Using the general principle that stronger visual effects should be used to show greater quantities (G1.3), we can establish a guideline for the use of saturation in color coding. Because there are few discriminable steps in saturation, and because of contrast effect that may occur if the background is variable, only a few saturation levels can be reliably judged.

[G4.7] If using color saturation to encode numerical quantity, use greater saturation to represent greater numerical quantities. Avoid using a saturation sequence to encode more than three values.

Brown

Brown is one of the most mysterious colors. Brown is dark yellow. Whereas people talk about a light green or a dark green, a light blue or a dark blue, they do not talk about dark yellow. When colors in the vicinity of yellow and orange yellow are darkened, they turn to shades of brown and olive green. Unlike red, blue, and green, brown requires that there be a reference white somewhere in the vicinity for it to be perceived. Brown appears qualitatively different from orange yellow.

There is no such thing as an isolated brown light in a dark room, but when a yellow or yellowish orange is presented with a bright white surround, brown appears. The relevance to visualization is that, if color sets are being devised for the purposes of color coding—for example, a set of blues, a set of reds, a set of greens, and a set of yellows—in the case of yellows, brown may not be recognized as a set member.

Applications of Color in Visualization

So far, this chapter has been mainly a presentation of the basic theory underlying color vision and color measurement. Now we shift the emphasis to applications of color, for which new theory will be introduced only as needed. We will examine four different application areas: color selection interfaces, color labeling, color sequences for map coding, and color reproduction. Each of these presents a different set of problems, and each benefits from an analysis in terms of the human perception of color. We will use these applications to develop guidelines and continue to develop theory.

Application 1: Color Specification Interfaces and Color Spaces

In data visualization software, drawing applications, and CAD systems, it is often essential to let users choose their own colors. There are a number of approaches to this user interface problem. The user can be given a set of controls to specify a point in a three-dimensional color space, a set of color names to choose from, or a palette of predefined color samples.

Color Spaces

The simplest color interface to implement on a computer involves giving someone controls to adjust the amounts of red, green, and blue light that combine to make a patch of color on a monitor. The controls can take the form of sliders, or the user can simply type in three numbers. This provides access, in a straightforward way, to any point within the RGB color cube shown in Figure 4.5; however, although it is simple, many people find this kind of control confusing. For example, most people do not know that to get yellow you must add red and green. There have been many attempts to make color interfaces easier to use.

Many of the most widely used color interfaces in computer graphics are based on the hue, saturation, and value (HSV) color space (Smith, 1978). This is a simple transformation from HSV coordinates to *RGB* monitor coordinates. *Hue*, in Smith's scheme, represents an approximation to the visible spectrum by interpolating in sequence from red to yellow (= red + green) to green to cyan (= green + blue) to blue to purple (= blue + red) and back to red. *Saturation* is the distance from neutral monitor values, on the white–gray–black axis, to the purest hue possible given the limits of monitor primaries. Figure 4.17 shows how hue and saturation can be laid out in two dimensions, with hue on one axis and saturation on the other, based on the HSV transformation of monitor primaries. As Figure 4.16(b) shows, HSV creates only the crudest approximation to perceptually equal saturation contours. *Value* is the name given to the black–white axis. Some color specification interfaces based on HSV allow the user to control hue, saturation, and value variables with three sliders.

Because color research has shown the luminance channel to be very different from the chromatic (red–green, yellow–blue) channels, it is a good idea to separate a luminance

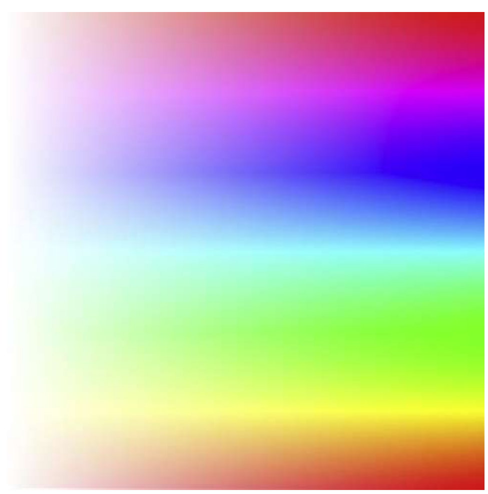

Figure 4.17 This plot shows hue and saturation, based on Smith's (1978) transformation of the monitor primaries.

(or lightness) dimension from the chromatic dimensions in a color specification interface. In addition, because the chromatic channels are perceived integrally, it is usually best to lay out the various hue and saturation choices on a plane, but not as shown in Figure 4.17, as this devotes far too much space to neutral colors and does not reflect the perceptual structure of color space derived from the color opponent channels. Figure 4.18 provides a selection of much better layouts. All are compromises among the constraints of colors produced by computer monitors, the desire to produce a neat geometric space, and the goal of producing a perceptually meaningful representation of a color plane orthogonal to the luminance channel.

> **[G4.8]** In an interface for specifying colors, consider laying out the red–green and yellow–blue channel information on a plane. Use a separate control for specifying the dark–light dimension.

A common interface method is to provide a single slider control for the black–white dimension and to lay out the two opponent color dimensions on a chromatic plane. The idea of laying out colors on a plane has a long history; for example, a color circle is a feature of a color textbook created for artists by Rood (1897). With the invention of computer graphics, it has become far simpler to create and control colors, and many ways of laying out colors are now available.

Figure 4.18(a) shows a color circle with red, green, yellow, and blue defining opposing axes. Many such color circles have been devised over the past century. They differ mainly in the spacing of colors around the periphery.

Figure 4.18(b) shows a color triangle with the monitor primaries, red, green, and blue, at the corners. This color layout is convenient because it has the property that mixtures of two colors will lie on a line between them (assuming proper calibration); however,

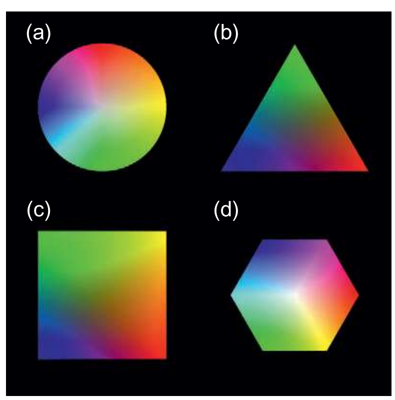

Figure 4.18 A sampling of four different geometric color layouts, each of them embodying the idea of a chromatic plane. (a) Circle. (b) Triangle. (c) Square. (d) Hexagon.

because of linear interpolation, only a very weak yellow occurs between the red and green corners (50% red, 50% green). The strongest yellow on a monitor comes from having both red and yellow at full strength.

Figure 4.18(c) shows a color square with the opponent color primaries, red–green and yellow–blue, at opposite corners (Ware & Cowan, 1990).

Figure 4.18(d) shows a color hexagon with the colors red, yellow, green, cyan, blue, and magenta at the corners. This represents a plane through the single-hexcone color model (Smith, 1978). The hexagon representation has the advantage that it gives both the monitor primaries (red, green, and blue) and the print primaries (cyan, magenta, and yellow) prominent positions around the circumference.

To create a color interface using one of these color planes, it is necessary to allow the user to pick a sample from the color plane and adjust its lightness with a luminance slider or some other control. In some interfaces, when the luminance slider is moved, the entire plane of colors becomes lighter and darker according to the currently selected level. For those interested in implementing color interfaces, algorithms for a number of color geometries can be found in Foley et al. (1990).

Another valuable addition to a color design interface is a method for showing a color sample on differently colored backgrounds. This allows the designer to understand how contrast effects can affect the appearance of particular color samples.

[G4.9] In an interface for designing visualization color schemes, consider providing a method for showing colors against different backgrounds.

The problem of the best color selection interface is by no means resolved. Experimental studies have failed to show that one way of controlling color is substantially better than another (Schwarz et al., 1987; Douglas & Kirkpatrick, 1996). Douglas and Kirkpatrick, however, have provided evidence that good feedback about the location of the color being adjusted in color space can help in the process.

Color Naming Systems

The facts that there are so few widely agreed upon color names and that color memory is so poor suggest that choosing colors by name will not be useful except for the simplest applications. People agree on red, green, yellow, blue, black, and white as labels, but not much more; nevertheless, it is possible to remember a rather large number of color names and use them accurately under controlled conditions. Displays in paint stores generally have a standard illuminant and standard background for sample strips containing several hundred samples. Under these circumstances, the specialist can remember and use as many as 1000 color names, but many of the names are idiosyncratic; the colors corresponding to *taupe*, *fiesta red*, and *primrose* are imprecisely defined for most of us. In addition, as soon as these colors are removed from the standard booth, they will change their appearance because of illumination-induced adaptation and contrast effects.

The Natural Color System (NCS), a standardized color naming system, has been developed based on Hering's opponent color theory (1920). NCS was developed in Sweden and is widely used in England and other European countries. In NCS, colors are characterized by the amounts of redness, greenness, yellowness, blueness, blackness, and whiteness that they contain. As shown in Figure 4.19, red, green, yellow, and blue lie at the ends of two orthogonal axes. Intervening "pure" colors lie on the circle circumference, and these are given numbers by sharing out 100 arbitrary units; thus, a yellowish orange might be given the value Y70R30, meaning 70 parts yellow and 30 parts red. Colors are also given independent values on a black–white axis by allocating a blackness value between 0 and 100. A third color attribute, intensity (roughly corresponding to saturation), describes the distance from the grayscale axis. In NCS, for example, the color *spring nymph* becomes 0030-G80Y20, which expands to blackness 00, intensity 30, green 80, and yellow 20 (Jackson et al., 1994). The NCS system combines some of the advantages of a color geometry with a reasonably intuitive and precise naming system.

In North America, other systems are more popular than NCS. The Pantone® system is widely used in the printing industry, and the Munsell system is an important reference for surface colors. The Munsell system is useful because it provides a set of standard color chips designed to represent equal perceptual spacing in a three-dimensional mesh. (Munsell color chips and viewing booths are available commercially, as are Pantone products.) The NCS, Pantone, and Munsell systems were originally designed to be used with carefully printed paper samples providing the reference colors, but computer-based interfaces to these systems have been developed as part of illustration

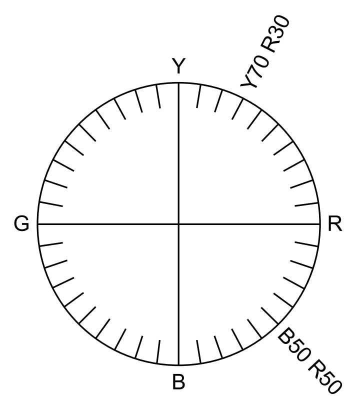

Figure 4.19 The Natural Color System (NCS) circle, defined midway between black and white. Two example color names are shown in addition to the "pure" opponent color primaries. One is an orange yellow and the other is purple.

and design packages. Rhodes and Luo (1996) described a software package that enables transformations between the different systems using the CIE as an intermediate standard.

Color Palettes

When the user wishes to use only a small set of standardized colors, providing a color palette is a good solution to the color selection problem. Often, color selection palettes are laid out in a regular order according to one of the color geometries defined previously. It is useful to provide a facility for the user to develop a personal palette. This allows for consistency in color style across a number of visualization displays.

> **[G4.10]** To support the use of easy-to-remember and consistent color codes, consider providing color palettes for designers.

Sometimes a color palette is based on one of the standard color sets used by the fabric industry or the paint industry. If this is the case, the monitor must be calibrated so that colors actually appear as specified and it must be placed in a standardized viewing environment.

Application 2: Color for Labeling (Nominal Codes)

Suppose we wish to create a visualization where colored symbols represent companies from different industrial sectors—red for manufacturing, green for finance, blue for retail, and so on. The technical name for this kind of labeling is *nominal information coding*. A nominal code does not have to be orderable; it simply must be remembered and recognized. Color can be extremely effective when we wish to make it easy for someone to classify visual symbols into separate categories; giving the objects distinctive colors is often the best solution. One of the reasons why color is often preferred is that the alternatives are generally worse. For example, if we try to create grayscale codes that are easily remembered and unlikely to be confused, we find that four is about the limit: white, light gray, dark gray, and black. Given that white will probably be used for the background and black is likely to be used for text, this leaves only two. In addition, using the gray scale as a nominal code may interfere with shape or detail perception. Chromatic coding can often be employed in a way that only minimally interferes with data presented on the luminance channel. Many perceptual factors must be considered when choosing a set of color labels.

Distinctness. A uniform color space, such as *CIEluv*, can be used to determine the degree of perceived difference between two colors that are placed close together. It might be thought that an algorithm based on *CIEluv* could be used to simply choose a set of colors that are most widely separated, but most color scheme design problems are too complex for this; background colors, symbol sizes, and application-specific requirements all must be taken into account. Also, when we are concerned with the ability to distinguish a color rapidly from a set of other colors, different rules may apply. Bauer et al. (1996) showed that the target color should lie outside the convex hull of the surrounding colors in the CIE color space. This concept is illustrated in Figure 4.20.

Unique hues. The unique hues—red, green, yellow, and blue, as well as black and white—are special in terms of the opponent process model. These colors are also special in the color vocabularies of languages worldwide. Clearly, these colors provide natural choices when a small set of color codes is required.

[G4.11] Consider using red, green, yellow, and blue to color code small symbols.

Contrast with background. In many displays, color-coded objects can be expected to appear on a variety of backgrounds. Simultaneous contrast with background colors can dramatically alter color appearance, making one color look like another. This is one reason why it is advisable to have only a small set of color codes. A method for reducing contrast effects is to place a thin white or black border around the color-coded object. This device is commonly used with signal lights; for example, train signals are displayed on large black background discs. In addition, we should never display codes using purely chromatic differences with the background. There should be a significant luminance difference in addition to the color difference.

[G4.12] For small color-coded symbols, ensure luminance contrast with the background as well as large chromatic differences with the background.

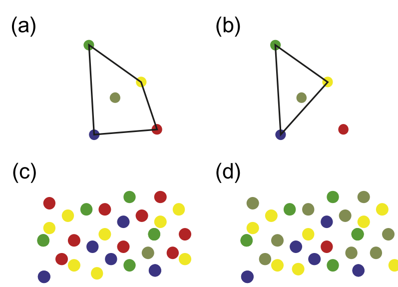

Figure 4.20 The convex hull of a set of colors is defined as the area within a rubber band that is stretched around the colors when they are defined in CIE tristimulus space. Although illustrated in two dimensions here, the concept can easily be extended to three dimensions. (a) Gray is within the convex hull of red, green, yellow, and blue. (b) Red lies outside the convex hull of green, blue, yellow, and gray. (c) The gray dot is difficult to find in a set of red, green, yellow, and blue dots. (d) The red dot is easy to find in a set of green, blue, yellow, and gray dots.

[G4.13] If colored symbols may be nearly isoluminant against parts of the background, add a border having a highly contrasting luminance value to the color, for example, black around a yellow symbol or white around a dark blue symbol.

Figure 4.21 illustrates this principle with a variety of colors against a variety of backgrounds.

Color blindness. Because there is a substantial color-blind population, it may be desirable to use colors that can be distinguished even by people who are color blind. Recall that the majority of color-blind people cannot distinguish colors that differ in a red–green direction. Almost everyone can distinguish colors that vary in a yellow–blue direction, as shown in Figure 4.8. Unfortunately, this drastically reduces the design choices that are available.

[G4.14] To create a set of symbol colors that can be distinguished by most color-blind individuals, ensure variation in the yellow–blue direction.

Figure 4.8 shows the lines defining colors that can be discriminated by most color-blind individuals.

Number. Although color coding is an excellent way to display category information, only a small number of codes can be rapidly perceived. Estimates vary between about five and ten codes (Healey, 1996).

[G4.15] Do not use more than ten colors for coding symbols if reliable identification is required, especially if the symbols are to be used against a variety of backgrounds.

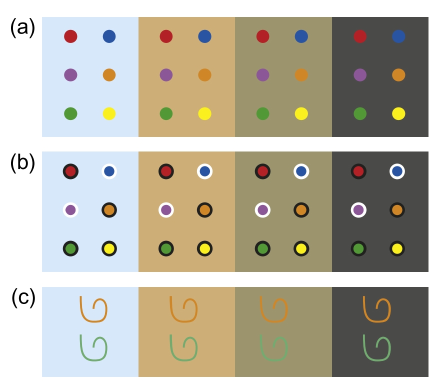

Figure 4.21 (a) Note that at least one member of the set of six symbols lacks distinctness against each background. (b) Adding a luminance contrast border ensures distinctness against all backgrounds. (c) Showing color-coded lines can be especially problematic.

Field size. To avoid the small-field color blindness illustrated in Figure 4.9, do not use very small color-coded areas. In general, the larger the area that is color coded, the more easily colors can be distinguished. Small objects that are color coded should have strong, highly saturated colors for maximum discrimination as already stated in G4.1. When large areas of color coding are used (for example, with map regions), the colors should be of low saturation and differ only slightly from one another. This enables small, vivid color-coded targets to be perceived against background regions.

> **[G4.16]** Use low-saturation colors to color code large areas. Generally, light colors will be best because there is more room in color space in the high-lightness region than in the low-lightness region.

> **[G4.17]** When color coding large background areas overlaid with small colored symbols, consider using all low-saturation, high-value (pastel) colors for the background, together with high-saturation symbols on the foreground.

Figure 4.22 shows two examples, one that follows these guidelines and one that contradicts them.

The goal of highlighting is to make some small subset of a display clearly distinct from the rest, and the same principles apply to the highlighting of text or other features in a display.

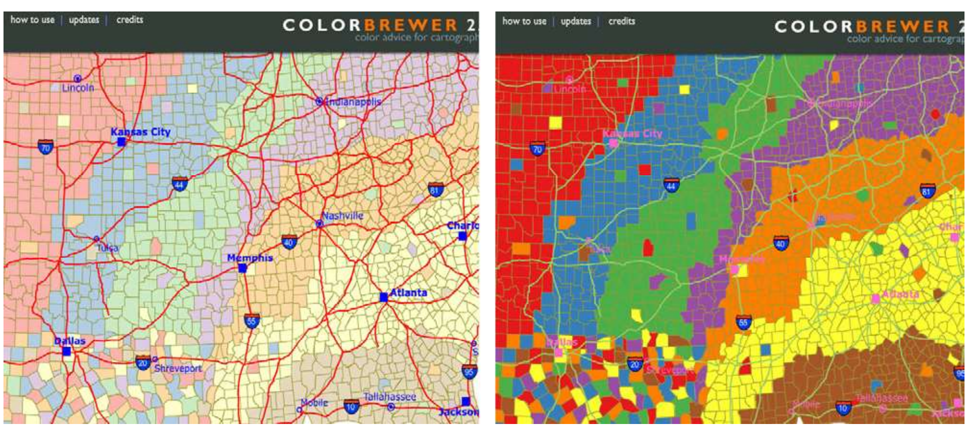

Figure 4.22 On the left is a map using low-saturation light colors for the area coding and high-saturation dark colors for the town and city symbols and linear features. On the right, a much worse solution shows high-saturation coding for areas and low-saturation symbols and linear features. Maps were generated using ColorBrewer2 (http://colorbrewer2.org).

> **[G4.18]** When highlighting text by changing the color of the font, it is important to maintain luminance contrast with the background. With a white background, high-saturation dark colors must be used to change the font color. Alternatively, when changing the background color, low-saturation light colors should be used if the text is black on white.

Figure 4.23 illustrates these two alternatives.

> ***Conventions.*** Color-coding conventions must sometimes be taken into account. Some common conventions are red = hot, red = danger, blue = cold, green = life, and green = go. It is important to keep in mind, however, that these conventions do not necessarily cross cultural borders. In China, for example, red means life and good fortune, and green sometimes means death.

The following is a list of 12 colors recommended for use in coding: red, green, yellow, blue, black, white, pink, cyan, gray, orange, brown, purple. They are illustrated in Figure 4.24. These colors have widely agreed upon category names and are reasonably far apart in color space. The first four colors, together with black and white, are chosen because they are the unique colors that mark the ends of the opponent color axes. The entire set corresponds to the 11 color names found to be the most common in the cross-cultural study carried out by Berlin and Kay (1969), with the addition of cyan.

(a) Higlighting text by changing the characters must be done using high saturation colors that contrast with the background.

(b)

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;

    public class ColorText extends Applet
{
    public void init()
    {
            red = 100;
            green = 255;
            blue = 20;
    }
    public void paint (Graphics g)
    {
            Gr.setColor(new Color(red, green, blue));
            Gr.drawString("ColoredText". 30,50);
    }
    private int red;
    private int green;
    private int blue;
}
```

Figure 4.23 Two different methods for highlighting black text. (a) Change text itself using a relatively dark, high-saturation color. (b) Change text background using low-saturation light colors. Both maintain luminance contrast.

The colors in the first set of six would normally be used before choosing any from the second set of six.

Sometimes it is useful to group color codes into families. This can be done by using hue as a primary attribute denoting family membership, with secondary values mapped to a combination of saturation and lightness. Figure 4.25 illustrates some examples. Generally, we cannot expect to get away with more than two different color steps in each family. The canonical red, green, and blue hues make good categories for defining families. Yellow is not so good because dark yellow is perceived as belonging to a different family and yellow has few discriminable saturation steps. Family members then can be distinguished from one another by saturation, as in Figure 4.25(a), or, even better, by saturation and lightness, as in Figure 4.25(b). Interior designers often consider a family of warm colors (nearer to red in color space) to be distinct from a family of cool colors (nearer to blue and green in color space), although the psychological validity of this is questionable.

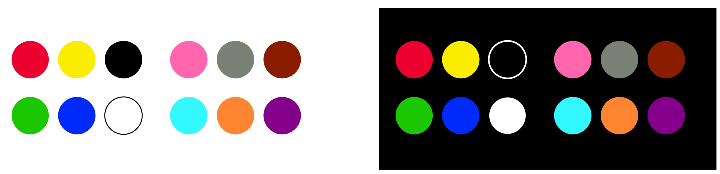

Figure 4.24 A set of 12 colors for use in labeling. The same colors are shown on a white and a black background.

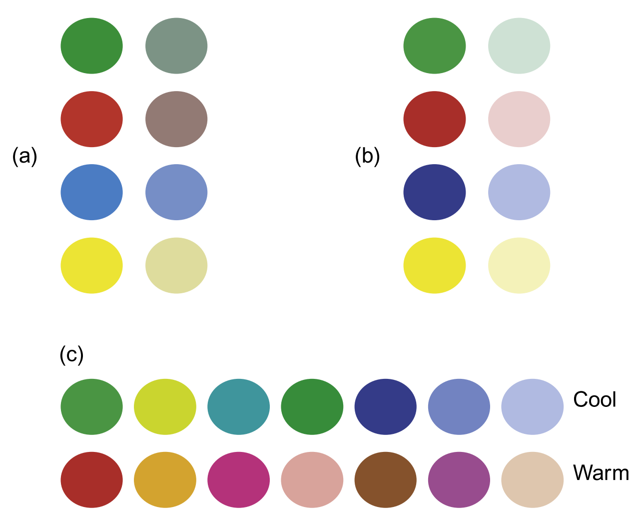

Figure 4.25 Families of colors. (a) Pairs related by hue; family members differ in saturation. (b) Pairs related by hue; family members differ in saturation and lightness. (c) A family of cool hues and a family of warm hues.

Application 3: Color Sequences for Data Maps

Somewhere in almost every newspaper and on every weather website is a map where regions are colored differently to show the forecast temperatures. Red is used to show hot weather, blue is used to show cold weather, and other colors are arranged in between, often using the colors of the rainbow, blue–cyan–green–yellow–orange–red.

Pseudocoloring is the technique of representing continuously varying map values using a sequence of colors. The result is sometimes called a *choropleth* map. Pseudocoloring is used widely for astronomical radiation charts, medical imaging, and many other scientific applications. Geographers use a well-defined color sequence to display height above sea level—lowlands are always colored green, which evokes vegetation, and the scale continues upward, through brown, to white at the peaks of mountains.

The most common coding scheme used in data visualization is a color sequence that approximates the physical spectrum, like that shown in Figure 4.26(b). Although this sequence is frequently used in physics and other disciplines and has some useful properties, it is not a perceptual sequence. This can be demonstrated by the following test. Give someone a series of gray paint chips and ask them to place them in order. They will happily comply with either a dark-to-light ordering or a light-to-dark ordering. Give the same person paint chips with the colors red, green, yellow, and blue and ask them to place them in order, and the result will be varied. For most people, the request will not seem particularly meaningful. They may even use an alphabetical ordering. This demonstrates that the whole spectrum is not perceptually ordered, although short sections of it are. For example, sections from red to yellow, yellow to green, and green to blue all vary monotonically (they continuously increase or decrease) on both the red–green and yellow–blue channels. Figure 4.27 shows seven different color sequences, but which is best and why?

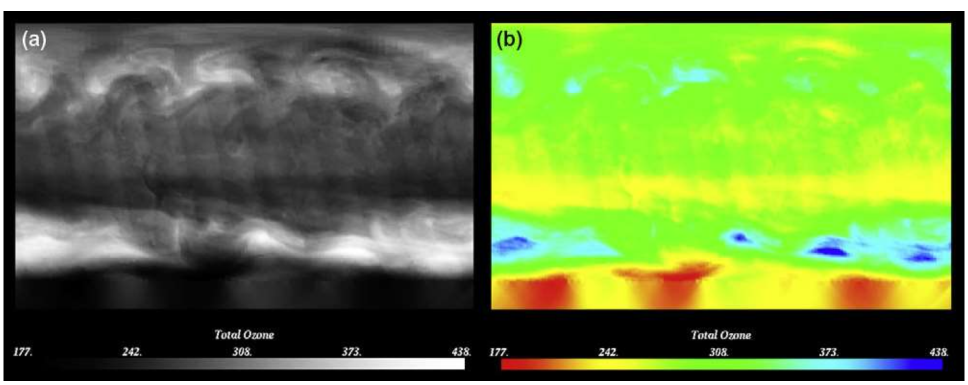

Figure 4.26 The same data showing ozone concentrations in the southern hemisphere is represented using (a) grayscale and (b) spectrum approximation pseudocolor sequences. (*Images courtesy of Penny Rheingans (Rheingans, 1999).*)

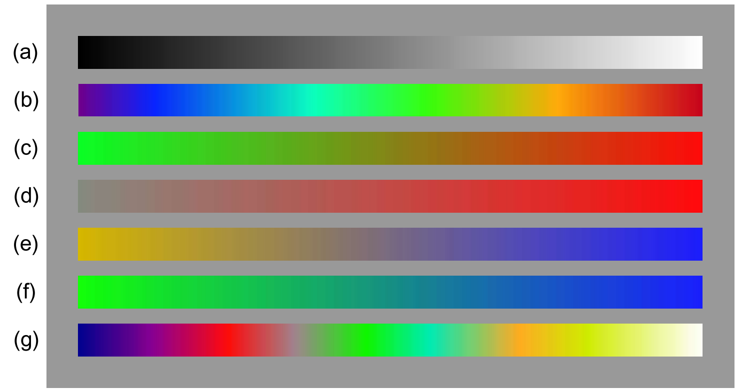

Figure 4.27 Seven different color sequences: (a) Grayscale. (b) Spectrum approximation. (c) Red–green. (d) Saturation. (e, f) Two sequences that will be perceived by people suffering from the most common forms of color blindness. (g) Sequence of colors in which each color is lighter than the previous one.

Form and Quantity

Sometimes we want to see the forms in a data set. Where are the highs and lows, the ridges and spirals, in a map of ozone in the atmosphere? Sometimes we want to be able to read the quantities. What is the temperature going to be in my part of the world tomorrow?

Color theory predicts that different color sequences have very different properties in this regard (Ware, 1988). Because the luminance channel helps us see forms, a grayscale sequence should allow us to see forms much better than pure color sequences (no luminance variation). See Figure 4.26(a). The highs are white, the lows are black, and complex swirling patterns can be seen in the ozone concentrations. Look at Figure 4.26(b). Here red, green, and blue areas clearly stand out, but this visual segmentation is meaningless; it is not clear which areas are high and low, and much less detail is seen overall.

Experimental studies have confirmed that grayscale maps are much better for form perception (Ware, 1988; Kindlmann et al., 2004). In spite of this, a recent survey of papers containing pseudocolored maps found that more than 50% used an approximation to the physical spectrum—a rainbow as a color sequence (Borland & Taylor, 2007). The same paper argued that this color sequence "*hinders this task* [of effectively conveying information] *by confusing, obscuring, and actively misleading.*"

Nevertheless, there are advantages to the spectrum approximation color sequence. The first is that it results in much lower errors in reading values from a key. Ware (1988) found 17% scale errors with a grayscale map and only 2.5% error with a spectrum approximation. There are two reasons for this. A spectrum color sequence can convey significantly more legible steps than a simple blue-to-red sequence (that is sometimes used for coding temperature).

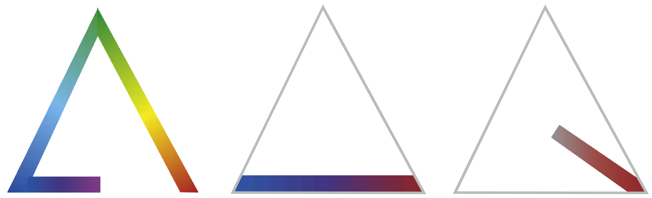

Figure 4.28 Sequences on a chromaticity diagram. (a) Spectrum approximation. (b) Blue–red sequence. (c) Saturation sequence.

Consider the sequences illustrated in Figure 4.28. These are based on the *RBG* triangle in the *CIEluv* uniform color space. A typical spectrum sequence actually starts with purple (not a true spectrum hue) and cycles clockwise through the spectrum colors to red. This path has more than two and a half times as many discriminable steps according to *CIEluv* than the red–blue sequence, and it has four times as many legible steps as a saturation sequence. Also, according to the *CIEluv* and *CIElab* color difference standards, there are about twice as many discernable steps in a traversal across the chromatic dimensions of color space as there are traversing the luminance dimension (Mahny, 1994). Another cause of errors in reading map values using a key is simultaneous contrast between parts of the display (Cleveland & McGill, 1983; Ware, 1988; Brewer, 1996b). These errors may be reduced in the spectrum sequence scale because the colors surrounding a particular point (and inducing contrast effects) are likely to partially cancel each other.

There are also sometimes semantic reasons for using a spectrum approximation sequence. Consider the case of the display of temperature in weather maps. The color blue is associated with cold. The color red is associated with heat. Having green and yellow in between provides a convenient method for conveying intermediate temperatures. In the case of a weather map, and many other visualizations, the ability to read quantity as well as see patterns in the data is essential. We may need to be able to read temperatures to better than 5 degrees over a 50-degree or greater range. This requires ten or more discriminable steps, something that is impossible to achieve with the blue-to-red sequence. Finally, the use of spectrum approximations for temperature maps is deeply embedded in large parts of our culture. Using this well-established standard can have huge efficiencies because it eliminates the need to learn something new.

[G4.19] Use a spectrum approximation pseudocolor sequence for applications where its use is deeply embedded in the culture of users. This kind of color sequence can also be used where the most important requirement is reading map values using a key. If this sequence is used, the spacing of the colors should be carefully chosen to provide discriminable steps.

Still, if it is important to show detail in the data, then it is essential to make that detail stand out using the luminance (black–white) channel because of the capacity of this channel to convey high-spatial-frequency information (Ware, 1988; Rogowitz & Treinish, 1996). Also, if form perception is the primary consideration, a sequence that trends upward or downward in luminance will be better.

Some authors have recommended that, for clarity, color sequences should constitute a straight line through a perceptual color space, such as *CIEluv* or *CIElab* (Robertson & O'Callaghan, 1988; Levkowitz & Herman, 1992). This would rule out the spectrum approximation sequence. Further, Spence et al. (1999) found that a color sequence combining variation in brightness, saturation, and hue was the most effective in a task requiring the rapid detection of low and high points in an image.

A better choice may be to design a sequence that cycles through a variety of colors, each one lighter than the previous. Sometimes this is called a *spiral color sequence*, because it can be thought of as spiraling upward in color space. Such a sequence can combine the advantages of monotonicity in luminance, so as to show form and detail, as well as reduce contrast-induced errors and enable accurate readings from a color key (Ware, 1988; Levkowitz & Herman, 1992; Kindelmann et al., 2004).

> **[G4.20]** If it is important to see highs, lows, and other patterns at a glance, use a pseudocolor sequence that monotonically increases or decreases in luminance. If reading values from a key is also important, cycle through a variety of hues while trending upward or downward in luminance.

The designer of such a sequence can take advantage of the fact that monitor blue has much lower luminance than monitor red, which in turn has lower luminance than monitor green. Yellow, being the sum of red and green, has a very high luminance, almost equal to white. This is the basis for the sequence design shown on the right in Figure 4.29.

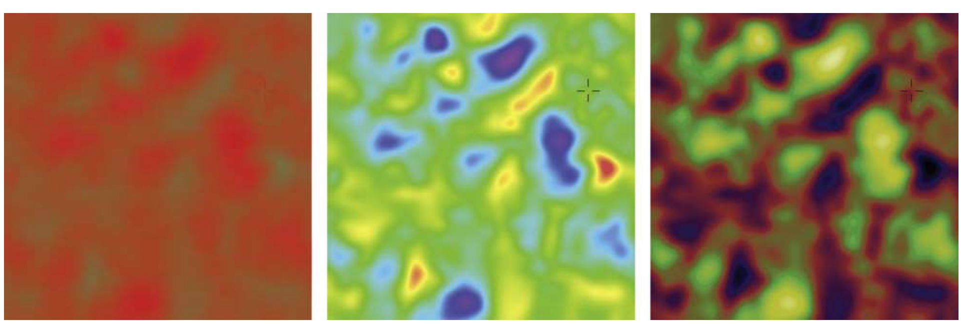

Figure 4.29 The same data represented with saturation, spectrum, and spiral color sequences. The spiral sequence makes it possible to easily see both the highs and lows, as well as read values accurately from a key.

Interval Pseudocolor Sequences

An interval sequence is one in which each unit step of the sequence represents an equal change in magnitude of the characteristic being displayed across the whole range of the sequence. In terms of color, this suggests using a uniform color space in which equal perceptual steps correspond to equal metric steps (Robertson & O'Callaghan, 1988). Using a contour map, not a color sequence, is the traditional way to display an interval sequence. Isovalue contour maps show the pattern of equal heights or other physical attributes with great precision, but using them to understand the overall shape of a terrain or an energy field takes considerable skill and experience. To support unskilled map readers, contours can be usefully combined with pseudocoloring, as shown in Figure 4.30(a). Even better may be a stepped pseudocolor sequence as shown in Figure 4.30(b).

Ratio Pseudocolors

A ratio sequence is an interval sequence that has a true zero and all that this implies: The sign of a value is significant; one value can be twice as large as another. Expressing this in a color sequence is a tall order. No known visualization technique is capable of accurately conveying ratios with any precision; however, a sequence can be designed that effectively expresses a zero point and numbers above and below zero. Brewer (1996a) called such sequences *diverging sequences*, whereas Spence and Efendov (2001) called them *bipolar sequences*. Such sequences typically use a neutral value on one or more

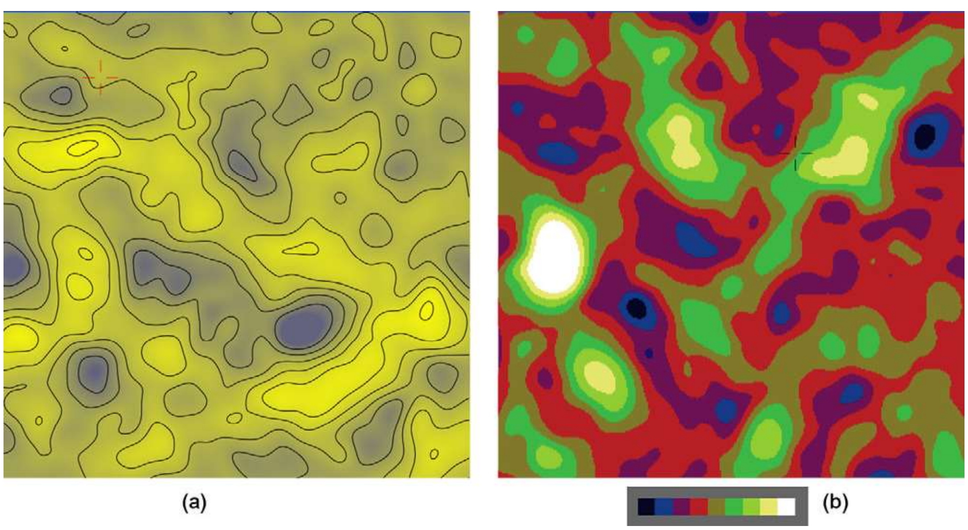

Figure 4.30 (a) Contours can show equal intervals in the data although numerical labels must be added for most applications. (b) A sequence of colors in discrete steps may be more reliably read using a key than a smoothly blended sequence.

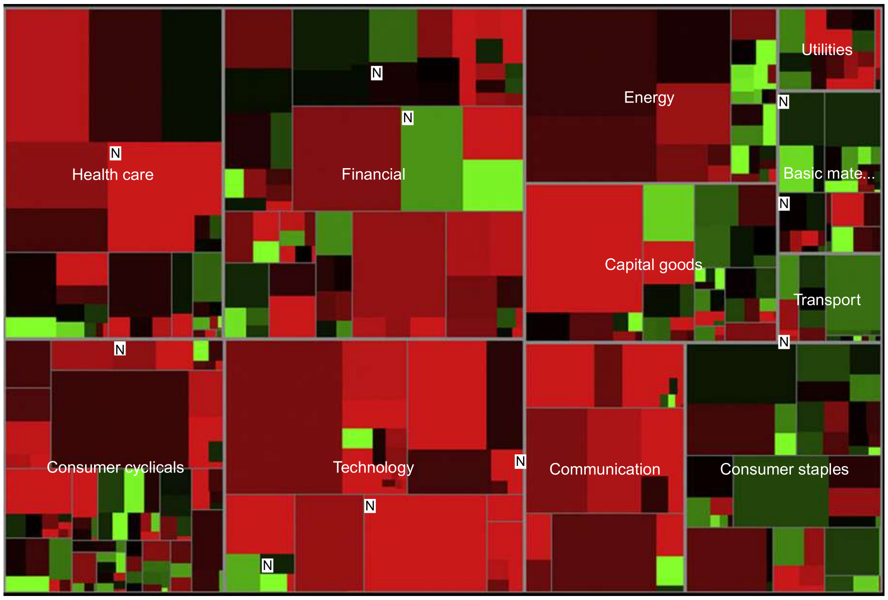

Figure 4.31 A color sequence with black representing zero. Increasing positive values are shown by increasing amounts of red. Increasing negative values are shown by increasing amounts of green. The map itself is a form of treemap (Johnson & Shneiderman, 1991). (*Courtesy of SmartMoney.com.*)

opponent channels to represent zero, and diverging colors (on one or more channels) to represent positive and negative quantities. For example, gray may be used to represent zero, increasing redness to represent positive quantities, and increasing blueness to represent negative quantities. In a target detection study, Spence and Efendov (2001) found that a red–green sequence was most effective, confirming the greater information-carrying capacity of this channel compared to the yellow–blue channel.

The example in Figure 4.31 shows a map of the stock market provided by SmartMoney .com. Market capitalization is represented by area, luminance encodes the magnitude of value change in the past year, and green–red encodes gains and losses. The website also gives users the option of a yellow–blue coding, suitable for most color-blind individuals.

Sequences for the Color Blind

Some color sequences will not be perceived by people who suffer from the common forms of color blindness: protanopia and deuteranopia. Both cause an inability to discriminate red from green. Sequences that vary mainly on a black-to-white scale or on a yellow-to-blue dimension (this includes green to blue and red to blue) will still be clear to color-blind people. Two sequences that will be acceptable to these individuals are shown in Figure 4.27(e, f). Meyer and Greenberg (1988) provided a detailed analysis of color sequences designed for common forms of color blindness.

Bivariate Color Sequences

Because color is three dimensional, it is possible to display two or even three dimensions using pseudocoloring (Trumbo, 1981). Indeed, this is commonly done in the case of satellite images, in which invisible parts of the spectrum are mapped to the red, green, and blue monitor primaries.

Although this mapping is simple to implement and corresponds to capabilities of the display device (which usually has red, green, and blue phosphors), such a scheme does not map the data values to perceptual channels. In general, it is better to map data dimensions to perceptual color dimensions. For example:

Variable one → hue

Variable two → saturation

or

Variable one → hue

Variable two → lightness

Figure 4.32 gives an example of a bivariate color sequence from Brewer (1996a) that maps one variable to yellow–blue variation and the other to a combination of light–dark variation and saturation. It suffers from the usual problem that the low-saturation colors are difficult to distinguish.

As a word of caution, it should be noted that bivariate color maps are notoriously difficult to read. Wainer and Francolini (1980) carried out an empirical evaluation of a color sequence designed for U.S. census data and found that it was essentially unintelligible. One approach to a solution is to apply a uniform color space, and Robertson and O'Callaghan (1986) discussed how to do this. But, distinctness may not lead to something that is interpretable. We do not seem to be able to read different color dimensions in a way that is highly separable.

Pseudocoloring is not the only way to display a two-dimensional scalar field. Generally, when the goal is to display two variables on the same map, it may be better to use visual texture, height difference, or another channel for one variable and color for the other, in this way mapping data dimensions to more perceptually separable dimensions. Mapping the scalar field to artificial height and shading the resulting surface with an artificial light source using standard computer graphics techniques is another alternative. These methods are discussed later in the book.

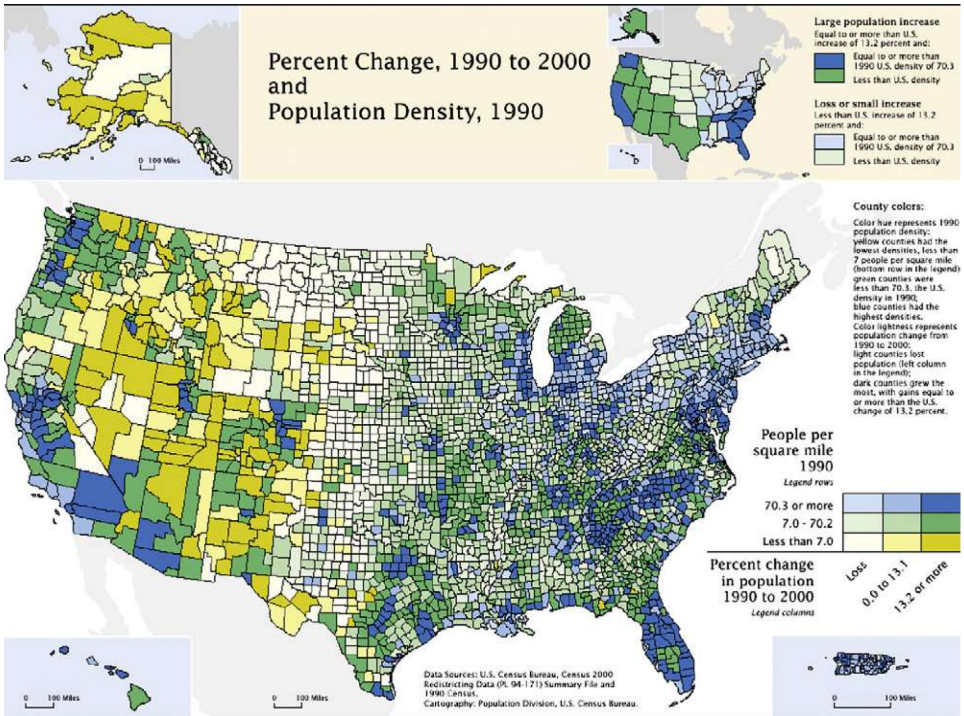

Figure 4.32 A bivariate coloring scheme using saturation and lightness for one variable and yellow–green–blue hue variation for the other. (*Courtesy of Cindy Brewer.*)

Many considerations go into making a color sequence that displays quantities without significant distortions, and this makes it unlikely that any predefined set of colors will exactly suit a particular data set and visualization goal. To show both overall form and detail and to provide the ability to read values from a key, it is often desirable to emphasize certain features in the data by deliberately using a nonuniform sequence; assigning more variation in color to a particular data range will lead to its visual emphasis and better discrimination of those values. Generally, the best way to achieve an effective color sequence is to place a good color editing tool in the hands of someone who understands both the data display requirement and the perceptual issues of color sequence construction (Guitard & Ware, 1990).

Application 4: Color Reproduction

The problem of color reproduction is essentially one of transferring color appearances from one display device, such as a computer monitor, to another device, such as a sheet of paper. The colors that can be reproduced on a sheet of paper depend on such factors as the color and intensity of the illumination. Northern daylight is much bluer than direct sunlight or tungsten light, which are both quite yellow, and is prized by artists for this reason. Halogen light is more balanced. Also, monitor colors can be reproduced only within the range of printing inks; therefore, it is neither possible nor meaningful to reproduce colors directly using a standard measurement system such as the CIE *XYZ* tristimulus values.

As we have discussed, the visual system is built to perceive relationships between colors rather than absolute values. For this reason, the solution to the color reproduction problem lies in preserving the color relationships as much as possible, not the absolute values. It is also important to preserve the white point in some way, because of the role of white as a reference in judging other colors.

Stone et al. (1988) described a process of gamut mapping designed to preserve color appearance in a transformation between one device and another. The set of all colors that can be produced by a device is called the *gamut* of that device. The gamut of a monitor is larger than that of a color printer (roughly the gamut of surface colors shown in Figure 4.7). Stone et al. described the following set of heuristic principles to create good mapping from one device to another:

1. The gray axis of the image should be preserved. What is perceived as white on a monitor should become whatever color is perceived as white on paper.
2. Maximum luminance contrast (black to white) is desirable.
3. Few colors should lie outside the destination gamut.
4. Hue and saturation shifts should be minimized.
5. An overall increase of color saturation is preferable to a decrease.

Figure 4.33 illustrates, in two dimensions, what is in fact a three-dimensional set of geometric transformations designed to accomplish the principles of gamut mapping. In this example, the process is a transformation from a monitor image to a paper hard copy, but the same principles and methods apply to transformations between other devices.

- **Calibration.** The first step is to calibrate the monitor and the printing device in a common reference system. Both can be characterized in terms of CIE tristimulus values. The calibration of the color printer must assume a particular illuminant.
- **Range scaling.** To equate the luminance range of the source and destination images, the monitor gamut is scaled about the black point until the white of the monitor has the same luminance as the white of the paper on the target printer.
- **Rotation.** What we perceive as neutral white on the monitor and on the printed paper can be very different, depending on the illumination. In general, in a printed image, the white is defined by the color of the paper. Monitor white is usually defined by the color that results when the red, green, and blue monitor primaries are set to their maximum values. To equate the monitor white with the paper white, the monitor gamut is rotated so as to make the white axes colinear.

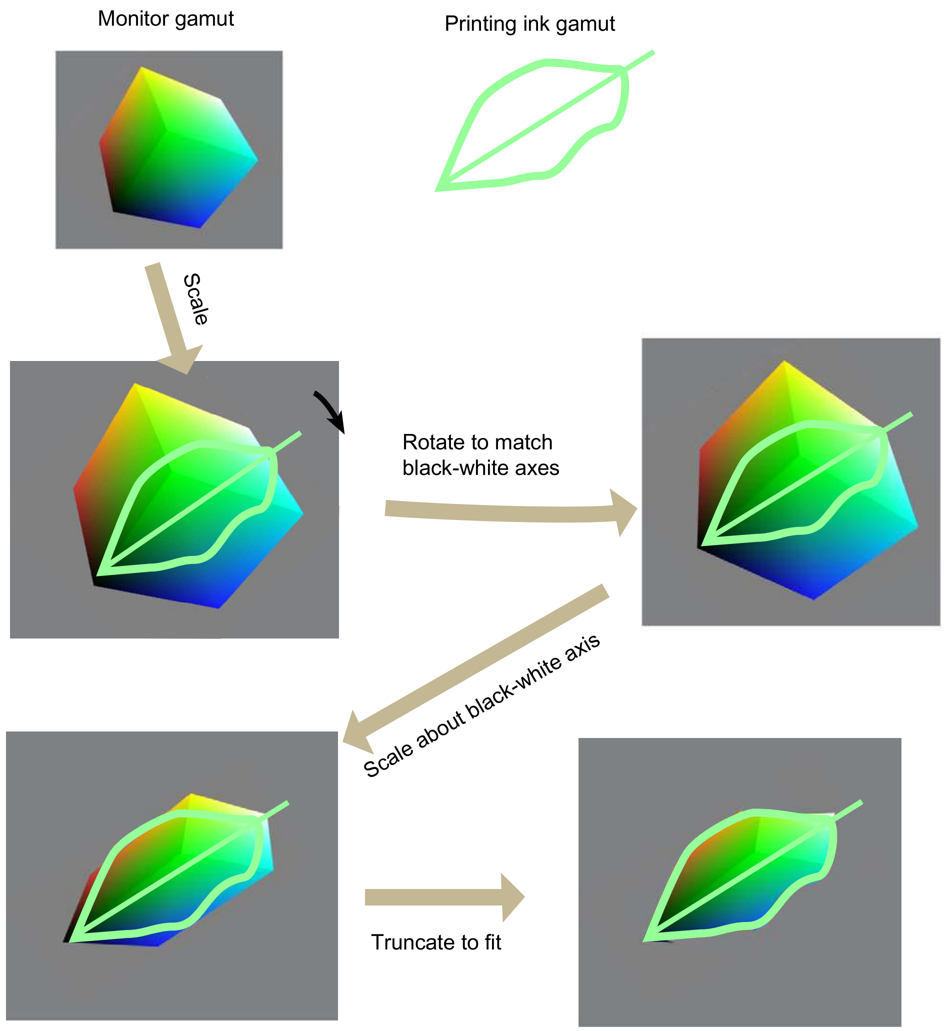

Figure 4.33 Illustration of the basic geometric operations in gamut mapping between two devices, as defined by Stone et al. (1988).

- **Saturation scaling.** Because colors can be achieved on a monitor that cannot be reproduced on paper, the monitor gamut is scaled radially with respect to the black–white axis to bring the monitor gamut within the range of the printing gamut. It may be preferable to leave a few colors outside the range of the target device and simply truncate them to the nearest color on the printing-ink gamut boundary.

For a number of reasons, it may not always be possible to apply these rules automatically. Different images may have different scaling requirements; some may consist of pastel colors that should not be made too vivid, whereas others may have vivid colors that must be truncated.

The approach adopted by Stone et al. (1988) is to design a set of tools that support these transformations, making it easy for an educated technician to produce a good result; however, this elaborate process is not feasible with off-the-shelf printers and routine color printing. In these cases, the printer drivers will contain heuristics designed to produce generally satisfactory results. They will contain assumptions about such things as the gamma value of the monitor displaying the original image and methods for dealing with oversaturated colors. Sometimes, the heuristics embedded in devices can lead to problems. In our laboratory, we usually find it necessary to start a visualization process with somewhat muted colors to avoid oversaturated colors on videotape or in paper reproduction.

Another issue that is important in color reproduction is the ability of the output device to display smooth color changes. Neural lateral inhibition within the visual system tends to amplify small artificial boundaries in smooth gradients of color as Mach bands. This sensitivity makes it difficult to display smoothly shaded images without artifacts. Because most output devices cannot reproduce the 16 million colors that can be created with a monitor, considerable effort has gone into techniques for generating a pattern of color dots to create the overall impression of a smooth color change. Making the dots look random is important to avoid aliasing artifacts (discussed in Chapter 2). Unless care is taken, artifacts of color reproduction can produce spurious patterns in scientific images.

Conclusion

There has been more research on the use of color in visualization than any other perceptual issue. Nevertheless, the important lessons are relatively few, and mostly they can be derived from opponent process theory. There are two chromatic channels (red–green and yellow–blue) and a luminance channel. Because of the low spatial resolution of the chromatic channels, small symbols should have high-saturation colors. Because of chromatic contrast in the opponent channels, we can only expect to have a few color symbols reliably identifiable. Contrast effects also make it desirable that larger regions should be less strongly colored in general.

It is impossible to keep a discussion of color entirely segregated in one chapter. Color affects every aspect of visualization and is mentioned in many other chapters, especially Chapter 5, which places color in the context of other methods for coding information.