

© Copyright 1998

Eric J. Stollnitz

Reproducing Color Images with Custom Inks

by Eric J. Stollnitz

A doctoral dissertation
Department of Applied Mathematics
University of Washington
1998

Abstract

Reproducing Color Images with Custom Inks

by Eric J. Stollnitz

University of Washington

Advisor

Associate Professor David H. Salesin

Computer Science and Engineering

Chairperson of Supervisory Committee

Professor Loyce M. Adams

Applied Mathematics

This dissertation investigates the general problem of reproducing color images on an off-set printing press using custom inks in any combination and number. Many mathematical and algorithmic challenges arise when printing with inks other than the standard process colors (cyan, magenta, yellow, and black), particularly as the number of inks increases. These challenges include the development of gamut mapping strategies, efficient ink selection strategies, and robust methods for computing color separations in situations that may be either overconstrained or underconstrained. In addition, the demands of high-quality color printing require an accurate physical model of the colors that result from overprinting multiple inks using halftoning, including the effects of trapping, dot gain, and the interreflection of light between ink layers. As we explore these issues, we present new algorithms and physical models that together comprise a system capable of choosing optimal inks for an image and generating the appropriate color separations. Finally, we present some printed examples demonstrating the promise of our techniques.

⊕ Table of Contents ⊕

List of Figures	iii
List of Tables	v
Glossary	vii
Notation	xi
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Methodology	2
1.3 Contributions	3
1.4 Overview	3
Chapter 2: Background and related work	5
2.1 Color and color spaces	5
2.2 Black-and-white halftone printing	7
2.3 Color halftone printing	9
2.4 Printing with inks other than process color inks	11
2.5 Gamut mapping	13
2.6 Computing separations	14
Chapter 3: Modeling printing gamuts	17
3.1 The Neugebauer model of color halftoning	17
3.2 Adding trapping to the Neugebauer model	19
3.3 Adding dot gain to the Neugebauer model	21
3.4 Modeling the printing primaries	22

Chapter 4: Gamut mapping	27
4.1 Gamut mapping for one ink	28
4.2 Gamut mapping for two inks	31
4.3 Gamut mapping for three or more inks	36
Chapter 5: Selecting inks	43
5.1 Ink-selection objective function	43
5.2 Ink-selection optimization algorithm	44
Chapter 6: Computing separations	51
6.1 Separations for one ink	52
6.2 Separations for two inks	52
6.3 Separations for three or more inks	54
Chapter 7: Implementation and experiments	61
7.1 Implementation features	61
7.2 Experimental determination of model parameters	65
7.3 Printed examples	74
7.4 Timings and complexity	80
Chapter 8: Conclusion	83
8.1 Summary	83
8.2 Extensions	85
8.3 Future directions	88
Bibliography	91
Appendix A: Data for papers and inks	97
Printed examples	pocket

✦ List of Figures ✦

2-1	Relationships between color spaces	7
2-2	Black-and-white halftone printing	8
2-3	Color halftone printing	9
3-1	Three-ink gamut modeled by the Neugebauer equations	18
3-2	Four-ink gamut modeled by the Neugebauer equations	19
3-3	Imperfect trapping of overlapping inks	20
3-4	Modeling a layer of ink as an ideal filter	22
3-5	Modeling a layer of ink as a reflecting filter	23
3-6	Kubelka-Munk model of a layer of ink	24
4-1	Monotonically increasing mappings	28
4-2	Effects of luminance mappings on an image	29
4-3	Steps of gamut mapping for one ink	32
4-4	Axes associated with a typical duotone gamut	33
4-5	Steps of gamut mapping for two inks	35
4-6	Coordinate systems of three gamut mapping methods	37
4-7	Steps of gamut mapping for three or more inks	41
5-1	Comparison of ink selection optimization algorithms	48
6-1	Image pyramid used in multiresolution color separation	56
6-2	Steps of the multiresolution separation algorithm	57
6-3	Comparison of separation optimization algorithms	59
7-1	The interactive system's image viewer	62
7-2	The interactive system's color palette and ink selector	64
7-3	The interactive system's gamut visualization tool	64
7-4	Typical dot gain curves	68

7-5	Experimentally determined trapping fractions	71
7-6	Experimentally determined Fresnel reflection coefficients	72
7-7	Error computed for each model variation	73
A-1	Paper reflectance spectra	98
A-2	Ink model parameters	102

✦ **List of Tables** ✦

7-1	Variations used in fitting ink models to data	70
7-2	Chromaticities used in color conversion	76
7-3	Statistics for process-color examples	80
7-4	Statistics for custom-ink examples	81
A-1	Paper reflectance spectra	100
A-2	Dot-gain parameters	101
A-3	Fresnel reflection coefficients	101
A-4	Trapping fractions	101
A-5	Ink model parameters	105

✦ Glossary ✦

CMYK: the process color inks: cyan, magenta, yellow, and black.

COATED PAPER: paper treated with a thin clay coating to improve its smoothness. The coating gives the paper a somewhat shiny finish, and keeps ink from being absorbed and spreading out in the paper fibers (contrast with **UNCOATED PAPER**).

CONTINUOUS-TONE: a reproduction of an image in which ink thickness or pigment density varies continuously to portray the gradation of tone in the image (contrast with **HALFTONE**).

CONTRAST: difference of tone between the darker and the lighter parts of an image.

CONVENTIONAL SCREENING: a technique of producing a halftone consisting of a regular grid of dots of varying sizes (contrast with **STOCHASTIC SCREENING**).

DOT GAIN: a printing artifact and optical phenomenon in which halftone dots appear larger than desired, causing changes in colors or tones.

DUOTONE: reproduction of an image (traditionally a **GRAYSCALE IMAGE**) using two inks.

FRESNEL REFLECTION: reflection that occurs at a planar interface between two materials with differing indices of refraction.

GAMUT: the set of all colors displayable or reproducible by a given device.

GAMUT MAPPING: a function or algorithm capable of assigning to each input color a color lying within the gamut of a particular device.

GRAVURE: a printing process using engraved metal plates, in which the ink rests in tiny cells onto which paper is pressed to create the image (contrast with **LETTERPRESS** and **LITHOGRAPHY**).

GRAYSCALE IMAGE: an image defined by a single numerical value for each pixel indicating how light or dark that pixel should appear.

HALFTONE: reproduction of an image in which ink is either present or absent at each location; the gradation of tone in the image is reproduced by varying the size (in **CONVENTIONAL SCREENING**) or spacing (in **STOCHASTIC SCREENING**) of dots of ink.

HUE: the psychological property of a color differentiating red, orange, yellow, green, blue, and purple.

LAB COLOR SPACE: a color space designed to be perceptually more uniform than XYZ color space. In $L^*a^*b^*$ space, L^* represents lightness while a^* and b^* together encode saturation and hue.

LETTERPRESS: a printing process in which ink is transferred to paper from raised portions of a printing plate (contrast with **GRAVURE** and **LITHOGRAPHY**).

LITHOGRAPHY: a printing process utilizing a metal or plastic plate that is treated photochemically so that the image to be printed repels water and accepts oil-based ink, while the blank areas accept water and repel ink (contrast with **GRAVURE** and **LETTERPRESS**).

LUV COLOR SPACE: a color space designed to be perceptually more uniform than XYZ color space, and slightly less costly to work with than $L^*a^*b^*$ space. In $L^*u^*v^*$ space, L^* represents lightness while u^* and v^* together encode saturation and hue.

MOIRÉ: an undesirable artifact in color halftones printed using conventional screening. When grids of dots are printed at different angles for four or more inks, the pattern of dots can produce a distracting low-frequency interference pattern.

NEUGEBAUER MODEL: a mathematical model of the colors produced by printing multiple halftone images atop one another using different inks. The color of a small area is determined by the area-weighted average of the colors of each possible overprinted combination of the inks.

OFFSET PRINTING: a type of lithography in which the image is transferred from a cylindrical plate to a cylindrical rubber blanket, which in turn offsets the image onto a sheet of paper.

OVERPRINTING: printing with one ink over an area already printed with another ink.

N-TONE: reproduction of a color image using n custom inks, rather than the standard process colors.

PLATE: a rectangular sheet of metal, plastic, or paper bearing an engraved image (in **GRAVURE** printing) or ink-receptive image (in **LETTERPRESS** and **LITHOGRAPHY**).

PRINTING PRIMARIES: the discrete set of colors produced by halftone printing, including the paper color, the color of each ink printed individually, and the color of every possible overprinted combination of inks. For n inks, there are 2^n printing primaries.

PROCESS COLOR PRINTING: the use of cyan, magenta, yellow, and black inks to reproduce a color image.

PROOF: a representation of how a printed job is intended to look; the reference that the press operators strive to match.

REFLECTANCE SPECTRUM: the fraction of light reflected by a material, as a function of wavelength.

REGISTRATION: the precise alignment of two or more overprinted images, controlled by adjusting the positions of plates on the printing press.

RGB COLOR SPACE: the color space used by television and computer monitors. A color in *RGB* space is represented by three numbers corresponding to the signals applied to red, green, and blue phosphors in the display.

SEPARATION:

1. a single-channel image prescribing how much of a given ink will be used for each pixel of a color image.
2. the process of converting a color image to a set of single-channel images, one for each ink.

SPECTROPHOTOMETER: an instrument for measuring the reflectance of a surface. The sample surface is illuminated by a known light source, and the reflected light is separated into its component wavelengths by a prism or diffraction grating before being measured by a photodetector.

SPOT COLOR: a color printed with a custom ink, rather than with a combination of the standard process colors.

STOCHASTIC SCREENING OR FREQUENCY-MODULATED SCREENING: an alternative to conventional screening that produces halftones from very small, pseudorandomly placed dots. Tone is controlled by the spacing of dots rather than by their size.

TRANSMITTANCE SPECTRUM: the fraction of light passing completely through a material, as a function of wavelength.

TRAPPING:

1. the probability of an ink adhering to (being “trapped” by) the paper and inks upon which it is printed.
2. the modification of separations so that adjacent areas of solid color actually overlap, thereby avoiding gaps between colors when slight misregistration occurs.

UNCOATED PAPER: paper without a clay coating, typically having a rough surface that absorbs more ink than **COATED PAPERS**.

XYZ COLOR SPACE: a device-independent standard for specifying colors according to their luminance Y and two components X and Z encoding their hue and saturation. The XYZ coordinates of a color can be obtained by integrating a reflectance spectrum against three response functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$.

✦ Notation ✦

symbol	meaning
c	an arbitrary color
$c(\alpha)$	the predicted color for ink amounts α
\hat{c}	a gamut-mapped color for which separations are desired
$f(\alpha)$	objective function of separation algorithm
g_p	the color of paper (one of the printing primaries)
g_{pi}	the color of ink i on paper p
g_{pij}	the color of ink j atop ink i and paper p
h	hue angle in ellipsoidal gamut mapping coordinate system
$M(\lambda)$	a measured reflectance spectrum
n	the number of inks used in a reproduction
P	projection direction used in duotone gamut mapping
r	radial coordinate in ellipsoidal gamut mapping coordinate system
$R(\lambda)$	reflectance spectrum
$\tilde{R}(\lambda)$	back-surface reflectance spectrum
S	ink-spread direction used in duotone gamut mapping
t_{pi}	trapping fraction for ink i on paper p
t_{pij}	trapping fraction for ink j on ink i on paper p
$T(\lambda)$	transmittance spectrum
w	weight used in separation objective function
X	the X direction in XYZ color space
Y	the Y direction in XYZ color space
Z	the Z direction in XYZ color space

symbol	meaning
α_i	actual fractional area coverage of ink i
$\bar{\alpha}_i$	specified fractional area coverage of ink i
$\boldsymbol{\alpha}$	vector of all fractional area coverages: $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$
$\boldsymbol{\alpha}_{\text{ref}}$	vector of reference coverage values for separation algorithm
α_{limit}	maximum total amount of ink that will adhere to paper
β	exponent controlling the slopes of cubic mappings
γ_i	dot gain parameter for ink i
κ	parameter controlling n -tone gamut mapping coordinate system
λ	wavelength of visible light
ρ_{ai}	Fresnel reflection coefficient for light traveling from air to ink
ρ_{ia}	Fresnel reflection coefficient for light traveling from ink to air
ρ_{pi}	Fresnel reflection coefficient for light traveling from paper to ink
ρ_{ip}	Fresnel reflection coefficient for light traveling from ink to paper
ϕ	angular variable in ellipsoidal gamut mapping coordinate system

✦ Acknowledgments ✦

I am indebted to many people who helped make this dissertation possible. Thanks go first of all to my parents, Janet and Fred Stollnitz, who have always encouraged my academic endeavors. My advisor, David Salesin, deserves many thanks for his guidance and encouragement, not to mention his ability to remind me of the bigger picture surrounding my research. I am grateful to the other members of my committee, Loyce Adams, Randy LeVeque, and Terry Rockafellar, for their participation and valuable input. Many thanks also go to Victor Ostromoukhov of the Ecole Polytechnique Fédérale de Lausanne, for his helpful collaboration both during his stay in Seattle and after, and especially for his willingness to oversee the robotic measurement of more than 150,000 spectra.

Joanna Power and Brad West deserve special recognition as coauthors of our SIGGRAPH article on duotone printing. I enjoyed our collaboration and learned from it—particularly from the successes and frustrations they encountered in implementing the first effective system for reproducing color images with two inks. Joanna also gets the award for most acknowledged in this dissertation, for in addition to having worked on this project, she is a great friend and a wonderful housemate. My appreciation goes to my equally wonderful housemates, Deirdre and Eva, for helping distract me from an otherwise unending stream of work, never raising the subjects of color printing and computer graphics. Likewise, my sister Nancy and her family as well as my extended family of west coast relatives have often provided welcome respites from academic work.

Thanks go to Roger Hersch and Frédéric Pighin for helpful discussions. I appreciate the dual roles Pat Lewis has played, as a collaborator in our research discussions, and as a developer at Numinous Technologies, one of the supporters of this work. I am grateful to Eric Mosher, Skip Corbitt, and others at University of Washington's Publications Services, as well as to Larry Jablinske and others at Safeco Insurance's Graphics & Printing Services. Without their generous donations of time, resources, and expertise, this research would have come to a grinding halt without any printed results. Thanks go to Heath Printers for quickly and cheerfully handling our last-minute printing jobs.

Many others deserve recognition as well. I feel fortunate to have interacted with past and present members of the graphics research group, and particularly with Per Christensen, whose research brought me into the group. I also appreciate the infectious enthusiasm with which K. K. Tung and Ed Lazowska have played the roles of Applied Math and Computer Science department chairs; some of their enthusiasm has surely rubbed off on me. Finally, thanks to Karen Sakahara and Alicen Smith, who provided assistance above and beyond the call of duty at the approach of any number of deadlines.

To Mom and Dad.

✦ Introduction ✦

It is of interest . . . that, regardless of the number of impressions, the inks may be selected solely on the basis of their color gamut. Their colors need not be cyan, magenta, and yellow; nor is it required that they be transparent. The way is therefore opened for entirely new printing processes.

—Hardy and Wurzburg, 1948 [Hardy & Wurzburg 48]

1.1 Motivation

Fifty years ago, the promise of color printing with custom inks appeared imminent. The advantages of such a process are clearly numerous. Freed from the same fixed set of *process color* inks—cyan, magenta, yellow, and black—it should be possible to print more vibrant colors for art reproductions, annual reports, and packaging. Moreover, if the inks are chosen specifically for the particular image being reproduced, it should be possible in many cases to achieve these vivid colors with just a small number of inks—perhaps four—and perhaps at no greater cost than using the four process colors. In addition, it is common today to print boxes and wrappers with four process inks (for images) plus two *spot colors* for corporate logos or large areas of background. By selecting custom inks that complement the required spot colors, we might achieve better quality with six inks or comparable quality with fewer inks.

In recent years, several new color printing processes have been proposed that use a fixed set of six or more standard printing inks [Boll 94, Ostromoukhov 93, Takaghi et al. 94]. For those willing to use more inks, these new processes do provide more vivid color reproduction. However, Hardy and Wurzburg’s fifty-year-old vision of printing with arbitrary custom inks remains elusive. This dissertation explores the difficult problems that stand in the way of selecting and using custom inks.

1.2 Methodology

The world of color printing encompasses many technologies, including photographic processes, nonimpact printers (ink jet, thermal wax transfer, color laser, and dye sublimation), as well as printing presses (gravure, letterpress, and lithography). While virtually all of these technologies stand to benefit from more widespread use of custom inks, pigments, and toners, we limited our research to offset lithography, the most common type of commercial printing. We first investigated the possibility of reproducing color images using only two inks—called *duotone* printing [Power et al. 96]. Later, we expanded our work’s scope to printing with arbitrary numbers of inks, which we term *n-tone* printing [Stollnitz et al. 98].

There are a number of problems common to both duotone and *n-tone* printing. For one, it is very difficult to derive a physical model that accurately predicts how arbitrary inks will interact when printed together, in superposition and in juxtaposition using halftoning. In addition to optical effects, the model must take into account physical effects such as trapping and dot gain.

Furthermore, the set of colors, or gamut, reproducible by multiple custom inks has an irregular, nonconvex shape. Creating efficient and reliable gamut mapping algorithms for smoothly mapping image colors to the colors that can be achieved with a given set of inks is a nontrivial problem.

Choosing the best custom inks to use for a given image is another difficult problem—in this case, a combinatorial challenge, particularly as the number of inks used for printing gets large.

Finally, once the inks are chosen, we must compute separations for the given image. While for two inks there is always a simple analytic solution, for three or more inks this problem can become either overconstrained or underconstrained. The problem becomes overconstrained when the color to be printed cannot be achieved with quantities of ink between 0 and 100%. It is underconstrained when there are two or more ways of achieving the same color using different ink combinations. This situation arises wherever the gamut is doubly covered, a commonplace occurrence with four or more inks.

We address these challenges in detail with new physical models and algorithms, then demonstrate the potential of our approach with printed examples. Although a great deal more work remains to be done before Hardy and Wurzburg’s vision is achieved in its entirety, this research takes significant steps toward that goal.

1.3 Contributions

This dissertation lays out a general framework for multicolor printing with custom inks. The techniques we describe show promise for solving several longstanding problems in color printing:

- We modify existing models of ink layering and color halftoning to account for the effects of trapping and dot gain. We also present the steps needed to fit the model parameters to experimental measurements.
- We introduce a gamut mapping algorithm that is parameterized to effectively fill a gap between two predominant strategies in the literature.
- We describe how stochastic combinatorial optimization methods (simulated annealing and genetic algorithms) can be modified to find the papers and inks well-suited to reproducing a given image.
- Finally, we develop a robust multiresolution algorithm that, given any combination of inks, computes separations that are as smooth as the input image.

In a broader sense, the work presented here offers new opportunities to the printing industry. Our techniques for reproducing color images as duotones fill a niche between four-color process printing and current two-color printing practices (which typically result in a monochrome image with a slight color cast). Those who cannot afford to print their advertising, packaging, or newsletters using process color, or who do not require color fidelity, can still print colorful images with two carefully chosen inks.

By using more than two custom inks, n -tone printing offers the opportunity to reproduce images with the same or better color fidelity than process color printing, at the same or lower cost, while making efficient use of spot colors. This opportunity should be a boon to advertisers and packaging printers, who are already accustomed to printing with nonstandard inks.

1.4 Overview

The next chapter presents background material and previously published work relevant to color printing. Chapter 3 develops a mathematical model of the gamut of printable colors

for a given choice of paper and inks. Chapter 4 discusses algorithms for mapping the set of image colors into the printable gamut. Chapter 5 presents the combinatorial optimization problem of finding the best paper and inks for an image. Chapter 6 introduces an algorithm for computing separations for any image and any combination of inks. Chapter 7 describes our experimental results, including a variety of images printed with custom inks using our techniques. It also includes a discussion of the experimental procedure we used to determine the gamut model's parameters, and some features of our interactive application. We conclude in Chapter 8 with a summary of our work and ways in which it can be extended.

✦ Background and related work ✦

Before we dive into the steps required to print an image with custom inks, we present some background material and previously published research relevant to color, color spaces, and halftone printing.

2.1 Color and color spaces

Because this entire dissertation is essentially about color, we require a precise way of quantifying color. We give a rudimentary introduction here; the interested reader can refer to standard books on color science [Judd & Wyszecki 75, Judd 79, Wyszecki & Stiles 82]. Color is determined by the intensity of light in the range of visible wavelengths from about 400nm to 700nm. The color of a material like paper or ink can be quantified by the fraction of light it reflects at each visible wavelength; this function is called the material's *reflectance spectrum*. The experiments we describe in Chapter 7 rely on reflectance measurements of paper and ink samples we made using a device called a spectrophotometer; other devices and techniques for measuring reflectance are discussed by Kortüm [Kortüm 69].

Although the space of reflectance spectra is infinite in dimension, humans perceive color with only three types of receptors (the three kinds of cones in the retina). In addition to this physiological evidence, there is psychological evidence that human perception of color is three-dimensional: we can only describe three independent aspects of a color, typically its lightness, its saturation, and its hue; all other terms denote particular combinations of these aspects [Judd & Wyszecki 75, page 34]. Experiments confirm that every color can be matched by a linear combination of three primary wavelengths (roughly corresponding to red, green, and blue) [Hering 64]. Thus, we will consider colors to be three-dimensional quantities whenever possible.

We most often use the *XYZ* color space, developed in 1931 by the Commission Internationale de l'Éclairage (CIE) to standardize the specification of colors. A reflectance spectrum can be converted to *XYZ* coordinates by integrating the spectral information against three functions \bar{x} , \bar{y} , and \bar{z} [Judd & Wyszecki 75, pages 125–153 and 472–479]. The *Y* co-

ordinate of a color measures its *luminance* (a function of brightness), while together the X and Z coordinates encode hue and saturation information. The XYZ color space is additive, meaning that the color resulting from the superposition of two colored light sources can be calculated by simply adding the coefficients of the two known colors.

The computer graphics community is more familiar with the RGB color space, which refers to the red, green, and blue phosphors of a video display monitor. The RGB color space is device-dependent, since the relationship between XYZ coordinates and RGB coordinates depends on the colors and signal responses of the monitor phosphors. If we correct for any nonlinearity in the phosphor responses and we know the XYZ coordinates of the device's red, green, and blue primaries, we can convert between RGB and XYZ coordinates with a simple linear transform [Foley et al. 90, pages 585–587].

While the XYZ and RGB color spaces are convenient to work with because they are linear and additive, they do not provide a good measure of the differences between colors. In order to select optimal inks, we will rely heavily on the measurement of differences between the colors of an input image and the predicted colors of a printed image. We make use of a *perceptually uniform color space*, in which the difference between two colors as perceived by the human eye can be measured as the distance between points. For example, two colors c_1 and c_2 separated by some distance d in a perceptually uniform color space appear about as different as two other colors c_3 and c_4 separated by the same distance d . The CIE developed two color spaces intended to be more perceptually uniform than XYZ coordinates: $L^*a^*b^*$ and $L^*u^*v^*$. Both color spaces require the definition of a *reference white*, which is usually taken to be a standard light source defined by the CIE. In both spaces, L^* indicates brightness and has a value of 100 for reference white; these spaces are defined so that distances on the order of 1 or 2 units constitute just barely perceptible differences in color. Conversion between XYZ and either $L^*a^*b^*$ or $L^*u^*v^*$ coordinates is accomplished using transformations that are nonlinear but not terribly complicated [Glassner 95, pages 59–66].

Figure 2-1 illustrates the relationships between each of the color spaces discussed above. Note that although most color space transformations are invertible, there is no conversion from XYZ to reflectance, since a given XYZ point corresponds to many reflectance spectra.

There are many other three-dimensional color spaces; among these are a handful of recently developed spaces intended to be more uniform than either $L^*a^*b^*$ or $L^*u^*v^*$. Termed *color appearance spaces*, they attempt to account for the effect on color perception of viewing conditions such as the ambient lighting and the color that surrounds the sample or image of interest [Fairchild & Berns 93, Hunt 91, Luo et al. 96]. Viewing conditions are particu-

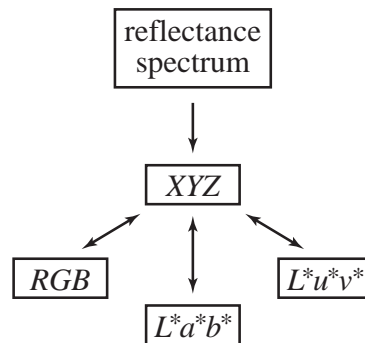


Figure 2-1 Relationships between color spaces.

larly important to the appearance of image colors because of the phenomenon known as chromatic adaptation: the human visual system generally perceives the brightest and least saturated visible color to be white regardless of its actual reflectance [von Kries 70]. Despite many theoretical and experimental advances in color appearance models, determining the effect of viewing conditions on a color image's appearance remains an open research problem. Despite its faults, we prefer to rely on the $L^*a^*b^*$ color space when computing color differences rather than explore the evolving field of color appearance spaces.

2.2 Black-and-white halftone printing

In offset printing, along with many other printing processes, an ink is either present or absent at each location on the page; there is no continuous transition from unprinted to fully printed. Reproducing a grayscale image with black ink on white paper therefore requires some finesse to fool the eye into seeing shades of gray when only black and white are present. The most familiar solution is to use a regular grid of ink dots in which the size of the dots increases as the darkness of the image increases, as shown in Figure 2-2. The image formed from the dots is known as a *halftone*, in contrast to the original *continuous-tone* image.

The process of making a halftone is called *screening*, referring to the original technique of photographing a continuous-tone image through a wire screen onto high-contrast film to produce a regular grid of varying sized dots. Computers have largely replaced photographic techniques of producing halftones. One of the alternatives to conventional screening made possible by computers is artistic screening, which uses variations of miniature pictures or letters instead of dots [Ostromoukhov & Hersch 95]. Another alternative is stochastic screening (also called frequency-modulated screening), in which the darkness of the im-

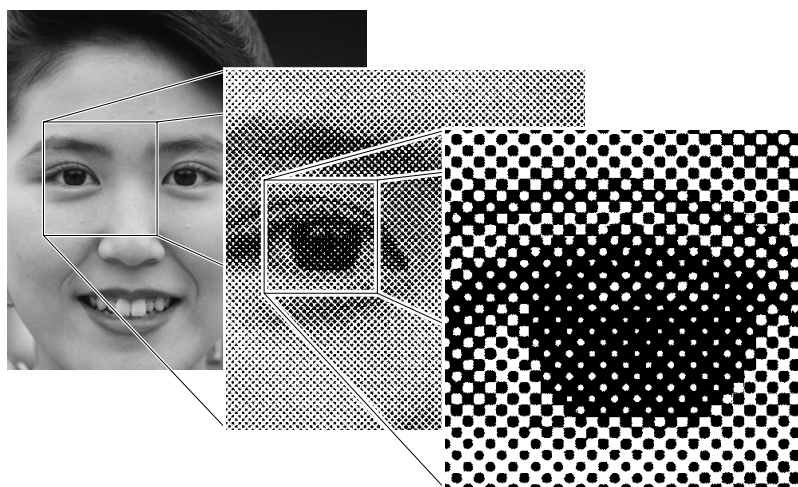


Figure 2-2 Black-and-white halftone reproduction of a grayscale image.

age is achieved by varying the spacing between pseudo-randomly placed dots that are all equally small [Sirén et al. 95]. Stochastic screening is better than conventional screening at reproducing fine details in dark areas because all the dots are small, but this advantage comes at the cost of more complicated algorithms, greater production cost, and greater sensitivity to unintended variation in dot size. While the current research is not focused on screening algorithms or the specific type of halftone used, we need to compensate for the variations in dot size that can occur in the halftones.

The dots in a halftone image often appear larger on the printed page than they do on the printing plate, an effect known as *dot gain*. The dots of ink appear larger for two reasons: ink spreads out on the paper (*physical dot gain*), and some of the light entering the paper is scattered until it emerges through dots of ink (*optical dot gain*). Mathematical models of dot gain range from simple empirically determined formulas to complicated physical simulations. Gustavson, Kruse, and colleagues have developed one of the more complex models, involving the convolution of a dot's shape with a point-spread function [Gustavson 95, Gustavson & Kruse 95, Kruse & Gustavson 95, Kruse & Gustavson 96]. Chang et al. applied numerical optimization to find the relationship between the area intended to be covered with ink and the area actually covered with ink for a number of halftone patterns other than conventional dots. Arney et al. introduced a two-parameter empirical model to account for variations in reflectance as dot size changes [Arney et al. 95a, Arney et al. 95b]. We describe a similar empirical model with only one parameter in Chapter 3 and discuss our experimental method for determining the dot gain parameter for each ink in Chapter 7.

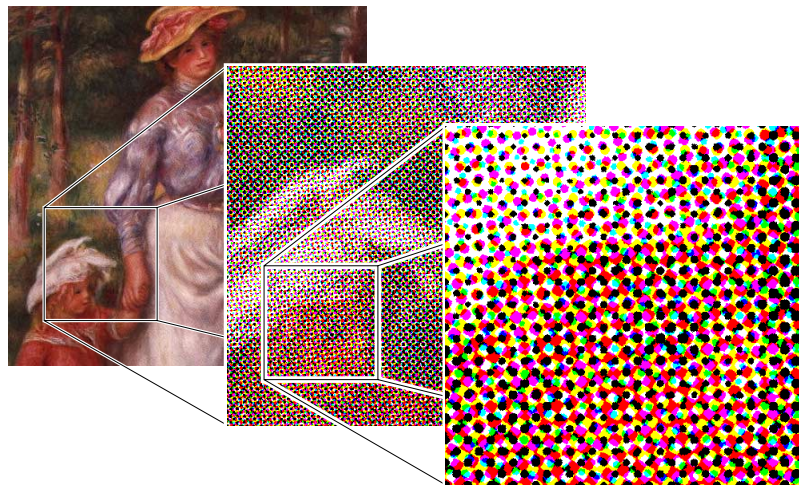


Figure 2-3 Color halftone reproduction of a continuous-tone color image.

2.3 Color halftone printing

In color halftone printing, a continuous-tone image is reproduced by printing a number of monochrome halftone images atop one another, each using a different ink. Color halftone printing differs from color dithering on monitors in that subtractive effects as well as additive effects play a role. The subtractive effect of superimposing dots of different color produces the set of *printing primaries* for a particular set of inks. For example, for cyan, magenta, and yellow ink printed on white paper, the set of printing primaries is cyan, magenta, yellow, blue (cyan + magenta), green (cyan + yellow), red (magenta + yellow), black (cyan + magenta + yellow), and white (no ink). The additive effect of juxtaposing dots of different sizes produces the entire set, or *gamut*, of colors that can be achieved by printing halftone images using a particular set of inks. Figure 2-3 illustrates the reproduction of a color image using cyan, magenta, yellow, and black inks. Further information on color halftones and color reproduction in general can be found in the classic texts by Hunt [Hunt 95] and Yule [Yule 67].

Any mathematical model of the gamut of colors achieved by halftone printing can be described as a function from ink amounts to the color produced by printing with those ink amounts. For n inks, this function maps an n -dimensional box (the n ink amounts lie between 0 and 1) to a region of three-dimensional color space. A variety of models have been proposed, each of which can be classified as either theoretical or empirical. Theoretical models can produce accurate predictions from relatively few data samples, but they require

detailed knowledge of the physical properties of the printing process. Empirical models are based only on measurement and observation; they require little knowledge of the underlying processes involved in printing, but instead depend on a large number of data samples.

A number of research articles are devoted to the development of empirical models for particular printing processes. For instance, Emmel et al. describe an empirical model of ink-jet printing [Emmel et al. 95], while Inui gives an empirical model for three subtractive dyes [Inui 93]. Lindbloom generates empirical models of three-ink printing gamuts by applying polynomial warping functions to deform a cube to fit measured color samples [Lindbloom 89]. Kress and Stevens present a similar model that fits the colors of a three-ink gamut using trivariate Bézier functions [Kress & Stevens 94]. Herzog and Hill also model three-ink gamuts empirically, but using polynomial distortions of a cylindrical color space [Herzog & Hill 95]. The challenge in each of these approaches is obtaining enough data to form an accurate empirical model. Because our aim is to print with custom inks in arbitrary combinations, we must be able to predict the colors that will result when we print with inks that have not already been measured in combination. We therefore rely on a theoretical model rather than an empirical model.

Most theoretical models of color halftone printing used today are based on the equations published by Neugebauer in 1937 [Neugebauer 37]. His model assumes that small dots of color are printed in such a way that their edges are sharply defined, their overlapping areas are distributed randomly, and within each overlapping area each ink is either completely present or completely absent. His model also assumes we know the colors of the *printing primaries*: the paper color, the color of each ink printed on paper alone, and the color of each overprinted combination of inks. Under these conditions, the Neugebauer equations state that the overall color of a small area seen from a distance is simply the area-weighted average of the colors of the printing primaries. For example, in a square printed with cyan, magenta and yellow ink on white paper, the contribution of blue is given by the fraction of the square covered by cyan and magenta but not yellow. If α_1 , α_2 , and α_3 are the amounts of cyan, magenta and yellow ink printed, then the contribution of blue is $\alpha_1\alpha_2(1 - \alpha_3)$.

Neugebauer's model was originally formulated for three inks, but it can easily be generalized to any number of inks. Furthermore, though the colors of the printing primaries are typically represented using coordinates in the *XYZ* color space, the model applies equally well to reflectance spectra or any linear transformation of *XYZ* coordinates.

The influence of the Neugebauer equations is widespread in the color printing literature; Rhodes provides a historical perspective on their impact [Rhodes 90]. Of the many

published variations of Neugebauer's model, we briefly point out a few here. Clapper and Yule modified the Neugebauer model to distinguish between the paper and the remaining printing primaries; their goal was to account for the scattering of light that occurs within many papers [Clapper & Yule 53]. Yule and Nielsen attempted to model the nonlinearities in their data for a single halftoned ink by applying a power law to the printing primaries before computing their area-weighted average [Yule & Nielsen 51]. The power they used was determined empirically to fit measurements, but their choice was later justified by physical arguments [Ruckdeschel & Hauser 78]. Many subsequent publications apply the same power law to the Neugebauer model, but with the printing primaries represented as spectra rather than tristimulus colors [Daligault & Archinard 93, Rolleston & Balasubramanian 93, Viggiano 90]. Further modifications to Neugebauer's model were made by Balasubramanian to more accurately determine the areas of printing primaries when inks are printed with their halftone grids aligned [Balasubramanian 95].

The Neugebauer model and those derived from it assume that the colors of the printing primaries are known. Once again, there are empirical and theoretical approaches to determining the printing primary colors. While it is reasonable to measure samples of all the printing primaries for a small set of inks, it becomes prohibitively expensive and time-consuming to measure all possible overprinted combinations for a large set of inks. Only a few research articles rely on theoretical models to predict the printing primaries: Emmel et al. use Beer's law to predict the color of overlapping transparent inks, thereby modeling an ink-jet printer gamut with one measurement per ink [Emmel et al. 96]. Berns uses the Kubelka-Munk model of colorant layers to accurately predict the gamut of a dye transfer printer from only 45 measured samples [Berns 93]. Liu describes a model for process color printing incorporating aspects of both the Kubelka-Munk layering model and the Neugebauer halftone model [Liu 91]. We model the printing gamuts of custom inks using a similar approach in Chapter 3, where we combine the Kubelka-Munk and Neugebauer equations while taking into account the effects of dot gain and trapping.

2.4 Printing with inks other than process color inks

The core idea of our research is to print images with inks other than the standard process colors. In recent years, a number of multicolor printing systems have been developed to achieve greater color fidelity than the process inks. The PANTONE Hexachrome system adds an orange and a green to the four process colors. Ostromoukhov's heptatone printing

method adds orange, green, and purple [Ostromoukhov 93]. Boll adds red, green, and blue [Boll 94]. Takaghi et al. dispense with process cyan and introduce a nine-ink system using red, orange, yellow, green, blue, purple, magenta, black, and gray inks [Takaghi et al. 94]. All of these systems rely on a fixed set of inks to print all images, in contrast to our approach of selecting custom inks for each image.

When we began researching custom-ink printing, we limited ourselves to the problem of reproducing full-color images with two inks. We found that many others have attempted to print color images with two inks, but their success has been limited by fixed choices of inks. For example, Harrington and Klassen describe a technique for creating a “highlight color image,” a specialized duotone in which one ink is black [Harrington & Klassen 93]. Southworth presents an algorithm (based on a photographic process developed by Wallace Edwards) for separating a color image to be printed with a red ink and a black, blue, or green ink [Southworth 95]. His approach amounts to adding a fraction of a standard yellow separation to a standard magenta separation, printing this result with red ink, and then adding a fraction of the yellow separation to a standard cyan separation, printing the result with black ink (or blue or green if the image content warrants). Our duotone research attempts to be more general than either of these approaches by choosing the best pair of inks for an image.

An interesting discussion related to duotone printing appears in Schläpfer and Widmer’s exploration of the relative sizes of various color gamuts [Schläpfer & Widmer 93]. They point out that the ideal pair of inks for printing arbitrary images as duotones should have wide, nonoverlapping transmittance spectra, with one ink transmitting in the lower half of the range of visible wavelengths and the other ink in the upper half. These properties ensure that the individual inks are as bright as possible, while their combination forms an achromatic color (making it possible to print grays). Schläpfer and Widmer distinguish PANTONE Process Blue and PANTONE Orange 021 as most closely satisfying these requirements. Although the authors measure a test chart printed with these inks, they do not endeavor to print any images. Coincidentally, our ink optimization algorithm chose Process Blue and a brownish orange ink not far from Orange 021 for several duotone reproductions of facial portraits (see Section 7.3).

In the vast color printing literature, the only article in which we found mention of the need for combinatorial optimization was one by Iwata and Marcu in which they briefly touch on the subject of choosing a good printing order for a fixed set of inks printed on fabric [Iwata & Marcu 94]. Selecting optimal colors is more prevalent in the computer graphics literature, where display devices with limited color resolution require that images be por-

trayed with a small palette of colors. Kolpatzik and Bouman present an algorithm for choosing a universal color palette that is optimized for dithered displays of all possible image colors [Kolpatzik & Bouman 95], while Iverson and Riskin describe how to find an optimal palette for simultaneous display of a number of given images [Iverson & Riskin 93]. While these palette optimization techniques rely on vector quantization methods, the ink selection algorithm we present in Chapter 5 is based on stochastic combinatorial optimization.

2.5 Gamut mapping

All printing processes are limited in the gamut of colors they can produce, and therefore in the images they can accurately reproduce. In the case of three-dimensional computer-synthesized imagery, one way to deal with this limitation is by redefining the original object colors so that the rendered image will lie inside the printable gamut [Glassner et al. 95]. However, in the vast majority of cases, the image we want to print comes from a photograph rather than a computer-generated rendering. We need a mapping that transforms image colors into printable colors without introducing discontinuities or objectionable color shifts.

Studies have shown that the least objectionable gamut mappings are those that preserve hue at the expense of luminance and saturation [Gentile et al. 90, Montag & Fairchild 96, Montag & Fairchild 97]. In accordance with these findings, most existing gamut mapping techniques maintain hue while compressing each color's luminance and saturation in one of two ways: either toward a gray of equal luminance, or toward a fixed gray of medium luminance [Johnson 96, Laihanen 87, Stone et al. 88, Wolski et al. 94]. The former approach preserves luminance relationships, but tends to desaturate brightly colored highlights until they become white. The latter approach keeps the highlights more saturated, but reduces their luminance at the same time, resulting in a reordering of brightnesses in the image. Which of these two approaches is more desirable depends upon the image content [Laihanen 87]. In Chapter 4, we develop a continuous family of gamut mappings filling the gap between these two predominant strategies, allowing the user more flexibility to achieve a quality reproduction.

There are alternatives to the simple geometric gamut mapping strategies listed above. Some authors, for example, recommend mapping dark colors differently from light colors, and achromatic colors differently from chromatic ones [Lindbloom 89, Wolski et al. 94]. Spaulding et al. construct an implicit function that warps a volume of color space into the volume occupied by the printing gamut [Spaulding et al. 95]. Their technique is based on

methods that were originally developed to warp images for computer-generated animations. The warping function is defined by establishing correspondences between key colors (or lines or planes of colors) in the image color space and the printing gamut. While the authors describe how to make reasonable correspondences when mapping an *RGB* monitor gamut to a process-color printing gamut, it is not clear how to generalize these constraints for the gamuts of arbitrary custom inks.

2.6 Computing separations

Once we have chosen the inks for a particular image and applied a gamut mapping algorithm to bring the image colors into the printable gamut, our only remaining task is to produce *separations* for each of the inks. A separation is a continuous-tone monochrome image encoding at each pixel the fractional area coverage desired of a given ink. Thus, for each color generated by the gamut mapping algorithm, we need to invert the gamut model to find the ink amounts that will reproduce that color.

Some authors rely on very simple gamut models in order to accelerate the computation of separations. For example, some of the multicolor extensions to process-color printing mentioned above assume the printing gamut can be approximated by a convex union of tetrahedra, where each tetrahedron has black ink as one vertex, white paper as another, and two adjacent chromatic ink colors as the third and fourth vertices [Iwata & Marcu 94, Marcu & Abe 94a, Marcu & Abe 94b, Ostromoukhov 93]. The advantage of this representation is that each color in the gamut model can be uniquely produced by a combination of black ink and two chromatic inks. The task of computing separations reduces to the simple problem of finding the tetrahedron containing a given color and computing that color's barycentric coordinates within the tetrahedron. Unfortunately, in reality ink amounts are often nonlinearly related to barycentric coordinates and printing gamuts are often concave, violating the assumptions of these separation techniques.

Takaghi et al. avoid these pitfalls in their approach to nine-ink printing by using neural networks to solve for separations [Takaghi et al. 94]. They associate a neural network with each triple of inks and train it to produce the ink amounts corresponding to a number of measured sample colors. Once training is complete, the neural networks can provide fairly accurate ink amounts for colors other than those in the training set.

We rely on Neugebauer's equations to model printing gamuts; we need to invert this model in order to compute separations. The equations are relatively straightforward to in-

vert in the case of one or two inks. Hardy and Wurzburg developed analog feedback circuits for inverting the three-ink case [Hardy & Wurzburg 48]. More recently Mahy and Delabastita showed that analytic solutions exist [Mahy & Delabastita 96]. However, for more than three inks analytic techniques fail because we cannot invert a function that maps from four or more ink amounts to a three-dimensional color space. We tackle this under-constrained problem using continuous optimization in Chapter 6, where we present a novel algorithm capable of robustly computing separations for any number of inks.

✦ Modeling printing gamuts ✦

In order to find the best combination of paper and inks from many possible choices, we need a mathematical model of the gamut of printable colors that results from any particular choice. The accuracy of the gamut model is not so crucial when printing with one or two inks, since the user cannot expect a perfect reproduction from a duotone. By contrast, users can be expected to be much more critical when printing with three or more inks, and therefore n -tone printing requires a much more accurate gamut model.

In the following sections we present the mathematical notation of the Neugebauer model of color halftoning, then extend the model to account for two effects commonly seen in offset printing: imperfect trapping and dot gain. The resulting model of color printing gamuts requires that we know the colors achieved by overprinting combinations of inks. As we cannot always measure these overprinted colors, we rely on mathematical models of colorant layers to predict these combinations. In the final section of this chapter, we present a range of colorant layering models and describe how they can be incorporated into the Neugebauer model. Throughout the chapter, we note which quantities are directly measurable and which can be derived from measurements, though we postpone until Chapter 7 a discussion of the experimental procedure we use to fit the gamut model's parameters to measured data.

3.1 The Neugebauer model of color halftoning

Neugebauer's model in its original formulation gives the overall color \mathbf{c} of an area printed with three inks as a function of the three coverage values α_1 , α_2 , and α_3 . The model also requires that we know the printing primaries. We refer to the color of paper as \mathbf{g}_p , the color of paper covered by the first ink as \mathbf{g}_{p1} , the color of paper printed with the first and second inks as \mathbf{g}_{p12} , and so on. With this notation we can write the Neugebauer equations for three

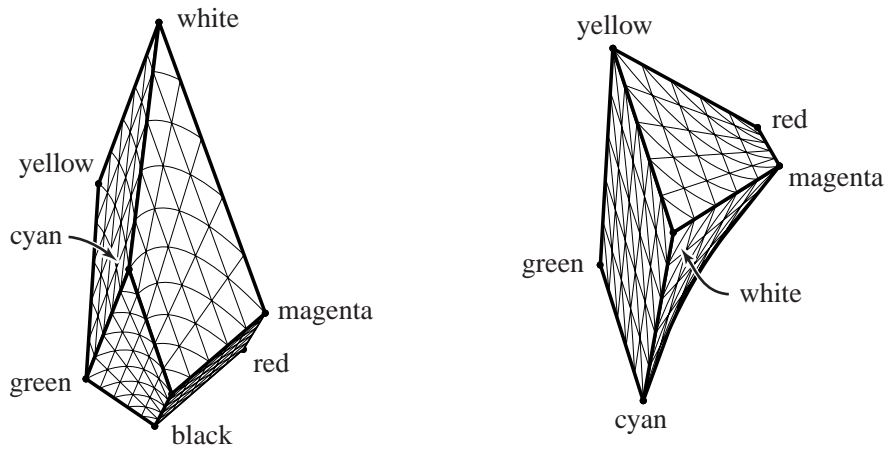


Figure 3-1 Front and top views of a three-ink gamut modeled by the Neugebauer equations.

inks as follows:

$$\begin{aligned}
 \mathbf{c}(\alpha_1, \alpha_2, \alpha_3) = & (1 - \alpha_1)(1 - \alpha_2)(1 - \alpha_3) \mathbf{g}_p \\
 & + \alpha_1(1 - \alpha_2)(1 - \alpha_3) \mathbf{g}_{p1} \\
 & + (1 - \alpha_1)\alpha_2(1 - \alpha_3) \mathbf{g}_{p2} \\
 & + \alpha_1\alpha_2(1 - \alpha_3) \mathbf{g}_{p12} \\
 & + (1 - \alpha_1)(1 - \alpha_2)\alpha_3 \mathbf{g}_{p3} \\
 & + \alpha_1(1 - \alpha_2)\alpha_3 \mathbf{g}_{p13} \\
 & + (1 - \alpha_1)\alpha_2\alpha_3 \mathbf{g}_{p23} \\
 & + \alpha_1\alpha_2\alpha_3 \mathbf{g}_{p123}
 \end{aligned}$$

The set of printable colors is generated by the function $\mathbf{c}(\alpha_1, \alpha_2, \alpha_3)$ as each of the three ink amounts is independently varied between 0 and 1. The resulting gamut is a trilinear volume in a three-dimensional color space, as illustrated in Figure 3-1. Each of the six surfaces bounding the gamut is a bilinear surface, and each of the twelve edges is a straight line segment. Leaving out one of the inks (by setting one α_i to zero) reveals that the gamut of two inks is just one of the bilinear surfaces bounding the three-ink gamut. Likewise, setting two α values to zero yields the gamut of a single ink: a straight line segment connecting the color of paper with the color of ink on paper.

Neugebauer's model is easily generalized from its original three-color formulation to incorporate any number of inks. For n inks, there are 2^n printing primaries (since each ink is either present or absent in a primary). We denote the fraction of area printed with ink i by α_i , and we write these ink amounts collectively as $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$. The general form of

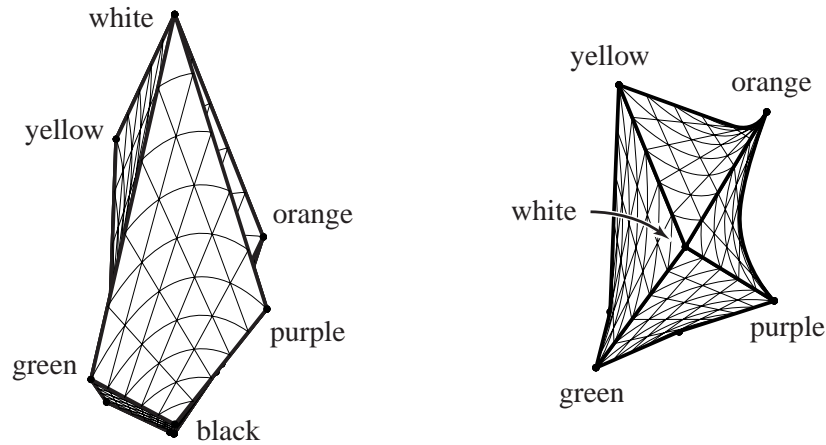


Figure 3-2 Front and top views of a four-ink gamut modeled by the Neugebauer equations.

the Neugebauer model for n inks can be written as a sum over the printing primaries (which we index here by a variable q for convenience):

$$\mathbf{c}(\boldsymbol{\alpha}) = \sum_{q=1}^{2^n} \left[\prod_{i=1}^n f(q, i, \alpha_i) \right] \mathbf{g}_q$$

$$\text{where } f(q, i, \alpha) = \begin{cases} \alpha & \text{if primary } q \text{ includes ink } i \\ 1 - \alpha & \text{otherwise} \end{cases}$$

Illustrations of the three-ink case shown in Figure 3-1 and the four-ink case in Figure 3-2 demonstrate that printing gamuts can be nonconvex—a fact that will later preclude some simple gamut mapping and color separation techniques.

3.2 Adding trapping to the Neugebauer model

Implicit in the Neugebauer model is the assumption that if we intend to cover a fraction α_i of an area with ink i , we can actually achieve that fractional coverage. In reality, because of the physical properties of inks and papers, some of the ink on the printing plate may not stick to the printed page, resulting in colors different from the Neugebauer model's predictions. The portion of ink that does stick is said to be “trapped” by the paper and previously printed inks. Liu assumed that incomplete trapping causes an ink layer to be uniformly reduced in thickness, thereby altering the colors of the printing primaries [Liu 91]. However, examination of printed samples under a microscope reveals that the colors of overprinted regions are far less uniform than the colors of the paper and the inks printed individually, as

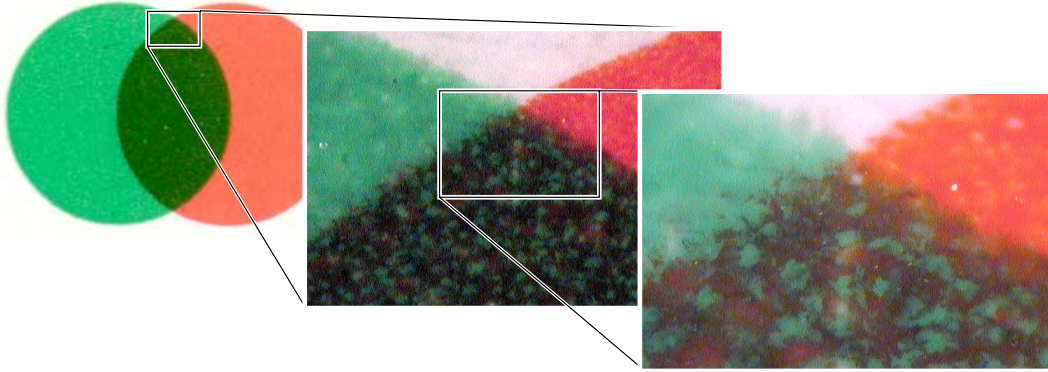


Figure 3-3 Imperfect trapping of overlapping inks: a circle of PANTONE 172 printed atop a circle of PANTONE 354, magnified about four times actual size by a flatbed scanner, and about 30 and 80 times by a microscope. The overprinted region appears blotchy because the second ink does not always stick to the areas already printed with the first ink.

shown in Figure 3-3. Instead of modeling imperfect trapping by a decrease in layer thickness, we use probabilities to describe the likelihood that a given ink will be trapped by the surface it is printed upon. To our knowledge, the approach described here is one of the novel contributions of our research.

We will denote by t_{p1} the probability that ink 1 that sticks to paper, and by t_{pij} the probability that ink j that sticks to ink i (the *trapping fraction* for ink j on ink i on paper). With this convention, we can model the color \mathbf{c} we get by *trying* to cover a fraction α_1 of the paper with ink 1:

$$\begin{aligned} \mathbf{c}(\boldsymbol{\alpha}) = & (1 - t_{p1}\alpha_1) \mathbf{g}_p \\ & + (t_{p1}\alpha_1) \mathbf{g}_{p1} \end{aligned} \quad (3.1)$$

Now suppose we print a second ink on top of that result. Of the area that was the color of paper \mathbf{g}_p , a fraction $t_{p2}\alpha_2$ will get covered by ink 2 and become \mathbf{g}_{p2} , while the rest will stay the same. Likewise, in the area that was colored \mathbf{g}_{p1} , a fraction $t_{p12}\alpha_2$ will be overprinted with ink 2 and become \mathbf{g}_{p12} , while the rest will stay the same. Thus, the result is a weighted average of four colors (as in the Neugebauer model):

$$\begin{aligned} \mathbf{c}(\boldsymbol{\alpha}) = & (1 - t_{p1}\alpha_1) (1 - t_{p2}\alpha_2) \mathbf{g}_p \\ & + (t_{p1}\alpha_1) (1 - t_{p12}\alpha_2) \mathbf{g}_{p1} \\ & + (1 - t_{p1}\alpha_1) (t_{p2}\alpha_2) \mathbf{g}_{p2} \\ & + (t_{p1}\alpha_1) (t_{p12}\alpha_2) \mathbf{g}_{p12} \end{aligned} \quad (3.2)$$

If we add a third ink, the result will be a weighted average of eight colors, and we need seven trapping fractions:

$$\begin{aligned}
\mathbf{c}(\boldsymbol{\alpha}) = & (1 - t_{p1}\alpha_1) (1 - t_{p2}\alpha_2) (1 - t_{p3}\alpha_3) \mathbf{g}_p \\
& + (t_{p1}\alpha_1) (1 - t_{p12}\alpha_2) (1 - t_{p13}\alpha_3) \mathbf{g}_{p1} \\
& + (1 - t_{p1}\alpha_1) (t_{p2}\alpha_2) (1 - t_{p23}\alpha_3) \mathbf{g}_{p2} \\
& + (t_{p1}\alpha_1) (t_{p12}\alpha_2) (1 - t_{p123}\alpha_3) \mathbf{g}_{p12} \\
& + (1 - t_{p1}\alpha_1) (1 - t_{p2}\alpha_2) (t_{p3}\alpha_3) \mathbf{g}_{p3} \\
& + (t_{p1}\alpha_1) (1 - t_{p12}\alpha_2) (t_{p13}\alpha_3) \mathbf{g}_{p13} \\
& + (1 - t_{p1}\alpha_1) (t_{p2}\alpha_2) (t_{p23}\alpha_3) \mathbf{g}_{p23} \\
& + (t_{p1}\alpha_1) (t_{p12}\alpha_2) (t_{p123}\alpha_3) \mathbf{g}_{p123}
\end{aligned} \tag{3.3}$$

In general, for n inks the resulting color will be a weighted average of the 2^n printing primaries, where the weights depend on $2^n - 1$ trapping fractions. The shape of a printing gamut model that includes these trapping fractions is essentially the same as the original Neugebauer model's shape; it is still a multilinear interpolant, but with some of the corner points moved to reduce its range.

3.3 Adding dot gain to the Neugebauer model

In addition to the effects of trapping, offset printing is subject to *dot gain*. The halftoned dots of an ink appear larger than they should for two reasons: ink spreads out on the paper (*physical dot gain*), and some of the light entering the paper is scattered until it emerges through dots of ink (*optical dot gain*). We can account for both varieties of dot gain using an empirical model that corrects the value of α_i for each ink. When we produce a halftone separation that specifies a coverage $\bar{\alpha}_i$, we find that one minus the actual coverage in the printed result is approximated very closely by a power law:

$$1 - \alpha_i = (1 - \bar{\alpha}_i)^{\gamma_i} \tag{3.4}$$

The parameter γ_i associated with ink i can be determined from experimental data using standard curve-fitting techniques, as discussed in Chapter 7. Note that if we desire an actual coverage of α_i , we can always solve the equation above for $\bar{\alpha}_i$, the coverage we should specify.

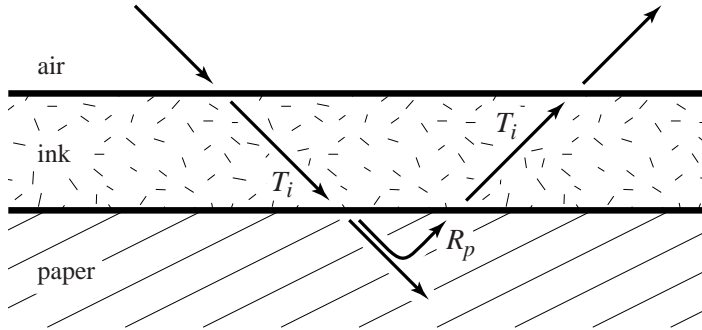


Figure 3-4 Modeling a layer of ink as an ideal filter.

3.4 Modeling the printing primaries

We have so far assumed that we know the colors of the printing primaries. While we can easily measure these colors for a small set of inks (like the process inks) on a small set of papers, it is impractical to do so for all the combinations that could be chosen from large sets of inks and papers. If we want to print on a new paper without measuring all our inks on that paper, we need a model capable of predicting the primaries. There are many levels of complexity we can introduce into a model; we will start by presenting a simple model and progress to more complicated ones.

A single ink layer

If we assume that a layer of ink acts as an ideal filter, we need to know only how much light it transmits at each wavelength λ . We will write the transmittance of the ink as T_i , and the reflectance of paper as R_p (for some wavelength λ). As Figure 3-4 illustrates, the reflectance of ink on paper is given by the amount of light that penetrates the ink (T_i), reflects off the paper (R_p), and emerges through the ink again (T_i):

$$R_{pi} = T_i^2 R_p \quad (3.5)$$

We can measure R_{pi} and R_p using a spectrophotometer, but not T_i because it is a property of the ink layer without paper. However, we can characterize an ink by printing an identical layer of that ink on a variety of papers, measuring R_p and R_{pi} for each paper, then fitting T_i to the model. Unfortunately, a single transmittance spectrum may not be enough information to accurately model an ink on paper, let alone one ink atop another.

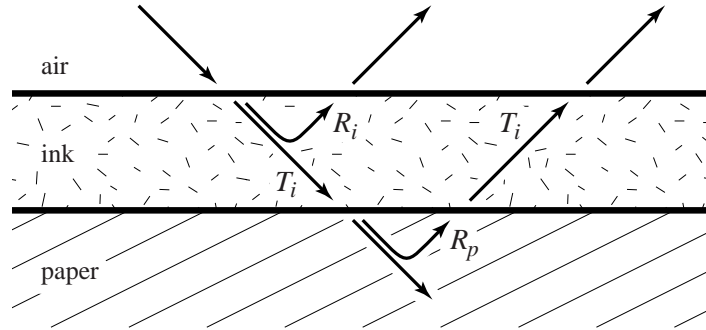


Figure 3-5 Modeling a layer of ink as a reflecting filter.

One problem with the simple model above is that inks reflect some light in addition to absorbing and transmitting light. If we introduce a reflectance R_i for the ink, as shown in Figure 3-5, we have

$$R_{pi} = R_i + T_i^2 R_p \quad (3.6)$$

Once again, we can measure R_p and R_{pi} for a single ink on a variety of papers, and fit R_i and T_i to the model.

Equation (3.6) is subsumed by a more general layering model, known as the Kubelka-Munk model (described by Judd and Wyszecki [Judd & Wyszecki 75, pages 420–438] and by Kortüm [Kortüm 69], among others). As Figure 3-6 illustrates, light can reflect any number of times between the ink and paper before finally exiting the ink layer, making the reflectance of ink on paper an infinite sum of terms:

$$R_{pi} = R_i + T_i^2 R_p (1 + \tilde{R}_i R_p + \tilde{R}_i^2 R_p^2 + \cdots) = R_i + \frac{T_i^2 R_p}{1 - \tilde{R}_i R_p} \quad (3.7)$$

The reflectance of the back side of the ink layer, \tilde{R}_i , can differ from the reflectance of its front side because the layer may be inhomogeneous. Now we must fit three reflectance spectra (R_i , \tilde{R}_i , and T_i) to measured data in order to characterize an ink. If \tilde{R}_i is identically zero, we are left with equation (3.6); if R_i is also zero, we are left with equation (3.5).

The Kubelka-Munk model can be derived from physical principles, but only under certain assumptions. One assumption is that all the layers have the same index of refraction. However, the index of refraction of a colorant layer is typically between 1.45 and 1.6, while that of air is very nearly 1 [Judd & Wyszecki 75, page 398]. As a result of the difference in indices, some of the incident light will undergo Fresnel reflection at the material interface.

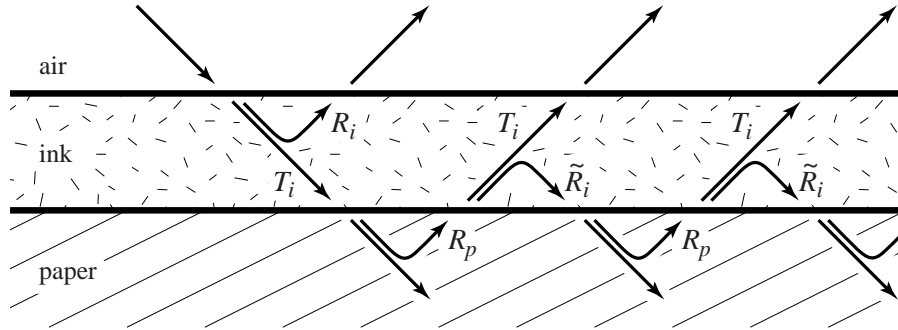


Figure 3-6 Kubelka-Munk model of a layer of ink.

The reflectance R_{pi} we have modeled in the equations above is actually the reflectance we would measure if the sample and spectrophotometer were immersed in an oil with the same index of refraction as the ink layer. We can correct this value to account for Fresnel reflection at the boundary between ink and air using a construction similar to that of the Kubelka-Munk model. Suppose ρ_{ai} is the fraction of diffuse light traveling from air to ink that is reflected by the surface of the ink layer, and ρ_{ia} is the surface reflectance for light going from ink to air.¹ Then, according to Saunderson [Saunderson 42], we find the corrected reflectance R'_{pi} of a layer of ink on paper by modifying the prediction given by equation (3.5), (3.6), or (3.7) as follows:

$$R'_{pi} = \rho_{ai} + (1 - \rho_{ai})(1 - \rho_{ia}) \frac{R_{pi}}{1 - \rho_{ia}R_{pi}} \quad (3.8)$$

Fresnel reflection may also occur at the boundary between ink and paper if the interface is planar. In this case, we adjust our earlier equations using surface reflection coefficients ρ_{ip} (for light going from ink to paper) and ρ_{pi} (for the opposite direction). We simply replace R_p (a reflectance measured in air) in equations (3.5), (3.6), or (3.7) with a reflectance R'_p adjusted to the ink's index of refraction:

$$R'_p = \rho_{ip} + (1 - \rho_{ip})(1 - \rho_{pi}) \frac{R_p}{1 - \rho_{pi}R_p} \quad (3.9)$$

Note that in the previous two equations R_p and R'_{pi} are measurable quantities, while the Fresnel reflection coefficients as well as R_i , \tilde{R}_i , and T_i are characteristics of the ink that we need

¹ In theory, these Fresnel reflection coefficients are functions of wavelength. In practice, their variations with wavelength are negligible when compared to the transmission and reflection properties of the ink pigments themselves, so we treat them as constants.

to derive from other measurements. We discuss numerical methods of determining these model parameters from data when we present our experimental work in Chapter 7.

Multiple ink layers

Our goal in introducing ink layering models is to model the colors of the printing primaries, particularly those that consist of multiple overlapping layers of inks. Assuming we know the reflectance and transmittance properties of each ink, we can adapt the Kubelka-Munk layering model of equation (3.7) to predict the reflectance of more than one ink layer on paper. If we print on paper p using ink i followed by ink j , we can compute the resulting reflectance R_{pij} from the reflectance of ink i on paper:

$$R_{pij} = R_j + \frac{T_j^2 R_{pi}}{1 - \tilde{R}_j R_{pi}} \quad (3.10)$$

This approach—adding a layer to a previously computed reflectance—generalizes to a procedure for computing the reflectance of any number of ink layers. Here are the steps we must take, including the modifications to account for Fresnel reflection:

1. Adjust the measured paper reflectance to account for Fresnel reflection at the interface between paper and ink using equation (3.9):

$$R_p' = \rho_{ip} + (1 - \rho_{ip})(1 - \rho_{pi}) \frac{R_p}{1 - \rho_{pi} R_p}$$

2. Use the Kubelka-Munk model to find the reflectance of the first ink on paper:

$$R_{pi} = R_i + \frac{T_i^2 R_p'}{1 - \tilde{R}_i R_p'}$$

3. Compute the reflectance of two ink layers from the reflectance of the first:

$$R_{pij} = R_j + \frac{T_j^2 R_{pi}}{1 - \tilde{R}_j R_{pi}}$$

4. Compute the reflectance of three ink layers from the reflectance of two layers:

$$R_{pijk} = R_k + \frac{T_k^2 R_{pij}}{1 - \tilde{R}_k R_{pij}}$$

5. Continue in this manner for as many ink layers as needed.
6. Adjust the final reflectance to account for Fresnel reflection at the interface between air and ink using equation (3.8):

$$R'_{p_{ijk}\dots} = \rho_{ai} + (1 - \rho_{ai})(1 - \rho_{ia}) \frac{R_{p_{ijk}\dots}}{1 - \rho_{ia}R_{p_{ijk}\dots}}$$

We can follow the recipe above to predict the reflectance spectrum of each overprinted combination for a given set of inks, thereby obtaining values for the printing primaries of the gamut model in Section 3.2. Even with our trapping and dot gain modifications, the Neugebauer model is linear in the printing primaries, and therefore applies equally well to primaries specified by reflectance spectra or by *XYZ* colors. We will make the most efficient use of the gamut model by assigning to $\mathbf{g}_{p_{ijk}\dots}$ the result of converting to *XYZ* coordinates the reflectance spectra obtained in step 6 above. As we mentioned in the previous chapter, this conversion is accomplished by integrating the reflectance spectra against three response functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ [Judd & Wyszecki 75, pages 125–153 and 472–479].

✦ Gamut mapping ✦

Suppose we want to reproduce an image using a particular combination of paper and inks. We can use the model presented in the previous chapter to predict the gamut of all printable colors associated with this choice of paper and inks. In most cases, many of the image colors will lie outside the gamut of printable colors. Our goal in this chapter is to define a gamut mapping function that associates a printable color with each of the original image colors without introducing unnecessary color distortion into the image's appearance. The result of applying this mapping will be a preview image predicting the printed appearance of the original image.

Stone et al. summarize the principles required to preserve the appearance of an image during the course of gamut mapping [Stone et al. 88]. In order of decreasing importance, these principles are as follows:

1. The gray axis of the image should be preserved.
2. Maximum luminance contrast is desirable.
3. Few colors should lie outside the destination gamut.
4. Hue and saturation shifts should be minimized.
5. It is better to increase than to decrease the color saturation.

As noted by Stone et al., these are only guidelines, and their relative importance may change according to the image or the printing gamut. Because the number of inks significantly affects the shape of the printing gamut, we prioritize the principles above very differently for one, two, and three or more inks. The following sections describe the gamut mapping algorithms we use in these different cases. The one- and two-ink cases are simple enough that applying the gamut mapping to a color produces not only a new color, but also the ink amounts α needed to reproduce that color. In the case of three or more inks, however, we map each color to a color lying within the geometric boundary of the printing

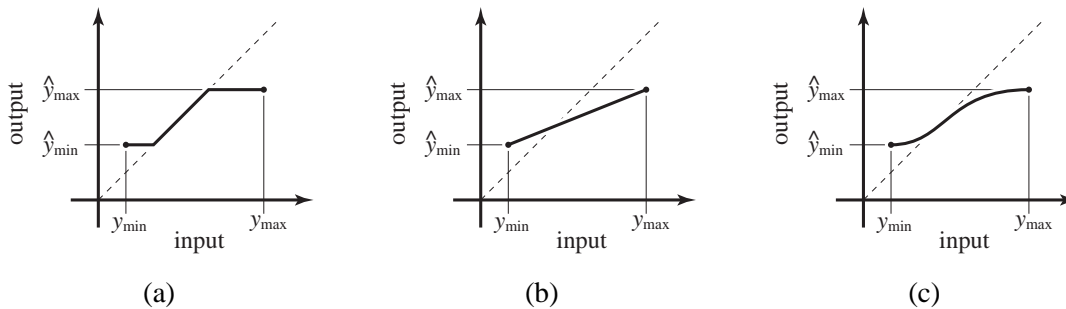


Figure 4-1 Monotonically increasing mappings: (a) clamped mapping, (b) linear mapping, and (c) cubic mapping.

gamut without determining the corresponding ink amounts; we leave the problem of finding α values for a printable color until we discuss color separations in Chapter 6.

4.1 Gamut mapping for one ink

The gamut of colors that can be printed with one ink is given by equation (3.1); it consists of a line segment in color space, parameterized by α_1 . With such a limited printing gamut, we are hard pressed to satisfy the principles listed above. Unless we are printing with black ink on white paper, we have little hope of maintaining grays in the image (the first principle above). We can, however, maximize the luminance contrast in our reproduction, by maintaining luminance relationships in the image while using the full range of printable colors.¹ Because the printing gamut is one-dimensional, once we choose to maximize luminance contrast there is little we can do to prevent changes in hue and saturation.

The first step of our gamut mapping algorithm for one ink is to remap the input image's range of luminance values $[y_{\min}, y_{\max}]$ to lie within the luminance range of the printing gamut $[\hat{y}_{\min}, \hat{y}_{\max}]$. The range of luminance values in an image is almost always larger than the printing gamut's range, so some compression is required. Gamut mapping experiments in the literature typically accomplish this compression by either clamping luminance values to the available range, as in Figure 4-1(a), or by a linear rescaling, as in Figure 4-1(b) [Gentile et al. 90, Montag & Fairchild 97].

¹ We assume here that the luminance of the paper printed with ink differs significantly from the luminance of paper alone—a safe assumption, except in the rare case of a very opaque ink differing from the paper in hue but not luminance.

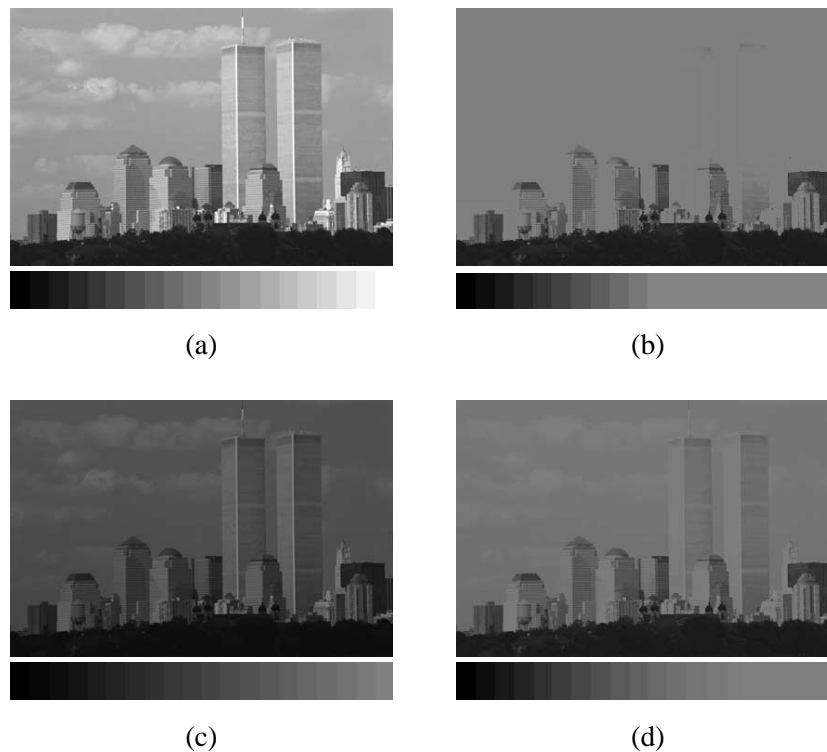


Figure 4-2 Effects of exaggerated luminance mappings on an image: (a) original image, (b) clamped mapping of luminance, (c) linear mapping of luminance, and (d) cubic mapping of luminance.

We can write the clamped mapping of Figure 4-1(a) in mathematical form as follows:

$$\mathit{clamp}(y) = \begin{cases} \hat{y}_{\min} & \text{if } y \leq \hat{y}_{\min} \\ \hat{y}_{\max} & \text{if } y \geq \hat{y}_{\max} \\ y & \text{otherwise} \end{cases}$$

The advantage of clamping is that it preserves exactly those luminance values that are printable. However, any unprintable luminance values are collapsed by clamping to either the darkest or lightest luminance level, rendering distinct image colors indistinguishable. This effect can be seen by comparing the grayscale image in Figure 4-2(a) to a version that has been clamped at the 50% luminance level in Figure 4-2(b). Further artifacts can appear when smooth gradations of luminance in the original image develop discontinuities in their first derivative because of the kinks in a clamped mapping.

We can avoid many of the disadvantages of clamping by using a linear mapping, which

we write as follows:

$$\text{linear}(y) = \left(\frac{y_{\max} - y}{y_{\max} - y_{\min}} \right) \hat{y}_{\min} + \left(\frac{y - y_{\min}}{y_{\max} - y_{\min}} \right) \hat{y}_{\max}$$

The drawback of this alternative is that a linear mapping compresses all the image's luminance values, even though some values could have been reproduced accurately. The effect of a linear mapping is illustrated in Figure 4-2(c), where the luminance ranges from 0 to 50% of the original luminance values.

We are not limited to clamped and linear mappings, however; we can use any monotonically increasing function to accomplish the compression of luminance. The ideal mapping would interpolate the two endpoints of the luminance range, be both smooth and invertible, and come as close as possible to preserving luminance values (i.e., lie near the identity function), as illustrated in Figure 4-1(c). In our work on duotone printing, we aimed to meet these goals with a mapping based on two parametric cubic Bézier curves [Power et al. 96]. We subsequently found that a single cubic function works equally well and costs less to construct and evaluate.

For a given input range $[y_{\min}, y_{\max}]$ and output range $[\hat{y}_{\min}, \hat{y}_{\max}]$, we form a cubic mapping from the composition of three functions:

$$\text{cubic}(y) = (C_3 \circ C_2 \circ C_1)(y)$$

The first function linearly transforms the input parameter y to lie in the interval $[0, 1]$:

$$C_1(y) = (y - y_{\min}) / (y_{\max} - y_{\min})$$

Next, we construct a cubic function $C_2(C_1) = \mu_1 C_1^3 + \mu_2 C_1^2 + \mu_3 C_1 + \mu_4$. We determine the coefficients μ_1, \dots, μ_4 by requiring this cubic to map 0 to itself and 1 to itself while constraining the slopes at 0 and 1 to be the constants (as yet undetermined) σ_{\min} and σ_{\max} , respectively:

$$C_2(0) = \mu_4 = 0$$

$$C_2(1) = \mu_1 + \mu_2 + \mu_3 = 1$$

$$C_2'(0) = \mu_3 = \sigma_{\min}$$

$$C_2'(1) = 3\mu_1 + 2\mu_2 + \mu_3 = \sigma_{\max}$$

Solving these linear constraints, we obtain

$$C_2(C_1) = (\sigma_{\min} + \sigma_{\max} - 2) C_1^3 + (3 - 2\sigma_{\min} - \sigma_{\max}) C_1^2 + \sigma_{\min} C_1$$

The third function comprising the cubic mapping is another linear transform that maps values of C_2 from the unit interval to the output range $[\hat{y}_{\min}, \hat{y}_{\max}]$:

$$C_3(C_2) = (1 - C_2) \hat{y}_{\min} + C_2 \hat{y}_{\max}$$

We now need only determine the values of σ_{\min} and σ_{\max} . We want to find slopes between 0 and 1 that yield an increasing function with the desired properties. In the degenerate case where the output range reduces to a single value ($\hat{y}_{\min} = \hat{y}_{\max}$), the cubic function should have slopes of zero at both ends. When the output range of luminance values exceeds the input range, we can actually preserve luminance values. Therefore, we want the slope at the left endpoint to be 1 when $\hat{y}_{\min} \leq y_{\min}$, and we want the slope at the right endpoint to be 1 when the $\hat{y}_{\max} \geq y_{\max}$. We can easily meet these constraints as follows:

$$\sigma_{\min} = \min \left\{ 1, \left(\frac{\hat{y}_{\max} - \hat{y}_{\min}}{\hat{y}_{\max} - y_{\min}} \right)^\beta \right\}$$

$$\sigma_{\max} = \min \left\{ 1, \left(\frac{\hat{y}_{\max} - \hat{y}_{\min}}{y_{\max} - \hat{y}_{\min}} \right)^\beta \right\}$$

We can choose any positive value for the exponent β . Trial and error indicates that setting $\beta = 4$ yields a curve that approaches the identity function near the middle of the input range. The image in Figure 4-2(d) illustrates the results of a cubic mapping of luminance. This image maintains more contrast in the dark colors than the linearly mapped version, without completely obliterating the World Trade Center as the clamped version did.

After applying one of the mappings described above to the luminance values of each image color, only one step remains to complete the gamut mapping. We project each of the image colors onto the line segment representing the printing gamut, taking care to preserve the luminance values we just obtained. Computing the projection of a color onto the gamut is intimately tied to computing the value of α_1 , the amount of ink required to produce the gamut-mapped color. More on this topic appears in Chapter 6 where we discuss color separations. Figure 4-3 illustrates the geometric effect in color space of both gamut mapping steps (luminance compression and projection) on a set of image colors.

4.2 Gamut mapping for two inks

The gamut produced by two inks is a bilinear surface given by equation (3.2). As in the one-ink case, the gamut has lower dimensionality than the space of colors, and therefore we must

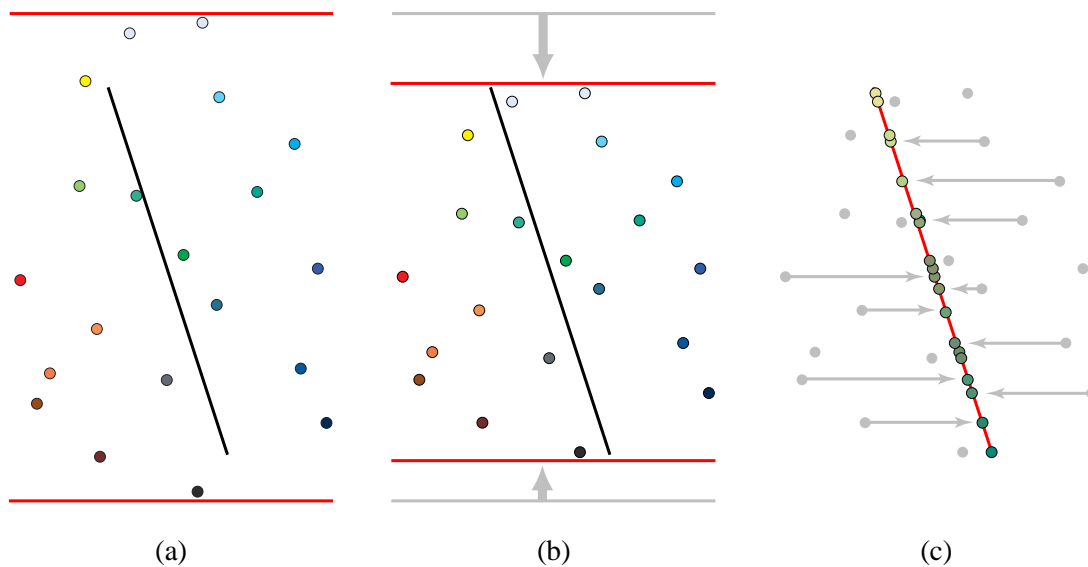


Figure 4-3 Steps of gamut mapping for one ink: (a) original colors, (b) colors after luminance mapping, and (c) colors after projection.

resort to projection within any gamut mapping. At the core of our work on duotone printing is a method of mapping image colors to a two-ink gamut while preserving as much color information as possible [Power et al. 96].

The basic idea behind our mapping technique is to define an orthogonal axis system based on the duotone gamut, then compress the image colors in two of the directions and use parallel projection along the third. The choice of directions is clearly critical to the effectiveness of the mapping and corresponds to preserving certain relationships between image colors at the expense of other relationships.

1. **Preserve relative luminance.** With two inks, we are typically unable to preserve grays (the first principle at the beginning of the chapter), but we can maximize luminance contrast. The Y axis of the XYZ color space corresponds exactly to the luminance direction, and therefore we take Y to be the direction of our first transformation.
2. **Preserve ink-spread separation.** The second relationship our mapping preserves is separation in the direction of most hue variation on the duotone gamut. The curve on the gamut between the two individual ink primaries describes the widest variation in hue achievable with the selected paper and inks. Thus, the vector $\mathbf{g}_{p2} - \mathbf{g}_{p1}$ is the ideal direction for the second transformation. However, our duotone mapping requires an orthog-

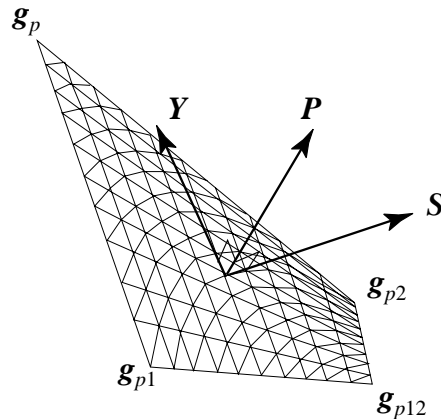


Figure 4-4 Axes associated with a typical duotone gamut.

onal axis system, and we have already chosen the Y axis. Therefore we orthogonalize $\mathbf{g}_{p2} - \mathbf{g}_{p1}$ with respect to Y and use the resulting *ink-spread direction* S as the direction of the second transformation.

- 3. Sacrifice normal separation.** Separation must be sacrificed in some direction in order to map points from three-dimensional space onto a surface. As we have chosen two axes and require mutual orthogonality, the *projection direction* P is already defined by the cross product $Y \times S$. The P axis approximates the average normal of the bilinear surface, which is the direction of least color variation on the gamut. It is therefore a good direction in which to sacrifice separation.

For a given two-ink printing gamut, the constraints listed above completely define the orthogonal axis system we use for our duotone mapping. A typical example is illustrated in Figure 4-4. Once the axis system is defined, gamut mapping for two inks takes place in three steps. First, we compress the luminance values of all image colors to fit within the luminance range of the printing gamut. We accomplish this compression exactly as we did for one-ink gamuts, by mapping y values to \hat{y} values using one of the three functions shown in Figure 4-1.

The second transformation is along the ink-spread axis S . This transformation depends upon luminance, since at some luminance values the duotone gamut is wide and at others it consists of a single point. Consider a particular luminance value \hat{y} between \hat{y}_{\min} and \hat{y}_{\max} . We define \hat{s}_{\min} and \hat{s}_{\max} to be the s -values of the points found by intersecting the edges of

the duotone gamut with a plane of constant luminance \hat{y} . The ink-spread transformation at luminance value \hat{y} brings the s -values of all image colors with luminance value \hat{y} , which range from s_{\min} to s_{\max} , into the range $[\hat{s}_{\min}, \hat{s}_{\max}]$.

In our implementation, the non-uniformity along Y is handled by separating colors into bins according to y -value and calculating a different transformation for each bin. The coherence among colors in most natural images prevents this discrete approach from introducing noticeable discontinuities into a duotone print. In order to guarantee smoothness in the gamut mapping, we can apply piecewise-linear (rather than piecewise-constant) interpolation to the values of s_{\min} and s_{\max} stored with each bin.

Although using linear interpolation improves the coherence of mappings from one bin to the next, the interpolated values of s_{\min} and s_{\max} do not strictly bound the s values of the image colors. We developed an alternative method that does bound the s values and simultaneously achieves greater smoothness. We define both $s_{\min}(y)$ and $s_{\max}(y)$ to be cubic spline functions of luminance whose coefficients are initially unknown. We then determine the coefficients by solving a constrained optimization problem: find the coefficients closest to zero for each curve, subject to the constraint that the s_{\min} curve lies to the left of every image color in the ink-spread direction and the s_{\max} curve lies to the right. Both the objective function and the constraints of this problem are linear in the unknown coefficients, and therefore we can find a global minimum by applying the simplex method [Press et al. 92]. The result is a pair of curves that define an ink-spread mapping that varies smoothly with luminance. While this solution is elegant, we ordinarily cannot distinguish its results from those of linear interpolation.

The ink-spread transformation for any particular y value, like the luminance transformation, can make use of any of the monotonically increasing functions in Figure 4-1 to map the s -values of image colors to the s -values available on the gamut.

After transforming each image color by the luminance and ink-spread mappings above, the final step of a duotone gamut mapping is to project the colors in the P direction onto the gamut. We find the projection of a given color by solving a set of nonlinear equations for the intersection of a line and the bilinear surface defined by the Neugebauer model. We postpone a derivation of the analytic solution to this problem until our discussion of color separations in Chapter 6, since projecting a color onto the gamut also yields the ink amounts α_1 and α_2 required for the duotone printing process.

Figure 4-5 illustrates the geometric effect in color space of the three steps above (luminance compression, ink-spread compression, and projection) on a set of image colors.

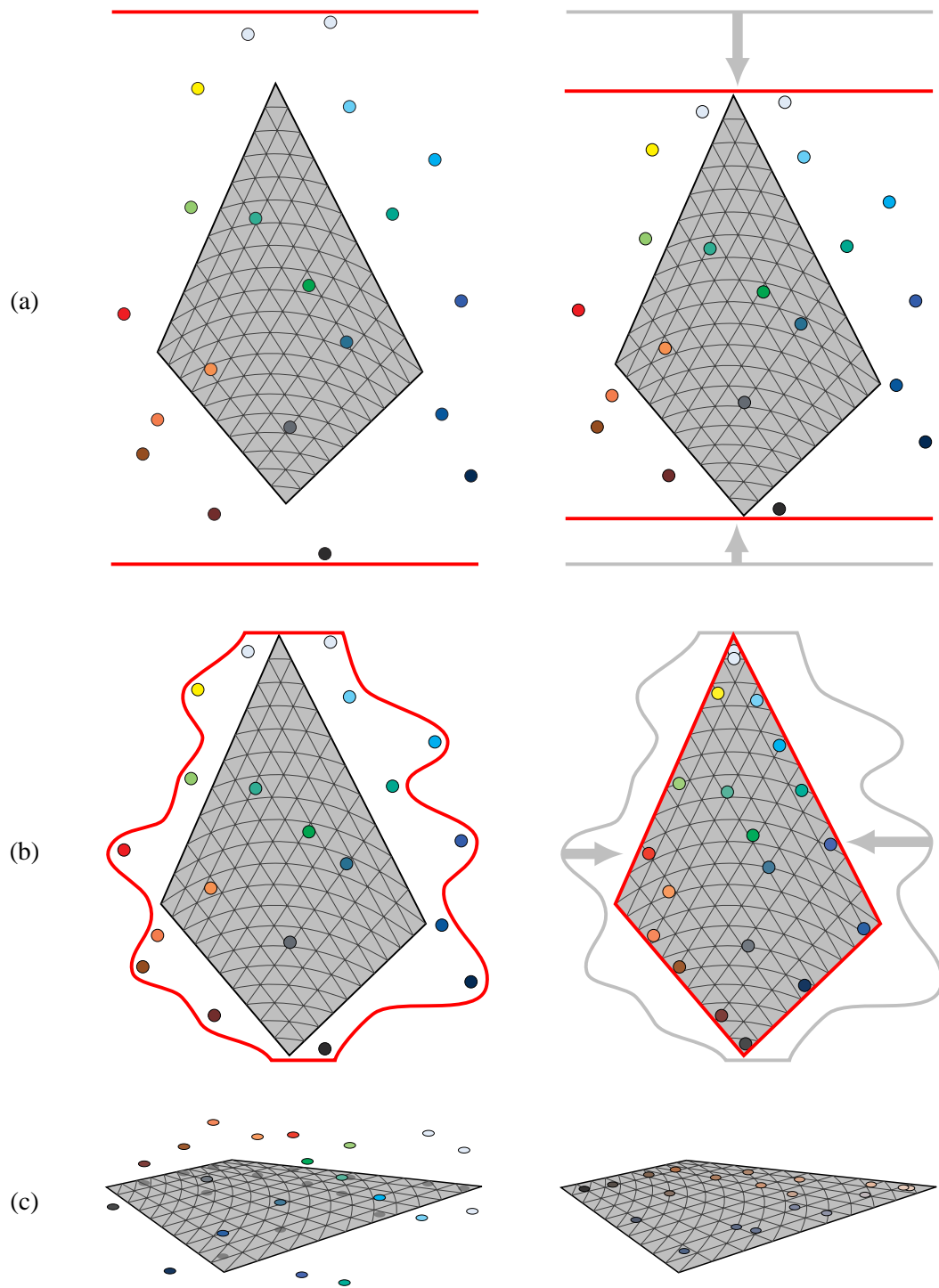


Figure 4-5 Steps of gamut mapping for two inks: (a) before and after luminance mapping, (b) before and after ink-spread compression, and (c) before and after projection.

4.3 Gamut mapping for three or more inks

The gamut of three or more inks occupies a volume in color space, though from the examples in Figures 3-1 and 3-2, it is clear that this volume may have an unusual nonconvex shape.

When we are dealing with an n -tone printing gamut occupying a volume, we need not project the image colors onto a lower-dimensional space as we do for one- and two-ink gamuts. We have the option of preserving the hue or saturation of image colors (principle 4 from the beginning of the chapter), in addition to maximizing luminance contrast. According to a number of articles that address gamut mapping, it is most important to maintain the hue of a color, while allowing its saturation to change in order to fit within the printing gamut [Gentile et al. 90, Johnson 96, Laihanen 87, Montag & Fairchild 97, Stone et al. 88, Wolski et al. 94]. In the sections that follow, we devise a hue-preserving gamut mapping strategy that is more general than existing ones, then present its coordinate system and algorithmic details.

4.3.1 Strategy of n -tone mapping

Among the hue-preserving gamut mapping algorithms in the literature, there are two predominant strategies for altering luminance and saturation: the first reduces saturation, leaving luminance fixed [Laihanen 87, MacDonald 93, Stone et al. 88]; the second simultaneously alters luminance and saturation toward the central gray of the gamut [Laihanen 87, MacDonald 93].

Because the first approach maps colors into the printing gamut by reducing their radial distances from a central gray axis, we refer to it as a “cylindrical” mapping. Likewise, because the second approach reduces each color’s distance from a single central gray point, we refer to it as a “spherical” mapping. Laihanen notes that depending on the image being reproduced, one may be preferable to the other [Laihanen 87]. The cylindrical mapping has the advantage of preserving luminance relationships, but it tends to desaturate brightly colored highlights until they become white. The spherical mapping keeps the highlights more saturated, but reduces their luminance at the same time, resulting in a reordering of brightnesses in the image.

In order to obtain some of the advantages of both the cylindrical mapping and the spherical mapping, we developed a parameterized family of intermediate mappings. While we could simply interpolate between the color given by the cylindrical mapping and the color

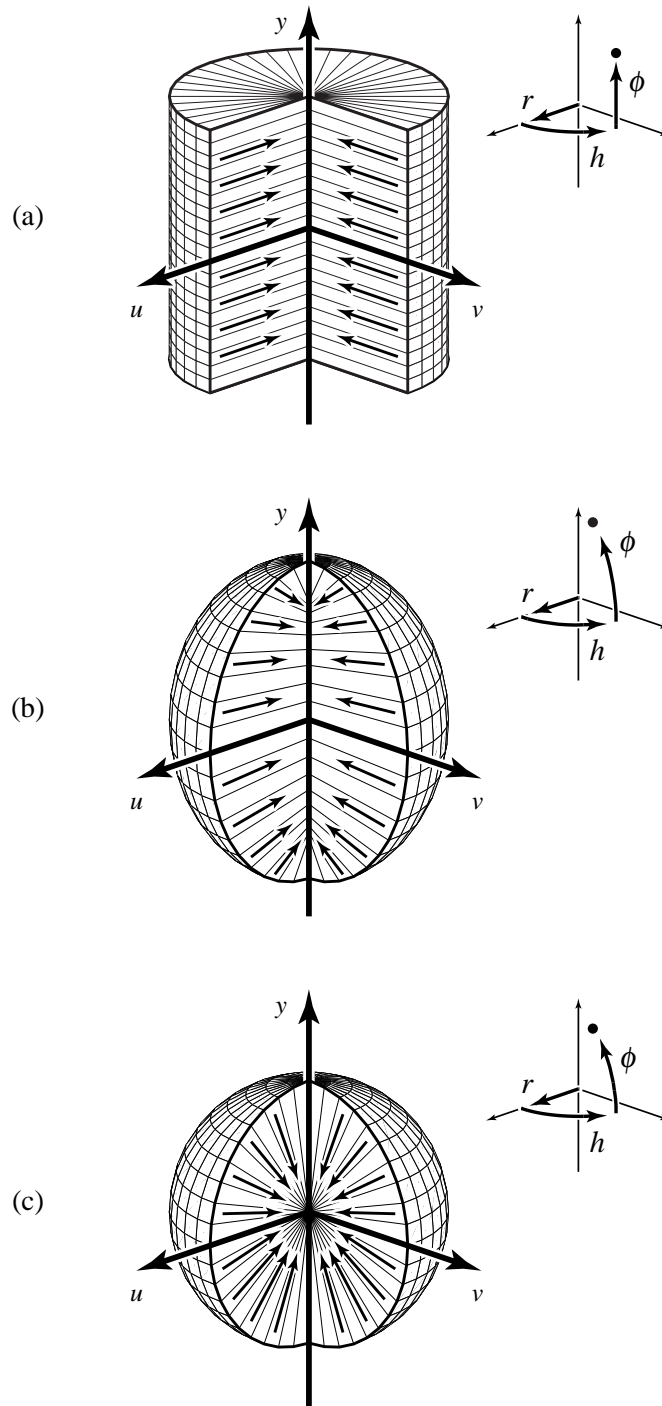


Figure 4-6 The coordinate systems used by three gamut mapping methods chosen from a continuum: (a) cylindrical ($\kappa = 0$), (b) ellipsoidal ($\kappa = 0.3$), and (c) spherical ($\kappa = 1$). Arrows on the cutaway surfaces indicate the directions in which colors are compressed.

given by the spherical mapping, there would be no guarantee that the result would lie in the printing gamut (because gamuts are not necessarily convex). Instead, we vary the locus of colors that serve as the centers of projection in the mapping: the cylindrical mapping moves colors toward a fixed line segment along rays orthogonal to a cylinder; the spherical mapping moves colors toward a single point along rays orthogonal to a sphere; our new mapping moves colors toward a line segment whose length is parameterized, along rays orthogonal to an ellipsoid. Figure 4-6 illustrates the directions in which colors are compressed by each type of mapping. These directions are made explicit in the following section.

4.3.2 Coordinate system of n -tone mapping

The implementation of our n -tone gamut mapping makes use of a special-purpose coordinate system that varies according to the parameter κ . The coordinate system yields a cylindrical mapping when $\kappa = 0$, a spherical mapping when $\kappa = 1$, and an ellipsoidal mapping for intermediate values. Transforming an XYZ color into this coordinate system takes place in two stages. The first is a linear transform that rewrites the color as a triple (u, v, y) , where y represents luminance and u and v hold the chrominance information. This linear transform shears the dark-to-light axis of the printing gamut (while preserving luminance) until it parallels the luminance direction, then applies a uniform scaling and translation that brings the darkest point of the printing gamut to $(0, 0, -1)$ and the lightest to $(0, 0, 1)$. Our linear transform is similar to that of Stone et al. [Stone et al. 88, Section 5.2], but we have replaced their rotation with a shear in order to preserve luminance relationships throughout the gamut mapping process.

The second stage of the transformation converts the triple (u, v, y) to curvilinear coordinates (r, h, ϕ) , where h represents hue, and r and ϕ indirectly encode luminance and saturation. These new coordinates are found by inverting the following equations:

$$\begin{aligned} u &= r \cos h \cos \phi \\ v &= r \sin h \cos \phi \\ y &= (1 - \kappa^2 + \kappa r) \sin \phi \end{aligned}$$

In this coordinate system, lines of constant h and ϕ trace out the normals to the ellipsoid $(u^2 + v^2)/\kappa^2 + y^2 = 1$; these are the lines along which we compress out-of-gamut colors. Note that when $\kappa = 1$, the transformation above gives the standard conversion between spherical and Cartesian coordinates, which is easily inverted. Likewise, when $\kappa = 0$ the equations are only a slight modification of standard cylindrical coordinates, and are also easily inverted.

However, for intermediate values of κ , we lack an analytic solution (except in some special cases) and therefore resort to Newton's method to solve a nonlinear equation for ϕ .

To be more specific, here are the steps we take to compute (r, h, ϕ) coordinates from (u, v, y) coordinates when κ is neither 0 nor 1.

- From the ratio of the first two equations above, we know that $h = \tan^{-1} v/u$.
- If $y = 0$, then $\phi = 0$ and $r = \sqrt{u^2 + v^2}$.
- If $u = v = 0$ and $|y| \geq 1 - \kappa^2$, then $\phi = \text{sign}(y) \pi/2$ and $r = (|y| - 1 + \kappa^2)/\kappa$.
- If $u = v = 0$ and $|y| < 1 - \kappa^2$, then $\phi = \sin^{-1} y/(1 - \kappa^2)$ and $r = 0$.
- Otherwise, we attempt to solve the equation $y = (1 - \kappa^2 + \kappa\sqrt{u^2 + v^2}/\cos \phi) \sin \phi$ for ϕ using Newton's method, beginning with an initial guess of $\phi_0 = \tan^{-1} 1/\kappa$.
- If the iterations fail to converge, we use the same initial guess in Newton's method to solve an alternate form of the equation: $\kappa\sqrt{u^2 + v^2} = (y/\sin \phi - (1 - \kappa^2)) \cos \phi$.

4.3.3 Steps in n -tone mapping

The first step in our gamut mapping technique for three or more inks is to apply a global mapping to the luminance values of the source image's colors, just as we do for one or two inks. For most values of κ we can skip this first step because subsequent steps will also adjust luminance; it is only when $\kappa = 0$ that we must compress luminance in order for a cylindrical mapping to get all colors into the printing gamut.

The second step is to divide the set of directions that are parameterized by h and ϕ into a two-dimensional array of bins, as indicated by the grid lines in Figure 4-6. The number of divisions in each direction determines the storage, efficiency, and accuracy of subsequent mapping steps; numbers near 20 are adequate for quick previews, while numbers near 100 are more suitable for high-quality output.

Next, we determine the maximum extent in the r direction of the printing gamut within each (h, ϕ) bin. For each bin, we construct a ray centered within that bin, and intersect it with each of the bilinear surfaces that bound the gamut. (The formulas used to compute these intersections are the same as the ones used for projection and separation in the duotone case; they are derived in Section 6.2.) We associate with the bin the largest r value encountered in these intersection tests, which we call \hat{r}_{\max} .

We also store with each bin a quantity r_{\max} , defined as the largest r value of all image colors lying in that bin. If the printing gamut exceeds the image gamut for some bin, we set r_{\max} equal to \hat{r}_{\max} , so that gamut mapping will not spread similar colors apart.

The final step is to apply a mapping to the r value of each image color, where the mapping varies from one bin to the next. Given an image color, we determine the four bins closest to the (h, ϕ) coordinates of that color, and apply bilinear interpolation to the corresponding values of r_{\max} and \hat{r}_{\max} .² We construct a function that maps zero to zero and the interpolated value of r_{\max} to the interpolated value of \hat{r}_{\max} . The intermediate values of r can be computed using the clamped, linear, or cubic mappings presented earlier (with $r_{\min} = \hat{r}_{\min} = 0$).

Figure 4-7 illustrates the geometric effect in color space of both gamut mapping steps (luminance compression and radial compression) on a set of image colors. Once all the image colors have been mapped, we can convert them from (r, h, ϕ) coordinates back to XYZ coordinates, and they should all lie within the printing gamut. By replacing each color in the original image with its corresponding mapped color, we obtain a preview image predicting the printed appearance of the original.

² We could alternatively use the simplex method as described in Section 4.2 to construct a bicubic spline function $r_{\max}(h, \phi)$ that tightly bounds the r values of the image colors, but bilinear interpolation appears smooth enough for practical purposes.

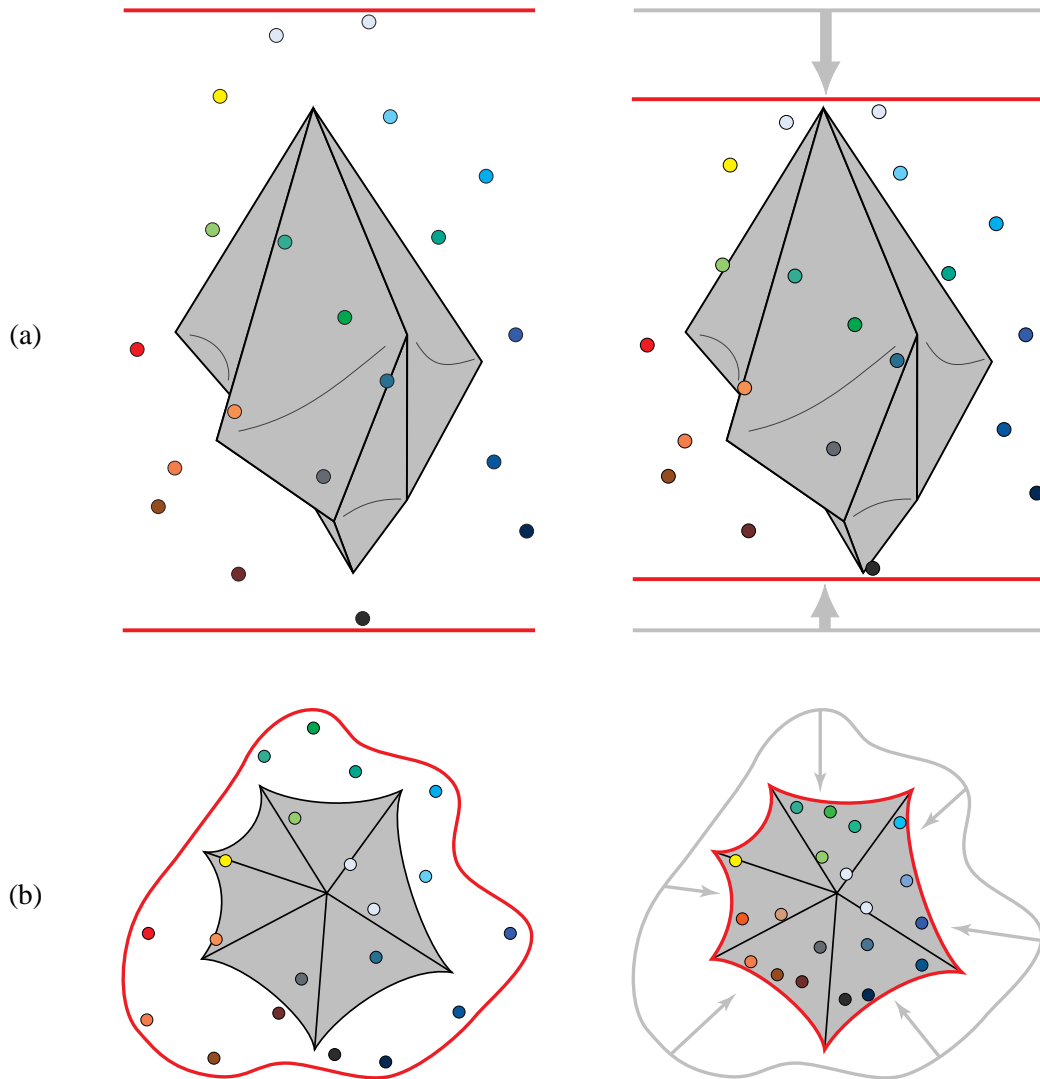


Figure 4-7 Steps of gamut mapping for three or more inks: (a) before and after luminance mapping, (b) before and after radial compression.

✦ **Selecting inks** ✦

Throughout Chapters 3 and 4 we treated the paper and inks as though they were known. Our goal, however, is to find the optimal combination of paper and inks for a given image. This task is difficult because there may be a huge number of possible choices, most of which will result in poor reproductions. Often it is not obvious whether or not a choice of paper and inks will reproduce an image well; therefore, we are not yet willing to rely on heuristic rules for accepting or rejecting combinations. Instead, we pose the problem of selecting paper and inks as a combinatorial optimization problem and apply general-purpose algorithms to solve it. We describe below our objective function and a number of alternative optimization algorithms.

5.1 Ink-selection objective function

The role of our objective function for ink selection is to quantify how well a given choice of paper and inks can reproduce a given image. The first step toward this end is to apply the gamut mapping algorithm discussed in Chapter 4; each image color is modified by the mapping, and we obtain a new image predicting the printed appearance of the original. Then we compare this preview image pixel-by-pixel to the original image. It is important to accurately measure the difference a human would perceive between the preview colors and the original colors. Therefore, we convert all the colors from XYZ coordinates to $L^*a^*b^*$ coordinates, where we can use the L^2 norm to measure perceptual color differences. The value we assign to our objective function is the average over all pixels of the L^2 distance between the preview and original image.

The efficiency of the objective function is an important concern because it gets evaluated often. Fortunately, we don't need to apply our gamut mapping algorithm to each of the tens or hundreds of thousands of distinct colors contained in a typical high-resolution scan. Instead, we use Heckbert's median-cut algorithm [Heckbert 82] to quantize the image to about 2,000 distinct colors while maintaining its general appearance. Then we need only apply the gamut mapping algorithm to these quantized colors, and weight each color's

$L^*a^*b^*$ distance according to the number of pixels represented by that color in the original image.

There are other methods of reducing the number of colors in an image: statistical clustering, for example, or octree quantization [Gervautz & Purgathofer 88]. Some of these techniques may be better than the median-cut algorithm at differentiating important representative colors from unimportant outlying colors, and may therefore provide a more accurate means of evaluating the quality of gamut-mapped images. Another way to improve the accuracy of the ink selection process is to first use a reduced set of colors to get an approximate solution, then use the complete set of original colors to refine that solution. These possibilities are left for future research, as is the difficult (but potentially very rewarding) task of finding heuristic rules to quickly accept or reject a given combination of paper and inks.

5.2 Ink-selection optimization algorithm

There are several flavors of optimizers suited to combinatorial problems like ours: brute-force search, dynamic programming, simulated annealing, and genetic algorithms, to name a few. Because of the time it takes to evaluate every possible combination of inks, we can rely on brute-force search only for small problems, such as choosing two or three inks from the 14 fundamental PANTONE inks used as ingredients in mixing other colors. For larger problems, brute force becomes prohibitive: there are nearly 500,000 ways to choose two inks from the 1,000 PANTONE inks, 10^{10} ways to choose four inks from this set, and 10^{15} ways to choose six inks.

In order to develop an ink-selection algorithm that scales well to large problems, we turn to stochastic optimization techniques. These techniques, which include random descent, simulated annealing, and genetic algorithms, rely on random changes to perturb intermediate solutions away from local minima in the objective function.

Stochastic optimization methods

Random descent starts with an initial combination of paper and inks, and at each iteration, generates a random variation on the current combination. The current combination is replaced whenever the newly generated combination receives a better rating from the objective function. We can apply randomness to the generation of new combinations in a couple of ways: we could pick the next paper and inks to try completely haphazardly, or we could choose randomly from among a few papers and inks closest to the current ones. There are

benefits to both approaches, since haphazard choices can shift our attention from local minima to global minima, while small changes are necessary to refine good solutions into incrementally better ones.

This logic suggests that to find a global minimum, we might try large random changes when we begin optimizing and gradually shift to smaller changes later on. Simulated annealing follows a similar strategy, but instead of varying the size of random changes, it varies the criteria by which changes are accepted or rejected. At first, annealing allows changes that increase the objective function by a significant amount. Over the course of the optimization, random changes that increase the objective function become less and less tolerable, until finally only decreases are allowed. This strategy permits the solution to escape from local minima early in the process, later becoming trapped (we hope) at a global minimum. By analogy with the physical simulation from which the method is derived, annealing optimizers require a cooling schedule that prescribes how much the objective function may increase as a function of time. Otten and van Ginneken cover in depth the cooling schedule and other details of the simulated annealing algorithm [Otten & van Ginneken 89].

Genetic algorithms provide a different approach to stochastic optimization. Instead of making random changes to one candidate solution, a genetic algorithm maintains a population of candidate solutions. At each iteration, the algorithm chooses one of the population members to undergo mutation or two of the members to combine via crossover. Mutation is essentially the same operation as is used in random descent and simulated annealing: in our situation, the current paper and inks are replaced by a random choice of similarly colored paper and inks. Crossover, on the other hand, takes two combinations and forms a new one by randomly choosing between them to determine the paper and each of the inks. Regardless of whether a new combination is generated by mutation or crossover, we decide whether or not to include it in the population based on its objective function value. If the new combination outperforms the worst member of the population, then the worst one is replaced by the new one. In this manner, the entire population improves as better solutions are found by altering or combining existing ones. We repeat the process for a fixed number of iterations, or until some number of iterations fails to yield a decrease in the objective function. In contrast to random descent and simulated annealing, which produce just one solution, the output of the genetic algorithm is a list of possible choices of papers and inks, ranked according to their objective function values.

Our implementation

We originally chose to use a simulated annealing algorithm to choose pairs of inks for duotone reproduction because of its ability to find global minima and relatively good local minima [Power et al. 96]. We later decided to use a genetic algorithm, mainly because it maintains a population of candidate solutions, each of which can be presented to the user as a possibility when the optimizer is finished. In fact, we made our simulated annealer slightly closer to a genetic algorithm by using multiple initial conditions to obtain a variety of solutions; likewise, we made our genetic algorithm slightly closer to simulated annealing by assigning a time-varying probability distribution to each of the genetic operators. We could easily call both of our approaches hybrids of the conventional annealing and genetic algorithms.

Following suggestions made by Davis [Davis 91], we include in our implementation a number of variations on the “standard” genetic algorithm. We keep the population of candidate combinations sorted according to their respective objective function values and disallow repeated combinations in the population. At each iteration, we choose a genetic operator according to a time-varying probability distribution, and we choose its operands randomly from the current population. The operators for our application include global mutation, local mutation, and crossover. Global mutation operates on one combination, changing the paper and inks completely randomly; this operator is more likely to be chosen in early iterations. Local mutation also operates on one combination, but changes the paper and inks only to nearby colors; this operator is more likely in later iterations. Crossover takes two combinations, from which it chooses each of the papers and inks with 50% probability. The likelihood of selecting the crossover operation starts out high, and gradually declines to zero, so as to prevent one solution from dominating the population.

There are a few details of our optimization that remain to be discussed, including the initial conditions and the exact nature of the local mutation and crossover operators. Each member of the initial population is chosen completely at random from the papers and ink over which we are optimizing. We may be able to find solutions faster by generating the initial combinations heuristically (say by selecting ink colors similar to the predominant saturated colors in the original image), but we have not explored that possibility.

The local mutation operator generates a new combination from an existing one by replacing each ink or paper with one of a similar color. In order to make this operation efficient, we determine before optimization the set of inks that neighbor (in $L^*a^*b^*$ color space) each of the available inks, and likewise for the papers. The ideal data structure for this pre-

processing step is a three-dimensional Voronoi diagram, whose dual graph would provide us with a list of neighbors for each ink. However, because we process each set of inks and papers only once, we can afford to use a less elegant brute-force algorithm to search out the five or ten nearest neighbors of each ink or paper. Once we ascertain the closest neighbors, we symmetrize the relationship so that if a is a neighbor of b , then b is also a neighbor of a .

Given two combinations of paper and inks, the crossover operator selects the first ink to be printed by choosing the first ink of either one combination or the other with equal probability. Likewise, the second ink to be printed is chosen to be the second ink of one of the combinations, and so on. This approach can produce poor results from good inputs. For example, given a pair of blue and light orange inks and a pair of dark orange and cyan inks, crossover generates the less-than-helpful pair of light orange and dark orange inks one quarter of the time. We could eliminate this possibility by applying bipartite matching [Papadimitriou & Steiglitz 82, pages 221–226, 248–255] to the inks in the two combinations before performing crossover. We would choose one ink from the two orange colors because of their similarity, and the other from cyan and blue because of their proximity in color space. Though this particular improvement may turn out to be worthwhile, introducing complexity into the genetic algorithm's operators is less appealing than encoding rules for quick rejection of bad combinations into the objective function itself.

We compare the performance of our genetic algorithm to brute force search and random descent in Figure 5-1 for several images. In each case, we limit the optimizers to choose inks from the set of sixteen inks whose characteristics we determine in Chapter 7. The first graph shows that brute force works as well as the stochastic methods for choosing just three inks—the optimizer can explore all 3,360 permutations of three inks chosen from sixteen, and it happens to find good solutions early on. The remaining convergence graphs indicate that the genetic algorithm performs slightly better than random descent, and both these stochastic algorithms do significantly better than brute force as the number of inks increases.

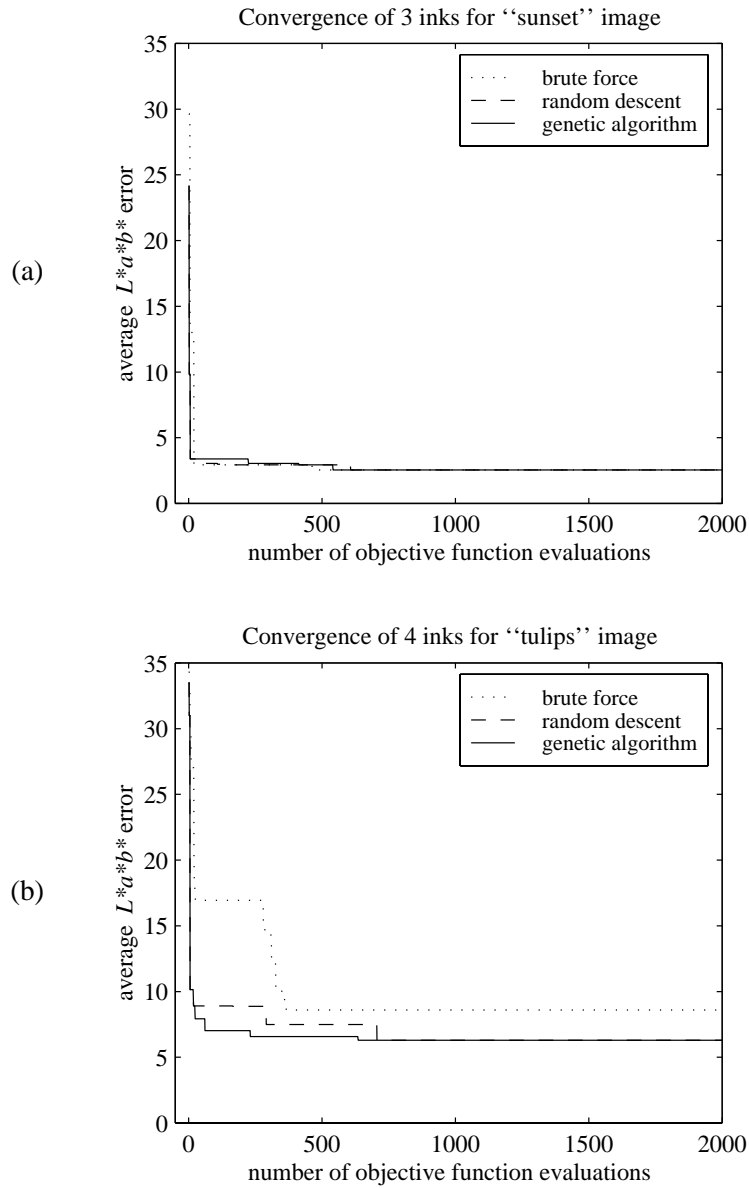


Figure 5-1 Comparison of combinatorial optimization algorithms for selecting inks: (a) 3 inks selected for "sunset" image, and (b) 4 inks selected for "tulips" image. (*continued*)

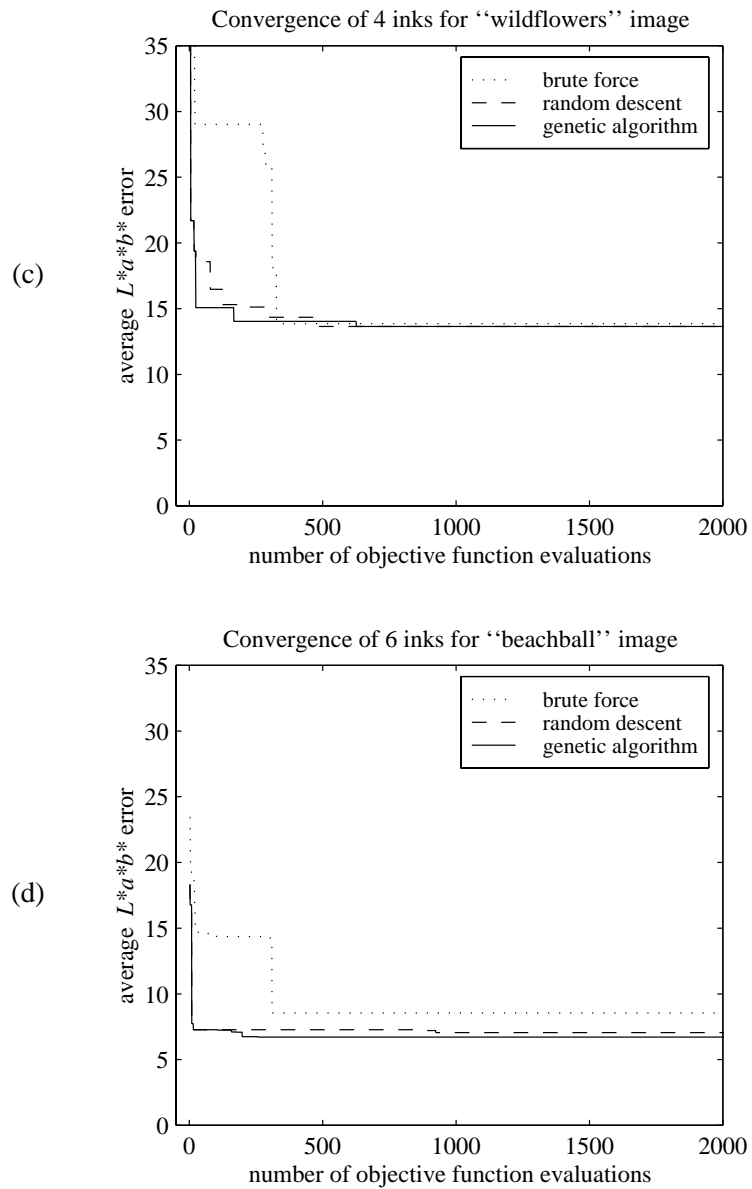


Figure 5-1 (continued) Comparison of combinatorial optimization algorithms for selecting inks: (c) 4 inks selected for “wildflowers” image, and (d) 6 inks selected for “beach ball” image.

✦ Computing separations ✦

Once we have applied the ink selection techniques Chapter 5 to choose inks and the gamut mapping techniques of Chapter 4 to generate a preview image, our only remaining task is to produce separations for each of the inks. More precisely, for each color generated by the gamut mapping algorithm, we need to find the amount of each ink required to reproduce that color. The mathematical model of printing gamuts that we developed in Chapter 3 gives tristimulus colors as a function of the ink amounts α , whereas now we want to find the ink amounts as a function of color. In general, determining this inverse function is no simple matter, mainly because each tristimulus component of our model is a nonlinear function of the α values. The exceptions are the one-ink and two-ink printing gamuts, which are sufficiently simple that this inverse function can be derived analytically. We treat these special cases in the first two sections below.

For three inks, it is possible to invert the Neugebauer model so long as the printing primaries are all distinguishable from one another [Mahy & Delabastita 96]. However, the inverse function is underdetermined when the three inks are degenerate or when there are more than three inks; there may be many α values that yield the same color. There is another difficulty that arises in practice: some of the colors for which we wish to compute separations may be slightly out of gamut because of imprecisions in the gamut mapping stage, yet we still need to find α values between zero and one that reproduce similar colors.

While a number of authors treat the printing gamut of multiple inks as a convex union of tetrahedra [Iwata & Marcu 94, Marcu & Abe 94a, Ostromoukhov 93, Takaghi et al. 94], we are loath to take that approach when we know that many gamuts are actually concave. Instead, for gamuts of three or more inks, we rely on a continuous optimization technique to find the separations that most closely reproduce a desired color while meeting the physical constraints of the printing process. The details of our objective function and optimization algorithm are given in the final section of this chapter.

6.1 Separations for one ink

The gamut of colors that can be printed with one ink is given by equation (3.1), which we repeat here for convenience:

$$\mathbf{c}(\alpha) = (1 - t_{p1}\alpha_1) \mathbf{g}_p + t_{p1}\alpha_1 \mathbf{g}_{p1} \quad (6.1)$$

Clearly, if we are given a color \mathbf{c} within this gamut, we can rearrange terms to find α_1 using any one of the three components of the tristimulus colors involved. More precisely, α_1 must satisfy all three components of the following linear system:

$$t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \alpha_1 = \mathbf{c} - \mathbf{g}_p$$

In Section 4.1, we pointed out that we could simultaneously project an out-of-gamut color $\hat{\mathbf{c}}$ onto the gamut and find the ink amount α_1 . Assuming that the gamut mapping process has already adjusted the luminance of *targetcol* to lie within the range of printable values, we want the projection to preserve the luminance (the Y component) of $\hat{\mathbf{c}}$. We therefore write the in-gamut color as $\mathbf{c} = \hat{\mathbf{c}} + \mu\mathbf{X} + \nu\mathbf{Z}$, where μ and ν are unknown numbers. Substituting this expression in the equation above gives

$$t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \alpha_1 = \hat{\mathbf{c}} + \mu\mathbf{X} + \nu\mathbf{Z} - \mathbf{g}_p$$

Observe that when we take the dot product of both sides of this equation with \mathbf{Y} , we eliminate the unknown variables μ and ν because \mathbf{X} , \mathbf{Y} , and \mathbf{Z} are mutually orthogonal. We therefore arrive at a single linear equation for the unknown ink amount α_1 :

$$t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \cdot \mathbf{Y} \alpha_1 = (\hat{\mathbf{c}} - \mathbf{g}_p) \cdot \mathbf{Y}$$

The solution is of course

$$\alpha_1 = \frac{(\hat{\mathbf{c}} - \mathbf{g}_p) \cdot \mathbf{Y}}{t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \cdot \mathbf{Y}}$$

The corresponding projected color is given by substituting this value in equation (6.1).

6.2 Separations for two inks

Computing separations for duotone printing involves slightly more work than for just one ink, but we can still use an analytic solution. In fact, the technique we present here simultaneously solves for ink amounts and accomplishes the last step of our gamut mapping algorithm for two inks, the projection of a color onto a bilinear gamut described in Section 4.2.

Suppose we have a color $\hat{\mathbf{c}}$ that we want to project in the direction \mathbf{P} onto a duotone gamut defined by the printing primaries \mathbf{g}_p , \mathbf{g}_{p1} , \mathbf{g}_{p2} , and \mathbf{g}_{p12} . The solution is a color \mathbf{c} that lies on the line passing through $\hat{\mathbf{c}}$ in the direction \mathbf{P} , so we know $\mathbf{c} = \hat{\mathbf{c}} + \eta\mathbf{P}$ for some real number η . The solution must also lie on the bilinear surface defining the duotone gamut:

$$\begin{aligned} \mathbf{c} = \hat{\mathbf{c}} + \eta\mathbf{P} &= (1 - t_{p1}\alpha_1) (1 - t_{p2}\alpha_2) \mathbf{g}_p \\ &+ (t_{p1}\alpha_1) (1 - t_{p12}\alpha_2) \mathbf{g}_{p1} \\ &+ (1 - t_{p1}\alpha_1) (t_{p2}\alpha_2) \mathbf{g}_{p2} \\ &+ (t_{p1}\alpha_1) (t_{p12}\alpha_2) \mathbf{g}_{p12} \end{aligned} \quad (6.2)$$

In what follows, we will solve for the unknown ink amounts α_1 and α_2 , rather than the projection point \mathbf{c} .

First, we rewrite equation (6.2) by grouping the terms differently:

$$\begin{aligned} 0 &= (\mathbf{g}_p - \hat{\mathbf{c}} - \eta\mathbf{P}) \\ &+ t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \alpha_1 \\ &+ t_{p2}(\mathbf{g}_{p2} - \mathbf{g}_p) \alpha_2 \\ &+ t_{p1}[t_{p2}(\mathbf{g}_p - \mathbf{g}_{p2}) + t_{p12}(\mathbf{g}_{p12} - \mathbf{g}_{p1})] \alpha_1\alpha_2 \end{aligned}$$

Notice that if we take the dot product of both sides of this equation with either the \mathbf{S} direction or the \mathbf{Y} direction, we eliminate η because \mathbf{P} , \mathbf{S} , and \mathbf{Y} are defined to be mutually orthogonal. We can write the two equations that result from these dot products as follows:

$$\mu_1 + \mu_2\alpha_1 + \mu_3\alpha_2 + \mu_4\alpha_1\alpha_2 = 0 \quad (6.3)$$

$$\nu_1 + \nu_2\alpha_1 + \nu_3\alpha_2 + \nu_4\alpha_1\alpha_2 = 0 \quad (6.4)$$

The constants μ_1, \dots, μ_4 and ν_1, \dots, ν_4 are given by

$$\begin{aligned} \mu_1 &= (\mathbf{g}_p - \hat{\mathbf{c}}) \cdot \mathbf{S} & \nu_1 &= (\mathbf{g}_p - \hat{\mathbf{c}}) \cdot \mathbf{Y} \\ \mu_2 &= t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \cdot \mathbf{S} & \nu_2 &= t_{p1}(\mathbf{g}_{p1} - \mathbf{g}_p) \cdot \mathbf{Y} \\ \mu_3 &= t_{p2}(\mathbf{g}_{p2} - \mathbf{g}_p) \cdot \mathbf{S} & \nu_3 &= t_{p2}(\mathbf{g}_{p2} - \mathbf{g}_p) \cdot \mathbf{Y} \\ \mu_4 &= t_{p1}[t_{p12}(\mathbf{g}_{p12} - \mathbf{g}_{p1}) \cdot \mathbf{S} - \mu_3] & \nu_4 &= t_{p1}[t_{p12}(\mathbf{g}_{p12} - \mathbf{g}_{p1}) \cdot \mathbf{Y} - \nu_3] \end{aligned}$$

Next, we can solve equation (6.3) for α_2 in terms of α_1 to get

$$\alpha_2 = -\frac{\mu_1 + \mu_2\alpha_1}{\mu_3 + \mu_4\alpha_1} \quad (6.5)$$

When we substitute this expression for α_2 into equation (6.4) and simplify, we obtain a quadratic equation for α_1 alone:

$$\xi_1 \alpha_1^2 + \xi_2 \alpha_1 + \xi_3 = 0 \quad \text{where} \quad \begin{cases} \xi_1 = \mu_4 \nu_2 - \mu_2 \nu_4 \\ \xi_2 = \mu_4 \nu_1 - \mu_1 \nu_4 + \mu_3 \nu_2 - \mu_2 \nu_3 \\ \xi_3 = \mu_3 \nu_1 - \mu_1 \nu_3 \end{cases}$$

Therefore, the solution for α_1 is given by

$$\alpha_1 = \begin{cases} \frac{-\xi_2 \pm \sqrt{\xi_2^2 - 4\xi_1\xi_3}}{2\xi_1} & \text{if } \xi_1 \neq 0 \\ -\frac{\xi_3}{\xi_2} & \text{if } \xi_1 = 0 \end{cases}$$

When $\xi_1 \neq 0$, there are two possible projections of $\hat{\mathbf{c}}$ onto the bilinear surface; we choose the solution for α_1 that results in a value between zero and one. Once we obtain α_1 , we can compute α_2 from equation (6.5) and the projection point \mathbf{c} from equation (6.2).

Note that the method described above essentially computes the intersection of a ray with a bounded portion of a bilinear surface, which is exactly the operation that Section 4.3 called for to determine the extent of a multicolor printing gamut in a given direction. Thus, the algebraic derivation given here turns out to be useful to us in more than one setting.

6.3 Separations for three or more inks

The analytic approach we used to invert the one- and two-ink gamut models can be extended to deal with three-ink gamuts as well. Mahy and Delabastita show that algebraic manipulation can transform the three trilinear equations of the Neugebauer model into a single sixth-degree polynomial in one unknown, whose roots can then be determined robustly using Laguerre's method [Mahy & Delabastita 96]. As many as six triplets of ink amounts may satisfy the equations, though only the real-valued solutions falling between 0 and 1 are of interest.

Analytic approaches to computing separations become less tractable and less helpful for more than three inks. Because the function $\mathbf{c}(\boldsymbol{\alpha})$ maps from a multidimensional parameter space to a three-dimensional color space, it must be many-to-one. For any given color there is a continuous subspace of ink amounts capable of producing that color, from which we must choose one solution. Instead of pursuing a cumbersome analytic solution for more than three inks, we rely on continuous optimization to find the ink amounts that most closely

reproduce a desired color. When dealing with three inks, we also use optimization rather than Mahy and Delabastita’s solution, simply to avoid the cost of discovering degeneracies, manipulating polynomials, finding roots, and checking solutions. The details of our technique are described below.

6.3.1 Separation objective function

Suppose we want to determine ink amounts that will reproduce a color $\hat{\mathbf{c}}$ that was produced by applying gamut mapping to one of the original image’s colors. The goal of the current optimization is to find the ink amounts $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ for which our gamut model yields the color $\mathbf{c}(\boldsymbol{\alpha})$ closest to $\hat{\mathbf{c}}$. The physical limitations of printing require that we constrain each of the α_i variables to lie between 0 and 1. We further require the sum of the ink amounts to be less than the constant α_{limit} , called the “ink limit.” This constraint ensures that the total amount of ink does not exceed the point at which ink no longer adheres to the page. The ink limit depends upon the press and the paper, but typically lies in the range $3 \leq \alpha_{\text{limit}} \leq 4$.

In order to make use of widely available algorithms for unconstrained optimization, we reformulate each of the constraints as a penalty term in our objective function. We also include a term whose purpose is to draw the ink amounts to a unique solution when there would otherwise be more than one optimal answer. The objective function is therefore comprised of four terms:

$$f(\boldsymbol{\alpha}; \hat{\mathbf{c}}, \boldsymbol{\alpha}_{\text{ref}}) = f_{\text{distance}}(\boldsymbol{\alpha}; \hat{\mathbf{c}}) + f_{\text{bounds}}(\boldsymbol{\alpha}) + f_{\text{limit}}(\boldsymbol{\alpha}) + f_{\text{ref}}(\boldsymbol{\alpha}; \boldsymbol{\alpha}_{\text{ref}})$$

The four terms are defined as follows:

$$\begin{aligned} f_{\text{distance}}(\boldsymbol{\alpha}; \hat{\mathbf{c}}) &= w_1 \|\mathbf{c}(\boldsymbol{\alpha}) - \hat{\mathbf{c}}\|^2 \\ f_{\text{bounds}}(\boldsymbol{\alpha}) &= w_2 \sum_i \max(0, -\alpha_i, \alpha_i - 1)^2 \\ f_{\text{limit}}(\boldsymbol{\alpha}) &= w_3 \max(0, \sum_i \alpha_i - \alpha_{\text{limit}})^2 \\ f_{\text{ref}}(\boldsymbol{\alpha}; \boldsymbol{\alpha}_{\text{ref}}) &= w_4 \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\text{ref}}\|^2 \end{aligned}$$

The first term of the objective function is just the square of the distance in XYZ color space between the desired color $\hat{\mathbf{c}}$ and the color our gamut model predicts from the $\boldsymbol{\alpha}$ values. For an in-gamut color, the optimizer should find a solution where this term is zero. If because of numerical inaccuracies in our gamut mapping we are trying to find ink amounts for a color that is slightly outside the gamut, minimizing the first term above is akin to projecting onto the nearest point of the gamut, as was done by Stone et al. before computing

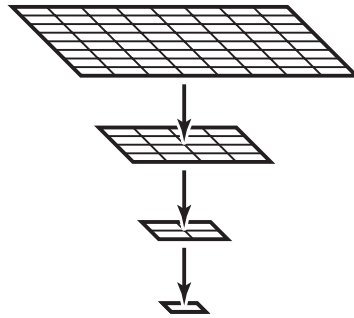


Figure 6-1 Image pyramid used in multiresolution color separation.

separations [Stone et al. 88]. We arbitrarily choose the weight $w_1 = 0.005$, and set the remaining weights by trial and error based on the magnitudes of the terms.

The second term bounds each of the ink amounts between 0 and 1, thereby moving out-of-gamut colors to in-gamut separations. The third term keeps the total amount of ink from exceeding the ink limit. We typically set $w_2 = w_3 = 1000$ so that ink amounts violating these constraints are strongly penalized.

The final term of the objective function allows us to achieve a unique solution when there are multiple ways to produce the same color. We do so by finding the solution that is closest to a given set of reference values α_{ref} . We use a relatively small weight for this term ($w_4 = 0.025$) so that it does not prevent the color $c(\alpha)$ from matching \hat{c} . We can set each component of α_{ref} to 0 or 1 to minimize or maximize the amount of ink used, or choose 0.5 for an intermediate solution.

Unfortunately, we find that these simple choices of reference values often lead to separations containing artificial discontinuities. Because the ink amounts used for adjacent pixels are computed independently, slightly different colors may result in very different separations. While in theory these separations will produce similar colors when printed, in reality even the slightest misregistration reveals the discontinuities. Even the best printing presses are subject to some deviation from perfect registration, so we cannot afford to ignore these artifacts our separation approach can produce. To avoid such artifacts, we try to compute separations that are just as smooth as the input image. One possible solution is to set α_{ref} to the ink amounts of the most similar color among the four adjacent pixels that have already been separated. This approach eliminates many artifacts, but because of the asymmetry inherent in processing pixels from left to right and top to bottom, it may still produce discontinuities in some directions.

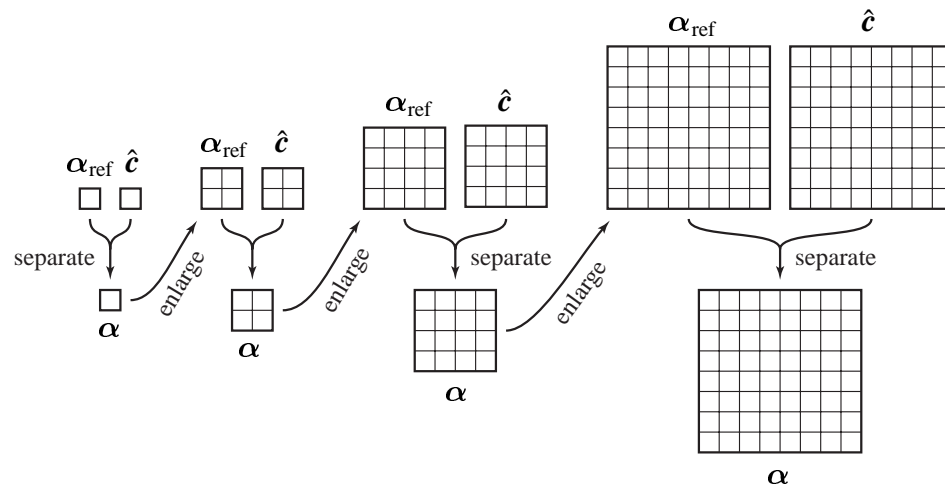


Figure 6-2 Steps of the multiresolution separation algorithm. At each resolution, we compute the α values that come closest to α_{ref} while reproducing the color \hat{c} . The resulting separation images of α values are then enlarged to form the reference values α_{ref} at the next resolution level.

We can generate much better separations using a multiresolution algorithm based on image pyramids. The central idea is to compute for each pixel the ink amounts that produce the right color and are as close as possible to the ink amounts of the entire surrounding neighborhood. Of course, the ink amounts that best reproduce a pixel's neighborhood depend in turn on a larger neighborhood, and thus we rely on a recursively defined image pyramid. First, we construct a pyramid of reduced images from the gamut-mapped source image by repeatedly applying a low-pass filter followed by downsampling. Each image has half the resolution of the previous one, and therefore one quarter of the number of pixels, as illustrated in Figure 6-1. We use a separable low-pass filter with coefficients $\frac{1}{16}(1, 4, 6, 4, 1)$ to construct our image pyramid. Further details on image pyramids and their uses in computer graphics can be found in the work by Adelson, Burt, and colleagues [Adelson et al. 84, Burt 81, Burt & Adelson 83].

In the next step of the multiresolution algorithm, we compute separations for the lowest-resolution image (with each component of α_{ref} set to 0.5). Then for each higher resolution version of the image, we calculate new separations, using as the reference values α_{ref} an enlarged version of the current separations, as shown in Figure 6-2. The enlargement operation consists of upsampling an image, followed by filtering with the same low-pass filter coefficients used for reduction [Burt & Adelson 83]. The end result is a set of separations at the highest resolution that maintain the smoothness of the original color image. Smooth-

ness comes at the cost of computing separations for the entire image pyramid, but this is only 4/3 the work of computing separations just at the highest level.

6.3.2 Separation optimization algorithm

There are a plethora of continuous optimization algorithms we could apply to the separation problem. We can choose among them by considering the amount of information they require and their rates of convergence. Because we can efficiently compute the first partial derivatives of all the terms in our objective function, we can use optimization techniques that achieve quadratic convergence rates. These include the conjugate gradient method and “quasi-Newton” methods, among others. We can also compute the second partial derivatives of the objective function, allowing us to use Newton’s method or globally convergent adaptations of Newton’s method like the “dogleg” method.

Figure 6-3 shows a comparison of the performance of three numerical optimization algorithms: the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method, the conjugate gradient method, and the dogleg adaptation of Newton’s method. The graphs show statistics for each of thirteen typical images. The dogleg method is the slowest of the three algorithms, taking many more iterations to converge and performing a costly evaluation of the objective’s second derivatives at each iteration. The other two methods converge much more quickly, with the BFGS technique taking less time than the conjugate gradient method in almost all cases. Press et al. give detailed descriptions of the latter two algorithms [Press et al. 92, pages 420–430].

Each of the numerical optimization methods mentioned above requires an initial guess at the α values for each pixel. One possibility is to set each α_i to a constant (one half, for example) before computing separations for each pixel. This turns out to be quite naive, though, since the number of iterations required to converge on a solution depends heavily on how close the initial guess is to a minimum. Newton’s method will not even converge if the starting values are not close enough to a minimum. We can reduce the number of iterations significantly by using the α values of the previous pixel as an initial guess for the current pixel, since the colors of adjacent pixels are usually similar. Another alternative is to use the values of α_{ref} computed by the multiresolution separation algorithm as initial values of α . This choice works well because the separations for any particular pixel are likely to be similar to the separations of the average color of the surrounding neighborhood.

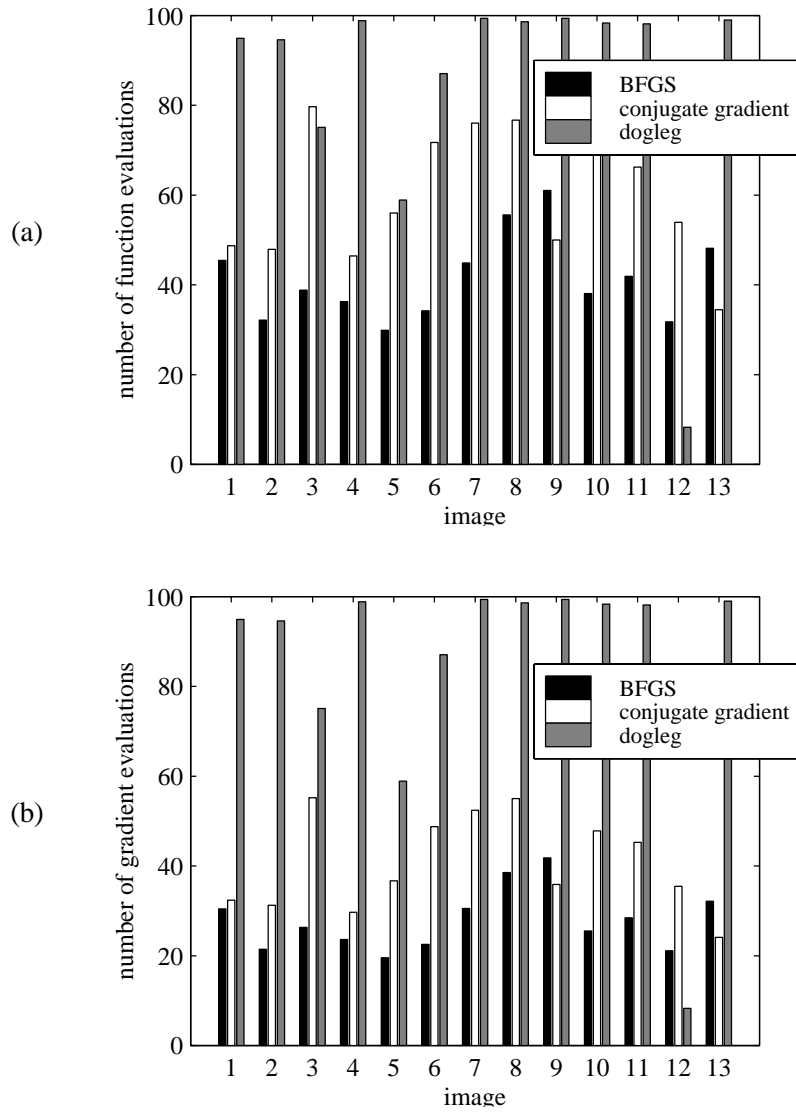


Figure 6-3 Comparison of continuous optimization algorithms for computing separations: (a) average number of objective function evaluations per pixel, and (b) average number of gradient evaluations per pixel. *(continued)*

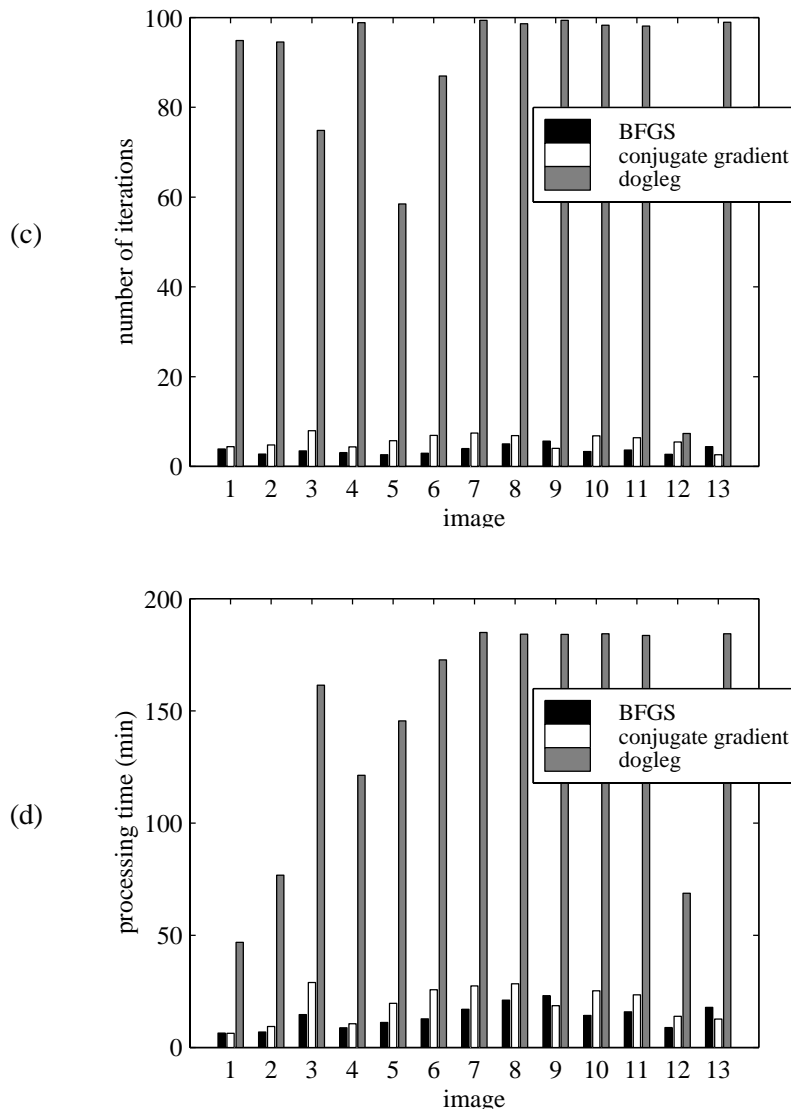


Figure 6-3 (continued) Comparison of continuous optimization algorithms for computing separations: (c) average number of iterations per pixel, and (d) total time to compute separations.

✦ Implementation and experiments ✦

In this chapter, we present practical aspects of our image reproduction system as well as details of our printing experiments. We begin by describing how our application incorporates the models and algorithms of the previous chapters. A good deal of this description is devoted to the user interface, for though interactivity is not strictly required to generate custom-ink color reproductions, we find that the experience and subjective judgement of a human user often improve the results. The chapter continues with a discussion of how we acquire the data required as input to our system. In addition to an input image, we need to provide the parameters of our gamut model, including the trapping, dot gain, and spectral characteristics of every paper and ink we might use. Most of these characteristics cannot be measured directly, but we present a data-fitting scheme that allows us to derive them from experimental data. The subsequent section presents the end result of our complete system: color images printed with a variety of custom inks. Finally, we conclude the chapter with a discussion of the time required by each stage of our system.

7.1 Implementation features

We developed a system incorporating the gamut model of Chapter 3 and the gamut mapping, ink selection, and separation algorithms of Chapters 4, 5, and 6. The system, which is written in C++, has two modes of operation: batch mode and interactive mode. In batch mode, the system takes the following steps:

- 1. Read the input.** The system reads an image and a set of constraints on the paper and inks to be used. The constraints can specify which paper to use, or they can delimit a subset of the available papers from which the system may choose. Likewise, each ink may be fixed by some prior decision, or allowed to vary within a particular subset of all available inks. The input also includes the parameter settings needed to control the remaining steps of the process.

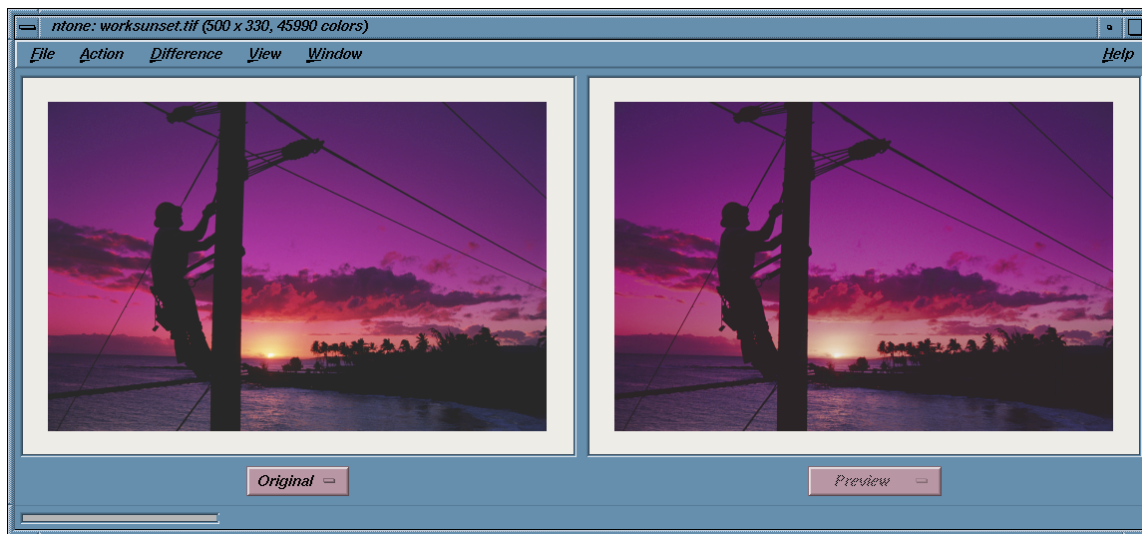


Figure 7-1 The interactive system's image viewer.

- 2. Find the optimal paper and inks.** The system applies the combinatorial optimization techniques of Chapter 5 to choose the best paper and inks satisfying the given constraints. We can skip this step if the constraints precisely specify all the printing materials.
- 3. Perform gamut mapping.** The system maps each of the original image's colors into the printing gamut using the methods of Chapter 4. The result is an image providing a preview of the printed appearance of the original.
- 4. Compute color separations.** The separation algorithm of the previous chapter generates ink amounts corresponding to each pixel of the gamut-mapped preview image.
- 5. Combine the separations.** As a check on the effectiveness of the gamut mapping and the accuracy of the separation algorithm, the system uses the gamut model to combine the ink amounts for each pixel into a "proof" image portraying the precise colors the separations will produce when printed. One of the options available in this step is to translate the separations a few pixels in different directions before combining them, thereby producing a proof that simulates misalignment of the printing plates. This artificial misregistration is very helpful in detecting the artifacts of the poorer separation methods that compelled us to develop our multiresolution algorithm. We can also highlight in

the proof any pixels for which the ink amounts exceed the ink limit for a given paper and printing press.

- 6. Output the results.** The final step saves the preview image, the color separations, and the proof image. We also automatically generate a script capable of including these items in a page layout within a desktop publishing application.

The interactive mode of our application performs each of these tasks as well, but provides more flexibility by allowing the user to view intermediate results and alter the parameters of the algorithms at any point in the process. Of particular use is the ability to fine-tune the inks that the combinatorial optimizer selects by trying similar combinations. The gamut mapping stage quickly provides a preview of the image's printed appearance for the user to compare to the original image. Once the user settles on a choice of inks, she can instruct the system to begin the more time-consuming process of computing the appropriate color separations.

The application's user interface includes three major components: the image viewer, the color palette, and the gamut visualization tool. The image viewer, pictured in Figure 7-1, displays two images side-by-side for comparison. The left side always shows the original input image, while the right side can display the preview image resulting from gamut mapping, or any one of the separations, or the proof image obtained by recombining the separations. The user can pan and zoom these two views simultaneously to compare any region of interest. She also has the option of choosing an ink simply by selecting a color in the original image, which instructs the system to find the closest available ink to that color.

The currently selected paper and inks are displayed in the color palette, pictured in Figure 7-2(a). In the figure, the user has specified a paper and three inks. The question mark in the space reserved for the fourth ink indicates that the user wants the optimizer to choose the best candidates from the specified PANTONE set of inks. The three constrained inks were chosen from sample swatches of the available inks, presented in the ink selection window shown in Figure 7-2(b).

Choosing inks by hand is facilitated by the gamut visualization tool shown in Figure 7-3. This tool provides a three-dimensional view of color space that can be rotated, translated, and magnified interactively. The visualization displays the printing gamut corresponding to the current choice of paper and inks, as well as the location in color space of the original image's colors. By observing how many image colors lie outside the printing gamut, the user gets a sense of how well the current choice of paper and inks will reproduce the image.

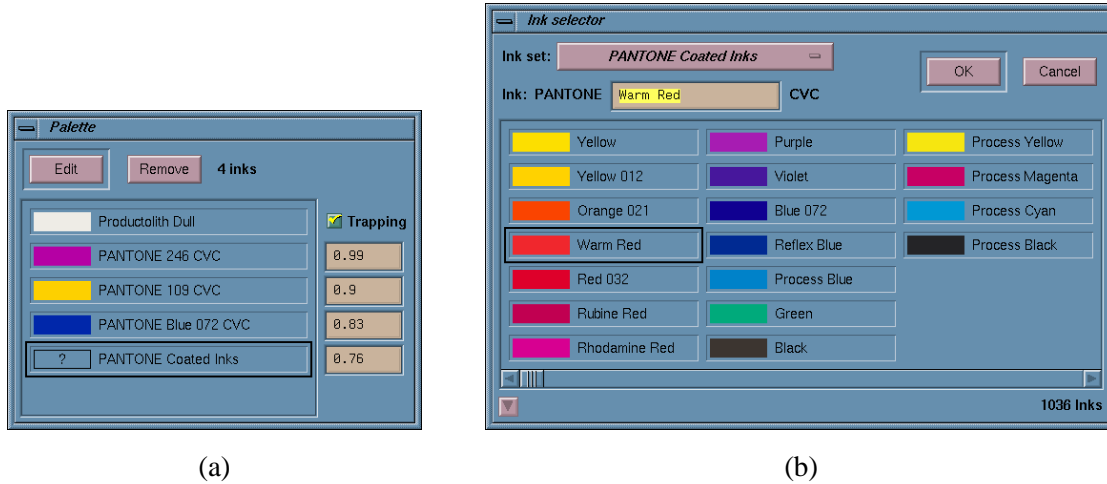


Figure 7-2 The interactive system's color palette (a) and ink selector (b).

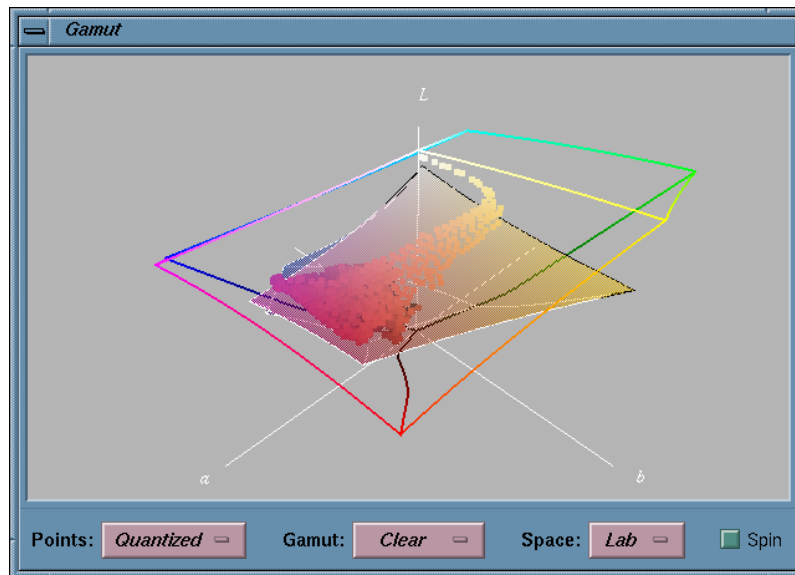


Figure 7-3 The interactive system's gamut visualization tool. This example displays in $L^*a^*b^*$ color space the printing gamut corresponding to the paper and first three inks shown in Figure 7-2(a). The cluster of dots shows the location of colors in the original image of Figure 7-1, while the large outlined volume shows the extent of all monitor colors.

If the gamut has an obvious shortcoming, the user can choose new inks by selecting colors directly according to their three-dimensional locations within the color space.

7.2 Experimental determination of model parameters

We cannot visualize or model printing gamuts unless we have data characterizing the papers and inks available to us. Our early work on duotone printing relied on spectral data included in LightSource's Colortron color measurement package. The data included spectral reflectance values for each of about 1000 PANTONE inks printed on white paper. For each ink, this data set provided only one spectral measurement, and therefore we could only derive one reflectance or transmittance parameter from it. Because printing inks are not very opaque, we assumed the inks were non-reflective ($R_i = \tilde{R}_i = 0$) and derived the transmittance T_i for each ink from the data. The ink layering model we described in the duotone research article essentially amounts to the ideal filter model of equation (3.5), with the addition of a small constant correction for Fresnel reflectance at the interface between air and ink [Power et al. 96]. We used this layering model to predict the printing primaries for the Neugebauer model, which we used with a table look-up for dot gain and without any modifications to account for trapping.

While many aspects of our early gamut model were admittedly simplistic, it performed admirably for two-ink gamuts and allowed us to make some striking duotone reproductions. The need for better data and improvements in the gamut model became apparent when we broadened the scope of our research to include printing with more than two inks. In effect, increasing the number of inks puts our approach in competition with process color printing, and therefore we require more accuracy of the gamut model.

To improve our model, we conducted an experiment whose goal was to determine R_p for a variety of papers, and the parameters R_i , \tilde{R}_i , T_i , and γ_i for a number of inks. We also wanted to find how the trapping fractions t_{pi} , t_{pji} , t_{pijk} , and so on depend on paper and inks. Finally, we were curious to see if the corrections for Fresnel reflection in equations (3.8) and (3.9) are necessary.

We printed a systematic rectangular grid of varying colors using 18 inks that we refer to as the PANTONE *primaries* (the 14 PANTONE colors used to mix other inks, and the four standard process colors) on 13 papers (five colored uncoated papers, four glossy coated white papers, and four matte coated white papers). The experiment was divided into seven runs, with the ordering and colors of the six inks in the press varying from one run to the

next. The grid of 2,745 samples included finely spaced steps of each individual ink, as well as coarsely spaced steps of all possible combinations of three inks. In all, we obtained nearly 250,000 printed color samples. Using a spectrophotometer mounted on a robot arm, we measured absolute spectral reflectances of 153,720 samples from five uncoated colored papers and three coated white papers.

7.2.1 *Characterizing papers*

The first task we accomplished with the experimental data was to determine the reflectance of each paper. The measurements included more than 500 reflectance spectra of each unprinted paper from a number of different sheets and from varying locations on those sheets. Due to the variations in color normally found in papers, the measurements varied by as much as 0.1 in absolute reflectance within each wavelength band. We obtained a single reflectance spectrum for each paper by assigning to R_p the median value of the measured values within each band. Using the median helped to eliminate the occasional outlying value caused by dirty or contaminated samples. The results of our analysis of the eight papers that we measured appear in tabular and graphical form in Appendix A.

7.2.2 *Characterizing inks*

Our next task was to characterize the inks by finding their dot gain, trapping, reflectance, transmittance, and Fresnel reflection parameters. We could see that some of the ink samples were poorly printed, in some cases because an ink was contaminated by impurities and in others because the press operator had failed to keep the supply of ink constant across the width of the page. We decided to discard from our data set all the samples within a given run that incorporated any ink that was printed poorly in that run. We detected these inks algorithmically by examining the range of variation in samples printed with just one ink. For a given ink in a given run, we took the difference between the maximum and minimum measured reflectances (in each wavelength band) for that ink, and labeled the ink as poorly printed if these differences averaged more than 0.1 over all wavelengths. In effect, we discarded inks that varied more on average than the papers did at their worst.

Among the seven print runs, we found ten instances of inks that printed poorly. Of these, two inks (PANTONE Process Yellow and PANTONE Violet) were not printed in any other runs, and therefore our experiment provided no useful data on these inks. We later obtained better samples of Process Yellow along with the other three process colors and two spot

colors used in our example images, though these additional samples were only printed on white coated paper.

Dot gain

The property of an ink that is most straightforward to determine is its dot gain. Recall from equation (3.1) that our gamut model for a single halftoned ink consists of a straight line segment connecting the paper color and the color of the ink printed on paper at 100% coverage. This is true even in the color space of the 36-band spectral measurements made by our spectrophotometer. Translating equation (3.1) into spectral terms, we can write the measured reflectance spectrum of ink i printed with a coverage of α_i as follows:

$$M(\alpha_i) = (1 - t_{pi} \alpha_i) R_p + t_{pi} \alpha_i R_{pi}$$

Because this equation is linear in α_i , we can rewrite it completely in terms of quantities we can measure:

$$M(\alpha_i) = M(0) + \alpha_i (M(1) - M(0))$$

In this form, we can see that the measurement of an ink printed at coverage α_i should lie a fraction α_i along the line segment in color space between the measured color of paper $M(0)$ and the measured color of the full-coverage ink $M(1)$. In practice, the measurements of halftoned samples deviate slightly from a straight line because of variations in paper color or simply because our model's assumptions are faulty. However, we can determine the actual coverage value α_i (as opposed to the intended coverage value $\bar{\alpha}_i$) by projecting the halftoned color $M(\alpha_i)$ orthogonally onto the line segment connecting $M(0)$ and $M(1)$:

$$\alpha_i = \frac{(M(\alpha_i) - M(0)) \cdot (M(1) - M(0))}{\|M(1) - M(0)\|^2}$$

When we plot the resulting value of α_i versus the corresponding coverage value $\bar{\alpha}_i$ that was specified for a sequence of 32 halftoned samples of a particular ink on a given paper, we get a graph like the one shown in Figure 7-4. The behavior of the data points can be captured remarkably well by a single parameter γ_i in a power law curve (also shown in the figure):

$$1 - \alpha_i = (1 - \bar{\alpha}_i)^{\gamma_i}$$

We used a nonlinear curve-fitting procedure provided by the Matlab package to determine γ_i values for each of the eighteen inks on one of the coated white papers. The results ranged from 0.43 for PANTONE Process Black to 0.64 for PANTONE Yellow 012 (though there was no obvious pattern to the numbers).

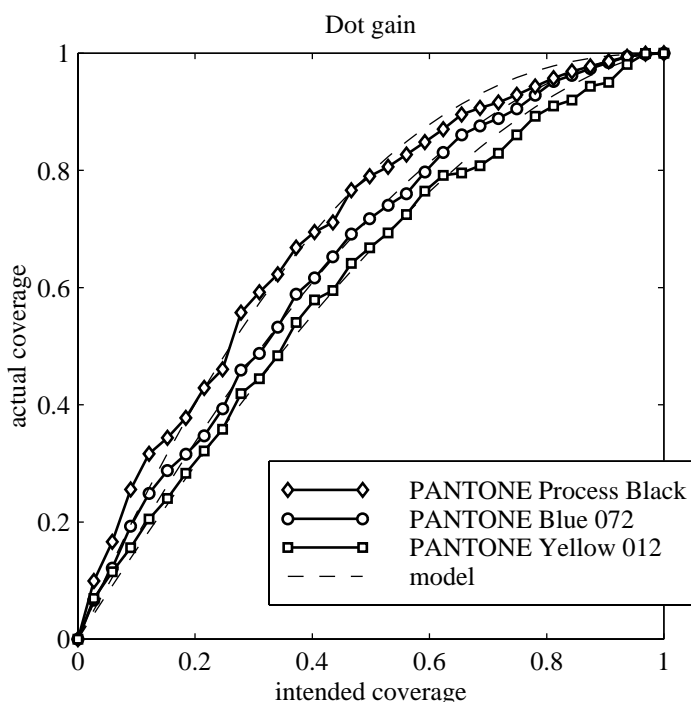


Figure 7-4 Typical dot gain curves obtained from experimental data.

Reflectance, transmittance, trapping, and Fresnel coefficients

It is convenient that we can determine dot gain independently of the other parameters of the gamut model. The remaining properties of inks (reflectance, transmittance, Fresnel coefficients, and trapping fractions) are interrelated in such a way that we cannot easily determine one independently of the others. In order to distinguish between light reflected by an ink and light reflected by the paper after transmission through that ink, we need measurements of the ink printed atop various differently colored backgrounds. Our systematic printed grid offered us samples of each ink printed on many different paper colors, as well as samples of each ink printed atop a number of other inks. We originally used samples of an ink printed individually on all the papers to determine the spectral properties of that ink, but we found that the difference between coated and uncoated papers prevented the model from achieving a good fit to the data. We therefore focused on fitting the spectral parameters of each ink based on samples printed on just one white paper, but in combination with other inks.

The process of fitting model parameters to data can be posed as a large continuous optimization problem. The objective is to minimize the perceptual difference between colors predicted by the gamut model and the colors we measured—in other words, the average

$L^*a^*b^*$ distance between modeled and measured colors. In our case, we want to optimize over the 36-dimensional spectra R_i , \tilde{R}_i , and T_i for 16 inks, along with four Fresnel coefficients and a multitude of trapping fractions. Of course, optimizing nearly 2,000 unknowns can be terribly time-consuming, particularly when one evaluation of the objective function involves comparing thousands of spectral reflectances. A more efficient approach is to split the variables according to their dependencies: we treat R_i , \tilde{R}_i , and T_i as functions of ink and wavelength, trapping fractions as functions of the paper and inks involved, and the four Fresnel coefficients as unknown constants. With this splitting in mind, we developed a Matlab procedure that iteratively improves the parameters of our gamut model to achieve the best fit to the measured samples. The following pseudocode outlines the structure of the procedure without the encumbrance of Matlab's technicalities.

```

function FitModel
  initialize all  $T_i$  values to 1
  initialize all  $R_i$  and  $\tilde{R}_i$  values to 0
  initialize all  $t_{pi}$ ,  $t_{pij}$ , and  $t_{pijk}$  values to 1
  initialize  $\rho_{ai}$ ,  $\rho_{ia}$ ,  $\rho_{ip}$ , and  $\rho_{pi}$  to 0
  error := EvaluateError( $T_i, R_i, \dots, \rho_{ip}, \rho_{pi}$ )
  repeat
    for each ink  $i$  and wavelength band  $\lambda$ 
      OptimizeSpectra( $T_i(\lambda), R_i(\lambda), \tilde{R}_i(\lambda)$ )
    for each paper  $p$  and ink  $i$ 
      OptimizeTrapping( $t_{pi}$ )
    for each paper  $p$ , ink  $i$ , and ink  $j$ 
      OptimizeTrapping( $t_{pij}$ )
    for each paper  $p$ , ink  $i$ , ink  $j$ , and ink  $k$ 
      OptimizeTrapping( $t_{pijk}$ )
    OptimizeFresnel( $\rho_{ai}, \rho_{ia}, \rho_{ip}, \rho_{pi}$ )
    lastError := error
    error := EvaluateError( $T_i, R_i, \dots, \rho_{ip}, \rho_{pi}$ )
  until lastError – error <  $\epsilon$ 
end function

```

The *EvaluateError* function, as its name implies, calculates how well a gamut model with the current ink parameters fits the measured spectral data. We compute this error metric by averaging (over all solid-colored samples in the data set for a given paper) the norm of the difference in $L^*a^*b^*$ color space between predicted and measured colors. In order to predict the color of a given printed sample, we apply our ink layering model to another measured sample with all but the topmost ink. For example, to predict the color of a sample including

Table 7-1 Labels used for the variations used in fitting ink models to data. Here ρ stands for all four Fresnel reflection coefficients and t stands for all trapping fractions. Similarly, T , R , and \tilde{R} refer to all wavelengths of transmittance, reflectance, and back surface reflectance of all inks.

	$\rho = 0$		optimal ρ	
	$t = 1$	optimal t	$t = 1$	optimal t
optimal T ; $R = \tilde{R} = 0$	A	B	C	D
optimal T and R ; $\tilde{R} = 0$	E	F	G	H
optimal T , R , and \tilde{R}	I	J	K	L

inks i , j , and k , we first find a location nearby on the page where a sample M_{ij} has been printed with inks i and j but not k .¹ We then model the effect of printing ink k atop the sample M_{ij} , and compare the result to the measured sample M_{ijk} with all three inks. The prediction relies on the Kubelka-Munk model of equation (3.7) and the Fresnel reflection corrections of equations (3.8) and (3.9) to determine ideally trapped printing primaries, then applies the modified Neugebauer equations of Section 3.2 to account for imperfect trapping. Note that dot gain does not play a role in the evaluation function since we consider only samples that are printed with 100% coverage.

The *OptimizeSpectra*, *OptimizeTrapping*, and *OptimizeFresnel* functions each use one of Matlab's constrained nonlinear optimization procedures to minimize the *EvaluateError* function over a subset of the ink parameters. We specialized the three optimization functions to efficiently compute just the portion of the objective function that is affected by their particular ink parameters. For example, *OptimizeSpectra* is charged with finding the best reflectance and transmittance parameters for one wavelength band of one ink at a time, assuming the current trapping and Fresnel coefficients are held fixed. Therefore, *OptimizeSpectra* needs to compare predictions to measurements only for samples whose topmost ink is the one in question.

Model variations

As we pointed out at the beginning of this section, we are interested not only in obtaining parameter values for our model, but also in determining their relative importance. We

¹ We choose a sample that is nearby to reduce the influence on our calculation of the unavoidable variations in paper color and ink density across the page.

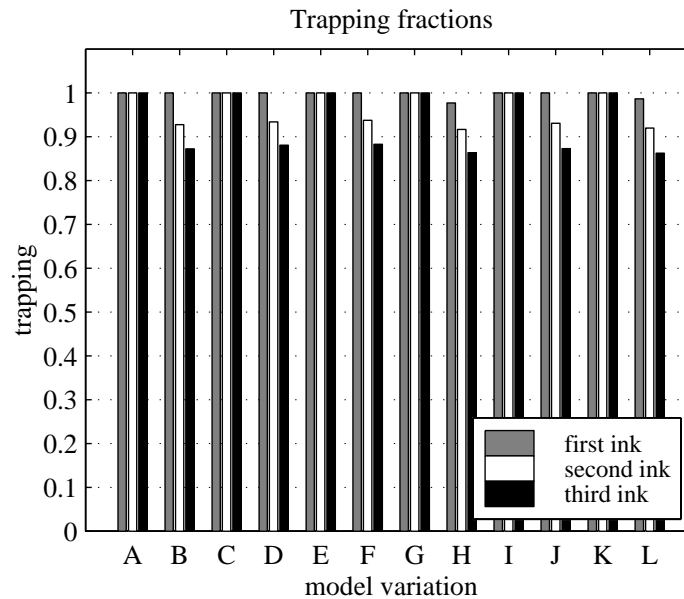


Figure 7-5 Experimentally determined trapping fractions for each model variation.

therefore used the procedure above to fit twelve variations of the model parameters to the measured data. Table 7-1 enumerates the variations, some of which constrain R_i and \tilde{R}_i to be zero, while others constrain only \tilde{R}_i , and still others permit both to be optimized. Likewise, some of the variations assume perfect trapping while others optimize the trapping fractions; some assume no Fresnel reflection while others optimize the Fresnel coefficients.

One of the conclusions we were able to draw from fitting the model parameters was that the amount of ink trapped by the page depends primarily on the type of paper (coated versus uncoated) and on the number of ink layers already printed. For example, coated papers typically trap between 98 and 100% of the first ink printed and between 88 and 95% of the second ink printed atop the first, regardless of the ink colors. We subsequently treated trapping fractions as functions only of the paper and the number of ink layers, rather than as functions of each particular ink. The results of optimizing these generic trapping fractions for the set of PANTONE primaries printed on Productolith Dull White paper are shown in Figure 7-5, for each of the twelve model variations. Half of the variations constrain the trapping fractions to be 1, while each of the remaining variations yields a similar sequence of decreasing trapping fractions.

The optimized Fresnel reflection coefficients for the same inks and model variations are shown in Figure 7-6. Again, half the variations constrain the parameters, in this case to be

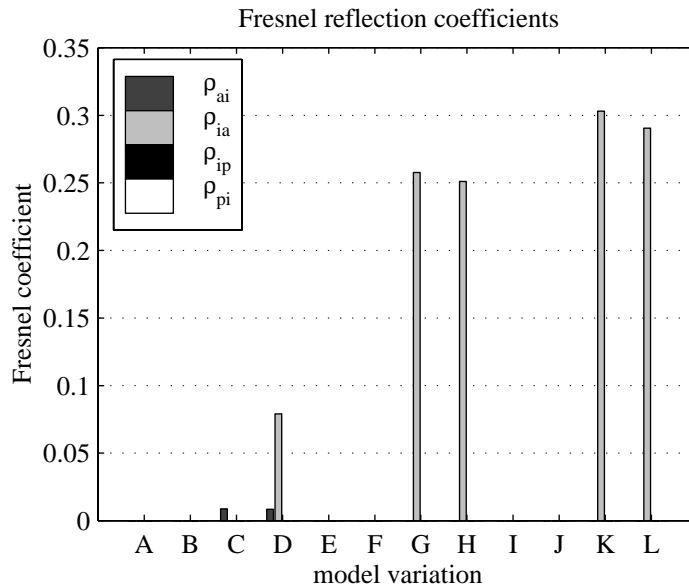


Figure 7-6 Experimentally determined Fresnel reflection coefficients for each model variation.

zero. The remaining variations produce quite different results, depending on whether or not we constrain the ink reflectances and back-surface reflectances to be zero. The inks are permitted to reflect light in models G, H, K, and L; the large values of ρ_{ia} in these cases indicate that a significant fraction of the light reflected by the ink pigments is actually trapped within the ink by internal reflection at the interface with air.

In addition to fitting our models to the data obtained for the PANTONE primaries, we also determined ink parameters for two smaller data sets. One of these consisted of the four process colors (cyan, magenta, yellow, and black, denoted CMYK), while the other included two spot colors (PANTONE 173 and PANTONE 394). Because these inks were printed on only one paper and with far fewer overprinted combinations than the primary data set, we were not able to fit all the model parameters for these inks. We relied on the trapping fractions and Fresnel coefficients obtained for the PANTONE primaries, holding these values fixed while we optimized the transmittance and reflectance parameters of the four process colors and the two spot colors. There was not enough data to determine the back-surface reflectance \tilde{R}_i , so we could not use model variations I, J, K, or L for any of these inks.

If the results of our model-fitting process can be reduced to one number, that number is the average $L^*a^*b^*$ error between predicted and measured colors. We graph this quantity for each of our model variations in Figure 7-7. The first graph compares these variations for the samples considered during optimization of the model parameters—that is, only the

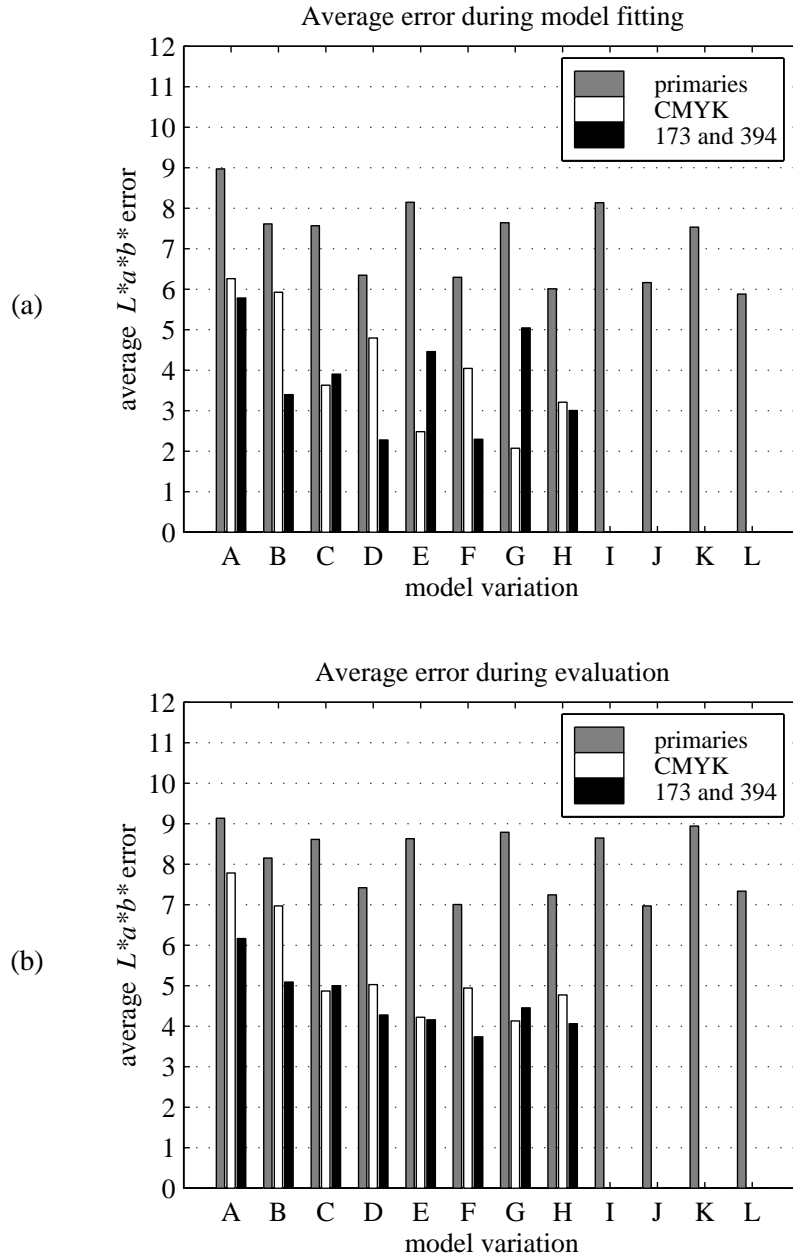


Figure 7-7 Error computed for each model variation: (a) error metric used during model fitting, including only solid-colored samples, and (b) error after evaluating all samples, including halftoned ones. Models I through L are not applicable to the CMYK data set or the 173 and 394 data set, so no errors are shown.

samples printed at 100% coverage. In Figure 7-7(b) we show the average error of each variation when we use the models to predict halftoned colors as well as solid colors. Both graphs indicate clearly that incorporating trapping fractions less than one into the gamut model improves the accuracy in fitting the larger data set of PANTONE primaries. The CMYK and spot color data sets also show an improved fit when the effect of trapping is included, though the improvement is less significant (perhaps because we could not optimize the trapping fractions for these inks).

We have a harder time drawing conclusions about the other differences between model variations. Using optimized values for ink reflectance R_i in variations E through H brings about a small improvement over variations A through D, while incorporating back-surface reflectance (variations I through L) has little to no beneficial effect. Optimizing over the Fresnel reflection coefficients appears to improve the fit of the models somewhat, but the physical plausibility of the resulting numbers deserves further investigation.

In general, we can conclude from our experiment in model fitting that variation H (optimized reflectance, transmittance, trapping, and Fresnel reflection, with no back surface reflectance) is the most promising model, with variations D and G not far behind. Because all the variations result in quite similar error values, we are left with the concern that a significant portion of those errors may be due to the variability inherent in paper colors and press operation. If that is the case and we cannot avoid making somewhat erroneous predictions, the simplicity of model variation A may make it more appealing than any of the others; the determination of ink transmittance alone reduces to a linear least-squares problem. Regardless of these considerations, the true test of which variation best suits our needs is a printing experiment involving images made with these models—the topic of the next section.

7.3 Printed examples

In our early research on reproducing color images as duotones, we attempted to reproduce with two or three inks the appearance of process-color prints of a variety of images. We briefly describe those results below, then move on to more recent results that were printed with more than two custom inks.

7.3.1 *Examples of duotone printing*

Example 1 depicts a painting by Cézanne printed as three duotones. The first two duotones, both printed with PANTONE 130 (a golden orange color) and Process Black, demonstrate the difference between the conventional approach to printing duotones and our approach. The conventional duotone, created using Adobe Photoshop, is essentially monochromatic; Photoshop determines the two ink amounts for each pixel as functions of the original image's luminance, disregarding hue and saturation information. Our approach, on the other hand, maintains some of the contrast in hue between the objects in the image. The third image in the series is printed with two colored inks selected by the simulated annealing optimizer (PANTONE 198 and 604) followed by Process Black. We treated printing with two colors and black as a special case in our duotone work by first removing as much black as possible from each color in the original image, applying the duotone gamut mapping, and then adding back just enough black ink to achieve the right luminance.

Often a pair of inks that works well for one image can be used for other images as well. As a case in point, the photograph of a koala in Example 2 and the first reproduction in Example 3 are both printed with PANTONE 144 and 546. The Renoir painting in Example 3, like many images, can be reproduced well using remarkably different pairs of inks. The second version in this example uses PANTONE 172 and 354, resulting in a quite different, but still successful reproduction.

Though skin tones are notoriously difficult to reproduce, photographs of people are not beyond the scope of two-color reproduction. Examples 4, 5, and 6 show three portrait photographs printed as duotones. All three duotones are printed with PANTONE 152 and Process Blue, inks that were selected by simulated annealing for the photograph in Example 4.

Example 7 shows a photograph of a sunset printed with two hand-picked inks (in this case, PANTONE 151 and 246) and Process Black. We took the same steps here as we did for the three-color print in Example 1: remove black from the original image, create a duotone, and return black to the printed image. The purples and oranges in sunset images are often outside the process color gamut, but we were able to portray these colors by choosing custom inks suited to this sunset.

The final example from our duotone project shows a painting by Schiele printed as a duotone on colored paper with inks selected by simulated annealing. By using yellow paper in addition to the green and red inks (PANTONE 329 and Warm Red) in Example 8, we can reproduce the three dominant hues present in the original image.

Table 7-2 Chromaticities used to convert *RGB* colors to device-independent coordinates.

	x	y
white	0.313	0.329
red	0.640	0.338
green	0.300	0.600
blue	0.150	0.060

7.3.2 Examples of *n*-tone printing

The duotone examples described above were intended to provide tolerable reproductions without necessarily matching the original image colors precisely. By contrast, printing images with four or more custom inks costs as much as or more than process color printing, and therefore to be competitive we need to make accurate reproductions.

The ultimate goal of any color reproduction is the faithful reproduction of physical reality. Unfortunately, it's not easy to obtain a physical image: full-spectrum cameras are rare and expensive. Instead, we often deal with simplified *RGB* images captured by tri-filter cameras and films. In this context, the original full-spectrum image is lost. Consequently, for want of anything better, an *objective* original image is replaced by a *subjective* original image—one that “looks good” on the monitor. We too use this widely accepted criterion and try to match the appearance of images on a calibrated monitor.

We selected a variety of stock photography images with which to test our algorithms. Our goal was to reproduce them as they appeared on our calibrated monitor (a BARCO Reference Calibrator, with stable and known phosphor chromaticities). The images, like most that are commercially available, provided device-dependent *RGB* colors, but without documented reference to any particular device. We therefore treated the colors as though they referred to the red, green, and blue primaries in both Kodak's PhotoCD standard and the HDTV (high-definition television) standard, using the chromaticities in Table 7-2 and a gamma correction of 1.5 to convert the colors to device-independent *XYZ* coordinates.

We used the algorithms discussed in Chapters 4, 5, and 6 to choose inks, apply gamut mappings, and produce separations for each of our example images. For some of the images, we picked the inks ourselves, while for others we used the genetic algorithm to choose the inks. The genetic algorithm provided us with a number of good solutions; we chose from these possibilities by considering the subjective appeal of the corresponding preview images. Because we could only print a limited number of inks on a page, we favored inks that helped reproduce other images on the same sheet.

Except where noted otherwise, we used the cylindrical gamut mapping strategy and the multiresolution separation algorithm. We applied these algorithms using three models of ink layering: variations A, D, and H from Section 7.2. Variation A treats each ink layer as an ideal filter with perfect trapping and no Fresnel reflections, transmitting some light and absorbing the rest. Variation D also treats inks as ideal filters, but incorporates adjustments to account for Fresnel reflection and imperfect trapping. Finally, variation H includes the reflectance of each ink in addition to the parameters of variation D.

We were unable to distinguish the results produced by variations D and H. Variation A, on the other hand, was noticeably worse than the other variations at reproducing the dark shadow areas of some images, presumably because we do not print enough ink when the model does not account for imperfect trapping. These conclusions were consistent from one printing press to another: we obtained virtually identical results on two different six-color sheet-fed offset presses made by Mitsubishi and Komori. All the examples discussed below were generated using model variation D—the simplest model that produced consistently good results. The separations were halftoned at 175 lines per inch using conventional screening and printed on 100-pound white Productolith dull coated paper.

- **Line worker at sunset:** We first use this image to demonstrate that our system can be used to compute separations for the standard process color inks. Example 9(a) displays a process color print made using Adobe Photoshop, while Example 9(b) shows our system’s process color reproduction. There are subtle differences between the two versions, particularly where Photoshop exaggerates the red in the underside of the clouds; our reproduction more accurately reflects the colors of our original monitor display.

Next, in Example 9(c) we present a color reproduction using only three inks. The image is printed with PANTONE Purple, Blue 072, and Process Yellow. This combination of custom inks was ranked second by the genetic algorithm after evaluating 5,000 possible triples of inks chosen from the 16 measured PANTONE primaries. We selected this combination rather than the one that ranked first because the differences in the objective function were statistically insignificant and we preferred the preview image in this case. Also note that we replaced PANTONE Yellow with Process Yellow since we were already using the latter ink for other reproductions on the same printed sheet. Because the process color versions include black ink, they are better able to reproduce the dark silhouetted areas in the image. On the other hand, despite

its shortcomings the three-color print demonstrates that color images can be reproduced effectively with three custom-selected inks.

- **Wildflowers:** We printed the image in Example 10 with four custom inks: PANTONE Blue 072, Process Magenta, Process Yellow, and Green. We selected this combination of inks from among the genetic algorithm’s top five choices of ink combinations for this image. Because there are many smooth color gradations where the picture is out of focus, this image provides a good opportunity to test the smoothness of our separation algorithm. When we set the reference ink amounts α_{ref} to zero in Example 10(a), the resulting reproduction exhibits many artificial discontinuities highlighting contours of the image that should be smoothly colored. In contrast, when we allow the multiresolution separation algorithm of Section 6.3.1 to set α_{ref} using an image pyramid, we obtain the smooth reproduction in Example 10(b).
- **Tulip background:** Example 11 displays reproductions of an image of tulips printed with PANTONE Warm Red, Purple, Process Yellow, and Green. These four custom inks comprise the first choice of the genetic algorithm after 5,000 evaluations of four inks chosen from the 16 PANTONE primaries.

We present several different reproductions of the image to illustrate the need for some of the components of our system. The first version of the tulips in Example 11(a) was made using all the features of our system, while each of the remaining versions is “defective” in some way. In Example 11(b), we have eliminated the correction for dot gain from the gamut model, yielding a very dark reproduction. Example 11(c) was produced with the dot gain correction, but without any gamut mapping. In this case, some of the image colors are clamped to the boundary of the gamut by the penalties in the separation objective function. The effect of clamping is particularly evident in the yellow flowers, where instead of smooth gradations of color we see solid regions of the incorrect hue. In the last version of this image, in Example 11(d), we applied gamut mapping but we intentionally used a coarse discretization of the gamut. While all our other n -tone reproductions use piecewise-linear interpolation between 64×64 bins, this version uses piecewise-constant interpolation between only 16×16 bins. The effect of using such a coarse gamut mapping is visible where the yellow flowers have turned tan. In order to move just a few very saturated yellows into the gamut, the mapping desaturates all the yellow hues in the same large bin.

- **Hats:** This example was printed with the four process colors and with six inks, five of which were chosen by the genetic algorithm. The optimizer selected PANTONE Process Black, Process Cyan, Process Magenta, Process Yellow, and Green. It also included Red 032, which we replaced with Orange 021 because the resulting combination yielded essentially the same preview and numerical error, but allowed us to improve the results for our next example.

We have included process-color and six-color versions produced by cylindrical, ellipsoidal, and spherical gamut mappings. The differences between the mappings are most apparent in the highlights on the yellow hat, which turn white under the cylindrical mapping of Examples 12(a) and (d), and flat yellow under the spherical mapping of Examples 12(c) and (f). The ellipsoidal mapping in Examples 12(b) and (e) produces a compromise that appears most like the monitor image, with highlights that are both brighter and whiter than the rest of the yellow hat. It comes as no surprise that by including orange and green inks in the six-ink reproductions, we can portray the vibrant orange and green hats in the original monitor image more accurately than with process color inks.

- **Boy and girl with beach ball:** Example 13(a) displays a process-color reproduction of a beach scene. In Example 13(b) we add PANTONE Orange 021 and Green inks (resulting in the same combination of inks as the previous example). This six-ink combination produces a more saturated orange stripe on the surfboard and more vivid green on the beach ball.

As in the tulips example, the remaining versions of the beach ball image illustrate the defects that result when we omit portions of our usual reproduction process. Example 13(c) appears too dark because we left out the correction for dot gain. Example 13(d) was produced without any gamut mapping, relying on the separation stage to make out-of-gamut colors printable. Most of the original image colors are within the gamut, but the cyan portions of the image stand out because these colors are clamped to the gamut's surface. The variation in Example 13(e) uses a coarse discretization of the gamut along with piecewise-constant interpolation of the gamut boundary. The result is an image with a dark stripe at the top and gray stripes in the water. Finally, in Example 13(f), we set $\alpha_{\text{ref}} = 0.5$ instead of using the multiresolution separation algorithm, only to obtain an image riddled with unwanted speckles.

Table 7-3 Statistics for process-color example images. Approximate computation times are given for an SGI O2 workstation with a 174 MHz R10000 processor.

image	resolution	distinct colors	gamut mapping	separations	
				inks	time
sunset	1000×660	120,056	4 sec	4	13 min
tulips	1000×660	349,584	8 sec	4	17 min
wildflowers	1000×660	200,013	6 sec	4	14 min
hats	1000×660	104,266	5 sec	4	11 min
beach ball	1000×660	188,285	5 sec	4	15 min
chocolates	660×660	103,695	—	4	9 min

- **Assorted chocolates:** We printed the images in Examples 14(a) through (c) using process colors and in Examples 14(d) through (f) using three hand-picked inks: PANTONE 173, Process Black, and 394. In versions (b) and (e) we intentionally offset each separation by 0.24 millimeters in a different direction to simulate poorly registered printing plates. The artifacts of misregistration are even more obvious in versions (c) and (f), where we offset each separation by 0.36 millimeters.

We hypothesized that misregistration artifacts (bands of incorrect hues at the edges of objects) would be less objectionable if we chose inks that closely matched the colors of objects in the image. We chose an image of chocolates because misregistration artifacts are particularly unappealing in pictures of foods, and because the limited range of colors in this image can be closely approximated by the gamut of just three inks (yellow, brown, and black). The misregistered process color print in Example 14(c) exhibits green and orange fringes around the nuts, while the corresponding artifacts in the three-color version of Example 14(f) are not nearly as noticeable. This example provides preliminary evidence that we might be able to modify the ink-selection objective function to choose inks that are robust with respect to misregistration.

7.4 Timings and complexity

Tables 7-3 and 7-4 presents some statistics for the example images discussed above, and the processing time required by each stage of our algorithm. The time it takes to select optimal inks varies widely depending on the combinatorial complexity of the problem: the time increases exponentially with the number of inks being selected and with the size of the ink set from which they are chosen. We suspect that by adding a few well-chosen heuristics to

Table 7-4 Statistics for custom-ink example images. Optimization chose inks from the 16 measured in the experiment described in Section 7.2. Approximate computation times are given for an SGI O2 workstation with a 174 MHz R10000 processor.

image	resolution	distinct colors	optimization		gamut mapping	separations	
			inks	time		inks	time
sunset	1000×660	120,056	3	5 min	2 sec	3	7 min
tulips	1000×660	349,584	4	15 min	8 sec	4	22 min
wildflowers	1000×660	200,013	4	15 min	6 sec	4	16 min
hats	1000×660	104,266	6	300 min	66 sec	6	51 min
beach ball	1000×660	188,285	—	—	71 sec	6	75 min
chocolates	660×660	103,695	—	—	—	3	6 min

our general-purpose optimizer, we could eliminate from consideration a large fraction of the bad combinations that we currently spend time evaluating. Our implementation can compute the optimal inks as a preprocessing step, or, in our interactive environment, the user can monitor the progress of the optimizer and interrupt it when it achieves an acceptable solution. Furthermore, because the genetic algorithm maintains a population of possible solutions, we present to the user a number of choices for comparison.

The gamut mapping stage takes only a few seconds, even for six-ink gamuts with fairly complex shapes, making it well suited for use in our interactive application. Users of our system can easily modify their choice of inks or the parameters of the gamut mapping, and quickly see the results of these changes.

The time required to compute separations is roughly proportional to the number of inks times the number of pixels in the original image. Because the separation objective function includes penalties, the continuous optimizer takes more iterations to converge for some colors—particularly those near or outside gamut boundaries—than others. Separations typically take too long to compute in an interactive setting, but fortunately the user can try out many choices of inks (looking at previews) before settling on a combination for which separations are needed. The system can then compute and save the separations as a postprocessing step.

The efficiency of the separator depends on how fast we can evaluate the objective function and its partial derivatives. The major component of these computations is the evaluation of the gamut model and its derivatives. Our original implementation required about seven hours to compute six-color separations (for the same processor and image resolutions listed in Tables 7-3 and 7-4) using the multiresolution separation algorithm, and three times

as long when the optimizer started with constant α values for each pixel. We made the separator four or five times faster by re-implementing special cases of our gamut model using factored versions of the Neugebauer equations. For example, instead of equation (3.3) we used the following more efficient expression for a three-color gamut:

$$\begin{aligned} \mathbf{c}(\boldsymbol{\alpha}) = & (1 - t_{p1}\alpha_1) \{ (1 - t_{p2}\alpha_2) [(1 - t_{p3}\alpha_3) \mathbf{g}_p + t_{p3}\alpha_3 \mathbf{g}_{p3}] \\ & + t_{p2}\alpha_2 [(1 - t_{p23}\alpha_3) \mathbf{g}_{p2} + t_{p23}\alpha_3 \mathbf{g}_{p23}] \} \\ & + t_{p1}\alpha_1 \{ (1 - t_{p12}\alpha_2) [(1 - t_{p13}\alpha_3) \mathbf{g}_{p1} + t_{p13}\alpha_3 \mathbf{g}_{p13}] \\ & + t_{p12}\alpha_2 [(1 - t_{p123}\alpha_3) \mathbf{g}_{p12} + t_{p123}\alpha_3 \mathbf{g}_{p123}] \} \end{aligned}$$

We expect that further tuning could make the separation algorithm execute at speeds suitable for our interactive application rather than the postprocessing stage.

✦ Conclusion ✦

In this dissertation we have laid out a general framework for multicolor printing with custom inks. This final chapter briefly reviews our contributions and discusses possible areas for future research.

8.1 Summary

When we first envisioned the reproduction of color images with custom inks, we could not find a single piece of research on choosing inks that are optimal for a given image. It may seem only natural to choose the best inks for the job, but developing a system for that purpose requires the right combination of ingredients: mathematical models, experimental data, effective algorithms, and computing hardware. We brought those ingredients together, introducing new algorithms in the process, and demonstrated their potential by printing color images with “entirely new printing processes,” as Hardy and Wurzburg might say. The specific contributions of this dissertation appear below.

- **Gamut modeling.** Our gamut model combines the Kubelka-Munk model of colorant layers with the Neugebauer model of color halftone printing. The advantage of this combination is that we can predict the printing gamut that will result from arbitrary choices of inks and papers without making any new measurements—a crucial advantage if we plan to optimize over inks and papers. We also incorporated in our gamut model modifications to Neugebauer’s equations that account for the effects of trapping and dot gain.
- **Experimental ink characterization.** We developed a procedure for determining the parameters of our gamut model from measurements of reflectance spectra. We require that the measurements include samples of each ink printed atop a variety of background colors (either different papers or different inks), but we do not require every possible overprinted combination of inks. The procedure determines the dot gain, transmittance, reflectance, trapping, and Fresnel reflection properties of each

ink. We avoid the expense of simultaneously fitting all the model parameters to a large set of data by using an iterative approach that repeatedly solves a sequence of smaller optimization problems until the parameters achieve the best fit to the data.

- **Gamut mapping.** We introduced a new ellipsoidal family of gamut mappings that effectively fills a gap between the existing cylindrical and spherical variations of gamut mapping. This parametrized family of mappings is not limited to the realm of offset printing presses and inks; it applies equally well to the adjustment of color images to fit the gamuts of other printing technologies, film output, and all manner of display devices.

Because we construct a gamut mapping that is customized for the image at hand, we can achieve a much more accurate reproduction than is possible with a gamut mapping designed to bring all monitor colors into the same printing gamut. As a case in point, our mapping will not alter an image whose colors all happen to fall within the printing gamut, while many other algorithms will shift the image colors merely because some colors in the monitor gamut (but not in the image) are not printable.

- **Ink selection.** We formulated the challenge of choosing papers and inks as a combinatorial optimization problem and developed simulated annealing and genetic algorithms particularly suited to that problem. These optimizers reliably find a variety of solutions, regardless of the number of inks and the constraints placed on them. The inks can be constrained to come from the one set or different sets; required spot colors can be held fixed while other inks are allowed to vary.
- **Color separation.** We developed a robust algorithm that computes separations for any given image and any given combination of inks. In contrast to separation methods that rely on approximation and interpolation formulas, our algorithm uses numerical optimization to find the ink amounts that will precisely reproduce each desired color. Our original implementation produced artifacts because each pixel was separated independently; we overcame this problem by introducing a new hierarchical algorithm that yields separations just as smooth as the input image.
- **User interface.** Our system evolved from a crude testbed to a well-integrated interactive application over the course of this research. While the design of the user interface

was never a central objective of the project, we developed a number of features helpful enough to deserve more widespread use. The ability to compare two images at any position and level of magnification is one such feature. The entire gamut visualization tool is another; this straightforward use of three-dimensional graphics greatly enriched our understanding of color printing and gamut mapping.

The previous chapter presented practical results of our experiments in custom-ink color image reproduction. We outlined there the steps required to fit the parameters of our gamut model to measured data, and exhibited a variety of printed images produced with our techniques. While we still see room for improvement in these results, they demonstrate the potential for making color reproductions with custom inks that are more accurate or less costly to produce than with standard process inks.

In general, n -tone printing offers the opportunity to match photo and monitor colors better than process color printing for two reasons: First, the inks are chosen specifically for the image, thereby giving us the best gamut of printable colors. Second, we construct a gamut mapping that is based not only on the range of printable colors, but also on the colors that are present in the image. Tailoring the inks *and* the gamut mapping function to suit an image allows us to achieve much greater accuracy than we could with a fixed set of inks or a fixed mapping.

8.2 Extensions

We envision many directions in which our work can be extended. First, there are some undesirable artifacts in our results that have yet to be eliminated. In particular, we occasionally encounter an image containing hues that are not present at all in the printing gamut, and our gamut mapping algorithm ends up compressing these hues into the gamut by desaturating them completely to shades of gray. The resulting reproductions may be objectionable because they contain gray areas, or because a smoothly colored object is portrayed with a sudden shift from a saturated hue to a desaturated gray. The key to solving this problem lies in developing a method of compressing slightly out-of-gamut hues to printable hues without crossing hue-name boundaries, and smoothly varying the amount of desaturation applied to the remaining colors whose hues cannot be reproduced.

Several other ways of extending our approach occurred to us in the course of developing and testing our techniques for custom-ink image reproduction. We briefly describe some of these extensions below.

- **Stochastic screening.** We used conventional halftone screening techniques to produce our printed experiments, but there are a number of reasons to use the more recently developed stochastic screening methods of halftoning. First, stochastic screening can reproduce finer details because the dots are uniformly small, rather than varying in size. Even more importantly, stochastic screening does not suffer from the moiré interference patterns that occur when the separations for four or more inks are printed with traditional halftones at different screen angles. To make use of stochastic screening, we need only measure a handful of printed samples of each ink to determine their dot gain characteristics; the rest of our model of printing gamuts is valid with any bi-level halftone technique.
- **Opaque inks.** Thus far, we have experimented only with standard printing inks that are quite translucent. We are interested in the effects achievable by printing opaque inks on dark papers, mainly because images reproduced that way would be something of a novelty. Although inks that acquire their colors by reflecting a lot of light will necessarily yield smaller printing gamuts than translucent inks, our gamut mapping strategies can still find ways of portraying any image. Nothing in our models of ink layering and halftoning prevent us from using opaque inks, which simply have much smaller transmittance spectra and larger reflectance spectra than the ones we have dealt with. All we require is a set of measurements of opaque inks printed atop a variety of different backgrounds to determine their characteristics.

One possible application of opaque inks is in the printing of “yellow pages” telephone directories, in which the pages are often dyed yellow before being printed. Images reproduced in these directories often look poor because cyan ink appears blue and magenta ink appears red when printed on yellow. Some directories achieve better quality at the cost of customizing the yellow printing on each page to avoid areas with images. By using opaque inks, we could conceivably match an image’s colors without concerning ourselves with the yellow background.

- **Direct measurement of trapping.** We first investigated trapping as a source of inaccuracy in our gamut model when we noticed that overprinted combinations of inks frequently appear mottled even when we request constant coverage. Examination of these samples under a microscope reveals patches of one color where the upper ink sticks and spots of a different color where the ink doesn’t stick, as shown in the photomicrograph in Figure 3-3. Our current technique for determining trapping fractions

relies on a fairly sophisticated numerical data fitting procedure that is simultaneously estimating other parameters of our ink layering model. We can imagine using an algorithm based on image-processing techniques to determine trapping fractions directly from photomicrographs, thereby confirming or improving upon our current values.

- **Perceptually uniform gamut mapping.** Our current gamut mapping implementation performs geometric manipulations of colors in either XYZ or RGB color space. These spaces are convenient because of their linear relationship to one another and to reflectance spectra, but unfortunately they are perceptually quite nonuniform. As a result, when our gamut mapping shifts colors toward a central axis or point, it does not accurately preserve hues (though we do take care to preserve luminance whenever possible). More than one author has recommended performing gamut mapping in a more perceptually uniform color space such as $L^*a^*b^*$ or Lu^*v^* [MacDonald 93, Wolski et al. 94].

Altering the color space in which our gamut mapping takes place is simple in concept, but a challenge arises in practice. We currently take advantage of the fact that the Neugebauer model of a multicolor gamut is bounded by bilinear surfaces in an additive color space. These surfaces are no longer bilinear once we apply a nonlinear transformation to obtain their representation in a perceptually uniform space, and therefore we cannot rely on the analytic techniques we used to determine the extent of the gamut in Chapter 4. We would instead need to develop a method for quickly and thoroughly sampling the gamut's boundary.

- **Localized gamut mapping strategies.** We have thus far relied on the user of our system to choose the parameter κ that controls whether the gamut mapping is cylindrical, ellipsoidal, or spherical. We could instead have the system automatically find the optimal κ for a given image and choice of inks. Furthermore, the ideal strategy for mapping image colors may vary from one region of color space to another. In fact, Montag and Fairchild have suggested using different gamut mapping strategies for light colors and dark colors [Montag & Fairchild 97]. We could extend our mapping to allow different values of κ in the upper and lower halves of the gamut's luminance range.
- **Extended ink selection objective.** We are considering a variety of changes to the way in which the optimizer chooses inks. We currently treat each color in the origi-

nal image as equally important, but we might instead measure the quality of a reproduction by weighting some colors more than others. For example, the system could use object recognition and texture analysis techniques of computer vision to automatically detect the colors most important to an image. Alternatively, the user could indicate to the optimizer which colors are most important by painting a weighting function over the original image; these weights would multiply each pixel's $L^*a^*b^*$ distance in the objective function. We could also obtain a more accurate estimate of color differences by using a color appearance space such as $RLab$ [Fairchild & Berns 93], $LLAB$ [Luo et al. 96], or Hunt's space [Hunt 91] instead of $L^*a^*b^*$.

With more terms in the objective function, we could try to minimize the cost of the materials, or reduce their environmental impact by favoring recycled papers and soy inks. We could maximize the longevity of the reproduction by favoring acid-free papers and fade-resistant inks, or reduce the impact of misregistration artifacts by favoring inks similar in color to the image subject matter.

As a more general extension, we might widen the set of variables over which we optimize to include items other than just the paper and inks. For instance, we could optimize the parameters controlling the gamut mapping and separation stages, and even the choice of which images to print. Or instead of selecting inks from a discrete set of possibilities, we could optimize over a set of coefficients determining the proportions in which to mix the primary inks to get completely new inks customized for an image. This approach changes the selection of inks from a combinatorial optimization problem to a continuous one, requiring the use of different numerical solution techniques.

8.3 Future directions

There are many ways in which the scope of our present work could be broadened. For instance, we could model metallic inks by including angular variation in the ink layering model, or capture the behavior of fluorescent inks and papers by treating reflectance and transmittance as functions of both incoming and outgoing wavelength. Both these possibilities pose challenges in terms of gathering the right data and formulating the ink layering model in a way that aids in determining separations.

Another direction for future work is the use of custom inks with printing devices other than offset presses. Many of the methods proposed in this dissertation apply equally well

to these other reproduction processes, including ink-jet, thermal wax transfer, dye sublimation, letterpress, and gravure printing. We could adapt our approach to these processes by substituting new gamut models for our current one; the gamut mapping, colorant selection, and separation calculations could in large part remain unchanged. Ink-jet printers are particularly appealing, because of their wide availability and the ease with which their ink cartridges are refilled or replaced. A number of recent articles have focused on modeling the gamuts achieved by the overlapping nonuniform dots produced by ink-jet printers [Daligault & Archinard 93, Emmel et al. 95, Emmel et al. 96]. The paucity of ink colors formulated for ink-jets may limit the migration of our techniques to these printers, but our separation algorithm may be of use even for process color inks.

As we mentioned earlier, well-chosen heuristics may help to speed up the selection of inks by eliminating poor combinations before any time is spent evaluating them. We already have some intuition for which inks will be helpful in reproducing a given image, but further experience and experimentation will be essential to find rules that approve good choices and reject bad ones. Going one step further than heuristics, it may be possible to train a neural network to perform ink selection.

In a broader context, we hope to see exciting new applications of custom-ink color reproduction emerge as the technology of digital printing presses improves. Recent changes include the development of machines capable of creating new printing plates directly from digital information (without the usual intermediate step of exposing film), presses that automatically exchange old printing plates for new ones, and presses that can alter an electrostatic image on the plate to customize each page as it is printed. The trend in printing is toward increasing automation and customizability, both of which are favorable for color reproduction with custom inks.

✦ Bibliography ✦

- [Adelson et al. 84] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, November/December 1984.
- [Arney et al. 95a] J. S. Arney, P. G. Engeldrum, and H. Zeng. An expanded Murray-Davies model of tone reproduction in halftone imaging. *Journal of Imaging Science and Technology*, 39(6):502–508, November–December 1995.
- [Arney et al. 95b] J. S. Arney, P. G. Engeldrum, and H. Zeng. A modified Murray-Davies model of halftone gray scales. In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1995.
- [Balasubramanian 95] R. Balasubramanian. A printer model for dot-on-dot halftone screens. In *Color Hard Copy and Graphic Arts IV*, volume 2413 of *Proceedings of the SPIE*, pages 356–364. SPIE, Bellingham, WA, 1995.
- [Berns 93] R. S. Berns. Spectral modeling of a dye diffusion thermal transfer printer. *Journal of Electronic Imaging*, 2(4):359–370, October 1993.
- [Boll 94] H. Boll. A color to colorant transformation for a seven ink process. In *Device-Independent Color Imaging*, volume 2170 of *Proceedings of the SPIE*, pages 108–118. SPIE, Bellingham, WA, 1994.
- [Burt & Adelson 83] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.
- [Burt 81] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16(1):20–51, May 1981.
- [Clapper & Yule 53] F. R. Clapper and J. A. C. Yule. The effect of multiple internal reflections on the densities of half-tone prints on paper. *Journal of the Optical Society of America*, 43(7):600–603, July 1953.
- [Daligault & Archinard 93] L. Daligault and P. Archinard. Predictive model for color ink jet printing. In *Color Hard Copy and the Graphic Arts II*, volume 1912 of *Proceedings of the SPIE*, pages 133–141. SPIE, Bellingham, WA, 1993.
- [Davis 91] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

- [Emmel et al. 95] P. Emmel, R. D. Hersch, and V. Ostromoukhov. A grid-based method for predicting the behaviour of colour printers. In *Proceedings of the IS&T/SID Color Imaging Conference: Color Science, Systems and Applications*, pages 71–77. Society for Imaging Science and Technology, Springfield, VA, 1995.
- [Emmel et al. 96] P. Emmel, I. Amidror, V. Ostromoukhov, and R. D. Hersch. Predicting the spectral behaviour of colour printers for transparent inks on transparent support. In *Proceedings of the IS&T/SID Color Imaging Conference: Color Science, Systems and Applications*, pages 86–91. Society for Imaging Science and Technology, Springfield, VA, 1996.
- [Fairchild & Berns 93] M. D. Fairchild and R. S. Berns. Image color-appearance specification through extension of CIELAB. *Color Research and Application*, 18(3):178–190, June 1993.
- [Foley et al. 90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, second edition, 1990.
- [Gentile et al. 90] R. S. Gentile, E. Walowit, and J. P. Allebach. A comparison of techniques for color gamut mismatch compensation. *Journal of Imaging Technology*, 16(5):176–181, October 1990.
- [Gervautz & Purgathofer 88] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. In N. Magnenat-Thalmann and D. Thalmann, editors, *New Trends in Computer Graphics: Proceedings of CG International '88*, pages 219–231. Springer-Verlag, Berlin, 1988.
- [Glassner 95] A. S. Glassner. *Principles of Digital Image Synthesis*, volume 1. Morgan Kaufmann, San Francisco, 1995.
- [Glassner et al. 95] A. S. Glassner, K. P. Fishkin, D. H. Marimont, and M. C. Stone. Device-directed rendering. *ACM Transactions on Graphics*, 14(1):58–76, January 1995.
- [Gustavson & Kruse 95] S. Gustavson and B. Kruse. Modelling optical dot gain by a 3-D model of light diffusion in paper. In *Third European Research Symposium on Image Analysis for Pulp and Paper Research and Production*. 1995.
- [Gustavson 95] S. Gustavson. Modelling of light scattering effects in print. Licentiate thesis, Linköping University, October 1995.
- [Hardy & Wurzburg 48] A. C. Hardy and F. L. Wurzburg, Jr. Color correction in color printing. *Journal of the Optical Society of America*, 38(4):300–307, April 1948.
- [Harrington & Klassen 93] S. J. Harrington and R. V. Klassen. Color printing having a highlight color image mapped from a full color image. U.S. Patent 5,237,517, August 17, 1993.

-
- [Heckbert 82] P. Heckbert. Color image quantization for frame buffer display. In *Proceedings of SIGGRAPH 82*, pages 297–307. ACM, New York, 1982.
- [Hering 64] E. Hering. *Outlines of a Theory of the Light Sense*. Harvard University Press, Cambridge, MA, 1964.
- [Herzog & Hill 95] P. G. Herzog and B. Hill. A new approach to the representation of color gamuts. In *Proceedings of the IS&T/SID Color Imaging Conference: Color Science, Systems and Applications*, pages 78–81. Society for Imaging Science and Technology, Springfield, VA, 1995.
- [Hunt 91] R. W. G. Hunt. Revised colour-appearance model for related and unrelated colours. *Color Research and Application*, 16(3):146–165, June 1991.
- [Hunt 95] R. W. G. Hunt. *The Reproduction of Colour*. Fountain Press, Kingston-upon-Thames, England, fifth edition, 1995.
- [Inui 93] M. Inui. Fast algorithm for computing color gamuts. *Color Research and Application*, 18(5):341–348, October 1993.
- [Iverson & Riskin 93] V. S. Iverson and E. A. Riskin. A fast method for combining palettes of color quantized images. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 1993.
- [Iwata & Marcu 94] K. Iwata and G. Marcu. Computer simulation of printed colors on textile materials. In *Color Hard Copy and Graphic Arts III*, volume 2171 of *Proceedings of the SPIE*, pages 228–238. SPIE, Bellingham, WA, 1994.
- [Johnson 96] T. Johnson. A complete colour reproduction model for graphic arts. In *Proceedings of the Technical Association of the Graphic Arts*, pages 1061–1076. Technical Association of the Graphic Arts, Rochester, NY, 1996.
- [Judd & Wyszecki 75] D. B. Judd and G. Wyszecki. *Color in Business, Science, and Industry*. John Wiley and Sons, New York, 1975.
- [Judd 79] D. B. Judd. *Contributions to Color Science*. National Bureau of Standards, Washington, 1979.
- [Kolpatzik & Bouman 95] B. W. Kolpatzik and C. A. Bouman. Optimized universal color palette design for error diffusion. *Journal of Electronic Imaging*, 4(2):131–143, April 1995.
- [Kortüm 69] G. Kortüm. *Reflectance Spectroscopy: Principles, Methods, Applications*, chapter 4, pages 103–169. Springer, New York, 1969.
- [Kress & Stevens 94] W. Kress and M. Stevens. Derivation of 3-dimensional gamut descriptors for graphic arts output devices. In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1994.

- [Kruse & Gustavson 95] B. Kruse and S. Gustavson. A new approach to dot gain modelling. In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1995.
- [Kruse & Gustavson 96] B. Kruse and S. Gustavson. Rendering of color on scattering media. In *Human Vision and Electronic Imaging*, volume 2657 of *Proceedings of the SPIE*, pages 422–431. SPIE, Bellingham, WA, 1996.
- [Laihanen 87] P. Laihanen. Colour reproduction theory based on the principles of colour science. In *Proceedings of the International Association of Research Institutes for the Graphic Arts Industry*, volume 19, pages 1–36. Pentech Press, London, 1987.
- [Lindbloom 89] B. J. Lindbloom. Accurate color reproduction for computer graphics applications. In *Proceedings of SIGGRAPH 89*, pages 117–126. ACM, New York, 1989.
- [Liu 91] Y. Liu. Spectral reflectance modification of Neugebauer equations. In *Proceedings of the Technical Association of the Graphic Arts*, pages 154–172. Technical Association of the Graphic Arts, Rochester, NY, 1991.
- [Luo et al. 96] M. R. Luo, M. C. Lo, and W. G. Kuo. The LLAB (l:c) colour model. *Color Research and Application*, 21(6):412–429, December 1996.
- [MacDonald 93] L. W. MacDonald. Gamut mapping in perceptual color space. In *Proceedings of the IS&T/SID Color Imaging Conference: Transforms & Transportability of Color*, pages 193–196. Society for Imaging Science and Technology, Springfield, VA, 1993.
- [Mahy & Delabastita 96] M. Mahy and P. Delabastita. Inversion of the Neugebauer equations. *Color Research and Application*, 21(6):401–411, December 1996.
- [Marcu & Abe 94a] G. Marcu and S. Abe. Color designing and simulation in non-conventional printing process. In *Applications of Digital Image Processing XVII*, volume 2298 of *Proceedings of the SPIE*, pages 216–223. SPIE, Bellingham, WA, 1994.
- [Marcu & Abe 94b] G. Marcu and S. Abe. Color separation for printing with non-standard inks. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 1002–1005. IEEE Compute Society, Los Alamitos, CA, 1994.
- [Montag & Fairchild 96] E. D. Montag and M. D. Fairchild. Simulated color gamut mapping using simple rendered images. In *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, volume 2658 of *Proceedings of the SPIE*, pages 316–325. SPIE, Bellingham, WA, 1996.
- [Montag & Fairchild 97] E. D. Montag and M. D. Fairchild. Psychophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries. *IEEE Transactions on Image Processing*, 6(7):977–989, July 1997.

-
- [Neugebauer 37] H. E. J. Neugebauer. Die Theoretischen Grundlagen des Mehrfarbendruckes (The theoretical foundation for multicolor printing). *Zeitschrift fuer Wissenschaftliche Photographie*, 36(4):73–89, 1937. Reprinted in [Sayanaï 90], pages 194–202.
- [Ostromoukhov & Hersch 95] V. Ostromoukhov and R. D. Hersch. Artistic screening. In *Proceedings of SIGGRAPH 95*, pages 219–228. ACM, New York, 1995.
- [Ostromoukhov 93] V. Ostromoukhov. Chromaticity gamut enhancement by heptatone multi-color printing. In *Device-Independent Color Imaging and Imaging Systems Integration*, volume 1909 of *Proceedings of the SPIE*, pages 139–151. SPIE, Bellingham, WA, 1993.
- [Otten & van Ginneken 89] R. H. J. M. Otten and L. P. P. P. van Ginneken. *The Annealing Algorithm*. Kluwer Academic Publishers, Boston, 1989.
- [Papadimitriou & Steiglitz 82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [Power et al. 96] J. L. Power, B. S. West, E. J. Stollnitz, and D. H. Salesin. Reproducing color images as duotones. In *Proceedings of SIGGRAPH 96*, pages 237–248. ACM, New York, 1996.
- [Press et al. 92] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Fetterling. *Numerical Recipes*. Cambridge University Press, New York, second edition, 1992.
- [Rhodes 90] W. Rhodes. Fifty years of the Neugebauer equations. In [Sayanaï 90], pages 7–18.
- [Rolleston & Balasubramanian 93] R. Rolleston and R. Balasubramanian. Accuracy of various types of Neugebauer model. In *Proceedings of the IS&T/SID Color Imaging Conference: Transforms & Transportability of Color*, pages 32–37. Society for Imaging Science and Technology, Springfield, VA, 1993.
- [Ruckdeschel & Hauser 78] F. R. Ruckdeschel and O. G. Hauser. Yule-Nielsen effect in printing: A physical analysis. *Applied Optics*, 17(21):3376–3383, November 1, 1978.
- [Saunderson 42] J. L. Saunderson. Calculation of the color of pigmented plastics. *Journal of the Optical Society of America*, 32(12):727–736, December 1942.
- [Sayanaï 90] K. Sayanaï, editor. *Neugebauer Memorial Seminar on Color Reproduction*, volume 1184 of *Proceedings of the SPIE*, Bellingham, WA, 1990. SPIE.
- [Schläpfer & Widmer 93] K. Schläpfer and E. Widmer. Which color gamut can be achieved in multicolor printing and in television? In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1993.

- [Sirén et al. 95] A. Sirén, T. Lehtonen, H. Juhola, and A. Karttunen. Experimenting with hifi-printing techniques. In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1995.
- [Southworth 95] M. Southworth. Printing color images with two non-process color inks, toners or paints. In *Proceedings of the Technical Association of the Graphic Arts*. Technical Association of the Graphic Arts, Rochester, NY, 1995.
- [Spaulding et al. 95] K. E. Spaulding, R. N. Ellson, and J. R. Sullivan. UltraColor: A new gamut mapping strategy. In *Device-independent color imaging II*, volume 2414 of *Proceedings of the SPIE*, pages 61–68. SPIE, Bellingham, WA, 1995.
- [Stollnitz et al. 98] E. J. Stollnitz, V. Ostromoukhov, and D. H. Salesin. Reproducing color images using custom inks. In *Proceedings of SIGGRAPH 98*. ACM, New York, 1998. To appear.
- [Stone et al. 88] M. C. Stone, W. B. Cowan, and J. C. Beatty. Color gamut mapping and the printing of digital color images. *ACM Transactions on Graphics*, 7(4):249–292, October 1988.
- [Takaghi et al. 94] A. Takaghi, T. Ozeki, Y. Ogata, and S. Minato. Faithful color printing for computer generated image syntheses with highly saturated component inks. In *Proceedings of the IS&T/SID Color Imaging Conference: Color Science, Systems and Applications*, pages 108–111. Society for Imaging Science and Technology, Springfield, VA, 1994.
- [Viggiano 90] J. A. S. Viggiano. Modeling the color of multi-colored halftones. In *Proceedings of the Technical Association of the Graphic Arts*, pages 44–62. Technical Association of the Graphic Arts, Rochester, NY, 1990.
- [von Kries 70] J. von Kries. Chromatic adaptation. In D. L. MacAdam, editor, *Sources of Color Science*, pages 109–119. MIT Press, Cambridge, MA, 1970.
- [Wolski et al. 94] M. Wolski, J. P. Allebach, and C. A. Bouman. Gamut mapping: Squeezing the most out of your color system. In *Proceedings of the IS&T/SID Color Imaging Conference: Color Science, Systems and Applications*, pages 89–92. Society for Imaging Science and Technology, Springfield, VA, 1994.
- [Wyszecki & Stiles 82] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, 1982.
- [Yule & Nielsen 51] J. A. C. Yule and W. J. Nielsen. The penetration of light into paper and its effect on halftone reproduction. In *Proceedings of the Technical Association of the Graphic Arts*, volume 3, pages 65–76. Technical Association of the Graphic Arts, Rochester, NY, 1951.
- [Yule 67] J. A. C. Yule. *Principles of Color Reproduction*. Wiley, New York, 1967.

✦ **Data for papers and inks** ✦

This appendix provides data on the papers and inks that we used in the printing experiments of Chapter 7. The spectral reflectance measurements of eight papers are presented graphically in Figure A-1 and numerically in Table A-1. Following these in Table A-2 are the dot gain parameters for the 16 PANTONE primaries that we measured. Next come the optimal Fresnel coefficients in Table A-3 and the optimal trapping fractions in Table A-4, for the same set of inks modeled using the variations labeled A, D, and H in Section 7.2. Finally, we present the spectra we obtained for each ink's transmittance and reflectance, graphically in Figure A-2 and numerically Table A-5.

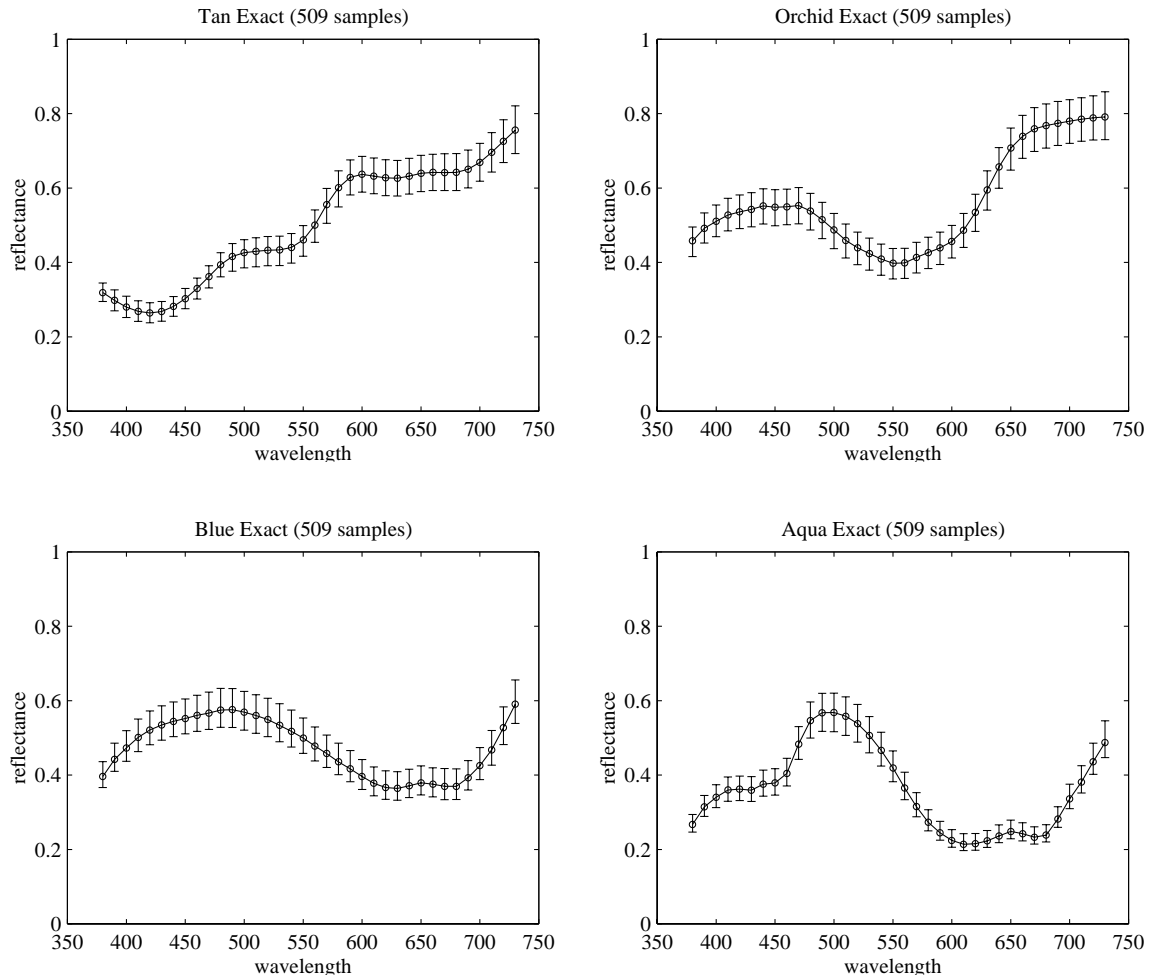


Figure A-1 Paper reflectance spectra. Data points indicate the median value at each wavelength, while error bars indicate the range of measurements. *(continued)*

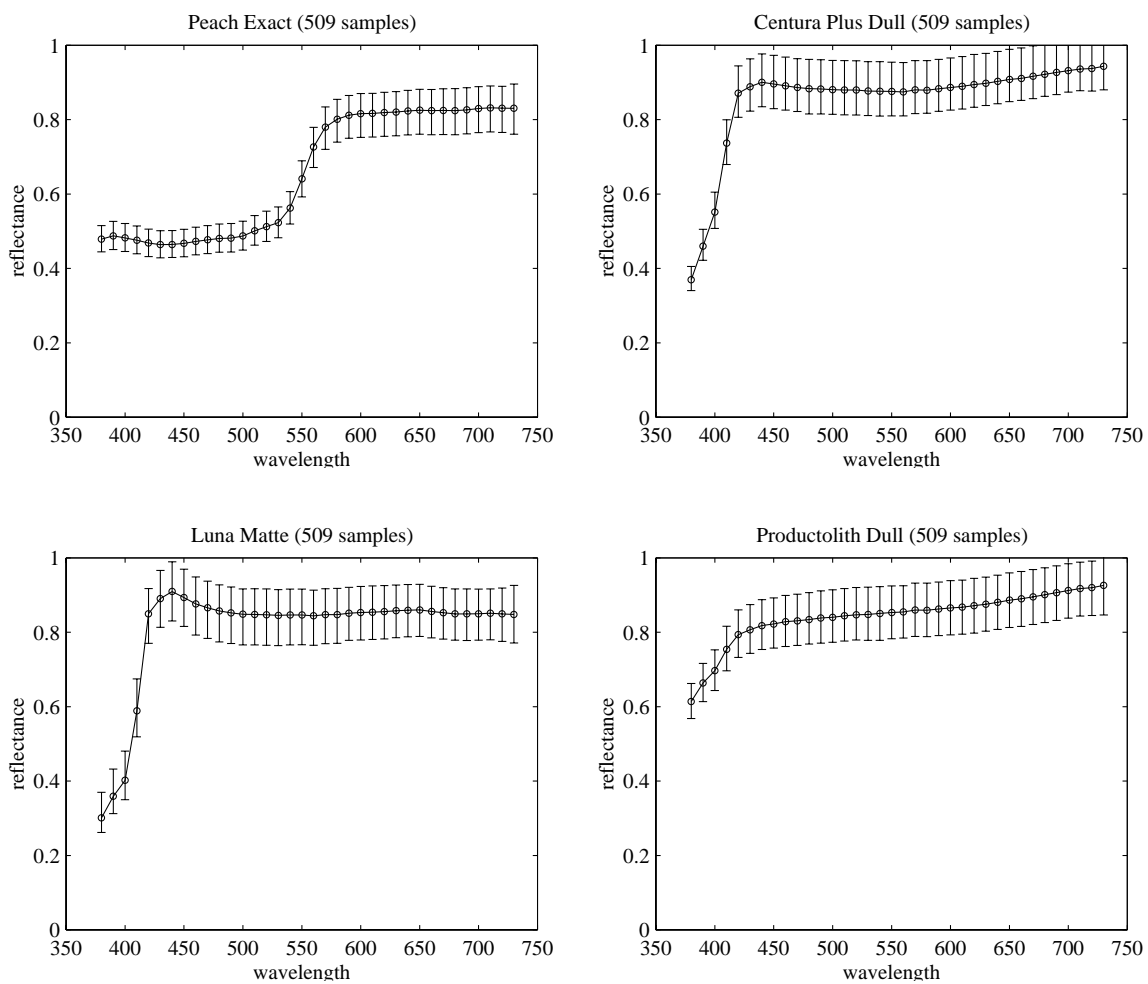


Figure A-1 (continued) Paper reflectance spectra. Data points indicate the median value at each wavelength, while error bars indicate the range of measurements.

Table A-1 Paper reflectance spectra, given by the median value at each wavelength, from 509 measurements of each paper.

λ (nm)	Tan Exact	Orchid Exact	Blue Exact	Aqua Exact	Peach Exact	Centura Plus Dull	Luna Matte	Productolith Dull
380	0.3227	0.4565	0.3968	0.2675	0.4787	0.3674	0.3011	0.6175
390	0.3010	0.4908	0.4424	0.3140	0.4878	0.4556	0.3584	0.6671
400	0.2824	0.5088	0.4741	0.3400	0.4822	0.5476	0.4015	0.7007
410	0.2709	0.5258	0.5022	0.3592	0.4757	0.7320	0.5831	0.7594
420	0.2667	0.5348	0.5216	0.3614	0.4683	0.8665	0.8456	0.7991
430	0.2705	0.5403	0.5351	0.3593	0.4640	0.8847	0.8881	0.8122
440	0.2836	0.5500	0.5458	0.3750	0.4647	0.8974	0.9076	0.8236
450	0.3057	0.5465	0.5535	0.3781	0.4677	0.8924	0.8915	0.8282
460	0.3336	0.5487	0.5620	0.4044	0.4730	0.8886	0.8748	0.8336
470	0.3653	0.5508	0.5683	0.4833	0.4772	0.8834	0.8661	0.8363
480	0.3977	0.5346	0.5760	0.5462	0.4810	0.8817	0.8576	0.8397
490	0.4201	0.5109	0.5769	0.5673	0.4818	0.8807	0.8527	0.8434
500	0.4296	0.4833	0.5700	0.5680	0.4882	0.8787	0.8487	0.8454
510	0.4331	0.4567	0.5611	0.5584	0.5027	0.8780	0.8488	0.8482
520	0.4361	0.4374	0.5510	0.5385	0.5143	0.8774	0.8483	0.8518
530	0.4371	0.4225	0.5356	0.5070	0.5252	0.8748	0.8468	0.8522
540	0.4434	0.4074	0.5199	0.4670	0.5667	0.8738	0.8483	0.8551
550	0.4642	0.3963	0.5015	0.4189	0.6455	0.8741	0.8482	0.8577
560	0.5039	0.3971	0.4802	0.3657	0.7278	0.8728	0.8458	0.8586
570	0.5602	0.4120	0.4592	0.3150	0.7800	0.8782	0.8491	0.8642
580	0.6056	0.4259	0.4378	0.2732	0.8013	0.8774	0.8497	0.8639
590	0.6340	0.4387	0.4191	0.2449	0.8119	0.8812	0.8524	0.8671
600	0.6417	0.4562	0.3996	0.2243	0.8168	0.8846	0.8537	0.8702
610	0.6376	0.4863	0.3808	0.2145	0.8176	0.8878	0.8551	0.8723
620	0.6329	0.5340	0.3693	0.2156	0.8200	0.8927	0.8568	0.8761
630	0.6323	0.5939	0.3664	0.2230	0.8222	0.8963	0.8597	0.8793
640	0.6377	0.6559	0.3738	0.2369	0.8248	0.9008	0.8620	0.8841
650	0.6455	0.7072	0.3818	0.2487	0.8280	0.9061	0.8625	0.8898
660	0.6473	0.7391	0.3790	0.2429	0.8271	0.9090	0.8586	0.8937
670	0.6476	0.7588	0.3733	0.2336	0.8271	0.9143	0.8547	0.8990
680	0.6473	0.7681	0.3739	0.2392	0.8272	0.9201	0.8517	0.9044
690	0.6552	0.7742	0.3953	0.2822	0.8294	0.9253	0.8517	0.9101
700	0.6735	0.7796	0.4280	0.3367	0.8334	0.9303	0.8527	0.9159
710	0.7005	0.7847	0.4713	0.3827	0.8365	0.9345	0.8544	0.9211
720	0.7320	0.7890	0.5315	0.4378	0.8355	0.9362	0.8507	0.9234
730	0.7632	0.7921	0.5951	0.4883	0.8369	0.9407	0.8519	0.9289

Table A-2 Dot gain parameters for 16 PANTONE inks.

PANTONE ink	γ
Yellow	0.5607
Yellow 012	0.6355
Orange 021	0.5261
Warm Red	0.6031
Red 032	0.4952
Rubine Red	0.4682
Rhodamine Red	0.6094
Purple	0.4680
Blue 072	0.5063
Reflex Blue	0.5346
Process Blue	0.5094
Green	0.5195
Black	0.4589
Process Magenta	0.4385
Process Cyan	0.5934
Process Black	0.4276

Table A-3 Fresnel reflection coefficients for model variations A, D, and H.

variation	ρ_{ai}	ρ_{ia}	ρ_{ip}	ρ_{pi}
A	0.0000	0.0000	0.0000	0.0000
D	0.0085	0.0790	0.0000	0.0000
H	0.0000	0.2511	0.0000	0.0000

Table A-4 Trapping fractions for model variations A, D, and H.

variation	t_{pi}	t_{pij}	t_{pijk}
A	1.0000	1.0000	1.0000
D	1.0000	0.9338	0.8808
H	0.9772	0.9169	0.8634

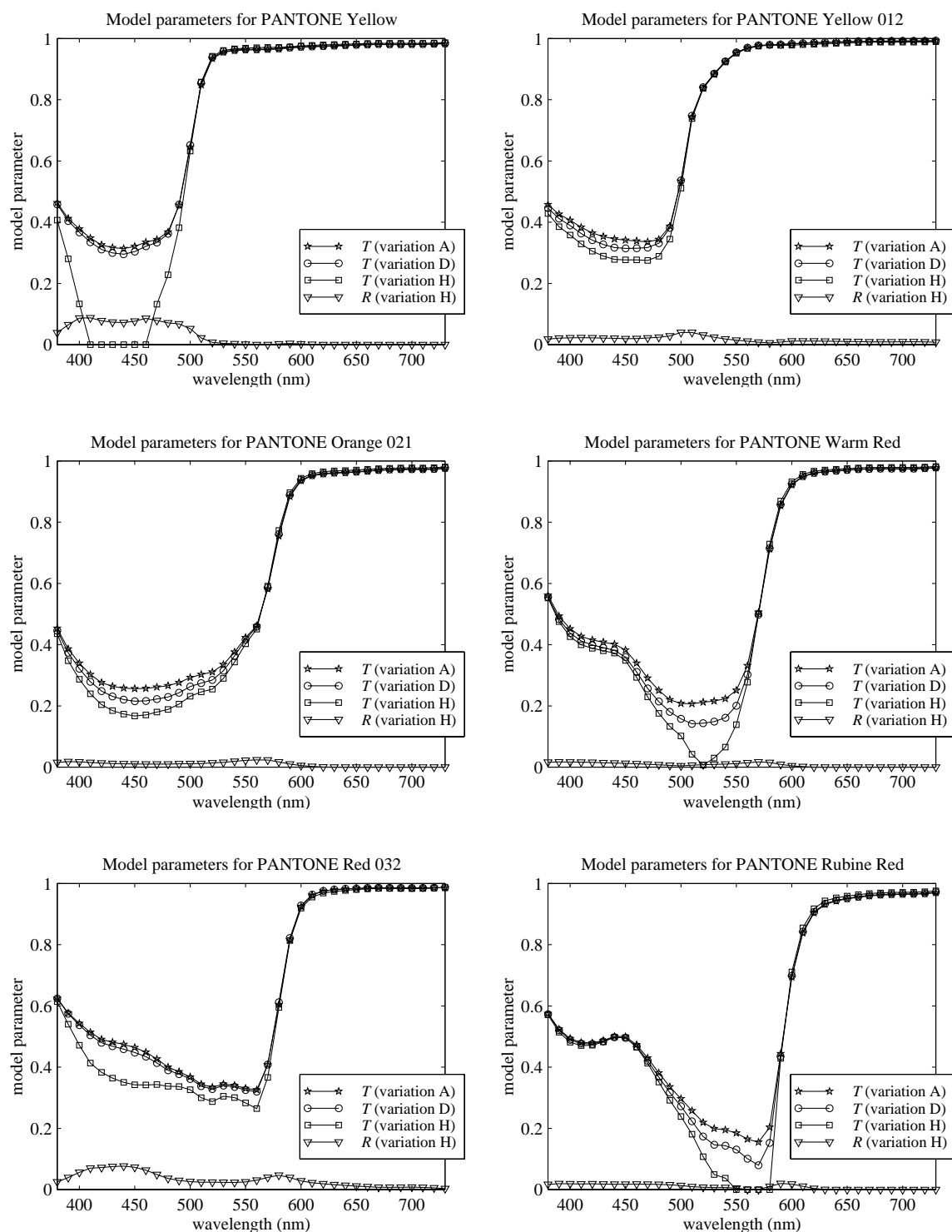


Figure A-2 Ink model parameters obtained by fitting model variations A, D, and H to measured samples of 16 PANTONE inks. *(continued)*

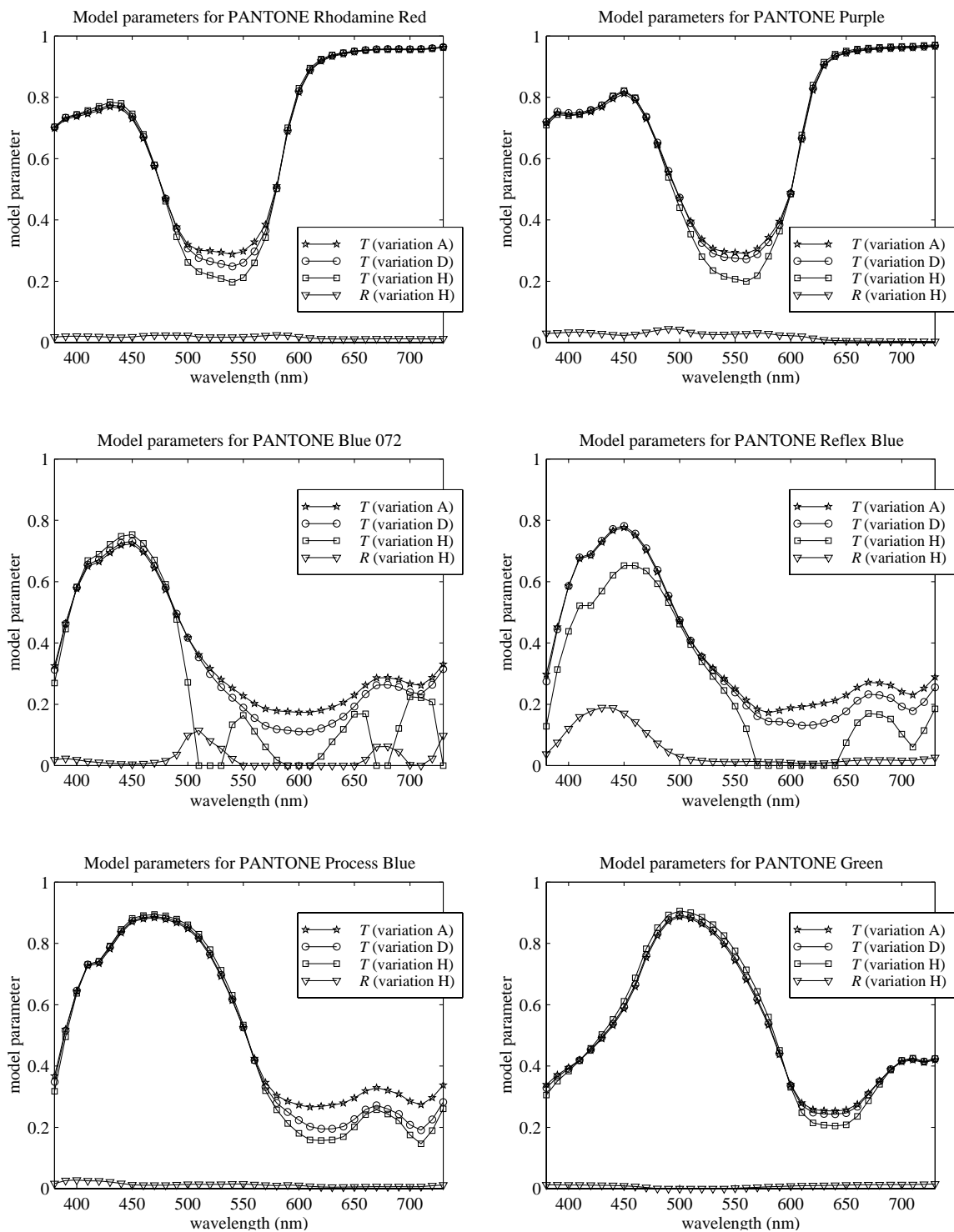


Figure A-2 (continued) Ink model parameters.

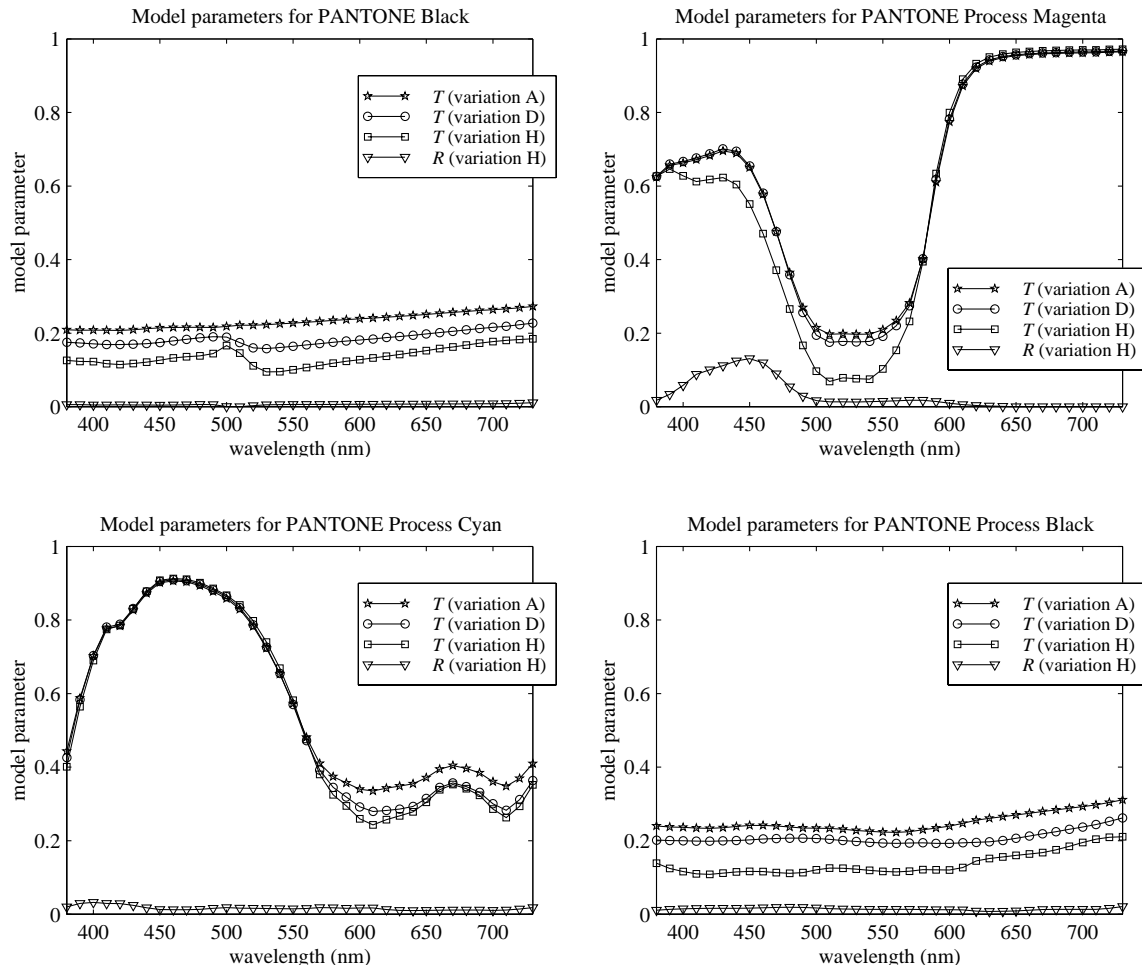


Figure A-2 (continued) Ink model parameters.

Table A-5 Ink model parameters obtained by fitting model variations A, D, and H to measured samples of 16 PANTONE inks. (continued)

PANTONE Yellow						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.4635	380	0.4583	380	0.4066	0.0388
390	0.4118	390	0.4042	390	0.2804	0.0646
400	0.3778	400	0.3672	400	0.1330	0.0865
410	0.3481	410	0.3350	410	0.0000	0.0876
420	0.3259	420	0.3109	420	0.0000	0.0782
430	0.3169	430	0.2997	430	0.0000	0.0734
440	0.3138	440	0.2953	440	0.0000	0.0719
450	0.3211	450	0.3041	450	0.0000	0.0766
460	0.3348	460	0.3217	460	0.0000	0.0859
470	0.3432	470	0.3335	470	0.1321	0.0792
480	0.3677	480	0.3624	480	0.2284	0.0709
490	0.4553	490	0.4568	490	0.3821	0.0675
500	0.6452	500	0.6526	500	0.6328	0.0522
510	0.8486	510	0.8548	510	0.8581	0.0219
520	0.9347	520	0.9379	520	0.9422	0.0076
530	0.9552	530	0.9575	530	0.9612	0.0043
540	0.9606	540	0.9626	540	0.9665	0.0028
550	0.9624	550	0.9644	550	0.9688	0.0015
560	0.9631	560	0.9650	560	0.9702	0.0001
570	0.9641	570	0.9659	570	0.9710	0.0001
580	0.9659	580	0.9676	580	0.9713	0.0022
590	0.9690	590	0.9707	590	0.9731	0.0037
600	0.9714	600	0.9729	600	0.9758	0.0024
610	0.9728	610	0.9743	610	0.9779	0.0008
620	0.9742	620	0.9756	620	0.9794	0.0000
630	0.9756	630	0.9768	630	0.9805	0.0000
640	0.9768	640	0.9780	640	0.9815	0.0000
650	0.9782	650	0.9793	650	0.9826	0.0000
660	0.9796	660	0.9807	660	0.9838	0.0000
670	0.9804	670	0.9814	670	0.9844	0.0000
680	0.9801	680	0.9812	680	0.9843	0.0000
690	0.9802	690	0.9813	690	0.9843	0.0000
700	0.9803	700	0.9814	700	0.9844	0.0000
710	0.9804	710	0.9815	710	0.9845	0.0000
720	0.9813	720	0.9824	720	0.9853	0.0000
730	0.9831	730	0.9841	730	0.9867	0.0000

Table A-5 (continued) Ink model parameters.

PANTONE Yellow 012						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.4576	380	0.4459	380	0.4285	0.0181
390	0.4263	390	0.4117	390	0.3852	0.0213
400	0.4064	400	0.3890	400	0.3578	0.0223
410	0.3835	410	0.3630	410	0.3291	0.0225
420	0.3644	420	0.3415	420	0.3052	0.0220
430	0.3534	430	0.3274	430	0.2895	0.0214
440	0.3450	440	0.3172	440	0.2783	0.0206
450	0.3415	450	0.3143	450	0.2772	0.0195
460	0.3385	460	0.3151	460	0.2770	0.0200
470	0.3359	470	0.3173	470	0.2750	0.0220
480	0.3439	480	0.3314	480	0.2888	0.0241
490	0.3868	490	0.3811	490	0.3445	0.0284
500	0.5339	500	0.5359	500	0.5116	0.0399
510	0.7439	510	0.7482	510	0.7386	0.0399
520	0.8374	520	0.8410	520	0.8375	0.0311
530	0.8824	530	0.8855	530	0.8840	0.0242
540	0.9234	540	0.9258	540	0.9242	0.0187
550	0.9532	550	0.9549	550	0.9525	0.0146
560	0.9688	560	0.9700	560	0.9675	0.0114
570	0.9762	570	0.9771	570	0.9756	0.0080
580	0.9793	580	0.9798	580	0.9789	0.0055
590	0.9805	590	0.9808	590	0.9779	0.0084
600	0.9825	600	0.9829	600	0.9782	0.0116
610	0.9844	610	0.9848	610	0.9799	0.0120
620	0.9860	620	0.9865	620	0.9817	0.0117
630	0.9871	630	0.9876	630	0.9830	0.0114
640	0.9883	640	0.9888	640	0.9842	0.0110
650	0.9894	650	0.9898	650	0.9854	0.0106
660	0.9907	660	0.9910	660	0.9869	0.0099
670	0.9912	670	0.9916	670	0.9878	0.0091
680	0.9913	680	0.9917	680	0.9880	0.0090
690	0.9916	690	0.9920	690	0.9881	0.0091
700	0.9919	700	0.9922	700	0.9881	0.0095
710	0.9918	710	0.9921	710	0.9880	0.0096
720	0.9922	720	0.9926	720	0.9888	0.0089
730	0.9924	730	0.9928	730	0.9895	0.0081

Table A-5 (continued) Ink model parameters.

PANTONE Orange 021						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.4538	380	0.4461	380	0.4355	0.0150
390	0.3860	390	0.3738	390	0.3476	0.0182
400	0.3397	400	0.3217	400	0.2872	0.0173
410	0.3024	410	0.2788	410	0.2394	0.0151
420	0.2762	420	0.2487	420	0.2041	0.0134
430	0.2635	430	0.2309	430	0.1846	0.0117
440	0.2582	440	0.2206	440	0.1731	0.0108
450	0.2563	450	0.2152	450	0.1672	0.0103
460	0.2572	460	0.2164	460	0.1704	0.0099
470	0.2614	470	0.2224	470	0.1798	0.0097
480	0.2661	480	0.2294	480	0.1887	0.0101
490	0.2762	490	0.2430	490	0.2061	0.0107
500	0.2925	500	0.2632	500	0.2314	0.0115
510	0.3028	510	0.2748	510	0.2456	0.0121
520	0.3105	520	0.2846	520	0.2549	0.0137
530	0.3351	530	0.3150	530	0.2897	0.0158
540	0.3760	540	0.3630	540	0.3434	0.0192
550	0.4233	550	0.4165	550	0.4031	0.0220
560	0.4618	560	0.4589	560	0.4507	0.0234
570	0.5827	570	0.5862	570	0.5915	0.0232
580	0.7548	580	0.7611	580	0.7732	0.0173
590	0.8841	590	0.8885	590	0.8969	0.0098
600	0.9354	600	0.9381	600	0.9437	0.0051
610	0.9515	610	0.9535	610	0.9589	0.0025
620	0.9575	620	0.9593	620	0.9652	0.0003
630	0.9603	630	0.9619	630	0.9676	0.0000
640	0.9619	640	0.9633	640	0.9689	0.0000
650	0.9637	650	0.9651	650	0.9705	0.0000
660	0.9666	660	0.9679	660	0.9728	0.0000
670	0.9692	670	0.9704	670	0.9749	0.0000
680	0.9707	680	0.9719	680	0.9762	0.0000
690	0.9720	690	0.9731	690	0.9772	0.0000
700	0.9723	700	0.9734	700	0.9775	0.0000
710	0.9721	710	0.9732	710	0.9774	0.0000
720	0.9730	720	0.9741	720	0.9782	0.0000
730	0.9757	730	0.9767	730	0.9804	0.0000

Table A-5 (continued) Ink model parameters.

PANTONE Warm Red						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.5614	380	0.5552	380	0.5524	0.0159
390	0.4936	390	0.4835	390	0.4752	0.0170
400	0.4516	400	0.4381	400	0.4263	0.0167
410	0.4272	410	0.4118	410	0.3997	0.0162
420	0.4152	420	0.3992	420	0.3881	0.0155
430	0.4079	430	0.3903	430	0.3803	0.0144
440	0.4014	440	0.3814	440	0.3726	0.0133
450	0.3825	450	0.3592	450	0.3483	0.0124
460	0.3394	460	0.3108	460	0.2933	0.0110
470	0.2908	470	0.2572	470	0.2298	0.0092
480	0.2505	480	0.2141	480	0.1759	0.0076
490	0.2203	490	0.1804	490	0.1333	0.0057
500	0.2076	500	0.1579	500	0.1017	0.0044
510	0.2071	510	0.1417	510	0.0419	0.0063
520	0.2120	520	0.1434	520	0.0066	0.0081
530	0.2157	530	0.1488	530	0.0300	0.0089
540	0.2237	540	0.1613	540	0.0653	0.0100
550	0.2510	550	0.2006	550	0.1390	0.0116
560	0.3317	560	0.3025	560	0.2774	0.0143
570	0.5057	570	0.4980	570	0.5020	0.0173
580	0.7120	580	0.7145	580	0.7288	0.0153
590	0.8539	590	0.8575	590	0.8692	0.0090
600	0.9217	600	0.9242	600	0.9321	0.0042
610	0.9490	610	0.9506	610	0.9566	0.0020
620	0.9599	620	0.9611	620	0.9668	0.0001
630	0.9646	630	0.9656	630	0.9708	0.0000
640	0.9674	640	0.9683	640	0.9731	0.0000
650	0.9701	650	0.9709	650	0.9753	0.0000
660	0.9724	660	0.9731	660	0.9773	0.0000
670	0.9739	670	0.9746	670	0.9785	0.0000
680	0.9742	680	0.9750	680	0.9789	0.0000
690	0.9743	690	0.9751	690	0.9790	0.0000
700	0.9744	700	0.9751	700	0.9791	0.0000
710	0.9743	710	0.9751	710	0.9790	0.0000
720	0.9754	720	0.9761	720	0.9799	0.0000
730	0.9779	730	0.9785	730	0.9819	0.0000

Table A-5 (continued) Ink model parameters.

PANTONE Red 032						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.6249	380	0.6246	380	0.6139	0.0249
390	0.5769	390	0.5743	390	0.5405	0.0394
400	0.5432	400	0.5372	400	0.4715	0.0562
410	0.5136	410	0.5044	410	0.4133	0.0701
420	0.4905	420	0.4797	420	0.3830	0.0718
430	0.4817	430	0.4684	430	0.3640	0.0751
440	0.4742	440	0.4590	440	0.3501	0.0767
450	0.4640	450	0.4481	450	0.3425	0.0729
460	0.4489	460	0.4344	460	0.3419	0.0635
470	0.4265	470	0.4140	470	0.3434	0.0490
480	0.3999	480	0.3895	480	0.3376	0.0365
490	0.3847	490	0.3772	490	0.3367	0.0299
500	0.3680	500	0.3618	500	0.3260	0.0260
510	0.3444	510	0.3381	510	0.3003	0.0240
520	0.3344	520	0.3282	520	0.2874	0.0243
530	0.3448	530	0.3401	530	0.3050	0.0237
540	0.3403	540	0.3357	540	0.3005	0.0232
550	0.3305	550	0.3254	550	0.2832	0.0253
560	0.3256	560	0.3203	560	0.2648	0.0307
570	0.4078	570	0.4082	570	0.3665	0.0396
580	0.6048	580	0.6119	580	0.5950	0.0469
590	0.8150	590	0.8214	590	0.8139	0.0394
600	0.9247	600	0.9276	600	0.9197	0.0283
610	0.9621	610	0.9634	610	0.9553	0.0225
620	0.9752	620	0.9760	620	0.9687	0.0187
630	0.9797	630	0.9803	630	0.9742	0.0160
640	0.9819	640	0.9824	640	0.9774	0.0136
650	0.9837	650	0.9841	650	0.9802	0.0115
660	0.9850	660	0.9854	660	0.9824	0.0093
670	0.9856	670	0.9861	670	0.9839	0.0078
680	0.9852	680	0.9856	680	0.9837	0.0075
690	0.9850	690	0.9855	690	0.9837	0.0074
700	0.9850	700	0.9855	700	0.9835	0.0078
710	0.9849	710	0.9854	710	0.9837	0.0073
720	0.9858	720	0.9863	720	0.9854	0.0056
730	0.9862	730	0.9866	730	0.9870	0.0032

Table A-5 (continued) Ink model parameters.

PANTONE Rubine Red						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.5748	380	0.5731	380	0.5708	0.0169
390	0.5240	390	0.5211	390	0.5141	0.0193
400	0.4935	400	0.4893	400	0.4814	0.0190
410	0.4807	410	0.4768	410	0.4709	0.0187
420	0.4790	420	0.4757	420	0.4718	0.0187
430	0.4872	430	0.4840	430	0.4824	0.0181
440	0.5007	440	0.4975	440	0.4984	0.0177
450	0.4995	450	0.4954	450	0.4967	0.0175
460	0.4730	460	0.4671	460	0.4650	0.0179
470	0.4299	470	0.4208	470	0.4127	0.0177
480	0.3811	480	0.3676	480	0.3511	0.0170
490	0.3356	490	0.3169	490	0.2917	0.0155
500	0.2970	500	0.2724	500	0.2386	0.0133
510	0.2574	510	0.2231	510	0.1807	0.0096
520	0.2194	520	0.1731	520	0.1075	0.0075
530	0.1991	530	0.1463	530	0.0494	0.0065
540	0.1953	540	0.1436	540	0.0376	0.0065
550	0.1848	550	0.1307	550	0.0000	0.0049
560	0.1650	560	0.1010	560	0.0000	0.0003
570	0.1549	570	0.0796	570	0.0000	0.0000
580	0.2035	580	0.1531	580	0.0000	0.0106
590	0.4441	590	0.4355	590	0.4291	0.0198
600	0.6943	600	0.6981	600	0.7113	0.0179
610	0.8382	610	0.8426	610	0.8549	0.0107
620	0.9041	620	0.9073	620	0.9178	0.0038
630	0.9311	630	0.9336	630	0.9431	0.0003
640	0.9429	640	0.9450	640	0.9532	0.0000
650	0.9497	650	0.9516	650	0.9590	0.0000
660	0.9551	660	0.9570	660	0.9635	0.0000
670	0.9594	670	0.9611	670	0.9671	0.0000
680	0.9620	680	0.9636	680	0.9693	0.0000
690	0.9637	690	0.9653	690	0.9708	0.0000
700	0.9643	700	0.9659	700	0.9712	0.0000
710	0.9648	710	0.9663	710	0.9716	0.0000
720	0.9665	720	0.9679	720	0.9730	0.0000
730	0.9699	730	0.9712	730	0.9759	0.0000

Table A-5 (continued) Ink model parameters.

PANTONE Rhodamine Red						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.6991	380	0.7021	380	0.7036	0.0180
390	0.7292	390	0.7334	390	0.7350	0.0202
400	0.7373	400	0.7418	400	0.7444	0.0207
410	0.7462	410	0.7514	410	0.7570	0.0202
420	0.7566	420	0.7624	420	0.7703	0.0193
430	0.7688	430	0.7747	430	0.7839	0.0177
440	0.7643	440	0.7701	440	0.7804	0.0172
450	0.7303	450	0.7361	450	0.7464	0.0184
460	0.6659	460	0.6713	460	0.6793	0.0216
470	0.5738	470	0.5778	470	0.5799	0.0237
480	0.4693	480	0.4697	480	0.4608	0.0238
490	0.3766	490	0.3720	490	0.3448	0.0237
500	0.3192	500	0.3074	500	0.2612	0.0231
510	0.3011	510	0.2760	510	0.2310	0.0182
520	0.2988	520	0.2649	520	0.2190	0.0176
530	0.2941	530	0.2568	530	0.2087	0.0173
540	0.2880	540	0.2487	540	0.1965	0.0175
550	0.2976	550	0.2607	550	0.2118	0.0185
560	0.3276	560	0.2974	560	0.2601	0.0199
570	0.3850	570	0.3646	570	0.3425	0.0222
580	0.5108	580	0.5038	580	0.5018	0.0244
590	0.6890	590	0.6912	590	0.7009	0.0231
600	0.8163	600	0.8203	600	0.8294	0.0185
610	0.8858	610	0.8893	610	0.8953	0.0146
620	0.9181	620	0.9209	620	0.9248	0.0124
630	0.9332	630	0.9356	630	0.9382	0.0117
640	0.9415	640	0.9437	640	0.9455	0.0114
650	0.9477	650	0.9497	650	0.9507	0.0115
660	0.9524	660	0.9543	660	0.9544	0.0118
670	0.9548	670	0.9566	670	0.9562	0.0122
680	0.9555	680	0.9573	680	0.9569	0.0123
690	0.9557	690	0.9575	690	0.9571	0.0123
700	0.9552	700	0.9571	700	0.9569	0.0122
710	0.9556	710	0.9574	710	0.9573	0.0120
720	0.9585	720	0.9602	720	0.9597	0.0121
730	0.9625	730	0.9642	730	0.9629	0.0123

Table A-5 (continued) Ink model parameters.

PANTONE Purple						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.7170	380	0.7203	380	0.7094	0.0296
390	0.7483	390	0.7529	390	0.7437	0.0314
400	0.7436	400	0.7487	400	0.7395	0.0336
410	0.7436	410	0.7497	410	0.7437	0.0340
420	0.7522	420	0.7589	420	0.7568	0.0319
430	0.7673	430	0.7741	430	0.7747	0.0290
440	0.7950	440	0.8016	440	0.8048	0.0250
450	0.8118	450	0.8182	450	0.8221	0.0232
460	0.7892	460	0.7961	460	0.7989	0.0263
470	0.7295	470	0.7370	470	0.7359	0.0331
480	0.6454	480	0.6527	480	0.6443	0.0397
490	0.5543	490	0.5599	490	0.5388	0.0445
500	0.4706	500	0.4724	500	0.4398	0.0426
510	0.3952	510	0.3901	510	0.3538	0.0328
520	0.3369	520	0.3253	520	0.2796	0.0273
530	0.3063	530	0.2905	530	0.2350	0.0255
540	0.2958	540	0.2785	540	0.2155	0.0262
550	0.2934	550	0.2756	550	0.2071	0.0277
560	0.2901	560	0.2716	560	0.1986	0.0285
570	0.3055	570	0.2885	570	0.2183	0.0313
580	0.3422	580	0.3279	580	0.2814	0.0288
590	0.3947	590	0.3842	590	0.3633	0.0236
600	0.4898	600	0.4865	600	0.4843	0.0224
610	0.6620	610	0.6663	610	0.6776	0.0205
620	0.8230	620	0.8285	620	0.8400	0.0139
630	0.9030	630	0.9069	630	0.9146	0.0087
640	0.9322	640	0.9352	640	0.9409	0.0065
650	0.9441	650	0.9467	650	0.9514	0.0057
660	0.9507	660	0.9530	660	0.9572	0.0053
670	0.9549	670	0.9571	670	0.9609	0.0049
680	0.9574	680	0.9595	680	0.9632	0.0046
690	0.9593	690	0.9613	690	0.9649	0.0043
700	0.9604	700	0.9624	700	0.9660	0.0041
710	0.9615	710	0.9634	710	0.9671	0.0040
720	0.9636	720	0.9655	720	0.9690	0.0036
730	0.9660	730	0.9678	730	0.9713	0.0031

Table A-5 (continued) Ink model parameters.

PANTONE Blue 072						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.3253	380	0.3126	380	0.2695	0.0188
390	0.4638	390	0.4624	390	0.4455	0.0230
400	0.5773	400	0.5809	400	0.5828	0.0191
410	0.6497	410	0.6560	410	0.6690	0.0134
420	0.6649	420	0.6721	420	0.6896	0.0102
430	0.6943	430	0.7019	430	0.7219	0.0073
440	0.7186	440	0.7264	440	0.7481	0.0051
450	0.7235	450	0.7313	450	0.7537	0.0042
460	0.6954	460	0.7033	460	0.7251	0.0062
470	0.6439	470	0.6517	470	0.6711	0.0095
480	0.5727	480	0.5795	480	0.5912	0.0157
490	0.4908	490	0.4952	490	0.4767	0.0360
500	0.4170	500	0.4170	500	0.2715	0.0982
510	0.3614	510	0.3531	510	0.0000	0.1147
520	0.3161	520	0.2986	520	0.0000	0.0798
530	0.2803	530	0.2556	530	0.0000	0.0552
540	0.2523	540	0.2212	540	0.1335	0.0221
550	0.2272	550	0.1892	550	0.1639	0.0000
560	0.2020	560	0.1547	560	0.1116	0.0000
570	0.1850	570	0.1296	570	0.0608	0.0000
580	0.1783	580	0.1185	580	0.0184	0.0000
590	0.1763	590	0.1150	590	0.0000	0.0000
600	0.1737	600	0.1106	600	0.0000	0.0000
610	0.1739	610	0.1110	610	0.0000	0.0000
620	0.1799	620	0.1210	620	0.0298	0.0000
630	0.1903	630	0.1375	630	0.0777	0.0000
640	0.2054	640	0.1596	640	0.1183	0.0000
650	0.2296	650	0.1926	650	0.1676	0.0000
660	0.2625	660	0.2341	660	0.1690	0.0193
670	0.2861	670	0.2625	670	0.0000	0.0616
680	0.2869	680	0.2635	680	0.0000	0.0624
690	0.2808	690	0.2565	690	0.1212	0.0449
700	0.2668	700	0.2397	700	0.2243	0.0022
710	0.2616	710	0.2336	710	0.2216	0.0000
720	0.2872	720	0.2642	720	0.2076	0.0231
730	0.3306	730	0.3148	730	0.0000	0.0976

Table A-5 (continued) Ink model parameters.

PANTONE Reflex Blue						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.2975	380	0.2741	380	0.1284	0.0376
390	0.4519	390	0.4447	390	0.3136	0.0755
400	0.5857	400	0.5858	400	0.4382	0.1202
410	0.6751	410	0.6788	410	0.5219	0.1588
420	0.6851	420	0.6896	420	0.5221	0.1775
430	0.7280	430	0.7332	430	0.5689	0.1889
440	0.7666	440	0.7720	440	0.6212	0.1881
450	0.7762	450	0.7819	450	0.6522	0.1702
460	0.7503	460	0.7565	460	0.6523	0.1420
470	0.7017	470	0.7082	470	0.6347	0.1066
480	0.6313	480	0.6376	480	0.5932	0.0724
490	0.5495	490	0.5546	490	0.5313	0.0455
500	0.4715	500	0.4746	500	0.4616	0.0284
510	0.4067	510	0.4071	510	0.3950	0.0201
520	0.3572	520	0.3548	520	0.3384	0.0163
530	0.3178	530	0.3126	530	0.2905	0.0138
540	0.2838	540	0.2758	540	0.2452	0.0127
550	0.2502	550	0.2386	550	0.1936	0.0125
560	0.2129	560	0.1959	560	0.1201	0.0131
570	0.1843	570	0.1614	570	0.0000	0.0137
580	0.1722	580	0.1437	580	0.0000	0.0117
590	0.1799	590	0.1437	590	0.0000	0.0122
600	0.1880	600	0.1387	600	0.0000	0.0095
610	0.1918	610	0.1306	610	0.0000	0.0066
620	0.1973	620	0.1321	620	0.0000	0.0067
630	0.2035	630	0.1391	630	0.0000	0.0082
640	0.2122	640	0.1518	640	0.0000	0.0114
650	0.2296	650	0.1771	650	0.0741	0.0148
660	0.2549	660	0.2117	660	0.1398	0.0168
670	0.2711	670	0.2327	670	0.1700	0.0186
680	0.2692	680	0.2303	680	0.1666	0.0185
690	0.2617	690	0.2207	690	0.1519	0.0180
700	0.2406	700	0.1925	700	0.1027	0.0166
710	0.2303	710	0.1780	710	0.0602	0.0170
720	0.2517	720	0.2073	720	0.1148	0.0209
730	0.2887	730	0.2550	730	0.1850	0.0261

Table A-5 (continued) Ink model parameters.

PANTONE Process Blue						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.3673	380	0.3480	380	0.3175	0.0164
390	0.5194	390	0.5151	390	0.4953	0.0266
400	0.6440	400	0.6459	400	0.6375	0.0281
410	0.7268	410	0.7314	410	0.7312	0.0254
420	0.7346	420	0.7399	420	0.7420	0.0253
430	0.7819	430	0.7873	430	0.7917	0.0219
440	0.8345	440	0.8395	440	0.8456	0.0169
450	0.8705	450	0.8748	450	0.8815	0.0123
460	0.8803	460	0.8845	460	0.8911	0.0116
470	0.8833	470	0.8877	470	0.8944	0.0113
480	0.8784	480	0.8831	480	0.8902	0.0115
490	0.8669	490	0.8720	490	0.8795	0.0126
500	0.8474	500	0.8528	500	0.8609	0.0143
510	0.8134	510	0.8190	510	0.8299	0.0140
520	0.7598	520	0.7654	520	0.7797	0.0135
530	0.6920	530	0.6971	530	0.7128	0.0140
540	0.6139	540	0.6174	540	0.6313	0.0152
550	0.5252	550	0.5254	550	0.5342	0.0152
560	0.4247	560	0.4188	560	0.4187	0.0131
570	0.3463	570	0.3321	570	0.3195	0.0115
580	0.3034	580	0.2794	580	0.2571	0.0100
590	0.2850	590	0.2499	590	0.2125	0.0121
600	0.2729	600	0.2237	600	0.1802	0.0106
610	0.2661	610	0.2020	610	0.1590	0.0077
620	0.2686	620	0.1952	620	0.1574	0.0055
630	0.2725	630	0.1950	630	0.1591	0.0049
640	0.2791	640	0.2026	640	0.1695	0.0051
650	0.2957	650	0.2263	650	0.2011	0.0056
660	0.3187	660	0.2582	660	0.2412	0.0062
670	0.3289	670	0.2716	670	0.2577	0.0065
680	0.3205	680	0.2600	680	0.2435	0.0063
690	0.3086	690	0.2432	690	0.2215	0.0064
700	0.2851	700	0.2085	700	0.1754	0.0059
710	0.2738	710	0.1907	710	0.1462	0.0066
720	0.2966	720	0.2254	720	0.1895	0.0093
730	0.3376	730	0.2823	730	0.2603	0.0119

Table A-5 (continued) Ink model parameters.

PANTONE Green						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.3391	380	0.3280	380	0.3053	0.0124
390	0.3711	390	0.3644	390	0.3509	0.0123
400	0.3948	400	0.3909	400	0.3835	0.0119
410	0.4196	410	0.4189	410	0.4175	0.0118
420	0.4513	420	0.4533	420	0.4580	0.0115
430	0.4891	430	0.4928	430	0.5027	0.0108
440	0.5324	440	0.5377	440	0.5520	0.0102
450	0.5863	450	0.5931	450	0.6114	0.0091
460	0.6586	460	0.6664	460	0.6877	0.0072
470	0.7534	470	0.7611	470	0.7828	0.0038
480	0.8250	480	0.8316	480	0.8512	0.0005
490	0.8718	490	0.8772	490	0.8929	0.0000
500	0.8867	500	0.8916	500	0.9059	0.0000
510	0.8805	510	0.8857	510	0.9006	0.0000
520	0.8630	520	0.8687	520	0.8854	0.0000
530	0.8356	530	0.8421	530	0.8613	0.0000
540	0.7961	540	0.8035	540	0.8257	0.0000
550	0.7434	550	0.7515	550	0.7760	0.0012
560	0.6807	560	0.6890	560	0.7142	0.0029
570	0.6115	570	0.6194	570	0.6436	0.0045
580	0.5331	580	0.5394	580	0.5596	0.0059
590	0.4375	590	0.4405	590	0.4513	0.0073
600	0.3393	600	0.3366	600	0.3306	0.0083
610	0.2792	610	0.2714	610	0.2476	0.0092
620	0.2581	620	0.2482	620	0.2146	0.0100
630	0.2543	630	0.2441	630	0.2075	0.0105
640	0.2533	640	0.2431	640	0.2047	0.0110
650	0.2566	650	0.2468	650	0.2083	0.0118
660	0.2753	660	0.2677	660	0.2356	0.0125
670	0.3112	670	0.3071	670	0.2868	0.0128
680	0.3513	680	0.3503	680	0.3404	0.0129
690	0.3882	690	0.3896	690	0.3872	0.0131
700	0.4138	700	0.4167	700	0.4187	0.0131
710	0.4194	710	0.4227	710	0.4254	0.0134
720	0.4115	720	0.4144	720	0.4150	0.0140
730	0.4200	730	0.4234	730	0.4243	0.0152

Table A-5 (continued) Ink model parameters.

PANTONE Black						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.2095	380	0.1752	380	0.1262	0.0058
390	0.2077	390	0.1733	390	0.1231	0.0056
400	0.2085	400	0.1711	400	0.1227	0.0049
410	0.2081	410	0.1691	410	0.1170	0.0049
420	0.2074	420	0.1692	420	0.1147	0.0051
430	0.2093	430	0.1698	430	0.1175	0.0047
440	0.2119	440	0.1716	440	0.1212	0.0045
450	0.2145	450	0.1747	450	0.1265	0.0045
460	0.2156	460	0.1789	460	0.1325	0.0046
470	0.2159	470	0.1833	470	0.1358	0.0054
480	0.2158	480	0.1876	480	0.1381	0.0063
490	0.2157	490	0.1903	490	0.1445	0.0058
500	0.2184	500	0.1892	500	0.1663	0.0003
510	0.2215	510	0.1741	510	0.1460	0.0000
520	0.2216	520	0.1599	520	0.1110	0.0027
530	0.2230	530	0.1583	530	0.0945	0.0050
540	0.2251	540	0.1608	540	0.0953	0.0057
550	0.2275	550	0.1644	550	0.1004	0.0060
560	0.2298	560	0.1680	560	0.1065	0.0061
570	0.2322	570	0.1716	570	0.1123	0.0063
580	0.2347	580	0.1752	580	0.1184	0.0064
590	0.2369	590	0.1784	590	0.1235	0.0065
600	0.2389	600	0.1813	600	0.1279	0.0066
610	0.2411	610	0.1844	610	0.1324	0.0068
620	0.2435	620	0.1878	620	0.1371	0.0070
630	0.2458	630	0.1910	630	0.1422	0.0071
640	0.2482	640	0.1942	640	0.1471	0.0072
650	0.2509	650	0.1979	650	0.1526	0.0072
660	0.2538	660	0.2016	660	0.1580	0.0074
670	0.2565	670	0.2052	670	0.1626	0.0077
680	0.2590	680	0.2087	680	0.1677	0.0077
690	0.2617	690	0.2125	690	0.1729	0.0079
700	0.2640	700	0.2158	700	0.1771	0.0081
710	0.2661	710	0.2190	710	0.1804	0.0085
720	0.2692	720	0.2229	720	0.1826	0.0097
730	0.2728	730	0.2271	730	0.1848	0.0111

Table A-5 (continued) Ink model parameters.

PANTONE Process Magenta						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.6239	380	0.6263	380	0.6270	0.0176
390	0.6560	390	0.6597	390	0.6462	0.0334
400	0.6629	400	0.6669	400	0.6280	0.0584
410	0.6717	410	0.6764	410	0.6122	0.0882
420	0.6826	420	0.6882	420	0.6181	0.1003
430	0.6956	430	0.7011	430	0.6230	0.1116
440	0.6893	440	0.6948	440	0.6042	0.1246
450	0.6502	450	0.6551	450	0.5509	0.1307
460	0.5773	460	0.5809	460	0.4706	0.1195
470	0.4760	470	0.4762	470	0.3713	0.0895
480	0.3644	480	0.3589	480	0.2656	0.0541
490	0.2695	490	0.2563	490	0.1666	0.0289
500	0.2144	500	0.1946	500	0.0965	0.0168
510	0.1969	510	0.1750	510	0.0686	0.0134
520	0.1982	520	0.1775	520	0.0786	0.0130
530	0.1969	530	0.1763	530	0.0765	0.0129
540	0.1976	540	0.1775	540	0.0747	0.0136
550	0.2091	550	0.1914	550	0.1027	0.0147
560	0.2339	560	0.2202	560	0.1535	0.0159
570	0.2818	570	0.2740	570	0.2319	0.0173
580	0.4015	580	0.4026	580	0.3945	0.0174
590	0.6102	590	0.6182	590	0.6340	0.0144
600	0.7744	600	0.7824	600	0.7998	0.0096
610	0.8720	610	0.8777	610	0.8906	0.0055
620	0.9196	620	0.9236	620	0.9327	0.0031
630	0.9400	630	0.9431	630	0.9505	0.0019
640	0.9495	640	0.9522	640	0.9590	0.0008
650	0.9545	650	0.9570	650	0.9636	0.0000
660	0.9576	660	0.9599	660	0.9661	0.0000
670	0.9595	670	0.9618	670	0.9677	0.0000
680	0.9608	680	0.9630	680	0.9688	0.0000
690	0.9617	690	0.9638	690	0.9695	0.0000
700	0.9622	700	0.9644	700	0.9700	0.0000
710	0.9626	710	0.9648	710	0.9704	0.0000
720	0.9640	720	0.9660	720	0.9714	0.0000
730	0.9648	730	0.9668	730	0.9721	0.0000

Table A-5 (continued) Ink model parameters.

PANTONE Process Cyan						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.4434	380	0.4261	380	0.4012	0.0200
390	0.5875	390	0.5833	390	0.5642	0.0295
400	0.7021	400	0.7033	400	0.6900	0.0322
410	0.7771	410	0.7806	410	0.7745	0.0293
420	0.7843	420	0.7885	420	0.7850	0.0287
430	0.8266	430	0.8309	430	0.8305	0.0240
440	0.8725	440	0.8763	440	0.8785	0.0175
450	0.9016	450	0.9048	450	0.9081	0.0129
460	0.9062	460	0.9095	460	0.9129	0.0122
470	0.9037	470	0.9073	470	0.9109	0.0122
480	0.8935	480	0.8977	480	0.9016	0.0134
490	0.8774	490	0.8822	490	0.8857	0.0160
500	0.8585	500	0.8636	500	0.8674	0.0180
510	0.8291	510	0.8336	510	0.8410	0.0167
520	0.7828	520	0.7868	520	0.7979	0.0160
530	0.7233	530	0.7264	530	0.7402	0.0155
540	0.6531	540	0.6543	540	0.6690	0.0150
550	0.5725	550	0.5699	550	0.5821	0.0149
560	0.4810	560	0.4717	560	0.4752	0.0156
570	0.4101	570	0.3919	570	0.3803	0.0180
580	0.3740	580	0.3454	580	0.3248	0.0175
590	0.3571	590	0.3184	590	0.2946	0.0162
600	0.3401	600	0.2921	600	0.2596	0.0170
610	0.3351	610	0.2794	610	0.2431	0.0170
620	0.3426	620	0.2824	620	0.2573	0.0130
630	0.3483	630	0.2860	630	0.2676	0.0106
640	0.3547	640	0.2933	640	0.2782	0.0100
650	0.3713	650	0.3149	650	0.3045	0.0102
660	0.3945	660	0.3445	660	0.3387	0.0110
670	0.4045	670	0.3572	670	0.3528	0.0116
680	0.3965	680	0.3470	680	0.3415	0.0113
690	0.3844	690	0.3317	690	0.3234	0.0112
700	0.3606	700	0.3000	700	0.2862	0.0105
710	0.3479	710	0.2828	710	0.2630	0.0113
720	0.3695	720	0.3117	720	0.2935	0.0143
730	0.4098	730	0.3637	730	0.3516	0.0179

Table A-5 (continued) Ink model parameters.

PANTONE Process Black						
variation A		variation D		variation H		
λ	T	λ	T	λ	T	R
380	0.2400	380	0.2018	380	0.1386	0.0111
390	0.2372	390	0.2002	390	0.1246	0.0132
400	0.2363	400	0.1998	400	0.1158	0.0146
410	0.2346	410	0.1987	410	0.1100	0.0156
420	0.2332	420	0.1983	420	0.1084	0.0160
430	0.2353	430	0.1993	430	0.1118	0.0160
440	0.2385	440	0.2005	440	0.1149	0.0160
450	0.2413	450	0.2022	450	0.1165	0.0165
460	0.2416	460	0.2045	460	0.1162	0.0175
470	0.2399	470	0.2060	470	0.1134	0.0185
480	0.2371	480	0.2068	480	0.1113	0.0191
490	0.2346	490	0.2067	490	0.1132	0.0187
500	0.2338	500	0.2061	500	0.1206	0.0171
510	0.2333	510	0.2034	510	0.1254	0.0152
520	0.2309	520	0.2004	520	0.1250	0.0142
530	0.2279	530	0.1977	530	0.1227	0.0136
540	0.2254	540	0.1954	540	0.1192	0.0134
550	0.2236	550	0.1939	550	0.1165	0.0133
560	0.2226	560	0.1929	560	0.1152	0.0132
570	0.2245	570	0.1934	570	0.1168	0.0131
580	0.2295	580	0.1939	580	0.1213	0.0126
590	0.2344	590	0.1927	590	0.1208	0.0123
600	0.2397	600	0.1926	600	0.1204	0.0125
610	0.2473	610	0.1941	610	0.1268	0.0120
620	0.2555	620	0.1958	620	0.1447	0.0086
630	0.2609	630	0.1971	630	0.1517	0.0074
640	0.2652	640	0.2008	640	0.1558	0.0079
650	0.2698	650	0.2064	650	0.1602	0.0092
660	0.2748	660	0.2126	660	0.1636	0.0110
670	0.2794	670	0.2188	670	0.1678	0.0127
680	0.2835	680	0.2244	680	0.1753	0.0131
690	0.2883	690	0.2310	690	0.1845	0.0134
700	0.2927	700	0.2371	700	0.1949	0.0131
710	0.2976	710	0.2438	710	0.2039	0.0134
720	0.3042	720	0.2526	720	0.2093	0.0161
730	0.3112	730	0.2617	730	0.2107	0.0209

✦ **Vita** ✦

Eric Stollnitz was born in 1970 in Ithaca, New York. His participation in Cornell University's child psychology studies at the impressionable age of two might have led to his current disposition toward scientific research. After attending elementary, junior high, and high school in Silver Spring, Maryland, Eric pursued a liberal arts education at Swarthmore College in Pennsylvania, where he studied physics and computer science. A few months in a research program at Los Alamos National Laboratory convinced him that there are far more idyllic settings than the east coast, prompting a move to Seattle in 1992. Six years later, having published a number of articles as well as a book, Eric emerged from the cocoon of academia to look for a job.