Linotype-Hell

Technical Information

Color Spaces and PostScript Level 2

Most people who do color work require accurate, repeatable color. They want to be able to predict from their monitor display just what a final printed piece will look like. Unfortunately, there are many reasons why this is difficult to do. This document will cover the reasons for these difficulties, and also discuss how PostScript** Level 2 hopes to solve them.

Device-independent color

Device independence is one of the foundations of PostScript. A device-independent job may be sent to many different output devices, and each of those devices produces the best output of which it is capable. When PostScript Level 2 is released, it will include features to promote device-independent color. Device-independent color should give users the ability to move accurately and consistently between scanner, monitor, and output device so that the final result will be the best that the output device can provide.

Device-independent color requires accurate conversions between the color spaces that are in use in the graphic arts: RGB for monitors and scanners, and CMYK for most printers. Color conversion can be thought of as a type of device independence. If you can transfer files between devices and still maintain color integrity, then you have achieved device-independent color.

Representing Color

There are two common means of displaying images: on a monitor or on a printed surface. Color monitors use glowing phosphors of three colors to produce an image on screen. These colors are red, green, and blue (commonly referred to as RGB.) Most color printed material is reproduced using four colors of pigment: cyan, magenta, yellow, and black (commonly referred to as CMYK.) (For a fuller explanation, see the technical information piece entitled Color in Printing, part number 3056.) RGB phosphors and CMYK pigments differ from monitor to monitor and from press to press. And although printed colors can be described by the halftone dot percent (see the technical information piece entitled Screen Ruling and Resolution, part number 3050,) two colors printed with the exact same dot percentages, but with different sets of inks or perhaps on a different paper stock, may appear quite different.

There is no universal standard for RGB or CMYK. There isn’t even a defined monitor white or paper white. Because of this, both RGB and CMYK leave much to be desired in terms of color matching. Add to this that a conversion must be done to take values from the monitor display and reproduce them on paper (and that there is no set agreed upon way to do this conversion) and you will see why color at present can be such an inexact science.

A good analogy for this is the conversion of measurements between meters and yards. For a gross estimate, you can assume that the number of meters is approximately equal to the number of yards. But if you measured a mile thinking meters were yards, you would be off by more than the length of a football field. If on top of this, the length of a yard changed from day to day or varied between countries, then you can start to see the dilemma that color conversions face.

There are simple conversions between RGB and CMYK, but they are prone to inaccuracies for the reasons described above (ie., not all RGB phosphors are the same, nor are all CMYK pigments). The best way to do an accurate conversion is to use a consistent reference that is not dependent on phosphors or pigments. The hardest problem with this is deciding upon that reference system. Color scientists have developed several, and they have trouble deciding which one is best for a wide range of applications.

Color spaces

A color space is a way of describing colors in three dimensions. Color spaces therefore, are usually built using three coordinates. This explains the preponderance of three-letter color acronyms (RGB, CMY, HSV, XYZ, L*a*b*).

Hue is the generic name of a color.
Saturation is the relative strength of a color.
Value is a measure of the white or black component of a color.

One of the easiest color spaces to describe is HSV (or Hue, Saturation, and Value). Hue is the generic color name, for example: red, orange, blue, green, yellow, etc. Saturation (sometimes called chroma or color strength) is the relative strength of a color. A neutral gray has no saturation, while a vivid red is extremely saturated. Value, sometimes referred to as intensity or brightness, is a measure of the white or black component of a color.

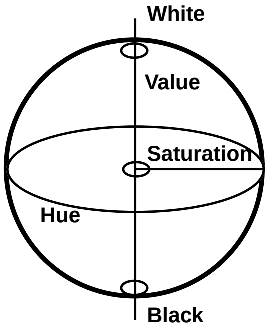

Figure 1 - A globe as a model of a color space.

Using these three descriptors you can build a color space. Imagine for a moment a globe (see Figure 1). The equator would describe hue, with red, yellow, green, cyan, blue, and magenta lining the tropics. The axis running from the north to the south pole would be value (with white at the north and black at the south. The saturation of any hue would be described by a line running horizontally from the surface of the globe to the value axis. The closer you get to the value axis, the less the saturation of any given color. This makes sense since all the values on the axis range from white to gray to black (which are not saturated at all.) Red, for example, would fall on the equator, as you approached the center of the earth the color would become less and less saturated until finally at the center it would be a neutral gray.

The globe is a simplistic model of color description. But it makes it fairly easy to pick out colors. (Those of you who have used the Apple** color wheel will be familiar with how today's software programs display a color space.) But the color globe doesn't give you an easy way of finding exact colors, nor does it accurately describe the range of colors, or even the ability with which the eye can distinguish between neighboring colors. To do this, it helps to have a more sophisticated method of measuring color, a consistent reference space that allows valid conversions between RGB and CMYK.

CIE

The Commission International de l'Eclairage (CIE) is a group of color scientists who joined together to create a color standard. CIE standards are based on the premise that you can't measure the color of an object without knowing something about the observer and the light source The first CIE standard was published in 1931. In it, the color scientists picked a standard light source, and created a standard observer based on experiments in which people looked at color patches under controlled circumstances. This is the so-called CIE XYZ color space that all other CIE color spaces are based on. Another color space, CIE L*a*b* (pronounced L-star, a-star, b-star) was released in 1976. CIE L*a*b* takes CIE XYZ, and sends it through a non-linear transformation that makes the colors within it appear more equally-spaced visually. This makes it easier for users to understand just how different neighboring colors are. PostScript Level 2 will have a PostScript procedure for decoding CIE L*a*b*.

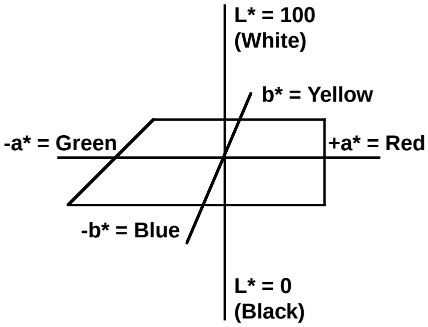

*Figure 2 - CIE L*a*b* diagram.*

CIE L*a*b* is an opponent type system. This means that white/black, red/green, and yellow/blue are set up as the ends of each axis in the color space. The theory states that something can't be pure white and pure black at the same time, and that mixtures of white and black can be designated along an axis between the two. The same can be said of red/green and blue/yellow. L* refers to the value (lightness) axis (where white = 100, black = 0), a* refers to the red/green axis (+a = red, -a = green), and b* refers to the yellow/blue axis (+b = yellow, -b = blue). (See Figure 2.)

In PostScript Level 2, Adobe has built a flexible framework for applications to use any number of device-independent color spaces based on the international standard CIE XYZ space. CIE L*a*b* is one such device-independent color space, but not everyone agrees that it is the best. Some prefer Kodak's new color space, Photo YCC (which is also based on CIE). Photo YCC will be used in Kodak's new compact disc product (for transferring 35mm slides to CD and ultimately viewing them on video). This immediately guarantees that many images will be available in Photo YCC. But since the hooks exist in PostScript Level 2 for any CIE space, the real issue is speed and conversion. If more than one space is in common use, that means more work for developers, more complex programs, and potentially slower program response.

Tonal range and color gamut

Tonal range and color gamut play an important role in device-independent color. Tonal range describes the range between the darkest black and the whitest white that a monitor or printer can create. You might hear tonal range referred to as dynamic range or density range.

If you were to imagine a line where one end was the absolute brightest white possible in the world, and the other end represented the absolute darkest black, you could place other tonal ranges on this line. (See Figure 3.)

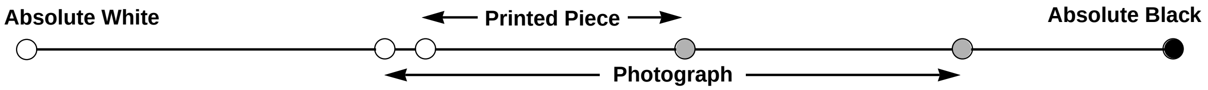

Figure 3 - Tonal range.

You can see that if you have a photograph whose darkest black is darker than the blackest ink, then you will have to somehow compress the tonal range of the photograph to reproduce it.

With color, the concept is more complex because you have to look at three rather than two dimensions. Visualizing a three-dimensional color space is harder because we normally only think of a slice of the space at a time. (If you think back to the color globe described earlier, you could slice the globe at any point to see a cross section, however you can't envision it all at once.)

Color gamut refers to all the colors that a device can reproduce. (You could think of the Apple color wheel as showing the color gamut of a monitor, one slice at a time.) The color gamut includes all of the colors that fit in the color space. Color gamuts may differ from one device to the next. (See Figure 4.)

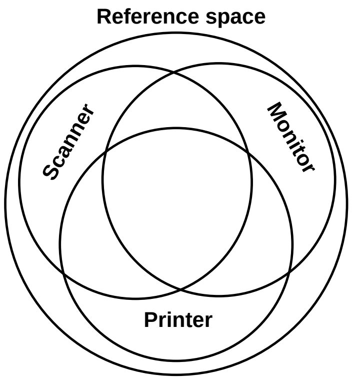

Figure 4 - Hypothetical color gamuts of a scanner, monitor, and printer (all enclosed within a reference color space.)

Figure 4 shows slices of the color gamuts of a hypothetical scanner, monitor, and printer, all falling within an overall reference space. This shows that there may be some colors that can be seen by a scanner but not reproduced on a monitor or printer. This leaves a big question:

- How do you approximate a color that falls out of a device's color gamut?

If a color (see Figure 5, part A) that appears outside of a device's gamut needs to be reproduced, some concessions must be made. The color, of course, could be approximated by substituting the closest color in the gamut, but that would mean that the relationship between it and a color that actually fell within the gamut would be altered (see Figure 5, part B). What would better preserve the relationship would be a slight altering of the existing color (see Figure 5, part C).

To address this problem, PostScript Level 2 provides the ability to create a color rendering dictionary. These color rendering dictionaries adjust output for the press, ink, and paper that will be used for a job. You might have several such dictionaries, one for each of the types of work that you do. However, for these dictionaries to be effective, the issues of characterization and calibration must be addressed.

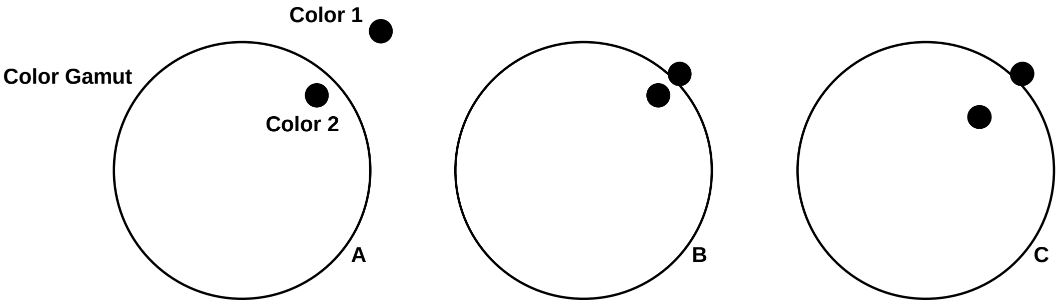

Figure 5 - Handling colors that fall out of the color gamut.

Characterization and calibration

One of the problems in maintaining accurate color is in knowing the inherent variation in the scanners, monitors, and printers used to make reproductions. In other words, what good does a consistent reference color space do for you if the devices you use vary over time? This is where characterization and calibration become important.

There are many components to calibrating a device or a process, and we have to start looking at each one. It might be easiest to divide this up into three parts: characterization, device calibration, process calibration.

Characterization involves knowing just what a device is capable of, and how much variation is likely to occur over time. Knowing this you can get an idea of how tightly you will be able to control the process. This amounts to observation without making any changes.

Device calibration would involve the controls you have over a given device to tighten the range of variation, and adjust it to get the results that you want. Some devices may have no (or limited controls) in which case you can do little more than characterize. Other devices may be able to be controlled much more accurately. A good example of device calibration is a monitor that has a tool which attaches to the screen and takes measurements from a test pattern. From this feedback it can control the state of the monitor.

If you are trying to control the process from beginning to end, you need to look at **process calibration**. With process calibration you take into account all factors involved for all devices. To do this you would have to (once all devices are stable and calibrated) send a color test through an entire system. With these results, you can develop a feedback loop that calibrates the entire process. Color rendering dictionaries will facilitate these types of calibration in PostScript Level 2. Process calibration requires sophisticated measurement tools, and strict adherence to device calibration.

Conclusion

Many components of PostScript Level 2 are not specifically color-related. Some (like compression algorithms) will ultimately play an important role in the use of color. However the most basic color improvement in PostScript Level 2 is its use of a standard, device-independent, color space which will provide greater control over color output.

Comments

Please direct any questions or comments to:

Jim Hamilton, Marketing Department
Linotype-Hell Company
425 Oser Avenue
Hauppauge, NY 11788

Part Number 3065, 4/91