



Principles and Practices of Chromatic Adaptation

Jan 9, 2021

Colorimetry provides a scientific and quantitative way to measure and compare color. By specifying a set of primaries (e.g., RGB or XYZ) and the tristimulus values, we could tell whether two light stimuli will generate the same color perception. However, colorimetry works best for a fixed viewing condition. Thus, to capture color perception when changing the viewing environment we must model key factors in a viewing condition, one of which is the viewing **illuminant**.

The significance of the viewing illuminant is that human beings change their color perception by adapting to the color of the illuminant through a process known as *chromatic adaptation*. Chromatic adaptation is fascinating; it's rooted in fundamental neuroscience and at the same time has wide applications in practice. This article describes some interesting and perhaps confusing aspects of chromatic adaptation, and discusses a few applications of chromatic adaptation.

Table of Contents

1. Principles of von Kries Chromatic Adaptation
2. Usecase A: Cross-Platform Color Reproduction
3. Usecase B: Equi-Appearance Color-Space Transformation
4. Usecase C: Color Correction in Camera Imaging
5. Usecase D: White Balance in Camera Imaging
6. Summary

1. Principles of von Kries Chromatic Adaptation

We provide a brief account for chromatic adaptation first. There are many excellent resources that provide a comprehensive treatment of chromatic adaptation. Two references that I find helpful are Color Imaging: Fundamentals and Applications (<https://dl.acm.org/doi/book/10.5555/1386684>) by Reinhard et al., and of course Color Appearance Models (<https://onlinelibrary.wiley.com/doi/book/10.1002/9781118653128>) by Fairchild.

In everyday life we encounter a wide range of viewing environments, each of which has a different illuminant ranging from incandescent light to daylight, but we have relatively constant visual experience. For instance, a piece of white paper or a white wall will appear the same white color to you whether you view it under incandescent light or under noon sunlight. This is because our visual system adapts to the illuminant insofar as we could discount the (color of the) illuminant to some extent, i.e., chromatic adaptation. Note that there are other forms of adaptation. For instance, we adapt to very dark or very bright environments, giving rise to the notions of light and dark adaptation, respectively. This article focuses on chromatic adaptation.

1.1 The Theory

How do we adapt? Johannes von Kries (https://en.wikipedia.org/wiki/Johannes_von_Kries) hypothesized that we adapt by scaling the responses of the three types of cone under different illuminants, each of which scales independently. In matrix form, von Kries adaptation is expressed as the following equation, where $[L, M, S]^T$ represents the unadapted cone responses of the neutral point, $[L_a, M_a, S_a]^T$ represents the adapted cone responses, and α, β, γ are the scaling factors (gains) of each cone type, respectively.

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

How exactly does each cone type scale? That is, what are the values of α, β, γ ? von Kries' theory is that the cone responses scale in such a way that a neutral point appears the same color under different illuminants. This basic assumption/observation would allow us to determine the scaling factors, as will see next.

But first of all, what is a neutral point? A neutral point is a point that reflects all wavelengths equally without loss of energy. That is, the spectral reflectance of a neutral point is universally 1. From the colorimetry perspective, the color of a neutral point is obviously the color of the illuminant itself, because $\int \Phi(\lambda)R(\lambda)L(\lambda)d\lambda = \int \Phi(\lambda)L(\lambda)d\lambda$ when $R(\lambda) = 1$ for any λ . A neutral point is sometimes also called a perfectly white diffuser. Since common illuminants appear white-ish, the color of the neutral point under a given illuminant is also called the white point of the illuminant.

While in reality no material is truly a perfectly white diffuser, a piece of white paper or a white wall comes close. That's why you will often hear people say things like "white always appears white under different lights." What they really mean, in precise terms, is that the neutral point maintains a constant perceived color under different illuminants.

Another way to look at this neutral point color constancy assumption is that the cone responses of a neutral point, after adaptation, are constant, independent of an illuminant. Taking this perspective, it's easy to calculate the scaling factors. Here is the idea. Let's use $[L_w, M_w, S_w]^T$ to denote the unadapted cone

responses of the illuminant itself. The unadapted cone responses of the neutral point under this illuminant must be $[L_w, M_w, S_w]^T$ too, based on the definition of a neutral point. The neutral point color constancy basically states that the adapted cone responses of the neutral point under this illuminant should be constant, i.e., independent of the illuminant $[L_w, M_w, S_w]^T$. Let's say the responses after adaptation are $[Q, P, R]^T$, where Q, P, R are just three constants independent of illuminant.

For now, we'll use $\mathbf{1}$ for Q, P, R , but we will show later the exact values of them don't matter in practical applications. As a result:

$$\alpha = \frac{P}{L_w} = \frac{1}{L_w}$$

$$\beta = \frac{Q}{M_w} = \frac{1}{M_w}$$

$$\gamma = \frac{R}{S_w} = \frac{1}{S_w}$$

This gives the familiar chromatic adaptation equation below, where Φ_w denotes the SPD of the illuminant. As we will see later, the exact numerical values of the constants are not important for practical applications.

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{bmatrix} \frac{1}{L_w} & 0 & 0 \\ 0 & \frac{1}{M_w} & 0 \\ 0 & 0 & \frac{1}{S_w} \end{bmatrix} \times \begin{bmatrix} L \\ M \\ S \end{bmatrix},$$

where

$$\begin{bmatrix} L_w \\ M_w \\ S_w \end{bmatrix} = \begin{bmatrix} \int \Phi_w(\lambda) L(\lambda) d\lambda \\ \int \Phi_w(\lambda) M(\lambda) d\lambda \\ \int \Phi_w(\lambda) S(\lambda) d\lambda \end{bmatrix}.$$

The von Kries chromatic adaptation model ensures that a neutral point does have a constant color appearance under different illuminants. This is obvious from the equation below, where $[L_{w1}, M_{w1}, S_{w1}]^T$ and $[L_{w2}, M_{w2}, S_{w2}]^T$ are two arbitrary illuminants $w1$ and $w2$.

$$\begin{bmatrix} \frac{1}{L_{w1}} & 0 & 0 \\ 0 & \frac{1}{M_{w1}} & 0 \\ 0 & 0 & \frac{1}{S_{w1}} \end{bmatrix} \times \begin{bmatrix} L_{w1} \\ M_{w1} \\ S_{w1} \end{bmatrix} = \begin{bmatrix} \frac{1}{L_{w2}} & 0 & 0 \\ 0 & \frac{1}{M_{w2}} & 0 \\ 0 & 0 & \frac{1}{S_{w2}} \end{bmatrix} \times \begin{bmatrix} L_{w2} \\ M_{w2} \\ S_{w2} \end{bmatrix}$$

However, other (non-neutral/white) colors, albeit adapted, are *not* perfectly adapted. This can be seen from the following inequality, where $[L_1, M_1, S_1]^T$ and $[L_2, M_2, S_2]^T$ are the unadapted responses of a non-neutral material under two illuminants $w1$ and $w2$, respectively. Intuitively, while an apple is seen as red-ish both under incandescent and daylight, you could probably still tell the slight difference.

$$\begin{bmatrix} \frac{1}{L_{w1}} & 0 & 0 \\ 0 & \frac{1}{M_{w1}} & 0 \\ 0 & 0 & \frac{1}{S_{w1}} \end{bmatrix} \times \begin{bmatrix} L_1 \\ M_1 \\ S_1 \end{bmatrix} \neq \begin{bmatrix} \frac{1}{L_{w2}} & 0 & 0 \\ 0 & \frac{1}{M_{w2}} & 0 \\ 0 & 0 & \frac{1}{S_{w2}} \end{bmatrix} \times \begin{bmatrix} L_2 \\ M_2 \\ S_2 \end{bmatrix}$$

Critically, von Kries chromatic adaptation operates in the LMS cone space. That is, the chromatic adaptation matrix is applied to the cone responses. If a color stimulus is expressed in other color spaces, e.g., CIE XYZ, we would have to first transform the color to the cone space before the precise von Kries adaptation can be applied.

Finally, it is worth noting that in practice we don't fully adapt even for white colors either. However, von Kries chromatic adaptation works remarkably well in most cases, and is what we will assume for the rest of this article. More complicated chromatic adaptation mechanisms can be found in Reinhard et al. 2008 (<https://dl.acm.org/doi/book/10.5555/1386684>) and Fairchild 2013 (<https://onlinelibrary.wiley.com/doi/book/10.1002/9781118653128>).

1.2 From Chromatic Adaptation to Color Appearance Models

Hopefully it is clear by now that colorimetry itself can't fully explain subjective color perception, which depends on not only the XYZ tristimulus values of a color stimulus but also the viewing condition. Chromatic adaptation moves one step toward subjective color perception by considering the illuminant.

Illuminant, however, is just one aspect of viewing condition. In reality, many other aspects of the viewing condition will affect the color perception. For instance, the immediate proximal field and the background of the color stimulus would also affect the color perception. The best example is perhaps the simultaneous lightness contrast effect (Figure 11.7 in Reinhard et al. 2008 (<https://dl.acm.org/doi/book/10.5555/1386684>)).

The previous adaptation states, i.e., the recent history of viewing conditions, also affect the color perception. A clever demonstration is the recent finding that chromatic adaptation is irreversible (<https://onlinelibrary.wiley.com/doi/abs/10.1002/col.22228>).

The subjective color perception is also very much dependent on the cognitive state of the observer. For instance, when observing an orange-ish wall, the color perception will be very different if the observer thinks the wall itself is orange to begin with versus if the observer thinks that the wall is being illuminated by street lights (Figure 11.6 in Reinhard et al. 2008 (<https://dl.acm.org/doi/book/10.5555/1386684>)).

Color appearance model comprehensively takes into account different aspects of the viewing condition, which is beyond the scope of this article. We here focus on just the chromatic adaptation aspect of color appearance modeling.

1.3 Emissive vs. Reflective Media

Sometimes you will see people say that we adapt to the white point of a color space, or we adapt to the white point of a display. What do they mean? Don't we adapt to the illuminant? Why all of a sudden we are adapting to the display or a color space?

It turns out the way we adapt changes slightly depending on whether we are looking at an emissive media, e.g., a computer display, versus a reflective media, e.g., a piece of white paper. When viewing a surface color, the observer is adapted to the surrounding illuminant. When viewing a display, the observer is (primarily) adapted to the display white point (i.e., the white point of the color space currently used by the display).

For instance if we are viewing a display that uses the sRGB color space, we adapt to D65. Why so? Intuitively, we adapt to the display white point because that's the prevailing/predominant color to the observer, similar to the effect of an illuminant. But what if a white pixel is not even presented to the viewer? How can we still adapt to the white point of the display? The explanation is debatable and sometimes a mystery, but the theory (<https://www.sciencedirect.com/science/article/pii/S2352154619300336>) is that we could estimate the equivalent illuminant from familiar objects in the images, essentially a series of visual cues.

Since the reference white of a color space is usually chosen from the CIE Standard Illuminants, which closely resemble common light sources (see Table 3 (https://www.babelcolor.com/index_htm_files/A%20review%20of%20RGB%20color%20spaces.pdf) in an article from BabelColor), we sometimes just mix the two cases together. But they are different precisely speaking: one is adapting to the chosen reference white on a display device (usually a standard illuminant), and the other is adapting to the light source.

One final note: when we look at a display under some illuminant, we actually have a "mixed adaptation" (<https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-27-3-2855&id=404429>), which can be thought of as a weighted sum between adapting to the display white point and the illuminant. Roughly, the adaptation is weighted more toward the display white, but when the display is relatively dimmer and dimmer (e.g., viewing a smartphone display under sunlight), it might have less effect on adaptation. Apple's True Tone Display (<https://www.pocket-lint.com/tablets/news/apple/137264-what-is-apple-true-tone-display>) is partially based on the notion of mixed adaptation. True Tone displays try to provide a consistent color appearance across different lighting environments by changing the display white according to the ambient lighting (<https://onlinelibrary.wiley.com/doi/abs/10.1002/sdtp.13057>).

2. Usecase A: Cross-Platform Color Reproduction

Suppose there is a color patch on an sRGB display (D65 white point), and we want to print out the color patch. Assuming that the print will be viewed under a D50 illuminant, what color (in terms of the XYZ tristimulus values) should we print so that the color appearance on the print when viewed under D50

matches the color appearance of the patch when viewed on the display? This is a typical real-world problem where the color appearance needs to precisely be maintained when viewed under different viewing conditions.

2.1 How?

Let's denote the tristimulus values of the color patch on the sRGB display $[X, Y, Z]_{D65}$, where the subscript **D65** indicates the adaptation state when the color is viewed from the display, and denote the tristimulus values of the color to be printed/calculated $[X, Y, Z]_{D50}$, where the subscript **D50**, again, indicates the adaptation state when the print is viewed. We can use the basic chromatic adaptation theory described above to derive $[X, Y, Z]_{D50}$, as shown below.

$$\begin{bmatrix} \frac{1}{L_{D65}} & 0 & 0 \\ 0 & \frac{1}{M_{D65}} & 0 \\ 0 & 0 & \frac{1}{S_{D65}} \end{bmatrix} \times T_{xyz2lms} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D65} = \begin{bmatrix} \frac{1}{L_{D50}} & 0 & 0 \\ 0 & \frac{1}{M_{D50}} & 0 \\ 0 & 0 & \frac{1}{S_{D50}} \end{bmatrix} \times T_{xyz2lms} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D50}$$

The left side of the equation is the adapted LMS cone responses of the color patch when viewing on the sRGB display. The right side of the equation is the adapted LMS cone responses when viewing on the print under the D50 illuminant. The equation holds because the color appearance of the color patch is to be maintained. $T_{xyz2lms}$ is the linear transformation from the XYZ color space and the LMS color space, which is necessary because the von Kries chromatic adaptation operates in the LMS cone space, whereas we are given the XYZ tristimulus.

Solving the equation gives $[X, Y, Z]_{D50}$:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D50} = T_{xyz2lms}^{-1} \times \begin{bmatrix} \frac{L_{D50}}{L_{D65}} & 0 & 0 \\ 0 & \frac{M_{D50}}{M_{D65}} & 0 \\ 0 & 0 & \frac{S_{D50}}{S_{D65}} \end{bmatrix} \times T_{xyz2lms} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D65}$$

The diagonal matrix chromatically adapts cone responses from D65 to D50. The product of $T_{xyz2lms}^{-1}$, the diagonal matrix, and $T_{xyz2lms}$ chromatically adapts XYZ values from D65 to D50, and we will use $M_{d65-d50}$ to represent this (composite) XYZ space adaptation matrix. This equation allows us to "predict" what color to produce to match a particular color appearance. Whether $[X, Y, Z]_{D50}$ can actually be produced in the print is a completely different matter known as gamut mapping, which is beyond the scope of this article.

An interesting observation from this calculation is that even if the constant adapted responses $[Q, P, R]^T$ don't take the value of $[1, 1, 1]^T$, the result would still be the same because Q, P, R will get canceled out in the calculation.

2.2 Another Perspective

A useful way to look at the chromatic adaptation process above is that we have essentially derived a new color space from the sRGB color space, where every color $[X, Y, Z]_{D50}$ in the new color space when viewed under D50 (i.e., when eyes adapt to D50) will have the same color appearance as that of $[X, Y, Z]_{D65}$ when viewed under D65 (i.e., when eyes adapt to D65), if $[X, Y, Z]_{D50}$ and $[X, Y, Z]_{D65}$ satisfy the equation above.

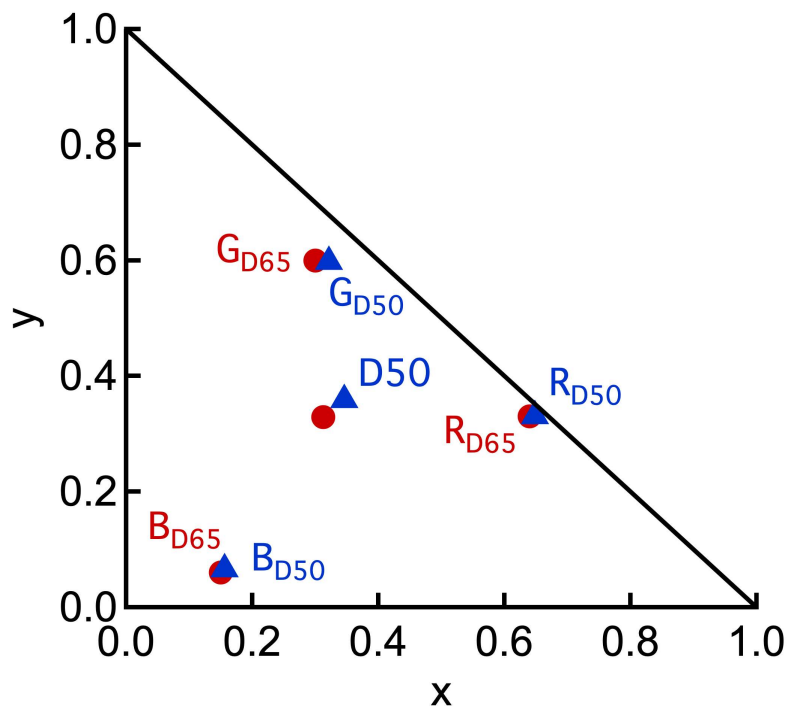
The color gamut of this new D50-adapted sRGB color space is still a parallelepiped in the XYZ color space, because the transformation is linear. It's just that the primaries and the white point's XYZ values have changed.

Specifically, the xy chromaticities of the three primaries and the white point (D65) in D65-adapted (i.e., the original) sRGB (<https://en.wikipedia.org/wiki/SRGB>) is:

Red		Green		Blue		White (D65)	
x	y	x	y	x	y	x	y
0.6400	0.3300	0.3000	0.6000	0.1500	0.0600	0.31271	0.32902

The xy chromaticities of the three primaries and the white point (D50) in D50-adapted sRGB (<https://ninedegreesbelow.com/photography/xyz-rgb.html#table2>) is:

Red		Green		Blue		White (D50)	
x	y	x	y	x	y	x	y
0.6485	0.3308	0.3212	0.5978	0.1559	0.0660	0.34567	0.3585



The figure above just plots the xy chromaticities of the primaries under D65 and D50 adaptations. We can see that the adaptation from D65 to D50 is not dramatic but noticeable. An interesting observation is that, in changing the illuminant from D65 to D50, the primaries move roughly along the same direction from D65 to D50.

3. Usecase B: Equi-Appearance Color-Space Transformation

Building on top of the example above, which converts XYZ values across adaptation states, we will discuss a more practical example, where we have to convert between different RGB color spaces while preserving color appearance.

3.1 What Is It?

Let's say you edit a photo on an sRGB display and save it in the sRGB format. Later you want to display the photo on a DCI-P3 (<https://en.wikipedia.org/wiki/DCI-P3>) display. Of course you can't just interpret the sRGB values directly as DCI-P3 values; instead, you want to apply a transformation that converts an sRGB color \mathbf{X} into a DCI-P3 color \mathbf{Y} such as \mathbf{X} viewed on the sRGB display and \mathbf{Y} viewed on the DCI-P3 display have the same appearance. The question is, how to perform this transformation?

Normally when we think of color space transformation we do that in a pure colorimetric way. If we convert a color \mathbf{X} from sRGB to a color \mathbf{Y} in DCI-P3 in a pure colorimetric way by applying the sRGB to DCI-P3 transformation matrix, we can say that color \mathbf{X} and color \mathbf{Y} will appear to have the same color --- assuming we have the same adaptation state when viewing \mathbf{X} and \mathbf{Y} . But that's not a correct assumption.

Why? Because your adaptation state is D65 when viewing \mathbf{X} on the sRGB display but the adaptation state becomes about D60 (<https://www.color.org/chardata/rgb/DCIP3.xalter>) when viewing \mathbf{Y} on the DCI-P3 display.

Therefore, in practice color space transformation must consider chromatic adaptation. In the example above, what we want is that \mathbf{X} viewed on an sRGB display and \mathbf{Y} viewed on a DCI-P3 display should have the same appearance --- even though we will have two different adaptation states when viewing \mathbf{X} and \mathbf{Y} . I call this **equi-appearance color-space transformation**. What this means is that to describe the *appearance* of a color, color space itself isn't enough; we must also specify the adaptation state/viewing illuminant. ICC color management introduces the notion of a *working space*, which, in very simple terms, is basically a color space (in the colorimetric sense) combined with a viewing (reference) illuminant, the combination of which uniquely specifies the appearance of a color.

To simplify transformation, ICC uses the idea of a Profile Connection Space (<http://www.color.org/profile.xalter>) (PCS), which is a special working space with a particular color space (CIE XYZ) and a particular adaptation state (D50). The idea is that any color \mathbf{X} in the source color space will be first converted to an equi-appearance color \mathbf{T} in the PCS, and then \mathbf{T} will be converted to the an equi-appearance color \mathbf{Y} in the target color space. Specifically for our example above, we will first convert sRGB color \mathbf{X} , which is under D65 adaptation (i.e., assuming it's viewed on an sRGB display) to an equi-appearance color \mathbf{T} in XYZ under D50 adaptation. We then transform \mathbf{T} in XYZ under D50 adaptation to an equi-appearance color \mathbf{Y} in DCI-P3, which is under D60 adaptation. That is, \mathbf{X} under D65, \mathbf{T} under D50, and \mathbf{Y} under D60 all have the same color appearance. Of course in theory the PCS can use any arbitrary color space different from XYZ and also an arbitrary adaptation state different from D50, but the point is that the color transformation needs to preserve color appearance!

3.2 How?

How do we go from \mathbf{X} in sRGB under D65 adaptation to an equi-appearance \mathbf{T} in XYZ under D50 adaptation? We will do that in two steps. First, we will convert \mathbf{X} in sRGB under D65 to \mathbf{T}' in XYZ under D65. This step is purely colorimetric, as the adaptation state is unchanged. This step uses the canonical sRGB to XYZ transformation matrix (https://en.wikipedia.org/wiki/SRGB#From_sRGB_to_CIE_XYZ); let's call it $\mathbf{M}_{srgb-xyz}$. The second step is going from \mathbf{T}' in XYZ under D65 adaptation to \mathbf{T} in XYZ under D50 adaptation, and that's the D65 to D50 chromatic adaptation matrix we derived in Section 2, whose concrete values can also be found at Bruce Lindbloom's site here (http://brucelindbloom.com/index.html?Eqn_ChromAdapt.html); let's call it $\mathbf{M}_{d65-d50}$. So the matrix that transforms \mathbf{X} in sRGB under D65 to an equi-appearance color \mathbf{T} in XYZ under D50 would simply be: $\mathbf{M}_{d65-d50}\mathbf{M}_{srgb-xyz}$, whose concrete values are shown in the last entry of the second table on this page (http://brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html) (which uses Bradford-adapted matrix for chromatic adaptation rather than von-Kris matrix we shown above).

Similarly, the matrix that transforms T in XYZ under D50 to an equi-appearance color Y in DCI-P3 under D60 is: $(M_{d60-d50}M_{dcip3-xyz})^{-1}$. Here are two ways to understand this. You can think of $(M_{d60-d50}M_{dcip3-xyz})$ as the matrix that transforms Y to T while maintaining appearance, and the inverse of that matrix does the opposite, which is what we want. Alternatively, that matrix is equal to $(M_{dcip3-xyz}^{-1}M_{d60-d50}^{-1})$ which is in turn $(M_{xyz-dcip3}M_{d50-d65})$, which basically first adapts T from D50 to D60 in XYZ, and then transforms from XYZ to DCI-P3, this time using the canonical colometric matrix $M_{xyz-dcip3}$ since the adaptation state remains unchanged.

3.3 A Few Notes

Why PCS? Why can't we just directly transform X from sRGB+D65 to Y in DCI-P3+D50 through one single transformation $(M_{d60-d50}M_{dcip3-xyz})^{-1}(M_{d65-d50}M_{srgb-xyz})$? Nothing wrong with that in theory, but it's unscalable. If we have M source color spaces and N target color spaces, we would need to encode MN different transformation matrices, but if we use a PCS, we need only $M + N$ transformation matrices.

ICC Profile. Chromatic adaptation is critical to color management, whose central goal is to maintain color appearance across media, which is what ICC cares about. To facilitate color management, ICC's idea is that any image should be associated with a input (camera/scanner) *profile*, which includes, among many things, the matrix that transforms colors in the source color space to equi-appearance values in the PCS, which, again, is XYZ+D50. So if an image is encoded in sRGB, that matrix will essentially be $M_{d65-d50}M_{srgb-xyz}$. This profile can then later be used when this image is read and manipulated on another platform with a different color space and adaptation state. The target platform will have a output *profile* given the specific display/printer, which includes a matrix that transforms colors from PCS to equi-appearance colors in the output color space. So if the output media is a DCI-P3 display, the matrix would essentially be $(M_{d60-d50}M_{dcip3-xyz})^{-1}$. These profiles allow the correct appearance-preserving transformation can be performed, a key to color management.

If you are on a Mac, you can use the built-in ColorSync Utility (<https://support.apple.com/guide/colorsync-utility/welcome/mac>) tool to see all the ICC profiles that's on your Mac. In an ICC profile, the data under the `chad` tag is essentially the chromatic adaptation matrix from the color space's original reference white (e.g. D65 for sRGB) to D50, which is the adaptation state of the PCS.

Also, ICC specifies D50 as the reference illuminant for all its working space (and the PCS). Why (<https://ninedegreesbelow.com/photography/xyz-rgb.html#ICC>)? *"Because the ICC is heavily oriented toward facilitating the making of prints on paper, and D50 is the preferred reference white for evaluating prints on paper."* Of course, using D50 as the reference illuminant in a working space doesn't mean that only D50 can be used as the viewing illuminant in practice. It simply means that the associated tristimulus values are pre-adapted to D50. For instance, if you take a look at the sRGB ICC profile, the XYZ values of

the red primary is $[0.436, 0.222, 0.014]$. If you calculate the chromaticities, you will see that they match the D50-adapted R primary above. See Part D5 of this article (<https://ninedegreesbelow.com/photography/xyz-rgb.html#ICC>) by Elle Stone for more details.

Android. Modern software that deal with images (and OSes) do perform color management. If you ever need to develop applications that deal with images and photos, make sure you perform color management, too. Android, for instance, provides APIs for color management starting from Android 8.0 (API level 26). This allows you to render colors in non-sRGB color spaces (<https://developer.android.com/reference/android/graphics/ColorSpace.Connector>) (usually wider gamuts) through correct color space transformations (<https://developer.android.com/reference/android/graphics/ColorSpace.Connector>).

4. Usecase C: Color Correction in Camera Imaging

Another interesting usecase of chromatic adaptation is to help obtain the color correction matrices (<http://www.strollswithmydog.com/determining-forward-color-matrix/>) in the camera imaging pipeline under different illuminants. Note that color correction by itself is not the same as chromatic adaptation, but the latter can make the former easier.

4.1 What is Color Correction?

Briefly, the spectral sensitivity functions (in RGB) of any camera are most likely different from the color matching functions of any known color space. Therefore, it is necessary to convert the raw tristimulus values captured by the camera from the camera-native color space to a known color space, say XYZ, for subsequent processing. This transformation is also called color correction, and is typically done through a linear transformation, just like how most color space conversions are done.

Determining the transformation matrix, a.k.a., color correction matrix (CCM), is done as follows. First we take a picture of a set of color patches, whose spectral reflectances function (SRF) are known, under a given illuminant, say D50. A common set of color patches are the Macbeth ColorChecker (<https://en.wikipedia.org/wiki/ColorChecker>), which contains 24 different patches. For each color patch i , we 1) read the corresponding raw tristimulus values $[R_i, G_i, B_i]$ in the camera-native color space (after demosaicking) and 2) calculate its XYZ tristimulus values $[X_i, Y_i, Z_i]$ using the illuminant's SPD and the patch's SRF.

An ideal CCM would perfectly convert each $[R_i, G_i, B_i]$ to the corresponding $[X_i, Y_i, Z_i]$, hence the equation below:

$$\begin{bmatrix} X_1, & \dots, & X_{24} \\ Y_1, & \dots, & Y_{24} \\ Z_1, & \dots, & Z_{24} \end{bmatrix} = \begin{bmatrix} T_{00}, & T_{01}, & T_{02} \\ T_{10}, & T_{11}, & T_{12} \\ T_{20}, & T_{21}, & T_{22} \end{bmatrix} \times \begin{bmatrix} R_1, & \dots, & R_{24} \\ G_1, & \dots, & G_{24} \\ B_1, & \dots, & B_{24} \end{bmatrix}$$

Of course color correction isn't going to be perfect (because the camera spectral sensitivities have to be designed with a set of practical constraints), our goal here is to find the CMM to minimize the conversion error. This is formulated as a linear optimization problem (e.g., linear least squares). If the CCM is perfect, that is, the camera raw color space is precisely a linear transformation away from XYZ, then the camera is said to satisfy the Luther Condition

(https://en.wikipedia.org/wiki/Tristimulus_colorimeter#:~:text=A%20camera%20or%20colorimeter%20is,of%20the%20filters%20is%20a) and that the camera space is colorimetric (in that we can use the camera to measure color).

4.2 The Challenge

So far so good. The challenge, however, is that the CCM depends on the illuminant under which the color patch image is taken. This is because the illuminant dictates the raw tristimulus values taken by the camera as well as the XYZ tristimulus values. Modern cameras usually calculate (pre-calibrate) different CCMs for a set of common illuminants. When the capturing illuminant is different from the preset illuminants, the camera imaging pipeline would usually interpolate the CCM

(https://openaccess.thecvf.com/content_cvpr_2018/papers/Karaimer_Improving_Color_Reproduction_CVPR_2018_paper.pdf) using the CCMs of the pre-calculated illuminants.

So if we want to consider, say, N different illuminants, we would need to calculate N CCMs. This is conceptually easy by taking N different measurements, one for each illuminant. The interesting question is what if you don't know the spectral reflectances of the color patches, which we need to get the "ground-truth" XYZ tristimulus values for a particular illuminant?

4.3 Chromatic Adaptation to the Rescue

Here is the interesting thing. The color patches on the Macbeth ColorChecker (<https://en.wikipedia.org/wiki/ColorChecker>) are chosen "*to have consistent color appearance under a variety of lighting conditions.*" To what extent this is true is debatable, but for some illuminants that are not far apart (e.g., D50 and D65) we can roughly assume that color patches have similar color appearance under them. What this means is that for each patch, given the XYZ values under one illuminant, we can predict the XYZ values of a color patch under a different illuminant using chromatic adaptation without knowing the spectral reflectance of the color patch!

As an example, Section 2.1.5 in this

(https://www.babelcolor.com/index_htm_files/A%20review%20of%20RGB%20color%20spaces.pdf) article from BabelColor uses chromatic adaptation to predict the XYZ values of a color patch (Blue Flower) in the ColorChecker under D50 given the XYZ values of the same patch under D65. This is feasible because the color patch is carefully picked such that it has the same color appearance under D50 and D65. Essentially, this becomes a cross-media color reproduction problem (usecase 1), where we calculate the XYZ values viewed under D50 that would match the color appearance of the given XYZ values viewed under D65.

Since the patch maintains a constant color appearance under both illuminants, the actual measured XYZ values under D50 match very closely the predicted XYZ values through chromatic adaptation, as shown in the article ($[24.60, 23.40, 34.54]_{predicted}$ vs. $[24.64, 23.41, 34.43]_{measured}$).

A small detail to note is that the BabelColor article uses Bradford chromatic adaptation, which is conceptually the same as von Kries chromatic adaptation albeit with differences that make it more reliable. But you get the idea.

A big caveat here: using chromatic adaptation to estimate the XYZ values of an illuminant from the XYZ values of another illuminant is just a rough estimation, because it assumes that the color appearance of the object is the same between the two illuminants, which is sort of OK if the two illuminants are similar, e.g., D65 and D50, but generally not true; you would not produce very precise results if you want to get XYZ under A from D65. The ability of an object to maintain constant color appearance under illuminants is measured by the Color Inconstancy Index (<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1478-4408.2003.tb00184.x>) (CII). As you can see, the Blue Flower patch even under the very similar D50 and D65 illuminants would still have a non-zero CII since the predicted and measured XYZ under D50 are not the same.

5. Usecase D: White Balance in Camera Imaging

A key challenge in digital camera imaging is white balance, which arises from the fact that the capturing illuminant Φ_c when a photo is taken by the photographer is most likely different from the viewing illuminant Φ_v when the photo is viewed by a viewer. Why does this matter? The photographer is adapted to Φ_c while the camera, without care, would not adapt and simply tries to accurately capture the XYZ/LMS values of the scene points (which more or less is the case if we have a good color correction mechanism); when the photo captured by the camera is presented to a user, the user will adapt to Φ_v . Therefore, the user will not perceive the color the same as what the photographer perceives when the photo is taken.

5.1 White Balance Using Chromatic Adaptation

The goal of white balance is to adjust the raw camera RGB values in such a way that the photo, when presented to the viewer under Φ_v , will have the same color appearance as that of the original scene under Φ_c . This means we will have to perform a chromatic adaptation from Φ_c to Φ_v . The only complexity is that chromatic adaptation needs to operate in a colorimetric space, but the raw camera values are not in a colorimetric space --- we have to perform a color correction first (as described before)!

Let $[R, G, B]^T$ denote the raw camera RGB values (before color correction), the following equation generates the linear sRGB values $[R_v, G_v, B_v]^T$ that will have the same color appearance, when viewed under Φ_v , as the scene viewed under Φ_c :

$$\begin{bmatrix} R_v \\ G_v \\ B_v \end{bmatrix} = M_{xyz-srgb} \times M_{\Phi_c-\Phi_v} \times T_{cam-xyz} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

where $T_{cam-xyz}$ denotes the CCM of the camera. Note that we are not considering observer metamerism here. That is, we assume that the cone responses of the photographer and the viewer are the same. If not, the matter will be much more complicated, which is beyond the scope of this article.

5.2 Auto White Balance, a.k.a., Estimating Capturing Illuminant

Looking at the equation above, to use the precise von Kries chromatic adaptation for white balance, we need to know four things: 1) the raw camera RGB values, 2) the CCM (i.e., $T_{cam-xyz}$), 3) the viewing illuminant Φ_v , and 4) the capturing illuminant Φ_c ; The latter two collectively determine $M_{\Phi_c-\Phi_v}$.

Let's focus on the 4th item, the capturing illuminant, for now. Perhaps the most difficult part in white balancing is to estimate the capturing illuminant Φ_c , which is usually quite difficult to know precisely. In fact, the main job of an auto white balance (AWB) is to (implicitly or explicitly) estimate the capturing illuminant. Some AWB methods use a separate sensor that measures the illuminant and keeps the illuminant as the metadata. Others use pure computational methods/heuristics. Perhaps not surprisingly people even train Deep Neural Networks to estimate capturing illuminant (<https://bmvc2019.org/wp-content/uploads/papers/0105-paper.pdf>).

Alternatively, we could manually estimate the capturing illuminant -- if we are lucky. If the scene that you are capturing happens to contain a (near) neutral point such as a piece of white paper or a white wall, then the XYZ values of the illuminant should be the same as the XYZ values of the neutral point as captured by the camera, which we can estimate from the raw camera RGB values of the neutral point. This is why cameras (and smartphone cameras with the help of professional camera Apps) will allow you to manually specify a white point. This helps estimating the capturing illuminant immensely.

5.3 White Balance and Color Correction --- Which Comes First?

Strictly speaking, color correction should be performed before white balance, because white balance should operate in a colorimetric space (e.g., LMS or XYZ). However, many cameras including DSLR and smartphones perform white balance before color correction, where white balancing is done by scaling the raw RGB values. How can this be correct? You can read this excellent article (<https://www.spiedigitallibrary.org/journals/optical-engineering/volume-59/issue-11/110801/Color-conversion-matrices-in-digital-cameras-a-tutorial/10.1117/1.OE.59.11.110801.full?SSO=1>) (Section 5) by Andrew Rowlands for details, but long story short what cameras do is to re-express the equation above as:

$$\begin{bmatrix} R_v \\ G_v \\ B_v \end{bmatrix} = R \times D \times \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

where:

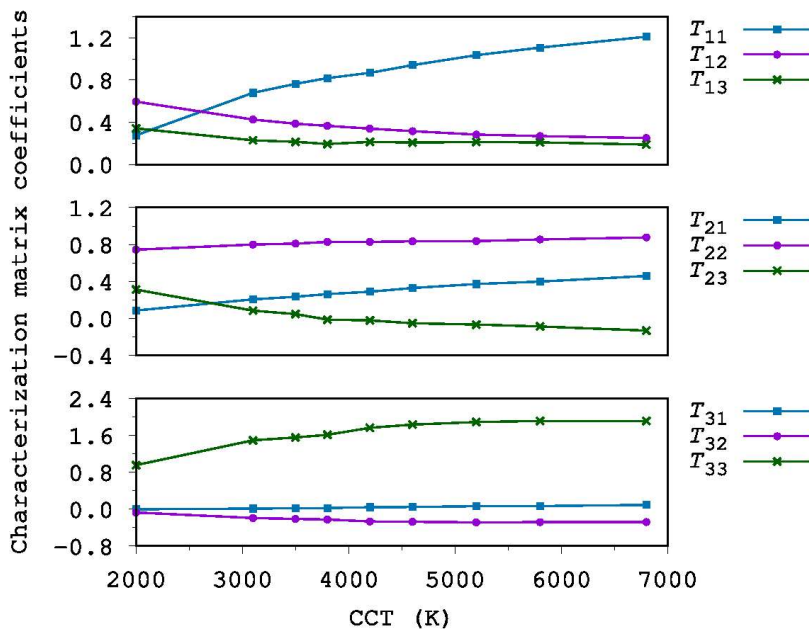
$$M_{xyz-srgb} \times M_{\Phi_c-\Phi_v} \times T_{cam-xyz} = R \times D$$

D is a diagonal matrix that scales the raw RGB values, and R is derived accordingly. The diagonal matrix D is calculated such that the capturing illuminant's raw RGB values is transformed to $[1, 1, 1]$. Since after chromatic adaption the scene illuminant will become $[1, 1, 1]$ in linear sRGB as well, each row in R sum to 1. That is, R transform $[x, x, x]$ to $[x, x, x]$ for arbitrary x , i.e., rotating around the gray axis. That's why Rowlands call R a rotation matrix.

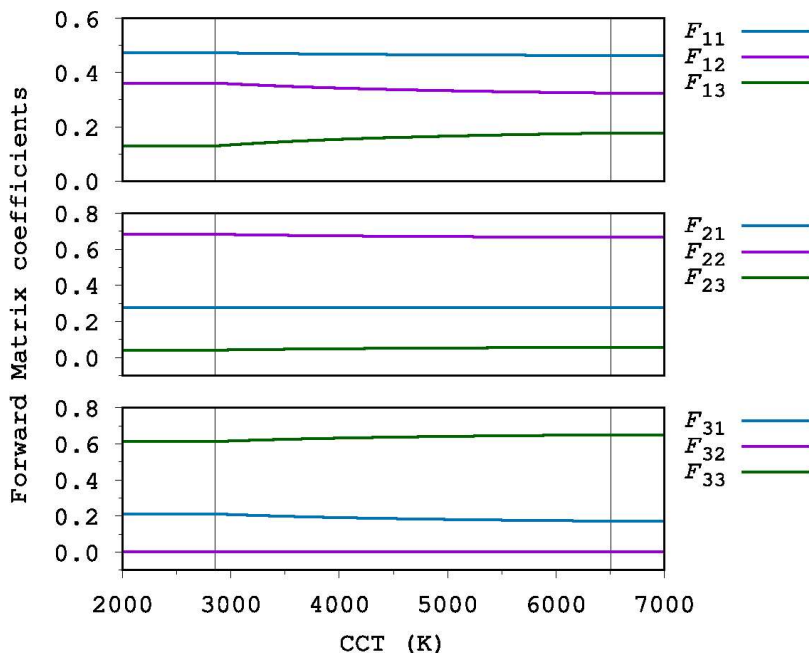
Digital camera photography literature calls scaling the raw RGB values by the diagonal matrix D "white balance" and transformation by matrix R "color correction", which you now should know is technically incorrect or at least confusing. Both matrices are encoded in, or derivable from, raw images (e.g., Adobe's DNG format). For instance, you can retrieve R and D using the RawPy (<https://letmaik.github.io/rawpy/api/rawpy.RawPy.html>) Python package from `camera_whitebalance` and `color_matrix` parameters, respectively.

According to Rowlands, there are two main incentives for this mathematical re-expression. First, the CCM is only approximate and contains error: "In particular, color errors that have been minimized in a nonlinear color space such as CIELAB will be unevenly amplified, so the color conversion will no longer be optimal." White-balancing in raw RGB minimizes the impact of color correction errors on white balancing.

Second and more importantly, doing white balance in RAW allows the subsequent color correction to be less dependent on illuminant, which simplifies camera design and improves color rendering quality. Recall that the traditional CCM is heavily dependent on the illuminant (see Figure 5 in Rowlands' article (<https://www.spiedigitallibrary.org/journals/optical-engineering/volume-59/issue-11/110801/Color-conversion-matrices-in-digital-cameras-a-tutorial/10.1117/1.OE.59.11.110801.full?SSO=1>), which I copy and paste below), and online interpolation is needed if the capturing illuminant is very different from the pre-calibrated illuminants, which typically need to be a lot in order to minimize the interpolation error.



It turns out that if we were to scale the raw RGB first (e.g., white balancing in raw), the new CCM will be much more stable under different illuminants (see Figure 8 in Rowlands' article (<https://www.spiedigitallibrary.org/journals/optical-engineering/volume-59/issue-11/110801/Color-conversion-matrices-in-digital-cameras-a-tutorial/10.1117/1.OE.59.11.110801.full?SSO=1>), which I again copy and past below). Compare this figure with the figure above; you can see that the coefficients in the new CCM have much less variation across illuminant CCTs. That way, cameras can afford to have CCMs for just a few pre-calibrated illuminants without online interpolation.



According to Rowlands, traditional cameras and DCRaw (<https://dechifro.org/dcrow/>) perform white balance (in raw) before color correction while smartphone cameras conventionally take the other approach, although there is evidence (<https://karaimer.github.io/camera-pipeline/>) that smartphone cameras are increasingly taking the former approach too. Adobe's DNG converter (<https://helpx.adobe.com/photoshop/using/adobe-dng-converter.html>) provides both modes.

6. Summary

Many real-world usecases deal with color appearance, ranging from color management to digital camera imaging. In simple terms, color appearance = colorimetry + viewing condition. Illuminant is one of the most important aspects of viewing condition. Chromatic adaptation provides a way to take into account the effect of the illuminant. When we want to specify a particular color appearance, it's important to specify both the XYZ values and the illuminant.

Acknowledgement

Hao Xie (<https://www.linkedin.com/in/hao-xie/>) from RIT meticulously went through the technical details of this article. Mike Murdoch (<https://sites.google.com/view/michaeljmurdoch/>) from RIT helped me better understand Apple's True Tone display. Andrew Rowlands (<http://www.andyrowlands.com/index.html>) patiently answered my questions on the interactions between white balance and color correction.