as the discrete cosine transform (DCT) and – in the case of JPEG 2000 – the discrete wavelet transform (DWT).

Data compression

Data compression has the goal of reducing the number of bits required to store or convey text, numeric, binary, image, sound, or other data. High performance is obtained by exploiting statistical properties of the data.

The reduction comes at the expense of some computational effort to compress and decompress. Data compression is, by definition, lossless: Decompression recovers exactly, bit for bit (or byte for byte), the data that was presented to the compressor.

Salomon, David (2008), *A Concise Introduction to Data Compression* (Springer).

Sayood, Khalid (2005), *Introduction to Data Compression*, Third edition (Elsevier/Morgan-Kaufmann).

Binary data typical of general computer applications often has patterns of repeating byte strings. Most data compression techniques, including *run-length encoding* (RLE) and *Lempel-Ziv-Welch* (LZW), accomplish compression by taking advantage of repeated strings; performance is highly dependent upon the data being compressed.

Image compression

Image data typically has strong vertical, horizontal, and spatial correlations among samples of the same colour component. When the RLE and LZW algorithms are applied to bilevel or pseudocolour image data stored in scan-line order, horizontal correlation among pixels can be exploited to some degree; such techniques usually result in modest compression (perhaps 2:1).

A data compression algorithm can be designed to exploit the statistics of image data, as opposed to arbitrary binary data; improved compression is then possible. For example, the ITU-T fax standard for bilevel image data exploits vertical and horizontal correlation to achieve typical compression ratios higher than RLE or LZW typically achieve. In the absence of channel errors, data compression (even of images) is lossless, by definition: Decompression reproduces, bit-for-bit, the data presented to the compressor.

Lossy compression

The term "perceptually lossless" signifies an attempt to minimize the perceptibility of compression errors. There are no standards or industry practices to determine to what extent that goal is achieved. Thus, the term is indistinct.

Lossless data compression can be optimized to achieve modest compression of continuous-tone (greyscale or truecolour) image data. However, if exact reconstruction is not required, the characteristics of human perception can be exploited to achieve dramatically higher compression ratios: Image or sound data can be subject to *lossy* compression, provided that any impairments introduced are not overly perceptible. Lossy compression techniques are not appropriate for bilevel or pseudocolour images; however, they are very effective for greyscale or truecolour images, both stills and video.

Format	Uncompressed data rate [MB/s]	Compression ratio Motion-JPEG	MPEG-2	H.264
SD (480*i*30, 576*i*25)	20	15:1 (e.g., DV25)	45:1 (e.g,. DVD)	90:1
HD (720*p*60, 1080*i*30)	120	20:1	75:1 (e.g., ATSC)	100:1 (e.g., Blu-ray)

Table 16.1 **Approximate compression ratios** for SD and HD video distribution systems

Transform techniques are effective for compression of continuous-tone (greyscale or truecolour) image data. The *discrete cosine transform* (DCT) has been developed and optimized over the last few decades; it is the method of choice for continuous-tone image compression. *JPEG* refers to a lossy compression method for still images. *MPEG* refers to a lossy compression standard for video sequences; MPEG-2 is used in digital television distribution (e.g., ATSC and DVB), and in DVD. *H.264* refers to a lossy compression standard for video sequences. H.264 is highly effective for HD; it is used in satellite, cable, and telco (IPTV) systems, and in Blu-ray. These techniques will all be described in subsequent sections.

Internet protocol television (IPTV) concerns video and audio delivered over TCP/IP networks.

Table 16.1 compares typical compression ratios of M-JPEG and MPEG-2, for SD and HD.

In the context of compression of video or audio, the term *codec* refers to an en*CO*der and/or a *DEC*oder.

Encoders and decoders in compression systems are not to be confused with composite video (NTSC or PAL) encoders or decoders.

JPEG

JPEG stands for *Joint Photographic Experts Group*, constituted by ISO and IEC in collaboration with ITU-T (the former CCITT).

In 1992, the JPEG committee adopted a standard based upon DCT transform coding, suitable for compressing greyscale or truecolour still images. This was before the world-wide web: The standard was expected to be used for colour fax! JPEG was quickly adopted and widely deployed for still images in desktop graphics and digital photography. The *M-JPEG* variant can be used for motion sequences; the DV scheme uses an M-JPEG-like algorithm. Details are presented in *JPEG and motion-JPEG (M-JPEG) compression,* on page 491.

A JPEG compressor ordinarily transforms *R'G'B'* to $Y'C_BC_R$, then applies 4:2:0 chroma subsampling to effect 2:1 compression prior to the transform coding steps. (In desktop graphics, this 2:1 factor is included in the compression ratio.) JPEG has provisions to compress *R'G'B'* data directly, without subsampling.

Compression ratio	Quality/application	Example SD tape formats
2:1	"Visually lossless" studio video	Digital Betacam
3.3:1	Excellent-quality studio video	DVCPRO50, D-9 (Digital-S)
6.6:1	Good-quality studio video; consumer digital video	D-7 (DVCPRO), DVCAM, consumer DV

Table 16.2 **Approximate compression ratios of M-JPEG** for SD applications

Motion-JPEG

The JPEG algorithm – though not the ISO/IEC JPEG standard – has been adapted to compress motion video. Motion-JPEG simply compresses each field or frame of a video sequence as a self-contained compressed picture – each field or frame is *intra-coded*. Because pictures are compressed individually, an M-JPEG video sequence can be easily edited; however, no advantage is taken of temporal coherence.

Video data is almost always presented to an M-JPEG compression system in $Y'C_BC_R$ subsampled form. (In video, the 2:1 factor due to chroma subsampling is generally not included in the compression ratio.)

JPEG and motion-JPEG (M-JPEG) compression is described on page 491. DV compression is described on page 505.

The M-JPEG technique achieves compression ratios ranging from about 2:1 to about 20:1. The 20 MB/s data rate of SD can be compressed to about 20 Mb/s, suitable for recording on consumer digital videotape (e.g., DVC). M-JPEG compression ratios and tape formats are summarized in Table 16.2.

JPEG 2000

Taubman, David S. and Marcellin, Michael W. (2002), *JPEG-2000: Image Compression Fundamentals, Standards and Practice* (Norwell, Mass.: Kluwer).

Between 1995 and 2000, the JPEG committee developed a compression standard for continuous-tone colour still images. The effort culminated in the JPEG 2000 standard, which is based upon discrete wavelet transform (DWT) techniques. DCI standards for digital cinema use JPEG 2000 compression. An adaptation of JPEG 2000 accommodates motion sequences, where each (progressive) frame is coded individually without reference to any other frame. Although the "core" JPEG 2000 coding system is intended to be royalty and license-free, intellectual property rights (IPR) concerns have inhibited JPEG 2000 commercialization.

Mezzanine compression

Uncompressed recording is taken to be the upper floor; formats for distribution are at ground level. The mezzanine lies between.

DV and its derivatives and relatives are common in the studio. In the last several years, several software-based codecs suitable for acquisition and postproduction have emerged: Dirac/VC-2, DNxHD/VC-3, and Apple ProRes. The term *mezzanine* is used for such codecs, signifying compression rates intermediate between uncompressed recording and consumer distribution.

Dirac PRO (VC-2)

An open-source development project led by the BBC developed video compression technology collectively named *Dirac* (in honour of the Nobel prize-winning physicist) for lossy mezzanine-level intra-frame wavelet compression. Dirac PRO handles a wide range of video formats, but is optimized to compress 10-bit, 4:2:2 1080*p* video to bit rates between about 50 Mb/s and 165 Mb/s. SMPTE has standardized Dirac PRO in the ST 2042 series of standards (also known as VC-2).

The Dirac PRO bitstream includes parameter values to identify the complexity of the coded bit-stream through the use of profiles and levels. These values enable a decoder to easily establish whether it has the capability to decode the bit-stream. Profiles and levels are defined.

DNxHD (VC-3)

Avid implemented a family of compression systems that are now widely deployed in postproduction. One of these, denoted DNxHD by Avid, has been standardized by SMPTE as ST 2042-1, colloquially called VC-3. Bit rate for HD ranges from about 60 Mb/s to 220 Mb/s. Compressed data is typically carried in MXF files.

Apple ProRes

Apple implemented a set of proprietary variable bit rate, intra-frame DCT-based codecs called *ProRes,* now used fairly widely in acquisition and in postproduction. ProRes resembles motion-JPEG; however, the scheme is proprietary and details aren't published. The scheme was designed for implementation in software; however, hardware-based codecs are commercially available.

RedCode; CineForm

Several intraframe wavelet-based codecs have been developed and deployed for digital cinema acquisition, and can be used for HD. Red's RedCode is a proprietary form of JPEG 2000; GoPro's CineForm also uses wavelet compression. Both of these schemes individually compress Bayer mosaic data prior to demosaicking.

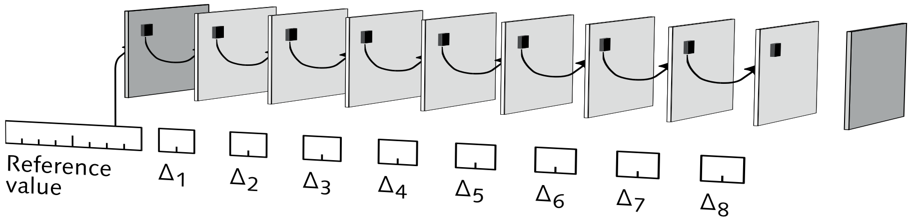


Figure 16.1 **Interpicture coding** exploits the similarity between successive pictures in video. First, a reference picture is transmitted (typically using intrapicture compression). Then, pixel differences to successive pictures are computed by the encoder and transmitted. The decoder reconstructs successive pictures by accumulating the differences. The scheme is effective provided that the difference information can be coded more compactly than the raw picture information.

MPEG

Apart from scene changes, there is a statistical likelihood that successive pictures in a video sequence are very similar. In fact, it is *necessary* that successive pictures are mostly similar: If this were not the case, human vision could make no sense of the sequence!

The efficiency of transform coding can be increased by a factor of 10 or more by exploiting the inherent temporal redundancy of video. The *MPEG* standard was developed by the *Moving Picture Experts Group* within ISO and IEC. In MPEG, intra-frame compression is used to provide an initial, self-contained picture – a *reference* picture, upon which predictions of succeeding pictures can be based. The encoder then transmits pixel differences, that is, prediction errors, or *residuals* – from the reference, as sketched in Figure 16.1. The method is termed *interframe coding.* (In interlaced video, differences between *fields* may be used, so the method is more accurately described as *interpicture* coding.)

The M in MPEG stands for *moving*, not *motion!* If you read a document that errs in this detail, the accuracy of the document is suspect!

Once a reference picture has been received by the decoder, it provides a basis for predicting succeeding pictures. This estimate is improved when the decoder receives the residuals. The scheme is effective provided that the residuals can be coded more compactly than the raw picture information.

Motion in a video sequence causes displacement of scene elements with respect to the image array. A fast-moving image element may easily move 20 pixels in one frame time. In the presence of motion, a pixel at a certain location may take quite different values in successive pictures. Motion is liable to cause prediction

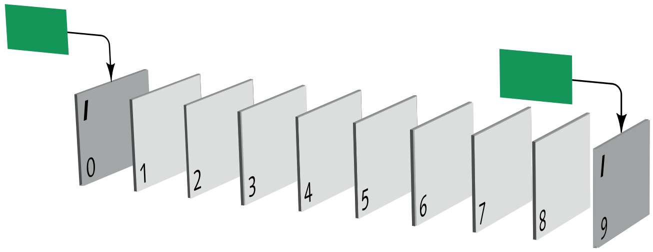


Figure 16.2 **An MPEG group of pictures (GoP).** The GoP depicted here has nine pictures, numbered 0 through 8. Picture 9 is the first picture of the next GoP. I-picture 0 is decoded from the coded data depicted as a green block. Here, the *intra-count* (N) is 9.

error information to grow in size to the point where the advantage of interframe coding would be negated.

One form of motion-compensated prediction forms an integer-valued MV, and copies a block of pixels from a reference picture for use as a prediction block. A more sophisticated form computes motion vectors to half-pixel precision, and interpolates between reference pixels (for example, by averaging). The latter is *motion-compensated interpolation.* Although the simple technique involves just copying, it's also called interpolation.

However, image elements tend to retain their spatial structure even when moving. MPEG overcomes the problem of motion between pictures by using *motion-compensated prediction* (MCP). The encoder is equipped with *motion estimation* (ME) circuitry that computes *motion vectors*. The encoder then displaces the pixel values of the reference picture by the estimated motion – a process called *motion compensated interpolation* – then computes residuals from the motion-compensated reference. The encoder compresses the residuals using a JPEG-like technique, then transmits the motion vectors and the compressed residuals.

Based upon the received motion vectors, the decoder mimics the motion compensated interpolation of the encoder to obtain a predictor much more effective than the undisplaced reference picture. The received residuals are then applied (by simple addition) to reconstruct an approximation of the encoder's picture.

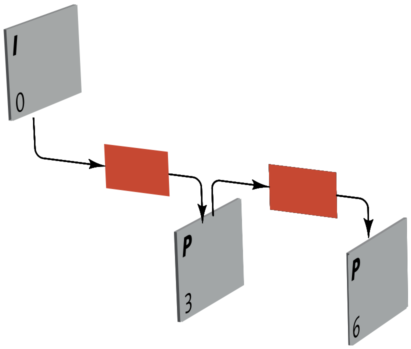
The MPEG suite of standards specifies the properties of a compliant bitstream and the algorithms that are implemented by a decoder. Any encoder that produces compliant ("legal") bistreams is considered MPEG-compliant, even if it produces poor-quality images.

Picture coding types (I, P, B)

When encoding interlaced source material, an MPEG-2 encoder can choose to code each field as a picture or each frame as a picture, as I will describe on page 518. In this chapter, and in Chapter 47, the term *picture* can refer to either a field or a frame.

In MPEG, a video sequence is partitioned into successive *groups of pictures* (GoPs). The first picture in each GoP is coded using a JPEG-like algorithm, independently of other pictures. This is an *intra* or *I-picture*. Once reconstructed, an I-picture becomes a reference picture available for use in predicting neighboring (*nonintra*) pictures. The example GoP sketched in Figure 16.2 above comprises nine pictures.

Figure 16.3 **An MPEG P-picture** contains elements forward-predicted from a preceding reference picture, which may be an I-picture or a P-picture. Here, a P-picture (3) is predicted from an I-picture (0). Once decoded, that P-picture becomes the predictor for a following P-picture (6).



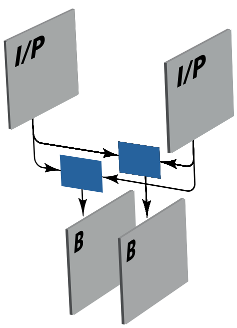
A *P-picture* contains elements that are predicted from the most recent anchor frame. Once a P-picture is reconstructed, it is displayed; in addition, it becomes a new anchor. I-pictures and P-pictures form a two-layer hierarchy. An I-picture and two dependent P-pictures are depicted in Figure 16.3 above.

The term *bidirectional prediction* was historically used, suggesting reverse and forward directions in time. It is often the case that pairs of motion vectors associated with B-pictures point in opposite directions in the image plane. In H.264, the term *bipredictive* is used, and is more accurate in H.264's context.

MPEG provides an optional third hierarchical level whereby *B-pictures* may be interposed between anchor pictures. Elements of a B-picture are typically *bipredicted* by averaging motion-compensated elements from the past reference picture and motion-compensated elements from the future reference picture. (At the encoder's discretion, elements of a B-picture may be unidirectionally forward-interpolated from the preceding reference, or unidirectionally backward-predicted from the following reference.) Each B-picture is reconstructed, displayed, then discarded: No decoded B-picture forms the basis for any prediction. Using B-pictures delivers a substantial gain in compression efficiency compared to encoding with just I- and P-pictures, but incurs a coding latency.

Figure 16.4 below depicts two B-pictures.

Figure 16.4 **An MPEG B-picture** is generally predicted from the average of the preceding reference picture and the following ("future") reference picture. (At the encoder's option, a B-picture may be unidirectionally forward-predicted from the preceding reference, or unidirectionally backward-predicted from the following reference.)



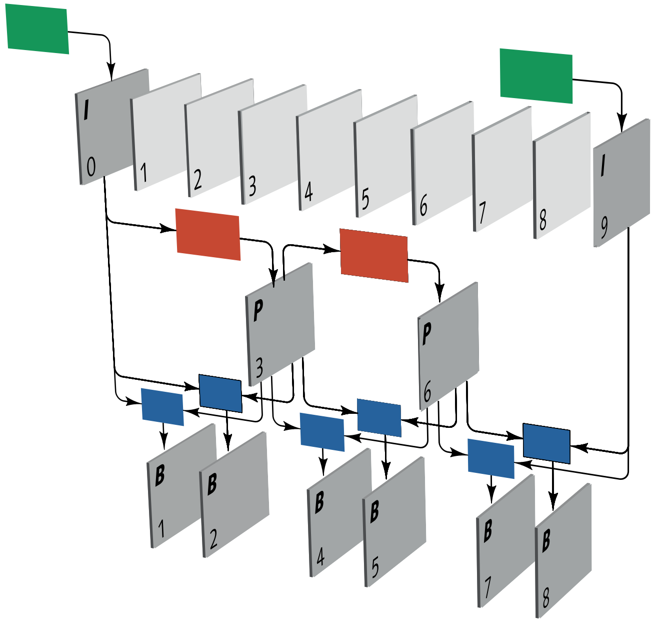


Figure 16.5 **The three-level MPEG picture hierarchy.** This sketch shows a regular GoP structure with an I-picture interval of N=9, and a reference picture interval of M=3. This example represents a simple encoder that emits a fixed schedule of I-, B-, and P-pictures; this structure can be described as IBBPBBPBB. The example shows an *open GoP:* B-pictures following the GoP's last P-picture are permitted to use backward prediction from the I-picture of the following GoP. Such prediction precludes editing of the bitstream between GoPs. A *closed GoP* permits no such prediction, so the bitstream can be edited between GoPs. Closed GoPs lose some efficiency.

The three-level MPEG picture hierarchy is summarized in Figure 16.5 above; this example has the structure IBBPBBPBB.

A 15-frame "long" GoP structured IBBPBBPBBPBBPBB is common for broadcasting. A sophisticated encoder may produce irregular GoP patterns.

A simple encoder typically produces a bitstream having a fixed schedule of I-, P-, and B-pictures. A typical GoP structure is denoted IBBPBBPBBPBBPBB. At 30 pictures per second, there are two such GoPs per second. Periodic GoP structure can be described by a pair of integers N and M; N is the number of pictures from one I-picture (inclusive) to the next (exclusive), and M is the number of pictures from one anchor picture (inclusive) to the next (exclusive). If $M = 1$, there are no B-pictures. Figure 16.5 shows a regular GoP structure with an I-picture interval of $N = 9$ and an anchor-picture interval of $M = 3$. The $M = 3$ component indicates two B-pictures between anchor pictures. Rarely do more than 2 B-pictures intervene between reference pictures.

Coded B-pictures in a GoP depend upon P- and I-pictures; coded P-pictures depend upon earlier P-pictures and I-pictures. Owing to these interdependencies, an MPEG sequence cannot be edited, except at GoP boundaries, unless the sequence is decoded, edited, and subsequently reencoded. MPEG is very suitable for distribution, but owing to its inability to be edited without impairment at arbitrary points, MPEG is generally unsuitable for production. In the specialization of MPEG-2 called *I-frame only MPEG-2*, every GoP is a single I-frame. This is conceptually equivalent to Motion-JPEG, but has the great benefit of an international standard. (Another variant of MPEG-2, the *simple profile,* has no B-pictures.)

I have introduced MPEG as if all elements of every P-picture and all elements of every B-picture are coded similarly. But even a picture that is generally well predicted by the past reference picture may have a few regions that cannot effectively be predicted. In MPEG, the image is tiled into *macroblocks* of 16×16 luma samples, and the encoder is given the option to code *any* particular macroblock in *intra* mode – that is, independently of any prediction. A compact code signals that a macroblock should be *skipped,* in which case the motion-compensated prediction is used without modification. In a B-picture, the encoder can decide on a macroblock-by-macroblock basis to code using forward prediction, backward prediction, or biprediction. Formally, an I-picture contains only I-macroblocks; a P-picture has at least one P-macroblock, and a B-picture has at least one B-macroblock.

Reordering

In a sequence without B-pictures, I- and P-pictures are encoded then stored or transmitted in the obvious order. However, when B-pictures are used, the decoder typically needs to access the past anchor picture and the future anchor picture to reconstruct a B-picture.

Figure 16.6 **Example GoP**

$\mathbf{I_0}B_1B_2\mathbf{P_3}B_4B_5\mathbf{P_6}B_7B_8$

Figure 16.7 **Example 9-picture GoP without B-pictures**

$\mathbf{I_0P_1P_2P_3P_4P_5P_6P_7P_8}$

Consider an encoder about to compress the sequence in Figure 16.6 (where anchor pictures I_0, P_3, and P_6 are written in boldface). The coded B_1 and B_2 pictures may be backward predicted from P_3, so the encoder must buffer the uncompressed B_1 and B_2 pictures until P_3 is coded: Only when coding of P_3 is complete can coding of B_1 start. Using B-pictures incurs a penalty in encoding delay. (If the sequence were coded without B-pictures, as depicted in Figure 16.7, transmission of the coded information for P_1 would not be subject to this two-picture delay.) Coding delay (latency) can make MPEG with B-pictures unsuitable for realtime two-way applications such as teleconferencing.

If the coded 9-picture GoP of Figure 16.6 were transmitted in the order shown, then the decoder would have to hold the coded B_1 and B_2 data in a buffer while receiving and decoding P_3; only when decoding of P_3 was complete could decoding of B_1 start. The encoder must buffer the B_1 and B_2 pictures no matter what; however, to minimize buffer memory at the decoder, MPEG-2 specifies that coded B-picture information is transmitted after the coded reference picture.

Figure 16.8 **GoP reordered for transmission**

$I_0 P_3 B_1 B_2 P_6 B_4 B_5 (I_9) B_7 B_8$

Figure 16.8 indicates picture data as reordered for transmission. I have placed I_9 in parentheses because it belongs to the next GoP (the GoP header precedes it). Here, B_7 and B_8 follow the GoP header.

MPEG-1

ISO/IEC 11172-1, *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 1: Systems* [MPEG-1].

The original MPEG effort resulted in a standard now called MPEG-1, which was deployed in multimedia applications. MPEG-1 was optimized for the coding of progressive 352×240 images at 30 frames per second (240*p*30). MPEG-1 has no provision for interlace. When 480*i*29.97 or 576*i*25 video is coded with MPEG-1 at typical data rates, the first field of each frame is coded as if it were progressive, and the second field is dropped. At its intended data rate of about 1.5 Mb/s, MPEG-1 delivers VHS-quality images.

For broadcast, MPEG-1 has been superseded by MPEG-2. An MPEG-2 decoder must decode MPEG-1 *constrained-parameter bitstream* (CPB) sequences – to be discussed in the caption to Table 47.1, on page 515 – so I will not discuss MPEG-1 further.

MPEG-2

The MPEG-2 effort was initiated to extend MPEG-1 to interlaced scanning, to larger pictures, and to data rates much higher than 1.5 Mb/s. MPEG-2 is standardized in a series of documents from ISO/IEC; MPEG-2 is widely deployed for the distribution of digital television (DTV) including SD and HD (for example, ATSC), and is the (only) video compression scheme for DVD.

Many MPEG terms – such as *frame, picture,* and *macroblock* – can refer to elements of the source video, to the corresponding elements in the coded bitstream, or to the corresponding elements in the reconstructed video. It is generally clear from context which is meant.

MPEG-2 accommodates both progressive and interlaced material. A video frame can be coded directly as a *frame-structured picture*. Alternatively, a video frame (typically originated from an interlaced source) may be coded as a pair of *field-structured pictures* – a top-field picture and a bottom-field picture. The two fields are time-offset by half the frame time, and are intended for interlaced display. Field pictures always come in pairs having opposite parity (top/bottom or bottom/top). Both pictures in a field pair have the same picture coding type (I, P, or B), except that an I-field may be followed by a P-field (in which case the pair effectively serves as an I-frame).

The MPEG IMX variant of MPEG-2, for studio use, is described below. The HDV variant of MPEG-2, for consumer use, is described on page 161. MPEG-2 video compression is detailed starting on page 513.

Other MPEGs

While the MPEG-2 work was underway, an MPEG-3 effort was launched to address HD. The MPEG-3 committee concluded early on that MPEG-2, at high data rate, would accommodate HD; consequently, the MPEG-3 effort was abandoned. I'll discuss MPEG-4 below. MPEG numbers above 4 are capricious.

ISO/IEC 15938-1, *Information technology – Multimedia content description interface – Part 1: Systems.*

MPEG-7, titled *Multimedia Content Description Interface,* standardizes a description of various types of multimedia information (metadata). In my view, MPEG-7 is not relevant to handling studio- or distribution-quality video signals.

ISO/IEC 21000, *Multimedia framework (MPEG-21).*

According to ISO, MPEG-21 "defines an open framework for multimedia delivery and consumption, with both the content creator and content consumer as focal points. The vision for MPEG-21 is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities." In my view, MPEG-21 is not relevant to handling studio- or distribution-quality video.

MPEG IMX

MPEG IMX is a Sony trademark; IMX is not an MPEG designation.

Sony's original Digital Betacam for 480*i* and 576*i* SD used proprietary motion-JPEG-like compression. The first products were videotape recorders denoted

Betacam SX, having a data rate of about 18 Mb/s. Follow-on products adopted I-frame-only MPEG-2 422P@ML, denoted MPEG IMX, having data rates of 30 Mb/s, 40 Mb/s, or 50 Mb/s (IMX30, IMX40, IMX50). MPEG IMX videotape recorders were commercialized. Today it is common to place or wrap IMX compressed video in an MXF file.

XDCAM is Sony's designation for a line of products using a variety of compression systems and a variety of physical media. MPEG IMX compression is one of the compression systems available in the XDCAM line. Recording on optical disc media is possible.

MPEG-4

ISO/IEC 14496, *Information technology – Coding of audio-visual objects.*

The original goal of the MPEG-4 effort was video coding at very low bit rates. The video compression system that resulted is standardized as MPEG-4 Part 2 *Visual; it differes from MPEG-2 and from H.264.* ISO/IEC co-published the ITU-T H.264 standard as MPEG-4 Part 10, so the term *MPEG-4* alone is ambiguous.

MPEG-4 Part 2 defines the *advanced simple profile* (ASP), implemented by DivX and Xvid. ASP is not useful for professional quality video. Even in ASP's intended application domain, low bit-rate video, H.264 – to be described in a moment – has proven to have better performance. Consequently, ASP has fallen out of favour.

MPEG-4 Part 2 also defines a profile called *Simple Studio Profile* (SStP). This profile is used in Sony's HDCAM SR, at very high bit rates (the other end of the bit rate spectrum for which MPEG-4 was conceived). HDCAM SR is widely used in HD, both on tape and in files. Apart from HDCAM SR, the simple studio profile of MPEG-4 sees very limited use.

ISO/IEC 14496-12:2004, *Information technology – Coding of audio-visual objects – Part 12: ISO base media file format.*

Part 12 of the MPEG-4 suite of standards defines the *ISO Base Media File Format,* which defines a general container structure for time-based media files. The format is used in desktop video (most commonly with MPEG-4 Part 2/ASP video, in *mp4* files), but is rarely if ever used in professional video distribution.

H.264

ISO/IEC 14496-10:2009, *Coding of audio-visual objects – Part 10: Advanced Video Coding* [AVC].

ITU-T Rec. H.264, *Advanced video coding for generic audiovisual services*.

An effort to extend MPEG-2 coding was undertaken by the *Joint Video Team* (JVT). During development, this effort worked toward *advanced video coding* (AVC). The resulting standard is promulgated jointly by by ITU-T (who call it H.264) and by ISO/IEC (who call it MPEG-4 Part 10, despite its having little to do with the rest of MPEG-4). H.264 encoding is roughly 1.5 or 2 times as efficient as MPEG-2 – that is, H.264 typically allows encoding at somewhat more than half the bit rate of MPEG-2 for similar picture quality levels. The Blu-ray standard mandates inclusion of H.264 decoding in consumer players. Details are found in *H.264 video compression*, on page 537.

AVC-Intra

An I-frame-only specialization of H.264 has been introduced for professional video acquisition and production, called *AVC-Intra.* Ten-bit vIdeo in 720*p*, 1080*i*, or 1080*p*24 format is compressed to data rates of 50 Mb/s or 100 Mb/s.

In *AVC-Intra 50*, 1920×1080 images are subsampled to 1440×1080, 4:2:0 prior to compression; H.264 Hi10P Intra profile at level 4 is used. 1280×720 images are subsampled to 960×720; 4:2:0; H.264 level 3.2 is used.

AVC-Intra 100 codes 1280×720 and 1920×1080 natively, uses 4:2:2 chroma subsampling, and codes to H.264 Hi422P Intra profile at level 4.1.

WM9, WM10, VC-1 codecs

SMPTE 421M, *VC-1 Compressed Video Bitstream Format and Decoding Process.*

SMPTE RP 227, *VC-1 Bitstream Transport Encodings.*

SMPTE RP 228, *VC-1 Decoder and Bitstream Conformance.*

Microsoft developed a series of video codecs culminating in Windows Media Video versions 9 and 10 (WMV9 and WMV10). These codecs are conceptually similar to – but differ in detail from – H.264. Microsoft submitted the WMV9 codec to SMPTE, where with slight changes it was standardized as VC-1 (corresponding to Windows Media Video 9 – WMV9 or WM9 – Studio profile).

Microsoft developed a proprietary implementation of an encoder and decoder for WM9 (and its successor, Windows Media Video 10). The VC-1 specification was used to produce a “clean-room” implementation of a decoder whose reference source code is publicly available.

The Blu-ray standard mandates VC-1 decoding capability in consumer players, and hundreds of Blu-ray discs were mastered using VC-1. If you're developing Blu-ray players, you'll need to implement VC-1 decoding. However, Microsoft stopped development of the system. VC-1 is now moribund, and VC-1 will not be discussed further here.

Compression for CE acquisition

Traditional consumer electronics equipment has been hardware-based. Two video compression schemes dominate CE today: the HDV scheme, based upon MPEG-2, and the AVCHD scheme, based upon H.264. However, delivery of video across IP networks for display on PCs now has huge economic importance. In a few moments, I'll discuss compression technologies optimized for IP transport and software-based decoding.

HDV

HDV – high definition video, presumably – refers to a compression system suitable for consumer electronic equipment. Long-GoP interframe MPEG-2 compression is used, with $Y'C_BC_R$ 8-bit 4:2:0 video at several different frame rates.

720*p* HDV (sometimes called *HDV1*) offers frame rates of 25, 30, 50, or 60 Hz, has an image structure of 1280×720 with 4:2:0 chroma subsampling, compresses according to MP@HL, and records an MPEG-2 transport stream (TS) with a short (6-frame) GoP at a data rate of about 19 Mb/s.

1080*i* HDV at 25 Hz or 29.97 Hz (sometimes called *HDV2*) downsamples to 1440×1080 with 4:2:0 chroma subsampling, compresses according to MP@H14, and records a packetized elementary stream (PES) with a long (15-frame) GoP at a data rate of about 25 Mb/s. The luma sample rate is 55.6875 MHz – three-quarters of the studio-standard 74.25 MHz rate.

HDV accommodates MPEG-1 Layer II stereo audio. Consumer camcorders are available using MiniDV cassette media, hard drive media, and flash media.

AVCHD

A specialization of H.264 for consumer use is called *AVCHD.* AVCHD compresses 720*p*, 1080*i*, and 1080*p*24 video using long-GoP H.264 coding, to bit rates between about 6 Mb/s and 18 Mb/s. Dolby Digital audio coding is used. AVCHD has been adapted to semi-professional use for recording on 12 cm DVD-R media, SDHC flash memory cards, or hard disk drives.

Compression for IP transport to consumers

The proliferation of personal computers, notebook computers, handheld devices, and tablets, most having WiFi connection and all able to execute software-based video decompression, has led to software-based codec implementations having performance beyond MPEG-2.

Apple has invested in, and has widely deployed, H.264 codecs. H.264 was outlined earlier in this chapter, and is detailed in Chapter 48 on page 537.

Google prefers WebM, to be outlined below.

There is a handful of proprietary codecs in this application domain, including RealPlayer and Adobe Flash Video. (Recent Flash players include H.264 decoding.)

VP8 ("WebM") codec

www.webmproject.org

In 2010, Google acquired a company called On2 that had, over a decade or more, developed a series of proprietary software-based codecs for video distribution. Google released the VP8 codec as open-source, and introduced the *WebM* system for web (IP-based) distribution of video to consumers. WebM comprises video encoded by the VP8 codec and audio encoded by the *Vorbis* codec, both wrapped in the *Matroska* file wrapper. Google's motivation is to avoid paying royalties – and to avoid having its customers pay royalties – on MPEG-2 or H.264 intellectual property (patent) licences.

Dirac (basic)

Dirac seems to be the BBC's preferred designation for the long-GoP wavelet codec intended for use in video distribution. Here we use the qualifier *basic* to distinguish it from Dirac PRO (see page 151).

Basic Dirac is a long-GoP motion-compensated interframe wavelet codec developmed by the BBC and its partners. It has been widely deployed by the BBC for SD distribution between 2–4 Mb/s and for HD distribution between 15–18 Mb/s.

Streams and files 17

A file is an ordered sequence of bytes explicitly having a start and an end, characterized by storage. A stream is characterized by realtime data transfer of unbounded duration on a unidirectional channel – that is, with no upstream channel for flow control, acknowledgement, or retransmission request. Table 17.1 provides a general summary of the characteristics of files and streams.

A file …	*A stream …*
… has predefined beginning and end	… has indeterminate beginning and end
… usually involves storage media	… usually involves an external data interconnect
… permits “random access” to data	… involves sequential data access, typically starting midstream
… has structure imposed at a high level; data is arranged arbitrarily	… has structure imposed at a low level; data is arranged to minimize buffering
… has no need for embedded delimiters	… contains embedded delimiters by which essence elements can be identified “on the fly”
… transfer usually occurs across a general-purpose network	… transfer usually occurs across a data interconnect
… transfer is usually free-running	… transfer is usually synchronized to a timing reference
… transfer typically has variable data (bit) rate (VBR)	… transfer typically has constant data (bit) rate (CBR)
… transfer data integrity is guaranteed, but data transfer rate isn't (best effort)	… transfer data rate is guaranteed, but data integrity isn't (errors may intrude)
… transfer typically involves upstream communication; transfers are generally acknowledged	… transfer typically has no upstream communication; transfers are generally not acknowledged

Table 17.1 **Files and streams** are compared.

A stream is "live," and suitable for realtime interface. A file is not intrinsically "live." A file may be operated on or exchanged slower than realtime, in realtime, or faster than realtime. A portion of a stream can be recorded as a file, and a file can be streamed across an interface; however, generally, streams are structured for realtime use across interfaces and files are structured for nonrealtime use on storage media. Generally, video interfaces (and videotape recorders) are characterized as streams; video storage is characterized as files.

Historical overview

Video signals were historically conceived as streams. VTRs recorded continuous streams (traditionally omitting the vertical blanking interval, and in DVTRs, omitting the horizontal blanking intervals as well).

A stream interface conveys elements (analogous to historical analog video sync) that permit synchronization on the fly: A receiver can connect to a stream at any time, and begin operation within a fraction of a second. Stream formats are designed to have a property of locality whereby essence elements to be presented simultaneously – typically, video and the associated audio – are located nearby in the stream, so as to bound the required buffer storage capacity, and to bound latency to access the essence required for presentation.

Historically, uncompressed digital video was streamed in the studio in realtime across SDI or HD-SDI interfaces. SDI and HD-SDI timing was designed such that analog video could be obtained simply by stripping off the stream synchronization and ancillary elements, performing digital-to-analog conversion, and inserting analog sync. Almost no buffer storage was required.

As digital recording and playback of compressed digital video became possible, the SDTI specialization of SDI was designed to "wrap" compressed video, then audio, for conveyance across SDI. Various compression schemes such as MPEG IMX and DV100 were accommodated. DV video can be conveyed in streaming mode across IEEE 1394 interfaces.

As computing and networking technology advanced, it became feasible to store compressed video, then uncompressed video, in files. It became common to exchange these files across TCP/IP Ethernet networks – first at 100 Mb/s, then 1 Gb/s, and soon, 10 Gb/s.

As commodity IT networking technology improved, the schemes that were used to package compressed video for transport across specialized interfaces such as SDTI were adapted to general-purpose networking. Compressed video in formats such as MPEG IMX and DV100 were stored in files as raw bytestreams. The MXF file format emerged as a mechanism to store video and audio essence in a more structured manner. Standards emerged to store compressed video in MXF files. An MXF file need not contain essence: It can refer to essence stored in separate files. Some formats for A/V storage in MXF files are "stream-friendly" in the sense that video and audio essence is stored in the MXF file in proximity to each other, suitable for playout with a minimum of restructuring. Other formats have higher-level structure more suitable to postproduction (for example, storing video and audio in separate files referred to by the main MXF file).

The Apple iTunes model differs: Files are transferred for later playback.

Today, file-based workflows are widely used in production and postproduction; however, stream-based techniques continue to dominate distribution of professionally produced content. Services such as YouTube, Hulu, and Netflix distribute video to consumers across what I call the big wooly internet – however, service and quality levels of these systems are lower than those associated with television broadcasting: The pictures aren't at HD quality level. They stutter, and the audio loses sync. *Internet-protocol television* (IPTV) refers to adaptations of commodity TCP/IP-based networking to achieve the service and quality levels of broadcasting.

In professional video distribution, the file-based production/postproduction world and the stream-based distribution world meet at the playout server. The playout server includes a disk store with an associated file system. On the production side, the server is accessed asynchronously using IT networking. On the distribution side, a stream access mechanism reads files according to a timeline driven by house sync and timecode, and throttles playback accordingly. Realtime decompression of compressed video may be required. Dedicated stream interfaces then launch the content into the distribution network.

Physical layer

Serial digital video SDI and HD-SDI interfaces are based upon 10-bit words that are serialized, "scrambled," then conveyed unidirectionally as a bitstream onto a single wire. The scrambling technique permits payload data transfer rate to equal the bit rate on the wire; however, signalling sync requires certain data values to be prohibited from appearing in video.

Commodity computer interfaces are based upon 8-bit bytes. Historically, data was serialized onto a single conductor (e.g., Ethernet); however, a number of computer-oriented interfaces (e.g., PCIe and Thunderbolt) serialize data onto multiple "lanes." In some physical interfaces, data is typically mapped – for example, using the 8b/10b scheme – so that all 8-bit byte values can be conveyed across the interface, while allowing the receiver to recover the clock and the data framing for arbitrary data. The bit rate in the channel of such encodings is somewhat higher than the payload rate – in the case of 8b/10b mapping, 1.25 times higher. Other interfaces (e.g., DVI and HDMI) use a dedicated clock wire to establish timing; arbitrary data can then be serialized and transferred without data value restrictions. Some interfaces reverse the direction of data transfer across each conductor (or pair); others have dedicated wires (or pairs) in each direction.

Stream interfaces

SDI, HD-SDI

SDI was designed for uncompressed 4:2:2 SD; it has a data rate of 270 Mb/s. (SDI was adapted to SDTI for compressed video; however, SDTI is now largely obsolete.) HD-SDI was designed for uncompressed 4:2:2 HD; it has a data rate of about 1.5 Gb/s. Recently, a 3 Mb/s adaptation of HD-SDI has been standardized and commercialized. Details are found in SDI and HD-SDI interfaces, on page 429.

DVI, HDMI, and DisplayPort

DVI, *HDMI*, and *DisplayPort* are digital interfaces designed for connection of computer graphics subsystems to displays, across cables at lengths up to 3 m. Apart from a very low-rate reverse channel – *display data channel*, DDC – that communicates display characteristics upstream to the graphics subsystem, DVI and HDMI are unidirectional.

Thunderbolt

The Thunderbolt logo depicts lightning, not thunder! The trademark is registered to Intel.

Thunderbolt refers to a bidirectional interface designed by Intel (apparently in collaboration with Apple). The scheme is intended to connect a computer and its peripheral devices; it achieves a data rate of about 10 Gb/s in each direction, and drives cables up to 3 m. Thunderbolt links can be daisy-chained. DisplayPort predates Thunderbolt; nevertheless, DisplayPort (with a Mini DisplayPort connector) is a specialization of Thunderbolt in which – apart from a very low-speed reverse channel to implement DDC – the data flow is unidirectional. A Thunderbolt link is capable of carrying two uncompressed 1080*p*60 $R'G'B'$ signal sets. Data flow across Thunderbolt is organized into packets; data exchange is based upon the protocols of PCI Express (PCIe), a commodity computer interface.

MPEG-2

Part 2, *Systems,* of the MPEG-2 specification defines two multiplexing schemes. An MPEG-2 *program stream* is a multiplexing scheme appropriate for one program stored on or conveyed across relatively error-free media; an MPEG-2 *transport stream* is a multiplexing scheme appropriate for one or more programs stored on or conveyed across relatively error-prone media. Both of these schemes are outlined in MPEG-2 storage and transport, on page 555. They are generally appropriate for distribution, but not for acquisition or production. (An exception is Sony's MPEG IMX scheme for SD, whose I-B frame structure and fairly high data rate enable use in production.)

IEEE 1394 (FireWire, i.LINK)

IEEE 1394, *Standard for a High Performance Serial Bus*.

In 1995, the IEEE standardized a general-purpose high-speed serial bus capable of connecting up to 63 devices in a tree-shaped network through point-to-point connections. The link conveys data across two shielded twisted pairs (STP), and operates at 100 Mb/s, 400 Mb/s, or 800 Mb/s. Each point-to-point segment is limited to 4.5 m; there is a limit of 72 m across the breadth of a network. Asynchronous and isochronous modes are provided; the latter accommodates realtime traffic. Apple computer refers to the interface by their trademark *FireWire*. Sony's trademark is *i.LINK*.

Though not practical for uncompressed video, IEEE 1394 has performance and features that make it highly practical for compressed video streams, particularly DV.

The standard 6-pin connector provides power for peripheral devices. Sony commonly uses a 4-pin connector not compliant with the IEEE standard. A node may have either 4-pin or 6-pin connectors. Power is absent from the 4-pin connector; many people find the 4-pin connector to be mechanically flimsy.

IEC 61883-1, *Consumer audio/video equipment – Digital interface – Part 1: General.* See also parts 2 through 5.

IEC has standardized the transmission of digital video over IEEE 1394. Video is digitized according to BT.601, then motion-JPEG coded (using the DV standard) at about 25 Mb/s; this is colloquially known as 1394/DV25 (or DV25-over-1394). DV coding has been adapted to 100 Mb/s for HD (DV100); a standard for DV100-over-1394 has been adopted by IEC.

HTTP live streaming (HLS)

http: hypertext transfer protocol

Apple has introduced a streaming adaptation of the *http* protocol that was devised for the World Wide Web. The scheme has aspects of stream transfer and aspects of file transfer.

The playlist is an Extended M3U Playlist file, either an .m3u file encoded in US-ASCII or an .m3u8 file encoded in UTF-8.

The server segments programs into a sequence of media files each having short playback duration, typically a small integer number of seconds, 10 or so. Typically, a segment is represented by several media files containing the same program material coded at different bit rates. Each media file is an MPEG-2 transport stream including a program association table (PAT) and a program mapping table (PMT); these alllow stream decoding to start at any segment without requiring access to any earlier segment.

gzip: Gnu zip, a lossless data compression technique

The server also prepares a playlist (index) file that associates time offsets to a set of filenames (actually, URLs) of media files along with their associated bit rates and sequence numbers. The playlist file can be gzip-compressed.

The client establishes the bit rate that it expects to be sustained, then accesses the playlist file to establish the filenames of the content at the desired time and the desired bit rate. The client issues an http request for the named file, and plays it.

There is no predetermined bound on program length: The server is free to add additional media files, and (atomically) update the playlist file accordingly. To continue playing, at some suitable time before that file has played out (making provision for estimated latencies), a request is issued for the next file in the playlist. The client can dynamically switch between different bit rates as required by network performance. Random access and "trick" mode access is permitted.

Audio-only files are accommodated. Provisions are made for encryption.

The scheme is called *live* streaming. For a truly live "broadcast," the playlist file can contain as few as three segments, with clients accessing segments according to their time of joining and their data rate requirements. The playlist file is updated every few seconds, appending and removing media files as necessary. There is a certain latency in access – typically several seconds, much longer than the typical latency of a live television broadcast using MPEG-2. There's no requirement for content to be "live": All of the segments of a show can be present in the playlist file; a playlist can be appended to as a program proceeds, with older segments being retained.

All data – both playlist files and media files – is transferred using normal web protocols on port 80: The upstream http GET request for a playlist file looks like a request for a web page, and the downstream transfer of a media file looks like transfer of an admittedly large web page. Router, proxy, and firewall issues are rare.

HTTP live streaming is implemented in Mac OS X and in iPhones.

A comparable scheme called *dynamic adaptive streaming over http* (DASH) is in the process of being standardized by ISO/IEC MPEG.

Metadata 18

This chapter differs in tone from other chapters in this book. I'm a skeptic concerning metadata.

You know you're in trouble when the Wikipedia page for *Metadata* starts "The term metadata is an ambiguous term …" [accessed 2011-10-18].

Metadata presents problems – therefore opportunities, therefore commercial activities, therefore products. However, in my view the video industry hasn't achieved a sufficiently broad understanding of the deep principles of metadata that any general approach can be set out.

Consider an audio file, storing 200 million audio sample pairs at 44.1 kHz representing a performance of Beethoven's Symphony No. 9, *Choral.* To recreate that sound approximating the way it was experienced by the original audience, you'll need to know the sample rate.

The sample rate could be provided in a paper document, perhaps a standard. To enable general purpose decoders and players, it makes sense to encode sample rate in the file, perhaps in the file header.

Is such an encoded sample rate *data* or *metadata?*

I argue that it's *data,* because the intended auditory experience cannot be attained without knowing it.

You may feel that this example – call it Example 0 – is contrived and irrelevant. Let me present five further examples. Example 1 is conceptually a small step from Example 0; we proceed (with increasing complexity and increasing relevance to professional video) to Example 5, which concerns a highly topical issue in video engineering. I claim that Example 5 exhibits the same philosophical dilemma as Example 0:

What's *data,* and what's *metadata?*

While this dilemma persists, a chapter entitled *Metadata* must ask questions instead of providing answers.

Metadata Example 1: CD-DA

CD-DA was defined by the Sony and Philips "Red Book," which IEC subsequently standardized as IEC 60908.

CD-DA abbreviates *compact disc-digital audio.* CD-DA was conceived by Philips and Sony to store hi-fi digital stereo audio at 16 bits per sample and 44.1 kHz sample rate (that is, a data rate of about 1.5 Mb/s) on optical media having capacity of about 660 MB.

After a few years, the CD proponents adopted the *CD Text* standard, augmenting the Red Book to allow recording text-based metadata. But by then it was too late.

The original "Red Book" specification for CD-DA did not include any provision for album title, artist name, song titles, liner notes, or any other text information. This information was printed on the CD jacket; apparently Sony and Philips thought that providing such information in digital form would be redundant! The CD format not only lacked the metadata but also lacked any provisions for a unique ID.

Today some people would call the table of contents *technical metadata.* I consider it to be data: Without the ToC, the user cannot put the system to its intended use – playing songs.

The recorded CD-DA media did – of necessity – include a table of contents giving track count, track start times, and track durations (to $^1/_{75}$ s accuracy). The audiophile and software engineer Ti Kan realized that this information could be "hashed" into a 32-bit number and treated as an ersatz unique ID. As CDs became popular, Kan (assisted by Steve Scherf) created the CDDB service, a database to store community-contributed metadata associated with their codes. CDDB was originally a community-driven service, but became a commercial entity – first CDDB, Inc. (in 1995), then Gracenote (in 2000, acquired by Sony in 2008).

So, CD albums have metadata – but not reliably sourced by, or under direct control of, content creators.

The lesson for the system designer is this: What constitutes "data" and what constitutes "metadata" is coloured by your view of the boundaries of your system. Sony and Philips apparently thought of the CD system as distributing prerecorded digital audio. Today, we think of the CD system as distributing music to consumers. There's a subtle difference that changes the notion of what's data and what's metadata.

When the MP3 audio compression system was created, the developers made provisions for ID3 tags to convey metadata sourced by the content creators.

ITU-R BR.1352-2, *Broadcast Wave Format (BWF).*

The BWF file format commonly used for broadcast audio includes a "parameter" called *nSamplesPerSec* giving the sample rate. The parameter is carried in a "BWF Metadata Chunk." Is the sample rate metadata?

Metadata Example 2: .yuv files

The ".yuv" file format was introduced by Abekas in the late 1980s to store uncompressed video. Given samples of 8-bit $Y'C_BC_R$, 4:2:2 interlaced video in raster order, the file format definition is essentially as follows:

> Store successive image rows, where each row is a sequence of 4-byte elements [C_{B_0}, Y_0', C_{R_0}, Y_1'] where subscript 0 signifies an even-numbered luma sample location and subscript 1 signifies odd.

There is no header in a .yuv file – in particular, there is no provision for storing the count of frames, image rows, or image columns. The format was introduced to store 720×480 video. Later, it was applied to 720×576. It could potentially be applied to 720×481, 720×483, 720×486, or 704×480. It has been used in the codec research community for 1280×720*p* and 1920×1080*i*.

Consider the reading of .yuv files constrained to be 720×480 or 720×576. Most of the time the format can be determined by dividing the file's bytecount by 1440, then dividing by 480 and 576 in turn to see which quotient is an integer. But that approach doesn't always work. For example, a 4,147,200-byte file could be six frames of 480*i* or five frames of 576*i*.

Reliable file interpretation is attained only by agreement between sender and receiver – or expressed more properly in terms of files, between writer and reader – that is, *outside* the scope of transfer of the file itself.

Imagine extending the .yuv file format by prepending a file header comprising three 32-bit words: a count of the number of frames, a count of the number of image rows, and a count of the number of image columns. Is the header data or metadata? If your "system" is defined in advance as being 480*i*, then the counts in the header are inessential, auxiliary information – call it metadata. But if your "system" is multiformat, then the counts are most certainly *data,* because reliable interpretation of the image portion of the file is impossible without the numbers in the header.

The conclusion is this: What comprises "metadata" depends upon what you consider to be your "system." The larger, more inclusive, and more general your system – the less you depend upon context – the more your metadata turns into data.

Metadata Example 3: RFF

See *2-3 pulldown*, on page 405.

Since about 1953, a dominant source of television content has been movies – first on photochemical film, then in digital form. For more than half a century, movies have been intended for display at a frame rate of 24 Hz. The expedient solution to match movie frame rate to the historical 59.94 Hz field rate of North American television is to slow the movie to 23.976 Hz, then impose *2-3 pulldown* whereby successive movie frames are displayed twice, then three times, twice, then three times, and so on. A certain degree of motion stutter results, but is not objectionable to consumers. Certain video frames – *M-frames*, see Figure 34.1 on page 405 – comprise fields from two different movie frames.

In about 1990 it became feasible for consumer television receivers to eliminate the display twitter artifact of interlaced display by deinterlacing (by digital means) and displaying frames at 59.94 Hz. Owing to the prevalence of "film" material, deinterlacing required detection and treatment of the M-frames.

The technique adopted compares elements of the image data of successive video fields to see if a 2-3 pattern can be discerned. If a sustained 2-3 sequence is detected, then the source is presumed to be 24 Hz; frames are assembled accordingly. As CE technology progressed, receivers became more and more dependent upon such algorithms, to the point today that a high-quality digital television processor chip may dedicate a hundred thousand gates to the task. The problem is that implementations aren't necessarily reliable, and different implementations aren't consistent.

The problem arose at a time when broadcasting of "line 21" closed caption data was becoming commonplace, transmitting roughly 16 bits per field. The 2-3 problem could have been nipped in the bud by including one bit per field signalling the film pulldown.

The MPEG-2 system accommodates 24 Hz material through the *repeat first field* (RFF) flag conveyed in the Picture Coding Extension. The flag causes the first decoded field of a field pair to be repeated. MPEG-2's RFF can be considered a metadata "hint": Satisfactory performance is obtained ignoring it, but improved performance is obtained by using it.

Metadata Example 4: JPEG/JFIF

JFIF mandates $Y'C_BC_R$ with BT.601 luma-chroma matrixing; however, 4:2:0 chroma subsampling in JFIF is sited interstitially both horizontally and vertically, unlike BT.601.

At its inception, the JPEG committee decided to avoid colour space wars: Its scope was established as compressing and decompressing image data, without concern for what colours the data represented. They accommodated one-channel greyscale image data, three-channel image data such as *RGB*, and four-channel image data such as *CMYK*. Chroma subsampling was recognized as providing a big compression gain – a 2:1 factor in the case of 4:2:0 – so allowance was made to enable luma-chroma encoding as a preprocessing step.

JPEG development culminated before the World Wide Web emerged – JPEG's original target application was colour facsimile! The founders expected that system integrators would make provisions outside JPEG for reliable colour transfer. However, JPEG was rapidly adopted as a method of *exchanging* colour images in files, not just compressing them.

Hamilton, Eric (1992), *JPEG File Interchange Format*, Version 1.02 (Milpitas, Calif.: C-Cube Microsystems). This informal document was endorsed by ECMA, who made slight modifications and in June 2009 published ECMA TR/98 having the same title.

It became clear to a JPEG proponent, C-Cube, that confusion regarding colour spaces in the *exchange* of JPEG files threatened to inhibit commercialization. C-Cube quickly drafted a document (paper metadata) defining a file format called *JFIF,* stating that image data was to be coded in the $Y'C_BC_R$ colour space of BT.601 (but without the footroom and headroom). JFIF is clear on the arithmetic. However, implementation according to JFIF calls for a C_B/C_R reference range of ±128, that is, 257 integers – but 8-bit coding permits only 256 values! Meeting the specification produces grey having noninteger C_B and C_R values; decoded grey is bound to be coloured. Implementations (particularly the widely used *libjpeg*) use the standard scale factors but clip the C_B/C_R range to +127, thereby clipping pure blue and pure red.

To use JFIF for BT.709 HD $Y'C_BC_R$, to conform to the standard you must recode to ${}^{601}Y'C_BC_R$, compress, transfer, decompress, and finally recode to ${}^{709}Y'C_BC_R$.

JFIF explicitly mandates the BT.601 luma-chroma matrix. BT.601 says nothing of primaries, and JFIF does *not* say what primaries are intended. JFIF was embraced by the computer industry at exactly the time that the sRGB standard was being formulated using BT.709 primaries. In practice, JFIF uses BT.709 primaries.

The JFIF specification states, "RGB components calculated by linear conversion from YCbCr shall not be gamma corrected (gamma = 1.0)." This passage does *not* mean that linear-light components are encoded; instead, *decoding* is intended to conform to BT.601

practice, which (after $Y'C_BC_R$-to-$R'G'B'$ dematrixing) imposes a 2.4-power function on $R'G'B'$ components to produce display tristimulus values.

So, we have the following mess:

- The spec says "BT.601 $Y'C_BC_R$" but contrary to BT.601, and for no good reason, "full-swing" is used.
- The spec says C_B and C_R values range ±128, a range unattainable in 8-bit integer arithmetic. Implementations cope by clipping pure blue and pure red.
- The spec says "gamma = 1.0," but the intention is clearly not linear-light coding. The spec is otherwise silent on "gamma," but a 2.4-power law EOCF is implicit.
- The spec says "BT.601," evidently taking that to define colour primaries, but BT.601 is silent on colour primaries. In practice, the primaries of BT.709 are used.

You may be thinking, "this is just a story about a poorly written specification for paper metadata, and about nonconformant implementations." That assessment is mainly correct. The next example is analagous – but brings the poorly conceived metadata into the professional video data stream.

Metadata Example 5: Sequence display extension

Here I describe aspects of MPEG-2's sequence display extension. Identical metadata is conveyed in H.264's Annex E, *Video usability information* (VUI).

In a manner roughly comparable with MPEG-2's RFF flag, MPEG-2's sequence display extension provides a decoder with information concerning how the image data is intended to be displayed. The standard provides (in printed form, as paper metadata) tables giving RGB primary chromaticities (*color primaries*), transfer functions (*transfer characteristics*), and luma-chroma matrices (*matrix coefficients*). The bitstream conveys enumerated codes that serve as indexes into these tables. The same scheme is adopted in H.264. The tables (simplified, and augmented with my annotations) are summarized in Tables 18.1, 18.2, and 18.3.

DeMarsh, LeRoy E. (1993), "TV display phosphors/primaries: Some history," in *SMPTE J.* **102** (12): 1095–1098.

Color primaries code 4 designates NTSC 1953 primaries. As far as I am aware, no extant recorded video material uses those primaries; they had been abandoned before the introduction of the first VTR. A bitstream containing that value is nonsensical: If the code is encountered, it ought to be ignored by all decoders. (A well-meaning technician may have set the code thinking, "I'm broadcasting NTSC; this is the only setting that says NTSC; I'd better use it.")

Code	Interpretation
0	Forbidden
1	**BT.709**
2	Unspecified
3	Reserved/future
4	BT.470-6/NTSC 1953
5	EBU Tech. 3213
6	SMPTE RP 145
7	SMPTE 240M
8	"Generic film"

Table 18.1 ***Color primaries***. Entries shaded in red are obsolete; the NTSC 1953 entry is utterly obsolete. Codes 5 and 6 are unsuitable for HD. Code 8 "Generic film" (shaded in magenta) is inscrutable. No matter which code you place in the bitstream at encoding, your material will almost certainly be presented with BT.709 primaries (bolded).

Code	Interpretation
0	Forbidden
1	**BT.709**
2	Unspecified
3	Reserved/future
4	Display gamma 2.2†
5	Display gamma 2.8†
6	BT.709
7	SMPTE 240M
8	Linear
9	Log (10^2:1)
10	Log ($10^{2.5}$:1)
11	xvYCC
12	BT.1361
13 … 255	Reserved

Table 18.2 ***Transfer characteristics***. Where MPEG says *display gamma*, read *EOCF;* these entries are flagged†. All other entries define an *OECF.* Entries shaded magenta are impractical. The two codes in green-shaded rows have identical interpretations.

Code	Interpretation
0	Forbidden/GBR
1	**BT.709**
2	Unspecified
3	Reserved/future
4	BT.601
5	BT.601
6	BT.601
7	SMPTE 240M
8	$Y'C_GC_O$
9 … 255	Reserved

Table 18.3 ***Matrix coefficients***. The three enumerations in the green-shaded rows have identical interpretation. The GBR entry, in H.264's VUI, is for coding *R'G'B'* 4:4:4.

MPEG and H.264 have the pervasive conceptual model that only bitstreams and decoders are standardized. A compliant encoder emits only legal bistreams. Apart from that, no aspect of the encoder is standardized. So, *transfer characteristics* should be specified as EOCFs, not OECFs!

Concerning transfer functions (Table 18.2), MPEG-2 fails to distinguish OECF from EOCF; the table contains both. Duplicate codes are provided for BT.709. Code 5, display gamma 2.8, is never used in practice, even in Europe (where other standards mention 2.8). A linear transfer function, code 8, will give unacceptable picture quality when used with fewer than about 14 bits per component. The logarithmic encodings are unworkable: Either of these encodings would clip low-luminance colours in a manner objectionable to consumers.

Concerning the luma-chroma matrix (Table 18.3), the BT.601 setting is triplicated for no good reason.

These tables exemplify what I call the encyclopedic approach to metadata: All the possibilities are collected without regard for practical use cases; no guidance is offered concerning how to encode metadata or how to decode it.

Apple, Inc. (2011), *QuickTime File Format Specification* (July): 141.

Issues with SDE metadata must be handled by software developers. Here's what Apple says concerning the duplicate BT.709 codes in *transfer characteristics:*

> QuickTime writers should map [code] 6 to 1 when converting from *transfer_characteristics* …

MPEG cites the ITU-R document incorrectly: The cited document is a *Report* (Rep.), not a *Recommendation* (Rec.).

The MPEG-2 specification cites ITU-R Rec. BT.470-6. Concerning that reference, Apple writes,

> This information is both incomplete and obsolete.

We could add, "erroneous."

In the light of all this confusion, how should an encoder be configured?

- For SD material, set the primaries to EBU 3213 for 576*i* material and to SMPTE RP 145 for 480*i* material; set *transfer characteristics* to BT.709 and *matrix coefficients* to BT.601.
- For HD, declare BT.709 everywhere in the SDE.

A problem for ATSC encoders is that North American HD material is almost all mastered with SMPTE RP 145 primaries, and you're tempted to declare that; however, ATSC specifications call for BT.709 primaries, and virtually all consumer receivers display with BT.709 primaries. My suggestion is to declare BT.709, for two reasons: to be ATSC compliant, and to prepare for the future when regional primary sets are relics of that past.

What should a decoder do?

- For SD formats, if BT.709 is declared for *color primaries,* it's probably intended and should be respected; otherwise, expect EBU 3213 for 576*i* material and SMPTE RP 145 for 480*i* material. Any other code is nonsensical and should be treated as BT.709. Treat *transfer characteristics* as BT.709 no matter what is declared. Expect *matrix coefficients* to be BT.601; BT.709 could potentially be correct but should be treated with suspicion. Any other code is almost certainly wrong.
- For HD formats, expect BT.709 across the board. Any other codes are highly suspect.

I summarize these recommendations for decoder processing in Tables 18.4, 18.5, and 18.6.

Code	*Interpretation*
0	Forbidden
1	**Respect for HD or SD**
2	Unspecified
3	Reserved/future
4	Suspect (Use code 1)
5	**Respect for SD**
6	**Respect for SD**
7	Suspect (Use code 1)
8	Suspect (Use code 1)

Table 18.4 ***Color primaries* interpretation.**

Code	*Interpretation*
0	Forbidden
1	**Respect**
2	Unspecified
3	Reserved/future
4	Use code 1
5	Use code 1
6	Replace with code 1
7	Suspect (Use code 1)
8	Suspect (Use code 1)
9	Suspect (Use code 1)
10	Suspect (Use code 1)
11	xvYCC: Implement
12	Suspect (Use code 1)
13 … 255	Reserved

Table 18.5 ***Transfer characteristics* interpretation.** Code 11, xvYCC, might be used in future systems, but is highly unlikely to be encountered today.

Code	*Interpretation*
0	Forbidden/GBR
1	**Respect for HD**
2	Unspecified
3	Reserved/future
4	**Respect for SD**
5	**Respect for SD**
6	**Respect for SD**
7	Suspect (Use code 1)
8	$Y'C_GC_O$: Implement
9 … 255	Reserved

Table 18.6 ***Matrix coefficients* interpretation.** Code 8 ($Y'C_GC_O$) might be used in future systems. I am unaware of any decoder today that will decode $Y'C_GC_O$ colour space, and no functional benefit is evident. It is highly unlikely to be encountered.

Conclusions

In the studio, MXF enables reliable coding, storage, and transport of metadata in files. ANC packets can convey metadata across HD-SDI streams; however, there are no common, widely used practices concerning how metadata is handled alongside video passed through processing equipment.

Apart from MXF files and HD-SDI ANC metadata, studio equipment has no reliable metadata system.

Migration to file-based workflows enables reliable metadata processing. However, the conceptual problems of metadata (evident in my examples 0 through 5 above) remain. I suggest these guidelines for implementing metadata systems:

- Devise metadata carefully. Have clear rules for what metadata to encode, and how to interpret metadata at decoding time. Don't be encyclopedic: “Make the system as simple as possible, but no simpler” [Einstein].

- Metadata should be inessential, otherwise it would be data: Design your system to work *without* metadata, then add metadata as augmentation.
- Strive for a design such that adding metadata never impairs the operation of the base system.
- If information is already available in data, don't duplicate it in metadata: To do so opens the possibility (at a later date) of conflicting information.
- If in a particular system design it is possible for essence to be separated from its metadata, the format chosen for essence must be able to serve its purpose in the absence of metadata.

Stereoscopic ("3-D") video 19

The term *S3D* ("ess-three-dee") distinguishes stereoscopic 3-D from imagery having depth cues (particularly, perspective) but only one view. Computer-generated imagery (CGI) produces images synthesized from scene geometry; CGI can relatively easily produce stereo views. Some people consider the term *S3D* to be redundant – that which is stereoscopic is necessarily 3D.

Stereoscopic 3-D (S3D) refers to acquisition, processing, storage, distribution, and display of imagery in two views, one intended for the left eye and one for the right. The views are typically acquired from cameras acquiring the same scene from positions a short lateral distance apart. Stereo viewing presents an illusion. Unlike viewing the real world, the views do not change when the viewer moves his or her head. Nonetheless, for very carefully crafted material, the effect can be convincing, and in some cases, can add to storytelling.

Acquisition

Two cameras are most often used; however, many other arrangements have been demonstrated such as one lens and two imagers, and two lenses and one imager.

To acquire images from a real scene in professional content creation, two cameras are typically used, each including an imager and signal processing. To produce "normal" stereo the optical axes of the cameras are displaced by the same distance the typical viewer's eyes are separated – the *interocular distance* (also known as *interpupillary distance*), which for adults is between about 52 mm and 75 mm, with a mean of about 63.5 mm (2.5 in). Various effects can be achieved by changing the interaxial distance of the cameras: setting a wide camera interaxial distance collapses depth, and upon display makes the scene look smaller than it is; setting a narrow camera interaxial distance expands depth and upon viewing magnifies the scene. Misaligned cameras can lead to viewer discomfort.

S3D display

S3D display can be achieved with a dedicated display for each eye, in the manner of the historical View-

Master. Many virtual reality systems from the 1990s and 2000s used the technique, sometimes in combination with head tracking; however, consumers are not comfortable with head-mounted display equipment! Viewing at a distance is a commercial necessity.

For normal television viewing distance of about 3 m, several schemes are in use that multiplex the two views at the display device and separate the views at each viewers' pair of eyes: anaglyph, temporal multiplexing, polarization, wavelength multiplexing, parallax barrier autostereoscopy, and lenticular autostereoscopy. These techniques are outlined in the sections to follow.

The techniques to be described are almost always used with a single "native" 2-D display (either direct view, or projector). In this case, all of the techniques have the disadvantage that at best 50% of the light of the native 2-D display is available (and frequently, much less). Consequently, stereo 3-D display systems tend to be dim.

Anaglyph

Associating red with left conforms to the nautical convention that red signifies the port (left) side.

Imagery is created placing the red component of the left view into the red primary, and the green and blue components of the right view, into the three components of what would otherwise be a 2-D video stream. (Clearly, several assumptions that enable chroma subsampling and MPEG or H.264 encoding are broken.)

The display presents the left-eye image using the red primary and the right-eye image using green and blue.

The viewer wears glasses having a colour filter over each eye. A red filter is placed over the left eye – the left eye only sees the red primary of the signal, containing the left image. A cyan filter is placed over the right eye – the right eye sees dichromatic combinations of the green and blue components of the right image. Full colour is not present for every pixel for each eye; nonetheless, the viewer's visual system largely compensates the loss (albeit with some discomfort). The red/cyan scheme is most common, but anaglyph display can use other combinations of colours.

Owing to the ease of recording and transmission using standard 2-D video infrastructure (admittedly outside of its usual assumptions), the anaglyph scheme was used sporadically for years in both cinema and television, but has mostly fallen into disuse and is now generally considered a novelty.

Temporal multiplexing

Two views can be multiplexed in time: The display operates at (at least) twice the frame rate of the imagery and alternately presents the left-eye image and the right.

The viewer wears active shuttered glasses, synchronized with the display such that the right eye is blocked while the left image is displayed and the left eye is blocked while the right image is displayed.

Shutter synchronization is typically achieved through an infrared (IR) light beam that is pulsed at the frame rate, flooding the viewing area. Each set of glasses includes an IR receiver. (Bluetooth radio frequency synchronization has been proposed.)

The scheme dominates 3-D consumer television, and has limited use in cinema (XPAND 3D).

Polarization

An excellent outline of the physics of polarized light is given in this book: Reinhard, Erik et al. (2008), *Color Imaging: Fundamentals and Applications* (Wellesley, Mass.: A K Peters).

Many S3D display schemes involve polarized light. The simplest forms of polarization – those used commercially – are linear polarization (LP) and circular polarization (CP). The viewer wears passive polarized glasses; filters for two eyes have opposite polarizations.

Polarization can be time-multiplexed: The display operates at (at least) twice the frame rate of the imagery, and alternately presents the left-eye image (in one polarization) and the right-eye image (in the opposite polarization).

In the RealD system common in theatres, a "Z screen" is inserted in the light path at the projector, between the projection lens and the port glass. The Z screen is an electro-optical device that rapidly switches the polarity of circular polarization. The imager produces the left- and right-eye images time-sequentially; the Z screen is actuated in synchrony. (In the RealD system deployed in theatres as I write, there are three left-right cycles per $^1/_{24}$ s – that is, the display's modulator produces images at 144 Hz.) The technique has not been commercialized for direct-view displays.

Polarized projection can potentially produce both views at the same time – for example, by using a pair of projectors (or two image modulators sharing the same projection lens). However, such solutions are unpopular owing to their high cost. A single 4 K (4096×2160) projector can be adapted to display a 2 K (2048×1060) left image on the top and a like-sized right image on the bottom, then fitted with an optical device to oppositely polarize the two images and combine them for simultaneous display. The scheme has been commercialized for cinema by Sony.

In the system as commercialized, each image has 1060 rows, not 1080 as you might expect: 40 black rows lie between the two.

Polarized projection requires that the screen preserve polarization. Typical cinema screens depolarize, so "silver" – actually, aluminized – screens are used.

Opposite polarization of alternate image rows is typically achieved using a *film pattern retarder* (FPR).

For direct-view displays, polarization can be accomplished by fabricating polarizers of opposite polarity over alternate image rows of the display. Obviously, in 3-D operation, vertical resolution is halved compared to the native display capability. Such a display can be used for normal 2-D viewing without glasses (though with at best 50% of the 2-D light available).

A big advantage of polarized systems is the fact that the glasses are passive and inexpensive.

Wavelength multiplexing (Infitec/Dolby)

This technique was invented by Helmut Jorke at Daimler-Benz in Germany. The display operates at twice the frame rate of the imagery (or higher), and presents first the left-eye image, then the right, through different optical filters. The wavelength compositions of each pair (e.g., G_{LEFT} and G_{RIGHT}) are designed to be mostly nonoverlapping. The characteristics of the optical filters are compensated by signal processing to produce roughly metameric pairs – that is, although the wavelength composition of the pair of reds differ, the colours look roughly the same.

The viewer wears passive glasses, where each eye has a different optical filter roughly matching that of the projector. The left eye's filter rejects the wavelengths corresponding to R_{RIGHT}, G_{RIGHT}, and B_{RIGHT}; the right eye's filter rejects the wavelengths corresponding to R_{LEFT}, G_{LEFT}, and B_{LEFT}.

The wavelength multiplex scheme could simultaneously present left and right images. However, that mode hasn't been commercialized.

The Infitec scheme uses passive (albeit somewhat expensive) glasses, and does not require a polarization-preserving screen.

Dolby commercialized the scheme for 3-D cinema. It has not been commercialized for direct-view displays.

Autostereoscopic displays

Autostereoscopy refers to techniques that present stereoscopic imagery without the requirement for the viewer to wear glasses. Two techniques have received limited commercialization: the parallax barrier technique, and the lenticular technique.

Autostereoscopic displays typically create reasonable stereo across a small volume of the viewing space. The major problem is that the "sweet spot" is typically fairly small, and outside the sweet spot, the stereo effect is either dramatically reduced or vanishes entirely. Also, autostereoscopic displays sometimes have (unintended) viewing positions where the views are reversed, causing apparent depth inversion known as *pseudostereo.*

Parallax barrier display

Two views are displayed interleaved column-by-column on the same display surface. A short distance in front of the display lies a set of barriers that form slots through which, at normal viewing distance, alternate image columns can be viewed. The geometry of the barrier (pitch and position) is designed so that at a chosen optimal viewing location, one set of columns is visible to the left eye and the other is "shadowed" by the barrier; the situation is reversed for the right eye.

The technique has been commercialized in handheld devices (3-D cameras and cellphones).

Lenticular display

Two or more (N) views are interleaved on the display surface in N columns. A set of lenses is placed, one lenslet per N columns, over the display. The geometry of each lens is arranged to project the interleaved columns out into the space in front of the display. In the case of two views ($N = 2$), the left and right images lie in alternate beams.

Philips has demonstrated lenticular autostereoscopic display where several views ($N \approx 9$) are generated at the display by signal processing based upon a single 2-D image accompanied by a "depth map" (2-D + Z) that is encoded during postproduction or produced in graphics generation (for example, in PC gaming). The technique has had limited deployment in digital signage, but has not been commercialized for consumer use.

A depth map can fairly easily be created for CGI content, including computer games in consumers' premises. However, there is no widely available standard for conveying the depth map from the computer to the display. Depth map techniques do not directly deal with occlusion, so visual performance is limited.

Recording and compression

For a given image format (e.g., 1920×1080), S3D obtained through a pair of views obviously involves double the data rate of a single view. The challenges in transport and interface centre around the high data rate. Professional acquisition and postproduction usually involves doubling the data rate (and often doubling up the production equipment). For consumer recording and distribution, 3-D systems have been devised that use less than twice the data rate of 2-D imagery.

Many techniques have been devised to record S3D content and to transport S3D content through broadcasting distribution chains. Some distribution networks squeeze the left and right views 2:1 and abut them horizontally side-by-side (*SbS*) onto a single signal that can be conveyed through ordinary distribution networks.

The stereo high profile is related to the *multiview profile* (MVP) of H.264: Both are documented in Appendix H of the current revision.

The Blu-ray standard has been augmented with a mechanism to compress S3D content using the *stereo high profile* of H.264. The motion estimation and motion-compensated interpolation schemes of H.264 were devised to compactly code a sequence of images having a high degree of spatial correlation, where differences between the images are a consequence of elapsed time between their exposures. The left and right images of a stereo pair exhibit a high degree of spatial correlation, where differences between the images are a consequence of position shifts (*disparity*) induced by parallax. In typical SHP use, the right image is predicted by the left image after "motion" compensation by *disparity vectors* (comparable to motion vectors). Typical stereo can be coded at between about 1.3 or 1.6 times the data rate of 2-D imagery.

Some people might quote multiples as low as 1.2 or as high as 1.8.

Consumer interface and display

Previous sections have discussed acquisition and display of S3D imagery. Here, we'll discuss interface to the consumer display.

Dave LeHoty describes his home HDMI system as "1.3a with a steenkin' asterisk," alluding to the wide variety of versions and options that makes system integration difficult for the expert, let alone for the average consumer.

HDMI version 1.4a has a mandatory *frame packing 3D* structure that packs left and right eye 1920×1080 images into a 1920×2205 "container" having 45 blanking (black) lines separating the images. There are progressive and interlaced versions.

HDMI 1.4a also describes an interface using a 1920×1080 container to convey a 960×1080 left eye image and a 960×1080 right eye image, both horizontally squeezed 2:1, abutted side-by-side (*SbS*). Horizontal resolution suffers.

Finally, HDMI 1.4a describes an interface using a 1920×1080 container to convey a 1920×540 left eye image and a 1920×540 right eye image, both vertically squeezed 2:1, abutted top-and-bottom (*TaB*). Vertical resolution suffers.

There are many schemes. Confusion abounds.

I once visited a consumer electronics retailer where a stereoscopic 3-D movie was being played from a Blu-ray disc and conveyed across HDMI in side-by-side format – but displayed on a receiver whose 3-D processing was disabled. I adjusted my vergence to free-view the 3-D imagery, horizontally squished 2:1. A salesman approached and said, "That's not 3-D." I said, "Well, I'm seeing depth." Without missing a beat, and with full confidence, he said, "No, you're not."

Ghosting

Most of the display techniques that I have described exhibit the problem that light intended for the left eye "leaks" into the right eye, and vice versa. You could call it "crosstalk." Image artifacts created by such unwanted light are called *ghosts.*

There are several reasons for ghosting. In most displays, generation of light in response to the video signal is not instantaneous. For example, the LCD material of LCD displays takes a certain time to respond to the drive signal; the phosphors of PDPs have a certain decay time. When LCD and PDP displays are used for 3-D display using temporal multiplexing, if the display is still decaying while the opposite shutter opens, ghosting will result. In polarized displays, the polarizers (at both the display and the glasses) typically have incomplete extinction. In the Infitec scheme, practical optical filters have a certain degree of unwanted spectral overlap.

Reduction of ghosting to tolerable levels involves compensating the image data prior to its reaching the display. (In cinema, the processing is called *ghost-busting*). If a bright left-eye image element is anticipated to leak into the right, light can be artificially subtracted from the corresponding spatial location in the right image. Compensation is necessarily imperfect, though: If the corresponding location in the right image is black, no light can be subtracted, and the crosstalk persists. In cinema, compensation can potentially be accomplished either in mastering or in the projector's signal processing. Movie creators don't want to create separate masters for each 3-D display technology, so the second option is now usual.

Vergence and accommodation

The region of the human retina intersected by the optical axis is the *fovea;* it is a cluster of tightly packed cone photoreceptor cells. The fovea has an angular diameter of about 1°; it covers a small fraction of the visual field – a few tenths of a percent of the area corresponding to an HD image at normal viewing distance.

The oculomotor system of the eye includes muscles attached to the eyeball. The muscles "steer" the optical axis of each eye so that the fovea images light from the region of interest in the visual field. A few times per second, the muscles operate and the gaze shifts to a new point; the movement is called a *saccade.*

Vergence movements ideally involve rotation of the eyeballs with respect to the plane that joins their centres.

In normal binocular viewing of an actual scene, eye movements are made such that the optical axes of both eyes meet at the depth of the scene element of interest. The oculomotor system's control of the distance at which the optical axes meet is known as *vergence.*

Presbyopia is age-related loss of accommodation owing to the lens becoming less pliant. Even for people having normal vision, presbyopia typically makes reading glasses necessary beyond age 50.

The lens of the human eye is enclosed in a capsule that is somewhat pliable: The lens can change shape. Within the eyeball, surrounding the lens, is a muscle – the *ciliary* muscle. When the muscle is in its relaxed state, the lens capsule is at its flattest; the focal length of the lens is at its maximum. As the ciliary muscle contracts, the lens capsule becomes more spherical; focal length decreases, focussing on nearer objects. Muscle control over the lens is called *accommodation;* it is analagous to focusing of a camera lens.

In normal human vision viewing real objects, the vergence and accommodation systems work in concert.

In a stereo display of the kinds that I have described, both the left and right images are formed on the display surface, and that surface is a fixed distance away from the viewer. To keep the images sharp requires accommodation to the distance of the display screen – not to the apparent distance of the object that is formed by the stereo display system. As apparent depth departs from the screen distance – either to longer distance ("behind" the screen) or closer distance ("in front of" the screen), conflict between vergence and accommodation (*V-A conflict*) is likely to be experienced subconsciously by the viewer. Researchers have proven that V-A conflict is a major contributor to viewer discomfort in stereo 3-D.

Part 2

Theory

This page intentionally left blank

Filtering and sampling 20

This chapter explains how a one-dimensional signal is filtered and sampled prior to A-to-D conversion, and how it is reconstructed following D-to-A conversion. In the following chapter, *Resampling, interpolation, and decimation,* on page 221, I extend these concepts to conversions within the digital domain. In *Image digitization and reconstruction,* on page 237, I extend these concepts to the two dimensions of an image.

My explanation describes the original sampling of an analog signal waveform. If you are more comfortable remaining in the digital domain, consider the problem of shrinking a row of image samples by a factor of n (say, $n = 16$) to accomplish image resizing. You need to compute one output sample for each set of n input samples. This is the *resampling* problem in the digital domain. Its constraints are very similar to the constraints of original sampling of an analog signal.

When a one-dimensional signal (such as an audio signal) is digitized, each sample must encapsulate, in a single value, what might have begun as a complex analog waveform during the sample period. When a two-dimensional image is sampled, each sample encapsulates what might have begun as a potentially complex distribution of power over a small region of the image plane. In each case, a potentially large amount of information must be reduced to a single number.

Prior to sampling, detail within the sample interval must be discarded. The reduction of information prior to sampling is *prefiltering.* The challenge of sampling is to discard this information while avoiding the loss of information at scales larger than the sample pitch, all the time avoiding the introduction of artifacts. *Sampling theory* elaborates the conditions under which a signal can be sampled and accurately reconstructed, subject only to inevitable loss of detail that could not, in any event, be represented by a given number of samples in the digital domain.

Sampling theory was originally developed to describe one-dimensional signals such as audio, where the signal is a continuous function of the single dimension of time. Sampling theory has been extended to images, where an image is treated as a continuous function of two spatial coordinates (horizontal and vertical). Sampling theory can be further extended to the temporal sampling of moving images, where the third coordinate is time.

Sampling theorem

Assume that a signal to be digitized is well behaved, changing relatively slowly as a function of time. Consider the cosine signals shown in Figure 20.1 below, where the *x*-axis shows sample intervals. The top waveform is a cosine at the fraction 0.35 of the sampling rate f_S; the middle waveform is at $0.65 f_S$. The bottom row shows that identical samples result from sampling either of these waveforms: Either of the waveforms can masquerade as the same sample sequence. If the middle waveform is sampled, then reconstructed conventionally, the top waveform will result. This is the phenomenon of *aliasing*.

Symbol conventions used in this figure and following figures are as follows:

$\omega = 2\pi f_S$

$[\text{rad} \cdot \text{s}^{-1}]$

$t_S = \dfrac{1}{f_S}$

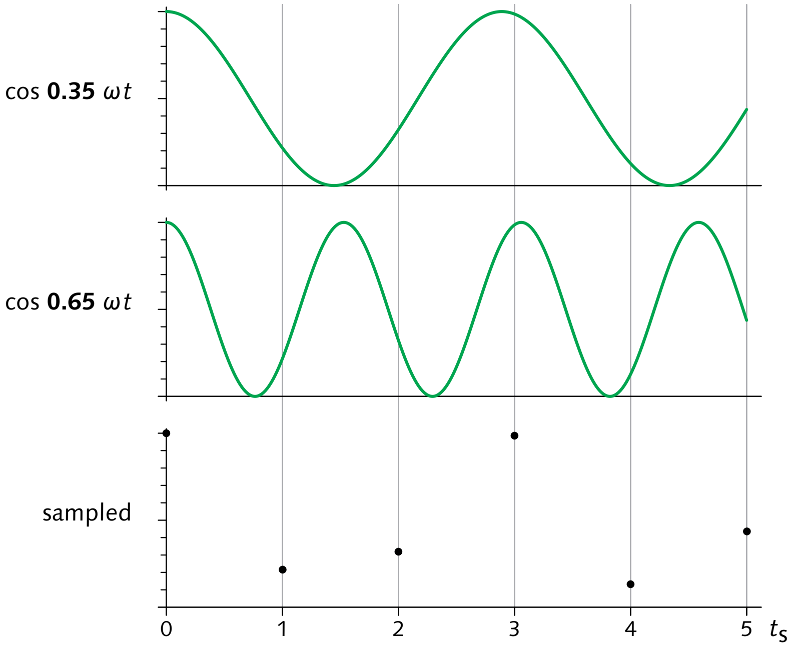


Figure 20.1 **Cosine waves less than and greater than $0.5 f_S$,** in this case at the fractions 0.35 and 0.65 of the sampling rate, produce exactly the same set of sampled values when point-sampled – they *alias*.

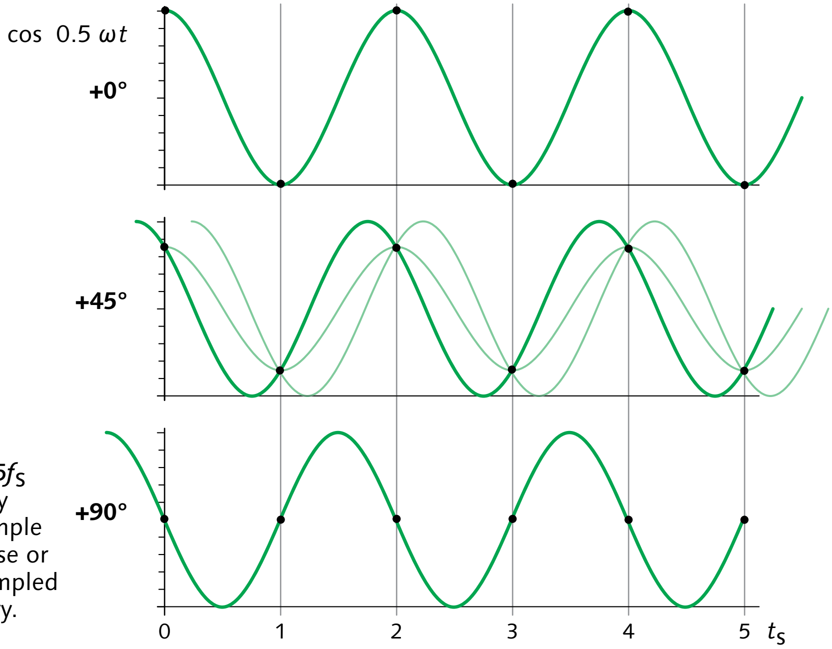


Figure 20.2 **Cosine waves at exactly 0.5f_S** cannot be accurately represented in a sample sequence if the phase or amplitude of the sampled waveform is arbitrary.

Sampling at exactly 0.5f_S

You might assume that a signal whose frequency is exactly half the sampling rate can be accurately represented by an alternating sequence of sample values, say, zero and one. In Figure 20.2 above, the series of samples in the top row is unambiguous (provided it is known that the amplitude of the waveform is unity). But the samples of the middle row could be generated from any of the three indicated waveforms, and the phase-shifted waveform in the bottom row has samples that are indistinguishable from a constant waveform having a value of 0.5. The inability to accurately analyze a signal at exactly half the sampling frequency leads to the strict "less-than" condition in the sampling sheorem, which I will now describe.

Nyquist essentially applied to signal processing a mathematical discovery made in 1915 by E.T. Whittaker. Later contributions were made by Claude Shannon (in the U.S.) and Aleksandr Kotelnikov (in Russia).

Harry Nyquist, at Bell Labs, published a paper in 1928 stating that to guarantee sampling of a signal without the introduction of aliases, all of the signal's frequency components must be contained strictly within half the sampling rate (now known as the *Nyquist rate*). If a signal meets this condition, it is said to satisfy the *Nyquist criterion*. The condition is usually

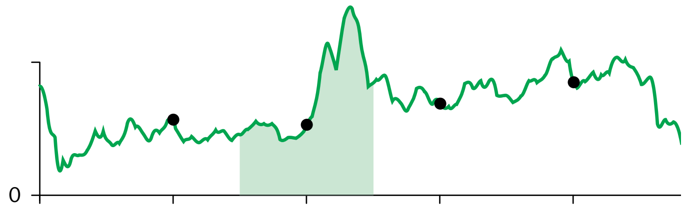


Figure 20.3 **Point sampling** runs the risk of choosing an extreme value that is not representative of the neighborhood surrounding the desired sample instant.

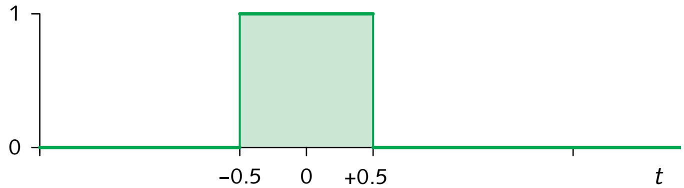


Figure 20.4 **The Box weighting function** (or "boxcar") has unity value throughout one sample interval; elsewhere, its value is zero.

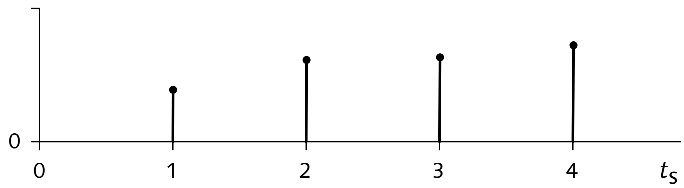


Figure 20.5 **Boxcar filtering** weights the input waveform with the boxcar weighting function: Each output sample is the average across one sample interval.

imposed by analog filtering, prior to sampling, that removes frequency components at $0.5f_S$ and higher. A filter must implement some sort of integration. In the example of Figure 20.1, no filtering was performed; the waveform was simply *point-sampled*. The lack of filtering admitted aliases. Figure 20.3 represents the waveform of an actual signal; point sampling at the indicated instants yields sample values that are not representative of the local neighborhood at each sampling instant.

Perhaps the most basic way to filter a waveform is to average the waveform across each sample period. Many different integration schemes are possible; these can be represented as weighting functions plotted as a function of time. Simple averaging uses the *boxcar* weighting function sketched in Figure 20.4; its value is unity during the sample period and zero outside that interval. Filtering with this weighting function is called *boxcar* filtering, since a sequence of these functions with different amplitudes resembles the profile of a freight train. Once the weighted values are formed the signal is represented by discrete values, plotted for this example in Figure 20.5. To plot these values as

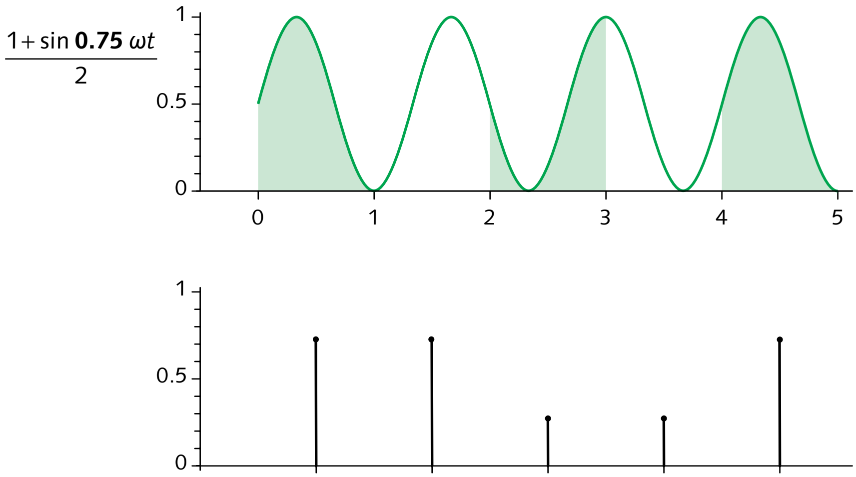


Figure 20.6 **Aliasing due to boxcar filtering.** The top graph shows a sine wave at $0.75 f_S$. The shaded area under the curve illustrates its integral computed by a boxcar function. The bottom graph shows that the sequence of resulting sample points is dominated by an alias at $0.25 f_S$.

amplitudes of a boxcar function would wrongly suggest that a boxcar function should be used as a reconstruction filter. The shading under the waveform of Figure 20.3 suggests box filtering.

A serious problem with boxcar filtering across each sample interval is evident in Figure 20.6 above. The top graph shows a sine wave at $0.75 f_S$; the signal exceeds the Nyquist frequency. The shaded regions show integration over intervals of one sample period. For the sine wave at $0.75 f_S$, sampled starting at zero phase, the first two integrated values are about 0.6061; the second two are about 0.3939. The dominant component of the filtered sample sequence, shown in the bottom graph, is one-quarter of the sampling frequency. Filtering using a one-sample-wide boxcar weighting function is inadequate to attenuate signal components above the Nyquist rate. An unwanted alias results.

Figure 20.6 provides another example of *aliasing:* Owing to a poor presampling filter, the sequence of sampled values exhibits a frequency component not present in the input signal. As this example shows, boxcar integration is not sufficient to prevent fairly serious aliasing.

Magnitude frequency response

To gain a general appreciation of aliasing, it is necessary to understand signals in the *frequency domain*. The previous section gave an example of inadequate filtering prior to sampling that created an unexpected alias upon sampling. You can determine whether a filter has an unexpected response at *any* frequency by presenting to the filter a signal that sweeps through all frequencies, from zero, through low frequencies, to some high frequency, plotting the response of the filter as you go. I graphed such a frequency sweep signal at the top of Figure 9.1, on page 98. The middle graph of that figure shows a response waveform typical of a low-pass filter (LPF), which attenuates high frequency signals. The magnitude response of that filter is shown in the bottom graph.

Strictly speaking, *amplitude* is an instantaneous measure that may take a positive or negative value. *Magnitude* is properly either an absolute value, or a squared or *root mean square* (RMS) value representative of amplitude over some time interval. The terms are often used interchangeably.

Magnitude response is the RMS average response over all phases of the input signal at each frequency. As you saw in the previous section, a filter's response can be strongly influenced by the phase of the input signal. To determine response at a particular frequency, you can test all phases at that frequency. Alternatively, provided the filter is linear, you can present just two signals – a cosine wave at the test frequency and a sine wave at the same frequency. The filter's magnitude response at any frequency is the absolute value of the vector sum of the responses to the sine and the cosine waves.

See *Linearity* on page 37.

Analytic and numerical procedures called *transforms* can be used to determine frequency response. The *Laplace transform* is appropriate for continuous functions, such as signals in the analog domain. The *Fourier transform* is appropriate for signals that are sampled periodically, or for signals that are themselves periodic. A variant intended for computation on data that has been sampled is the *discrete Fourier transform* (DFT). An elegant scheme for numerical computation of the DFT is the *fast Fourier transform* (FFT). The *z-transform* is essentially a generalization of the Fourier transform. All of these transforms represent mathematical ways to determine a system's response to sine waves over a range of frequencies and phases. The result of a transform is an expression or graph in terms of frequency.

Bracewell, Ronald N. (1985), *The Fourier Transform and Its Applications*, Second Edition (New York: McGraw-Hill).

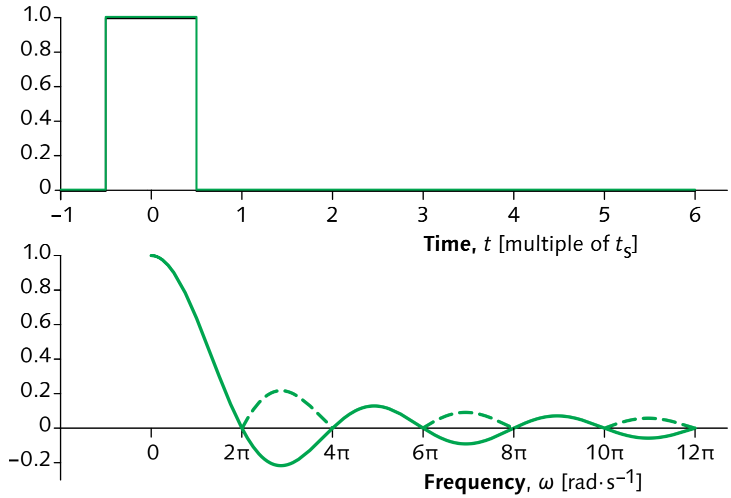


Figure 20.7 **Frequency response of a boxcar filter.** The top graph shows a boxcar weighting function, symmetrical around $t = 0$. Its frequency spectrum is a *sinc* function, shown underneath. The solid line shows that at certain frequencies, the filter causes phase inversion. Filter response is usually plotted as magnitude; phase inversion in the stopband is reflected as the absolute (magnitude) values shown in dashed lines.

$$\operatorname{sinc} \omega = \begin{cases} 1, & \omega = 0 \\ \dfrac{\sin \pi\omega}{\pi\omega}, & \omega \neq 0 \end{cases}$$

Eq 20.1 **The sinc function** (pronounced *sink*) is defined by this equation. Its argument is in radians per second (rad·s^{-1}); here I use the conventional symbol ω for that quantity. The term (sin x)/x (pronounced *sine ecks over ecks*) is often used synonymously with sinc, without mention of the units of the argument. If applied to frequency in hertz, the function could be written $(\sin 2\pi f)/2\pi f$.

sinc is unrelated to *sync* (synchronization).

Magnitude frequency response of a boxcar

The top graph of Figure 20.7 shows the weighting function of point sampling, as a function of time (in sample intervals). The Fourier transform of the boxcar function – that is, the magnitude frequency response of a boxcar weighting function – takes the shape of (sin x)/x. The response is graphed at the bottom of Figure 20.7, with the frequency axis in units of $\omega = 2\pi f_S$. Equation 20.1 in the margin defines the function. This function is so important that it has been given the special symbol *sinc*, introduced by Phillip M. Woodward in 1952 as a contraction of *sinus cardinalis*.

A presampling filter should have fairly uniform response below half the sample rate, to provide good sharpness, and needs to severely attenuate frequencies at and above half the sample rate, to achieve low aliasing. The bottom graph of Figure 20.7 shows that this requirement is not met by a boxcar weighting function. The graph of sinc predicts frequencies where aliasing can be introduced. Figure 20.6 showed an example of a sine wave at $0.75 f_S$; reading the value of sinc at 1.5 π from Figure 20.7 shows that aliasing is expected.

You can gain an intuitive understanding of the boxcar weighting function by considering that when the input frequency is such that an integer number of cycles lies under the boxcar, the response will be null. But when an integer number of cycles, plus a half-cycle, lies under the weighting function, the response will exhibit a local maximum that can admit an alias.

To obtain a presampling filter that rejects potential aliases, we need to pass low frequencies, up to almost half the sample rate, and reject frequencies above it. We need a frequency response that is constant at unity up to just below $0.5 f_S$, whereupon it drops to zero. We need a filter function whose *frequency* response – not time response – resembles a boxcar.

The sinc weighting function

Remarkably, the Fourier transform possesses the mathematical property of being its own inverse (within a scale factor). In Figure 20.7, the Fourier transform of a boxcar *weighting* function produced a sinc-shaped *frequency* response. Figure 20.8 opposite shows a sinc-shaped *weighting* function; it produces a boxcar-shaped *frequency* response. So, sinc weighting gives the *ideal lowpass filter* (ILPF), and sinc is the ideal temporal weighting function for use in a presampling filter. However, there are several theoretical and practical difficulties in using sinc. In practice, we approximate it.

A near-ideal filter in analog video is sometimes called a *brick wall* filter, though there is no precise definition of this term.

An analog filter's response is a function of frequency on the positive real axis. In analog signal theory, there is no upper bound on frequency. But in a digital filter the response to a test frequency f_T is identical to the response at f_T offset by any integer multiple of the sampling frequency: The frequency axis "wraps" at multiples of the sampling rate. Sampling theory also dictates "folding" around half the sample rate. Signal components having frequencies at or above the Nyquist rate cannot accurately be represented.

The temporal weighting functions used in video are usually symmetrical; nonetheless, they are usually graphed in a two-sided fashion. The frequency response of a filter suitable for real signals is symmetrical about

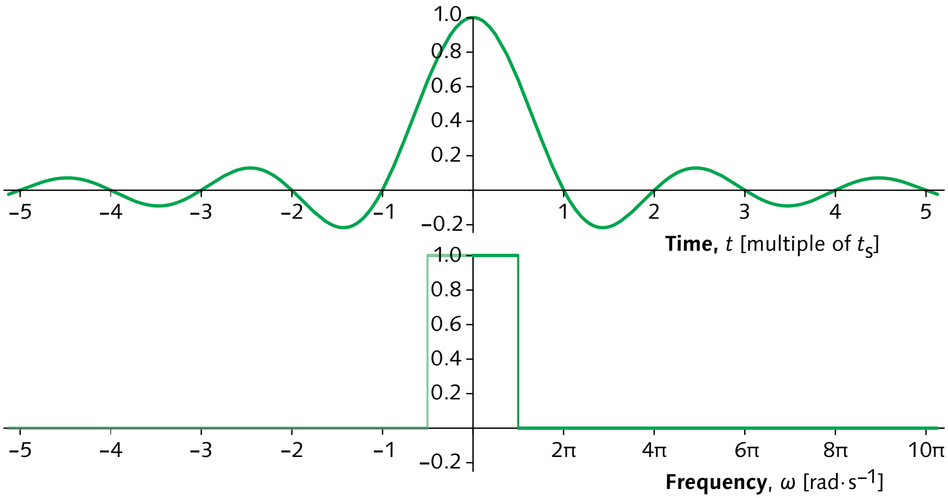


Figure 20.8 **The sinc weighting function** is shown in the top graph. Its frequency spectrum, shown underneath, has a boxcar shape: sinc weighting exhibits the ideal properties for a presampling filter. However, its infinite extent makes it physically unrealizable; also, its negative lobes make it unrealizable for transducers of light such as cameras, scanners, and displays. Many practical digital lowpass filters have coefficients that approximate samples of sinc.

zero; conventionally, frequency response is graphed in one-sided fashion starting at zero frequency ("DC"). Sometimes it is useful to consider or graph frequency response in two-sided style.

Frequency response of point sampling

The Fourier transform provides an analytical tool to examine frequency response: We can reexamine point sampling. Taking an instantaneous sample of a waveform is mathematically equivalent to using a weighting function that is unity at the sample instant, and zero everywhere else – the weighting function is an *impulse*. The Fourier transform of an impulse function is constant, unity, at all frequencies. A set of equally spaced impulses is an *impulse train;* its transform is also unity everywhere. The sampling operation is represented as multiplication by an impulse train. An unfiltered signal sampled by a set of impulses will admit aliases equally from all input frequencies.

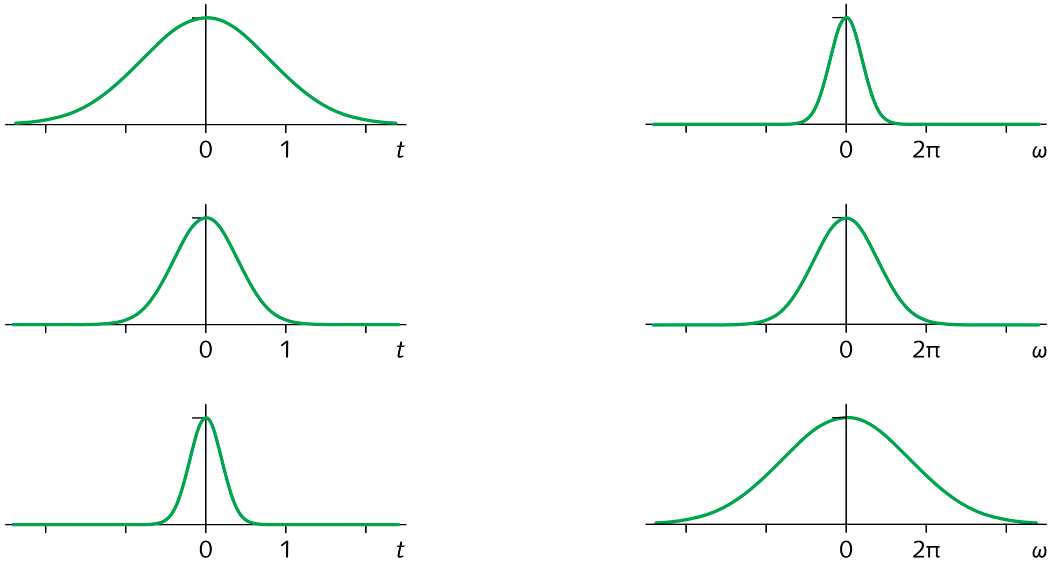


Figure 20.10 **Waveforms of three temporal extents** are shown on the left; the corresponding transforms are shown on the right. Spectral width is inversely proportional to temporal extent, not only for the Gaussians shown here, but for all waveforms.

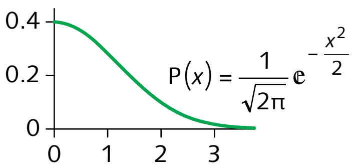


Figure 20.9 **A Gaussian function** is shown here in its one-sided form, with the scaling that is usual in statistics, where the function (augmented with mean and variance terms) is known as the *normal function*. Its integral is the *error function*, erf(x). The frequency response of cascaded Gaussian filters is Gaussian.

Fourier transform pairs

A Gaussian function – graphed in Figure 20.9 in the margin – is the identity function for the Fourier transform: It has the unique property of transforming to itself (within a scale factor). The Gaussian function has moderate spread both in the time domain and in the frequency domain; it has infinite extent, but becomes negligibly small more than a few units from the origin. The Gaussian function lies at the balance point between the distribution of power in the time domain and the distribution of power in the frequency domain.

Functions having short time durations transform to functions with widely distributed frequency components. Conversely, functions that are compact in their frequency representation transform to temporal functions with long duration. See Figure 20.10 above.

Figure 20.11 opposite shows Fourier transform pairs for several different functions. In the left column is a set of waveforms, with the Gaussian in the middle row; in the right column are the corresponding frequency spectra.

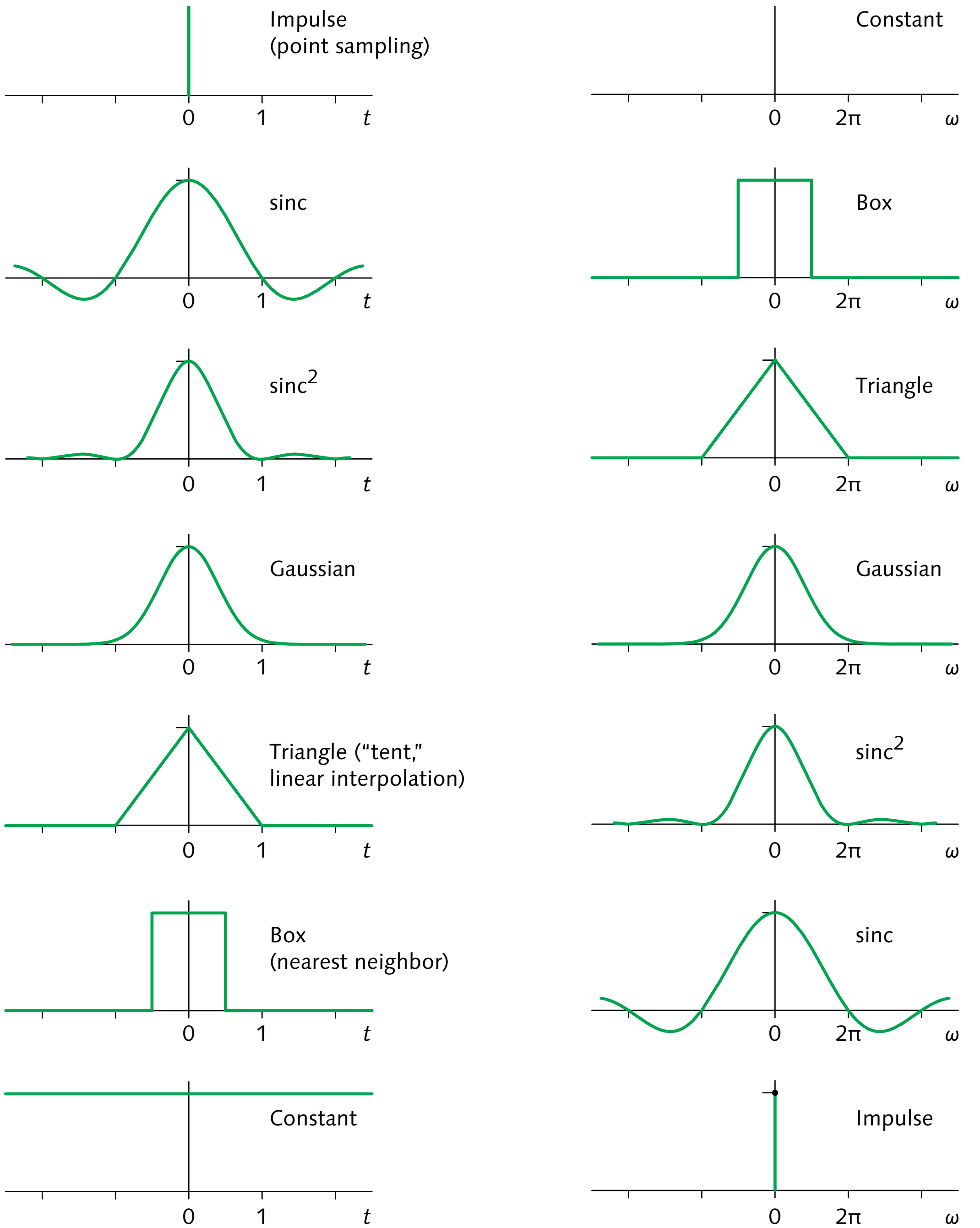


Figure 20.11 **Fourier transform pairs** for several functions are shown in these graphs. In the left column is a set of waveforms in the time domain; beside each waveform is its frequency spectrum.

Analog filters

Analog filtering is necessary prior to digitization, to bring a signal into the digital domain without aliases. I have described filtering as integration using different weighting functions; an antialiasing filter performs the integration using analog circuitry.

An analog filter performs integration by storing a magnetic field in an inductor (coil) using the electrical property of inductance (*L*), and/or by storing an electrical charge in a capacitor using the electrical property of capacitance (*C*). In low-performance filters, resistance (*R*) is used as well. An ordinary analog filter has an impulse response that is infinite in temporal extent.

The design of analog filters is best left to specialists.

Digital filters

Once digitized, a signal can be filtered directly in the digital domain. Design and implementation of such filters – in hardware, firmware, or software – is the domain of *digital signal processing* (DSP). Filters like the ones that I have been describing are implemented digitally by computing weighted sums of samples.

Averaging neighboring samples is the simplest form of *moving average* (MA) filter.

Perhaps the simplest digital filter is one that just sums adjacent samples; the weights in this case are [1, 1]. Figure 20.12 on the facing page shows the frequency response of such a [1, 1] filter. This filter offers minimal attenuation to very low frequencies; as signal frequency approaches half the sampling rate, the response follows a cosine curve to zero. This is a very simple, very cheap *lowpass filter* (LPF).

I have drawn in grey the filter's response from $0.5f_S$ to the sampling frequency. In a digital filter, frequencies in this region are indistinguishable from frequencies between $0.5f_S$ and 0. The gain of this filter at zero frequency (DC) is 2, the sum of its coefficients. Normally, the coefficients of such a filter are normalized to sum to unity, so that the overall DC gain of the filter is one. In this case the normalized coefficients would be [½, ½]. However, it is inconvenient to call this a [½, ½]-filter; colloquially, this is a [1, 1]-filter.

Digital filters can be implemented in software, firmware, or hardware. At the right side of each graph above, I show the block diagrams familiar to hardware

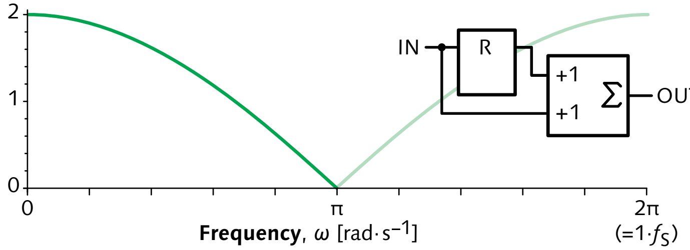


Figure 20.12 **A [1, 1] FIR filter** sums two adjacent samples; this forms a simple lowpass filter. I'll introduce the term *FIR* on page 207.

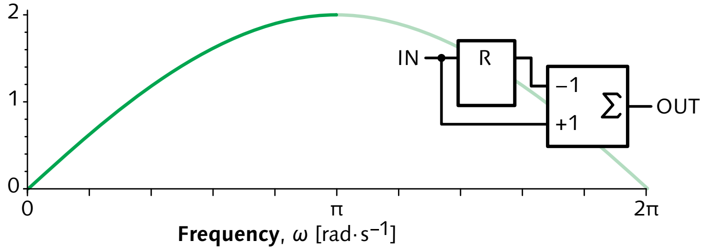


Figure 20.13 **A [1, –1] FIR filter** subtracts one sample from the previous sample; this forms a simple high-pass filter.

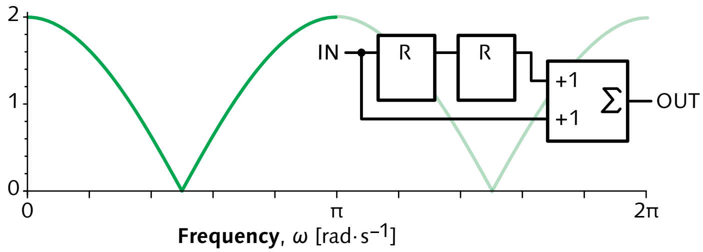


Figure 20.14 **A [1, 0, 1] FIR filter** averages a sample and the second preceding sample, ignoring the sample in between; this forms a bandreject ("notch," or "trap") filter at 0.25 f_S.

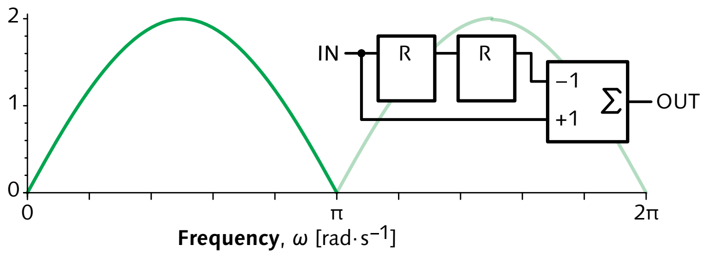


Figure 20.15 **A [1, 0, –1] FIR filter** subtracts one sample from the second previous sample, ignoring the sample in between; this forms a bandpass filter centered at 0.25 f_S.

designers. Each block labelled *R* designates a register; a series of these elements forms a shift register.

A simple *highpass filter* (HPF) is formed by subtracting each sample from the previous sample: This filter has weights [1, –1]. The response of this filter is graphed in Figure 20.13. In general, and in this case, a highpass filter is obtained when a lowpass-filtered version of a signal is subtracted from the unfiltered signal. The unfiltered signal can be considered as a two-tap filter having weights [1, 0]. Subtracting the weights

Subtracting a signal from a low-pass version of itself is equivalent to negating all of coefficients (weights) of the lowpass filter except the central coefficient, which is subtracted from unity.

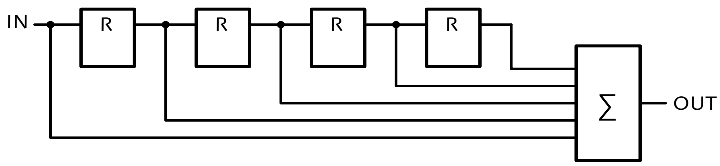


Figure 20.16 **A very simple 5-tap FIR filter** comprises four registers and an adder; five adjacent samples are summed. Prior to scaling to unity, the coefficients are [1, 1, 1, 1, 1].

A bandpass (bandstop) filter is considered *narrowband* if its passband (stopband) covers a 2:1 range of frequencies (*octave*) or less.

If a filter like that of Figure 20.16 has many taps, it needs many adders. Its arithmetic can be simplified by using an accumulator to form the running sum of input samples, another accumulator to form the running sum of outputs from the shift register, and a subtractor to take the difference of these sums. This structure is called a *cascaded integrator comb* (CIC).

[$\frac{1}{2}$, $\frac{1}{2}$] of the scaled lowpass filter from that yields the scaled weights [$\frac{1}{2}$, $-\frac{1}{2}$] of this highpass filter.

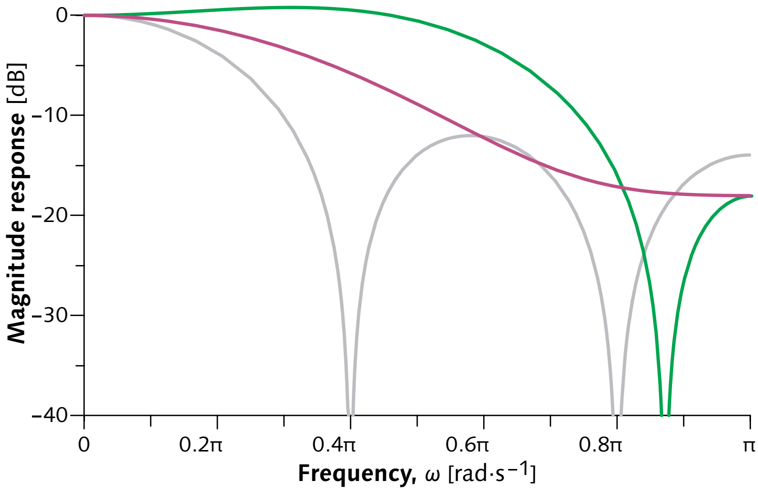
Figure 20.14 shows the response of a filter that adds a sample to the second previous sample, disregarding the central sample. The weights in this case are [1, 0, 1]. This forms a simple *bandreject filter* (BRF), also known as a *bandstop* or *notch filter,* or *trap.* Here, the response has a null at one quarter the sampling frequency. The scaled filter passes DC with no attenuation. This filter would make a mess of image data – if a picket fence whose pickets happened to lie at a frequency of $0.25f_S$ were processed through this filter, the pickets would average together and disappear! It is a bad idea to apply such a filter to image data, but this filter (and filters like it) can be very useful for signal processing functions.

Figure 20.15 shows the response of a filter that subtracts a sample from the second previous sample, disregarding the central sample. Its weights are [1, 0, –1]. This forms a simple *bandpass filter* (BPF). The weights sum to zero – this filter blocks DC. The BPF of this example is complementary to the [1, 0, 1] filter.

Figure 20.16 above shows the block diagram of a 5-tap FIR filter that sums five successive samples. As shown in the light grey curve in Figure 20.17 at the top of the facing page, this yields a lowpass filter. Its frequency response has two *zeros:* Any input signal at $0.2f_S$ or $0.4f_S$ will vanish; attenuation in the stopband reaches only about –12 dB, at $\frac{3}{10}$ of the sampling rate.

In the design of digital filters, control of frequency response is exercised in the choice of tap weights. Figure 20.18 at the bottom of the facing page shows the block diagram of a filter having fractional coefficients chosen from a Gaussian waveform. The magenta curve in Figure 20.17 shows that this set of tap weights yields a lowpass filter having a Gaussian frequency response. By using negative coefficients, low-frequency

Figure 20.17 **5-tap FIR filter responses** are shown for several choices of coefficient values (tap weights).

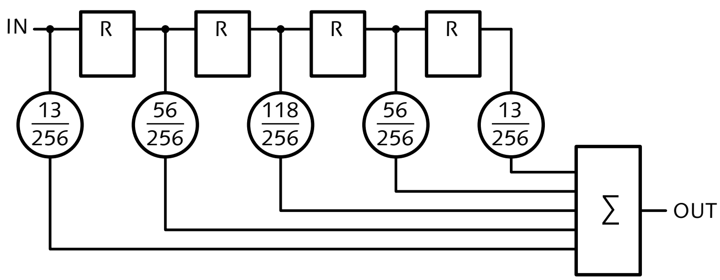


response can be extended without deteriorating performance at high frequencies. The black curve in Figure 20.17 shows the response of a filter having coefficients $[-32/256, 72/256, 176/256, 72/256, -32/256]$. This filter exhibits the same attenuation at high frequencies (about –18 dB) as the Gaussian, but has about twice the –6 dB frequency.

Negative coefficients, as in the last example here, potentially cause production of output samples that exceed unity. (In this example, output samples above unity are produced at input frequencies about $\omega=0.3\pi$, $1/6$ the sampling rate). If extreme values are clipped, artifacts will result. To avoid artifacts, the signal coding range must include suitable footroom and headroom.

The operation of an FIR filter amounts to multiplying a set of input samples by a set of filter coefficients (weights), and forming the appropriate set of sums of these products. The weighting can be implemented using multipliers or using table lookup techniques. With

Figure 20.18 **A 5-tap FIR filter including multipliers** has coefficients [13, 56, 118, 56, 13], scaled by $1/256$. The coefficients approximate a Gaussian; so does the frequency response. The multipliers can be implemented by table lookup.



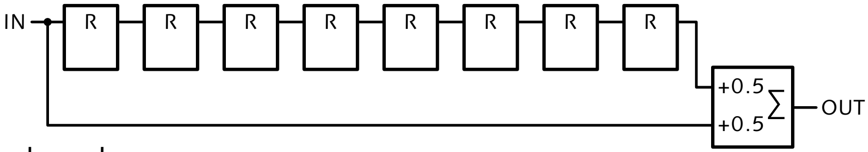


Figure 20.19 **A simple comb filter** includes several delay elements and an adder.

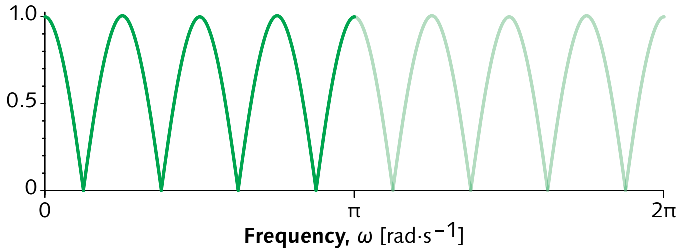


Figure 20.20 **The simple comb filter's response** resembles the teeth of a comb. This filter has unity response at zero frequency: It passes DC. A filter having weights [½, 0, 0, …, 0, −½] blocks DC.

respect to a complete set of input samples, this operation is called *convolution.* Ordinarily, convolution is conceptualized as taking place one multiplication at a time. An *n*-tap FIR filter can be implemented using a single multiplier-accumulator (MAC) component operating at *n* times the sample rate. A direct implementation with *n* multiplier components, or a multiplexed implementation with a single MAC, accepts input samples and delivers output samples in temporal order: Each coefficient needs to be presented to the filter *n* times. However, convolution is symmetrical with respect to input samples and coefficients: The same set of results can be produced by presenting filter coefficients one at a time to a MAC, and accumulating partial output sums for each output sample. FIR filters have many potential implementation structures.

For details concerning implementation structures, see the books by Lyons and Rorabaugh cited at the end of the chapter.

Figure 20.19 above shows the block diagram of an FIR filter having eight taps weighted [1, 0, 0, …, 0, 1]. The frequency response of this filter is shown in Figure 20.20. The response peaks when an exact integer number of cycles lie underneath the filter; it nulls when an integer-and-a-half number of cycles lie underneath. The peaks all have the same magnitude: The response is the same when exactly 1, 2, …, or *n* samples are within its window. The magnitude frequency response of such a filter has a shape resembling a comb, and such a filter is called a *comb filter.*

Impulse response

I have explained filtering as weighted integration along the time axis. I coined the term *temporal weighting function* to denote the weights. I consider my explanation of filtering in terms of its operation in the temporal domain to be more intuitive to a digital technologist than a more conventional explanation that starts in the frequency domain. But my term *temporal weighting function* is nonstandard, and I must now introduce the usual but nonintuitive term *impulse response*.

Details of the relationship between the Dirac delta, the Kronecker delta, and sampling in DSP are found on page 122 of Rorabaugh's book, cited at the end of the chapter.

An analog impulse signal has infinitesimal duration, infinite amplitude, and an integral of unity. (An analog impulse is conceptually equivalent to the Dirac or Kronecker deltas of mathematics.) A digital impulse signal is a solitary sample having unity amplitude amid a stream of zeros; The *impulse response* of a digital filter is its response to an input that is identically zero except for a solitary unity-valued sample.

Finite impulse response (FIR) filters

In each of the filters that I have described so far, only a few coefficients are nonzero. When a digital impulse is presented to such a filter, the result is simply the weighting coefficients scanned out in turn. The response to an impulse is limited in duration; the examples that I have described have *finite impulse response*. They are *FIR filters*. In these filters, the impulse response is identical to the set of coefficients. The digital filters that I described on page 202 implement temporal weighting directly. The impulse responses of these filters, scaled to unity, are [½, ½], [½, –½], [½, 0, ½], and [½, 0, –½], respectively.

In Equation 20.2, g is a sequence (whose index is enclosed in square brackets), not a function (whose argument would be in parentheses); s_j is sample number j.

The particular set of weights in Figure 20.18 approximate a sampled Gaussian waveform; so, the frequency response of this filter is approximately Gaussian. The action of this filter can be expressed algebraically:

Eq 20.2

$$g[j] = \frac{13}{256}s_{j-2} + \frac{56}{256}s_{j-1} + \frac{118}{256}s_j + \frac{56}{256}s_{j+1} + \frac{13}{256}s_{j+2}$$

Symmetry:
$f(x) = f(-x)$

Antisymmetry:
$f(x) = -f(-x)$

I have described impulse responses that are symmetrical around an instant in time. You might think $t=0$ should denote the beginning of time, but it is usually convenient to shift the time axis so that $t=0$ corresponds to the central point of a filter's impulse response. An FIR (or *nonrecursive*) filter has a limited number of coefficients that are nonzero. When the input impulse lies outside this interval, the response is zero. Most digital filters used in video are FIR filters, and most have impulse responses either symmetric or antisymmetric around $t = 0$.

You can view an FIR filter as having a fixed structure, with the data shifting along underneath. Alternatively, you might think of the *data* as being fixed, and the *filter* sliding across the data. Both notions are equivalent.

Physical realizability of a filter

In order to be implemented, a digital filter must be *physically realizable:* It is a practical necessity to have a temporal weighting function (impulse response) of limited duration. An FIR filter requires storage of several input samples, and it requires several multiplication operations to be performed during each sample period. The number of input samples stored is called the *order* of the filter, or its number of *taps*. If a particular filter has fixed coefficients, then its multiplications can be performed by table lookup. A straightforward technique can be used to exploit the symmetry of the impulse response to eliminate half the multiplications; this is often advantageous!

Here I use the word *truncation* to indicate the forcing to zero of a filter's weighting function beyond a certain tap. The nonzero coefficients in a weighting function may involve theoretical values that have been quantized to a certain number of bits. This *coefficient quantization* can be accomplished by *rounding* or by *truncation*. Be careful to distinguish between truncation of impulse response and truncation of coefficients.

When a temporal weighting function is truncated past a certain point, its transform – its frequency response characteristics – will suffer. The science and craft of filter design involves carefully choosing the order of the filter – that is, the position beyond which the weighting function is forced to zero. That position needs to be far enough from the center tap that the filter's high-frequency response is small enough to be negligible for the application.

Signal processing accommodates the use of impulse responses having negative values, and negative coefficients are common in digital signal processing. But image capture and image display involve sensing and generating light, which cannot have negative power, so negative weights cannot always be realized. If you study the transform pairs on page 201 you will see that your ability to tailor the frequency response of a filter is severely limited when you cannot use negative weights.

Impulse response is generally directly evident in the design of an FIR digital filter. Although it is possible to implement a boxcar filter directly in the analog domain, analog filters rarely implement temporal weighting directly, and the implementation of an analog filter generally bears a nonobvious relationship to its impulse response. Analog filters are best described in terms of Laplace transforms, not Fourier transforms. Impulse responses of analog filters are rarely considered directly in the design process. Despite the major conceptual and implementation differences, analog filters and FIR filters – and *IIR* filters, to be described – are all characterized by their frequency response.

Phase response (group delay)

Until now I have described the magnitude frequency response of filters. *Phase frequency* response – often called phase response – is also important. Consider a symmetrical FIR filter having 15 taps. No matter what the input signal, the output will have an effective delay of 8 sample periods, corresponding to the central sample of the filter's impulse response. The time delay of an FIR filter is constant, independent of frequency.

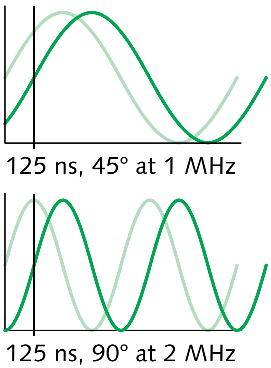


Figure 20.21 **Linear phase**

Consider a sine wave at 1 MHz, and a second sine wave at 1 MHz but delayed 125 ns. The situation is sketched in Figure 20.21 in the margin. The 125 ns delay could be expressed as a phase shift of 45° at 1 MHz. However, if the time delay remains constant and the frequency doubles, the phase offset doubles to 90°. With constant time delay, phase offset increases in direct (linear) proportion to the increase in frequency. Since in this condition phase delay is directly proportional to frequency, its synonym is *linear phase*. A closely related condition is *constant group delay*, where the first derivative of delay is constant but a fixed time delay may be present. All FIR filters exhibit constant group delay, but only symmetric FIR filters exhibit strictly linear phase.

It is characteristic of many filters – such as IIR filters, to be described in a moment – that delay varies somewhat as a function of frequency. An image signal contains many frequencies, produced by scene elements at different scales. If the horizontal displacement of a reproduced object were dependent upon

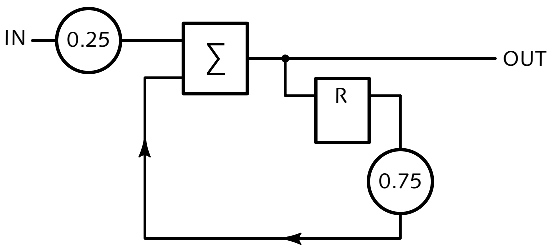


Figure 20.22 **An IIR ("recursive") filter** computes a weighted sum of input samples (here, just 0.25 times the current sample), and adds to this a weighted sum of previous result samples. Every IIR filter exhibits nonlinear phase response.

frequency, objectionable artifacts would result. Symmetric FIR filters exhibit linear phase in their passbands, and avoid this artifact. So, in image processing and in video, FIR filters are strongly preferred over other sorts of filters: Linear phase is a highly desirable property in a video system.

Infinite impulse response (IIR) filters

What a signal processing engineer calls an IIR filter is known in the finance and statistics communities as *autoregressive moving average* (ARMA).

The digital filters described so far have been members of the FIR class. A second class of digital filter is characterized by having a potentially *infinite impulse response* (IIR). An IIR (or *recursive*) filter computes a weighted sum of input samples – as is the case in an FIR filter – but adds to this a weighted sum of previous *output* samples.

A simple IIR is sketched in Figure 20.22: The input sample is weighted by $1/4$, and the previous output is weighted by $3/4$. These weighted values are summed to form the filter result. The filter result is then fed back to become an input to the computation of the next sample. The impulse response jumps rapidly upon the onset of the input impulse, and tails off over many samples. This is a simple one-tap lowpass filter; its time-domain response closely resembles an analog RC lowpass filter. A highpass filter is formed by taking the difference of the input sample from the previously stored filter result.

In an IIR filter having just one tap, the designer's ability to tailor frequency response is severely limited. An IIR filter can be extended by storing several previous filter results, and adding (or subtracting) a fraction of each to a fraction of the current input sample. In such a multitap IIR filter, a fine degree of control can be exercised over frequency response using just a handful of taps. Just three or four taps in an IIR filter can achieve frequency response that might take 20 taps in an FIR filter.

However, there's a catch: In an IIR filter, both attenuation and delay depend upon frequency. In the terminology of the previous section, an IIR filter exhibits nonlinear phase. Typically, low-frequency signals are delayed more than high-frequency signals. As I have explained, variation of delay as a function of frequency is potentially a very serious problem in video.

Compensation of undesired phase response in a filter is known as *equalization*. This is unrelated to the *equalization* pulses that form part of sync.

An IIR filter cannot have exactly linear phase, although a complex IIR filter can be designed to have arbitrarily small phase error. Because IIR filters usually have poor phase response, they are not ordinarily used in video. (A notable exception is the use of field- and frame-based IIR filters in temporal noise reduction, where the delay element comprises a field or frame of storage.)

The terms *nonrecursive* and *recursive* are best used to describe filter implementation structures.

Owing to the dependence of an IIR filter's result upon its previous results, an IIR filter is necessarily recursive. However, certain recursive filters have finite impulse response, so a recursive filter does *not* necessarily have infinite impulse response.

Lowpass filter

Here I represent frequency by the symbol ω, whose units are radians per second ($rad \cdot s^{-1}$). A digital filter scales with its sampling frequency; using ω is convenient because the sampling frequency is always $\omega = 2\pi$ and the half-sampling (Nyquist) frequency is always π.

Some people define bandwidth differently than I do.

A lowpass filter lets low frequencies pass undisturbed, but attenuates high frequencies. Figure 20.23 overleaf characterizes a lowpass filter. The response has a *passband*, where the filter's response is nearly unity; a *transition band*, where the response has intermediate values; and a *stopband*, where the filter's response is nearly zero. For a lowpass filter, the *corner frequency,* ω_C – sometimes called *bandwidth,* or *cutoff frequency* – is the frequency where the magnitude response of the filter has fallen 3 dB from its magnitude at a reference frequency (usually zero, or DC). In other words, at its corner frequency, the filter's response has fallen to 0.707 of its response at DC.

The passband is characterized by the passband edge frequency ω_P and the passband ripple δ_P (sometimes denoted δ_1). The stopband is characterized by its edge frequency ω_S and ripple δ_S (sometimes denoted δ_2). The *transition band* lies between ω_P and ω_S; it has width $\Delta\omega = \omega_S - \omega_P$.

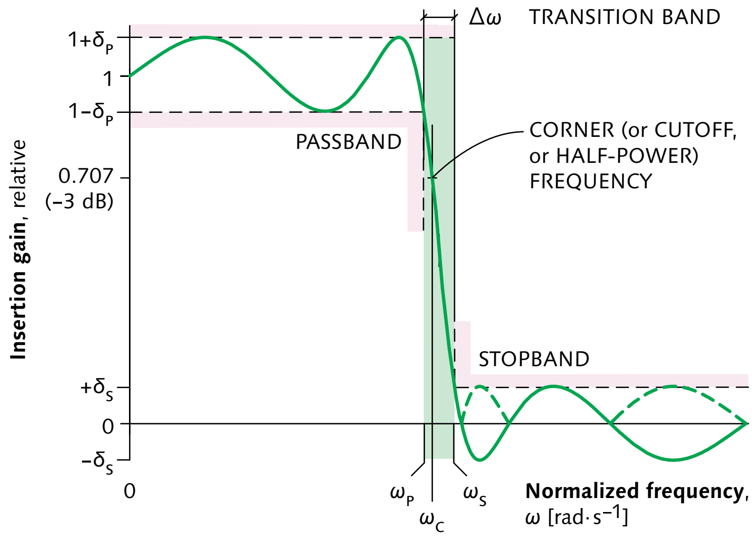


Figure 20.23 **Lowpass filter characterization.** A lowpass filter for use in video sampling or reconstruction has a corner frequency ω_C, where the attenuation is 0.707. (At the corner frequency, output power is half the input power.) In the *passband*, response is unity within δ_P, usually 1% or so. In the *stopband*, response is zero within δ_S, usually 1% or so. The *transition band* lies between the edge of the passband and the edge of the stopband; its width is $\Delta\omega$.

The complexity of a lowpass filter is roughly determined by its *normalized transition bandwidth* (or *transition ratio*) $\Delta\omega/2\pi$. The narrower the transition band, the more complex the filter. Also, the smaller the ripple in either the passband or the stopband, the more complex the filter. FIR filter tap count can be estimated by this formula, due to Bellanger:

Eq 20.3

$$N_e \approx \frac{2\pi}{\Delta\omega} \cdot \frac{2}{3} \log_{10}\left(\frac{1}{10\delta_P\delta_S}\right)$$

Bellanger, Maurice (2000), *Digital Processing of Signals: Theory and Practice*, Third Edition (Chichester, England: Wiley): 124.

In analog filter design, frequency response is generally graphed in log–log coordinates, with the frequency axis in units of log hertz (Hz), and magnitude response in decibels (dB). In digital filter design, frequency is usually graphed linearly from zero to half the sampling frequency. The passband and stopband response of a digital filter are usually graphed logarithmically; the passband response is often magnified to emphasize small departures from unity.

The templates standardized in BT.601 for a studio digital video presampling filter are shown in Figure 20.24 opposite. The response of a practical lowpass filter meeting this tremplate is shown in

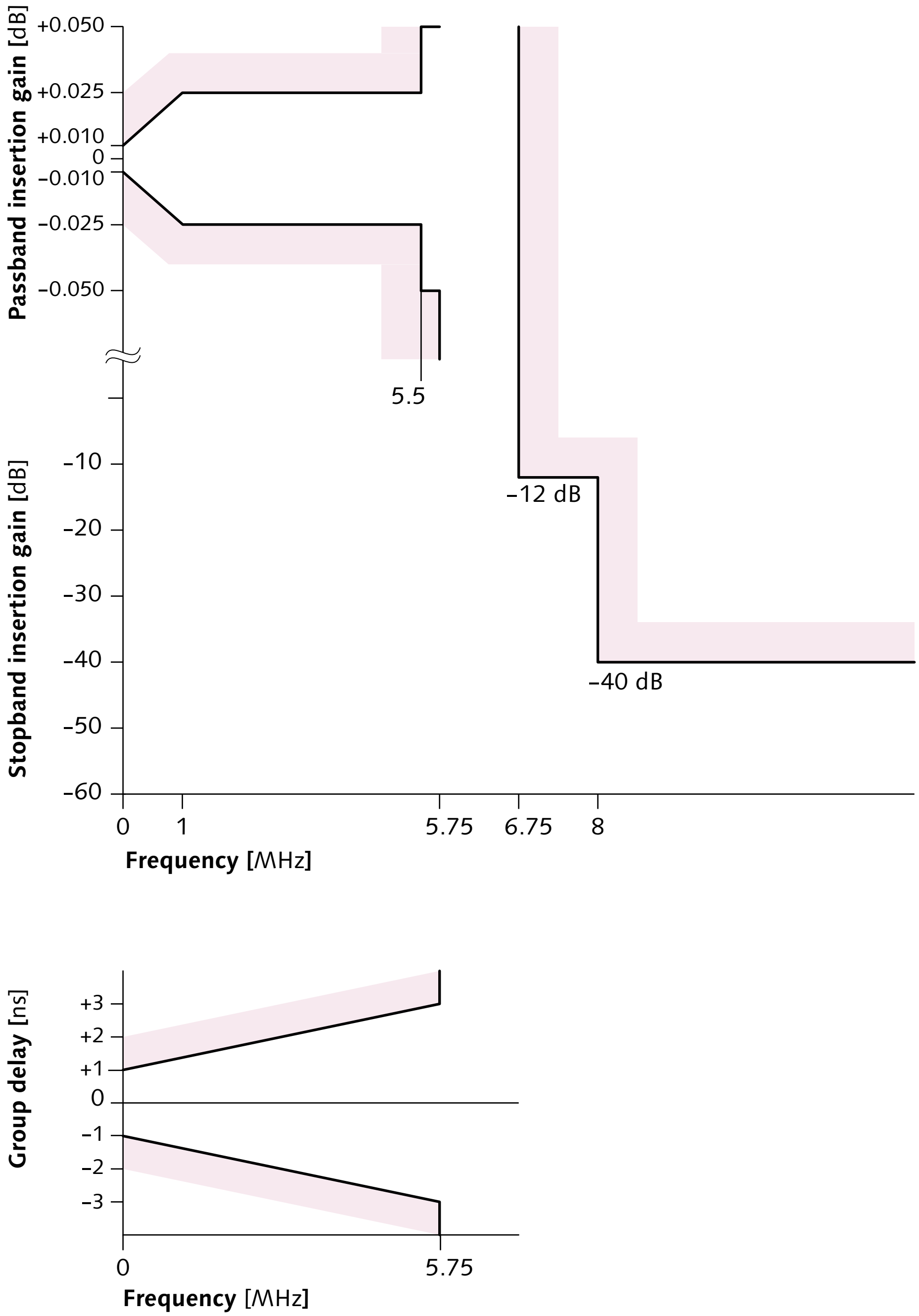


Figure 20.24 **BT.601 filter templates** are standardized for studio digital video systems in BT.601-5. The top template shows frequency response, detailing the passband (at the top) and the stopband (in the middle). The bottom template shows the group delay specification.

Figure 20.25, on page 215. This is a halfband filter, intended for use with a sampling frequency of 27 MHz; its corner frequency is $0.25f_S$. A consumer filter might have ripple two orders of magnitude worse than this.

Digital filter design

I describe *risetime* on page 543. In response to a step input, a Gaussian filter has a risetime very close to ⅓ of the period of one cycle at the corner frequency.

A simple way to design a digital filter is to use coefficients that comprise an appropriate number of point-samples of a theoretical impulse response. Coefficients beyond a certain point – the *order* of the filter – are simply omitted. Equation 20.4 implements a 9-tap filter that approximates a Gaussian:

Eq 20.4

$$g[j] = \frac{1s_{j-4} + 9s_{j-3} + 43s_{j-2} + 110s_{j-1} + 150s_j + 110s_{j+1} + 43s_{j+2} + 9s_{j+3} + 1s_{j+4}}{476}$$

Omission of coefficients causes frequency response to depart from the ideal. If the omitted coefficients are much greater than zero, actual frequency response can depart significantly from the ideal.

Another approach to digital filter design starts with the ILPF. Its infinite extent can be addressed by simply truncating the weights – that is, forcing the weights to zero – outside a certain interval, say outside the region 0±4 sample periods. This will have an unfortunate effect on the frequency response, however: The frequency response will exhibit overshoot and undershoot near the transition band.

We could use the term *weighting*, but sinc itself is a weighting function, so we choose a different word: *windowing*.

For details about windowing, see Lyons or Rorabaugh, at the end of the chapter, or WOLBERG, GEORGE (1990), *Digital Image Warping* (Los Alamitos, Calif.: IEEE).

Poor spectral behavior of a truncated sinc can be mitigated by applying a weighting function that peaks at unity at the center of the filter and diminishes gently to zero at the extremities of the interval. This is referred to as applying a *windowing* function. Design of a filter using the windowing method begins with scaling of sinc along the time axis to choose the corner frequency and choosing a suitable number of taps. Each tap weight is then computed as a sinc value multiplied by the corresponding window value. A sinc can be truncated through multiplication by a rectangular window. Perhaps the simplest nontrivial window has a triangular shape; this is also called the *Bartlett* window. The *von Hann* window (often wrongly called "Hanning") has a windowing function that is a single cycle of a raised cosine. Window functions such as von Hann are fixed by the corner frequency and the number of filter taps;

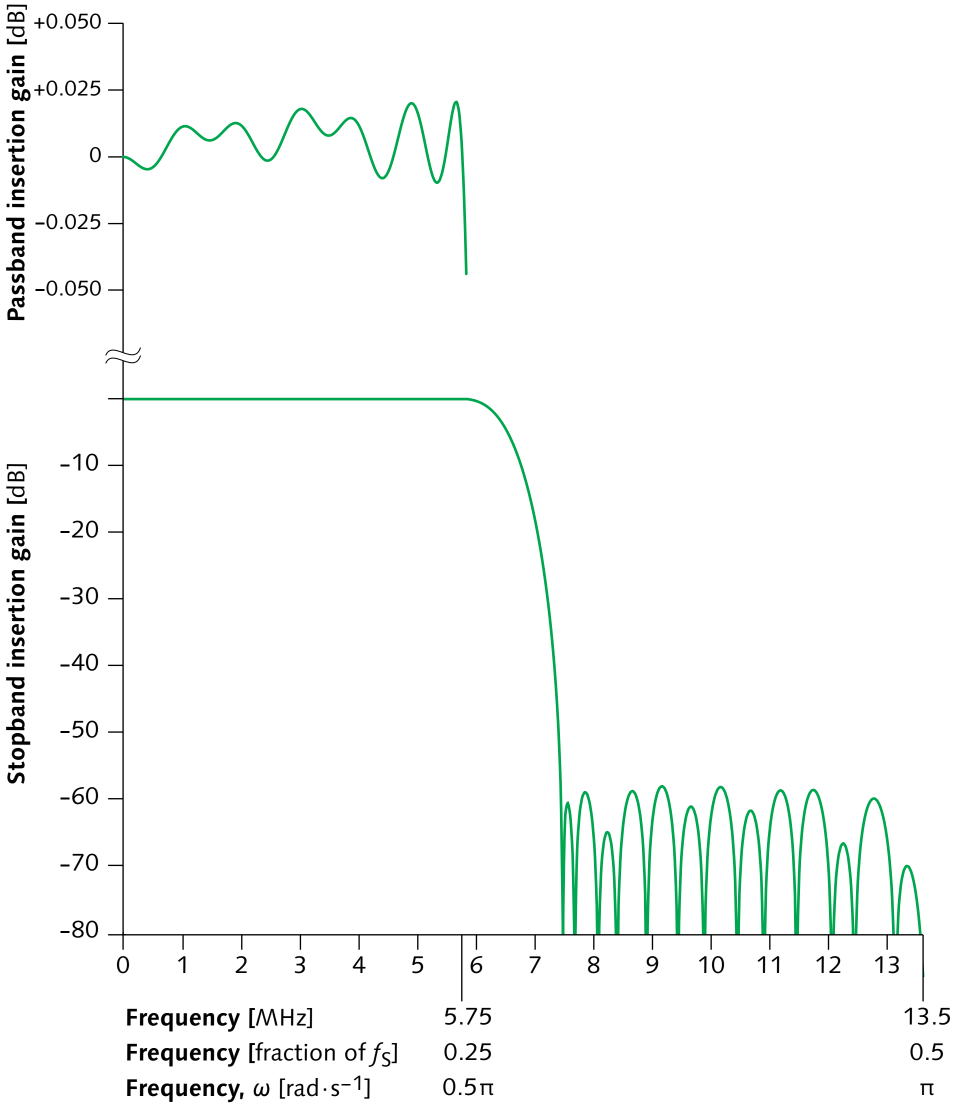


Figure 20.25 **Halfband filter.** This graph shows the frequency response of a practical filter whose corner is at one-quarter its sampling frequency of 27 MHz. The graph is linear in the abscissa (frequency) and logarithmic in the ordinate (response). The top portion shows that the passband has an overall gain of unity and a uniformity (*ripple*) of about ±0.02 dB: In the passband, its gain varies between about 0.997 and 1.003. The bottom portion shows that the stopband is rejected with an attenuation of about –60 dB: The filter has a gain of about 0.001 at these frequencies. This data, for the GF9102A halfband filter, was kindly provided by Gennum Corporation.

Figure 20.26 **A 25-tap lowpass FIR filter**

$$
\begin{aligned}
g[i] = {} & 0.098460\,s_{i-12} \\
& +0.009482\,s_{i-11} \\
& -0.013681\,s_{i-10} \\
& +0.020420\,s_{i-9} \\
& -0.029197\,s_{i-8} \\
& +0.039309\,s_{i-7} \\
& -0.050479\,s_{i-6} \\
& +0.061500\,s_{i-5} \\
& -0.071781\,s_{i-4} \\
& +0.080612\,s_{i-3} \\
& -0.087404\,s_{i-2} \\
& +0.091742\,s_{i-1} \\
& +0.906788\,s_{i} \\
& +0.091742\,s_{i+1} \\
& -0.087404\,s_{i+2} \\
& +0.080612\,s_{i+3} \\
& -0.071781\,s_{i+4} \\
& +0.061500\,s_{i+5} \\
& -0.050479\,s_{i+6} \\
& +0.039309\,s_{i+7} \\
& -0.029197\,s_{i+8} \\
& +0.020420\,s_{i+9} \\
& -0.013681\,s_{i+10} \\
& +0.009482\,s_{i+11} \\
& +0.098460\,s_{i+12}
\end{aligned}
$$

no control can be exercised over the width of the transition band. The *Kaiser* window has a single parameter that controls that width. For a given filter order, if the transition band is made narrower, then stopband attenuation is reduced. The Kaiser window parameter allows the designer to determine this tradeoff.

A windowed sinc filter has much better performance than a truncated sinc, and windowed design is so simple that there is no excuse to use sinc without windowing. In most engineering applications, however, filter performance is best characterized in the frequency domain, and the frequency-domain performance of windowed sinc filters is suboptimal: The performance of an *n*-tap windowed sinc filter can be bettered by an *n*-tap filter whose design has been suitably optimized.

Few closed-form methods are known to design optimum digital filters. Design of a high-performance filter usually involves successive approximation, optimizing by trading design parameters back and forth between the time and frequency domains. The classic method was published by J.H. McLellan, T.W. Parks, and L.R. Rabiner ("MPR"), based upon an algorithm developed by the Russian mathematician E.Ya. Remez. In the DSP community, the method is often called the "Remez exchange."

The coefficients of a high-quality lowpass filter for studio video are shown in Figure 20.26 in the margin.

Reconstruction

Digitization involves sampling and quantization; these operations are performed in an analog-to-digital converter (ADC). Whether the signal is quantized then sampled, or sampled then quantized, is relevant only within the ADC: The order of operations is immaterial outside that subsystem. Modern video ADCs quantize first, then sample.

I have explained that filtering is generally required prior to sampling in order to avoid the introduction of aliases. Avoidance of aliasing in the sampled domain has obvious importance. In order to avoid aliasing, an analog presampling filter needs to operate prior to analog-to-digital conversion. If aliasing is avoided, then the sampled signal can, according to Shannon's theorem, be reconstructed without aliases.

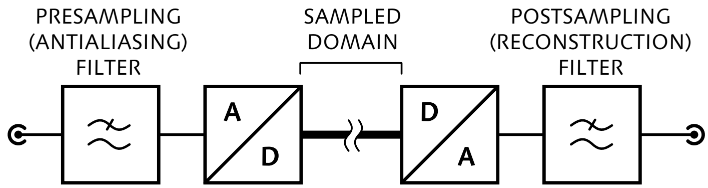


Figure 20.27 **Sampling and reconstruction**

To reconstruct an analog signal, an analog reconstruction filter is necessary following digital-to-analog (D-to-A) conversion. The overall flow is sketched in Figure 20.27 above.

Reconstruction close to 0.5f_S

Consider the example in Figure 20.28 below of a sine wave at 0.44f_S. This signal meets the sampling criterion, and can be perfectly represented in the digital domain. However, from an intuitive point of view, it is difficult to predict the underlying sinewave from samples 3, 4, 5, and 6 in the lower graph. When reconstructed using a Gaussian filter, the high-frequency signal vanishes. To be reconstructed accurately, a waveform with a significant amount of power near half the sampling rate must be reconstructed with a high-quality filter.

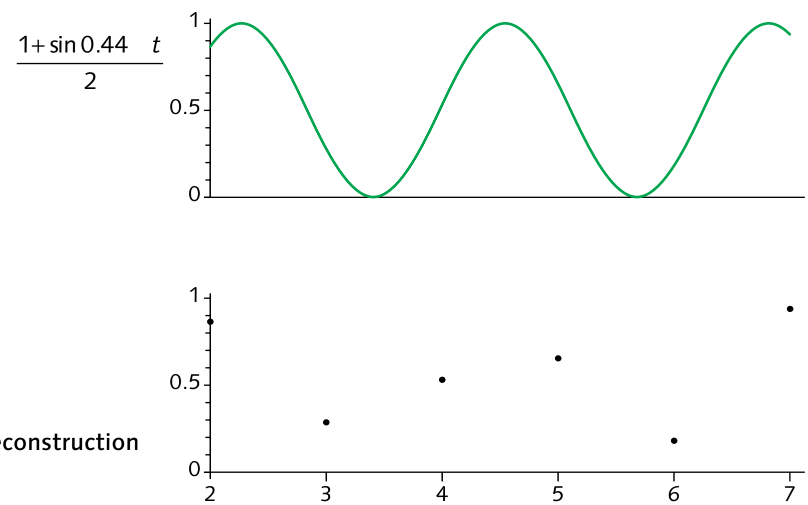


Figure 20.28 **Reconstruction close to 0.5f_S**

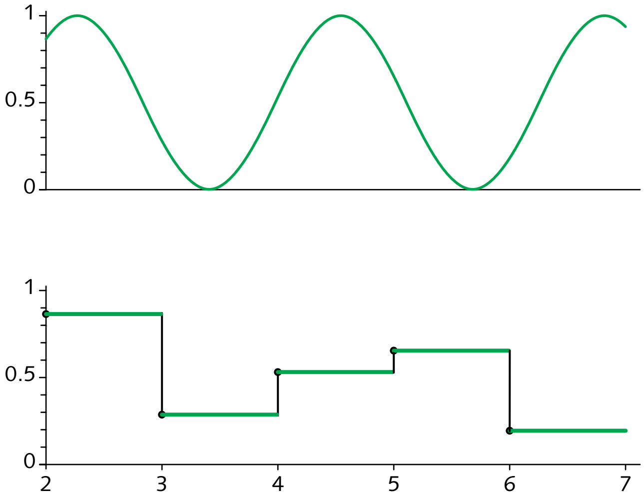


Figure 20.29 **D-to-A conversion with a boxcar waveform** is equivalent to a DAC producing an impulse train followed by a boxcar filter with its sinc response. Frequencies close to $0.5f_S$ are attenuated.

"(sin *x*)/*x*" correction

I place "(sin *x*)/*x*" in quotes: With the argument properly scaled it is (sin π*x*)/(π*x*), but it is almost always pronounced *sine-ecks-over-ecks*, with argument scaling implicit.

I have described how it is necessary for an analog reconstruction filter to follow digital-to-analog conversion. If the DAC produced an impulse "train" where the amplitude of each impulse was modulated by the corresponding code value, a classic lowpass filter would suffice: All would be well if the DAC output resembled my "point" graphs, with power at the sample instants and no power in between. Recall that a waveform comprising just unit impulses has uniform frequency response across the entire spectrum.

Unfortunately for analog reconstruction, a typical DAC does not produce an impulse waveform for each sample. It would be impractical to have a DAC with an impulse response, because signal power is proportional to the integral of the signal, and the amplitude of the impulses would have to be impractically high for the integral of the impulses to achieve adequate signal power. Instead, each converted sample value is held for the entire duration of the sample: A typical DAC produces a boxcar waveform. A boxcar waveform's frequency response is described by the sinc function.

You might consider a DAC's boxcar waveform to be a "sample-and-hold" operation, but that term is normally used in conjunction with an A-to-D converter, or circuitry that lies in front of an ADC.

In Figure 20.29 above, the top graph is a sine wave at $0.44f_S$; the bottom graph shows the boxcar waveform produced by a conventional DAC. Even with a high-quality reconstruction filter, whose response extends close to half the sampling rate, it is evident that

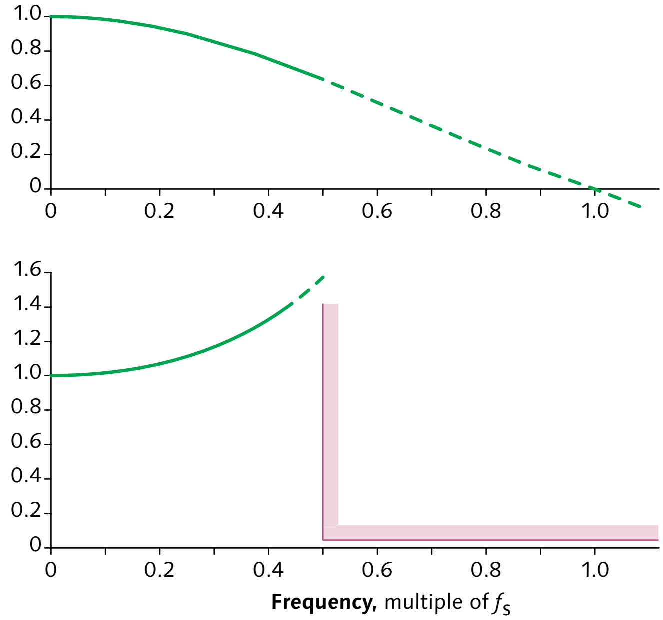


Figure 20.30 **"(sin *x*)/*x*" correction** is necessary following (or in principle, preceding) digital-to-analog conversion when a DAC with a typical boxcar output waveform is used. The frequency response of a boxcar-waveform DAC is shown in the upper graph. The lower graph shows the response of the "(sin *x*)/*x*" correction filter necessary to compensate its high frequency falloff.

reconstruction by a boxcar function reduces the magnitude of high-frequency components of the signal.

The DAC's holding of each sample value throughout the duration of its sample interval (*zero-order hold,* ZOH) corresponds to a filtering operation, with a frequency response of sinc. The top graph of Figure 20.30 shows the attenuation due to this phenomenon.

The effect is overcome by "(sin *x*)/*x*" *correction:* The frequency response of the reconstruction filter is modified to include peaking corresponding to the reciprocal of sinc. In the passband, the filter's response increases gradually to about 4 dB above its response at DC, to compensate the loss. Above the passband edge frequency, the response of the filter must decrease rapidly to produce a large attenuation near half the sampling frequency, to provide alias-free reconstruction. The bottom graph of Figure 20.30 shows the idealized response of a filter having "(sin *x*)/*x*" correction.

This chapter has detailed one-dimensional filtering. In *Image digitization and reconstruction,* I will introduce two- and three-dimensional sampling and filters.

Further reading

For an approachable introduction to the concepts, theory, and mathematics of digital signal processing (DSP), see Lyons. For an alternative point of view, see Rorabaugh's book; it includes the source code for programs to design filters – that is, to evaluate filter coefficients. For comprehensive and theoretical coverage of DSP, see Mitra and Kaiser.

Lyons, Richard G. (1997), *Understanding Digital Signal Processing* (Reading, Mass.: Addison Wesley).

McClellan, James H. and Parks, Thomas W. (2005), "A personal history of the Parks-McClellan algorithm," *IEEE Signal Processing Magazine* **22** (2): 82–86.

Mitra, Sanjit K., and James F. Kaiser (1993), *Handbook for Digital Signal Processing* (New York: Wiley).

Rorabaugh, C. Britton (1999), *DSP Primer* (New York: McGraw-Hill).

Resampling, interpolation, and decimation 21

In video and audio signal processing, it is often necessary to take a set of sample values and produce another set that approximates the samples that would have resulted had the original sampling occurred at different instants – at a different rate, or at a different phase. This is called *resampling.* (In PC parlance, resampling for the purpose of picture resizing is called *scaling.*) Resampling is an essential part of video processes such as these:

- Chroma subsampling (e.g., 4:4:4 to 4:2:2)
- Downconversion (e.g., HD to SD) and upconversion (e.g., SD to HD)
- Aspect ratio conversion (e.g., 4:3 to 16:9)
- Conversion among different sample rates of digital video standards (e.g., $4f_{SC}$ to 4:2:2, 13.5 MHz)
- Picture resizing in digital video effects (DVE)

One-dimensional resampling applies directly to digital audio, in applications such as changing sample rate from 48 kHz to 44.1 kHz. In video, 1-D resampling can be applied horizontally or vertically. Resampling can be extended to a two-dimensional array of samples. Two approaches are possible. A horizontal filter, then a vertical filter, can be applied in cascade (tandem) – this is the *separable* approach. Alternatively, a direct form of 2-D spatial interpolation can be implemented.

Upsampling produces more result samples than input samples. In audio, new samples can be estimated at a higher rate than the input, for example when digital audio sampled at 44.1 kHz is converted to the 48 kHz professional rate used with video. In video, upsampling is required in the spatial upconversion from 1280×720

I write resampling ratios in the form *input samples*:*output samples*. With my convention, a ratio less than unity is upsampling.

HD to 1920×1080 HD: 1280 samples in each input line must be converted to 1920 samples in the output, an upsampling ratio of 2:3.

One way to accomplish upsampling by an integer ratio of 1:n is to interpose $n-1$ zero samples between each pair of input samples. This causes the spectrum of the original signal to repeat at multiples of the original sampling rate. The repeated spectra are called "images." (This is a historical term stemming from radio; it has nothing to do with pictures!) These "images" are then eliminated (or at least attenuated) by an anti-imaging lowpass filter. In some upsampling structures, such as the Lagrange interpolator that I will describe later in this chapter, filtering and upsampling are intertwined.

Downsampling produces fewer result samples than input samples. In audio, new samples can be created at a lower rate than the input. In video, downsampling is required when converting $4f_{SC}$ NTSC digital video to BT.601 ("4:2:2") digital video: 910 samples in each input line must be converted to 858 samples in the output, a downsampling ratio of 35:33; for each 35 input samples, 33 output samples are produced.

In an original sample sequence, signal content from DC to nearly $0.5f_S$ can be represented. After downsampling, though, the new sample rate may be lower than that required by the signal bandwidth. After downsampling, meaningful signal content is limited by the Nyquist criterion at the *new* sampling rate – for example, after 4:1 downsampling, signal content is limited to ⅛ of the original sampling rate. To avoid the introduction of aliases, lowpass filtering is necessary prior to, or in conjunction with, downsampling. The corner frequency depends upon the downsampling ratio; for example, a 4:1 ratio requires a corner less than $0.125f_S$. Downsampling with an integer ratio of n:1 can be thought of as prefiltering (antialias filtering) for the new sampling rate, followed by the discarding of $n-1$ samples between original sample pairs.

Consider resampling pseudocolour data. If you treat the data as continuous, the resulting image is liable to contain colours not in the source. If you use nearest-neighbour resampling to avoid generating "new" sample values, geometry will suffer.

Resampling produces new samples that assume that neighbouring input samples are related by a continuous function. If the underlying function is not continuous, problems can be expected. For example, pseudocolour images are not continuous: They cannot be meaningfully resampled without creating artifacts.

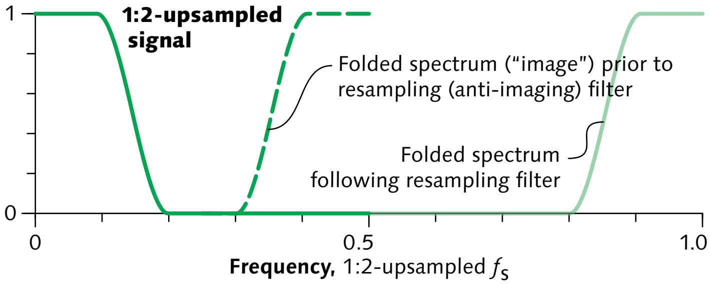


Figure 21.1 **Two-times upsampling** starts by interposing zero samples between original sample pairs. This would result in the folded spectral content of the original signal appearing in-band at the new rate. These "images" are removed by a resampling filter.

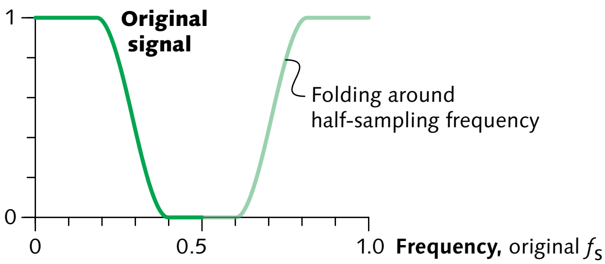


Figure 21.2 **An original signal** exhibits folding around half the sampling frequency. This is inconsequential providing that the signal is properly reconstructed. When the signal is upsampled or downsampled, the folded portion must be handled properly or aliasing will result.

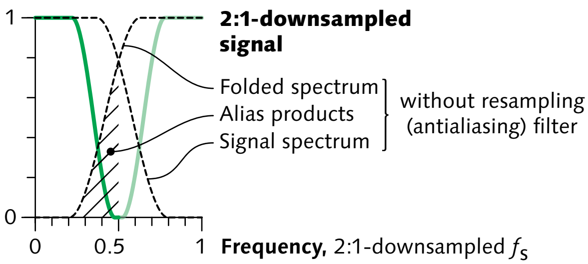


Figure 21.3 **Two-to-one downsampling** requires a resampling filter to meet the Nyquist criterion at the new sampling rate. The solid green line shows the spectrum of the filtered signal; the shaded line shows its folded portion. Resampling without filtering would preserve the original baseband spectrum, but folding around the new sampling rate would cause alias products shown here in the crosshatched region.

Figure 21.2, at the center above, sketches the spectrum of an original signal. Figure 21.1 shows the frequency domain considerations of upsampling; Figure 21.3 shows the frequency domain considerations of downsampling. These examples show ratios of 1:2 and 2:1; however, the concepts apply to resampling at any ratio.

2:1 downsampling

Colour video originates with *R'G'B'* components. Transcoding to $Y'C_BC_R$ is necessary if signals are to be used in the studio. The conversion involves matrixing (to $Y'C_BC_R$ in 4:4:4 form), then chroma subsampling to 4:2:2. Chroma subsampling requires a 2:1 downsampler. If this downsampling is attempted by simply dropping alternate samples, any signal content between the original $0.25f_S$ and $0.5f_S$ will cause aliasing in the result. Rejection of signal content at and above $0.25f_S$ is required. The required filter is usually implemented as an FIR lowpass filter having its corner frequency somewhat less than one-quarter of the (original) sampling frequency. After filtering, alternate result samples can be dropped. There is no need to calculate values that will subsequently be discarded, however! Efficient chroma subsamplers take advantage of that fact, interleaving the C_B and C_R components into a single filter.

In Figure 20.12, on page 203, I presented a very simple lowpass filter that simply averages two adjacent samples. That filter has a corner frequency of $0.25f_S$. However, it makes a slow transition from passband to stopband, and it has very poor attenuation in the stopband (above $0.25f_S$). It makes a poor resampling filter. More than two taps are required to give adequate performance in studio video subsampling.

In 4:2:2 video, chroma is cosited: Each chroma sample must be located at the site of a luma sample. A symmetrical filter having an even number of (nonzero) taps does not have this property. A downsampling filter for cosited chroma must have an odd number of taps.

Oversampling

I have explained the importance of prefiltering prior to A-to-D conversion, and of postfiltering following D-to-A conversion. Historically, these filters were implemented in the analog domain, using inductors and capacitors. In discrete form, these components are bulky and expensive. It is extremely difficult to incorporate inductive and capacitive elements with suitable values and precision onto integrated circuits. However, A-to-D and D-to-A converters are operating at higher and higher rates, and digital arithmetic has become very

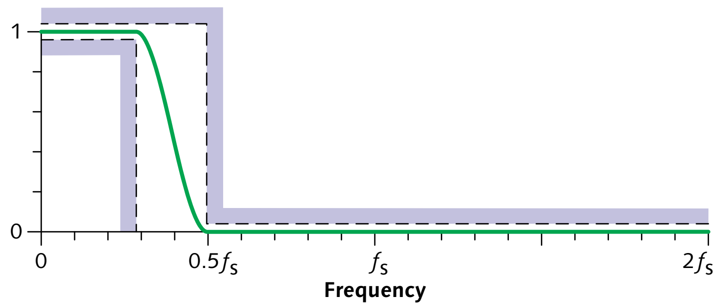


Figure 21.4 **An analog filter for direct sampling** must meet tight constraints, making it expensive.

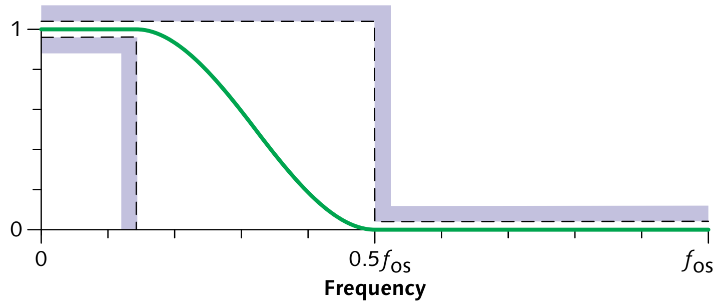


Figure 21.5 **An analog filter for 2×-oversampling** is much less demanding than a filter for direct sampling, because the difficult part of filtering – achieving a response comparable to that of Figure 21.4 – is relegated to the digital domain.

inexpensive. These circumstances have led to the emergence of *oversampling* as an economical alternative to complex analog presampling ("antialiasing") and post-sampling (reconstruction) filters.

The characteristics of a conventional analog presampling filter are critical: Attenuation must be quite low up to about 0.4 times the sample rate, and quite high above that. In a presampling filter for studio video, attenuation must be less than 1 dB or so up to about 5.5 MHz, and better than 40 or 50 dB above 6.75 MHz. This is a demanding transition ratio $\Delta\omega/\omega_S$. Figure 21.4 above (top) sketches the filter template of a conventional analog presampling filter.

For an explanation of transition ratio, see page 212.

An oversampling A-to-D converter operates at a multiple of the ultimate sampling rate – say at 27 MHz, twice the rate of BT.601 video. The converter is preceded by a cheap analog filter that severely attenuates components at 13.5 MHz and above. However, its characteristics between 5.5 MHz and 13.5 MHz are not critical. The demanding aspects of filtering in that region are left to a digital 2:1 downsampler. The transition ratio $\Delta\omega/\omega_S$ of the analog filter is greatly relaxed compared to direct conversion. In today's technology, the cost of the digital downsampler is less than the difference in cost between excellent and mediocre analog filtering. Complexity is moved from the analog domain to the digital domain; total system cost is reduced. Figure 21.5 (on page 225) sketches the template of an analog presampling filter appropriate for use preceding a 2x oversampled A-to-D converter.

In certain FIR filters whose corner is exactly $0.25 f_S$, half the coefficients are zero. This leads to a considerable reduction in complexity.

Figure 20.25, on page 215, showed the response of a 55-tap filter having a corner frequency of $0.25 f_S$. This is a *halfband* filter, intended for use following a 2×-oversampled A-to-D converter.

The approach to 2×-oversampled D-to-A conversion is comparable. The D-to-A device operates at 27 MHz; it is presented with a datastream that has been upsampled by a 1:2 ratio. For each input sample, the 2×-oversampling filter computes 2 output samples. One is computed at the effective location of the input sample, and the other is computed at an effective location halfway between input samples. The filter attenuates power between 6.75 MHz and 13.5 MHz. The analog postsampling filter need only reject components at and above 13.5 MHz. As in the 2×-oversampling A-to-D conversion, its performance between 6.75 MHz and 13.5 MHz isn't critical.

Interpolation

In the common case of interpolation horizontally across an image row, the argument *x* is horizontal position. Interpolating along the time axis, as in digital audio sample rate conversion, you could use the symbol *t* to represent time.

In mathematics, *interpolation* is the process of computing the value of a function or a putative function (call it $\tilde{g}$), for an arbitrary argument (x), given several function argument and value pairs $[x_i, s_i]$. There are many methods for interpolating, and many methods for constructing functions that interpolate.

Given two sample pairs $[x_0, s_0]$ and $[x_1, s_1]$, the linear interpolation function has this form:

$$\tilde{g}(x) = s_0 + \frac{x - x_0}{x_1 - x_0}(s_1 - s_0) \qquad \text{Eq 21.1}$$

In computer graphics, the linear interpolation operation is often called *LIRP* (pronounced *lerp*).

I symbolize the interpolating function as $\tilde{g}$; the symbol f is already taken to represent frequency. I write g with a tilde ($\tilde{g}$) to emphasize that it is an approximation.

The linear interpolation function can be rewritten as a weighted sum of the neighboring samples s_0 and s_1:

$$\tilde{g}(x) = c_0(x) \cdot s_0 + c_0(x) \cdot s_1 \qquad \text{Eq 21.2}$$

The weights depend upon the x (or t) coordinate:

$$c_0(x) = \frac{x_1 - x}{x_1 - x_0}; \qquad c_1(x) = \frac{x - x_0}{x_1 - x_0} \qquad \text{Eq 21.3}$$

Lagrange interpolation

J.L. Lagrange (1736–1813) developed a method of interpolation using polynomials. A cubic interpolation function is a polynomial of this form:

$$\tilde{g}(x) = ax^3 + bx^2 + cx + d \qquad \text{Eq 21.4}$$

Julius O. Smith calls this *Waring-Lagrange* interpolation, since Waring published it 16 years before Lagrange. See Smith's *Digital Audio Resampling Home Page*, <www-ccrma.stanford.edu/~jos/resample>.

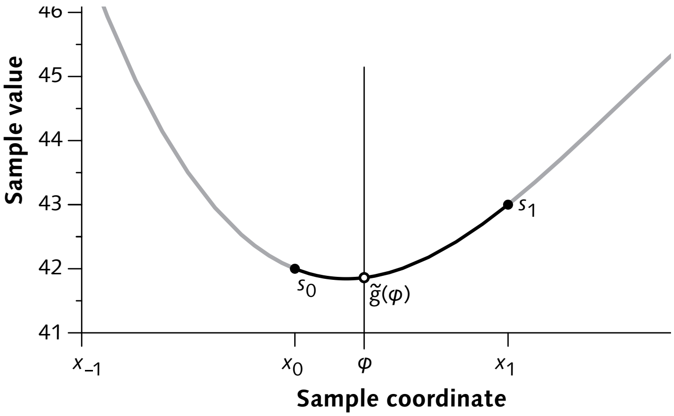
Interpolation involves choosing appropriate coefficients a, b, c, and d, based upon the given argument/value pairs $[x_j, s_j]$. Lagrange described a simple and elegant way of computing the coefficients.

Linear interpolation is just a special case of Lagrange interpolation of the first degree. (Directly using the value of the nearest neighbor can be considered zero-order interpolation.) There is a second-degree (quadratic) form; it is rarely used in signal processing.

In mathematics, to *interpolate* refers to the process that I have described. However, the same word is used to denote the property whereby an interpolating function produces values exactly equal to the original sample values (s_i) at the original sample coordinates (x_i). The Lagrange functions exhibit this property. You might guess that this property is a requirement of *any* interpolating function. However, in signal processing this is not a requirement – in fact, the interpolation functions used in video and audio rarely pass exactly through the original sample values. As a consequence of using the terminology of mathematics, in video we have the seemingly paradoxical situation that interpolation functions usually do not "interpolate"!

In principle, cubic interpolation could be undertaken for any argument x, even values outside the x-coordinate range of the four input samples. (Evaluation outside the interval $[x_{-1}, x_2]$ would be called *extrapolation*.) In digital video and audio, we limit x to the range between x_0 and x_1, so as to estimate the signal in the interval between the central two samples. To evaluate outside this interval, we substitute the input sample values $[s_{-1}, s_0, s_1, s_2]$ appropriately – for example, to

Figure 21.6 **Cubic interpolation** of a signal starts with equally spaced samples, in this example 47, 42, 43, and 46. The underlying function is estimated to be a cubic polynomial that passes through ("interpolates") all four samples. The polynomial is evaluated between the two central samples, as shown by the black segment. Here, evaluation is at phase offset φ. If the underlying function isn't a polynomial, small errors are produced.



evaluate between s_1 and s_2, we shift the input sample values left one place.

Eq 21.5

$$\varphi = \frac{x - x_0}{x_1 - x_0}; \qquad x_0 \le \varphi \le x_1$$

With uniform sampling (as in conventional digital video), when interpolating between the two central samples the argument x can be recast as the *phase offset*, or the *fractional phase* (φ, phi), at which a new sample is required between two central samples. (See Equation 21.5.) In abstract terms, φ lies between 0 and 1; in hardware, it is implemented as a binary or a rational fraction. In video, a 1-D interpolator is usually an FIR filter whose coefficients are functions of the phase offset. The weighting coefficients (c_i) are functions of the phase offset; they can be considered as basis functions.

UNSER, MICHAEL (1999), "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*: 22–38 (Nov.).

In signal processing, cubic (third-degree) interpolation is often used; the situation is sketched in Figure 21.6 above. In linear interpolation, one neighbor to the left and one to the right are needed. In cubic interpolation, we ordinarily interpolate in the central interval, using two original samples to the left and two to the right of the desired sample instant.

Equation 21.2 can be reformulated:

$$g(\varphi) = c_{-1}(\varphi) \cdot s_{-1} + c_0(\varphi) \cdot s_0 + c_1(\varphi) \cdot s_1 + c_2(\varphi) \cdot s_2 \qquad \text{Eq 21.6}$$

The function takes four sample values [s_{-1}, s_0, s_1, s_2] surrounding the interval of interest, and the phase offset φ between 0 and 1. The coefficients (c_i) are now functions of the argument φ; the interpolator forms a weighted sum of four sample values, where the weights are functions of the parameter φ; it returns an estimated value. (If the input samples are values of a polynomial not exceeding the third degree, then the values produced by a cubic Lagrange interpolator are exact, within roundoff error: Lagrange interpolation "interpolates"!)

If a 2-D image array is to be resampled at arbitrary x and y coordinate values, one approach is to apply a 1-D filter along one axis, then apply a 1-D filter along the other axis. This approach treats interpolation as a separable process, akin to the separable filtering that I will introduce on page 242. Surprisingly, this two-pass approach can be used to rotate an image; see Smith, cited in the margin. Alternatively, a 2×2 array (of 4 sample values) can be used for linear interpolation in 2 dimensions in one step – this is *bilinear* interpolation. A more sophisticated approach is to use a 4×4 array (of 16 sample values) as the basis for cubic interpolation in 2 dimensions – this is *bicubic* interpolation. (It is mathematically comparable to 15th-degree interpolation in one dimension.)

SMITH, ALVY RAY (1987), "Planar 2-pass texture mapping and warping," in *Computer Graphics* 21 (4): 12–19, 263–272 (July, *Proc. SIGGRAPH 87*).

Curves can be drawn in 2-space using a parameter u as the argument to each of two functions x(u) and y(u) that produce a 2-D coordinate pair for each value of u. Cubic polynomials can be used as x(u) and y(u). This approach can be extended to 3-space by adding a third function, z(u). Pierre Bézier developed a method, which is now widely used, to use cubic polynomials to describe curves and surfaces. Such curves are now known as *Bézier curves* or *Bézier splines*. The method is very important in the field of computer graphics; however, Bézier splines and their relatives are infrequently used in signal processing.

BARTELS, RICHARD H., JOHN C. BEATTY, and BRIAN A. BARSKY (1989), *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling* (San Francisco: Morgan Kaufmann).

Lagrange interpolation as filtering

Except for having 4 taps instead of 5, Equation 21.6 has identical form to the 5-tap Gaussian filter of Equation 20.2, on page 207! Lagrange interpolation can be viewed as a special case of FIR filtering, and can be analyzed as a filtering operation. In the previous chapter, *Filtering and sampling*, all of the examples were symmetric. Interpolation to produce samples exactly halfway between input samples, such as in a 2×-over-

Only symmetric FIR filters exhibit true linear phase. Other FIR filters exhibit very nearly linear phase, close enough to be considered to have linear phase in video and audio.

sampling DAC, is also symmetric. However, most interpolators are asymmetric.

There are four reasons why polynomial interpolation is generally unsuitable for video signals: Polynomial interpolation has unequal stopband ripple; nulls lie at fixed positions in the stopband; the interpolating function exhibits extreme behavior outside the central interval; and signals presented to the interpolator are somewhat noisy. I will address each of these issues in turn.

- Any Lagrange interpolator has a frequency response with unequal stopband ripple, sometimes highly unequal. That is generally undesirable in signal processing, and it is certainly undesirable in video.
- A Lagrange interpolator "interpolates" the original samples; this causes a magnitude frequency response that has periodic nulls ("zeros") whose frequencies are fixed by the order of the interpolator. In order for a filter designer to control stopband attenuation, he or she needs the freedom to place nulls judiciously. This freedom is not available in the design of a Lagrange interpolator.
- Conceptually, interpolation attempts to model, with a relatively simple function, the unknown function that generated the samples. The form of the function that we use should reflect the process that underlies generation of the signal. A cubic polynomial may deliver sensible interpolated values between the two central points. However, the value of any polynomial rapidly shoots off to plus or minus infinity at arguments outside the region where it is constrained by the original sample values. That property is at odds with the behavior of signals, which are constrained to lie within a limited range of values forever (say the abstract range 0 to 1 in video, or ±0.5 in audio).
- In signal processing, there is always some uncertainty in the sample values caused by noise accompanying the signal, quantization noise, and noise due to roundoff error in the calculations in the digital domain. When the source data is imperfect, it seems unreasonable to demand perfection of an interpolation function.

These four issues are addressed in signal processing by using interpolation functions that are not polynomials and that do not come from classical mathematics.

Instead, we usually use interpolation functions based upon the the sinc weighting function that I introduced on page 198. In signal processing, we usually design interpolators that do not "interpolate" the original sample values.

You can consider the entire stopband of an ideal sinc filter to contain an infinity of nulls. Mathematically, the sinc function represents the limit of Lagrange interpolation as the order of the polynomial approaches infinity. See Appendix A of Smith's *Digital Audio Resampling Home Page*, cited in the margin of page 227.

The ideal sinc weighting function has no distinct nulls in its frequency spectrum. When sinc is truncated and optimized to obtain a physically realizable filter, the stopband has a finite number of nulls. Unlike a Lagrange interpolator, these nulls do not have to be regularly spaced. It is the filter designer's ability to choose the frequencies for the zeros that allows him or her to tailor the filter's response.

Polyphase interpolators

The 720*p*60 and 1080*i*30 standards have an identical sampling rate (74.25 MHz). In the logic design of this example, there is a single clock domain.

Some video signal processing applications require upsampling at simple ratios. For example, conversion from 1280 S_{AL} to 1920 S_{AL} in an HD format converter requires 2:3 upsampling. An output sample is computed at one of three phases: either at the site of an input sample, or $1/3$ or $2/3$ of the way between input samples. The upsampler can be implemented as an FIR filter with just three sets of coefficients; the coefficients can be accessed from a lookup table addressed by φ.

Many interpolators involve ratios more complex than the 2:3 ratio of this example. For example, in conversion from $4f_{SC}$ NTSC to BT.601 (4:2:2), 910 input samples must be converted to 858 results. This involves a downsampling ratio of 35:33. Successive output samples are computed at an increment of $1\,2/33$ input samples. Every 33rd output sample is computed at the site of an input sample (0); other output samples are computed at input sample coordinates $1\,2/33$, $2\,4/33$, …, $16\,32/33$, $18\,1/33$, $19\,3/33$, …, $34\,31/33$. Addressing circuitry needs to increment a sample counter by one, and a fractional numerator by 2 modulo 33 (yielding the fraction $2/33$), at each output sample. Overflow from the fraction counter carries into the sample counter; this accounts for the missing input sample number 17 in the sample number sequence of this example. The required interpolation phases are at fractions $\varphi = 0$, $1/33$, $2/33$, $3/33$, …, $32/33$ between input samples.

In the logic design of this example, two clock domains are involved.

A straightforward approach to design of this interpolator in hardware is to drive an FIR filter at the input sample rate. At each input clock, the input sample values shift across the registers. Addressing circuitry implements a modulo-33 counter to keep track of phase – a *phase accumulator*. At each clock, one of 33 different sets of coefficients is applied to the filter. Each coefficient set is designed to introduce the appropriate phase shift. In this example, only 33 result samples are required every 35 input clocks: During 2 clocks of every 35, no result is produced.

This structure is called a *polyphase filter*. This example involves 33 phases; however, the number of *taps* required is independent of the number of *phases*. A 2×-oversampled prefilter, such I described on page 224, has just two phases. The halfband filter whose response is graphed in Figure 20.25, on page 215, would be suitable for this application; that filter has 55 taps.

Polyphase taps and phases

The number of *taps* required in a filter is determined by the degree of control that the designer needs to exercise over frequency response, and by how tightly the filters in each phase need to match each other. In many cases of consumer-grade video, cubic (4-tap) interpolation is sufficient. In studio video, 8 taps or more might be necessary, depending upon the performance to be achieved.

In a direct implementation of a polyphase FIR interpolator, the number of *phases* is determined by the arithmetic that relates the sampling rates. The number of phases determines the number of coefficient sets that need to be used. Coefficient sets are typically precomputed and stored in nonvolatile memory.

On page 231, I described a polyphase resampler having 33 phases. In some applications, the number of phases is impractically large to implement directly. This is the case for the 709379:540000 ratio required to convert from $4f_{SC}$ PAL to BT.601 (4:2:2), from about 922 active samples per line to about 702. In other applications, such as digital video effects, the number of phases is variable, and unknown in advance. Applications such as these can be addressed by an interpolator having a number of phases that is a suitable power of two, such as 256 phases. Phase offsets are computed to the appropriate degree of precision, and are then approximated to a binary fraction (in this case having 8 bits) to form the phase offset φ that is presented to the interpolator.

$$\frac{1}{512} = \frac{1}{2} \cdot \frac{1}{2^8}$$

If the interpolator implements 8 fractional bits of phase, then any computed output sample may exhibit a positional error of up to $\pm \frac{1}{512}$ of a sample interval. This is quite acceptable for component digital video. However, if the phase accumulator implements just 8 fractional bits, that positional error will accumulate as the incremental computation proceeds across the image row. In this example, with 922 active samples per line, the error could reach three or four sample intervals at the right-hand end of the line! This isn't tolerable. The solution is to choose a sufficient number of fractional bits in the phase accumulator to keep the cumulative error within limits. In this example, 13 bits are sufficient, but only 8 of those bits need to be presented to the interpolator.

Implementing polyphase interpolators

Polyphase interpolation is a specialization of FIR filtering; however, there are three major implementation differences. First, in a typical FIR filter, the input and output rates are the same; in a polyphase interpolator, the input and output rates are usually different. Second, FIR filters usually have fixed coefficients; in a polyphase FIR interpolator, the coefficients vary on a sample-by-sample basis. Third, typical FIR filters are symmetrical, but polyphase interpolators are not.

Generally speaking, for a small number of phases – perhaps 8 or fewer – the cost of an interpolator is dominated by the number of multiplication operations, which is proportional to the number of taps. Beyond about 8 taps, the cost of coefficient storage begins to be significant. The cost of the addressing circuitry depends only upon the number of phases.

In the 35:33 downsampler example, I discussed a hardware structure driven by the input sample rate. Suppose the hardware design requires that the interpolator be driven by the output clock. For 31 of each 33 output clocks, one input sample is consumed; however, for two clocks, two input samples are consumed. This places a constraint on memory system design: Either two paths from memory must be implemented, or the extra 44 samples per line must be accessed during the blanking interval, and be stored in a small buffer. It is easier to drive this interpolator from the input clock.

Consider a 33:35 upsampler, from BT.601 to $4f_{SC}$ NTSC. If driven from the output side, the interpolator produces one output sample per clock, and consumes at most one input sample per clock. (For 2 of the 35 output clocks, no input samples are consumed.) If driven from the input side, for 2 of the 33 input clocks, the interpolator must produce two output samples. This is likely to present problems to the design of the FIR filter and the output side memory system.

The lesson is this: The structure of a polyphase interpolator is simplified if it is driven from the high-rate side.

Decimation

In Lagrange interpolation, no account is taken of whether interpolation computes more or fewer output samples than input samples. However, in signal processing, there is a big difference between downsampling – where lowpass filtering is necessary to prevent aliasing – and upsampling, where lowpass filtering is necessary to suppress "imaging." In signal processing, the term *interpolation* generally implies upsampling, that is, resampling to any ratio of unity or greater. (The term *interpolation* also describes phase shift without sample rate change; think of this as the special case of upsampling with a ratio of 1:1.)

Taken literally, *decimation* involves a ratio of 10:9, not 10:1.

Downsampling with a ratio of 10:9 is analogous to the policy by which the Roman army dealt with treachery and mutiny among its soldiers: One in ten of the offending soldiers was put to death. Their term *decimation* has come to describe downsampling in general.

Lowpass filtering in decimation

Earlier in this chapter, I expressed chroma subsampling as 2:1 decimation. In a decimator, samples are lowpass filtered to attenuate components at and above half the *new* sampling rate; then samples are dropped. Obviously, samples that are about to be dropped need not be computed! Ordinarily, the sample-dropping and filtering are incorporated into the same circuit.

For details of interpolators and decimators, see CROCHIERE, RONALD E., and LAWRENCE R. RABINER (1983), *Multirate Digital Signal Processing* (New York: Prentice-Hall).

In the example of halfband decimation for chroma subsampling, I explained the necessity of lowpass filtering to $0.25f_S$. In the $4f_{SC}$ NTSC to BT.601 example that I presented in *Polyphase interpolators,* on page 231, the input and output sample rates were so similar that no special attention needed to be paid to bandlimiting at the resulting sample rate. If the downsampling ratio is much greater than unity – say 5:4, or greater – then the impulse response must incorporate a lowpass filtering (prefiltering, or antialiasing) function as well as phase shift. To avoid aliasing, the lowpass corner frequency must scale with the downsampling ratio. This may necessitate several sets of filter coefficients having different corner frequencies.

This page intentionally left blank

Image digitization and reconstruction 22

A sequence of still pictures captured and displayed at a sufficiently high rate – typically between 24 and 60 pictures per second – can create the illusion of motion, as I will describe further on page 51. Sampling in time, in combination with 2-D (spatial) sampling, causes digital video to be sampled in three axes – horizontal, vertical, and temporal – as sketched in Figure 22.1.

One-dimensional sampling theory was described in *Filtering and sampling,* on page 191.

One-dimensional sampling theory applies along each of the three axes. I sketch just three temporal samples, because temporal sample count is limited by the number of picture stores provided; picture stores are more expensive than linestores. I sketch five vertical samples: Each vertical sample is associated with a linestore.

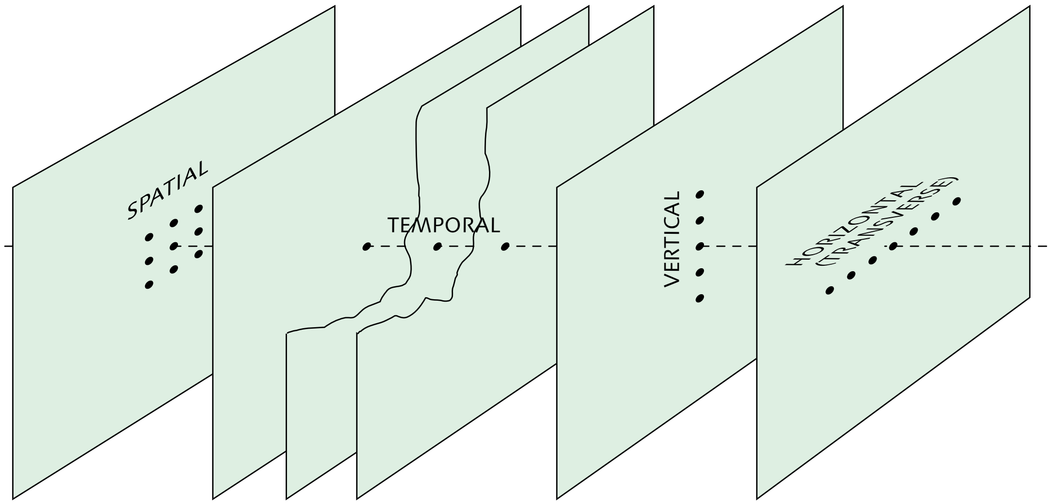


Figure 22.1 **Spatiotemporal domains**

At the far left of Figure 22.1 is a sketch of a two-dimensional *spatial* domain of a single image. Some image processing operations, such as certain kinds of filtering, can be performed separately on the horizontal and vertical axes, and have an effect in the spatial domain – these operations are called *separable*. Other processing operations cannot be separated into horizontal and vertical facets, and must be performed directly on a two-dimensional sample array. Two-dimensional sampling theory applies.

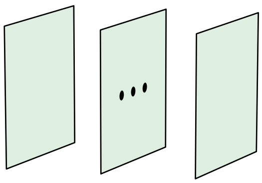


Figure 22.2 **Horizontal domain**

In Chapter 20, *Filtering and sampling,* on page 191, I described how to analyze a signal that is a function of the single dimension of time, such as an audio signal. Sampling theory also applies to a signal that is a function of one dimension of space, such as a single scan line (image row) of a video signal. This is the horizontal or *transverse* domain, sketched in Figure 22.2 in the margin. If an image is scanned line by line, the waveform of each line can be treated as an independent signal. The techniques of filtering and sampling in one dimension, discussed in Chapter 20, apply directly to this case.

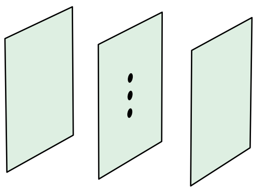


Figure 22.3 **Vertical domain**

Consider a set of points arranged vertically that originate at the same displacement along each of several successive image rows, as sketched in Figure 22.3. Those points can be considered to be sampled by the scanning process itself. Sampling theory can be used to understand the properties of these samples.

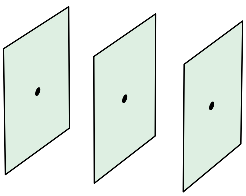


Figure 22.4 **Temporal domain**

A third dimension is introduced when a succession of images is temporally sampled to represent motion. Figure 22.4 depicts samples in the same column and the same row in three successive frames.

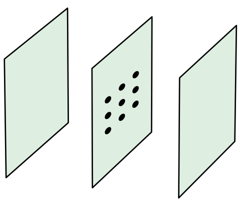


Figure 22.5 **Spatial domain**

Complex filters can act on two axes simultaneously. Figure 22.5 illustrates spatial sampling. The properties of the entire set of samples are considered all at once, and cannot necessarily be separated into independent horizontal and vertical aspects.

Spatial frequency domain

I explained in *Image structure,* on page 75, how a one-dimensional waveform in time transforms to a one-dimensional frequency spectrum. This concept can be extended to two dimensions: The two dimensions of space can be transformed into two-dimensional spatial

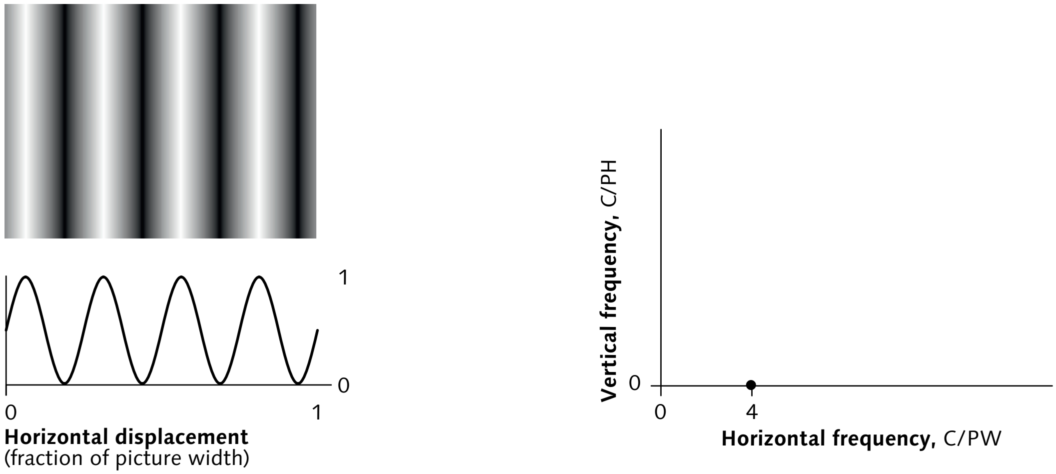


Figure 22.6 **Horizontal spatial frequency domain**

frequency. The content of an image can be expressed as horizontal and vertical spatial frequency components. Spatial frequency is plotted using cycles per picture width (C/PW) as an *x*-coordinate, and cycles per picture height (C/PH) as a *y*-coordinate. You can gain insight into the operation of an imaging system by exploring its spatial frequency response.

In the image at the top left of Figure 22.6 above, every image row has identical content: 4 cycles of a sine wave. Underneath the image, I sketch the time domain waveform of every line. Since every line is identical, no power is present in the vertical direction. Considered in the spatial domain, this image contains power at a single horizontal spatial frequency, 4 C/PW; there is no power at any vertical spatial frequency. All of the power of this image lies at spatial frequency [4, 0].

Figure 22.7 overleaf shows an image comprising a sine wave signal in the vertical direction. The height of the picture contains 3 cycles. The spatial frequency graph, to the right, shows that all of the power of the image is contained at coordinates [0, 3] of spatial frequency. In an image where each image row takes a constant value, all of the power is located on the *y*-axis of spatial frequency.

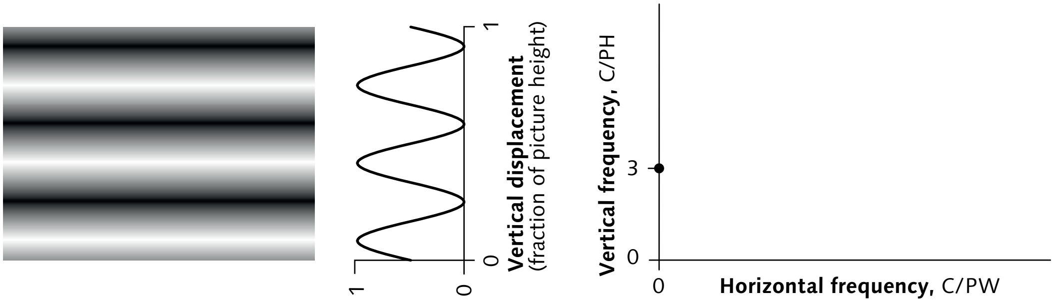


Figure 22.7 **Vertical spatial frequency domain**

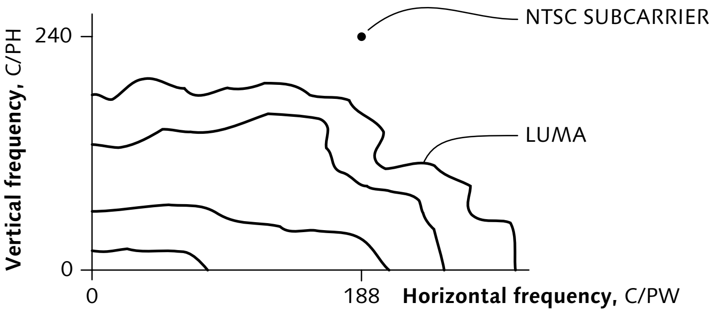


Figure 22.8 **The spatial frequency spectrum of 480*i* luma** is depicted in this plot, which resembles a topographical map. If the unmodulated NTSC subcarrier were included in image data, it would take the indicated position.

If an image comprises rows with identical content, all of the power will be concentrated on the horizontal axis of spatial frequency. If the content of successive scans lines varies slightly, the power will spread to nonzero vertical frequencies. An image of diagonal bars would occupy a single point in spatial frequency, displaced from the *x*-axis and displaced from the *y*-axis.

When spatial frequency is determined analytically using the two-dimensional Fourier transform, the result is plotted in the manner of Figure 22.8, where low vertical frequencies – that is, low *y* values – are at the bottom. When spatial frequency is computed numerically using discrete transforms, such as the 2-D *discrete Fourier transform* (DFT), the *fast Fourier transform* (FFT), or the *discrete cosine transform* (DCT), the result is usually presented in a matrix, where low vertical frequencies are at the top.

The spatial frequency that corresponds to half the vertical sampling rate depends on the number of picture lines. A 480*i* system has approximately 480 picture lines: 480 samples occupy the height of the picture, and the Nyquist frequency for vertical sampling is 240 C/PH. No vertical frequency in excess of this can be represented without aliasing.

In most images, successive rows and columns of samples (of *R'*, *G'*, *B'*, or of luma) are very similar; low frequencies predominate, and image power tends to cluster toward spatial frequency coordinates [0, 0]. Figure 22.8 sketches the spatial frequency spectrum of luma in a 480*i* system. If the unmodulated NTSC colour subcarrier were an image data signal, it would take the indicated location. In composite NTSC, chroma is modulated onto the subcarrier; the resulting modulated chroma can be thought of as occupying a particular region of the spatial frequency plane, as described in *Spatial frequency spectra of NTSC,* in Chapter 6 of *Composite NTSC and PAL: Legacy Video Systems*. In NTSC encoding, modulated chroma is then summed with luma; this causes the spectra to be overlaid. If the luma and chroma spectra overlap, cross-colour and cross-luma interference artifacts can result.

An *optical transfer function* (OTF) includes phase. The magnitude of an OTF is MTF; MTF disregards phase.

In optics, the terms *magnitude frequency response* and *bandwidth* are not used. An optical component, subsystem, or system is characterized by its *modulation transfer function* (MTF), a one-dimensional plot of horizontal or vertical spatial frequency response. (*Depth of modulation* is a single point quoted from this graph.) Technically, the MTF is the Fourier transform of the point spread function (PSF) or line spread function (LSF). By definition, the MTF relates to light intensity. Since negative light power is physically unrealizable, an MTF is measured by superimposing a high-frequency sinusoidal (modulating) wave onto a constant level, then taking the ratio of output modulation to input modulation.

Comb filtering

In *Finite impulse response (FIR) filters,* on page 207, I described FIR filters operating in the single dimension of time. If the samples are from a scan line of an image, the frequency response can be considered to represent horizontal spatial frequency (in units of C/PW), instead of temporal frequency (in cycles per second, or hertz).

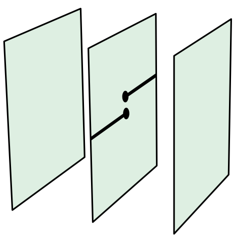


Figure 22.9 **Two samples, vertically arranged**

Consider a sample from a digital image sequence, and the sample immediately below, as sketched in Figure 22.9 in the margin. If the image has 640 active (picture) samples per line, and these two samples are presented to a comb filter like that of Figure 20.19, on page 206, but having 639 zero-samples between the two “ones,” then the action of the comb filter will be identical to the action of a filter having two taps weighted [1, 1] operating in the vertical direction. In Figure 20.12, on page 203, I graphed the frequency response of a one-dimensional [1, 1] filter. The graph in

Figure 22.10 **The response of a [1, 1] FIR filter** operating in the vertical domain, scaled for unity gain, is shown. This is a two-line (1*H*) comb filter. Magnitude falls as cos *ω*.

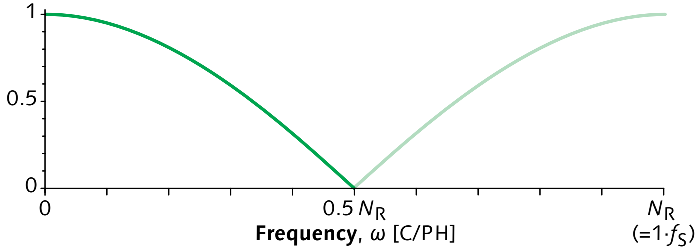


Figure 22.10 shows the response of the comb filter, expressed in terms of its response in the vertical direction. Here magnitude response is shown normalized for unity gain at DC; the filter has a response of about 0.707 (i.e., it is 3 db down) at one-quarter the vertical sampling frequency.

Spatial filtering

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 22.11 **Separable spatial filter examples**

Placing a [1, 1] horizontal lowpass filter in tandem (cascade) with a [1, 1] vertical lowpass filter is equivalent to computing a weighted sum of spatial samples using the weights indicated in the matrix on the left in Figure 22.11. Placing a [1, 2, 1] horizontal lowpass filter in tandem with a [1, 2, 1] vertical lowpass filter is equivalent to computing a weighted sum of spatial samples using the weights indicated in the matrix on the right in Figure 22.11. These are examples of *spatial filters*. These particular spatial filters are *separable:* They can be implemented using horizontal and vertical filters in tandem.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 22.12 **Inseparable spatial filter examples**

Many spatial filters are *inseparable:* Their computation must take place directly in the two-dimensional spatial domain; they cannot be implemented using cascaded one-dimensional horizontal and vertical filters. Examples of inseparable filters are given in the matrices in Figure 22.12.

Image presampling filters

In a video camera, continuous information must be subjected to a presampling ("antialiasing") filter. Aliasing is minimized by optical spatial lowpass filtering that is effected in the optical path, prior to conversion of the image signal to electronic form. MTF limitations in the lens impose some degree of filtering. An additional filter can be implemented as a discrete optical element (often employing the optical property of birefringence). Additionally, or alternatively, some degree of filtering may be imposed by optical properties of the photosensor itself.

In resampling, signal power is not constrained to remain positive; filters having negative weights can be used. The ILPF (see page 198) and other sinc-based filters have negative weights, but those filters often ring and exhibit poor visual performance. Schreiber and Troxel found well-designed sharpened Gaussian filters with $\sigma = 0.375$ to have superior performance to the ILPF. A filter that is optimized for a particular mathematical criterion does not necessarily produce the best-looking picture!

Schreiber, William F., and Donald E. Troxel (1985), "Transformations between continuous and discrete representations of images: A perceptual approach," in *IEEE Tr. on Pattern Analysis and Machine Intelligence* PAMI-7 (2): 178–186 (Mar.).

Image reconstruction filters

On page 76, I introduced "box filter" reconstruction. This is technically known as *sample-and-hold, zero-order hold,* or *nearest-neighbor* reconstruction.

In theory, ideal image reconstruction would be obtained by using a PSF which has a two-dimensional sinc distribution. This would be a two-dimensional version of the ILPF that I described for one dimension on page 198. However, a sinc function involves negative excursions. Light power cannot be negative, so a sinc filter cannot be used for presampling at an image capture device, and cannot be used as a reconstruction filter at a display device. A box-shaped distribution of sensitivity across each element of a sensor is easily implemented, as is a box-shaped distribution of intensity across each pixel of a display. However, like the one-dimensional boxcar of Chapter 20, a box distribution has significant response at high frequencies. Used at a sensor, a box filter will permit aliasing. Used in a display, scan-line or pixel structure is likely to be visible. If an external optical element such as a lens attenuates high spatial frequencies, then a box distribution might be suitable. A simple and practical choice for either capture or reconstruction is a Gaussian having a judiciously chosen half-power width. A Gaussian is a compromise that can achieve reasonably high resolution while minimizing aliasing and minimizing the visibility of the pixel (or scan-line) structure.

A *raised cosine* distribution is roughly similar to a Gaussian. See page 542.

Schreiber and Troxel suggest reconstruction with a sharpened Gaussian having $\sigma = 0.3$. See their paper cited in the marginal note above.

Spatial (2-D) oversampling

In image capture, as in reconstruction for image display, ideal theoretical performance would be obtained by using a PSF with a sinc distribution. However, a sinc function cannot be used directly in a transducer of light, because light power cannot be negative: Negative weights cannot be implemented. As in display reconstruction, a simple and practical choice for a direct presampling or reconstruction filter is a Gaussian having a judiciously chosen half-power width.

I have been describing direct sensors, where samples are taken directly from sensor elements, and direct displays, where samples directly energize display elements. In *Oversampling,* on page 224, I described a technique whereby a large number of directly acquired samples can be filtered to a lower sampling rate. That section discussed downsampling in one dimension, with the main goal of reducing the complexity of analog presampling or reconstruction filters. The oversampling technique can also be applied in two dimensions: A sensor can directly acquire a fairly large number of samples using a crude optical presampling filter, then use a sophisticated digital spatial filter to downsample.

The advantage of interlace – reducing scan-line visibility for a given bandwidth, spatial resolution, and flicker rate – is built upon the assumption that the sensor (camera), data transmission, and display all use identical scanning. If oversampling is feasible, the situation changes. Consider a receiver that accepts progressive image data (as in the top left of Figure 8.8, on page 91), but instead of displaying this data directly, it synthesizes data for a larger image array (as in the middle left of Figure 8.8). The synthetic data can be displayed with a spot size appropriate for the larger array, and all of the scan lines can be illuminated in each $^1/_{60}$ s instead of just half of them. This technique is *spatial oversampling* or *upsampling.* For a given level of scan-line visibility, this technique enables closer viewing distance than would be possible for progressive display.

Oversampling to double the number of lines displayed during a frame time is called *line doubling.*

Oversampling provides a mechanism for a sensor PSF or a display PSF to have negative weights, yielding a spatially "sharpened" filter. For example, a sharpened Gaussian PSF (such as anticipated by Schreiber 25 years ago) can be obtained, and can achieve performance better than a Gaussian. With a sufficient degree of oversampling, using sophisticated filters having sinc-like PSFs, the interchange signal can come arbitrarily close to the Nyquist limit. However, mathematical excellence does not necessarily translate to improved visual performance. Sharp filters are liable to ring, and thereby produce objectionable artifacts.

If negative weights are permitted in a PSF, then negative signal values can potentially result. Standard studio digital interfaces provide footroom that enables conveying moderate undershoot or overshoot. Using negative weights typically improves filter performance even if negative values are clipped after downsampling.

Similarly, if a display has many elements for each digital sample, a sophisticated digital upsampler can use negative weights. Negative values resulting from the filter's operation will eventually be clipped at the display itself, but again, improved performance could result.

If oversampling had been technologically feasible in 1941, or in 1953, then the NTSC would have undoubtedly chosen a progressive transmission standard. However, oversampling was not economical for SD studio systems until about 2005, when HD production became so prevalent that HD was in essence the oversampled studio standard for SDTV. Oversampling at consumer displays was not economical until about 2005. So, until about 2005, interlace retained an economic advantage both in the studio and in consumers' premises. However, in my view this advantage has now eroded, and it is likely that all future video system standards will have progressive scanning.

Perception and visual acuity 23

Properties of human vision are central to image system engineering. They determine how many pixels need to be provided per degree of picture angle, and how many bits are necessary to represent luminance (or tristimulus) levels. This chapter introduces the luminance discrimination and spatial properties of vision that inform image system engineering choices.

Retina

The human retina has four types of photoreceptor cells that respond to incident radiation with different spectral response curves. A retina has about 100 million *rod* cells, effective only at extremely low light levels; and about 5 million *cone* cells, of three types, that mediate colour vision. Since there is only one type of rod cell, what is loosely called *night vision* cannot discern colours.

BOYNTON, ROBERT M. (1979), *Human Color Vision* (New York: Holt, Rinehart and Winston).

WANDELL, BRIAN A. (1995), *Foundations of Vision* (Sunderland, Mass.: Sinauer Associates).

The cone cells are sensitive to longwave, mediumwave, and shortwave light – roughly, light in the red, green, and blue portions of the spectrum. Because there are just three types of colour photoreceptors, three numerical components are necessary and sufficient to describe colour: Colour vision is inherently *trichromatic*. To arrange for three components to mimic colour vision, suitable spectral sensitivity functions must be used; this topic will be discussed in *The CIE system of colorimetry,* on page 265.

Adaptation

Vision operates over a remarkable range of luminance levels – about eight orders of magnitude (decades), sketched in Figure 23.1. For about four decades at the low end of the range, the rods are active; vision at these light levels is called *scotopic*. For the top five or six decades, the cones are active; vision at these light levels is called *photopic*.

Mesopic vision takes place in the range of luminance levels where there is some overlap between rods and cones. Considered from the bottom of the photopic region, this is called *rod intrusion.* It is a research topic whether the rods have significance to colour image reproduction at usual luminance levels (such as in the cinema). For today's engineering purposes, the effect of rod intrusion is discounted.

During the course of the day we experience a wide range of illumination levels; adaptation adjusts accordingly, as sketched in Figure 23.2. From moonlight to sunlight, illuminance changes by a factor of about 200,000; adaptation causes the sensitivity of the visual system to reduce by about a factor of 1000. About one decade of adaptation is effected by the eye's iris – that is, by changes in pupil diameter (from about 2 mm to 8 mm). The main mechanism of adaptation is a photochemical process involving the *visual pigment* substance contained in the rods and the cones; it also involves neural mechanisms in the visual pathway.

Dark adaptation, to low luminance, is slow: Adaptation from a bright sunlit day to the darkness of a cinema can take a few minutes. Adaptation to higher luminance is rapid but can be discomforting, as you may have experienced when walking out of the cinema back into daylight.

Adaptation is a low-level phenomenon within the visual system; it is mainly controlled by total retinal illumination. Your adaptation state is closely related to the mean luminance in your field of view. In a dark viewing environment, such as a cinema, the image itself controls adaptation.

At a particular state of adaptation, vision can discern different luminances across about a 1000:1 range. When viewing a real scene, adaptation changes depending upon where in the scene your gaze is directed. In video and film, we are nearly always concerned with viewing at a known adaptation state, so a simultaneous contrast ratio of 1000:1 is adequate.

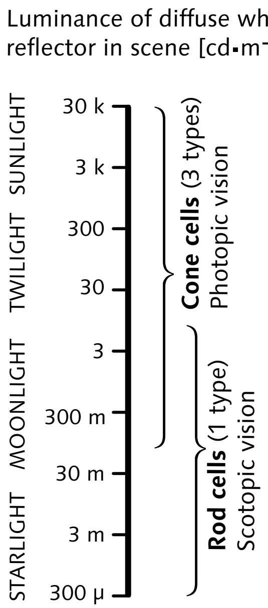


Figure 23.1 **Luminance range of vision**

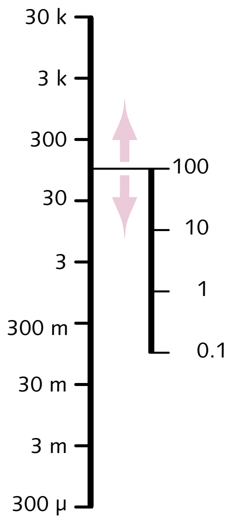


Figure 23.2 **Adaptation**

Diffuse white was described on page 117. This wide range of luminance levels is sometimes called *dynamic range*, but nothing is in motion!

For image reproduction purposes, our ability to distinguish luminance differences ordinarily extends over a ratio of luminance of about three decades – 10^3, or 1000:1 – that is, down to about 0.1% of diffuse white as portrayed on the display. Loosely speaking, luminance levels less than 0.1% of diffuse white appear just "black": Different luminances below that level are not ordinarily visually useful. Emergent *high dynamic range* (HDR) systems may increase that ratio.

Contrast sensitivity

Within the two-decade range of luminance that is useful for image reproduction, vision has a certain threshold of discrimination. It is convenient to express the discrimination capability in terms of *contrast threshold*, which is the ratio of a small test increment in luminance to the base luminance in a test stimulus having two adjacent patches of similar luminance.

Figure 23.3 below shows the pattern presented to an observer in an experiment to determine the contrast sensitivity of human vision. Most of the observer's field of vision is filled by a background luminance level, L_B, which fixes the observer's state of adaptation. In the central area of the field of vision are placed two adjacent patches having slightly different luminance levels, L and $L + \Delta L$. The experimenter presents stimuli having a wide range of test values with respect to the surround, that is, a wide range of L/L_B values. At each test luminance, the experimenter presents to the observer a range of luminance increments with respect to the test stimulus, that is, a range of $\Delta L/L$ values.

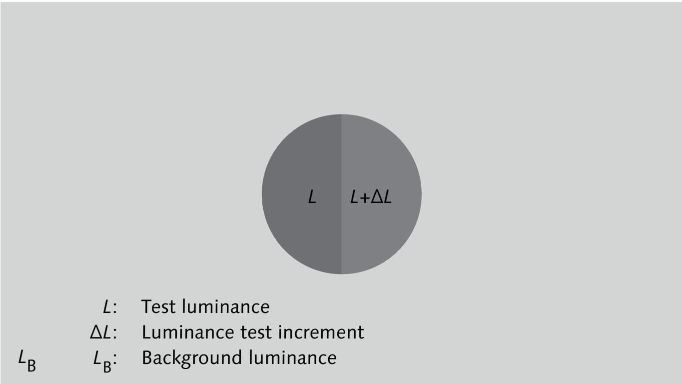


Figure 23.3 **A contrast sensitivity test pattern** is presented to an observer in an experiment to determine the contrast sensitivity of human vision. The observer is adapted to background having luminance L_B; a bipartite patch is viewed. The experimenter adjusts ΔL; the observer reports whether he or she detects a difference in lightness between the two patches.

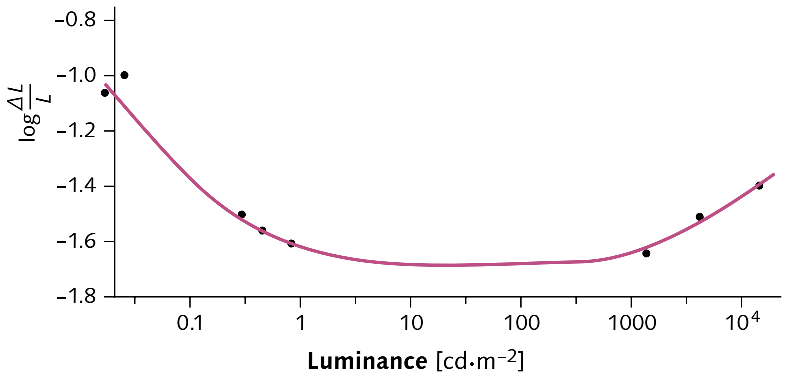


Figure 23.4 **Contrast sensitivity.** This graph is redrawn from Figure 3.4 of Schreiber's *Fundamentals of Electronic Imaging Systems*. Over a range of luminance values of about 300:1, the discrimination threshold of vision is approximately a constant ratio of luminance. The flat portion of the curve shows that the perceptual response to luminance – termed *lightness* – is approximately logarithmic. At very low luminance values, the curve departs from logarithmic behaviour and approximates a square-root; this characteristic is called the *de Vries-Rose law*.

Schreiber, William F. (1993), *Fundamentals of Electronic Imaging Systems*, Third Edition (Berlin: Springer-Verlag).

When this experiment is conducted, the relationship graphed in Figure 23.4 above is found: Plotting log $^{\Delta L}/_{L}$ as a function of log L reveals an interval of a few decades of luminance over which the discrimination capability of vision is about 1% of the test luminance level. This experiment leads to the conclusion that – for *threshold* discrimination of two adjacent patches of nearly identical luminance – the discrimination capability is roughly logarithmic.

The contrast sensitivity function begins to answer this question: What is the minimum number of discrete codes required to represent relative luminance over a particular range? In other words, what luminance codes can be thrown away without the observer noticing? On a linear luminance scale, to cover a 100:1 range with an increment of 0.01 takes $^{100}/_{0.01}$, or about 10,000 codes, requiring about 14 bits. If codes are spaced according to a *ratio* of 1.01, then only about 463 codes are required; codes can be represented in just 9 bits.(NTSC documents from the early 1950s used a contrast sensitivity of 2% and a contrast ratio of 30:1 to derive 172 steps; even today, 8 bits suffice for video distribution.)

$$\frac{\log 100}{\log 1.01} \approx 463; \quad 1.01^{463} \approx 100$$

$$\frac{\log 30}{\log 1.02} = 172$$

Fink, Donald G., ed. (1955), *Color Television Standards* (New York: McGraw-Hill): 201.

The logarithmic relationship relates to contrast sensitivity *at threshold:* We are measuring the ability of the visual system to discriminate between two nearly identical luminances. If you like, call this a *just noticeable* *difference* (JND), defined where the difference between two stimuli is detected as often as it is undetected.

Logarithmic coding rests on the assumption that the threshold function can be extended to large luminance ratios. Experiments have shown that this assumption does not hold very well. At a given state of adaptation, the discrimination capability of vision degrades at low luminances, below several percent of diffuse white. Over a wider range of luminance, strict adherence to logarithmic coding is not justified for perceptual reasons. Coding based upon a power law is found to be a better approximation to lightness response than a logarithmic function. In video, and in computing, power functions are used instead of logarithmic functions. Incidentally, other senses behave according to power functions, as shown in Table 23.1.

Stevens, Stanley S. (1975), *Psychophysics* (New York: Wiley).

Contrast sensitivity function (CSF)

Van Nes, Floris L., and Bouman, Maarten A. (1967), "Spatial modulation transfer in the human eye," in *J. Opt. Soc. Am.* **57** (3): 401–406.

The contrast sensitivity of vision is about 1% – that is, vision cannot distinguish two luminance levels if the ratio between them is less than about 1.01. That threshold applies to visual features of a certain angular extent, about ⅛°, for which vision has maximum ability to detect luminance differences. However, the contrast sensitivity of vision degrades for elements having angular subtense smaller or larger than about ⅛°.

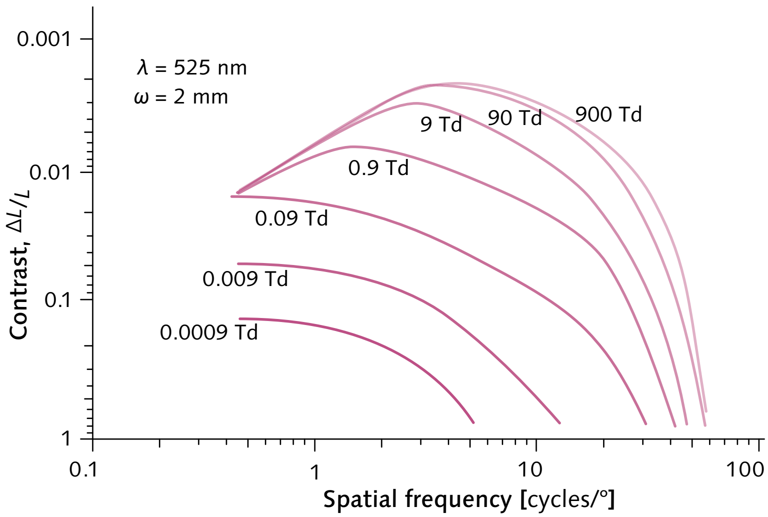
Barten, Peter G.J. (1999), *Contrast Sensitivity of the Human Eye and Its Effect on Image Quality* (Knegsel, Netherlands: HV Press). Also published by SPIE Press.

In vision science, rather than characterizing vision by its response to an individual small feature, we place many small elements side by side. The spacing of these elements is measured in terms of spatial frequency, in units of *cycles per degree* (CPD, or ~/°); each cycle comprises a dark element and a white element. At the limit, a cycle comprises two samples or two pixels in adjacent columns; in the vertical dimension, the smallest cycle corresponds to two adjacent image rows.

Percept	*Physical quantity*	*Power*
Loudness	Sound pressure level	0.67
Saltiness	Sodium chloride concentration	1.4
Smell	Concentration of aromatic molecules	0.6

Table 23.1 **Power functions in perception**

Figure 23.5 **The contrast sensitivity function (CSF)** of human vision varies with retinal illuminance, here shown in units of troland (Td). The curve at 9 Td, which typifies television viewing, peaks at about 4 cycles per degree (CPD, or ~/°). Below that spatial frequency, the eye acts as a differentiator; above it, the eye acts as an integrator.



Troland [Td] is a unit of retinal illuminance equal to object luminance (in $cd \cdot m^{-2}$) times pupillary aperture area (in mm^2).

Contrast sensitivity can also be plotted as a function of temporal modulation.

Figure 23.5 above shows a graph of the dependence of contrast sensitivity (on the *y*-axis) upon spatial frequency (on the *x*-axis, expressed in cycles per degree). Contrast sensitivity of 100 corresponds to a ratio of 1.01 (1%) being perceptible. The graph shows a family of curves, representing different adaptation levels, from very dark (0.0009 Td) to very bright (900 Td). The curve at 9 Td is typical of electronic displays.

For video engineering, three features of Figure 23.5 are important:

- First, the 90 Td curve has fallen to a contrast sensitivity of unity at about 60 cycles per degree. Vision isn't capable of perceiving spatial frequencies greater than this; a display need not reproduce detail higher than this frequency. This limit of vision sets an upper bound on the resolution (or bandwidth) that must be provided.
- Second, the peak of the 90 Td curve has a contrast sensitivity of about 1%; luminance ratios less than this need not be preserved. This limits the number of bits per pixel that must be provided.
- Third, the curve falls off at spatial frequencies below about one cycle per degree. Luminance can diminish (within limits) toward the edges of the image without the viewer's noticing; such was the case in traditional CRT consumer displays, though fall-off doesn't occur in LCD and PDP displays. Fall-off does occur in projection.

Campbell, Fergus W. and Robson, John G. (1968), "Application of Fourier analysis to the visibility of gratings," in *J. Physiol.* (London) **197**: 551–566.

In traditional video engineering, the spatial frequency and contrast sensitivity aspects of this graph are used independently. Most video compression systems, including JPEG and MPEG, exploit the interdependence of these two aspects, as will be explained in *JPEG and motion-JPEG (M-JPEG) compression,* on page 491.

This page intentionally left blank

Luminance and lightness 24

Perceptual coding is essential to maximize the performance of an image coding system. In commercial imaging, we rarely use pixel values proportional to luminance; instead, we use pixel values that approximate lightness. This chapter introduces luminance and lightness.

In *Colour science for video*, on page 287, I will describe how spectral power distributions (SPDs) in the range 400 nm to 700 nm are related to colours.

Relative luminance, denoted *Y*, is what I call a *linear-light* quantity; it is directly proportional to physical radiance weighted by the spectral sensitivity of human vision. Luminance involves light having wavelengths in the range of about 400 nm to 700 nm. (Luminance can also be computed as a properly weighted sum of linear-light red, green, and blue tristimulus components according to the principles and standards of the CIE.)

The term *luminance* is often carelessly and incorrectly used to refer to what is now properly called *luma*. See *Relative luminance*, on page 258, and Appendix A, *YUV and luminance considered harmful*, on page 567.

Video signal processing equipment does not compute the linear-light luminance of colour science; nor does it compute lightness. Instead, it computes an approximation of lightness, called *luma* (denoted *Y'*), as a weighted sum of *nonlinear* (gamma-corrected) *R'*, *G'*, and *B'* components. Luma is only loosely related to true (CIE) luminance. In *Constant luminance*, on page 107, I explained why video systems approximate lightness instead of computing it directly. I will detail the nonlinear coding used in video in *Gamma*, on page 315. In *Luma and colour differences*, on page 335, I will outline how luma is augmented with colour information.

Radiance, intensity

Image science concerns optical power incident upon the image plane of a sensor device, and optical power emergent from the image plane of a display device.

See *Introduction to radiometry and photometry*, on page 573. Some people believe that *light* is defined by what we can see; for them, electromagnetic radiation outside the band 360 nm to 830 nm isn't light!

Radiometry concerns the measurement of radiant optical power in the electromagnetic spectrum from 3×10^{11} Hz to 3×10^{16} Hz, corresponding to wavelengths from 1 mm down to 10 nm. There are four fundamental quantities in radiometry:

- Radiant optical power, *flux*, is expressed in units of watts [W].
- Radiant flux per unit area is *irradiance;* its units are watts per meter squared [$W \cdot m^{-2}$].
- Radiant flux in a certain direction – that is, radiant flux per unit of solid angle – is *radiant intensity;* its units are watts per steradian [$W \cdot sr^{-1}$].
- Flux in a certain direction, per unit area, is *radiance;* its units are watts per steradian per meter squared [$W \cdot sr^{-1} \cdot m^{-2}$].

Wideband radiance is measured with an instrument called a *radiometer*. A *spectroradiometer* measures spectral radiance – that is, radiance per unit wavelength incident upon the instrument. A *spectrophotometer* incorporates a light source, and measures spectral reflectance (or for an instrument specialized for film, spectral transmittance).

Photometry is essentially radiometry as sensed by human vision: In photometry, radiometric measurements are weighted by the spectral response of human vision (to be described). This involves wavelengths (symbolized λ) between 360 nm to 830 nm, or in practical terms, 400 nm to 700 nm. Each of the four fundamental quantities of radiometry – flux, irradiance, radiant intensity, and radiance – has an analog in photometry. The photometric quantities are *luminous flux*, *illuminance*, *luminous intensity*, and (absolute) *luminance.* In video engineering, luminance is the most important of these.

The unit of luminous intensity is the candela [cd]. It is one of the seven base units in the SI system; the others are meter, kilogram, second, ampere, kelvin, and mole.

Luminance

I presented a brief introduction to lightness terminology on page 27.

The *Commission Internationale de L'Éclairage* (CIE, or International Commission on Illumination) is the international body responsible for standards in the area of colour. The CIE defines brightness as *the attribute of a visual sensation according to which an area appears to exhibit more or less light*. Brightness is, by the CIE's definition, a subjective quantity: It cannot be measured.

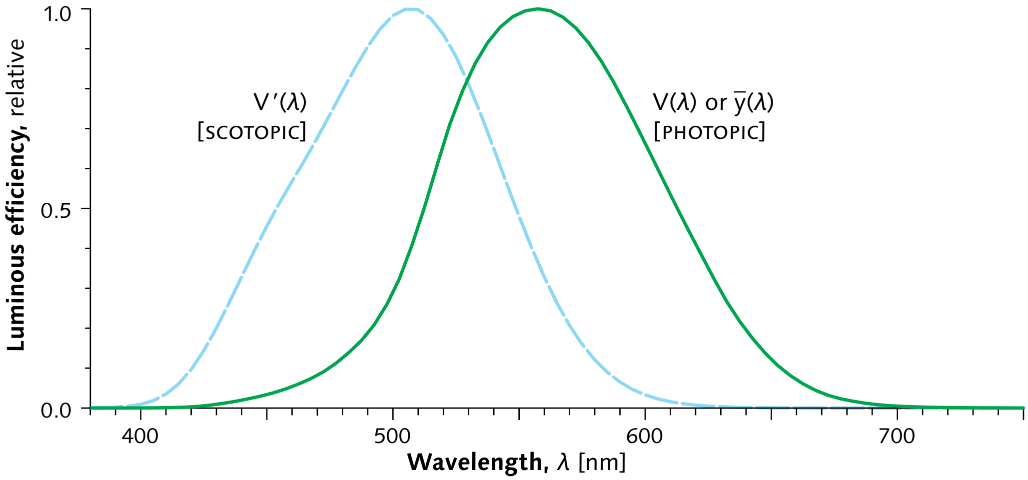


Figure 24.1 **Luminous efficiency functions.** The solid line indicates the luminance response of the cone photoreceptors – that is, the CIE *photopic* response. A monochrome scanner or camera must have this spectral response in order to correctly reproduce lightness. The peak occurs at about 555 nm, the wavelength of the brightest possible monochromatic 1 mW source. (The lightly shaded curve shows the *scotopic* response of the rod cells – loosely, the response of night vision. The increased relative luminance of shortwave light in scotopic vision is called the *Purkinje shift*.)

CIE Publication 15:2004, *Colorimetry*, 3rd Edition (Vienna, Austria: Commission Internationale de L'Éclairage).

The CIE has defined an objective quantity that is related to brightness. *Luminance* is defined as radiance weighted by the spectral sensitivity function – the sensitivity to power at different wavelengths – that is characteristic of vision. Put succinctly, brightness is apparent luminance.

The *luminous efficiency* of the CIE Standard Observer, denoted $\bar{y}(\lambda)$, is graphed as the solid line of Figure 24.1 above. The luminous efficiency function is also known as the $\bar{y}(\lambda)$ *colour-matching function* (CMF). It is defined numerically, is everywhere positive, and peaks at about 555 nm. When a spectral power distribution (SPD) is integrated using this weighting function, the result is *luminance*, symbolized L_V (or, where radiometry isn't in the context, just L). Luminance has units of candelas per meter squared, $cd \cdot m^{-2}$ (colloquially, "nits" or nt).

The y is pronounced *WYE-bar*. The luminous efficiency function is sometimes denoted V(λ), pronounced *VEE-lambda*.

In continuous terms, luminance is an integral of spectral radiance across the spectrum. It can be represented in discrete terms as a dot product. The magnitude of luminance is proportional to physical power; in that sense it resembles intensity. However, its spectral composition is intimately related to the lightness sensitivity of human vision.

You might intuitively associate pure luminance with grey, but a spectral power distribution having the shape of Figure 24.1 would *not* appear neutral grey! In fact, an SPD of that shape would appear distinctly green. As I will detail in *The CIE system of colorimetry,* on page 265, it is very important to distinguish analysis functions – called *colour-matching functions* (CMFs) of human vision, or the *spectral responsivity functions* (SRFs) of an image sensor – from synthesis functions, spectral power distributions (SPDs). The luminous efficiency function takes the role of an analysis function, not a synthesis function.

Relative luminance

Luminance factor is not a synonym for relative luminance: Luminance factor refers to the reflectance – relative to a perfect diffuse reflector – of a reflective surface.

In image reproduction – including photography, cinema, video, and print – we rarely, if ever, reproduce the absolute luminance of the original scene. Instead, we reproduce luminance roughly proportional to scene luminance, up to the maximum luminance available in the presentation medium. We process or record an approximation to *relative luminance*. To use the unqualified term *luminance* would suggest that we are processing or recording absolute luminance.

Once normalized to a specified or implied *reference white,* relative luminance is given the symbol *Y;* it has a purely numeric value (without units) which runs from 0 to 1 (which I prefer), or traditionally, 0 to 100. (Relative luminance is often called just "luminance.")

I will introduce *XYZ* and *LMS* in *The CIE system of colorimetry,* on page 265. I will introduce *RGB* in *Colour science for video,* on page 287.

Relative luminance, *Y*, is one of three distinguished tristimulus values. The other two, *X* and *Z*, are also unitless. Various other sets of tristimulus values, such as *LMS* and *RGB,* have an implied absolute reference, come in sets of three, and also carry no units.

Luminance from red, green, and blue

The luminous efficiency of vision peaks in the medium-wave region of the spectrum: If three monochromatic sources appear red, green, and blue, and have the same radiant power in the visible spectrum, then the green will appear the brightest of the three, the red will appear less bright, and the blue will be the darkest of the three. As a consequence of the luminous efficiency function, all saturated blue colours are quite dark, and all saturated yellows are quite light.

If the luminance of a scene element is to be sensed by a scanner or camera having a single spectral filter, then the spectral response of the scanner's filter must – in theory, at least – correspond to the luminous efficiency function of Figure 24.1. However, luminance can also be computed as a weighted sum of suitably chosen red, green, and blue tristimulus components. The coefficients are functions of vision, of the white reference, and of the particular red, green, and blue spectral weighting functions employed. For realistic choices of white point and primaries, the green coefficient is quite large, the blue coefficient is the smallest of the three, and the red coefficient has an intermediate value.

The primaries of contemporary video displays are standardized in BT.709. Weights computed from these primaries are appropriate to compute relative luminance from red, green, and blue tristimulus values for computer graphics, and for modern video cameras and modern displays in both SD and HD:

$$ {}^{709}Y = 0.2126\,R + 0.7152\,G + 0.0722\,B \qquad \text{Eq 24.1} $$

My notation is outlined in Figure 28.6, on page 343. The coefficients are derived in *Colour science for video*, on page 287.

To compute luminance using $(R+G+B)/3$ is at odds with the characteristics of vision.

For BT.709 primaries, luminance comprises roughly 21% power from the red (longwave) region of the spectrum, 72% from green (mediumwave), and 7% from blue (shortwave).

Blue has a small contribution to luminance. However, vision has excellent colour discrimination among blue hues. Equation 24.1 does not give you licence to assign fewer bits to blue than to red or green – in fact, it tells you nothing whatsoever about how many bits to assign to each channel.

Lightness (CIE L^*)

Lightness is defined by the CIE as *the brightness of an area judged relative to the brightness of* a *similarly illuminated area that appears to be white or highly transmitting*. Lightness is most succinctly described as apparent reflectance. Vision is attuned to estimating surface reflectance factors; lightness relates to that aspect of vision. The CIE's phrase "similarly illuminated area that appears white" involves the absolute luminance by which relative luminance is normalized. In digital imaging, the reference white luminance is ordinarily

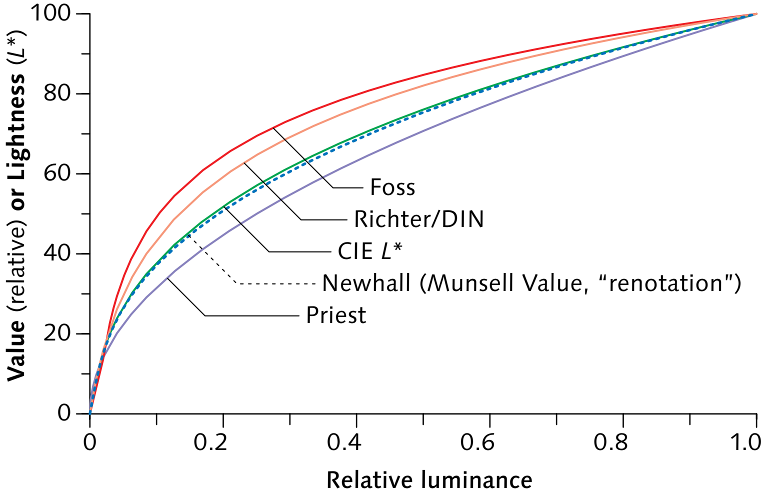


Figure 24.2 **Luminance and lightness.** The dependence of lightness (L^*) or value (V) upon relative luminance (Y) has been modeled by polynomials, power functions, and logarithms. In all of these systems, 18% "mid-grey" has lightness about halfway up the perceptual scale. This graph is adapted from Fig. 2 (6.3) in Wyszecki and Stiles, *Color Science* (cited on page 286).

closely related to the luminance of a perfectly diffusing reflector (PDR) in the scene, or the luminance at which such a scene element is presented (or will ultimately be presented) at a display.

In *Contrast sensitivity,* on page 249, I explained that vision has a nonlinear perceptual response to luminance. Vision scientists have proposed many functions that relate relative luminance to perceived lightness; several of these functions are graphed in Figure 24.2.

The L^* symbol is pronounced *EL-star*.

The computational version of lightness, denoted L^*, is defined by the CIE as a certain nonlinear function of relative luminance. In 1976, the CIE standardized *lightness*, L^* as an approximation of the lightness response of human vision. Other functions – such as *Munsell value* – specify alternate lightness scales, but the CIE L^* function is widely used and internationally standardized.

The L^* function has two segments: a linear segment near black, and a scaled and offset cube root ($^1/_3$-power) function everywhere else.

The 1976 version of the CIE standard expresses this definition of L^*:

Eq 24.2

$$L^*(Y) = \begin{cases} 903.3\,\dfrac{Y}{Y_N}; & \dfrac{Y}{Y_N} \le 0.008856 \\ 116\left(\dfrac{Y}{Y_N}\right)^{\frac{1}{3}} - 16; & 0.008856 < \dfrac{Y}{Y_N} \end{cases}$$

In the 2004 version of the standard, the decimals were replaced by exact rational fractions. Today's definition is equivalent to this:

Eq 24.3

$$L^*(Y) = \begin{cases} \left(\dfrac{116}{12}\right)^3 \dfrac{Y}{Y_N}; & \dfrac{Y}{Y_N} \le \left(\dfrac{24}{116}\right)^3 \\ 116\left(\dfrac{Y}{Y_N}\right)^{\frac{1}{3}} - 16; & \left(\dfrac{24}{116}\right)^3 < \dfrac{Y}{Y_N} \end{cases}$$

The argument *Y* is *relative* luminance, proportional to intensity. This quantity is already relative to some absolute white reference, typically the absolute luminance associated with a perfect (or imperfect, say 90%) diffuse reflector. The argument *Y* is tacitly assumed to lie on a scale whose maximum value (Y_N) is related to the viewer's adaptation state. The division by Y_N does *not* form relative luminance; rather, the normalization accommodates the tradition dating back to 1931 and earlier that tristimulus values lie on a 0 to 100 scale. For tristimulus reference range of 0 to 1, as I prefer, the division by Y_N can be omitted.

To compute L^* from optical density *D* in the range 0 to 2, use this relation:

$$L^* = 116 \cdot 10^{-D/3} - 16$$

The linear segment of L^* is convenient for mathematical reasons, but is not justified by visual perception: The utility of L^* is limited to a luminance ratio of about 100:1, and L^* values below 8 don't represent meaningful visual stimuli. (In graphics arts, luminance ratios up to about 300:1 are used with L^*.)

The exponent of the power function segment of the L^* function is $1/3$, but the scale factor of 116 and the offset of –16 modify the pure power function such that the best-fit pure power function has an exponent of 0.42, not $1/3$! L^* is *based* upon a cube root, but it is not best approximated by a cube root! The best pure-power function approximation to lightness is 100 times the 0.42-power of relative luminance.

My best-fit pure power function estimate is based upon numerical (Nelder-Mead) minimization of least-squares error on L^* values 0 through 100 in steps of 10. The same result is obtained by fitting 100 samples in linear-light space.

For television viewing, we typically set Y_N to reference white at the display. In television viewing, the viewer's adaptation is controlled both by the image itself and by elements in the field of view that are outside the image. In cinema, the viewer's adaptation is controlled mainly by the image itself. In cinema, setting Y_N to reference white is not necessarily appropriate; it may be more appropriate to set Y_N to the luminance of the representation of a perfect diffuse reflector in the displayed scene.

In a display system having contrast ratio of 100:1, L^* takes values between 9 and 100.

Relative luminance of 0.01 maps to L^* of almost exactly 9. You may find it convenient to keep in mind two exact mappings of L^*: Relative luminance of $^1/_{64}$ (0.015625) corresponds to L^* of exactly 13, and relative luminance of $^1/_8$ (0.125) corresponds to L^* of exactly 42 (which, as Douglas Adams would tell you, is the answer to Life, the Universe, and Everything).

ΔL^* is pronounced *delta EL-star*.

The difference between two L^* values, denoted ΔL^*, is a measure of perceptual "distance." In graphics arts, a difference of less than unity between two L^* values is generally considered to be imperceptible – that is, ΔL^* of unity is taken to lie at the threshold of discrimination. L^* is meaningless beyond about 200 – that is, beyond about 6.5 Y/Y_N.

In *Contrast sensitivity*, on page 249, I gave the example of logarithmic coding with a Weber contrast of 1.01. For reconstructing images for human viewing, it is never necessary to quantize relative luminance more finely than that. However, L^* suggests that a ratio of 1.01 is unnecessarily fine. The inverse L^* of 100 is unity; dividing that by the inverse L^* of 99 yields a Weber contrast of 1.025. The luminance ratio between adjacent L^* values increases as L^* falls, reaching 1.13 at L^* of 8 (at relative luminance of about 1%, corresponding to a contrast ratio of 100:1). L^* was standardized based upon estimation of lightness of diffusely reflecting surfaces; the linear segment below L^* of 8 was inserted for mathematical convenience. I consider estimating the visibility of lightness differences at luminance values less than 1% of white to be a research topic, and I recommend against using delta-L^* at such low luminances.

L^* provides one component of a *uniform colour space;* it can be described as *perceptually uniform*. Since we cannot directly measure the quantity in question, we cannot assign to it any strong properties of mathematical linearity; as far as I'm concerned, the term *perceptually linear* is not appropriate.

In Chapter 10, *Constant luminance*, I described how video systems use a luma signal (Y') that is an engineering approximation to lightness. The luma signal is only indirectly related to the relative luminance (Y) or the lightness (L^*) of colour science.

This page intentionally left blank

The CIE system of colorimetry 25

The *Commission Internationale de L'Éclairage* (CIE) has defined a system that maps a *spectral power distribution* (SPD) of physics into a triple of numerical values – CIE *XYZ* tristimulus values – that form the mathematical coordinates of colour space. In this chapter, I describe the CIE system. In the following chapter, *Colour science for video*, I will explain how these *XYZ* tristimulus values are related to linear-light *RGB* values.

Figure 25.1 **Example coordinate system**

Colour coordinates are analogous to coordinates on a map (see Figure 25.1). Cartographers have different map projections for different functions: Some projections preserve areas, others show latitudes and longitudes as straight lines. No single map projection fills all the needs of all map users. Analogously, there are many "colour spaces," and as in maps, no single coordinate system fills all of the needs of users.

In Chapter 24, *Luminance and lightness*, I introduced the linear-light quantity *luminance*. To reiterate, I use the term *luminance* and the symbol Y to refer to CIE luminance. I use the term *luma* and the symbol Y' to refer to the video component that conveys an approximation to lightness. Most of the quantities in this chapter, and in the following chapter *Colour science for video*, involve "linear-light" values that are proportional to intensity. In Chapter 10, *Constant luminance*, I related the theory of colour science to the practice of video. To approximate perceptual uniformity, video uses quantities such as R', G', B', and Y' that are *not* proportional to intensity.

Fundamentals of vision

About 8% of men and 0.4% of women have deficient colour vision, called *colour blindness*. Some people have fewer than three types of cones; some people have cones with altered spectral sensitivities.

Spectral properties can be measured in electron volts (eV); the visible spectrum encompasses the range from 3.1 eV to 1.6 eV. Sometimes *wave number,* the reciprocal of wavelength is used, ordinarily expressed in cm^{-1}.

As I explained in *Retina,* on page 247, human vision involves three types of colour photoreceptor *cone* cells, which respond to incident radiation having wavelengths (λ) from about 380 nm to 750 nm. The three cell types have different spectral responses; colour is the perceptual result of their absorption of light. Normal vision involves three types of cone cells, so three numerical values are necessary and sufficient to describe a colour: Normal human colour vision is inherently *trichromatic*.

Power distributions exist in the physical world; however, colour exists only in the eye and the brain. Isaac Newton put it this way, in 1675:

> "Indeed rays, properly expressed, are not coloured."

Definitions

On page 27, I outlined brightness, intensity, luminance, value, lightness, and tristimulus value. In Appendix B, *Introduction to radiometry and photometry,* on page 573, I give more rigorous definitions. In colour science, it is important to use these terms carefully. It is especially important to differentiate physical quantities (such as intensity and luminance), from perceptual quantities (such as lightness and value).

Hue is the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours, red, yellow, green, and blue, or a combination of two of them. Roughly speaking, if the dominant wavelength of a spectral power distribution shifts, the hue of the associated colour will shift.

Saturation is the colourfulness of an area, judged in proportion to its brightness. Saturation is a perceptual quantity; like brightness, it cannot be measured.

Bill Schreiber points out that the words *saturation* and *purity* are often used interchangeably, to the dismay of purists.

Purity is the ratio of the amount of a monochromatic stimulus to the amount of a specified achromatic stimulus which, when mixed additively, matches the colour in question. Purity is the objective correlate of saturation.

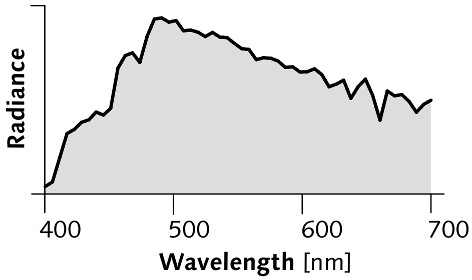
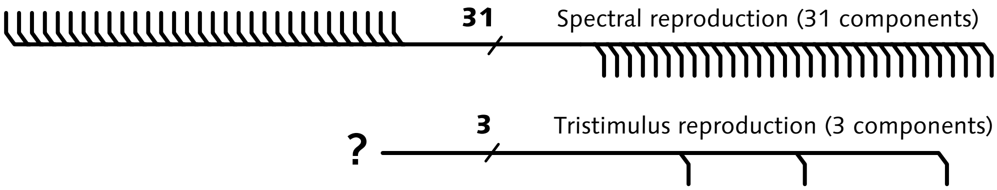


Figure 25.2 **Spectral and tristimulus colour reproduction.** A colour can be represented as a spectral power distribution (SPD), perhaps in 31 components representing power in 10 nm bands over the range 400 nm to 700 nm. However, owing to the trichromatic nature of human vision, if appropriate spectral weighting functions are used, three components suffice to represent colour. The SPD shown here is the CIE D_{65} daylight illuminant.



Spectral power distribution (SPD) and tristimulus

The more an SPD is concentrated near one wavelength, the more saturated the associated colour will be. A colour can be desaturated by adding light with power distributed across the visible spectrum.

The physical wavelength composition of light is expressed in a *spectral power distribution* (SPD), also known as *spectral radiance.* An SPD gives radiance [$W \cdot sr^{-1} \cdot m^{-2}$] or relative radiance as a function of wavelength, symbolized λ [nm]. An SPD representative of daylight is graphed at the upper left of Figure 25.2.

One way to reproduce a colour is to directly reproduce its spectral power distribution. This approach, termed *spectral reproduction*, is suitable for reproducing a single colour or a few colours. For example, the visible range of wavelengths from 400 nm to 700 nm could be divided into 31 bands, each 10 nm wide. However, using 31 components for each pixel is an impractical way to code an image. Owing to the trichromatic nature of vision, if suitable spectral weighting functions are used, any colour on its way to the eye can be described by just three components. This is called *tristimulus reproduction*.

Strictly speaking, *colorimetry* refers to the measurement of colour. In video, *colorimetry* is taken to encompass the transfer functions used to code linear *RGB* to *R'G'B'*, and the matrix that produces luma and colour difference signals. *Colorimetry* is spelled without *u*, even in England and Canada.

The science of *colorimetry* concerns the relationship between SPDs and colour. In 1931, the Commission Internationale de L'Éclairage (CIE) standardized weighting curves for a hypothetical *Standard Observer.* These curves – graphed in Figure 25.5, on page 271 – specify how an SPD can be transformed into three *tristimulus values* that specify a colour.

Pronounced *meh-ta-MAIR-ik* and *meh-TAM-er-ism*.

To specify a colour, it is not necessary to specify its spectrum – it suffices to specify its tristimulus values. To reproduce a colour, its spectrum need not be reproduced – it suffices to reproduce its tristimulus values. This is known as a *metameric* match. *Metamerism* occurs when a pair of spectrally distinct stimuli have the same tristimulus values.

The colours produced in reflective systems – such as photography, printing, or paint – depend not only upon the colourants and the substrate (media), but also on the SPD of the illumination. To guarantee that two coloured materials will match under illuminants having different SPDs, you may have to achieve a spectral match.

Spectral constraints

The relationship between spectral distributions and the three components of a colour value is usually explained starting from the famous colour-matching experiment. I will instead explain the relationship by illustrating the practical concerns of engineering the spectral filters required by a colour scanner or camera, using Figure 25.3 opposite.

For a textbook lowpass filter – but in the signal domain – see Figure 20.23 on page 212.

The top row shows the spectral sensitivity of three wideband optical filters having uniform response across each of the longwave, mediumwave, and shortwave regions of the spectrum. Most filters, whether for electrical signals or for optical power, are designed to have responses as uniform as possible across the passband, to have transition zones as narrow as possible, and to have maximum possible attenuation in the stopbands.

At the top right of Figure 25.3, I show two monochromatic sources, which appear saturated orange and red, analyzed by "textbook" bandpass filters. These two different wavelength distributions, which are seen as different colours, report the identical *RGB* triple [1, 0, 0]. The two SPDs are perceived as having different colours; however, this filter set reports identical *RGB* values. The wideband filter set senses colour incorrectly.

At first glance it may seem that the problem with the wideband filters is insufficient wavelength discrimination. The middle row of the example attempts to solve that problem by using three narrowband filters. The narrowband set solves one problem, but creates

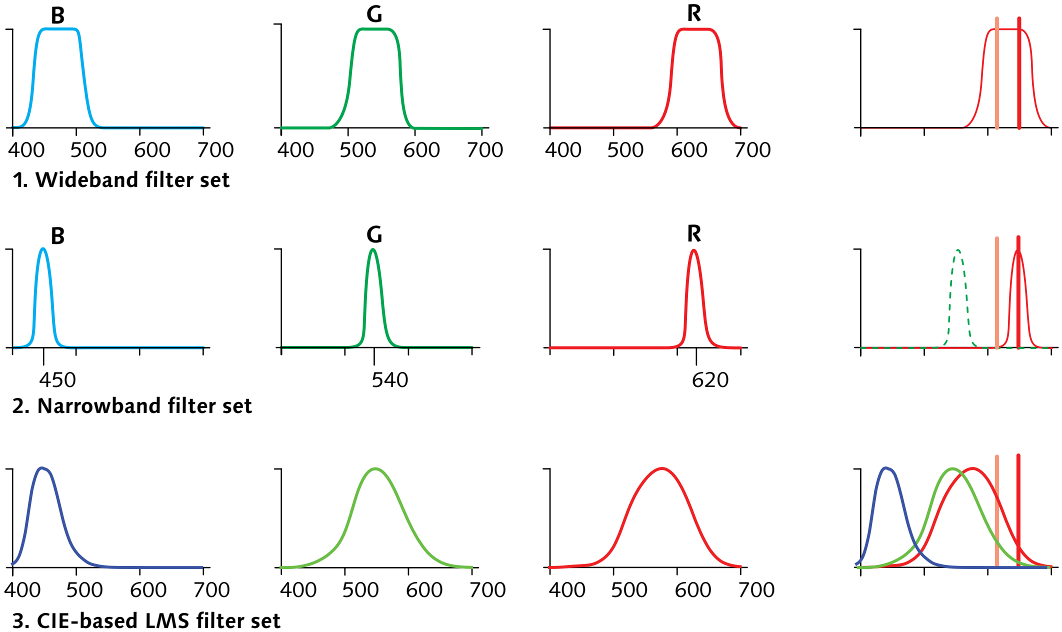


Figure 25.3 **Spectral constraints** are associated with scanners and cameras. **1. The wideband filter set** of the top row shows the spectral sensitivity of filters having uniform response across the shortwave, mediumwave, and longwave regions of the spectrum. Two monochromatic sources seen by the eye to have different colours – in this case, a saturated orange and a saturated red – cannot be distinguished by the filter set. **2. The narrowband filter set** in the middle row solves that problem, but creates another: Many monochromatic sources "fall between" the filters, and are sensed indistinguishably as black. To see colour as the eye does, the filter responses must closely relate to the colour response of the eye. **3. The CIE-based filter set** in the bottom row shows the *Hunt-Pointer-Estévez* (HPE) *colour-matching functions* (CMFs).

another: Many monochromatic sources "fall between" the filters. Here, the orange source reports an *RGB* triple of [0, 0, 0], identical to the result of scanning black.

Although my example is contrived, the problem is not. Ultimately, the test of whether a camera or scanner is successful is whether it reports distinct *RGB* triples if and only if human vision sees two SPDs as being different colours. For a scanner or a camera to see colour as the eye does, the filter sensitivity curves must be intimately related to the response of human vision – more specifically, the camera spectral sensitivities must be identical to the CIE CMFs, or a linear combination of

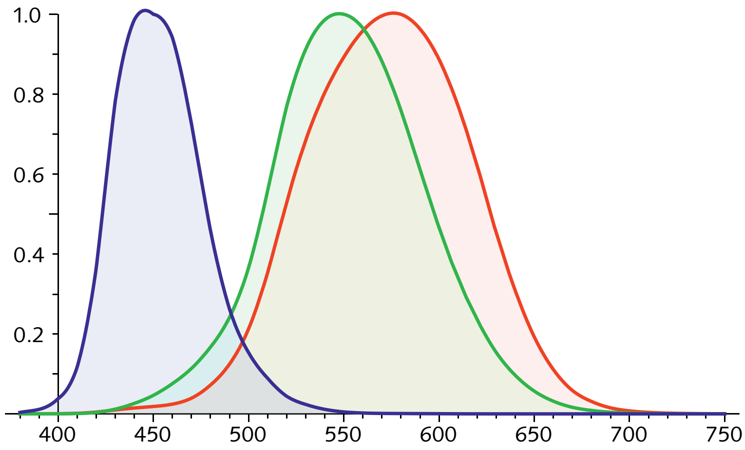


Figure 25.4 **The HPE colour-matching functions** estimate the responses of the three classes of *cone* photoreceptor cells. These are the *Hunt-Pointer-Estévez* (HPE) *colour-matching functions* (CMFs). In a practical camera, it is desirable for noise performance reasons to move the longwave ("red") response toward longer wavelengths; however, you do this, colour accuracy suffers.

What I call the *Maxwell-Ives criterion* is sometimes called *Luther-Ives*, or just *Luther.* In my view, James Clerk Maxwell and Herbert E. Ives mainly deserve the credit.

them. A camera that meets this requirement is said to conform to the *Maxwell-Ives criterion.*

The famous "colour-matching experiment" was devised during the 1920s to characterize the relationship between physical spectra and perceived colour. Today, we might seek the best approximation to the spectral sensitivities of the cone photoreceptor cells. Those functions are illustrated at the bottom of Figure 25.3, and they are graphed at larger scale in Figure 25.5. Different researchers prefer slightly different versions of these functions; the ones shown here are the *Hunt-Pointer-Estévez* (HPE) *cone fundamentals.*

CIE 15 (2004), *Colorimetry, 3rd Edition* (Vienna, Austria: Commission Internationale de L'Éclairage).

$\bar{x}$, $\bar{y}$, and $\bar{z}$ are pronounced *ECKS-bar, WYE-bar, ZEE-bar*.

Some authors refer to CMFs as *colour mixture curves*, or CMCs. That usage is best avoided, because CMC denotes a particular colour difference formula defined in British Standard BS:6923.

The CIE did not attempt to directly determine the responses of the cone cells. Instead, theirs was an indirect experiment.that measured mixtures of different spectral distributions that are required for human observers to match colours. In 1931 the CIE took data from these experiments, transformed the data according to certain mathematical principles, and standardized a set of spectral weighting functions that are related to the cone responses by a 3×3 matrix transform.

The CIE curves are called the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ *colour-matching functions* (CMFs) for the CIE Standard Observer, and are graphed in Figure 25.5. They are defined numerically; they are everywhere nonnegative.

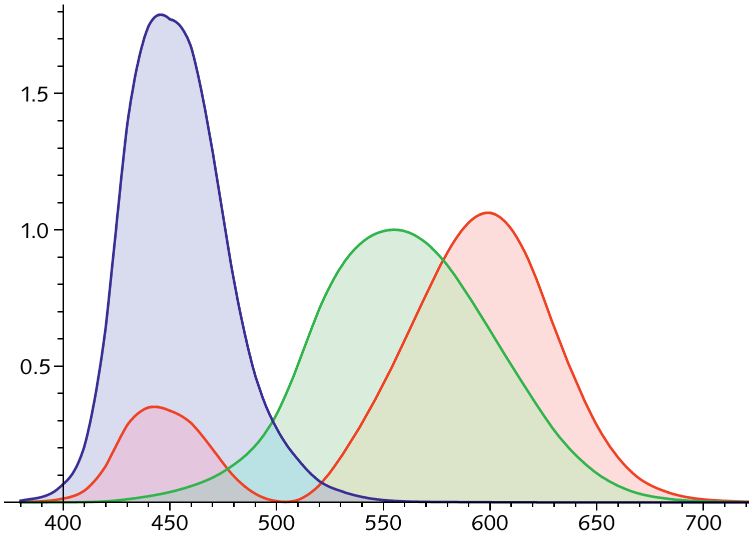


Figure 25.5 **CIE 1931, 2° colour-matching functions.** A sensor or camera must have these spectral response curves, or linear combinations of them, in order to capture all colours. However, practical considerations make this difficult. These are analysis functions; they are *not* comparable to spectral power distributions! The standard $\bar{y}(\lambda)$ function is scaled to unity at 560 nm. The $\bar{x}(\lambda)$ and $\bar{z}(\lambda)$ functions are scaled to match the integral of $\bar{y}(\lambda)$.

The CIE 1931 functions are appropriate to estimate the visual response to stimuli subtending angles of about 2° at the eye. In 1964, the CIE standardized a set of CMFs suitable for stimuli subtending about 10°; this set is generally not appropriate for image reproduction.

The functions of the CIE Standard Observer were standardized based upon experiments with visual colour matching. Research since then revealed the spectral sensitivities of the three types of cone cells – the *cone fundamentals*. We would expect the CIE CMFs to be intimately related to the properties of the retinal photoreceptors; many experimenters have related the cone fundamentals to CIE tristimulus values through 3×3 linear matrix transforms. None of the proposed mappings is very accurate, apparently owing to the intervention of high-level visual processing. For engineering purposes, the CIE functions suffice.

The term *sharpening* is used in the colour science community to describe certain 3×3 transforms of cone fundamentals; the "sharpening" is in the spectral domain. I consider the term to be unfortunate, because in image science, *sharpening* more sensibly refers to spatial phenomena.

The $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ CMFs each have one peak – each is "unimodal." However, the $\bar{x}(\lambda)$ CMF is bimodal, having a secondary peak between 400 nm and 500 nm. This "bump" does not directly reflect any physiological property of vision; it is best considered as a consequence of the mathematical process by which the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ curves are constructed.

CIE *XYZ* tristimulus

X, Y, and *Z* are pronounced *big-X, big-Y,* and *big-Z,* or *cap-X, cap-Y,* and *cap-Z*, to distinguish them from *little-x* and *little-y*, to be described in a moment.

Weighting an SPD under the $\bar{y}(\lambda)$ colour-matching function yields luminance (symbol *Y*), as I described on page 205. When luminance is augmented with two other values, computed in the same manner as luminance but using the $\bar{x}(\lambda)$ and $\bar{z}(\lambda)$ colour-matching functions, the resulting values are known as *XYZ tristimulus* values (denoted *X*, *Y*, and *Z*). *XYZ* values correlate to the spectral sensitivity of human vision. Their amplitudes – always nonnegative – are proportional to intensity.

Tristimulus values are computed from a continuous SPD by integrating the SPD under the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ colour-matching functions. In discrete form, tristimulus values are computed by a matrix multiplication, as illustrated in Figure 25.6 opposite.

Grassmann's Third Law:

Sources of the same colour produce identical effects in an additive mixture regardless of their spectral composition.

Human colour vision follows a principle of superposition known as Grassmann's Third Law: The set of tristimulus values computed from the sum of a set of SPDs is identical to the sum of the tristimulus values of each SPD. Due to this linearity of additive colour mixture, any set of three components that is a nontrivial linear combination of *X*, *Y,* and *Z* – such as *R, G,* and *B* – is also a set of tristimulus values. (In *Transformations between RGB and CIE XYZ,* on page 307, I will introduce related CMFs that produce *R, G,* and *B* tristimulus values.)

Luminance can be considered to be a distinguished tristimulus value that is meaningful on its own, and, exceptionally, carries units of $cd \cdot m^{-2}$. Apart from luminance, tristimuli come in sets of three, as the word suggests, and have no units.

This chapter accepts the CIE Standard Observer rather uncritically. Although the CIE Standard Observer is very useful and widely used, some researchers believe that it exhibits some problems and ought to be improved. For one well-informed and provocative view, see Thornton.

Thornton, William A. (1999), "Spectral sensitivities of the normal human visual system, color-matching functions and their principles, and how and why the two sets should coincide," in *Color Research and Application* **24** (2): 139–156 (Apr.).

$$
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} =
\begin{bmatrix}
0.0143 & 0.0004 & 0.0679 \\
0.0435 & 0.0012 & 0.2074 \\
0.1344 & 0.0040 & 0.6456 \\
0.2839 & 0.0116 & 1.3856 \\
0.3483 & 0.0230 & 1.7471 \\
0.3362 & 0.0380 & 1.7721 \\
0.2908 & 0.0600 & 1.6692 \\
0.1954 & 0.0910 & 1.2876 \\
0.0956 & 0.1390 & 0.8130 \\
0.0320 & 0.2080 & 0.4652 \\
0.0049 & 0.3230 & 0.2720 \\
0.0093 & 0.5030 & 0.1582 \\
0.0633 & 0.7100 & 0.0782 \\
0.1655 & 0.8620 & 0.0422 \\
0.2904 & 0.9540 & 0.0203 \\
0.4334 & 0.9950 & 0.0087 \\
0.5945 & 0.9950 & 0.0039 \\
0.7621 & 0.9520 & 0.0021 \\
0.9163 & 0.8700 & 0.0017 \\
1.0263 & 0.7570 & 0.0011 \\
1.0622 & 0.6310 & 0.0008 \\
1.0026 & 0.5030 & 0.0003 \\
0.8544 & 0.3810 & 0.0002 \\
0.6424 & 0.2650 & 0.0000 \\
0.4479 & 0.1750 & 0.0000 \\
0.2835 & 0.1070 & 0.0000 \\
0.1649 & 0.0610 & 0.0000 \\
0.0874 & 0.0320 & 0.0000 \\
0.0468 & 0.0170 & 0.0000 \\
0.0227 & 0.0082 & 0.0000 \\
0.0114 & 0.0041 & 0.0000
\end{bmatrix}^{\mathbf{T}}
\bullet
\begin{bmatrix}
82.75 \\ 91.49 \\ 93.43 \\ 86.68 \\ 104.86 \\ 117.01 \\ 117.81 \\ 114.86 \\ 115.92 \\ 108.81 \\ 109.35 \\ 107.80 \\ 104.79 \\ 107.69 \\ 104.41 \\ 104.05 \\ 100.00 \\ 96.33 \\ 95.79 \\ 88.69 \\ 90.01 \\ 89.60 \\ 87.70 \\ 83.29 \\ 83.70 \\ 80.03 \\ 80.21 \\ 82.28 \\ 78.28 \\ 69.72 \\ 71.61
\end{bmatrix}
\begin{matrix}
\text{400 nm} \\ \\ \\ \\ \\ \text{450 nm} \\ \\ \\ \\ \\ \text{500 nm} \\ \\ \\ \\ \\ \text{550 nm} \\ \\ \\ \\ \\ \text{600 nm} \\ \\ \\ \\ \\ \text{650 nm} \\ \\ \\ \\ \\ \text{700 nm}
\end{matrix}
$$

Figure 25.6 **Calculation of tristimulus values by matrix multiplication** starts with a column vector representing the SPD. The 31-element column vector in this example is a discrete version of CIE Illuminant D_{65} sampled at 10 nm intervals. The SPD is matrix-multiplied by a discrete version of the CIE $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ colour-matching functions (CMFs) of Figure 25.5, here in a 31×3 matrix (which is sometimes denoted $\boldsymbol{A}$). The superscript **T** denotes the matrix transpose operation. The result of the matrix multiplication is a set of *XYZ* tristimulus components.

In the caption to Figure 25.5, I mentioned that $\bar{y}(\lambda)$ is scaled to unity at 560 nm. In the 10 nm approximation given here, the value is not exactly unity owing to the CIE's interpolation procedure.

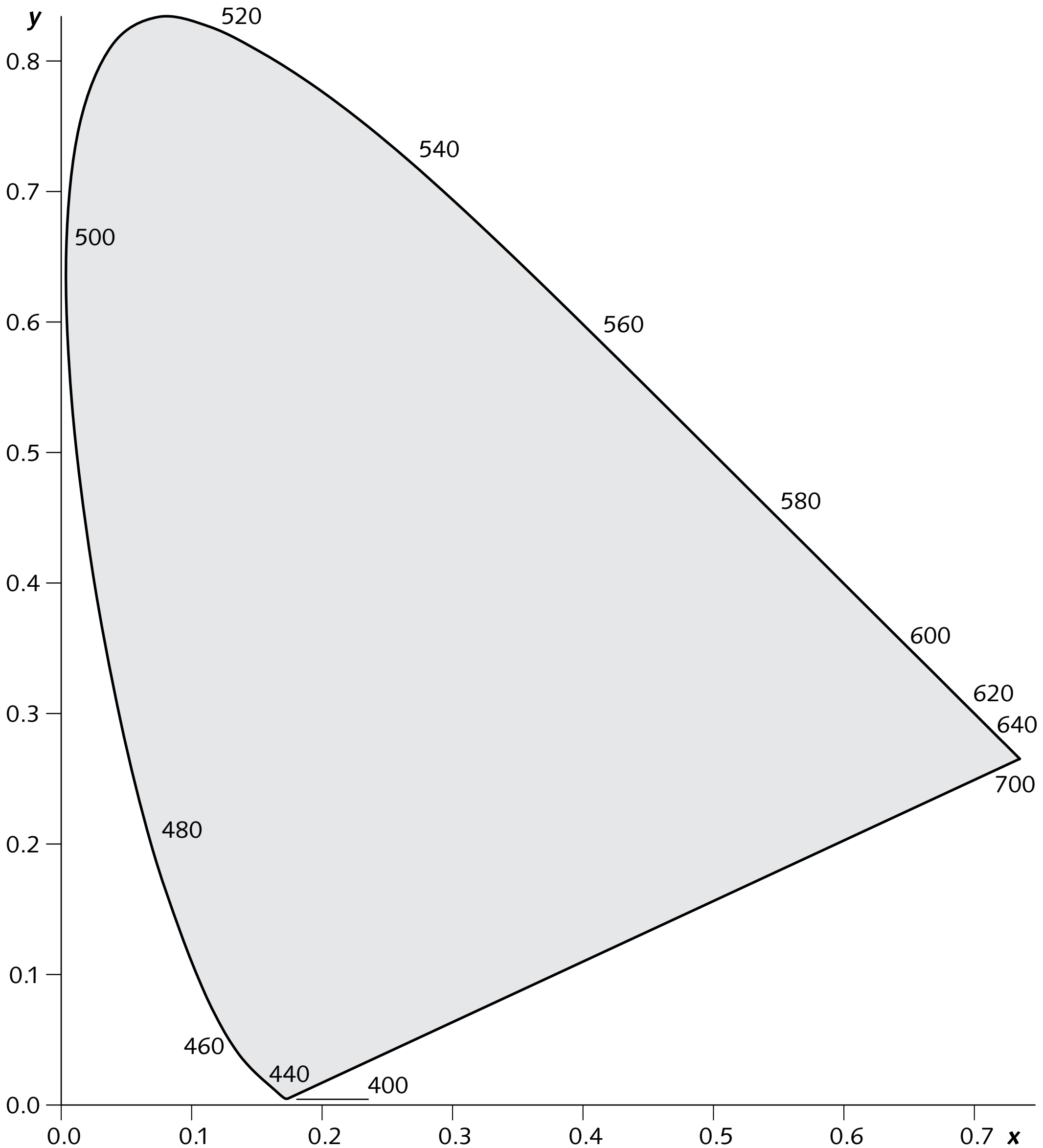


Figure 25.7 **CIE 1931 2° [x, y] chromaticity diagram.** The spectral locus is a horseshoe-shaped path swept by a monochromatic source as it is tuned from 400 nm to 700 nm. The *line of purples* traces SPDs that combine longwave and shortwave power but have no mediumwave power. All colours lie within the horseshoe-shaped region: Points outside this region are not colours.

This diagram is not a slice through [*X*, *Y*, *Z*] space! Instead, points in [*X*, *Y*, *Z*] project onto the plane of the diagram in a manner comparable to the perspective projection. White has [*X*, *Y*, *Z*] values near [1, 1, 1]; it projects to a point near the center of the diagram, in the region of [$\frac{1}{3}$, $\frac{1}{3}$]. Attempting to project black, at *XYZ* coordinates [0, 0, 0], would require dividing by zero in Equation 25.1: Black has no place in a chromaticity diagram.

CIE [x, y] chromaticity

It is convenient, for both conceptual understanding and for computation, to have a representation of "pure" colour in the absence of lightness. The CIE standardized a procedure for normalizing *XYZ* tristimulus values to obtain two *chromaticity* values x and y.

The x and y symbols are pronounced *little-x* and *little-y*.

Chromaticity values are computed by this projective transformation:

$$x = \frac{X}{X+Y+Z}; \qquad y = \frac{Y}{X+Y+Z} \qquad \text{Eq 25.1}$$

A third chromaticity coordinate, z, is defined, but is redundant since $x + y + z = 1$. The x and y chromaticity coordinates are abstract values that have no direct physical interpretation.

A colour can be specified by its chromaticity and luminance, in the form of an xyY triple. To recover X and Z tristimulus values from [x, y] chromaticities and luminance, use the inverse of Equation 25.1:

$$X = \frac{x}{y}Y; \qquad Z = \frac{1-x-y}{y}Y \qquad \text{Eq 25.2}$$

A colour plots as a point in an [x, y] *chromaticity diagram*, plotted in Figure 25.7 opposite.

In Figure 25.8 in the margin, I sketch several features of the [x, y] diagram. The important features lie on, or below and to the left of, the line $y = 1 - x$.

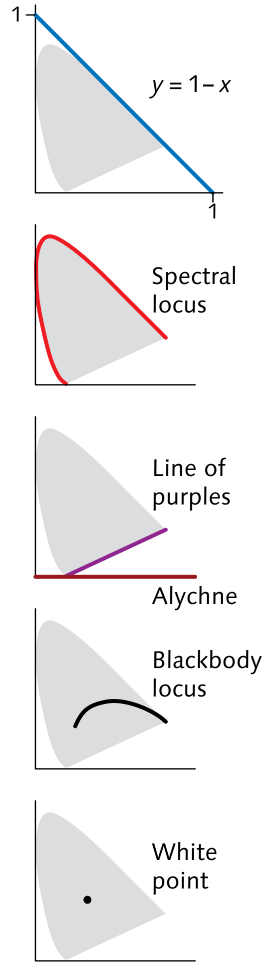


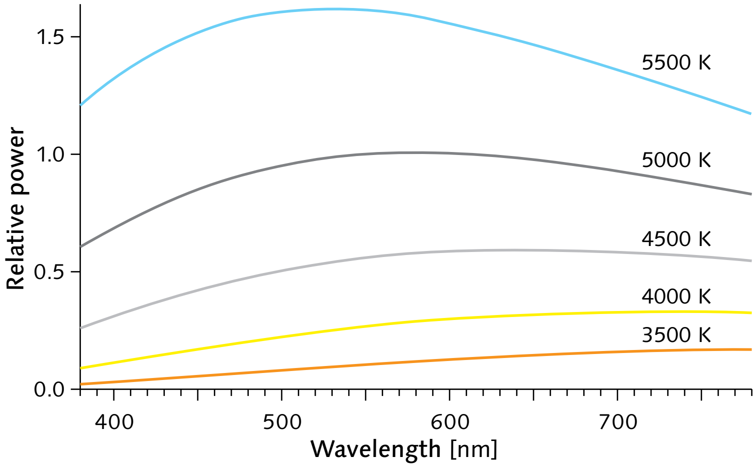
Figure 25.8 **CIE [x, y] chart features.**

When a narrowband (monochromatic) SPD comprising power at just one wavelength is swept across the range 400 nm to 700 nm, it traces the inverted-U (or horseshoe) shaped *spectral locus* in [x, y] coordinates.

The sensation of purple cannot be produced by a single wavelength; it requires a mixture of shortwave and longwave light. The *line of purples* on a chromaticity diagram joins the chromaticity of extreme blue (violet), containing only shortwave power, to the chromaticity of extreme red, containing only longwave power.

There is no unique physical or perceptual definition of white. Many important sources of illumination are blackbody radiators, whose chromaticity coordinates lie on the *blackbody locus* (sometimes called the *Planckian*

Figure 25.9 **SPDs of blackbody radiators** at several temperatures are graphed here. As the temperature increases, the absolute power increases and the peak of the spectral distribution shifts toward shorter wavelengths.



locus). Blackbody radiators will be discussed in the next section.

An SPD that appears white has CIE [*X*, *Y*, *Z*] values of about [1, 1, 1], and [*x*, *y*] coordinates in the region of [⅓, ⅓]: White plots in the central area of the chromaticity diagram. In the section *White,* on page 278, I will describe the SPDs associated with white.

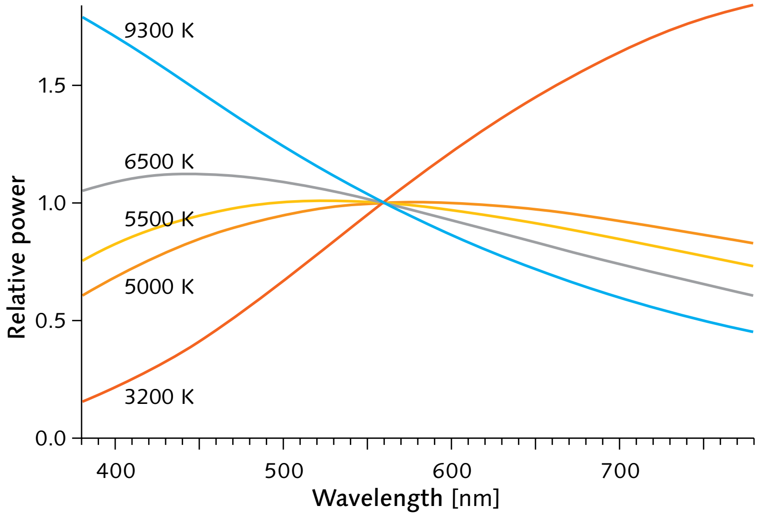
Any all-positive (*physical*, or *realizable*) SPD plots as a single point in the chromaticity diagram, within the region bounded by the spectral locus and the line of purples. All colours lie within this region; points outside this region are not associated with colours. It is silly to qualify "colour" by "visible," because colour is itself defined by vision – if it's invisible, it's not a colour!

In the projective transformation that forms *x* and *y*, any additive mixture (linear combination) of two SPDs – or two tristimulus values – plots on a straight line in the [*x*, *y*] plane. However, distances are not preserved, so chromaticity values do not combine linearly. Neither [*X*, *Y*, *Z*] nor [*x*, *y*] coordinates are perceptually uniform.

Blackbody radiation

Max Planck determined that the SPD radiated from a hot object – a *blackbody radiator* – is a function of the temperature to which the object is heated. Figure 25.9 above shows the SPDs of blackbody radiators at several temperatures. As temperature increases, the absolute

Figure 25.10 **SPDs of blackbody radiators, normalized** to equal power at 555 nm, are graphed here. The dramatically different spectral character of blackbody radiators at different temperatures is evident.



The symbol for Kelvin is properly written K (with no degree sign).

To a colour scientist, it's paradoxical that cold water faucets are colour-coded blue and hot water faucets are colour-coded red!

power increases and the spectral peak shifts toward shorter wavelengths. If the power of blackbody radiators is normalized at an arbitrary wavelength, dramatic differences in spectral character become evident, as illustrated in Figure 25.10 above.

Many sources of illumination have, at their core, a heated object, so it is useful to characterize an illuminant by specifying the absolute temperature (in units of kelvin, K) of a blackbody radiator having the same hue.

The *blackbody locus* is the path traced in [*x*, *y*] coordinates as the temperature of a blackbody source is raised. At low temperature, the source appears red ("red hot"). When a viewer is adapted to a white reference of CIE D_{65}, which I will describe in a moment, at about 2000 K, the source appears orange. Near 4000 K, it appears yellow; at about 6000 K, white. Above 10,000 K, it is blue hot.

Colour temperature

An illuminant may be characterized by a single colour temperature number – the temperature of a blackbody radiator that exactly matches the chromaticity of the source. If the match is approximate, the term *correlated colour temperature* (CCT) is used.

The 1960 [*u*, *v*] coordinates are described in the marginal note on page 281.

Colour temperature is sometimes augmented by a second number giving the closest distance in the deprecated CIE 1960 [*u*, *v*] coordinates of the colour from the blackbody locus – the arcane "minimum perceptible colour difference" (MPCD) units. I consider it more sensible to specify colour temperature in kelvin for intuitive purposes, accompanied by [*x*, *y*] or [*u'*, *v'*] chromaticity coordinates.

When a blackbody source's temperature sweeps from a low value (say 1000 K) to a high value (say 20,000 K), the chromaticity coordinate of the source sweeps out a path called the *blackbody locus* in the chromaticity diagram. (See Figure 25.8, on page 275.) Such a plot distributes temperatures in a highly nonuniform manner.

White

As I mentioned a moment ago, there is no unique definition of white: To achieve accurate colour, you must specify the SPD or the chromaticity of white. In additive mixture, to be detailed on page 288, the *white point* is the set of tristimulus values (or the luminance and chromaticity coordinates) of the colour reproduced by equal contributions of the red, green, and blue primaries. The colour of white is a function of the ratio – or *balance* – of power among the primary components. (In subtractive reproduction, the colour of white is determined by the SPD of the illumination, multiplied by the SPD of the uncoloured media.)

It is sometimes convenient for purposes of calculation to define white as an SPD whose power is uniform throughout the visible spectrum. This white reference is known as the *equal-energy illuminant*, denoted *CIE Illuminant E;* its CIE [*x*, *y*] coordinates are [$\frac{1}{3}$, $\frac{1}{3}$].

The CIE D illuminants are properly denoted with a two-digit subscript. CIE Illuminant D_{65} has a correlated colour temperature of about 6504 K.

A more realistic reference, approximating daylight, has been numerically specified by the CIE as Illuminant D_{65}. You should use this unless you have a good reason to use something else. The print industry commonly uses D_{50} and photography commonly uses D_{55}; these represent compromises between the conditions of indoor (tungsten) and daylight viewing. Figure 25.11 shows the SPDs of several standard illuminants; chromaticity coordinates are given in Table 25.1.

Concerning 9300 K, see page 311.

Many computer displays and many consumer television receivers have a default colour temperature setting of 9300 K. That white reference contains too much blue to achieve acceptable image reproduction in Europe or

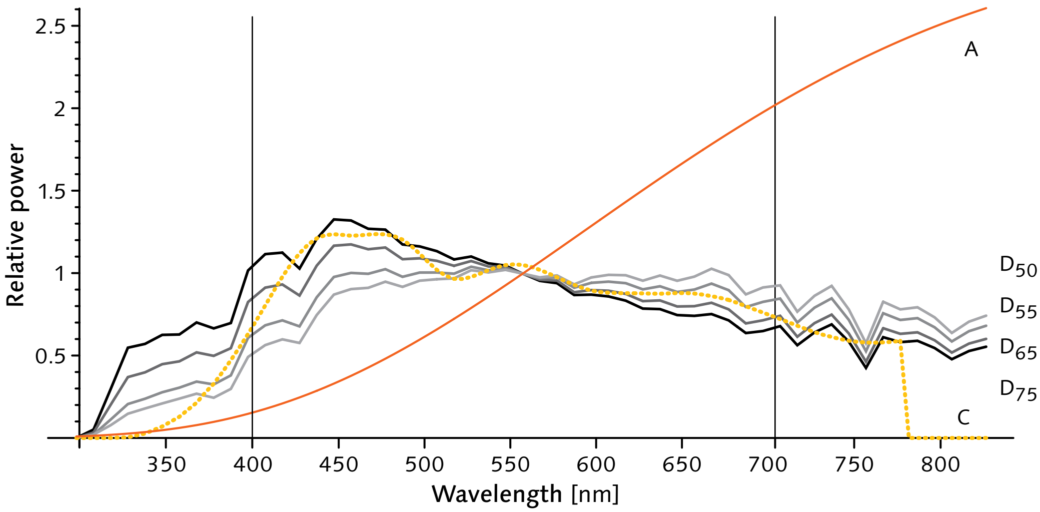


Figure 25.11 **CIE illuminants** are graphed here. Illuminant A is an obsolete standard representative of tungsten illumination; its SPD resembles the blackbody radiator at 3200 K shown in Figure 25.10, on page 277. Illuminant C was an early standard for daylight; it too is obsolete. The family of D illuminants represents daylight at several colour temperatures.

America; transform from D_{65} to 9300 K involves multiplying the BT.709 blue tristimulus value by about 1.3. However, there is a cultural preference in Asia for a more bluish reproduction than D_{65}; 9300 K is common in Asia (e.g., in studio displays in Japan).

Table 25.1 enumerates the chromaticity coordinates of several common white references:

Notation	*x*	*y*	*z*	u'_N	v'_N
1666.7 K (6000 mirek)	0.37683	0.38050	0.24267	0.2213	0.5027
CIE Ill. A (obsolete), ~2856 K	0.44757	0.40745	0.14498	0.2560	0.5243
CIE Ill. C (obsolete)	0.31006	0.31616	0.37378	0.2009	0.4609
CIE Ill. D_{50}	0.3457	0.3587	0.2956	0.2091	0.4882
CIE Ill. D_{55}	0.3325	0.3476	0.3199	0.2044	0.4801
CIE Ill. D_{65}, ~6504 K	0.312727	0.329024	0.358250	0.1978	0.4683
CIE Ill. E (equal-energy)	0.333334	0.333330	0.333336	0.2105	0.4737
9300 K (used in studio standards in Asia)	0.2830	0.2980	0.4190	0.1884	0.4463
∞ (0 mirek)	0.23704	0.236741	0.526219	0.1767	0.3970

Table 25.1 **White references.** The CIE D_{65} standard ubiquitous in SD and HD is highlighted.

The reciprocal of correlated colour temperature is somewhat more perceptually uniform than correlated colour temperature itself. Cinematographers use units of *mirek* (micro reciprocal kelvin [MK^{-1}]), that is, $10^6/t$, where t is in units of kelvin [K]. Mirek units are more perceptually uniform than kelvin. For typical video or cinema acquisition, CCT typically ranges from 2000 K to 10,000 K; that is, from 500 to 100 mirek.

The mirek unit is sometimes called *reciprocal mega-kelvin*, and was historically called *mired* ("micro reciprocal degree") .

Chromatic adaptation

Human vision adapts to the viewing environment. An image viewed in isolation – such as a 35 mm slide, or motion picture film projected in a dark room – creates its own white reference; a viewer will be quite tolerant of variation in white point. However, if the same image is viewed alongside an external white reference, or with a second image, differences in white point can be objectionable. Complete adaptation seems to be confined to colour temperatures from about 5000 K to 6500 K. Tungsten illumination, at about 3200 K, almost always appears somewhat yellow.

Tungsten illumination can't have a colour temperature higher than tungsten's melting point, 3695 K.

Perceptually uniform colour spaces

As I outlined in *Perceptual uniformity,* on page 30, a system is perceptually uniform if a small perturbation to a component value is approximately equally perceptible across the range of that value.

Luminance is not perceptually uniform. On page 259, I described how luminance can be transformed to lightness, denoted L^*, which is nearly perceptually uniform:

$$L^*(Y) = \begin{cases} \left(\frac{116}{12}\right)^3 \frac{Y}{Y_N}; & \frac{Y}{Y_N} \le \left(\frac{24}{116}\right)^3 \\ 116\left(\frac{Y}{Y_N}\right)^{\frac{1}{3}} - 16; & \left(\frac{24}{116}\right)^3 < \frac{Y}{Y_N} \end{cases} \qquad \text{Eq 25.3}$$

$L^*u^*v^*$ and $L^*a^*b^*$ are often written CIELUV and CIELAB; they are usually pronounced *SEA-love* and *SEA-lab*. The u^* and v^* quantities of colour science – and the u' and v' quantities, to be described – are unrelated to the U and V colour difference components of video.

Extending this concept to colour, *XYZ* and *RGB* tristimulus values, and *xyY* (chromaticity and luminance), are far from perceptually uniform. Finding a transformation of *XYZ* into a reasonably perceptually uniform space occupied the CIE for a decade, and in the end no single system could be agreed upon. In 1976, the CIE standardized two systems, $L^*u^*v^*$ and $L^*a^*b^*$, which I will now describe. In both systems, perceptual difference is approximated as Euclidean distance.

CIE *L*u*v**

Computation of CIE $L^*u^*v^*$ starts with a projective transformation of $[X, Y, Z]$ into intermediate u' and v' quantities:

$$u' = \frac{4\,X}{X + 15\,Y + 3\,Z}; \qquad v' = \frac{9\,Y}{X + 15\,Y + 3\,Z} \qquad \text{Eq 25.4}$$

Equivalently, u' and v' can be computed from x and y chromaticity:

$$u' = \frac{4\,x}{3 - 2\,x + 12\,y}; \qquad v' = \frac{9\,y}{3 - 2\,x + 12\,y} \qquad \text{Eq 25.5}$$

To recover X and Z tristimulus values from u' and v', use these relations:

$$X = \frac{9u'}{4v'}Y; \qquad Z = \frac{12 - 3u' - 20v'}{4v'}Y \qquad \text{Eq 25.6}$$

To recover x and y chromaticity from u' and v', use these relations:

$$x = \frac{9u'}{6u' - 16v' + 12}; \qquad y = \frac{4v'}{6u' - 16v' + 12} \qquad \text{Eq 25.7}$$

The primes in the CIE 1976 u' and v' quantities denote the successor to the obsolete 1960 CIE u and v quantities. $u = u'$; $v = \frac{2}{3}v'$ – that is, the 1960 v quantity underestimated visual perceptibility, and was multiplied by a factor of 1.5 to form the 1976 system. (To compute 1960 v, replace the numerator $9\,y$ in Eq 25.5 by $6\,y$.) The primes are not formally related to the primes in R', G', B', and Y', though all imply some degree of perceptual uniformity.

Since u' and v' are formed by a projective transformation, u' and v' coordinates are associated with a chromaticity diagram similar to the *CIE 1931 2° [x, y] chromaticity diagram* on page 274. You can use the $[u', v']$ diagram if you want to produce 2-D plots that are more suggestive of the perceptibility of colour differences than an $[x, y]$ plot would be. However, $[u', v']$ are subsequently multiplied by L^* (see Equation 25.8 below) to form $[u^*, v^*]$. That multiplication effectively enlarges the perceptual increment as luminance decreases. Perceptual differences in a $[u', v']$ diagram are dependant upon luminance, but that fact is not evident from the diagram: Be careful not to draw strong conclusions from the diagram.

To compute u^* and v^*, first compute L^*. Then compute u'_N and v'_N from your reference white X_N, Y_N, and Z_N. (The subscript *N* suggests *normalized*.) The u'_N and v'_N coordinates for several common white points are given in Table 25.1, *White references,* on page 279. (The $[x_N, y_N]$ coordinates for a colour temperature of infinity are about [0.237, 0.237]; the $[u'_N, v'_N]$ coordinates are about [0.177, 0.397].) Finally, compute u^* and v^*:

$$u^* = 13 \cdot L^*(Y) \cdot \left(u' - u'_N\right); \qquad v^* = 13 \cdot L^*(Y) \cdot \left(v' - v'_N\right) \qquad \text{Eq 25.8}$$

Gamut refers to the range of colours available in an imaging system. For gamuts typical of image reproduction, u^* and v^* values each range approximately ±100.

ΔE^* is pronounced *delta E-star*.

Euclidean distance in $L^*u^*v^*$ – denoted ΔE^*_{uv} – estimates the perceptibility of colour differences:

$$\Delta E^*_{uv} = \sqrt{\left(L^*_2 - L^*_1\right)^2 + \left(u^*_2 - u^*_1\right)^2 + \left(v^*_2 - v^*_1\right)^2} \qquad \text{Eq 25.9}$$

If ΔE^*_{uv} is unity or less, the colour difference is assumed to be imperceptible. However, $L^*u^*v^*$ does not achieve perceptual uniformity, it is merely an approximation. ΔE^*_{uv} values between about 1 and 4 may or may not be perceptible, depending upon the region of colour space being examined. ΔE^*_{uv} values greater than 4 are likely to be perceptible; whether such differences are objectionable depends upon circumstances.

A polar-coordinate version of the $[u^*, v^*]$ pair can be used to express chroma and hue:

$$C^*_{uv} = \sqrt{u^{*2} + v^{*2}}; \qquad h_{uv} = \tan^{-1} \frac{v^*}{u^*} \qquad \text{Eq 25.10}$$

In addition, there is a "psychometric saturation" term:

$$s_{uv} = \frac{C^*}{L^*} \qquad \text{Eq 25.11}$$

Chroma, hue, and saturation defined here are not directly related to saturation and hue in the *HSB, HSI, HSL, HSV*, and *IHS* systems used in computing and in digital image processing: Most of the published descriptions of these spaces, and most of the published formulæ, disregard the principles of colour science. In particular, the quantities called *lightness* and *value* are wildly inconsistent with their definitions in colour science.

CIE $L^*u^*v^*$ exhibits reasonable perceptual uniformity. $L^*u^*v^*$ has been common in video because the mapping of *XYZ*, *xyY*, and *RGB* to the *u'v'* coordinates is projective: Straight lines in any of these spaces map to straight lines in *u'v'*. Despite the convenience and utility of $L^*u^*v^*$, colour scientists today generally agree that better perceptual performance is exhibited by $L^*a^*b^*$, which I will now describe.

CIE *L*a*b** (CIELAB)

The quantities a^* and b^* are computed as follows:

Eq 25.12

$$a^* = \frac{125}{29}\left[\mathrm{L}^*\!\left(\frac{X}{X_\mathrm{N}}\right) - \mathrm{L}^*\!\left(\frac{Y}{Y_\mathrm{N}}\right)\right]; \qquad b^* = \frac{50}{29}\left[\mathrm{L}^*\!\left(\frac{Y}{Y_\mathrm{N}}\right) - \mathrm{L}^*\!\left(\frac{Z}{Z_\mathrm{N}}\right)\right]$$

The coefficients are approximately 4.310 and 1.724. My definition is written in an unusual way, using L^* instead of the traditional auxiliary function f. The definition of L^* involves a linear segment having C^1 continuity with a power function segment. That linear segment is incorporated (by way of L^*) into a^* and b^*.

The reference L^* range from black to white is zero to 100. For the BT.709 primaries typical of SD and HD, a^* and b^* are contained within the ranges [–87…+97] and [–108…+95] respectively, not including any undershoot, overshoot, or "illegal" or "invalid" C_BC_R values.

As in $L^*u^*v^*$, one unit of Euclidean distance in $L^*a^*b^*$ – denoted ΔE^*_{ab} – approximates the perceptibility of colour differences:

Eq 25.13

$$\Delta E^*_{ab} = \sqrt{\left(L^*_2 - L^*_1\right)^2 + \left(a^*_2 - a^*_1\right)^2 + \left(b^*_2 - b^*_1\right)^2}$$

If ΔE^*_{ab} is unity or less, the colour difference is taken to be imperceptible. However, $L^*a^*b^*$ does not achieve perceptual uniformity: It is merely an approximation.

A polar-coordinate version of the [a^*, b^*] pair can be used to express chroma and hue:

Eq 25.14

$$C^*_{ab} = \sqrt{a^{*2} + b^{*2}}; \qquad h_{ab} = \tan^{-1}\frac{b^*}{a^*}$$

The equations that form a^* and b^* coordinates are not projective transformations: Straight lines in $[x, y]$ do not transform to straight lines in $[a^*, b^*]$. The $[a^*, b^*]$ coordinates can be plotted in two dimensions, but such a plot is not a chromaticity diagram.

CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ summary

Both $L^*u^*v^*$ and $L^*a^*b^*$ improve the 80:1 or so perceptual nonuniformity of *XYZ* to perhaps 6:1. Both systems transform tristimulus values into a lightness component ranging from 0 to 100, and two colour components ranging approximately ±100. One unit of Euclidean distance in $L^*u^*v^*$ or $L^*a^*b^*$ corresponds roughly to a just noticeable difference (JND) of colour.

McCamy argues that under normal conditions 1,875,000 colours can be distinguished. See McCamy, Cam S. (1998), "On the number of discernible colors," in *Color Research and Application*, **23** (5): 337 (Oct.).

Consider that L^* ranges 0 to 100, and each of u^* and v^* range approximately ±100. A threshold of unity ΔE^*_{uv} defines four million colours. About one million colours can be distinguished by vision, so CIE $L^*u^*v^*$ is somewhat conservative. A million colours – or even the four million colours identified using a ΔE^*_{uv} or ΔE^*_{ab} threshold of unity – are well within the capacity of the 16.7 million colours available in a 24-bit truecolour system that uses perceptually appropriate transfer functions, such as the function of BT.709. (However, 24 bits per pixel are far short of the number required for adequate performance with linear-light coding.)

The $L^*u^*v^*$ or $L^*a^*b^*$ systems are most useful in colour specification. Both systems demand too much computation for economical realtime video processing, although both have been successfully applied to still image coding, particularly for printing. The complexity of the CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ calculations makes these systems generally unsuitable for image coding. The nonlinear $R'G'B'$ coding used in video is quite perceptually uniform, and has the advantage of being suitable for realtime processing. Keep in mind that $R'G'B'$ typically incorporates significant gamut limitation, whereas $L^*u^*v^*$ and CIE $L^*a^*b^*$ represent all colours. $L^*a^*b^*$ is sometimes used in desktop graphics with $[a^*, b^*]$ coordinates ranging from −128 to +127 (e.g., Photoshop). Even with these restrictions, CIE $L^*a^*b^*$ covers nearly all of the colours.

Colour specification and colour image coding

A colour specification system needs to be able to represent any colour with high precision. Since few colours are handled at a time, a specification system can be computationally complex. A system for colour specification must be intimately related to the CIE system. The systems useful for colour specification are CIE *XYZ* and its derivatives *xyY*, *u'v'*, $L^*u^*v^*$, and $L^*a^*b^*$.

A colour image is represented as an array of pixels, where each pixel contains three values that define a colour. As you have learned in this chapter, three components are necessary and sufficient to define any colour. (In printing it is convenient to add a fourth, black, component, giving CMYK.)

In theory, the three numerical values for image coding could be provided by a colour specification system; however, a practical image coding system needs to be computationally efficient, cannot afford unlimited precision, need not be intimately related to the CIE system, and generally needs to cover only a reasonably wide range of colours and not all possible colours. So image coding uses different systems than colour specification.

The systems useful for image coding are linear *RGB;* nonlinear *RGB* (usually denoted *R'G'B'*, with sRGB as one variant); nonlinear *CMY;* nonlinear *CMYK;* and derivatives of *R'G'B'*, such as $Y'C_BC_R$ and $Y'P_BP_R$. These systems are summarized in Figure 25.12.

If you manufacture cars, you have to match the paint on the door with the paint on the fender; colour specification will be necessary. You can afford quite a bit of computation, because there are only two coloured elements, the door and the fender. To convey a picture of the car, you may have a million coloured elements or more: Computation must be quite efficient, and an image coding system is called for.

Further reading

The bible of colorimetry is *Color Science*, by Wyszecki and Stiles. But it's daunting; it covers colour very generally, and contains no material specific to imaging.

For an approachable introduction to colour theory, accompanied by practical descriptions of image reproduction, consult Hunt's classic work.

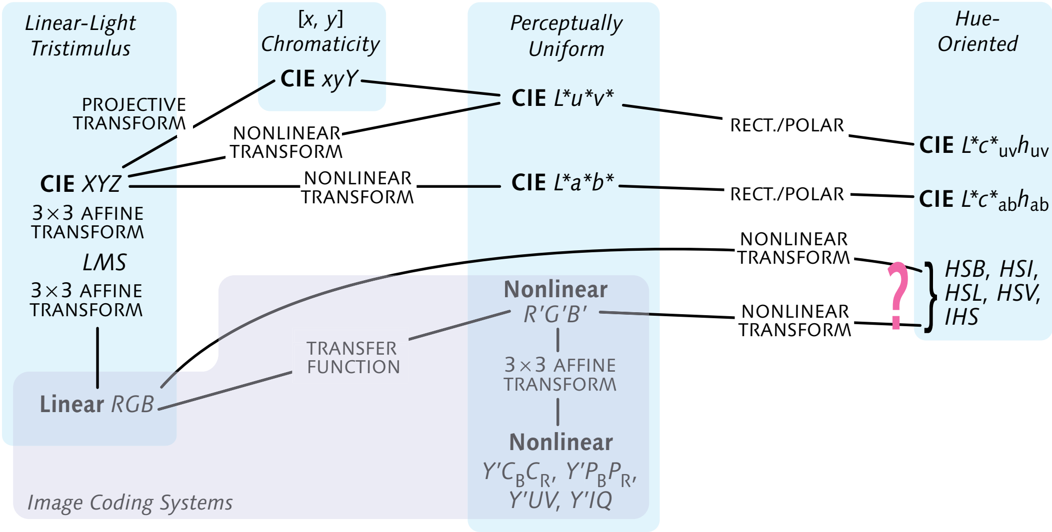


Figure 25.12 **Colour systems** are classified into four groups that are related by different kinds of transformations. Tristimulus systems, and perceptually uniform systems, are useful for image coding. (I flag *HSB*, *HSI*, *HSL*, *HSV*, and *IHS* with a question mark: These systems lack objective definition of colour.)

Berns' revision of the classic work by Billmeyer and Saltzman provides an excellent introduction to colour science. For an approachable, nonmathematical introduction to colour physics and perception, see Rossotti's book.

Wyszecki, Günter, and Stiles, W. Stanley (1982), *Color Science: Concepts and Methods, Quantitative Data and Formulæ*, Second Edition (New York: Wiley).

Hunt, Robert W. G., *The Reproduction of Colour*, Sixth Edition (Chichester, U.K.: Wiley, 2004).

Hunt, Robert W. G. and Pointer, Michael R. (2011), *Measuring Colour*, Fourth Edition (Chichester, U.K.: Wiley).

Berns, Roy S., (2000), *Billmeyer and Saltzman's Principles of Color Technology*, Third Edition (New York: Wiley).

Rossotti, Hazel (1983), *Colour: Why the World Isn't Grey* (Princeton, N.J.: Princeton Univ. Press).

Colour science for video 26

Classical colour science, explained in the previous chapter, establishes the basis for numerical description of colour. However, colour science is intended for the *specification* of colour, not for image coding. Although an understanding of colour science is necessary to achieve good colour performance in video, its strict application is impractical. This chapter explains the engineering compromises necessary to make practical cameras and practical coding systems.

Video processing is generally concerned with colour represented in three components derived from the scene, usually red, green, and blue, or components computed from these. Accurate colour reproduction depends on knowing exactly how the physical spectra of the original scene are transformed into these components, and exactly how the components are transformed to physical spectra at the display. These issues are the subject of this chapter.

Once red, green, and blue components of a scene are obtained, these components are transformed into other forms optimized for processing, recording, and transmission. This will be discussed in *Component video colour coding for SD,* on page 357, and *Component video colour coding for HD,* on page 369. (Although the BT.709 primaries are now used in both SD and HD, unfortunately, other colour coding aspects differ.)

The previous chapter explained how to analyze SPDs of scene elements into *XYZ* tristimulus values representing colour. The obvious way to present those colours is to arrange for the display system to reproduce those *XYZ* values. That approach works in many applications of colour reproduction, and it's the basis for colour in video. However, in image reproduction, direct recreation of the *XYZ* values is unsuitable for perceptual reasons. Some modifications are necessary to achieve subjectively acceptable results. Those modifications were described in *Constant luminance,* on page 107.

Should you wish to skip this chapter, remember that accurate description of colours expressed in terms of *RGB* coordinates depends on the characterization of the *RGB* primaries and their power ratios (white reference). If your system is standardized to use a fixed set of primaries throughout, as in SD and HD, you need not be concerned about different "flavours" of *RGB.* However, if your images have different primary sets in different stages or production – in digital cinema, or in digital still photography – it is a vital issue.

Additive reproduction (*RGB*)

In the previous chapter, I explained how a physical SPD can be analyzed into three components that represent colour. This section explains how those components can be mixed to present ("reproduce") colour.

So-called RGB+W displays were commercialized in the in the 1990s and early 2000s, mainly in colour-sequential DLP projectors. In an RGB+W display, the luminance of white is considerably greater than the sum of the luminances of red, green, and blue: High brightness is claimed; however, such displays do *not* exhibit additive colour mixture. As I write, virtually all presentations include pictorial imagery; customers demand proper colour portrayal, and *RGB+W* projectors have consequently fallen out of favour.

The simplest way to reproduce a range of colours is to mix the beams from three lights of different colours, as sketched in Figure 26.1 opposite. In physical terms, the spectra from each of the lights add together wavelength by wavelength to form the spectrum of the mixture. Physically and mathematically, the spectra add: The process is called *additive reproduction*.

I described Grassmann's Third Law on page 272: Colour vision obeys a principle of superposition, whereby the colour produced by any additive mixture of three primary SPDs can be predicted by adding the corresponding fractions of the *XYZ* tristimulus components of the primaries. The colours that can be formed from a particular set of *RGB* primaries are completely determined by the colours – tristimulus values, or luminance values and chromaticity coordinates – of the individual primaries. Subtractive reproduction, used in photography, cinema film, and commercial printing, is much more complicated: Colours in subtractive mixtures are not determined by the *colours* of the individual primaries, but by their *spectral* properties.

If you are unfamiliar with the term *luminance*, or the symbols *Y* or *Y'*, refer to *Luminance and lightness*, on page 255.

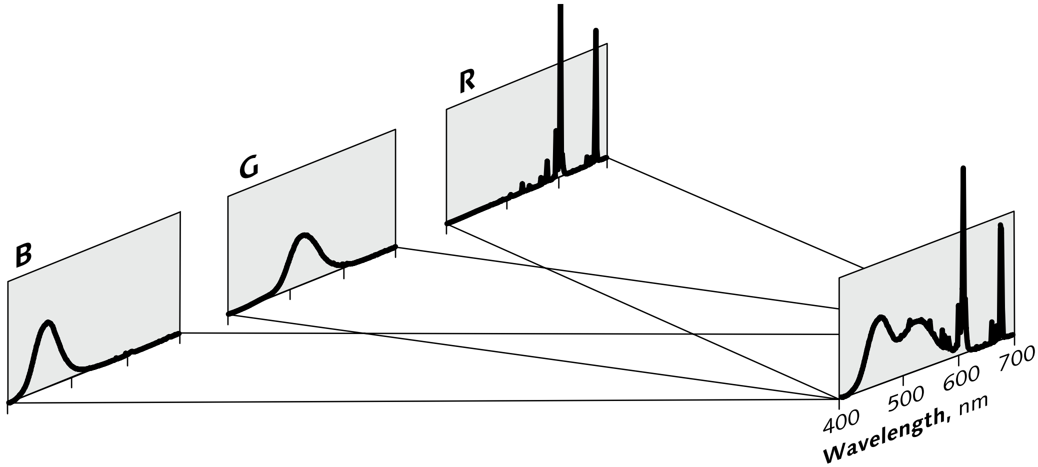


Figure 26.1 **Additive reproduction.** This diagram illustrates the physical process underlying additive colour mixture, as is used in video. Each primary has an independent, direct path to the image. The spectral power of the image is the sum of the spectra of the primaries. The colours of the mixtures are completely determined by the colours of the primaries; analysis and prediction of mixtures is reasonably simple. The SPDs shown here are those of a Sony Trinitron CRT.

Additive reproduction is employed directly in a video projector, where the spectra from a red beam, a green beam, and a blue beam are physically summed at the surface of the projection screen. Additive reproduction is also employed in a direct-view colour CRT, but through slightly indirect means. The screen of a CRT comprises small phosphor dots (triads) that, when illuminated by their respective electron beams, produce red, green, and blue light. When the screen is viewed from a sufficient distance, the spectra of these dots add in the lens and at the retina of the observer's eye.

The widest range of colours will be produced with primaries that individually appear red, green, and blue. When colour displays were exclusively CRTs, *RGB* systems were characterized by the chromaticities of their phosphors; we referred to *phosphor chromaticities*. To encompass newer devices that form colours without using phosphors, we now refer to *primary chromaticities* instead.

Three well chosen primaries can produce a large range of colours, but no finite set of primaries can cover all colours! An economic trade-off must be made that covers a wide range of colours with a very small number of primaries – preferably three.

Characterization of *RGB* primaries

An additive *RGB* system is specified by the chromaticities of its primaries and its white point. If you have an *RGB* image without information about its primary chromaticities, you cannot accurately reproduce the image. In Figure 26.2 opposite, I plot the primaries of a few *RGB* systems that I will discuss.

BT.709 specifies the primaries for HD. The BT.709 triangle is shaded in Figure 26.2.

The range of colours – or *gamut* – that can be formed from a given set of *RGB* primaries is given in the [*x*, *y*] chromaticity diagram by a triangle whose vertices are the chromaticities of the primaries. This two-dimensional plot doesn't tell the whole story, though: The range of [*x*, *y*] values that can be covered is a function of luminance. For example, BT.709's saturated blue colour at [0.15, 0.06] is only accessible at luminance below about 7% of white luminance; no chroma excursion is available at reference white! Gamut should be considered in three dimensions. I'll discuss gamut further on page 311.

In computing, the sRGB standard is now ubiquitous. The sRGB standard shares the BT.709 primaries. Many applications in desktop computing assume an sRGB interpretation unless other information accompanies the image.

The SMPTE/DCI P3 primaries that are standardized for D-cinema are overlaid on Figure 26.2.

Each of these systems will now be described in detail.

BT.709 primaries

ITU-R Rec. BT.709, *Parameter values for the HDTV standard for the studio and for international programme exchange*.

International agreement was obtained in 1990 by the former CCIR – now the ITU-R – on primaries for high-definition television (HD). The standard is formally denoted *Recommendation ITU-R BT.709* (formerly CCIR Rec. 709). I'll call it *BT.709*. Implausible though this sounds, the BT.709 chromaticities were agreed upon as the result of a political compromise that culminated in EBU red, EBU blue, and a green which is the average (rounded to 2 digits) of EBU green and SMPTE green! These primaries were adopted into the sRGB standard for computing and computer graphics. The BT.709 primaries are closely representative of contemporary displays in studio video. The chromaticities of the

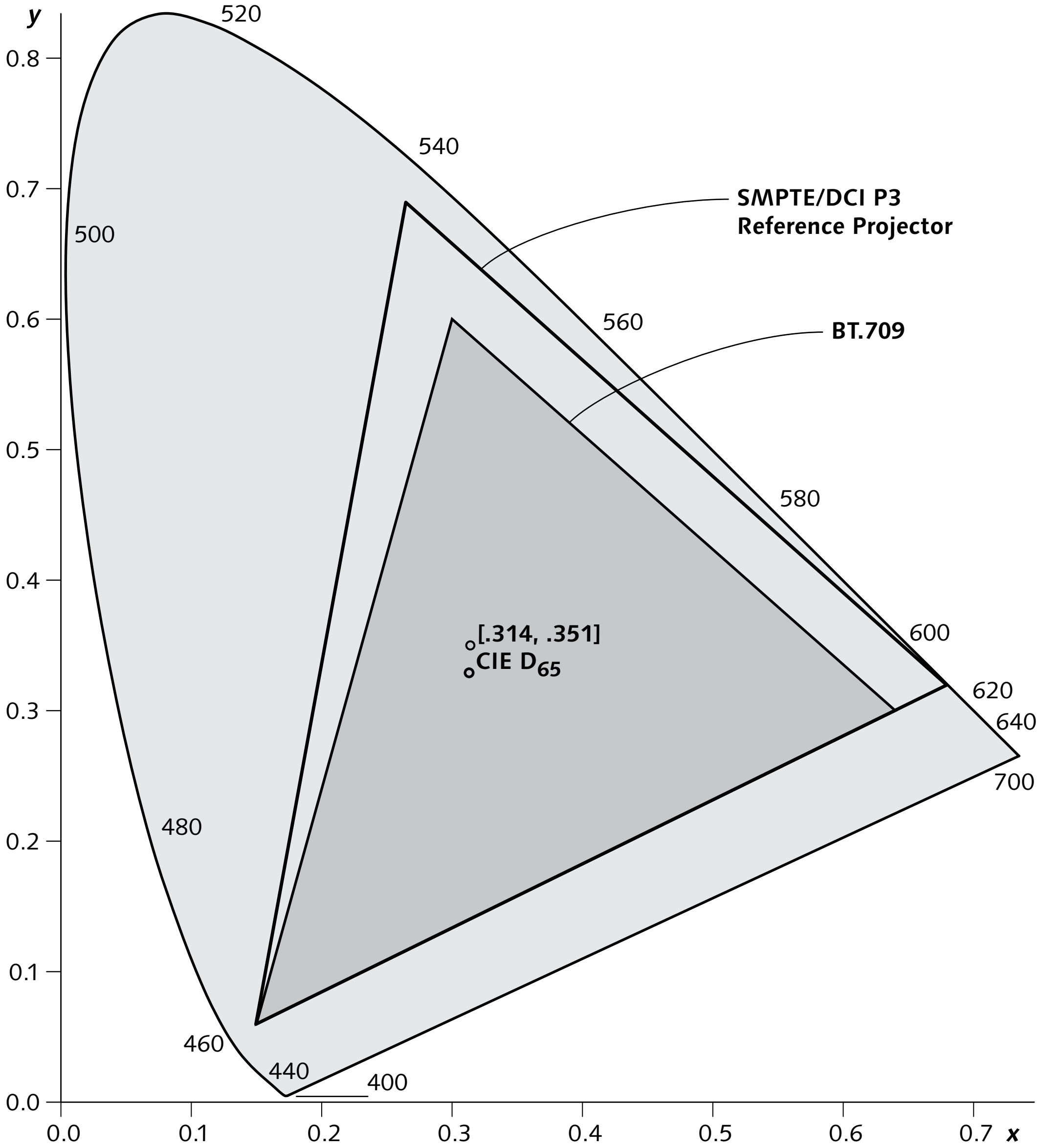


Figure 26.2 **The primaries of BT.709 and SMPTE/DCI P3** are compared. BT.709 is standard for HD worldwide, and is reasonably representative of SD; it incorporates the CIE D_{65} white point. The SMPTE/DCI P3 specification is used for D-cinema; its white point is [0.314, 0.351].

BT.709 primaries and its D_{65} white point are specified in Table 26.1:

Table 26.1 **BT.709 primaries** apply to 1280×720 and 1920×1080 HD systems; they are incorporated into the sRGB standard for desktop PCs.

	Red	*Green*	*Blue*	*White, D_{65}*
x	0.64	0.3	0.15	0.312727
y	0.33	0.6	0.06	0.329024
z	0.03	0.1	0.79	0.358249

Table 26.2 provides the relative luminance (*Y*) and [*x*, *y*] chromaticities of colourbars in BT.709 colour space:

	White	*Yellow*	*Cyan*	*Green*	*Magenta*	*Red*	*Blue*	*Black*
Y	1	0.927825	0.787327	0.715152	0.284848	0.212673	0.072175	0
x	0.312727	0.419320	0.224656	0.3	0.320938	0.64	0.15	*indeterminate*
y	0.329023	0.505246	0.328760	0.6	0.154190	0.33	0.06	*indeterminate*

Table 26.2 **Luminance and chromaticities of BT.709 colourbars**

The divisions by $X+Y+Z$ that form *x* and *y* effectively "explode" for a denominator of zero, reflected in the *indeterminate* entries for *x* and *y* of black in the table above. Black effectively covers the whole [*x*, *y*] diagram.

Video standards specify *RGB* chromaticities that are closely matched to practical displays. Physical display devices involve tolerances and uncertainties, but if you have a display that conforms to BT.709 within some tolerance, you can think of the display as being device-independent.

The importance of BT.709 as an interchange standard in studio video, broadcast television, and HD, and the firm perceptual basis of the standard, assures that its parameters will be used even by such devices as flat-panel displays that do not have the same physics as CRTs. However, there is no doubt that emerging display technologies will soon offer a wider colour gamut. SMPTE has adopted a standard for digital cinema that I will describe in a moment; that standard – SMPTE/DCI P3 – offers considerably wider gamut than BT.709. However, digital movies in their native P3 colour space are highly unlikely to be made available to consumers. IEC 61966-2-4 (xvYCC) purports to enable wide-gamut consumer video, but owing to the absence of any gamut-mapping mechanism I am highly skeptical concerning whether that claim will be realized by xvYCC.

Leggacy SD primaries

In 1953, the NTSC established primaries for emergent colour television. Those primaries were standardized by the FCC; they are now obsolete. RP 145 ("SMPTE-C") primaries have historically been used for SD in North America and Japan. EBU primaries have historically been used for SD in Europe. I detail these systems in Chapter 3, *Summary of obsolete RGB standards,* of *Composite NTSC and PAL: Legacy Video Systems*. The chromaticities of these sets of primaries are indicated below:

Table 26.3 **NTSC primaries (obsolete)** were used in 480*i* SD systems from 1953 until about 1970, when the primaries now documented in SMPTE RP 145 were adopted.

	Red	*Green*	*Blue*	*White CIE Ill. C*
x	0.67	0.21	0.14	0.310
y	0.33	0.71	0.08	0.316
z	0	0.08	0.78	0.374

Table 26.4 **EBU Tech. 3213 primaries** apply to 576*i* SD systems.

	Red	*Green*	*Blue*	*White,* D_{65}
x	0.640	0.290	0.150	0.3127
y	0.330	0.600	0.060	0.3290
z	0.030	0.110	0.790	0.3583

Table 26.5 **SMPTE RP 145 primaries** apply to 480*i* SD systems.

	Red	*Green*	*Blue*	*White,* D_{65}
x	0.630	0.310	0.155	0.3127
y	0.340	0.595	0.070	0.3290
z	0.030	0.095	0.775	0.3583

Interpreting RP 145 *RGB* values as BT.709 leads to colour errors of up to 20 ΔE, with an average error of about 14 ΔE. There are differences in primary chromaticities between the EBU standards and BT.709. Interpreting EBU *RGB* values as BT.709 leads to colour errors of up to 5 ΔE on the primaries, with an average error of about 2 ΔE.

Despite these differences between SD practice and BT.709, the BT.709 primaries are effectively being retrofitted into SD, albeit slowly.

sRGB system

IEC 61966-2-1, *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB.*

The sRGB standard is widely used in computing; the BT.709 primaries have been incorporated into sRGB. Beware that although the primary chromaticities are identical, the sRGB transfer function is somewhat different from the transfer functions standardized for studio video.

In the 1980s and 1990s, *RGB* image data in computing was commonly exchanged without information about primary chromaticities, white reference, or transfer function (to be described in *Gamma,* on page 315). If you have *RGB* image data without information about these parameters, you cannot accurately reproduce the image. The sRGB standard saw widespread deployment starting in about 2000, and today if you have an untagged RGB image it is safe to interpret the data according to sRGB.

SMPTE Free Scale (FS) primaries

SMPTE ST 2048-1, *2048×1080 and 4096×2160 Digital Cinematography Production Image Formats FS/709.* The notation *FS/709* is supposed to indicate that the standard applies either to imagery coded to FS-Gamut or to imagery coded to BT.709.

In 2011, SMPTE standardized a scheme termed *Free Scale Gamut* (FS-Gamut) that accommodates wide-gamut image data for 2 K and 4 K digital cinema production. (See *Free Scale Gamut, Free Scale Log,* on page 312.) A set of default primaries is defined. The primaries are nonphysical, so as to exceed the gamut of original camera negative film as normally processed and printed. The white point is D_{65}. The scheme is intended to encode scene-referred image data, typically using a quasilog coding specified in the same standard. The chromaticities of the default FS primaries and white are specified in Table 26.6:

Table 26.6 **SMPTE "Free Scale" default primaries** exceed the gamut of motion picture film as conventionally aquired, processed, and projected. Image data is scene-referred.

	Red	*Green*	*Blue*	*White*
x	0.7347	0.14	0.1	0.31272
y	0.2653	0.86	–0.02985	0.32903
z	0	0	0.92985	0.34065

AMPAS ACES primaries

AMPAS Specification S-2008-001, *Academy Color Encoding Specification (ACES).*

The Science and Technology Council (STC) of the Academy of Motion Picture Arts and Sciences (AMPAS) agreed upon a set of primaries for digital cinema acquisition and processing. ACES green and blue are nonphysical, so as to exceed the gamut of original

Some people will be surprised by the negative value of the ACES blue y coordinate: Adding ACES blue decreases luminance!

camera negative film as normally processed and printed. The standard has a white point approximating D_{60}. The system is intended to encode scene-referred image data. The chromaticities of the ACES primaries and white are specified in Table 26.7:

Table 26.7 **AMPAS ACES primaries** exceed the gamut of motion picture film as conventionally aquired, processed, and projected. ACES data is scene-referred.

	Red	*Green*	*Blue*	*White*
x	0.7347	0	0.0001	0.32168
y	0.2653	1	–0.0770	0.33767
z	0	0	1.0769	0.34065

SMPTE/DCI P3 primaries

SMPTE ST 428-1, *D-Cinema Distribution Master – Image Characteristics.*

Participants in the Digital Cinema Initiative (DCI) in the year 2000 achieved agreement upon primaries for a reference projector whose primaries approximately encompass the gamut of motion picture film as conventionally illuminated. Image data is display-referred. DCI's agreement has been promulgated as a SMPTE standard. The standard has a white point approximating current film practice; the tolerance on the white point specification encompasses D_{61}. The chromaticities of the DCI primaries and white are specified in Table 26.8:

Table 26.8 **SMPTE/DCI P3 primaries** approximately encompass the gamut of film, and are used for digital cinema. P3 data is display-referred.

	Red	*Green*	*Blue*	*White*
x	0.680	0.265	0.150	0.314
y	0.320	0.690	0.060	0.351
z	0	0.050	0.790	0.340

In establishing image data encoding standards for digital cinema, the DCI sought to make a "future-proof" standard that would accommodate wider colour gamut without the need to represent negative data values or data values above unity and without the necessity to maintain metadata. The standards adopted call for image data to be encoded in *XYZ* tristimulus values representing colours to be displayed. *XYZ* encoding amounts to using primaries having chromaticities [1, 0], [0, 1], and [0, 0], with a white reference of [$\frac{1}{3}$, $\frac{1}{3}$].

Despite DCI's adoption of *XYZ* coding, all digital cinema material mastered today is mastered to the DCI P3 gamut of the reference projector that I described above. I expect this situation to continue for the next 5 or 10 years. Issues of gamut mapping will eventually have to be addressed if new, wide-gamut material is to be sensibly displayed on legacy projectors. DCI standards are completely silent on issues of gamut mapping.

CMFs and SPDs

You might guess that you could implement a display whose primaries had spectral power distributions with the same shape as the CIE spectral analysis curves – the colour-matching functions for *XYZ*. You could make such a display, but when driven by *XYZ* tristimulus values, it would not properly reproduce colour. There are display primaries that reproduce colour accurately when driven by *XYZ* tristimuli, but the SPDs of those primaries do not have the same shape as the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ CMFs. To see why requires understanding a very subtle and important point about colour capture and reproduction.

To find a set of display primaries that reproduces colour according to *XYZ* tristimulus values would require constructing three SPDs that, when analyzed by the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ colour-matching functions, produced [1, 0, 0], [0, 1, 0], and [0, 0, 1], respectively. The $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ CMFs are positive across the entire spectrum. Producing [0, 1, 0] would require positive contribution from some wavelengths in the required primary SPDs, and that we could arrange; however, there is no wavelength that contributes to *Y* that does not also contribute positively to *X* or *Z*.

The solution to this dilemma is to force the *X* and *Z* contributions to zero by making the corresponding SPDs have negative power at certain wavelengths. Although this is not a problem for mathematics, or even for signal processing, an SPD with a negative lobe is not physically realizable in a transducer for light, because light power cannot go negative. So we cannot build a real display that responds directly to *XYZ*. But as you will see, the concept of negative SPDs – and *nonphysical SPDs* or *nonrealizable primaries* – is very useful in theory and in practice.

To understand the mathematical details of colour transforms, described in this section, you should be familiar with linear (matrix) algebra. If you are unfamiliar with linear algebra, see STRANG, GILBERT (1998), *Introduction to Linear Algebra*, Second Edition (Boston: Wellesley-Cambridge).

There are many ways to choose nonphysical primary SPDs that correspond to the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ colour-matching functions. One way is to arbitrarily choose three display primaries whose power is concentrated at three discrete wavelengths. Consider three display SPDs, each of which has some amount of power at 600 nm, 550 nm, and 470 nm. Sample the $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ functions of the matrix given earlier in *Calculation of tristimulus values by matrix multiplication*, on page 273, at those three wavelengths. This yields the tristimulus values shown in Table 26.9:

Table 26.9 **Example primaries** are used to explain the necessity of signal processing in accurate colour reproduction.

	Red, 600 nm	*Green, 550 nm*	*Blue, 470 nm*
X	1.0622	0.4334	0.1954
Y	0.6310	0.9950	0.0910
Z	0.0008	0.0087	1.2876

These coefficients can be expressed as a matrix, where the column vectors give the *XYZ* tristimulus values corresponding to pure red, green, and blue at the display, that is, [1, 0, 0], [0, 1, 0], and [0, 0, 1]. It is conventional to apply a scale factor in such a matrix to cause the middle row to sum to unity, since we wish to achieve only relative matches, not absolute:

Eq 26.1 This matrix is based upon *R*, *G*, and *B* components with unusual spectral distributions. For typical *R*, *G*, and *B*, see Eq 26.8.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.618637 & 0.252417 & 0.113803 \\ 0.367501 & 0.579499 & 0.052999 \\ 0.000466 & 0.005067 & 0.749913 \end{bmatrix} \bullet \begin{bmatrix} R_{600\,\mathrm{nm}} \\ G_{550\,\mathrm{nm}} \\ B_{470\,\mathrm{nm}} \end{bmatrix}$$

That matrix gives the transformation from *RGB* to *XYZ*. We are interested in the inverse transform, from *XYZ* to *RGB*, so invert the matrix:

Eq 26.2

$$\begin{bmatrix} R_{600\,\mathrm{nm}} \\ G_{550\,\mathrm{nm}} \\ B_{470\,\mathrm{nm}} \end{bmatrix} = \begin{bmatrix} 2.179151 & -0.946884 & -0.263777 \\ -1.382685 & 2.327499 & 0.045336 \\ 0.007989 & -0.015138 & 1.333346 \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The column vectors of the matrix in Equation 26.2 give, for each primary, the weights of each of the three discrete wavelengths that are required to display unit *XYZ* tristimulus values. The colour-matching functions for CIE *XYZ* are shown in Figure 26.3, *CMFs for CIE XYZ primaries*, on page 300. Opposite those functions, in Figure 26.4, is the corresponding set of primary SPDs. As expected, the display primaries have some negative spectral components: The primary SPDs are nonphysical. Any set of primaries that reproduces colour from

XYZ tristimulus values is necessarily *supersaturated*, more saturated than any realizable SPD could be.

To determine a set of physical SPDs that will reproduce colour when driven from *XYZ*, consider the problem in the other direction: Given a set of physically realizable display primaries, what CMFs are suitable to directly reproduce colour using mixtures of these primaries? In this case the matrix that relates *RGB* components to CIE *XYZ* tristimulus values is all-positive, but the CMFs required for analysis of the scene have negative portions: The analysis filters are nonrealizable.

Michael Brill and Bob Hunt agree that *R*, *G*, and *B* tristimulus values have no units. See HUNT, ROBERT W.G. (1997), "The heights of the CIE colour-matching functions," in *Color Research and Application*, **22** (5): 335–335 (Oct.).

Figure 26.6 shows a set of primary SPDs conformant to SMPTE 240M, similar to BT.709. Many different SPDs can produce an exact match to these chromaticities; the set shown is from a Sony Trinitron display. Figure 26.5 shows the corresponding colour-matching functions. As expected, the CMFs have negative lobes and are therefore not directly realizable; nonetheless, these are the idealized CMFs, or idealized *taking characterstics* – of the BT.709 primaries.

We conclude that we can use physically realizable analysis CMFs, as in the first example, where *XYZ* components are displayed directly. But this requires nonphysical display primary SPDs. Or we can use physical display primary SPDs, but this requires nonphysical analysis CMFs. As a consequence of the way colour vision works, there is no set of nonnegative display primary SPDs that corresponds to an all-positive set of analysis functions.

The escape from this conundrum is to impose a 3×3 matrix multiplication in the processing of the camera signals, instead of using the camera signals to directly drive the display. Consider these display primaries: monochromatic red at 600 nm, monochromatic green at 550 nm, and monochromatic blue at 470 nm. The 3×3 matrix of Equation 26.2 can be used to process *XYZ* values into components suitable to drive that display. Such signal processing is not just desirable; it is a necessity for achieving accurate colour reproduction!

To avoid the ambiguity of the term *pixel* – does it constitute one component or three? – I suggest that you call a sensor element a *photosite.*

In a "one-chip" camera, hardware or firmware performs spatial interpolation to reconstruct *R, G,* and *B* at each photosite. In a "three-chip" camera, dichroic filters are mounted on one or two glass blocks. In optical engineering, a glass block is called a prism, but it is not the prism that separates the wavelengths, it is the dichroic filters.

Kuniba, Hideyasu and Roy S. Berns, (2009), "Spectral sensitivity optimization of color image sensors considering photon shot noise," in *Journal of Electronic Imaging* **18** (2): 023002-1–023002-14.

Every colour video camera or digital still camera needs to sense the image through three different spectral characteristics. Digital still cameras and consumer camcorders typically have a single area array CCD sensor ("one chip"); each 2×2 tile of the array has sensor elements covered by three different types of filter. Typically, filters appearing red, green, and blue are used; the green filter is duplicated onto two of the photosites in the 2×2 tile. This approach loses light, and therefore sensitivity. Conventional studio video cameras separate incoming light using dichroic filters operating as beam splitters; each component has a dedicated CCD sensor ("3 CCD," or "3 CMOS"). Such an optical system separates different wavelength bands without absorbing any light, achieving about a factor of two higher sensitivity than a mosaic sensor.

Figure 26.7 shows the set of spectral sensitivity functions implemented by the beam splitter and filter ("prism") assembly of an actual HD camera. The functions are positive everywhere across the spectrum, so the filters are physically realizable. However, rather poor colour reproduction will result if these signals are used directly to drive a display having BT.709 primaries. Figure 26.8 shows the same set of camera analysis functions processed through a 3×3 matrix transform. The transformed components will reproduce colour more accurately – the more closely these curves resemble the ideal BT.709 CMFs of Figure 26.5, the more accurate the camera's colour reproduction will be.

In theory, and in practice, using a linear matrix to process the camera signals can capture *all* colours correctly. However, capturing all colours is seldom necessary in practice, as I will explain in the *Gamut* section below. Also, capturing the entire range of colours would incur a noise penalty, as I will describe in *Noise due to matrixing,* on page 308.

Normalization and scaling

The $\bar{y}(\lambda)$ CMF is standardized by the CIE such that its maximum value lies at unity. For the 10 nm CIE CMFs commonly used in image science, the $\bar{y}$ curve integrates to about 10.68. The $\bar{x}(\lambda)$ and $\bar{z}(\lambda)$ CMFs are scaled such that they integrate to the same value. The CIE derived its 10 nm CMFs by interpolation from its 1 nm curves;

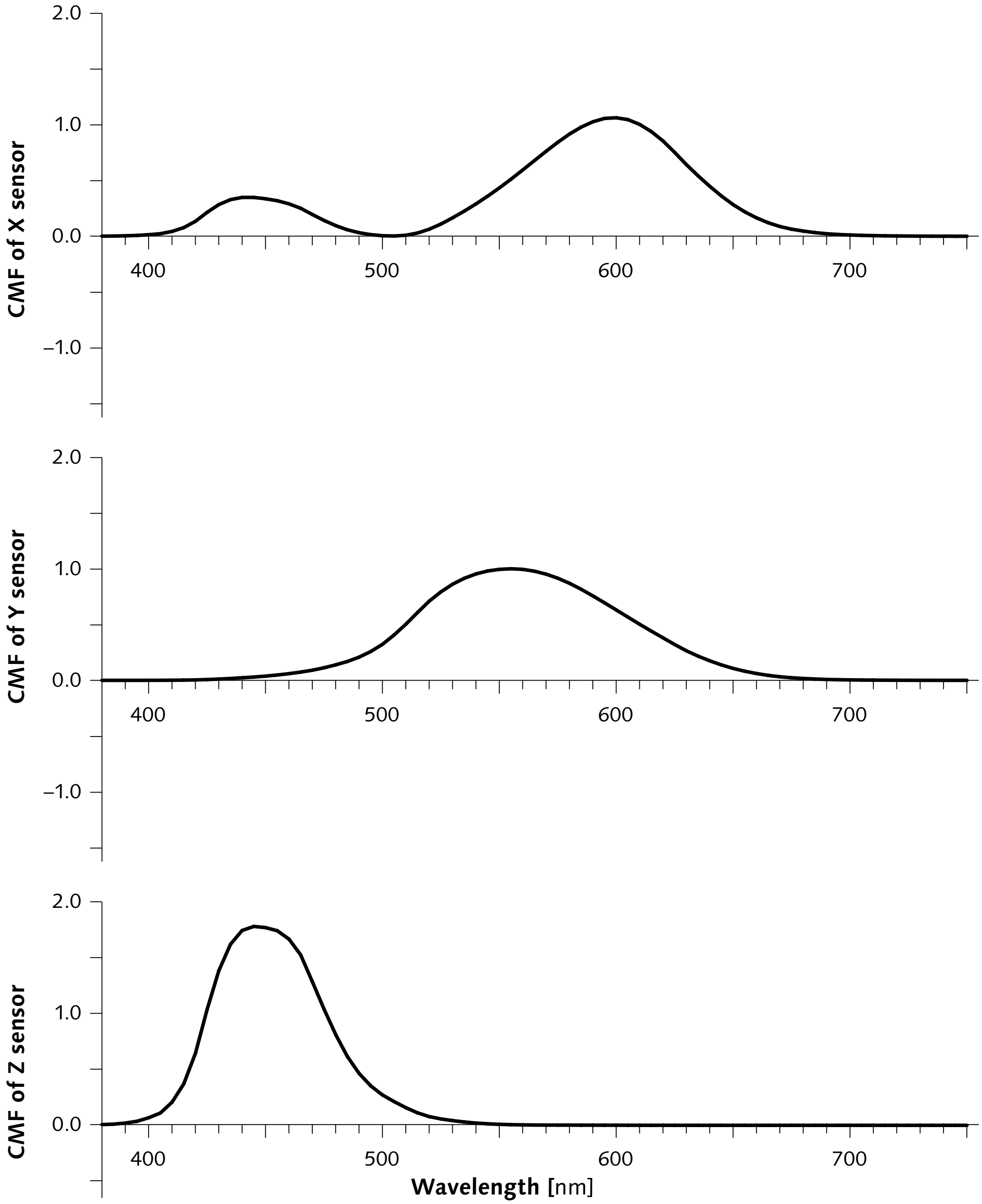


Figure 26.3 **CMFs for CIE *XYZ* primaries.** To acquire all colours in a scene requires filters having the CIE $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ spectral sensitivities. The functions are nonnegative, and therefore could be realized in practice. However, these functions are seldom used in actual cameras or scanners, for various engineering reasons.

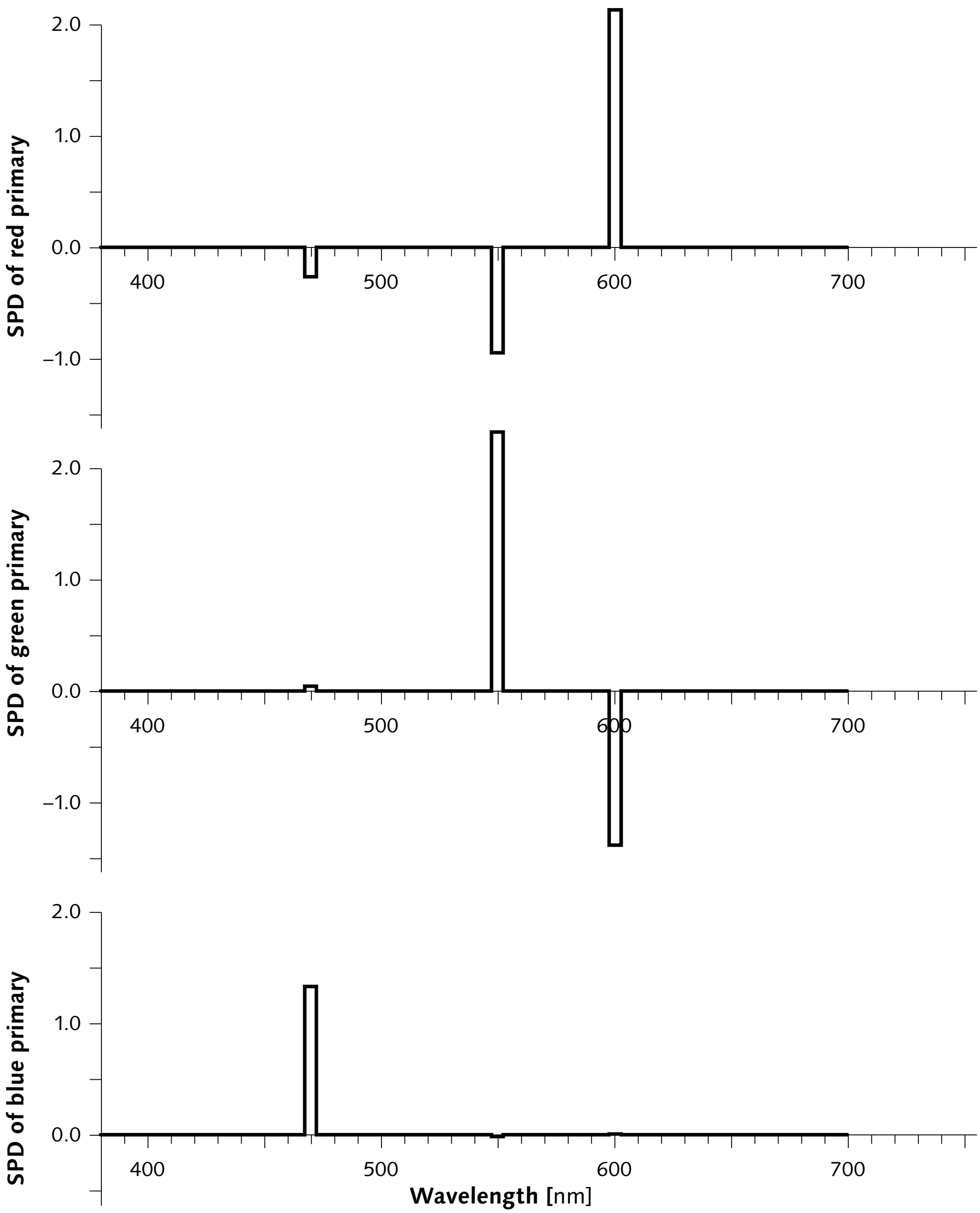


Figure 26.4 **SPDs for CIE *XYZ* primaries.** To directly reproduce a scene that has been analyzed using the CIE *XYZ* colour-matching functions requires *nonphysical* primaries having negative excursions, which cannot be realized in practice. Many different sets are possible. In this hypothetical example, the power in each primary is concentrated at the same three discrete wavelengths, 470, 550, and 600 nm.

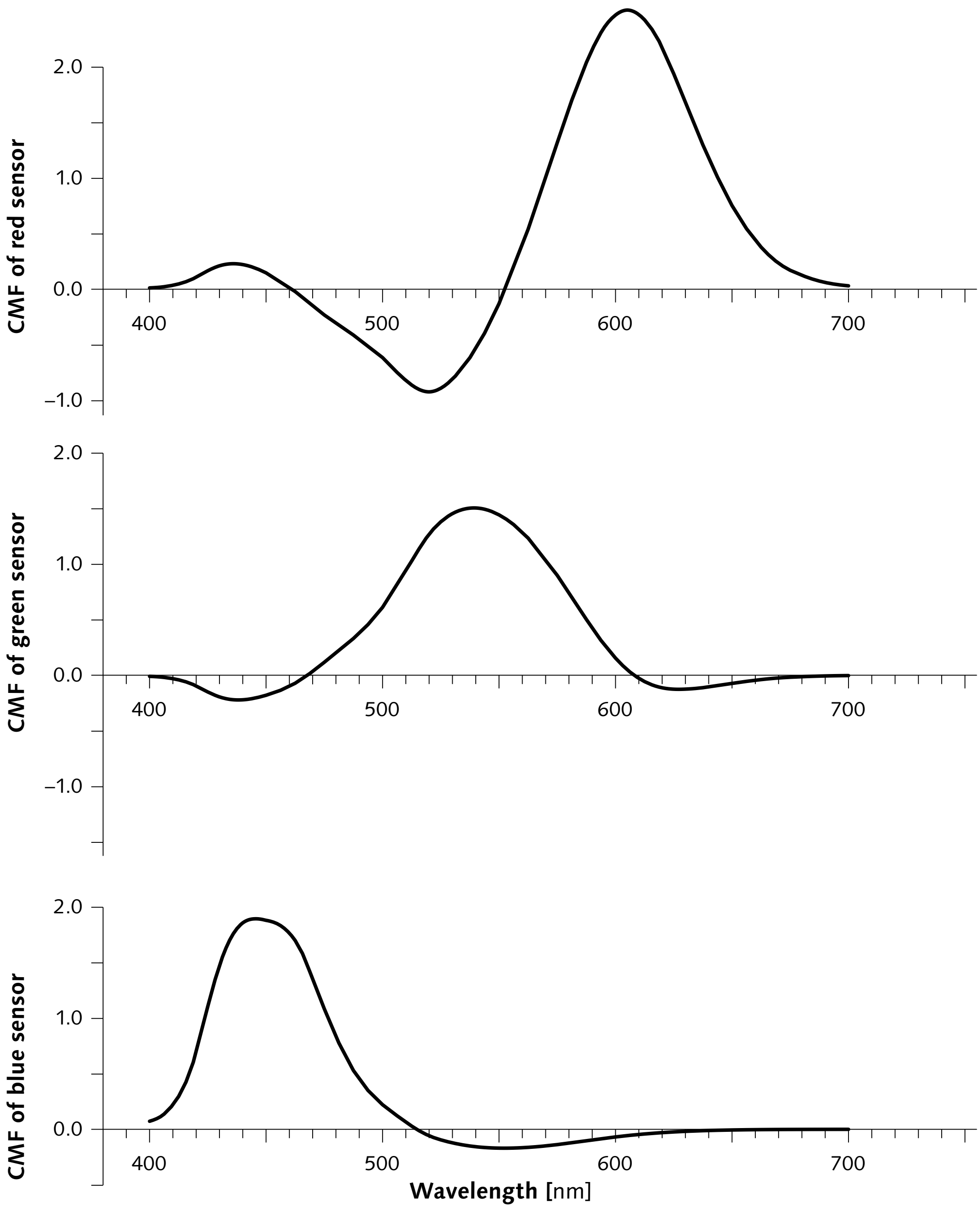


Figure 26.5 **CMFs for BT.709 primaries.** These analysis functions are theoretically correct to acquire *RGB* components for display using BT.709 primaries. The functions are not directly realizable in a camera or a scanner, due to their negative lobes; however, they can be realized by a 3×3 matrix transformation of the CIE *XYZ* colour-matching functions of Figure 26.3.

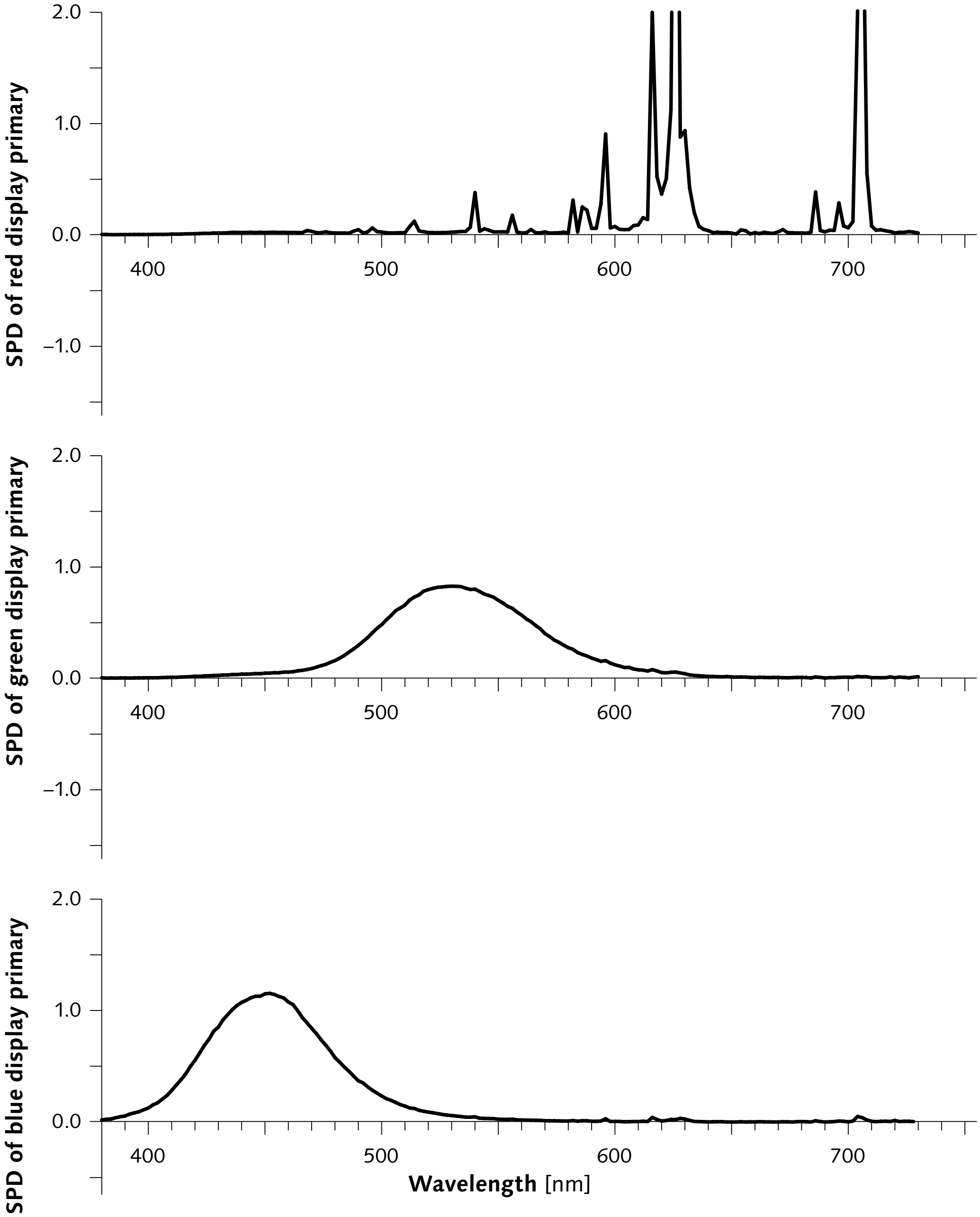


Figure 26.6 **SPDs for BT.709 display primaries.** This set of SPDs has chromaticity coordinates that conform to SMPTE RP 145, similar to BT.709. Many SPDs could produce the same chromaticity coordinates; this particular set is produced by a Sony Trinitron CRT display. The red primary uses *rare earth* phosphors that produce very narrow spectral distributions, different in character from the phosphors used for green or blue.

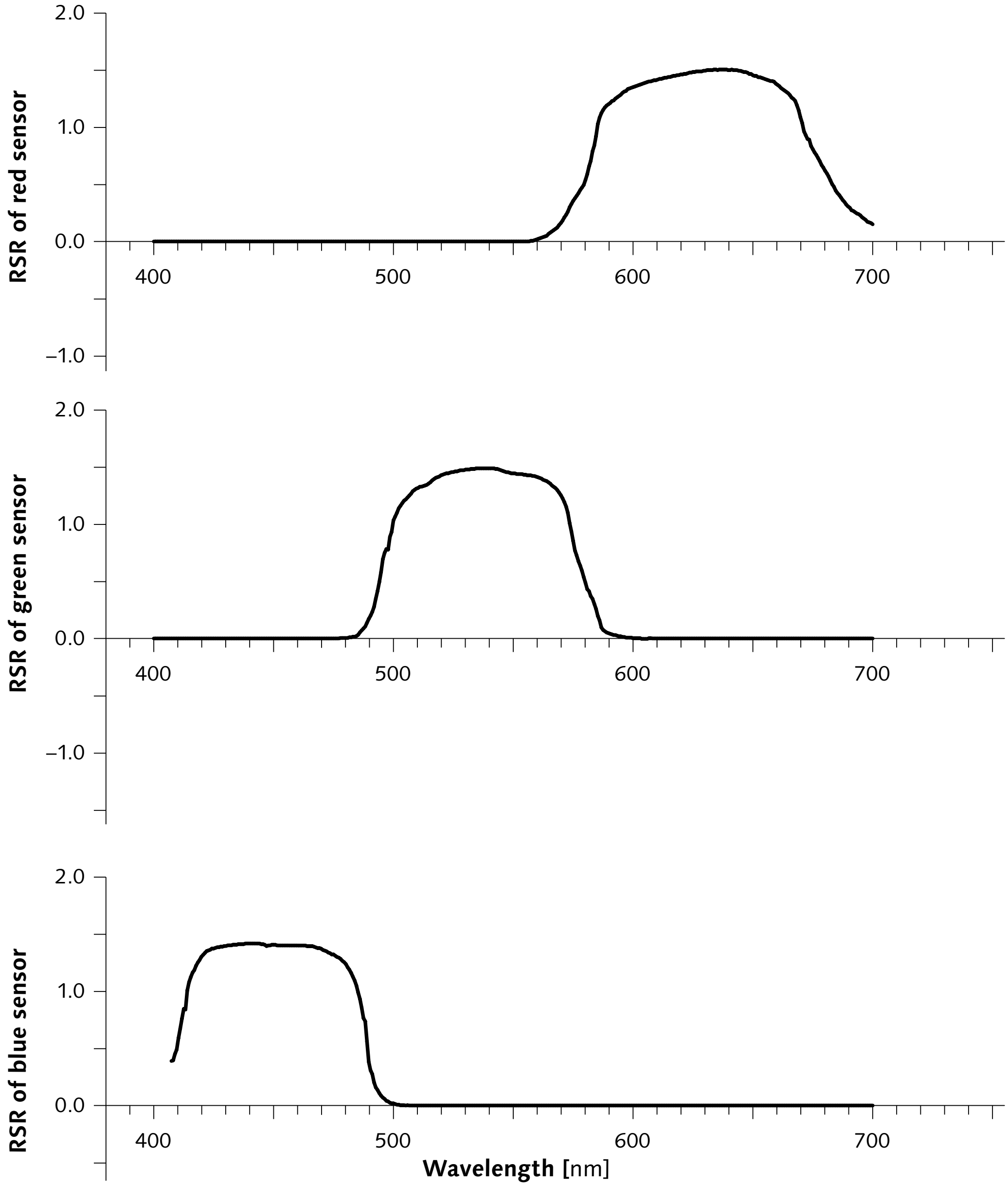


Figure 26.7 **Relative spectral responses (RSRs) for a real camera.** This set of spectral response functions is produced by the dichroic colour separation filters (*prism*) of a 2000-vintage beam-splitter CCD studio HD camera. I call these *relative spectral response* (RSR) functions.

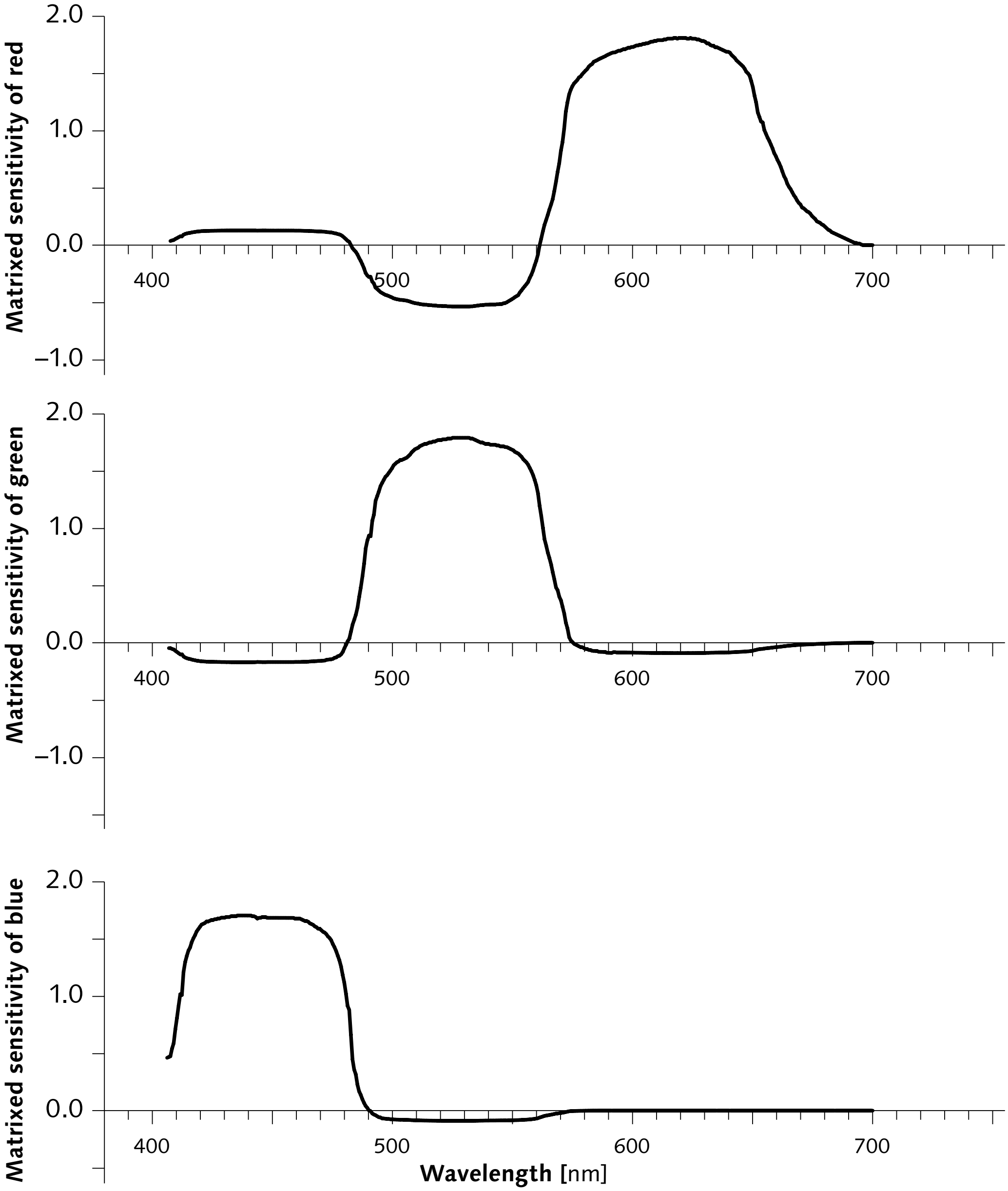


Figure 26.8 **Effective response after matrixing** for BT.709 primaries. These curves result from the analysis functions of Figure 26.7, opposite, being processed through a suitable 3×3 matrix. Colours as "seen" by this camera will be accurate to the extent that these curves match the ideal CMFs for BT.709 primaries shown in Figure 26.5.

the 10 nm $\bar{x}(\lambda)$ and $\bar{z}(\lambda)$ CMFs do not integrate to precisely the same value as $\bar{y}(\lambda)$ does.

CIE illuminants are specified scaled for unity at 560 nm. Illuminant E's SPD is represented as a vector of all ones. With the illuminant and CMF scaling specified by the CIE, Illuminant E's luminance is about 10.68.

In traditional practice, reference white has luminance of 100.

In applying the CIE standards to imaging, I find it convenient to normalize illuminants to a luminance of unity. With spectra represented as 31-element vectors from 400 nm to 700 nm at 10 nm intervals, Illuminant E then comprises a vector all of whose elements have the value $1/_{10.68}$, or about 0.0936.

Luminance coefficients

The luminance coefficients of any set of *XYZ* primary SPDs are [0, 1, 0]. Any *X* or *Z* primary is nonphysical; it has zero luminance.

Relative luminance can be formed as a properly weighted sum of *RGB* (linear-light) tristimulus components. The luminance coefficients can be computed starting with the chromaticities of the *RGB* primaries, here expressed in a matrix:

$$\boldsymbol{C} = \begin{bmatrix} x_R & x_G & x_B \\ y_R & y_G & y_B \\ z_R & z_G & z_B \end{bmatrix} \qquad \text{Eq 26.3}$$

Coefficients J_R, J_G, and J_B are computed from the chromaticities, and the white reference, as follows:

For the D_{65} white reference now standard in video, $\boldsymbol{C}^{-1}$ is multiplied by the vector [0.95, 1, 1.089].

$$\begin{bmatrix} J_R \\ J_G \\ J_B \end{bmatrix} = \boldsymbol{C}^{-1} \cdot \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} \cdot \frac{1}{y_W} \qquad \text{Eq 26.4}$$

Luminance can then be computed as follows:

$$Y = \begin{bmatrix} J_R y_R & J_G y_G & J_B y_B \end{bmatrix} \bullet \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{Eq 26.5}$$

This calculation can be extended to compute [*X*, *Y*, *Z*] from [*R*, *G*, *B*] of the specified chromaticity. First, compute a *normalized primary matrix* (NPM) denoted $\boldsymbol{T}$. The NPM depends upon the primaries and the white point of the [*R*, *G*, *B*] space:

$$\boldsymbol{T} = \boldsymbol{C} \bullet \begin{bmatrix} J_R & 0 & 0 \\ 0 & J_G & 0 \\ 0 & 0 & J_B \end{bmatrix} \qquad \text{Eq 26.6}$$

The elements J_R, J_G, and J_B of the diagonal matrix have the effect of scaling the corresponding columns of the chromaticity matrix, balancing the primary contributions to achieve the intended chromaticity of white. CIE tristimulus values [*X*, *Y*, *Z*] are then computed from the specified [*R*, *G*, *B*] as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{T} \bullet \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{Eq 26.7}$$

See *BT.601 luma* and *BT.709 luma*, on pages 346 and following.

As I explained in *Constant luminance*, on page 107, video systems compute luma as a weighted sum of *nonlinear R'G'B'* components. Although this calculation produces nonconstant-luminance (Livingston) errors, there is a second-order benefit in using the "theoretical" coefficients. The standard coefficients for SD are computed from the 1953 FCC NTSC primaries and CIE Illuminant C. The standard coefficients for HD are computed from BT.709 primaries and CIE D_{65}.

Transformations between *RGB* and CIE *XYZ*

RGB values in a particular set of primaries can be transformed to and from CIE *XYZ* by a 3×3 matrix transform. These transforms involve tristimulus values, that is, sets of three linear-light components that approximate the CIE colour-matching functions. CIE *XYZ* represents a special case of tristimulus values. In *XYZ*, any colour is represented by an all-positive set of values. SMPTE has standardized a procedure for computing these transformations.

SMPTE RP 177, *Derivation of Basic Television Color Equations*.

To transform from BT.709 *RGB* (with its D_{65} white point) into CIE *XYZ*, use the following transform:

Eq 26.8

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \bullet \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}$$

When constructing such a matrix for fixed-point calculation, take care when rounding to preserve the unity sum of the middle (luminance) row.

The middle row of this matrix gives the luminance coefficients of BT.709 (though BT.709 specifies four-digit values). Because white is normalized to unity, the middle row sums to unity. The column vectors are the *XYZ* tristimulus values of pure red, green, and blue. To recover primary chromaticities from such a matrix, compute x and y for each *RGB* column vector. To recover the white point, transform *RGB* = [1, 1, 1] to *XYZ*, then compute x and y according to Equation 25.1.

To transform from CIE *XYZ* into BT.709 *RGB*, invert the 3×3 matrix of Equation 26.8:

Eq 26.9

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Gamut is described on page 311.

This matrix has some negative coefficients: *XYZ* colours that are *out of gamut* for BT.709 *RGB* transform to *RGB* components where one or more components are negative or greater than unity.

Any *RGB* image data, or any matrix that purports to relate *RGB* to *XYZ*, should indicate the chromaticities of the *RGB* display primaries expected. If you encounter a matrix transform or image data without reference to any primary chromaticities, be very suspicious! Its originator may be unaware that *RGB* values must be associated with chromaticity specifications in order to have meaning for accurate colour.

Noise due to matrixing

Even if it were possible to display colours in the outer reaches of the chromaticity diagram, there would be a great practical disadvantage in doing so. Consider a camera that acquires *XYZ* tristimulus components, then transforms to BT.709 *RGB* according to Equation 26.9. The coefficient 3.240479 in the upper left-hand corner of the matrix in that equation determines the contribution from *X* at the camera into the red signal. An *X* component acquired with 4 digital codes of noise will inject 13 codes of noise into red: There is a noise penalty associated with the larger coefficients in the transform, and this penalty is quite significant in the design of a high-quality camera.

Transforms among *RGB* systems

The equations below transform between systems having the same illuminant. If the illuminant differs, a chromatic adaptation transform (CAT) may be required; in that case, a suitable 3×3 matrix (such as the Bradford transform) intervenes between $\boldsymbol{T}_D^{-1}$ and $\boldsymbol{T}_S$.

RGB values in a system employing one set of primaries can be transformed to another set by a 3×3 linear-light matrix transform. [*R*, *G*, *B*] tristimulus values in a source space (denoted with the subscript S) can be transformed into [*R*, *G*, *B*] tristimulus values in a destination space (denoted with the subscript D), using matrices $\boldsymbol{T}_S$ and $\boldsymbol{T}_D$ computed from the corresponding chromaticities and white points:

$$\begin{bmatrix} R_D \\ G_D \\ B_D \end{bmatrix} = \boldsymbol{T}_D^{-1} \bullet \boldsymbol{T}_S \bullet \begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} \qquad \text{Eq 26.10}$$

As an example, here is the transform from SMPTE RP 145 *RGB* (e.g., SMPTE 240M) to BT.709 *RGB*:

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 0.939555 & 0.050173 & 0.010272 \\ 0.017775 & 0.965795 & 0.016430 \\ -0.001622 & -0.004371 & 1.005993 \end{bmatrix} \bullet \begin{bmatrix} R_{145} \\ G_{145} \\ B_{145} \end{bmatrix} \qquad \text{Eq 26.11}$$

This matrix transforms EBU 3213 RGB to BT.709:

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 1.044036 & -0.044036 & 0 \\ 0 & 1 & 0 \\ 0 & 0.011797 & 0.988203 \end{bmatrix} \bullet \begin{bmatrix} R_{EBU} \\ G_{EBU} \\ B_{EBU} \end{bmatrix} \qquad \text{Eq 26.12}$$

To transform typical Sony Trinitron *RGB*, with D_{65} white reference, to BT.709, use this transform:

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 1.068706 & -0.078595 & 0.009890 \\ 0.024110 & 0.960070 & 0.015819 \\ 0.001735 & 0.029748 & 0.968517 \end{bmatrix} \bullet \begin{bmatrix} R_{SONY} \\ G_{SONY} \\ B_{SONY} \end{bmatrix} \qquad \text{Eq 26.13}$$

Transforming among *RGB* systems may lead to an *out of gamut RGB* result, where one or more *RGB* components are negative or greater than unity.

These transformations produce accurate results only when applied to tristimulus (linear-light) components. In principle, to transform nonlinear *R'G'B'* from one primary system to another requires application of the inverse transfer function to recover the tristimulus values, computation of the matrix multiplication, then reapplication of the transfer function. However, the transformation matrices of Equations 26.11, 26.12, and 26.13 are similar to the identity matrix: The diagonal terms are nearly unity, and the off-diagonal terms are nearly zero. In these cases, if the transform is computed in the nonlinear (gamma-corrected) *R'G'B'* domain, the resulting errors will be small.

Camera white reference

There is an implicit assumption in television that the camera operates as if the scene were illuminated by a source having the chromaticity of CIE D_{65}. In practice, television studios are often lit by tungsten lamps at around 3200 K, and scene illumination is often deficient in the shortwave (blue) region of the spectrum. This situation is compensated by *white balancing* – that is, by adjusting the gain of the red, green, and blue components at the camera so that a diffuse white object reports the values that would be reported if the scene illumination had the same tristimulus values as CIE D_{65}. In studio cameras, controls for white balance are available. In consumer cameras, activating WHITE BALANCE causes the camera to integrate red, green, and blue over the picture, and to adjust the gains so as to equalize the sums. (This approach to white balancing is sometimes called *integrate to grey*.)

Display white reference

SMPTE RP 166, *Critical Viewing Conditions for Evaluation of Color Television Pictures.*

EBU Tech. R23, *Procedure for the operational alignment of grade-1 colour picture monitors.*

In additive mixture, the illumination of the reproduced image is generated entirely by the display device. In particular, reproduced white is determined by the characteristics of the display, and is not dependent on the environment in which the display is viewed. In a completely dark viewing environment, such as a cinema theater, this is desirable; a wide range of chromaticities is accepted as "white." However, in an environment where the viewer's field of view encompasses objects other than the display, the viewer's notion of "white" is likely to be influenced or even dominated by what he or she perceives as "white" in the ambient. To avoid subjective mismatches, the chromaticity of white reproduced by the display and the chromaticity of white in the ambient should be reasonably close. SMPTE has standardized the chromaticity of reference white in studio displays. The standard specifies that luminance for reference white be reproduced at 120 $cd \cdot m^{-2}$, and surround conditions – basically, neutral grey at 10% of reference white – are outlined. In Europe, reference white luminance is specified in EBU R23 as 80 $cd \cdot m^{-2}$.

Modern blue CRT phosphors are more efficient with respect to human vision than red or green phosphors. Until recently, brightness was valued in computer displays more than colour accuracy. In a quest for a small brightness increment at the expense of a loss of colour accuracy, computer display manufacturers adopted a white point having a colour temperature of about 9300 K, producing a white having about 1.3 times as much blue as the standard CIE D_{65} white reference used in television. So, computer displays and computer pictures often look excessively blue. The situation can be corrected by adjusting or calibrating the display to a white reference with a lower colour temperature.

Studio video standards in Asia call for viewing with a 9300 K white reference. This practice apparently originates from a cultural preference regarding the portrayal of skin tones.

Gamut

Analyzing a scene with the CIE analysis functions produces distinct component triples for all colours. But when transformed into components suitable for a set of physical display primaries, some of those colours – those colours whose chromaticity coordinates lie outside the triangle formed by the primaries – will have negative component values. In addition, colours outside the triangle of the primaries may have one or two primary components that exceed unity. These colours cannot be correctly displayed. Display devices typically clip signals that have negative values and saturate signals whose values exceed unity. Visualized on the chromaticity diagram, a colour outside the triangle of the primaries is reproduced at a point on the boundary of the triangle.

If a camera is designed to capture all colours, its complexity is necessarily higher and its performance is necessarily worse than a camera designed to capture a smaller range of colours. Thankfully, the range of colours encountered in the natural and man-made world is a small fraction of all of the colours. Although it is necessary for an instrument such as a colorimeter to measure all colours, in an imaging system we are generally concerned with colours that occur frequently.

Pointer, Michael R. (1980), "The gamut of real surface colours," in *Color Research and Application* **5** (3): 143–155 (Fall).

M.R. Pointer characterized the distribution of frequently occurring *real surface colours*. The naturally occurring colours tend to lie in the central portion of the chromaticity diagram, where they can be encompassed by a well-chosen set of physical primaries. An imaging system performs well if it can display all or most of these colours. BT.709 does reasonably well; however, many of the colours of conventional offset printing – particularly in the cyan region – are not encompassed by all-positive BT.709 *RGB*. To accommodate such colours requires wide-gamut reproduction.

Wide-gamut reproduction

Poynton, Charles (2010), "Wide-gamut image capture," in *Proc. IS&T CGIV, Fourth European Conf. on Colour in Graphics and Imaging:* 471–482 (Joensuu, Finland).

For much of the history of colour television, cameras were designed to incorporate assumptions about the colour reproduction capabilities of colour CRTs. But nowadays, video production equipment is being used to originate images for a much wider range of applications than just television broadcast. The desire to make digital cameras suitable for originating images for this wider range of applications has led to proposals for video standards that accommodate a wider gamut.

Perhaps the first image coding system that accommodated linear-light (tristimulus) values below zero and above unity is described in Levinthal, Adam, and Thomas Porter (1984), "Chap: a SIMD graphics processor," in *Computer Graphics* **18** (3): 77–82 (July, *Proc. SIGGRAPH '84*).

The xvYCC ("x.v.Color") scheme is intended to be the basis for wide-gamut reproduction in future HD systems. The scheme is intended for use with *RGB* tristimulus values having BT.709 primaries, but with their range extended to –0.25 to +1.33, well outside the range 0 to 1. The excursions below zero and above unity allow *RGB* values to represent colours outside the triangle enclosed by the BT.709 primaries. When the extended *R'G'B'* values are matrixed, the resulting $Y'C_BC_R$ values lie within the "valid" range: Regions of $Y'C_BC_R$ space outside the "legal" *RGB* cube are exploited to convey wide-gamut colours.

Free Scale Gamut, Free Scale Log (FS-Gamut, FS-Log)

SMPTE ST 2048-1, *2048×1080 and 4096×2160 Digital Cinematography Production Image Formats FS/709.*

A recent SMPTE standard endorses wide-gamut imagery in production. "FS" stands for "Free Scale;" image data having arbitrary chromaticity can be conveyed. The standard uses the notation $R'_{FS}G'_{FS}B'_{FS}$ for wide-gamut colour components. The "709" component in the standard's title reflects the option to convey image data having BT.709 colorimetry. The default values for FS primaries reflect Sony "wide gamut" delivered by the F23, F35, and F65 cameras (see page 294). The standard provides no default values for the quasilog OECF.

ST 2048 contains many occurrences of "tristimulus value" where "chromaticity coordinate" is meant. Expect raised eyebrows among colour and image scientists.

The colour space is defined by the chromaticity coordinates of the primaries and white, and a parametricly defined quasilog OECF. Apart from toe and shoulder regions that are typically nonlinear, no provision is made for footroom or headroom. The standard does not specify how image data values are to be carried, but presumably more than 10 bits per component will be used (despite the quasilog).

The quasilog OECF is described by a set of four numerical parameters and a (fifth) "exposure" value k_{EXT}; $0 \le k_{EXT}$ indicates underexposure, $k_{EXT} = 1$ indicates correct exposure (default!), and $1 < k_{EXT}$ indicates overexposure.

Color VANC is pronounced *colour-VEE-ants.* The companion standard ST 2048-2 suggests placing Color VANC in the early portion of the active interval of line 18 in 1125-line interfaces.

The standard defines Color VANC, an ancillary data (ANC) packet carrying colour metadata – namely, the chromaticities of the primaries and reference white, the four parameters of the quasilog OECF function, k_{EXT}, and 12 numerical parameters concerned with the toe and knee (or shoulder) of the OECF. Presumably, DI ingest is expected to use the parameters carried by the Color VANC to construct a colour transform.

Further reading

For a highly readable short introduction to colour image coding, consult DeMarsh and Giorgianni. For a terse, complete technical treatment, read Schreiber.

For details of many aspects of colour imaging technology, consult either Kang (somewhat dated, now), or Sharma. For a discussion of nonlinear *RGB* in computer graphics, read Lindbloom's SIGGRAPH paper.

In a computer graphics system, once light is on its way to the eye, any tristimulus-based system can accurately represent colour. However, the interaction of light and objects involves spectra, not tristimulus values. In computer-generated imagery (CGI), the calculations actually involve sampled SPDs, even if only three samples (in this context, colour components) are used. Roy Hall discusses these issues.

DeMarsh, LeRoy E., and Edward J. Giorgianni (1989), "Color science for imaging systems," in *Physics Today:* 44–52 (Sep.).

Hall, Roy (1989), *Illumination and Color in Computer Generated Imagery* (New York: Springer).

Kang, Henry R. (1997), *Color Technology for Electronic Imaging Devices* (Bellingham, Wash.: SPIE).

Lindbloom, Bruce (1989), "Accurate color reproduction for computer graphics applications," in *Computer Graphics*, **23** (3): 117–126 (July).

Reinhard, Erik et al. (2008), *Color Imaging: Fundamentals and Applications* (Wellesley, Mass.: A K Peters).

Schreiber, William F. (1993), *Fundamentals of Electronic Imaging Systems,* Third Edition (Berlin: Springer-Verlag).

Sharma, Gaurav (2002), *Digital Color Imaging Handbook* (Boca Raton, Florida: CRC).

Gamma 27

In photography, video, and computer graphics, the *gamma* symbol (γ) represents a numerical parameter that estimates, in a single numerical parameter, the exponent of the assumed power function that maps from code (pixel) value to tristimulus value. Gamma is a mysterious and confusing subject, because it involves concepts from four disciplines: physics, perception, photography, and video. This chapter explains how gamma is related to each of these disciplines. Having a good understanding of the theory and practice of *gamma* will enable you to get good results when you create, process, and display pictures.

Luminance is proportional to intensity. For an introduction to the terms *brightness*, *intensity*, *luminance*, and *lightness*, see page 27. Further detail on luminance and lightness is found on page 255.

This chapter concerns the electronic display of images using video and computer graphics techniques and equipment. I deal mainly with the presentation of luminance, or, as a photographer would say, *tone scale*. Achieving good tone reproduction is one important step toward achieving good colour reproduction. (Other issues specific to colour reproduction were presented in the previous chapter, *Colour science for video*.)

Electro-optical conversion function (EOCF) refers to the function that characterizes conversion from the electrical signal domain into light, through some combination of signal processing and intrinsic display physics.

A *cathode-ray tube* (CRT) is inherently nonlinear: The luminance produced at the face of the display is a nonlinear function of each (R', G', and B') voltage input. From a strictly physical point of view, *gamma correction* at the camera can be thought of as precompensation for this nonlinearity in order to achieve correct reproduction of relative luminance.

Perceptual uniformity was introduced on page 8: Human perceptual response to luminance is quite nonuniform: The *lightness* sensation of vision is roughly the 0.42-power function of relative luminance. This nonlinearity needs to be considered if an image is to be coded to minimize the visibility of noise so as to make best perceptual use of a limited number of bits per pixel.

Combining the CRT nonlinearity (from physics), and lightness sensitivity (from perception) reveals an amazing coincidence: The nonlinearity of a CRT is remarkably similar to the *inverse* of the lightness sensitivity of human vision. Coding tristimulus value *RGB* into a gamma-corrected signal *R'G'B'* makes maximum perceptual use of each signal component. If gamma correction had not already been necessary for physical reasons at the CRT, we would have had to invent it for perceptual reasons. Modern displays such as LCDs and PDPs don't have CRT physics, but the CRT's nonlinearity has been replicated through signal processing.

Opto-electronic conversion function (OECF) refers to the transfer function in a scanner or camera that relates light power to signal code. In video, it's sometimes termed *opto-electronic transfer function*, OETF.

I will describe how video draws aspects of its handling of gamma from all of these areas: knowledge of the CRT from physics, knowledge of the nonuniformity of vision from perception, and knowledge of viewing conditions from photography.

Gamma in CRT physics

The electron gun of a CRT involves a theoretical relationship between voltage input and light output that a physicist calls a *five-halves power law:* Luminance produced at the face of the screen is in principle proportional to voltage input raised to the $5/2$ power. Luminance is roughly between the square and cube of the voltage. The numerical value of the exponent of this power function is represented by the Greek letter γ (gamma). CRT displays historically had behaviour that reasonably closely approximated this power function: Studio reference display CRTs have a numerical value of gamma quite close to 2.4.

Olson, Thor (1995), "Behind gamma's disguise," in *SMPTE Journal*, 104 (7): 452–458 (July).

Figure 27.1 opposite is a sketch of the power function that applies to the electron gun of a greyscale CRT, or to each of the red, green, and blue electron guns of a colour CRT. The three channels exhibit very similar, but not necessarily perfectly identical, responses.

The nonlinear voltage-to-luminance function of a CRT originates with the electrostatic interaction between the cathode, the grid, and the electron beam. The function is influenced to some extent by the mechanical structure of the electron gun. Contrary to

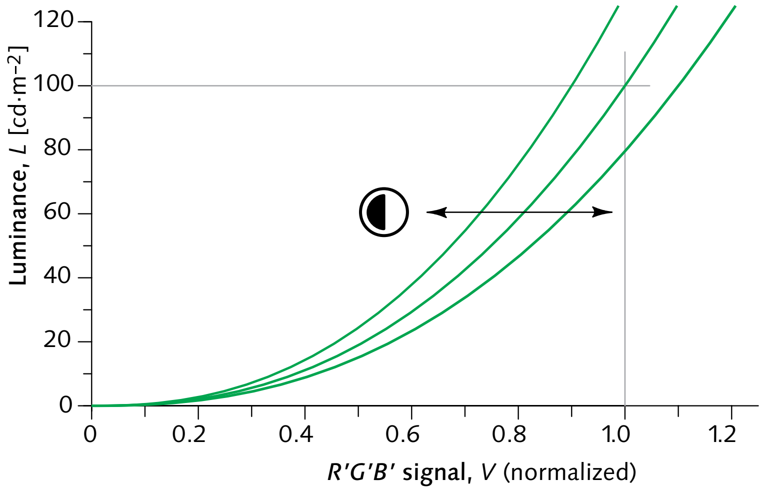


Figure 27.1 **Display electro-optical function (EOCF)** involves a nonlinear relationship between video signal and luminance, graphed here at GAIN settings of 0.9, 1.0, and 1.1 (effected by the poorly-named CONTRAST control). Luminance is approximately proportional to input signal voltage raised to the 2.4 power. The *gamma* of a display system – historically, that of a CRT – is the exponent of the assumed power function. Here the CONTRAST control is shown varying the gain of the video signal (on the *x*-axis), the way it's usually implemented; however, owing to the mathematical properties of a power function, scaling the luminance output would yield the same effect.

popular opinion, CRT phosphors themselves are quite linear, at least up to about eight-tenths of peak luminance. I denote the exponent the *decoding gamma,* γ_D.

Roberts, Alan (1993), "Measurement of display transfer characteristic (gamma, γ)," in *EBU Technical Review* 257: 32–40 (Autumn).

The value of decoding gamma (γ_D) for a typical, properly adjusted CRT in a studio environment ranges from about 2.3 to 2.4. Computer graphics practitioners sometimes claim numerical values of gamma wildly different from 2.4; however, such measurements often disregard two issues. First, the largest source of variation in the nonlinearity of a display is careless setting of the BRIGHTNESS (or BLACK LEVEL) control. Before a sensible measurement of gamma can be made, this control must be adjusted, as outlined on page 56, so that black-valued pixels are correctly displayed. Second, computer systems often have lookup tables (LUTs) that effect control over transfer functions. A gamma value dramatically different from 2.4 is often due to the function loaded into the LUT. For example, Macintosh computers prior to 2009 were said to have a gamma of 1.8; however, that value was a consequence of the default Macintosh LUT, not the Macintosh display itself (which has gamma between about 2.2 and 2.4).

In Mac OS X operating system version 10.6 ("*Snow Leopard*"), released in 2009, Apple adopted a default gamma of 2.2*: R'G'B'* values presented to the graphics subsystem are now interpreted as sRGB by default.

Understanding CRT physics is an important first step toward understanding gamma, but it isn't the whole story.

The amazing coincidence!

In *Luminance and lightness,* on page 255, I described the nonlinear relationship between luminance (a physical quantity) and lightness (a perceptual quantity): Lightness is approximately luminance raised to the 0.42-power. The previous section described how the nonlinear transfer function of a CRT relates a voltage signal to luminance. Here's the surprising coincidence:

> A CRT's signal-to-luminance function is very nearly the *inverse* of the luminance-to-lightness relationship of human vision.

In analog systems, we represent lightness information as a voltage, to be transformed into luminance by a CRT's power function. Digital systems simply digitize analog voltage. To minimize the perceptibility of noise, we use a perceptually uniform code. Amazingly, the CRT function is a near-perfect inverse of vision's lightness sensitivity: CRT voltage is effectively a perceptually uniform code! In displays such as LCDs and PDPs, we impose signal processing to mimic CRT behaviour.

Gamma in video

Many video engineers are unfamiliar with colour science. They consider only the first of these two purposes, and disregard, or remain ignorant of, the great importance of perceptually uniform coding.

In a video system, "gamma correction" is applied at the camera for the dual purposes of precompensating the nonlinearity of the display's CRT and coding into perceptually uniform space. Figure 27.2 summarizes the image reproduction situation for video. At the left, gamma correction is imposed at the camera; at the right, the display imposes the inverse function.

Coding into a perceptual domain was important in the early days of television because of the need to minimize the noise introduced by over-the-air analog transmission; the same considerations of noise visibility applied to analog videotape recording. These considerations also apply to the quantization error that is introduced upon digitization, when a linear-light signal is quantized to a limited number of bits. Consequently, it is universal to convey video signals in gamma-corrected form.

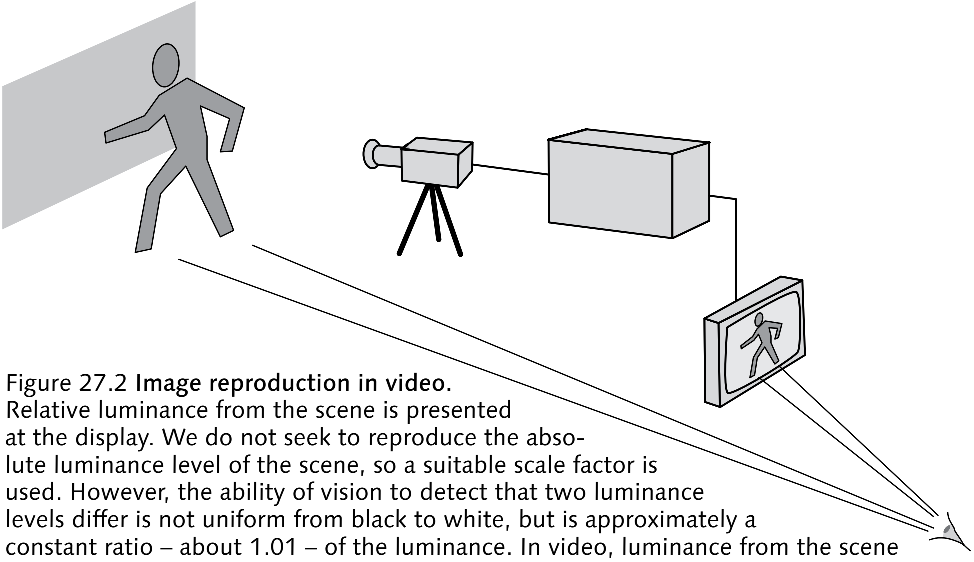


Figure 27.2 **Image reproduction in video.** Relative luminance from the scene is presented at the display. We do not seek to reproduce the absolute luminance level of the scene, so a suitable scale factor is used. However, the ability of vision to detect that two luminance levels differ is not uniform from black to white, but is approximately a constant ratio – about 1.01 – of the luminance. In video, luminance from the scene is transformed by a function similar to a square root into a nonlinear, perceptually uniform signal that is transmitted. The camera is designed to mimic the human visual system, in order to “see” lightness in the scene the same way that a human observer would. Noise introduced by the transmission system then has minimum perceptual impact. The nonlinear signal is transformed back to luminance at the display. In a CRT, a 2.4-power function is intrinsic; in other display technologies, a comparable power function is included in the signal processing.

Gamma correction is ordinarily based upon a *power* function, which has the form $y = x^a$ (where a is constant). Gamma correction is sometimes incorrectly claimed to be an *exponential* function, which has the form $y = a^x$ (where a is constant).

Gamma correction is unrelated to the gamma function $\Gamma(\cdot)$ of mathematics.

In a video camera, we precompensate for the CRT's nonlinearity by processing each of the *R, G,* and *B* tristimulus signals through a nonlinear transfer function. This process is known as gamma correction. The function required is approximately a square root. The curve is often not precisely a power function; nonetheless, I denote the best-fit exponent the *encoding gamma,* γ_E. In video, gamma correction is accomplished by analog (or sometimes digital) circuits at the camera. In computer graphics, gamma correction is usually accomplished by incorporating the nonlinear transfer function into a framebuffer's lookup table.

The importance of picture rendering, and the consequent requirement for different exponents for encoding (γ_E) and decoding (γ_D), have been poorly recognized and poorly documented in the development of video.

As explained in *Picture rendering,* on page 115, it is important for perceptual reasons to alter the tone scale of an image presented at a luminance substantially lower than that of the original scene, presented with limited contrast ratio, or viewed in a dim surround. The dim surround condition is characteristic of television viewing. In video, the alteration is accomplished at the camera by slightly undercompensating the actual power function of the CRT, to obtain an end-to-end power

Eq 27.1

$\gamma_E \approx 0.5; \gamma_D \approx 2.4;$

$\gamma_E \cdot \gamma_D \approx 1.2$

function whose exponent is about 1.2, as indicated in Equation 27.1 in the margin. This undercompensation achieves end-to-end reproduction that is subjectively correct (though not mathematically linear).

Opto-electronic conversion functions (OECFs)

What I call OECF, in accordance with the nomenclature of ISO 14524, is often called *opto-electronic transfer function,* OETF, in historical video literature.

Several different transfer functions have been standardized and are in use. In the sections to follow, I will detail these standards:

- BT.709 is an international standard that specifies the basic parameters of HD. Although intended for HD, it is representative of current SD technology, and it is being retrofitted into SD studio standards.

BT.1361 was established by ITU-R but never deployed. It is now moribund, superseded by xvYCC.

- The xvYCC "standard" extends $Y'C_BC_R$ and $Y'P_BP_R$ coding to accommodate a wide colour gamut. As I write, xvYCC is not deployed.
- sRGB refers to the standard transfer function of PCs.
- The transfer function of the original 1953 NTSC specification, often written $1/_{2.2}$, has been effectively superseded by BT.1886.
- The transfer function of European standards for 576*i* is often given as $1/_{2.8}$. Professional encoding has never expected a decoding gamma as high as 2.8. In any event, that value has been effectively superseded by BT.1886.

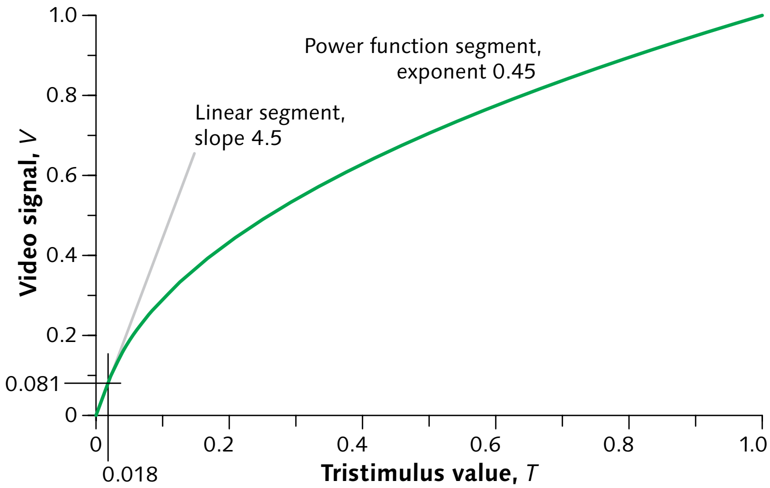
It is unclear from historical documents whether the classic NTSC 2.2 "gamma" and the classic EBU 2.8 "gamma" were intended to define the camera or the display! In entertainment imaging, the content creator has licence to manipulate image data at acquisition and at postproduction to yield the intended picture appearance, potentially completely independently of any standard OECF at a camera. The standard *EOCF* predominates: The EOCF establishes how image data is to be displayed in a manner faithful to the content creation process. The standard camera OECFs merely serve as engineering guidelines.

BT.709 OECF

ITU-R Rec. BT.709, *Basic parameter values for the HDTV standard for the studio and for international programme exchange*.

Figure 27.3 illustrates the transfer function defined by the international BT.709 standard for high-definition television (HD). It is based upon a pure power function with an exponent of 0.45. Theoretically, a pure power function suffices for gamma correction; however, the

Figure 27.3 **BT.709 OECF** is standardized as the reference mapping from scene tristimulus to video code in SD and HD.



slope of a pure power function (whose exponent is less than unity) is infinite at zero. In a practical system such as a video camera, in order to minimize noise in dark regions of the picture it is necessary to limit the slope (gain) of the function near black. BT.709 specifies a slope of 4.5 below a tristimulus value of +0.018. The pure power function segment of the curve is scaled and offset to maintain function and tangent continuity at the breakpoint.

The symbol *T* suggests tristimulus value; the same equation applies to *R*, *G* or *B*. The symbol *V* suggests *voltage*, or *video*, or [code/pixel] *value*. I write this unprimed.

Reference BT.709 encoding is as follows. The tristimulus (linear light) component is denoted *T*, and the resulting gamma-corrected video signal – one of *R'*, *G'*, or *B'* components – is denoted with a prime symbol, V_{709}. *R*, *G*, and *B* are processed through identical functions to obtain *R'*, *G'*, and *B'*:

$$V_{709} = \begin{cases} 4.5T; & 0 \le T < 0.018 \\ 1.099T^{0.45} - 0.099; & 0.018 \le T \le 1 \end{cases} \qquad \text{Eq 27.2}$$

The reference BT.709 encoding equation includes an exponent of 0.45. I call this the “advertised” exponent. Some people describe BT.709 as having “gamma of 0.45"; broadcast video camera GAMMA controls are calibrated in terms comparable to this value. However, the effect of the scale factor and offset terms make the overall power function very similar to a square root ($\gamma_E \approx 0.5$); the *effective* power function exponent – and the value appropriate for picture rendering calculations – is 0.5.

BT.709 encoding assumes that encoded *R'G'B'* signals will be converted to tristimulus values at a display with an EOCF close to a pure 2.4-power function:

$$T = V^{2.4} \qquad \text{Eq 27.3}$$

The product of the effective 0.5 exponent typically used at the camera and the 2.4 exponent at the display produces an end-to-end power of about 1.2, suitable for material acquired in a bright environment for display in a typical television viewing situation, as I explained in *Picture rendering,* on page 115. In 2011, ITU-R adopted BT.1886, which specifies a 2.4-power function EOCF for HD; see *Reference display and viewing conditions,* on page 427. Unfortunately, reference white luminance and viewing conditions aren't standardized.

To recover *RGB* values proportional to scene tristimulus values, *assuming that the camera was operated with "factory"* BT.709 settings, invert Equation 27.2:

$$T = \begin{cases} \dfrac{V_{709}}{4.5}; & 0 \le V_{709} < 0.081 \\ \left(\dfrac{V_{709} + 0.099}{1.099} \right)^{\frac{1}{0.45}}; & 0.081 \le V_{709} \le 1 \end{cases} \qquad \text{Eq 27.4}$$

Equation 27.4 is very similar to a square root. It does not incorporate correction for picture rendering: Recovered values are proportional to the *scene* tristimulus values, not to the intended *display* tristimulus values. BT.709 is misleading in its inclusion of this equation without discussing – or even mentioning – the issue of picture rendering.

For details of quantization to 8- or 10-bit components, see *Studio-swing (footroom and headroom),* on page 42.

SMPTE 240M OECF

SMPTE 240M, *1125-Line High-Definition Production Systems – Signal Parameters*.

SMPTE Standard 240M for 1125/60, 1035*i*30 HD was adopted two years before BT.709; virtually all HD equipment deployed in the decade 1988 to 1998 used the its parameters. For details, refer to the first edition of this book. The OECF specified in SMPTE 240M is intended to be used with a display EOCF comparable to that standardized (much later) in BT.1886.

sRGB transfer function

IEC 61966-2-1, *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB*.

The notation *sRGB* refers to a specification for colour image coding for personal computing, desktop graphics, and image exchange on the Internet.

The sRGB specificaton provides that a display will convert encoded *R'G'B'* signals using an EOCF that is a pure 2.2-power function.

$\gamma_E \approx 0.45 \approx \frac{1}{2.22}$

$\gamma_D \approx 2.4$

$0.45 \cdot 2.4 \approx 1.1$

See *Picture rendering*, on page 115.

The sRGB specification anticipates a higher ambient light level for viewing than typical broadcast studio practice associated with BT.709 encoding. Imagery originated with BT.709 encoding, displayed on a display with a 2.2-power, results in an end-to-end power of 1.1, considerably lower than the 1.2 end-to-end power produced by BT.709 encoding, but appropriate for the high display luminance, light surround, and poor contrast ratio typical of sRGB display environments.

The sRGB specification includes a function that ostensibly defines an OECF:

$$V_{\mathrm{sRGB}} = \begin{cases} 12.92T; & 0 \le T \le 0.0031308 \\ 1.055T^{\left(\frac{1}{2.4}\right)} - 0.055; & 0.0031308 < T \le 1 \end{cases} \qquad \text{Eq 27.5}$$

The standard is not explicit about the use of this function. Evidently it maps linear-light values to sRGB codes, and it includes a linear segment near black that you would expect in an OECF. The function resembles the BT.709 OECF. However, no account is taken of picture rendering. I conclude – and section 5.1 of the standard implies – that the function is intended to describe the mapping from the tristimulus values *presented on the display* to sRGB codes; in other words, sRGB coding is display referred. The encoding specified by sRGB is *inappropriate* when picture rendering is to be applied at the time of image capture – for example, when capturing a scene with a digital camera. For the latter purpose, BT.709 coding is appropriate.

Stokes, Michael, Matthew Anderson, Srinivasan Chandrasekar, and Ricardo Motta (1996), A *Standard Default Color Space for the Internet – sRGB* http://www.w3.org/Graphics/Color/sRGB.

Although Equation 27.5 contains the exponent $^1/_{2.4}$, which suggests "gamma of 0.42," the scale factor and the offset cause the overall function to approximate a pure 0.45-power function ($\gamma_E \approx 0.45$). It is misleading to describe sRGB as having "gamma of 0.42."

It is standard to code sRGB components in 8-bit form from 0 to 255, with no footroom and no headroom.

Figure 27.4 **BT.709, sRGB, and CIE *L** encoding functions** are compared. They are all approximately perceptually uniform; however, they are not sufficiently close to be interchangeable.

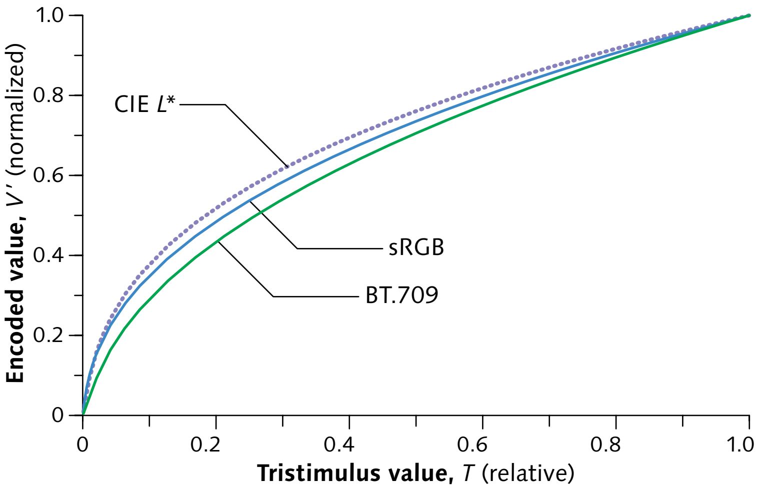


Figure 27.4 sketches the sRGB encoding function, overlaid on the BT.709 encoding and CIE *L** functions.

Transfer functions in SD

Historically, transfer functions for SD have been very poorly specified. The FCC NTSC standard adopted in 1953 referred to a "transfer gradient (gamma exponent) of 2.2." It isn't clear whether 2.2 was intended to characterize the camera's OECF or the display's EOCF. In any event, modern CRTs have power function laws very close to 2.4! The FCC statement is widely interpreted to suggest that *encoding* should approximate a power of $^1/_{2.2}$; the reciprocal of $^1/_{2.2}$, 0.45, appears in modern standards such as BT.709. However, as I mentioned on page 321, BT.709's effective overall curve is very close to a square root. The FCC specification should not be taken seriously: Use BT.709 for encoding.

Standards for 576*i* SD also have poorly specified transfer functions. An "assumed display power function" of 2.8 is mentioned in EBUspecifications; some people interpret this as suggesting an encoding exponent of $^1/_{2.8}$. However, the 2.8 value is unrealistically high. In fact, European displays are comparable to displays in other parts of the world, and encoding to BT.709 is appropriate.

Surprisingly, no current standards specify viewing conditions in the studio. Only in 2011 was a standard adopted that specifies the transfer function of an idealized studio display! In the absence of a studio display EOCF, consumer display manufacturers adopted their own (nonstandard) practices, one factor leading to unpredictable image display in consumers' premises.

Bit depth requirements

In Figure 10.1 on page 108, in Chapter 10's discussion of constant luminance, I indicated that conveying relative luminance directly would require about 11 bits. That observation stems from two facts. First, studio video experience proves that 8 bits is barely sufficient to convey gamma-corrected *R'G'B'* – that is, 2^8 (or 256) nonlinear levels are sufficient. Second, the transfer function used to derive gamma-corrected *R'G'B'* has a certain maximum slope; a maximum slope of 4.5 is specified in BT.709. The number of codes necessary in a linear-light representation is the product of these two factors: 256 times 4.5 is 1152, which requires 11 bits.

In BT.601 coding with 8 bits, the black-to-white range without footroom or headroom encompasses 220 levels. For linear-light coding of this range, 10 bits suffices:

$$4.5 \cdot 220 = 990; \;\; 990 < 2^{10}$$

In studio video, 8 bits per component barely suffice for distribution purposes. Some margin for roundoff error is required if the signals are subject to processing operations. For this reason, 10-bit studio video is now usual. To maintain 10-bit BT.709 accuracy in a linear-light system would require 12 bits per component. The BT.709 transfer function is suitable for video intended for display in the home, where contrast ratio is limited by the ambient environment. For higher-quality video, such as home theater, or for the adaptation of HD to digital cinema, we would like a higher maximum gain. When scaled to a lightness range of unity, CIE L^* has a maximum gain of 9.033; sRGB has a gain limit of 12.92. For these systems, linear-light representation requires 4 bits in excess of 10 on the nonlinear scale – that is, 14 bits per component.

$$4.5 \cdot 880 = 3960; \;\; 3960 < 2^{12}$$

If *RGB* or *XYZ* tristimulus components were conveyed directly, then 16 bits in each component would suffice for any realistic image-reproduction purpose. Linear-light 16-bit coding is now practical in high-end production, for example, scene-linear workflows using OpenEXR coding. For now, such approaches don't havve realtime hardware. In most applications, the nonlinear characteristics of perception are exploited and nonlinear image data coding is used.

Gamma in modern display devices

PDP and DLP devices are commonly described as employing PWM. However, it is not exactly the widths of the pulses that are being modulated, but the number of unit pulses per frame.

Modern display devices, such as liquid crystal displays (LCDs), have transfer functions different from that of CRTs. Plasma display panels (PDPs) and Digital Light Processors (DLPs) both achieve apparent continuous tone through *pulse width modulation* (PWM): They are intrinsically linear-light devices, with straight-line transfer functions. Linear-light devices, such as PDPs and DLPs, potentially suffer from the "code 100" problem explained on page 31: In linear-light, more than 8 bits per component are necessary to achieve high quality.

No matter what transfer function characterizes the display, it is economically important to encode image data in a manner that is well matched to perceptual requirements. The BT.1886 EOCF is well matched to CRTs, but more importantly, it is well matched to perception! The performance advantage of perceptual coding, the wide deployment of equipment that expects BT.1886 decoding, and the huge amount of program material already encoded to this standard preclude any attempt to establish new standards optimized to particular devices.

A display device whose transfer function differs from a CRT must incorporate local correction, to adapt from its intrinsic transfer function to the transfer function that has been standardized for image interchange.

Estimating gamma

Concerning the conversion between BT.601 levels and the full-swing levels commonly used in computing, see Figure 31.3, on page 384.

Knowing that a CRT is intrinsically nonlinear, and that its response is based on a power function, many researchers have attempted to summarize the nonlinearity of a CRT display in a single numerical parameter γ using this relationship, where V is code (or voltage) and T is luminance (or tristimulus value):

$$T = V^{D} \qquad \text{Eq 27.6}$$

The model forces zero voltage to map to zero luminance for *any* value of gamma. Owing to the model being "pegged" at zero, it cannot accommodate black-level errors: Black-level errors that displace the transfer function upward can be "fit" only by an estimate of gamma that is much smaller than 2.4. Black-level errors that displace the curve downward – saturating at zero over some portion of low voltages – can be "fit" only with an estimate of gamma that is much larger than 2.4. The only way the single *gamma* parameter can fit a black-level variation is to alter the curvature of the function. The apparent wide variability of gamma under this model has given gamma a bad reputation.

A much better model is obtained by fixing the exponent of the power function at 2.4, and using the single parameter to accommodate black-level error, ϵ:

$$T = (V + \epsilon)^{2.4} \qquad \text{Eq 27.7}$$

This model fits the observed nonlinearity much better than the variable-gamma model.

A simple technique to estimate gamma uses luminance measurements for video signal codes 0.08 and 0.8. Take the $\log_{10}$ of these two luminance values. The arithmetic difference between the two logs is a decent gamma estimate. The two video signal values are one decade apart; the 0.08 video signal is high enough to avoid potential black-level issues, and the 0.8 signal is low enough to avoid CRT saturation.

Equation 27.8 below is written with logs to base 10; however, because the ratio oof logs is taken, any base would do. In this calculation, luminance values L_1 through L_N must be strictly greater than L_0. If the video signal values are not at equal intervals, replace $^{i}/_{n}$ in the denominator by V_i where each V_i is the appropriate video signal level strictly between 0 and 1.

Figure 27.1, on page 317, graphs several pure 2.4-power functions. Gamma is 2.4 everywhere along these curves. Consider measuring a display at $n+1$ video signal values at equal intervals of $^{1}/_{n}$ between 0 and 1. (Usually ten values are used, 0.1, 0.2, …, 0.9, 1.0.) Using L_0 to symbolize the luminance produced by zero signal value and L_N to symbolize the luminance produced by unity signal value, average gamma can be estimated as follows:

$$\gamma = \frac{1}{9} \sum_{i=1}^{9} \frac{\log_{10} \dfrac{L\left(\frac{i}{10}\right)}{L(1)}}{\log_{10} \frac{i}{10}} \qquad \text{Eq 27.8}$$

EBU Tech. 3325 (2008), *Methods for the Measurement of the performance of Studio Monitors*, Version 1.1 (Sep.).

A variant of this formulation is described in EBU Tech. 3325; it is commonly used in home theatre calibration. In my view, this formulation of average gamma gives the luminance produced at reference white video level undue influence over the estimated gamma value. If reference white luminance is depressed – as will be the case for a CRT entering saturation, or an LCD mimicking that behaviour – then all of the contributing point-gamma values will be low, and the average gamma estimate will be reduced. In some formulations (such as that of the EBU), L_i in the numerator is replaced by $L_i - L_0$, subtracting the zero-code luminance; that subtraction leads to errors.

Berns, Roy S., Ricardo J. Motta, and Mark E. Gorzynski (1993), "CRT colorimetry," in *Color Research and Application* **18**: 299–325.

In my view, a better way to characterize gamma is to perform a numerical fit to an appropriate model, such as the GOGO model of Berns, Mottta, and Gorzynski.

Video displays have historically been aligned in the studio using the PLUGE test signal, adjusting BLACK LEVEL (BRIGHTNESS, or offset) so that 0% video and −2% video signals were just barely under the threshold of visibility. That procedure leaves video signal zero producing a small amount of light, typically between 0.01 and 0.1 nt for 100 nt reference white. For a pure 2.4-power function, the video signal corresponding to absolute, theoretical black is about −2% of the reference white signal, that is, around 10-bit interface code 32.

Cowan, William B. (1983), "An inexpensive scheme for calibration of a colour monitor in terms of CIE standard coordinates," in *Computer Graphics* **17** (3): 315–321 (July).

If you want to determine the nonlinearity of your display, consult the classic article by Cowan. In addition to describing how to measure the nonlinearity, he describes how to determine other characteristics of your display – such as the chromaticity of its white point and its primaries – that are important for accurate colour reproduction.

Gamma in video, CGI, and Macintosh

Transfer functions in video (and PC), computer-generated imagery, and Macintosh are sketched in the rows of Figure 27.5 opposite. Each row shows four function blocks; from left to right, these are a camera or scanner LUT, an image storage device, an output LUT, and a display.

In video, sketched in the top row, the camera applies a transfer function to accomplish gamma correction. Signals are then maintained in a perceptual domain throughout the system until conversion to tristimulus values at the display. I show the output LUT with a ramp that leaves data unaltered: Video systems conventionally use no LUT, but the comparison is clarified if I portray the four rows with the same blocks.

PC graphics hardware ordinarily implements lookup tables at the output of the framestore, as I detailed in *Raster images,* on page 67. However, most PC software

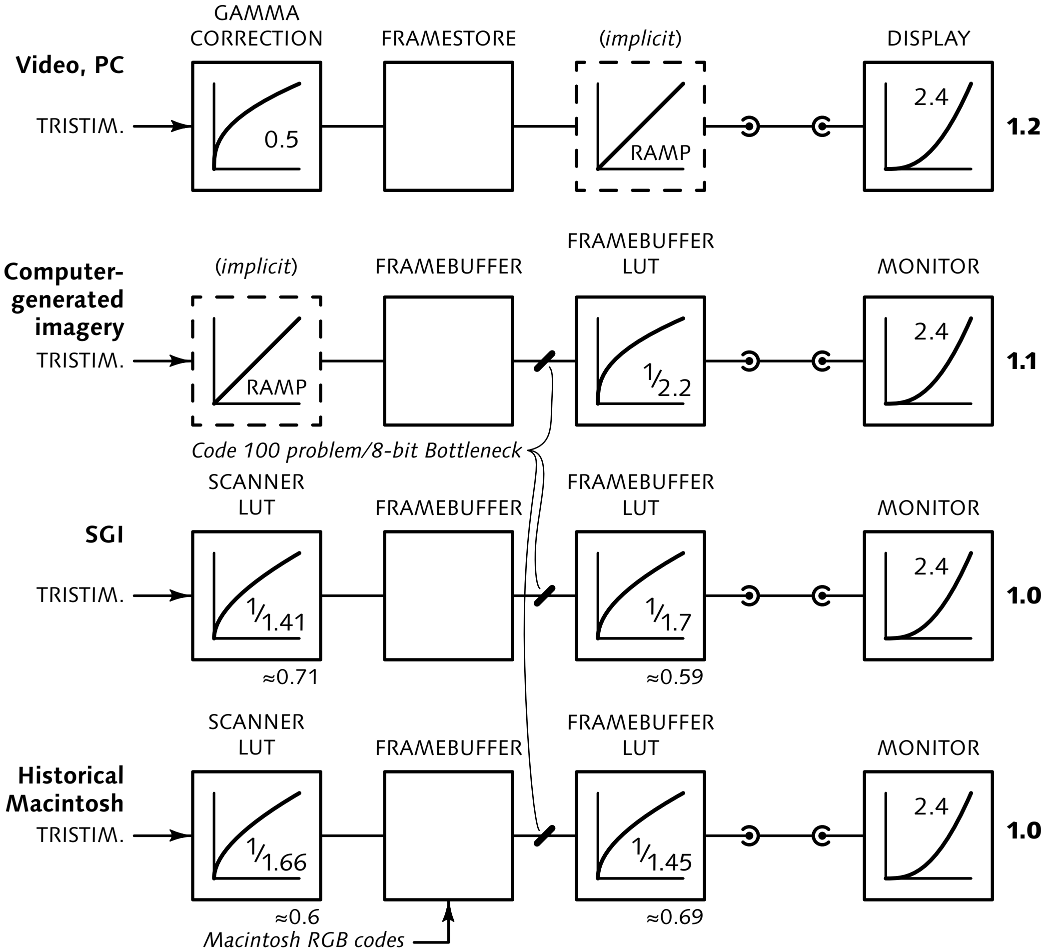


Figure 27.5 **Gamma in video, CGI, and Macintosh** are summarized in the rows of this diagram. Tristimulus signals enter from the left; the columns show the transfer functions of (respectively) a camera or scanner; the image storage device (framestore or framebuffer); output LUT; and the display.

In video, sketched in the top row, a transfer function that mimics vision is applied at the camera ("gamma correction"); the signal remains in perceptual space until the encoding is reversed by the display. (PCs have comparable signal encoding.) In computer graphics, sketched in the second row, calculations are performed in the linear-light domain, and gamma correction is applied in a LUT at the output of the framebuffer. Macintosh computers, sketched in the bottom row, take a hybrid approach: The scanner applies a $1/_{1.66}$ power, and a $1/_{1.45}$-power function is loaded into the LUT. Using $\gamma_E \approx 1/_{1.66}$ is appropriate for prerendered imagery, to produce an end-to-end exponent of 1.0.

The end-to-end power function exponent, or *picture rendering* (see page 115), is shown for each row by the number at the extreme right. This number is the product of the exponents across the system. Some people call this "system gamma," but that term is so widely misused that I reject it.

The Macintosh computer historically implemented a $^1/_{1.45}$-power function at the output LUT. John Knoll's *Gamma* Control Panel was commonly used to load the output LUT. When set to a gamma value *g*, the Control Panel loaded the LUT with a power function whose exponent is $^{2.61}/_g$. Strangely, gamma on Macintosh computers came to be quoted as the exponent applied prior to the framebuffer (whereas in other computers it is the exponent of the table loaded into the output LUT). So, the Mac's default gamma was said to be 1.8, not 1.45. A more reasonable value of display gamma of 2.2 results in tristimulus value proportional to code value raised to the 1.66-power (see Figure 27.6).

A Macintosh could be set to handle video (or PC) *R'G'B'* data by loading a ramp into its output LUT. Using Knoll's control panel, this is accomplished by setting gamma to 2.61.

JPEG/JFIF files originated on Macintosh historically represented *R*, *G*, and *B* display tristimulus values raised to the 0.6 power (that is, about $^1/_{1.65}$).

As of Mac OS X 10.6 ("Snow Leopard"), Macintosh software has been brought into conformance with the colour properties of sRGB.

accommodates display hardware without lookup tables. When the LUT is absent, code values map directly to voltage, and the situation is equivalent to video. So, the top row in the diagram pertains to PCs.

Computer graphics systems generally store tristimulus values in the framebuffer, and use hardware LUTs, in the path to the display, to gamma-correct on the fly. This is illustrated in the second row. Typically, a $^1/_{2.2}$-power function is loaded into the output LUT; in this case, picture rendering of 1.1 is achieved.

Macintosh computers, prior to Mac OS X 10.6, used the approach shown in the bottom row. The output LUT is, by default, loaded with a $^1/_{1.45}$-power function. The combination of the default LUT and the usual 2.4-power display function results in a 1.66-power function that relates Macintosh *R'G'B'* values (such as the values stored in a PICT file or data structure) to displayed tristimulus values.

If a desktop scanner is to produce Macintosh *R'G'B'* values that display relative luminance correctly, then a 1.66-power function must be loaded to the scanner LUT. In the typical Macintosh situation, the $^1/_{1.66}$, $^1/_{1.45}$, and 2.4 exponents combine to achieve an end-to-end exponent of unity. This is suitable for scanning photographs or offset printed matter, where picture rendering is already incorporated into the image.

For Macintosh *R'G'B'* values originated by application software, part of Macintosh gamma correction must be effected by application software prior to presentation of *R'G'B'* values to the Macintosh graphics subsystem; the remainder is accomplished in the output LUTs. When scanning, part of Macintosh gamma correction is effected by the LUT in the scanner driver, and the remainder is accomplished in the output LUTs.

Halftoned printing has a builtin nonlinearity, owing to the phenomenon of dot gain. Reflectance from the printed page is approximately proportional to the 1.8-power of [1–*CMYK*] code values. Macintosh *R'G'B'* values are not perceptually optimum; however, apparently by serendipity, Macintosh *R'G'B'* coding is nearly perfectly matched to the dot gain of halftone printing. This led to the dominance of Macintosh computers in graphic arts and prepress, and made "gamma 1.8" image encoding a de facto standard for graphic arts.

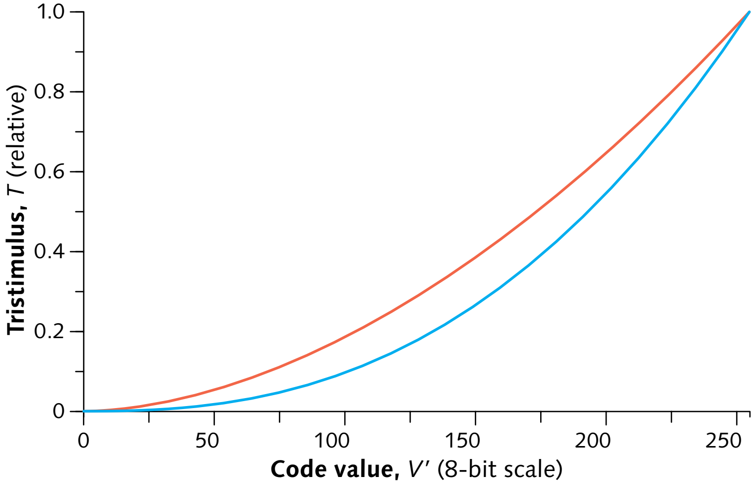


Figure 27.6 **Gamma PC and in classic Mac** are different, owing to the interpretation of *R′G′B′* code values by the display system. On a PC, the output LUT is either absent or programmed as if absent, and code values are subject to the 2.4-power function of the display (sketched in the lower curve). On a Mac prior to Mac OS X version 10.6 "Snow Leopard," the default output LUT imposes a $1/_{1.45}$-power function on the code values, then the display imposes its usual 2.4-power function; the concatenation of these two functions results in a 1.66-power function that relates Mac code value to displayed relative luminance, as sketched in the upper curve.

At the right-hand end of each row of Figure 27.5, on page 329, I have indicated in boldface type the rendering intent usually used. In video, I have shown an end-to-end power function of 1.2. For computer-generated imagery, I have shown the typical value of 1.1. For Macintosh, I have sketched the usual situation where prerendered images are being scanned; in this case, the end-to-end power function exponent is unity.

Correct display of computer image data depends upon knowing the transfer function that is expected at the output of the graphics subsystem. If an image that originates on a PC traverses the classic $1/_{1.45}$-power function of a pre-10.6 Macintosh LUT and a 2.4-power function display, midtones will display too light: Code 128 will produce luminance 1.5 times higher than intended. Conversely, if an image originates on a classic pre-10.6 Macintosh (where the $1/_{1.45}$-power function is expected), but is displayed on a PC (without this function), midtones will display much too dark. The relationship between default *R′G′B′* code values and displayed luminance factors for both PC and Mac is graphed in Figure 27.6.

Gamma shift refers to an undesired alteration of effective decoding gamma that results from inadvertent application of Macintosh-related gamma correction upon import or export of video involving a Macintosh computer. Gamma shift usually involves inadvertent application of a 1.45-power function or its inverse, a 0.69-power function.

Gamma in computer graphics

Computer-generated imagery (CGI) software systems generally perform calculations for lighting, shading, depth-cueing, and antialiasing using approximations to tristimulus values, so as to model the physical mixing of light. Values stored in the framebuffer are processed by hardware lookup tables on the fly on their way to the display. If linear-light values are stored in the framebuffer, the LUTs can accomplish gamma-correction. The power function at the CRT acts on the gamma-corrected signal voltages to display the correct luminance values at the face of the screen. Software systems usually provide a default gamma value and some method to change the default.

The BT.709 function is suitable for originating image data at high light levels (2000 lx or more) intended for viewing at about 100 nt in a dim surround. For other origination or viewing environments, see the comments on page 118.

The framebuffer's LUTs enable software to perform tricks to manipulate the appearance of the image data without changing the image data itself. To allow the user to make use of features such as accurate colour reproduction, applications should access lookup tables in the structured ways that are provided by the graphics system, and not by direct manipulation of the LUTs.

Gamma in pseudocolour

In *Pseudocolour,* on page 70, I described how the colour lookup table (CLUT) in a pseudocolour system contains values that are directly mapped to voltage at the display. It is conventional for a pseudocolour application program to provide, to a graphics system, *R'G'B'* colour values that are already gamma corrected for a typical monitor and typical viewing conditions. A pseudocolour image stored in a file is accompanied by a *colourmap* whose *R'G'B'* values are intended to be subject to an EOCF approximating a 2.4-power function at display.

Limitations of 8-bit linear coding

As mentioned in *Gamma in computer graphics,* on page 332, computer graphics systems that render synthetic imagery usually perform computations in the linear-light – or loosely, "intensity" – domain. Low-end graphics accelerators historically performed Gouraud shading in the linear-light domain, and stored 8-bit components in the framebuffer. In *The "code 100" problem and nonlinear image coding,* on page 31, I explained that linear-light representation cannot achieve high-quality images with just 8 bits per component; such images typically exhibit contouring. The visibility of contouring is enhanced by a perceptual effect called *Mach bands*; consequently, the contouring artifact is sometimes called *banding*.

High-end systems for computer-generated imagery (CGI) typically operate in the linear-light ("gamma = 1.0") domain using more than 8 bits per component (often floating point). Some systems perform gamma correction in software, then write gamma-corrected values into a limited-depth framebuffer. Other systems have "deep colour" framebuffers (having components with more than 8 bits, and often 16 bits); a unity ramp is loaded into the LUT of the framebuffer. This arrangement maximizes perceptual performance, and produces rendered imagery without the quantization artifacts of 8-bit linear-light coding.

Linear and nonlinear coding in CGI

Computer graphic standards often make no explicit mention of transfer function. Often, linear-light coding is implicit. However, in the JPEG standard there is no mention of transfer function but *nonlinear* (video-like) coding is implicit: Unacceptable results are obtained when JPEG is applied to linear-light data. All of these standards deal with *RGB* quantities; you might consider their *RGB* values to be comparable, but they're not!

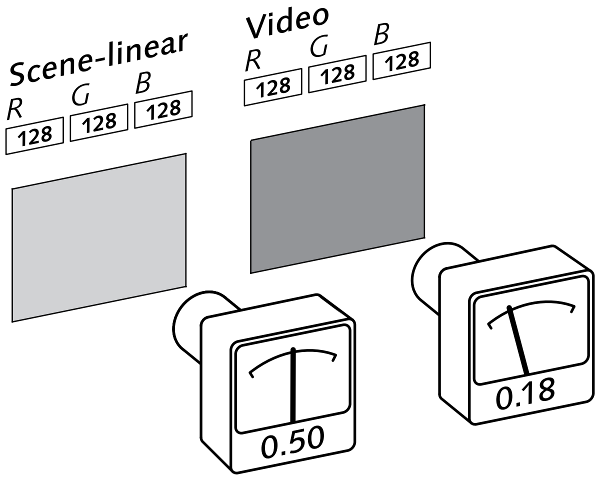


Figure 27.7 **Linear and nonlinear coding in imaging standards.** In linear-light standards, code [128, 128, 128] produces luminance halfway up the *physical* scale, a relative luminance of 0.5. In video, code [128, 128, 128] produces luminance halfway up the *perceptual* scale, only about 0.18 in relative luminance. Values are denoted *RGB* in both cases; however, the values are not comparable. The discrepancy exemplifies a serious problem in the exchange of image files.

What are loosely called *JPEG files* use the *JPEG File Interchange Format* (JFIF), cited in the margin of page 502. Version 1.02 of the JFIF specification states that linear-light coding (gamma 1.0) is used. That is seldom the case in practice; instead, image data is encoded expecting a 2.2-power EOCF. See page 328.

Figure 27.7 sketches two systems displaying the same *RGB* triple, [128, 128, 128]. A photometer reading the luminance displayed by a scene-linear system is shown at the left; a photometer reading luminance displayed by a video system is shown at the right. In scene-linear, the displayed luminance is halfway up the *physical* scale, a relative luminance of 0.5. In the video case, displayed luminance is halfway up the *perceptual* scale, only about 0.18 in relative luminance. Many graphics image files do not carry any transfer function information. If you exchange *RGB* image data without regard for transfer functions, huge differences will result when image data is displayed.

The digital image-processing literature rarely discriminates between linear and nonlinear coding. Also, when *intensity* is mentioned, be suspicious: Image data may be represented in linear-light form, *proportional* to intensity. However, a pixel component value is usually associated with a small area of a sensor or a display, so its units should include a per square meter ($\cdot m^{-2}$) term, so radiance, luminance, relative luminance, or tristimulus value are technically correct. All of these quantities are *proportional* to intensity, but they do not have units of intensity and they are not properly described as intensity values.

Luma and colour differences 28

This chapter describes colour coding systems that are used to convey image data derived from additive *RGB* primaries. I outline nonlinear *R'G'B'*, explain the formation of *luma*, denoted *Y'*, as a weighted sum of these nonlinear signals, and introduce the *colour difference* (chroma) components [*B'–Y'*, *R'–Y'*], [C_B, C_R], and [P_B, P_R].

The design of a video coding system is necessarily rooted in detailed knowledge of human colour perception. However, once this knowledge is embodied in a coding system, what remains is physics, mathematics, and signal processing. This chapter concerns only the latter domains.

Colour acuity

A monochrome video system ideally senses relative luminance, described on page 256. Luminance is then transformed by the gamma correction circuitry of the camera, as described in *Gamma in video,* on page 318, into a signal that takes into account the properties of lightness perception. At the receiver, the display – historically, the CRT itself – imposes the required inverse transfer function.

A colour image is sensed in three components, red, green, and blue, as described in *Additive reproduction (RGB),* on page 288. To minimize the visibility of noise or quantization, the *RGB* components should be coded nonlinearly.

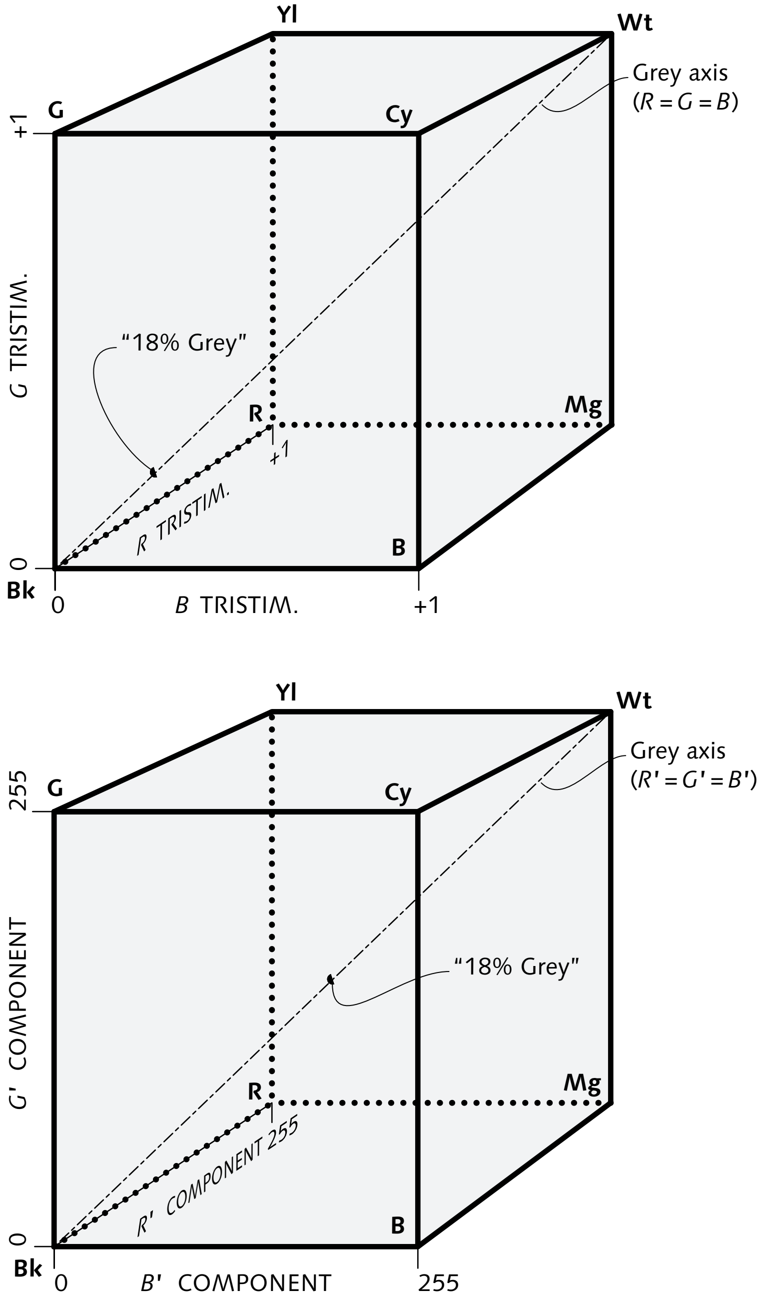


Figure 28.1 ***RGB* and *R'G'B'* cubes.** *RGB* components form the coordinates of a three-dimensional colour space; coordinate values between 0 and 1 define the unit cube. Linear coding, sketched at the top, has poor perceptual performance when 8 or even 10 bits are used for each component. In video, *RGB* components are subject to *gamma correction* to impose perceptual uniformity.

RGB and *R'G'B'* colour cubes

Red, green, and blue tristimulus (linear light) primary components, as detailed in *Colour science for video,* on page 287, can be considered to be the coordinates of a three-dimensional colour space. Coordinate values between zero and unity define the unit cube of this space, as sketched at the top of Figure 28.1 opposite. Linear-light coding is used in CGI, where physical light is simulated. However, as I explained in the previous chapter, *Gamma in video*, 8-bit linear-light coding exhibits poor perceptual performance: 12 or 14 bits per component are necessary to achieve excellent quality. The best perceptual use is made of a limited number of bits by using nonlinear coding that mimics the nonlinear lightness response of human vision. As introduced on page 27, and detailed in Chapter 27 *Gamma,* on page 315, in video, JPEG, MPEG, computing, digital still photography, and in many other domains a nonlinear transfer function is applied to *RGB* tristimulus signals to give nonlinearly coded (*gamma-corrected*) components, denoted with prime symbols: *R'G'B'*. Excellent image quality is obtained with 10-bit nonlinear coding with a transfer function similar to that of BT.709 or sRGB.

In PC graphics, 8-bit nonlinear coding is common: Each of *R', G',* and *B'* ranges from 0 through 255, inclusive, following the quantizer transfer function sketched in Figure 4.1, on page 37. The resulting *R'G'B'* cube is sketched at the bottom of Figure 28.1 opposite. A total of 2^{24} colours – that is, 16,777,216 colours – are representable. Not all of them can be distinguished visually; not all are perceptually useful; but they are all colours. Studio video uses headroom and footroom, as explained in *Studio-swing (footroom and headroom),* on page 42: 8-bit *R'G'B'* has 219 codes between black and white, for a total of 220^3 or 10,648,000 codewords.

In video, *codeword* (or *codepoint*) refers to a combination of three integer values such as [*R'*, *G'*, *B'*] or [*Y'*, C_B, C_R].

The drawback of conveying *R'G'B'* components of an image is that each component requires relatively high spatial resolution: Transmission or storage of a colour image using *R'G'B'* components requires a capacity three times that of a greyscale image. Human vision has considerably less spatial acuity for colour information than for lightness. Owing to the poor colour acuity of vision, a colour image can be coded into a wideband monochrome component representing lightness, and two narrowband components carrying colour information, each having substantially less spatial resolution than lightness. In analog video, each colour channel has bandwidth typically one-third that of the monochrome channel. In digital video, each colour channel has half the data rate (or data capacity) of the monochrome channel, or less. There is strong evidence that the human visual system forms an achromatic channel and two chromatic colour-difference channels at the retina.

Here the term *colour difference* refers to a signal formed as the difference of two gamma-corrected colour components. In other contexts, the term can refer to a numerical measure of the perceptual distance between two colours.

Green dominates luminance: Between 60% and 70% of luminance comprises green information. Signal-to-noise ratio is maximized if the colour signals on the other two components are chosen to be blue and red. The simplest way to "remove" lightness from blue and red is to subtract it, to form a pair of *colour difference* (or loosely, *chroma*) components.

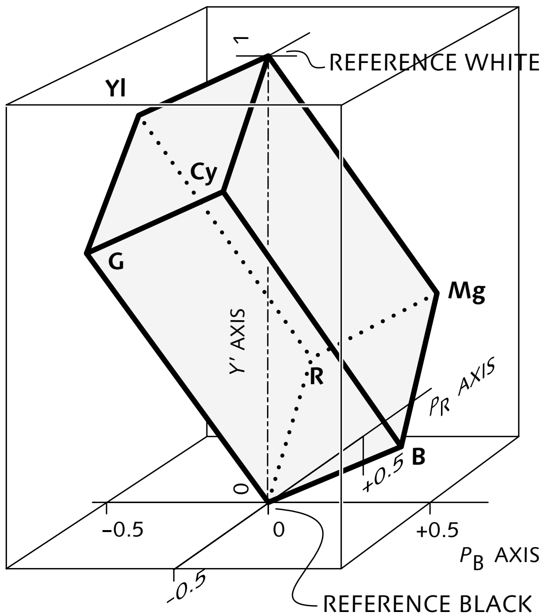
The monochrome component in colour video could have been based upon the luminance of colour science (a weighted sum of R, G, and B). Instead, as I explained in *Constant luminance,* on page 107, luma is formed as a weighted sum of R', G', and B', using coefficients similar or identical to those that would be used to compute luminance. Expressed in abstract terms, luma ranges 0 to 1. Colour difference components $B'-Y'$ and $R'-Y'$ are bipolar; each ranges nearly ±1.

In component analog video, $B'-Y'$ and $R'-Y'$ are scaled to form P_B and P_R components. In abstract terms, these range ±0.5. Figure 28.2 shows the unit $R'G'B'$ cube transformed into luma [Y', P_B, P_R]. (Various interface standards are in use; see page 359.) In component digital video, $B'-Y'$ and $R'-Y'$ are scaled to form C_B and C_R components. In 8-bit $Y'C_BC_R$ prior to the application of the interface offset, the luma axis of Figure 28.2 would be scaled by 219, and the chroma axes by 112.

I introduced interface offsets on page 44.

Once colour difference signals have been formed, they can be subsampled to reduce bandwidth or data capacity, without the observer's noticing, as I will explain in *Chroma subsampling, revisited,* on page 347.

Figure 28.2 A **$Y'P_BP_R$ cube** is formed when *R'*, *G'*, and *B'* are subject to a particular 3×3 matrix transform. The *valid R'G'B'* unit cube occupies about one-quarter of the volume of the $Y'P_BP_R$ unit cube. (The volume of the $Y'P_BP_R$ unit cube, the outer boundary of this sketch, is the same as the volume of the *R'G'B'* cube in Figure 28.1 on page 336; however, the useful codes occupy only the central parallelpiped here.) Luma and colour difference coding incurs a penalty in signal-to-noise ratio, but this disadvantage is compensated by the opportunity to subsample.



Izraelevitz, David, and Joshua L. Koslov (1982), "Code utilization for component-coded digital video," in *Tomorrow's Television* (Proc. 16th Annual SMPTE Television Conference): 22–30.

It is evident from Figure 28.2 that when *R'G'B'* signals are transformed into the $Y'P_BP_R$ space of analog video, the unit *R'G'B'* cube occupies only part of the volume of the unit $Y'P_BP_R$ cube: Only ¼ of the $Y'P_BP_R$ volume corresponds to *R'G'B'* values all between 0 and 1. Consequently, $Y'P_BP_R$ exhibits a loss of signal-to-noise ratio compared to *R'G'B'*. However, this disadvantage is offset by the opportunity to subsample.

In a *legal* signal, no component exceeds its reference excursion. Signal combinations that are *R'G'B'*-legal are termed *valid.* Signals within the $Y'P_BP_R$ unit cube are $Y'P_BP_R$-legal. However, about ¾ of these combinations correspond to *R'G'B'* combinations outside the *R'G'B'* unit cube: Although legal, these $Y'P_BP_R$ combinations are *invalid* – that is, they are *R'G'B'*-illegal.

In digital video, we refer to *codewords* instead of combinations. There are about 2.75 million valid codewords in 8-bit $Y'C_BC_R$, compared to 10.6 million in 8-bit studio *R'G'B'*. If *R'G'B'* is transcoded to 8-bit $Y'C_BC_R$, then transcoded back to *R'G'B'*, the resulting *R'G'B'* cannot have any more than 2.75 million colours.

$$\frac{\frac{1}{4}\cdot 220\cdot 225^2}{220^3} = \frac{2784375}{10648000} \approx 0.261$$

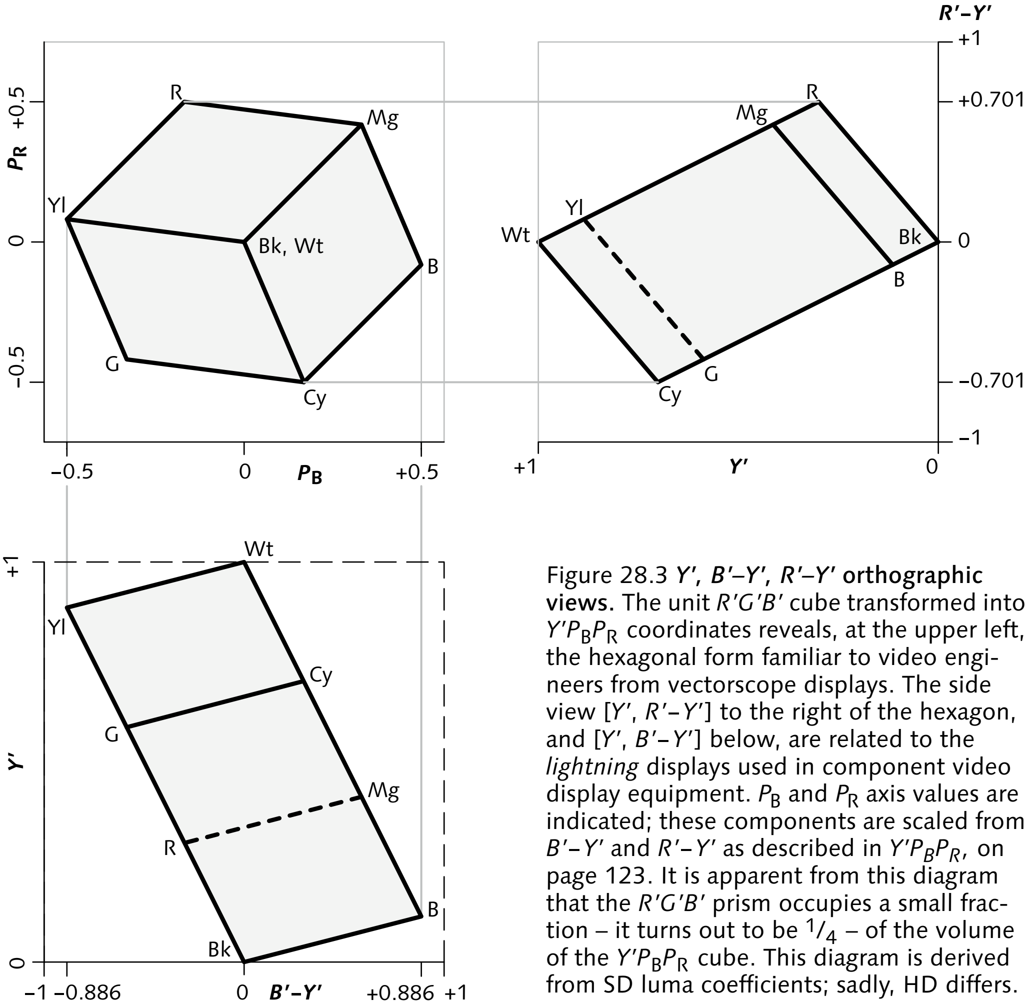


Figure 28.3 **Y', $B'{-}Y'$, $R'{-}Y'$ orthographic views.** The unit $R'G'B'$ cube transformed into $Y'P_BP_R$ coordinates reveals, at the upper left, the hexagonal form familiar to video engineers from vectorscope displays. The side view [Y', $R'{-}Y'$] to the right of the hexagon, and [Y', $B'{-}Y'$] below, are related to the *lightning* displays used in component video display equipment. P_B and P_R axis values are indicated; these components are scaled from $B'{-}Y'$ and $R'{-}Y'$ as described in *$Y'P_BP_R$*, on page 123. It is apparent from this diagram that the $R'G'B'$ prism occupies a small fraction – it turns out to be $^1/_4$ – of the volume of the $Y'P_BP_R$ cube. This diagram is derived from SD luma coefficients; sadly, HD differs.

C_BC_R components are comparable to P_BP_R components, but have codeword values ranging ±112 on the 8-bit scale instead of abstract values ranging ±0.5.

In Figure 28.2, the $Y'P_BP_R$ cube is portrayed off-axis. Figure 28.3 shows three orthographic views of the $R'G'B'$ prism in $Y'P_BP_R$-space. The luma axis, denoted Y', ranges 0 to 1. The chroma axes are annotated with both [$B'{-}Y'$, $R'{-}Y'$] scaling (where the components range ±0.886 and ±0.701, respectively), and P_BP_R scaling (where the components both range ±0.5). The extent of the volume of $Y'P_BP_R$ space that lies outside the $R'G'B'$ prism is apparent. The emergent xvYCC system, to be described, uses $Y'C_BC_R$ codewords outside the unit $R'G'B'$ prism – that is, formerly "invalid" codewords – to convey wide-gamut colour.

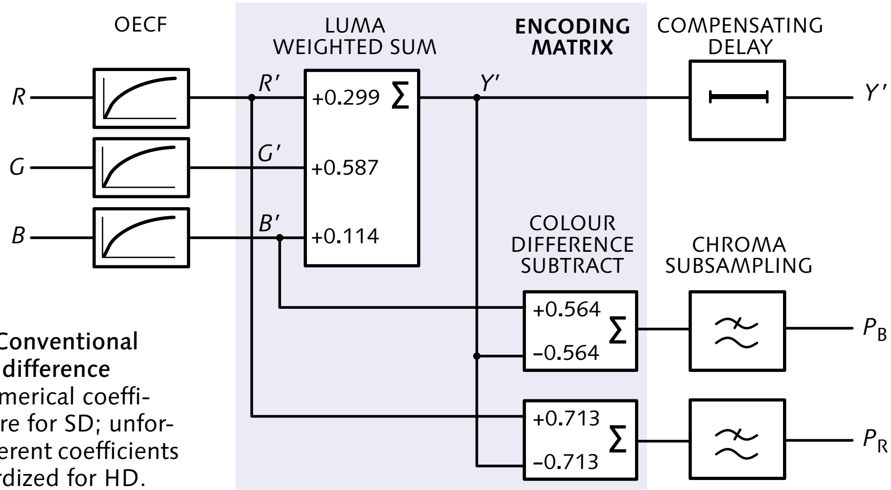


Figure 28.4 **Conventional luma/colour difference encoder.** Numerical coefficients here are for SD; unfortunately, different coefficients were standardized for HD.

Conventional luma/colour difference coding

I explained constant luminance on page 107. True constant luminance coding remains an intriguing possibility, but at present all video systems use nonconstant luminance coding, which I will now describe.

Figure 28.4 shows a time delay element in the luma path. Luma is delayed by a time interval equal to the transit delay of chroma through the chroma bandlimiting filters.

A conventional luma/colour difference encoder is shown in Figure 28.4 above. First, a nonlinear transfer function is applied to each of the red, green, and blue linear (tristimulus) components. Then luma is formed as a weighted sum of gamma-corrected *R′*, *G′*, and *B′* components. *B′*–*Y′* and *R′*–*Y′* colour difference components are formed by subtraction; in Figure 28.4, scaling to analog P_B and P_R components is indicated. Finally, the colour difference components are lowpass filtered.

Eq 28.1 BT.601 $Y'P_BP_R$ encoding matrix (for SD)

$$\mathbf{P} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix}$$

For the derivation of this matrix, and a more precise expression, see *PBPR components for SD,* on page 359.

The highlight rectangle in Figure 28.4 groups together the weighted adder that forms luma with the pair of colour difference subtractors; the combination is equivalent to matrix multiplication by the 3×3 matrix ***P*** shown in Equation 28.1 in the margin. The numerical values used in Equation 28.1, in Figure 28.4, and in subsequent figures in this chapter all reflect the BT.601 luma coefficients used in SD. Unfortunately, the coefficients for HD are different, as I will describe in *Component video colour coding for HD,* on page 369.

Figure 28.5 illustrates a conventional luma/colour difference decoder. In a digital decoder, the colour difference (chroma) components are horizontally (and,

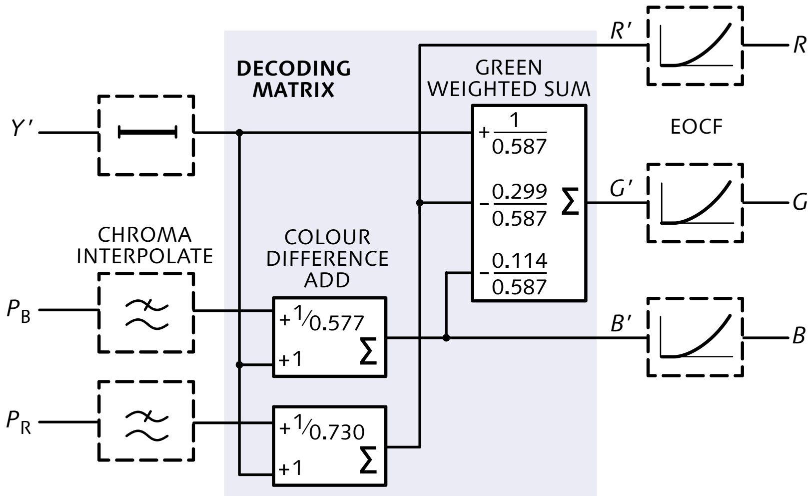


Figure 28.5 **Conventional luma/colour difference decoder,** Parameter values here are those of SD. In a historical analog SD television receiver, chroma interpolation is implicit, and requires no circuitry: I draw these components with dashed outlines. In a historical analog SD television receiver, the EOCF is inherent in the CRT, and similarly requires no components.

in some applications, spatially) interpolated; in an analog decoder, no circuitry is required to perform this function. Luma is added to the colour difference components to reconstruct nonlinear blue and red components. A weighted sum of luma, blue, and red is then formed to reconstruct the nonlinear green component.

Eq 28.2 BT.601 $Y'P_BP_R$ decoding matrix (for SD)

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & -0.714 \\ 1 & 1.772 & 0 \end{bmatrix}$$

The blue and red colour difference adders and the weighted adder that recovers green, all enclosed by the highlight rectangle in Figure 28.5, can be considered together as multiplication by the 3×3 matrix $\mathbf{P}^{-1}$ shown in Equation 28.2. These values are for SD; the matrix for HD is different.

To produce linear-light tristimulus components, all three components are subject to the inverse transfer function sketched at the right with dashed outlines. Usually, a decoder is used with a CRT that has an intrinsic 2.4-power function, or with some other display that incorporates a 2.4-power function; in either case, the transfer function need not be explicitly computed.

Luminance and luma notation

In *Luminance from red, green, and blue,* on page 258, I described how relative (linear-light) luminance, proportional to intensity, can be computed as an appropriately weighted sum of *RGB*.

In video, the luminance of colour science isn't computed. Instead, we compute a nonlinear quantity *luma* as a weighted sum of nonlinear (gamma-corrected) *R'G'B'*. The weights – or *luma coefficients* – are related to the luminance coefficients. The luma coefficients specified in BT.601 have been ubiquitous for SD, but new and different weights have been introduced in HD standards. In my opinion, the luma coefficients need not and should not have been changed for HD: Complexity is added to upconversion and downconversion in studio and consumer equipment, for no improvement in performance or quality.

Television standards documents historically used the prime symbol (′) – often combined with the letter *E* for voltage – to denote a component that incorporates gamma correction. For example, E'_R historically denoted the gamma-corrected red channel. Gamma correction is nowadays so taken for granted in video that the *E* and the prime symbol are usually elided. This has led to much confusion among people attempting to utilize video technology in other domains.

See Appendix A, *YUV and luminance considered harmful,* on page 567.

The existence of several standard sets of primary chromaticities, the introduction of new coefficients, and continuing confusion between luminance and luma all beg for a notation to distinguish among the many possible combinations. In the absence of any standard notation, I was compelled to invent my own. Figure 28.6 below sketches the notation that I use. The base symbol is *Y, R, G, or B.* The subscript denotes the standard specifying the chromaticities of the primaries

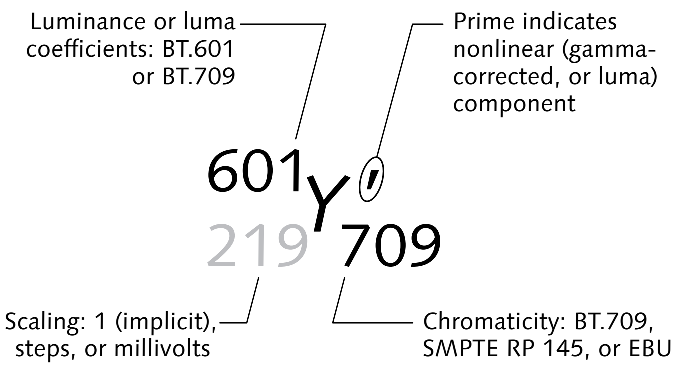
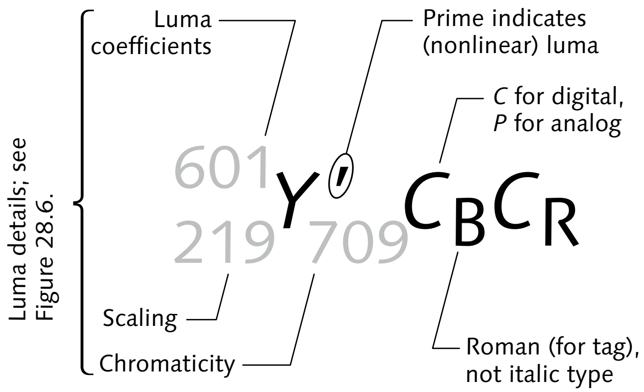


Figure 28.6 **Luminance and luma notation** is necessary because different primary chromaticity sets, different luma coefficients, and different component scale factors are in use. Unity scaling suffices for components in this chapter; in succeeding chapters, other scale factors will be introduced.

Figure 28.7 **Typesetting $Y'C_BC_R$** is a challenge! Luma coefficient set, scaling, and chromaticities are set out as in Figure 28.6. The prime should always be present, to distinguish *luma* from the *luminance* of colour science. *C* is appropriate for digital signals, *P* for analog. Subscripts B and R serve as tags, not variables: They should be in roman type, not italics. B comes before R.



and white. An unprimed letter indicates a linear-light tristimulus component (*R*, *G*, or *B*), or relative luminance (*Y*). A prime symbol (′) indicates a nonlinear (gamma-corrected) component (*R*′, *G*′, and *B*′), or luma (*Y*′).

For luminance or luma, a leading superscript indicates the standard specifying the weights used. Historically the weights of BT.601 were implicit, but recent HD standards such as BT.709 and SMPTE ST 274 call for different weights. Finally, the leading subscript indicates the overall scaling of the signal. If omitted, an overall scaling of unity is implicit, otherwise an integer such as 219, 255, or 874 specifies the black-to-white excursion in a digital system, or a number such as 661, 700, or 714 specifies the analog excursion in millivolts.

Typesetting $Y'C_BC_R$ (or $Y'P_BP_R$) is a challenge! I illustrate the main points in Figure 28.7 above. I augment *Y*′ with a leading superscript and subscript and a trailing subscript, according to the conventions of Figure 28.6. Without these elements, the intended colour cannot be determined with certainty. I place a single prime on the *Y*. Some authors prime the C_B and C_R as well, but I consider that practice to be obsessive and pedantic. Practical, deployed image coding systems are either perceptually coded in all three colour components or (rarely) fully linear-light in all three. Since there are no “hybrid” systems (linear-light luminance with nonlinear colour differences, or perceptually coded luma with linear-light colour differences), there is no need to triplicate the prime.

Nonlinear red, green, blue (*R'G'B'*)

Video originates with approximations of linear-light (*tristimulus*) *RGB* primary components, usually represented in abstract terms in the range 0 (black) to +1 (white). In order to meaningfully determine a colour from an *RGB* triple, the colorimetric properties of the primaries and the reference white – such as their CIE [*x*, *y*] chromaticity coordinates – must be known. Colorimetric properties of *RGB* components were discussed in *Colour science for video,* on page 287. In the absence of any specific information, use the BT.709 primaries and the CIE D_{65} white point.

In *Gamma,* on page 315, I described how lightness information is coded nonlinearly, in order to achieve good perceptual performance from a limited number of bits. In a colour system, the nonlinear transfer function described in that chapter is applied individually to each of the three *RGB* tristimulus components: From the set of *RGB* tristimulus (linear-light) values, three gamma-corrected primary signals are computed; each is approximately proportional to the square-root of the corresponding scene tristimulus value.

See page 320.

BT.709 standardizes an EOCF that ostensibly should be imposed at the camera. For tristimulus values greater than a few percent, the encoding is this:

$$\begin{aligned} R'_{709} &= 1.099R^{0.45} - 0.099 \\ G'_{709} &= 1.099G^{0.45} - 0.099 \\ B'_{709} &= 1.099B^{0.45} - 0.099 \end{aligned} \qquad \text{Eq 28.3}$$

However, what is important in "BT.709" coding is that encoded image data produces the intended picture appearance on the standard display device. BT.1886 standardizes the reference EOCF for HD. Encoding should be specified in terms of the inverse EOCF.

$$\begin{aligned} R'_{709} &= R^{1/2.4} \\ G'_{709} &= G^{1/2.4} \\ B'_{709} &= B^{1/2.4} \end{aligned} \qquad \text{Eq 28.4}$$

To encode sRGB is similar, but using a power of $^1/_{2.2}$ instead of $^1/_{2.4}$.

BT.601 luma

The following luma equation is standardized in BT.601 for SD, and also applies to JPEG/JFIF (in computing) and Exif (in digital still photography):

$$^{601}Y' = 0.299\,R' + 0.587\,G' + 0.114\,B' \qquad \text{Eq 28.5}$$

As mentioned a moment ago, the *E* and prime symbols originally used for video signals have been elided over the course of time, and this has led to ambiguity of the *Y* symbol between colour science and television.

The coefficients in the luma equation are based upon the sensitivity of human vision to each of the *RGB* primaries standardized for the coding. The low value of the blue coefficient is a consequence of saturated blue colours having low lightness. The luma coefficients are also a function of the white point, or more properly, the *chromaticity of reference white*.

The BT.601 luma coefficients were computed using the technique that I explained in *Luminance coefficients*, on page 306, using the historical NTSC primaries and white point of 1953; see Table 3.2, on page 25 of *Composite NTSC and PAL: Legacy Video Systems*.

In principle, luma coefficients should be derived from the primary and white chromaticities. The BT.601 luma coefficients of Equation 28.5 are the same as those established in 1953 by the NTSC from the primaries and white point then in use. Primaries actually used in consumer displays changed a few years after the adoption of NTSC. The primaries in use for 480*i* SD today are approximately those specified in SMPTE RP 145; the primaries in use for 576*i* SD are approximately those specified in EBU Tech. 3213. (These primary sets are slightly different; both sets very nearly match the primaries of BT.709.) Despite the change in primaries, the luma coefficients for SD video – both 480*i* and 576*i* – have remained unchanged from the values that were established in 1953. As a consequence of the change in primaries, the luma coefficients in SD no longer theoretically match the primaries. The mismatch has little practical significance.

The mismatch between the primaries and the luma coefficients of SD has little practical significance; however, the mismatch of luma coefficients between SD and HD has great practical significance!

BT.709 luma

International agreement on BT.709 was achieved in 1990 on the basis of “theoretically correct” luma coefficients derived from the BT.709 primaries:

$$^{709}Y' = 0.2126\,R' + 0.7152\,G' + 0.0722\,B' \qquad \text{Eq 28.6}$$

Chroma subsampling, revisited

The purpose of colour difference coding is to enable subsampling. In analog video, the colour difference components are subject to bandwidth reduction through the use of analog lowpass filters; horizontal colour detail is removed. In digital video, the chroma components are subsampled, or *decimated*, by filtering followed by the discarding of samples. Figure 12.3, *Chroma subsampling,* on page 124, sketches several digital subsampling schemes. In 4:2:2 subsampling, after filtering, alternate colour difference samples are discarded at the encoder. In 4:2:0, vertical chroma detail is removed as well. At the decoder, the missing samples are approximated by interpolation.

In analog chroma bandlimiting, and in digital subsampling, some colour detail is lost. However, owing to the poor colour acuity of vision, the loss cannot be detected by a viewer *at normal viewing distance*.

Some low-end digital video systems simply drop chroma pixels at the encoder without filtering, and replicate chroma pixels at the decoder. Discarding samples can be viewed as point sampling; that operation runs the risk of introducing aliases. Proper decimation and interpolation filters should be used; these should be designed according to the principles explained in *Filtering and sampling,* on page 191.

Luma/colour difference summary

When luma and colour difference coding is used for image interchange, it is important for the characteristics of red, green, and blue to be maintained from the input of the encoder to the output of the decoder. The chromaticities of the primaries were detailed in *Colour science for video,* on page 287, and mentioned in this chapter as they pertain to the encoding and decoding of luma. I have assumed that the characteristics of the primaries match across the whole system. The primaries upon which luma and colour difference coding are based are known as the *interchange* (or *transmission*) *primaries*.

In practice, a camera sensor may produce *RGB* components whose chromaticities do not match the interchange primaries. To achieve accurate colour reproduction in such a camera, it is necessary to insert

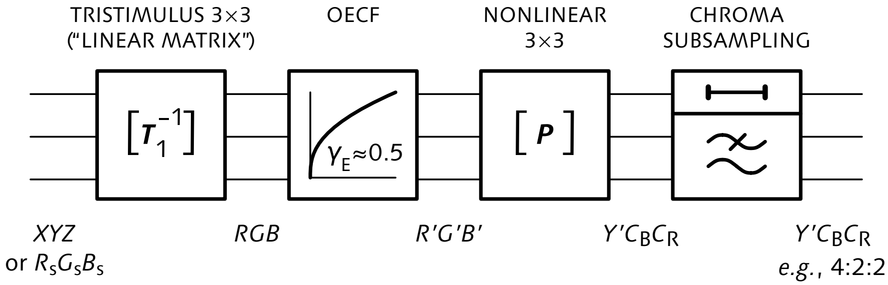


Figure 28.8 **A luma/colour difference encoder** involves the four stages summarized in this block diagram. First, linear-light (tristimulus) input signals are transformed through the "linear" matrix T_1^{-1} to produce *RGB* coded to the interchange primaries. Gamma correction is then applied. The matrix P then produces luma and two colour differences. The colour difference (chroma) signals are then subsampled; luma undergoes a compensating delay.

a 3×3 matrix that transforms tristimulus signals from the image capture primaries to the interchange primaries. (This is the "linear matrix" built into the camera.) Similarly, a decoder may be required to drive a display whose primaries are different from the interchange primaries; at the output of the decoder, it may be necessary to insert a 3×3 matrix that transforms from the interchange primaries to the image display primaries. (See page 309.)

I use T_1^{-1} to denote the encoding "linear matrix," to conform to the notation of *Luminance coefficients,* on page 306.

Figure 28.8 above summarizes luma/colour difference encoding. If image data originated in linear *XYZ* components, a 3×3 matrix transform (T_1^{-1}) would be applied to obtain linear *RGB* having chromaticities and white reference of the interchange primaries. For BT.709 interchange primaries standard for SD and HD, the matrix would be that of Equation 26.9, on page 308. More typically, image data originates in some device-dependent space that I denote $R_1G_1B_1$, and the 3×3 "linear matrix" transform (T_1^{-1}) is determined by the camera designer. See the sequence of Figures 26.3 through 26.8, starting on page 300, and the accompanying text and captions, to gain an appreciation for how such a matrix might be crafted. Practical cameras do not have spectral sensitivities that are linear combinations of the CIE colour matching functions, so they are not properly characterized by chromaticities. Nonetheless, once a linear matrix to a set of interchange primaries has been chosen, Equation 26.10 can be used to derive equivalent sensor primaries (the "taking primaries").

Interchange primaries are also called transmission primaries.

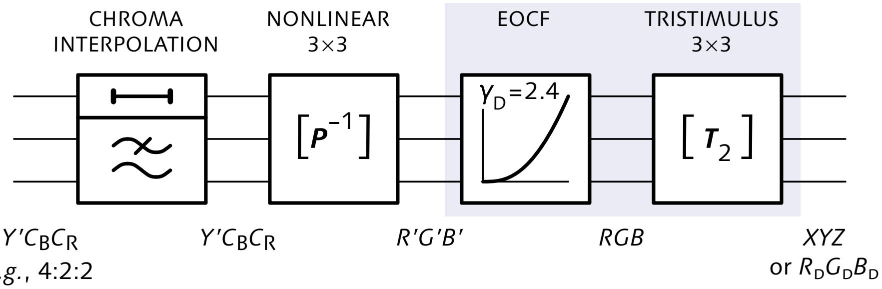


Figure 28.9 **A luma/colour difference decoder** involves the inverse of the four stages of Figure 28.8 in opposite order. First, subsampled colour difference (chroma) signals are interpolated; luma undergoes a compensating delay. The matrix $\boldsymbol{P}^{-1}$ then recovers *R′G′B′* from luma and two colour differences. A transfer function having an exponent of about 2.4 is then applied, which produces linear-light (tristimulus) signals *RGB*. If the display's primaries differ from the interchange primaries, *RGB* are transformed through the matrix $\boldsymbol{T}_2$ to produce appropriate $R_D G_D B_D$.

Figures 28.8 and 28.9 show 3×3 matrix transforms being used for two distinctly different tasks. When someone hands you a 3×3, you have to ascertain whether it is linearly or nonlinearly related to light power.

Once the linear matrix has been applied, each of the components is subject to a nonlinear transfer function (gamma correction) that produces nonlinear *R′G′B′*. These components are transformed through a 3×3 matrix ($\boldsymbol{P}$), to obtain luma and colour difference components $Y'C_BC_R$ or $Y'P_BP_R$. (This matrix depends upon the luma coefficients in use, and upon colour difference scale factors.) Then, if necessary, a chroma subsampling filter is applied to obtain subsampled colour difference components; luma is subject to a compensating delay.

A decoder uses the inverse operations of the encoder, in the opposite order, as sketched in Figure 28.9. In a digital decoder, the chroma interpolation filter reconstructs missing chroma samples; in an analog decoder, no explicit operation is needed. The 3×3 colour difference matrix ($\boldsymbol{P}^{-1}$) reconstructs nonlinear red, green, and blue primary components. The transfer functions restore the primary components to their linear-light tristimulus values. Finally, the tristimulus 3×3 matrix ($\boldsymbol{T}_2$) transforms from the primaries of the interchange standard to the primaries implemented in the display device.

When a decoder is intimately associated with a CRT display, the decoder's transfer function is performed by the nonlinear voltage-to-luminance relationship intrinsic to the CRT: No explicit operations are required for this step. However, to exploit this transfer function, the display primaries must be the same as – or at least very similar to – the interchange primaries.

The transfer functions of the decoder (or the CRT) are intertwined with gamma correction. As explained on page 115, an end-to-end power function having an exponent of about 1.2 is appropriate for typical television acquired at studio lighting levels and viewed at 100 nt in a dim surround. The encoder of Figure 28.8 imposes a 0.5-power function; the decoder of Figure 28.9 imposes a 2.4-power function. The product of these implements the end-to-end power function. If the native EOCF of a display device differs from that of a CRT, then decoding should include a transfer function that is the composition of a 2.4-power function and the inverse transfer function of the display device.

Owing to the dependence of the optimum end-to-end power function upon viewing conditions, there here ought to be a user control for rendering intent – perhaps even replacing BRIGHTNESS and CONTRAST – but there isn't!

When viewing a rather bright display (say 320 nt) in an average surround (say 20%), a 1.1 end-to-end power is appropriate; a 2.2-power EOCF (like that of sRGB) is appropriate. When viewing in a dark (0%) surround, a 1.3 end-to-end power is appropriate; a 2.6-power EOCF (like that of digital cinema) is appropriate.

If the display primaries match the interchange primaries, the decoder's 3×3 tristimulus matrix is not needed. If a display has primaries not too different from the interchange primaries, then it may be possible to compensate the primaries by applying a 3×3 matrix in the nonlinear domain. But if the primaries are quite different, it will be necessary to apply the transform between primaries in the tristimulus domain; see *Transforms among RGB systems,* on page 309.

SD and HD luma chaos

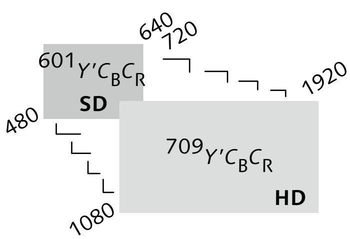


Figure 28.10 **Luma/colour difference flavors**

Although the concepts of $Y'P_BP_R$ and $Y'C_BC_R$ coding are identical in SD and HD, the BT.709 standard established a new set of luma coefficients for HD. That set differs dramatically from the luma coefficients for SD specified in BT.601. There are now two flavors of $Y'C_BC_R$ coding, as suggested by Figure 28.10 in the margin; I denote the flavors ${}^{601}Y'C_BC_R$ for SD, and ${}^{709}Y'C_BC_R$ for HD. Similarly, there are two flavors of $Y'P_BP_R$ for analog systems, ${}^{601}Y'P_BP_R$ for SD, and ${}^{709}Y'P_BP_R$ for HD.

In my view, it is extremely unfortunate that different coding was adopted: Image coding and decoding now depend on whether the picture is small (conventional

System	Luma coefficients	EOCF	Primary chromaticities
SD 480*i*	BT.601	*unspecified*	SMPTE RP 145
SD 576*i*	BT.601	*unspecified*	EBU Tech. 3213
HD 720*p*, 1080*i*, 1080*p*	BT.709	BT.1886	BT.709

Table 28.1 **luma coefficients, EOCF, and primary chromaticities** for video decoding and display circa 2011 are summarized. Encoding in all systems should be accomplished through forming *R'G'B'* values that yield the intended image appearance in the reference display and viewing conditions. For SD, there is no effective standard for either the reference EOCF or the viewing conditions. For HD, BT.1886 standardizes the reference EOCF, but not the viewing conditions.

video, SD) or large (HD); that dependence erodes the highly useful concept of resolution-independent production in the $Y'C_BC_R$ 4:2:2 and 4:2:0 domains. In my opinion, HD should have been standardized with the BT.601 luma coefficients. With things as they stand, the smorgasbord of colour-encoding parameters makes accurate image interchange extremely difficult. The situation is likely to get worse with time, not better.

Table 28.1 above summarizes the standards for primary chromaticities, transfer functions, and luma coefficients that are either implicit or explicit in several SD and HD standards. When video is converted among these standards, appropriate processing should be performed in order to preserve the intended colour.

It's sensible to use the term colourbar test *signal* (or *pattern*) instead of colourbar test *image*, because the signal is standardized, not the image.

The colourbar test signal is standardized in the *R'G'B'* domain, without any reference to primaries, transfer function, or luma coefficients. The colours of the bars depend upon which primary chromaticities are in use; the luma and colour difference levels of the bars depend upon which luma coefficients are in use. When colour conversions and standards conversions are properly performed, the colours and levels of the colourbar test signal will change!

Luma/colour difference component sets

These colour difference component sets, all based upon $B'-Y'$ and $R'-Y'$, are in use:

- $Y'P_BP_R$ coding is used in component analog video; P_B and P_R are scaled to have excursion nominally identical to that of luma. $Y'P_BP_R$ can be potentially based upon either BT.601 (for SD) or BT.709 (for HD). In 480*i*29.97 SD, three different analog interface standards are in use: EBU N10 "SMPTE," Sony, and Panasonic.
- $Y'C_BC_R$ coding is used for component digital video; C_B and C_R are scaled to have excursion $^{224}/_{219}$ that of luma. A "full-range" variant is used in JPEG/JFIF. $Y'C_BC_R$ can be potentially based upon BT.601 or BT.709 luma coefficients.

In Chapter 5, *NTSC and PAL Chroma modulation*, of *Composite NTSC and PAL: Legacy Video Systems*, I detail two additional component sets, now obsolete, whose proper use was limited to composite SD NTSC and PAL:

- *Y'UV* components are only applicable to composite NTSC and PAL systems. $B'-Y'$ and $R'-Y'$ are scaled so as to limit the excursion of the composite (luma plus modulated chroma) signal. *Y'UV* coding is always based upon BT.601 luma coefficients.
- *Y'IQ* components were historically used in composite NTSC systems from 1953 to about 1970. *UV* components were rotated 33°, and axis-exchanged, to enable wideband-*I* transmission. This obsolete technique has not been practiced since about 1970. *Y'IQ* coding was always based upon the luma coefficients now documented in BT.601.

The bewildering set of scale factors and luma coefficients in use is set out in Table 28.2A opposite for analog SD, Table 28.2B overleaf for digital SD and computing systems, and Table 28.2C for analog and digital HD. The following two chapters detail component colour coding for SD and HD, respectively.

	System	*Notation*	*Colour difference scaling*
1	Component analog video, 480*i* (EIA/CEA-770 and "SMPTE") and 576*i* EBU N10; also, 480*i* Panasonic M-II, zero setup (Japan)[a]	${}^{601}_{700}Y'_{145}P_BP_R$, ${}^{601}_{700}Y'_{EBU}P_BP_R$	The EBU N10 standard calls for 7:3 picture-to-sync ratio, 700 mV luma excursion with zero setup. P_B and P_R components are scaled individually to range ±350 mV, an excursion identical to luma.
2	Component analog video, 480*i* Sony, 7.5% setup[a]	${}^{601}_{661}Y'_{145}P_BP_R$	Sony de facto standards call for 10:4 picture-to-sync ratio, 7.5% setup, and black-to-white luma excursion of approximately 661 mV. P_B and P_R components are scaled individually to range $4/3$ times ±350 mV, that is, ±466$2/3$ mV.
3	Component analog video, 480*i* Sony, zero setup (Japan)[a]	${}^{601}_{714}Y'_{145}P_BP_R$	Sony de facto standards call for 10:4 picture-to-sync ratio, zero setup, and black-to-white luma excursion of approximately 714 mV. P_B and P_R components are scaled individually to range $4/3$ times ±350 mV, that is, ±466$2/3$ mV.
4	Component analog video, 480*i* Panasonic, 7.5% setup[a]	${}^{601}_{647}Y'_{145}P_BP_R$	Panasonic de facto standards call for 7:3 picture-to-sync ratio, 7.5% setup, and black-to-white luma excursion of approximately 647.5 mV. P_B and P_R components are scaled individually to range $37/40$ times ±350 mV, that is, ±323.75 mV.
5	Composite analog NTSC, PAL video (incl. S-video)	various, typ. ${}^{601}_{700}Y'_{EBU}UV$, ${}^{601}_{714}Y'_{145}IQ$, ${}^{601}_{714}Y'_{145}UV$	*U* and *V* are scaled to meet a joint constraint: Scaling is such that peak composite video – luma plus modulated chroma – is limited to $4/3$ of the blanking-to-white excursion. Rotation and exchange of axes (e.g., *I* and *Q*) cannot be distinguished after analog encoding. There is no standard component interface.

Table 28.2A **Colour difference systems for analog SD.** The EBU N10 levels indicated in the shaded (first) row are sensible but unpopular. Designers of 480*i* SD studio equipment were forced to implement configuration settings for three interface "standards": EBU N10 ("SMPTE"), Sony, and Panasonic.

a The component analog interface for consumer equipment (such as DVD players) is properly scaled $Y'P_BP_R$, according to EIA/CEA-770.2 (cited on page 454). Some consumer equipment was engineered and deployed with incorrect $Y'P_BP_R$ scaling. Certain consumer devices have rear-panel connectors labelled *Y*, *B*–*Y*, *R*–*Y*, or *YUV*; these designations are plainly wrong.

	System	Notation	Colour difference scaling
6	Component digital video: 4:2:0, 4:1:1, BT.601 4:2:2 (incl. M-JPEG, MPEG, DVD, DVC)	${}^{601}_{219}Y'_{145}C_BC_R$	BT.601 calls for luma range 0…219, offset +16 at the interface. C_B and C_R are scaled individually to range ±112, an excursion $^{224}/_{219}$ of luma, offset +128 at the interface. Codes 0 and 255 are prohibited.
7	Component digital stillframe JPEG (incl. JFIF 1.02), typical desktop publishing and the web. Transfer functions vary; see the marginal note on page 335.	${}^{601}_{255}Y'_{709}C_BC_R$	There is no comprehensive standard. Luma reference range is typically 0 through 255. C_B and C_R are typically scaled individually to a "full-swing" of ±128, an excursion $^{256}/_{255}$ that of luma. C_B and C_R codes +128 are clipped; fully saturated blue and fully saturated red cannot be represented.

Table 28.2B **Colour difference systems for digital SD and computing.** The scaling indicated in the first row is recommended. For details of obsolete SD systems, see the table *Colour difference systems for analog composite SD and digital 4fSC SD*, in Chapter 14 of *Composite NTSC and PAL: Legacy Video Systems*. (Row numbering here is discontinuous so as to mesh with that table.)

	System	Notation	Colour difference scaling
11	Component analog HD	${}^{709}_{700}Y'_{709}P_BP_R$	7:3 picture-to-sync ratio, 700 mV luma excursion with zero setup. P_B and P_R components are scaled individually to range ±350 mV, an excursion identical to luma.
12	Component digital HD (BT.709/BT.1886)	${}^{709}_{219}Y'_{709}C_BC_R$	BT.709 calls for luma range 0…219, offset +16 at the interface. C_B and C_R are scaled individually to range ±112, an excursion $^{224}/_{219}$ of luma, offset +128 at the interface. Codes 0 and 255 are prohibited.
13	Component digital HD (xvYCC)	${}^{\text{xvYCC}}_{219}Y'_{709}C_BC_R$	xvYCC $Y'C_BC_R$ is identical to BT.709 $Y'C_BC_R$, except that some codewords outside the $R'G'B'$ unit cube represent wide-gamut colours.

Table 28.2C **Colour difference systems for HD.** The luma coefficient set for HD differs significantly from that of SD.