Wednesday, July 18, 2018

12:53 PM

The Basics of Monitor Calibration

LAST UPDATED ON APRIL 8, 2018 BY [GUEST POSTER](https://photographylife.com/author/guest) [208 COMMENTS](https://photographylife.com/the-basics-of-monitor-calibration#comments)

The subject of monitor calibration and profiling can be quite difficult to understand not only for a beginner, but also for professionals working in the field. With so many different hardware and software components, color profiles, bit depth and other related terminologies, one can get quickly confused and lost, potentially ending up with a rather poor working environment. Having a badly-calibrated monitor is not only counter-productive, it is also potentially harmful for one’s business, especially when dealing with paying customers and clients. Due to the complexity of the topic, our team at Photography Life requested help from a real expert, who will be providing detailed information on how to properly calibrate monitors for photography needs. But first, some basic concepts need to be understood. This particular article is just an introduction to cover the basics of calibration and profiling, without going into too many technical details.

1) A Very Brief History

As you might already know, monitors, TVs, mobile devices, etc., can show us colors using a mixture or Red, Green and Blue (RGB) light. Common monitors try to cover a minimum standard color space known as “[sRGB](https://en.wikipedia.org/wiki/SRGB)” with their red, green and blue emitted light. The Internet and most computer content is meant for this particular color space. For historical reasons, sRGB and other similar color spaces like [Rec.709](https://en.wikipedia.org/wiki/Rec._709) cover the same gamut (subset of visible colors) as CRT monitors. This color space is not able to cover colors printable with current technology like offset printing or a domestic inkjet printer – there are colors like cyan-turquoise green that are printable with such devices, but cannot be shown on an sRGB monitor. That’s the main reason that leads professionals and photo hobbyists into seeking monitors with a wider gamut which covers a large percentage of color spaces like [AdobeRGB 1998](https://en.wikipedia.org/wiki/Adobe_RGB_color_space) or [eciRGBv2](http://www.eci.org/en/colourstandards/workingcolorspaces).

2) Color Management and Color Coordinates

The first thing photographers need to know is that their wide-gamut monitors are meant to be used in color-managed applications: applications that work in a color managed environment. For example, you have an sRGB 300×300 JPEG image that is just a green background (RGB values “0,255,0” in sRGB). With a common monitor (sRGB monitor) you can output its contents to the monitor directly, without conversions or color management, and you will see the green color that is “fairly close” to color information stored in that JPEG file. But if you do the same thing in a wide-gamut monitor configured to show its full gamut, that “0,255,0” RGB value will show native gamut green 255 and it will look over-saturated. This is where color management comes into play: if applications know the actual gamut of that monitor, they can translate this “0,255,0” sRGB value to another set of RGB values that represent the same color (or fairly close) in a bigger color space:

*sRGB 0,255,0 (green) -> 144,255,60 AdobeRGB (same sRGB green color)*

Such number transformations are possible, because colors (visible colors seen by humans) can be defined objectively as coordinates in a color space that covers human vision, like the CIE 1931 color space. There are several coordinate choices that map to color spaces like CIE 1931 XYZ (or just CIE XYZ onwards), which is a 3D coordinate system for visible colors with X, Y and Z coordinates.

Measuring color in CIE 1931 XYZ coordinates (the most used color coordinate system for measuring) is about weighting the spectral power distribution (SPD, distribution of how much light is coming to measurement device for each visible wavelength) against a “model” of human vision called CIE 1931 2º standard observer (or just “standard observer” to keep it short). [Wikipedia](https://en.wikipedia.org/wiki/CIE_1931_color_space) has a very good definition of CIE XYZ and where X, Y & Z coordinate values come from.

Like our world, a city is a 3D space: north-south, west-east, but also an up-down location of a building. A city may be a 3D space, but we find it useful to represent a city in a 2D plane, like a paper map with north-south, west-east locations. A similar approach is CIE xyY color space, derived from CIE XYZ. In that CIE xyY color space, XYZ values (3D coordinates) are normalized to lowercase x,y,z values with the condition x+y+z=1, a scale conversion. Since a Y coordinate (capital Y) is kept in CIE xyY (it’s a 3D coordinate system after-all), original XYZ values can be restored. CIE xy coordinates (without capital Y) represent a 2D plot of CIE 1931 XYZ color space, like a city map… and like in a 2D city map some information is discarded, but we get a picture of locations quickly. The concept of this CIE xy 2D plot (or other 2D plot of a 3D color space) is important for the next articles. There are other common 2D projections of other 3D color coordinate systems like CIE u’v’.

Another color coordinate system derived from CIE XYZ is CIE L\*a\*b\*. It has 3 coordinates (3D color space), L\* for luminance, a\* for a green-magenta axis and b\* for blue-yellow axis. It takes a reference CIE XYZ white for its definition so L\*=100, a\*=b\*=0 are the coordinates of reference white.

Since a 3D coordinate system for color does not carry information about actual SPD of the source, two light sources with different SPD may have the same coordinates, their SPD weighted against standard observer gives the same numbers. We see them with the same color. That’s called a metameric pair. That’s why we can capture colors in cameras and view them with colors close enough on a computer screen: different SPDs may have the same color coordinates.

Two pieces of paper or fabric may have the same color coordinates under some lighting conditions. That means SPDs of the reflected light weighted against standard observer are equal or close enough, but if we change the SPD light source, then reflected light SPD changes too, so color coordinates of each sample may drift away. That mismatch is called illuminant metameric failure: for one light source there is a color match but for the other there isn’t.

It may be possible that a human subject visual system has a different enough response from standard observer response. In that case, actual color coordinates (colors) “seen” by each observer will be different. This is called observer metameric failure. Human visual system response varies between subjects and with age, but a very huge percent of them are close enough to standard observer response: that means standard observer is a very good model. There is a limitation though: very narrow spikes in SPD from a light source (like a laser) will make those tiny differences between you and standard observer noticeable, but this is not a real issue for current WLED (sRGB) or GB-LED (wide-gamut) technology used in monitors, so don’t worry about it.

There are other metameric failure sources, but please note that metameric failures and metameric pairs are defined over pairs: two samples, two observers, two light sources…

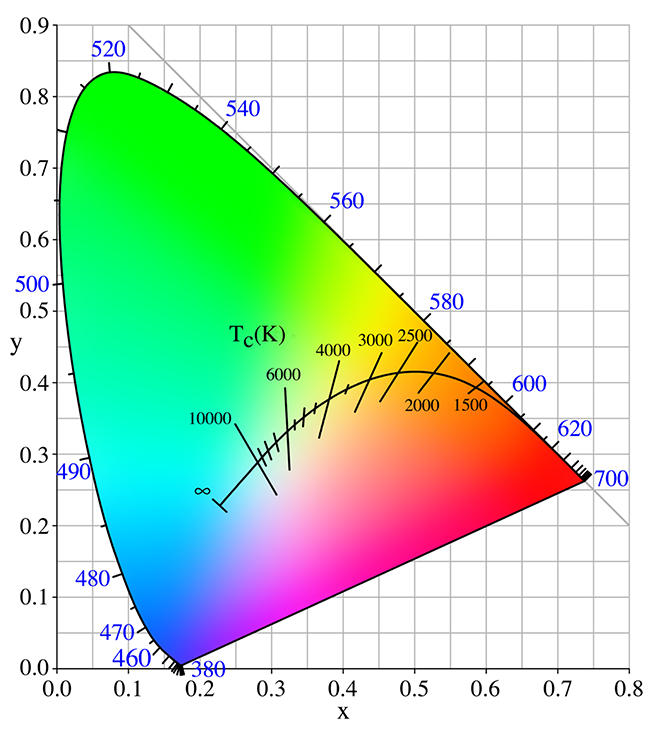
3) Color Distance

With coordinate systems for color, we are able to define a distance between colors, in the same way we define distance between points in 3D space or on a 2D map. The most useful of these distance definitions are not equal to “Euclidean Distance” (Pythagorean Theorem) of CIE XYZ coordinates, but a modified version to deal with the way humans perceive colors: a “perceptually uniform” definition of color distances, a distance definition that our eyes perceive as “equal distance jump” between neighbor colors. That distance is expressed in terms of deltaE units (dE). There are several revisions of these distance definitions as we acquired better understanding of human vision. The more common of these distances in order of increased accuracy are: dE76, dE94 and dE00, named after their year of definition. The most accurate is deltaE2000 (dE00) and dE76 is just Euclidean Distance of CIE L\*a\*b\* color space.

4) Color Correlated Temperature

Instead of using the proper way to name a color by its color coordinates, “whites” are usually addressed by a “color temperature” term expressed in degrees kelvin (K). In physics, there is an idealized physical body that radiates a spectral power distribution (SPD, energy distribution across wavelengths) related to its temperature (this is a very simplified version). A blackbody at 3000K radiates an SPD that we see as warm orange-red and at 8000K radiates an SPD we see as bluish white. Think of it as a model of incandescence, like if you have a forge and you start to warm up a piece of metal with fire until it glows. The lower blackbody temperature is, the “warmer color” (yellow-orange-red) we see. The higher blackbody temperature is, the “cooler color” (blue) we see. Color coordinates that match color from a blackbody SPD at its different temperatures is called blackbody locus on a 2D plot (like CIE xy) and it is a curve.

In the same way, we may define an SPD of daylight in its different warmer-cooler tones, each of which have color coordinates in CIE xy plane that when plotted together form a curve: daylight locus. Some of these daylight whites have specific names like “D65” for 6500K daylight SPD or D50 for 5000K daylight SPD and their color coordinates are a very common calibration target for monitors.



The segments that crosses blackbody locus curve have the same Correlated Color Temperature (CCT). Daylight locus runs as a parallel curve moved a few dE units towards cyan

Illustration courtesy of Wikimedia Commons

But we may have a “white” that does not exactly meet blackbody or daylight behavior, it’s just “near” them. We may define for such whites a “color correlated temperature” (CCT for blackbody, CDT for daylight), the color temperature of the closest white in those loci. Color correlated temperature is an indication of how yellow-blue (warm-cool) a white is, but it does NOT give us information on how far it is from blackbody or daylight loci, how “green” or “magenta” it is. You need to add to that color temperature a distance term, how far in dE terms it is from one of those loci. This concept is very important: correlated color temperature is not sufficient to give us information about whites – with a CCT or CDT we do not know how magenta or green a white point is, only how yellowish-bluish it is.

5) Profiles

Color managed applications need to know what the actual behavior of a monitor is, so that they can send proper RGB numbers for THAT monitor in order to show a color stored as RGB numbers in a defined color space inside an image or a photograph. To solve this problem we have “ICC profiles”. Other devices like scanners, printers, etc., use profiles to describe their behavior too.

To keep it simple, a monitor ICC profile (or just “profile”) is just a file with “.icm” or “.icc” extension that stores monitor color behavior for a specific configuration. Among other things we find the following in a monitor profile:

* Gamut: which are CIE XYZ coordinates of “full” red, green and blue of such monitor in its current state, the location of its primary colors.
* White Point: what the CIE XYZ coordinates are when monitor outputs white (“full” red, green and blue at the same time) in its current state.
* Tone Response Curve (TRC), also called gamma. This is a plot of how brightness rises (relative to full maximum output) as you send a bigger R, G or B input value to monitor from zero input to its full input value… in its current state. There is one TRC per RGB channel and it can be equal for R, G and B. A monitor with “true neutral grey” for all grey values (referred to a certain white) should have a red TRC = green TRC = blue TRC. It does not matter what the actual white is, since these TRCs are defined relative to each channel max output, not to actual cd/m2 output per channel.

There are several ways to store that information, which gives different types of profiles: matrix profiles, cLUT/table profiles… The simplest way is a matrix profile with 3 equal TRCs, this assumes that the monitor has a nearly ideal behavior. The more complex way to store that information is in a table with XYZ color coordinates for several RGB input values and 3 different TRC for each R, G and B channel, in order to capture any non-ideal behavior.

Profiles for different devices may have different white points or gamuts and it is not useful that every profile knows how to transform its own RGB coordinates to every other profile RGB coordinates with a given rendering intent. It is more useful to transform the RGB coordinates of a profile to a common neutral ground where color managed applications do the transformation from and to that neutral ground. This neutral ground is called Profile Connection Space (PCS). It has a big gamut equal to visible colors (whole CIE XYZ gamut, for example) and it usually has D50 as a reference white.

Each profile has information about how to transform its own coordinates from or to PCS for some rendering intents. Matrix profiles have information for only relative colorimetric intent transformation.

6) Rendering Intents

Color management also states a set of rules about what RGB numbers should be sent to a device (monitor, printer, etc) when a color defined as RGB numbers in a color space falls outside the device’s gamut. That color cannot be shown as intended, but a set of recommendations known as “rendering intents” deal with this situation in a more or less predictable way.

Some of them are:

* Absolute Colorimetric: This intent aims to show in-gamut colors as they are, clipping colors that the device cannot show.
* Relative Colorimetric: It is akin to absolute colorimetric, but when the two color spaces involved have different whites, gamut and its colors are “moved” from one white to the other. Color management is “relative” to each color space white.
* Perceptual: It is similar to relative colorimetric, but out of gamut colors are moved in-gamut, pushing or “deforming” inwards the already in-gamut colors. Although this preserves tonal relations in gradients since there is no clipping of out of gamut colors, in-gamut colors may not be shown as intended.

If in doubt, relative colorimetric is the safest choice: show me the colors that my device can display, with its current white point, the right way.

7) Calibration

Sometimes a monitor needs to be configured for a very specific white (CIE XYZ white point coordinates) or a tone response curve behavior (neutral grey and a specific gamma value) or to a certain brightness. Since white is the sum of red, green and blue output, we can lower the max brightness value of each channel until the white output matches our desired color of white (white point). In the same way we can vary the “middle” red, green and blue output, so that the resulting greys have the same color as white (so that they are neutral relative to white) or to have a specific brightness for each grey (gamma). We could write this information as a table: for each red input number to the monitor from zero to full input (table input) we want a specific red brightness, so we calculate which input number for the red channel behaves in that way (table output). The same applies to green and blue.

This process is known as calibration, to make a monitor behave in a particular way (or close to it). Information of what red, green and blue “numbers” should be fed to a device so we get the grey colors we want to show on that device, is called “calibration curves”. There is one per channel and they may be used to correct the white point too, since a monitor’s white is just the brightest of its grays.

Most monitors have button controls to lower maximum light output of R, G and B channel (you known them as “brightness”, “contrast” and “RGB Gain” controls), so white point can be fixed inside a monitor, without the help of external tools. Other displays cannot do that, because they lack such controls (like laptops).

A few monitors allow changing its grey response because they are able to store at user command a set of custom calibration curves in their own electronic components. If a monitor has such a feature, we say it has “hardware calibration”, because it has a LUT (Lookup Table) to store calibration curves. For monitors without such a feature, almost every graphics card (GPU to keep it short) inside a computer has a LUT for each DVI, HDMI or DisplayPort(DP)/Thunderbolt output.

Since we output discrete RGB numbers to a monitor, usually from 0 to 255 for each channel, and since a monitor accepts a discrete RGB number as input, usually from 0 to 255, then if we modify this one-to-one translation with a calibration curve, we may be introducing “gaps” or “jumps” in that 256 step stair. Such gaps may result in visible jumps between neighbor grey values and even coloration of some grays (red, green or blue tint in them). The bigger the gap, the more noticeable it is. It does not matter where those calibration curves are stored (inside the monitor or in a graphics card LUT) – a one-to-one transformation modification of a discrete value to another may result in noticeable gaps.

To avoid these issues, there is a mathematical tool known as “temporal dithering”. The basic concept is to use “time” to solve lack of step resolution, so it’s possible to create a “visual step” in the middle of the gap. For example, calibration curve says that for “128,128,128” RGB input number sent from a computer, the monitor should work as if “128.5, 123.75, 129.25” values were the actual input, in order to get a neutral grey with a desired brightness. If a monitor (or its internal components) only accepts discrete values from 0 to 255 in steps of 1 unit, not decimals, then rounding such transformation to “128,124,129” may result in an excess of green, a green tint for that grey, a gap or band (hence the term “banding”) in a grey gradient from black to white because of this rounding error. With the help of temporal dithering, we can “move” these “decimal values” to time with a device that only accepts 1 unit per step as input, just changing value for each time step, so overall value across a time interval will be our desired value, with decimal values. For example, let’s take an interval sequence from t1 to t4 for the same grey correction:

*t1:”128,124,129” -> t2:”129,124,129” -> t3:”128,123,130” -> t4:”129,124,129”*

Like in cinema, tiny time steps (fast enough frames) are not noticeable and our eyes perceive it as if the monitor (or its internal components) was fed with intended correction “128.5, 123.75,129.25”.

Monitors with hardware calibration have LUTs capable of storing high bit depth calibration curves (more than 8-bit, more than 256 steps, with “decimal values”) even if the input to the monitor is limited to 256 steps. With the help of temporal dithering, electronic units can output calibration to lower bit depth electronics (lower than LUT, without “decimal values support” like for example 8-bit – 256 steps) in a smooth way, without gaps. This results in smooth gradients thanks to high bit depth LUT AND dithering.

Monitors without hardware calibration need a graphics card with a high bit depth LUT and temporal dithering units in order to achieve the same thing. The ugly part of the tale is that more than half of graphics cards cannot do that: NVIDIA GeForce series and Intel Integrated Graphics cards cannot do it, so every calibration curve different from “no translation”/”no calibration” may result in awful banding artifacts. The bigger the gap in calibration curves, the more noticeable banding will be.

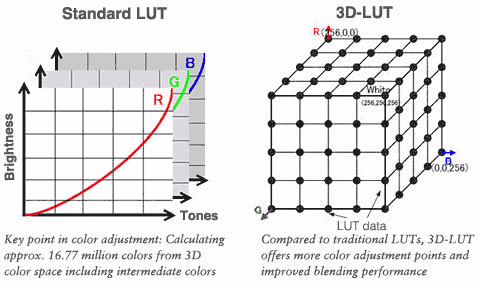
If you want (or are forced) to use graphics card calibration, it is HIGHLY recommended that you get an AMD/ATI graphics card (gamer “Radeon” or professional “FirePro”) or NVIDIA Quadro graphics card (professional market). This is the only way to avoid every kind of banding artifacts caused by calibration curves loaded in graphics card LUT.

Keep in mind that since monitors with hardware calibration are plugged to a graphics card, their behavior could be modified too by calibration curves stored in graphics card LUT (GPU LUT to keep it short). ICC profiles contain a tag called VCGT (video card gamma table) with calibration curves table that must be sent to graphics card LUT. For hardware calibration capable monitors, their ICC profiles contain a linear input=output calibration curve, so no graphics card calibration is applied when that ICC profile is active.

8) 3D LUT Calibration

Calibration curves allow us to get a desired white, with a desired TRC and neutral grey… but do not substantially modify the gamut of a monitor once applied to it. With wide-gamut monitors, it will be desirable in some situations to make them work like a common sRGB monitor for non-color managed applications, a gamut SMALLER than its native / full gamut. Since sRGB is a smaller gamut contained INSIDE wide-gamut monitor’s full gamut, sRGB colors are reproducible in those monitors: sRGB colors are just a combination of wide-gamut monitor’s R, G and B native values, a subset of its possible R, G, B values.

This could be seen as a table: for each R, G and B value of a smaller color space (like sRGB) we could write other R, G and B values which represent the same color in our wide-gamut monitor’s full gamut color space. Having 3 coordinates for each input, that table is “3D” in its inputs, hence the name “3D LUT”.



So a 3D LUT can “emulate” a color space smaller than or equal to the monitor’s native gamut color space. We call such 3D LUT calibration “emulated color space”. Hence we call “emulated sRGB” to a 3D LUT calibration that makes a wide-gamut monitor behave like a common sRGB monitor. A monitor could emulate other color spaces too, like AdobeRGB, DCI-P3, etc., even mimic other device’s behavior. This is an important feature, because such emulated color spaces as sRGB or Rec709 could be used without color management to display content that is meant to those color spaces (like non-color managed Internet browsers, or to output HDTV/DVD/BR content in a non-color managed video player program).

If such a table stored every 256 step R x G x B combination, it would result in a HUGE table with millions of entries. In order to simplify it, less than 256 steps per channel need to be taken, interpolating the other values between those steps. For example a 17x17x17 3D LUT results in less than 5000 entries, smaller than millions of entries with a 15=256/17 step between entries. Such 3D LUT assumes small and smooth variations of uncorrected monitor behavior. For example, if there was a big undesired behavior between step 6 (102/256 value) and step 7 (119/256 value), let’s say in 110 value, but such undesired behavior does not happen at 102 value nor 119…a 17x17x17 3D LUT cannot correct it. Such error correction “does not exist” for a 3D LUT with that step value between entries.

Since most calibrations aim for a neutral grey ideal behavior of a smaller or equal gamut than native gamut, we could simplify a 3D LUT to be small but to store correction for each input of a channel. This is done with a pre-LUT, matrix and post-LUT structures:

* pre-LUT and post-LUT are each just 3 usual LUT (like graphics card LUT) for calibration curves, one per channel, so there are 6 tables, 3+3.
* matrix is a way to express desired red, green and blue primary colors (gamut) in a combination of the full gamut of the monitor.

Most monitors with 3D LUT calibration use this approach: small, fast and accurate. Some high-end 3D LUT calibration systems allow clipping (relative colorimetric intent) when dealing with bigger than native gamut color spaces. For example Rec.2020 is a HUGE color space that usual wide-gamut monitors cannot cover at 100%. If we want to feed such a wide-gamut monitor with Rec.2020 content in a non-color managed environment, it is possible (if some hardware and software requirements are met) to write a 3D LUT calibration which shows Rec.2020 colors properly if they fall inside our monitor gamut, but clip Rec.2020 colors that cannot be shown with the device (out of gamut colors).

9) Uniformity

An ideal monitor should output the same color and brightness response for each of its pixels – it should be perfectly “uniform”. In a real world device, there are some deviations from this ideal uniformity. Since color could be objectively described with coordinates (CIE XYZ), there is a way to objectively express color difference between different zones of the monitor screen. The easiest way is to use deltaE2000 distance, but it stores distance in “color tint” and brightness in one number. It may be desirable to split that distance in brightness and “tint”, the latter being worse for non-uniformity: green or magenta ugly tints on monitors sides or corners. If you do not care about the actual color (hue) of the “tint” of the less uniform zone of screen and you just care about “how huge” (how bad and noticeable) it is, that partial color distance could be expressed in terms of deltaC distance.

With these uniformity deviation values, brightness and deltaC, we can express how bad color uniformity is for a display, in an easy to understood way. Keep in mind that there are several distance definitions and several ways of describe uniformity problems – this is just one of them. ISO norm 12646 in each of its revisions states has its own way of defining uniformity requirements in a PASS/FAIL test. These requirements are not met by most cheap and affordable monitors; a very large amount of them will get a FAIL test result.

But for most hobbyists and even some professionals with a more limited budget, a lower than 10-15% brightness variation from center and lower or equal than 2 deltaC “tint” variation from center are easier to meet and they are “good enough” (your mileage may vary). Bigger than 20% brightness variation and more than 3-4 deltaC variation from center should not be accepted for a monitor intended for image/photo editing… I would reject a unit with such bad uniformity even for a monitor used for multimedia/entertainment.

Color uniformity in terms of “tint” CANNOT be expressed properly in terms of correlated color temperature, because as seen previously, it does NOT give information about green-magenta deviations from blackbody or daylight white loci. Such correlated color temperature uniformity tests should be avoided for their inaccuracy (i1Profiler software for example is useless for color uniformity evaluation).

10) Measurement Devices

There are several devices on the market for color measurement. A serious discussion about the accuracy, speed and upgrade capabilities of each one of them involves talking about the math of CIE 1931 XYZ color space. Since the target audience of this article is not so technical, these formulas are out of the scope of this text. For further information CIE 1931 XYZ formulas are available online for free. There are lots of resources for those willing to learn the core math about color. So let’s start with a very basic understanding about those devices. The measuring process can be done in two ways and that gives us two types of color measurement devices.

10.1) Colorimeters

Colorimeters use filters placed before the measurement sensor as a way to mimic standard observer behavior. The closer the filter’s response to standard observer, the more accurate the colorimeter is. Old affordable colorimeter models had filters that fade over time (i1Display2, Spyder2, Spyder3), others were not accurate at all (old ones and the new Spyder4 & 5) and some models have very bad inter-instrument agreement (old ones & all Spyders, again) which means that if you buy 2 new units of the same model of these poorly-made colorimeters and test them against the same screen (without changing screen configuration), they may not agree in measurement by a huge margin. That means that the ONLY choice for affordable “non-lab grade” colorimeters is the [X-rite i1DisplayPro](https://www.amazon.com/X-Rite-EOSDIS3-i1Display-Pro-Calibration/dp/B0055MBQOW/ref=as_li_ss_tl?ie=UTF8&qid=1458130135&sr=8-1&keywords=X-rite+i1DisplayPro&linkCode=ll1&tag=photolife0c-20&linkId=07b53d98e1ea05882ea145ddbcf0b581) (also called i1d3) and their more limited brother [Color Munki Display](https://www.amazon.com/X-Rite-CMUNDIS-ColorMunki-Display/dp/B0055MBQOM/ref=as_li_ss_tl?ie=UTF8&linkCode=ll1&tag=photolife0c-20&linkId=18d0bd8803247380cf76472963752612). Munki Display is unable to work with monitor internal calibration software and is about 4-5 times slower, but it is cheaper in comparison.

i1DisplayPro has some superb features like:

* Non-fading filters
* Very fast measurement (not available on color munki display)
* Support for almost every software suitable for monitor internal calibration (not available on Color Munki Display)
* Accurate low light readings
* Works with [ArgyllCMS](http://www.argyllcms.com/) which is the best software for color measurement. It’s licensed under GNU license (free software) but you can actively support its development with a donation (PayPal)
* Stores its spectral sensitivity internally (its own “observer”), so with a more or less accurate sample of each monitor backlight type SPD (WLED, GB-LED…), it’s own inaccuracy can be corrected, because it is known where and how much its observer is different from standard observer. This is a key feature. Spyder 4 & 5 have this feature too, but their major flaws in other aspects make them an unsuitable alternative.

10.2) Spectrophotometers

Spectrophotometers measure the actual SPD data of the light and then internally or with computer software weights SPD data against the standard observer (or whatever observer user wants). It does not rely on the accuracy of filters… but this approach has some drawbacks:

* In order to capture SPD in an accurate way, high spectral resolution is needed, tiny wavelength steps are needed to capture actual SPD without errors.
* Since incoming light is split into different wavelengths and then measured for each wavelength slot (spectral resolution), measurements are noisy and low-light measurements are very noisy. This happens because just a small amount of incoming light arrives at each wavelength step sensor. That implies very slow measurements too, since sensors need more time to capture a certain “valid” (not noise) amount of light.
* Inaccuracies in the wavelength splitting process translate to inaccurate SPD measurement. Actual measurement could be of a shorter or longer wavelength than intended.

Despite these limitations, most of them have a nice feature: they come with a light source to measure reflected light from printed paper (you can profile printers) or fabrics. Affordable non-lab grade spectrophotometers are limited to the old model X-rite i1Pro and its new revision [i1Pro2](https://www.amazon.com/X-Rite-i1Basic-Pro-2-EO2BAS/dp/B007TCITRS/ref=as_li_ss_tl?ie=UTF8&linkCode=ll1&tag=photolife0c-20&linkId=6a7816e8fbbef62cd1ae480fbaf86b39). They are accurate enough devices for printer profiling and have wide software support (ArgyllCMS too with a custom driver). Their optical resolution is not very good for display readings since it is 10nm (3nm step high noise internal readings) and low-light dark color measurements for high contrast displays will be noisy. Anyway, they are able to take actual SPD readings so it’s possible to feed an i1DisplayPro with SPD data for newer or unknown display backlight technologies, providing fast & accurate readings with that colorimeter for every display: the two devices can work as a team to overcome their limitations.

X-rite has another non-lab grade cheap spectrophotometer, Color Munki Photo/Design, but it is an unreliable and inaccurate device with poor inter-instrument agreement. It cannot measure papers with optical brightening agents (OBAs) properly, because its light source has no UV content. It’s a poor performer hardware and like Spyders, it should be avoided.

That means that your choices for display measurement of monitors with hardware calibration are limited to i1DisplayPro and i1Pro/i1Pro2. Since the kind of IPS wide-gamut monitors used for photography and graphic art have very well-known SPD (WG CCFL or GBLED backlight) and those typical SPDs are bundled with i1DisplayPro driver, the natural choice is i1DisplayPro colorimeter. It is cheaper, it will be more accurate than 10nm noisy i1Pro2 readings and it is much faster.

If you need to profile your printer or to measure fabric colors too, you should get the two devices, [i1DisplayPro](https://www.amazon.com/X-Rite-EOSDIS3-i1Display-Pro-Calibration/dp/B0055MBQOW/ref=as_li_ss_tl?ie=UTF8&qid=1458130135&sr=8-1&keywords=X-rite+i1DisplayPro&linkCode=ll1&tag=photolife0c-20&linkId=07b53d98e1ea05882ea145ddbcf0b581) and [i1Pro2](https://www.amazon.com/X-Rite-i1Basic-Pro-2-EO2BAS/dp/B007TCITRS/ref=as_li_ss_tl?ie=UTF8&linkCode=ll1&tag=photolife0c-20&linkId=6a7816e8fbbef62cd1ae480fbaf86b39)(or i1DisplayPro and a used i1Pro as a cheaper option) to get the best of two worlds:

* Fast and accurate readings out of the box with photo editing wide-gamut monitor (i1DisplayPro)
* Printer profiling for every paper (i1Pro2)
* Fabric color measurement (i1Pro2)
* Fast and accurate readings for every display type, “well known” or unknown (measure SPD with i1Pro2, then feed i1Displaypro with that SPD data and use the colorimeter for actual color readings)
* ArgyllCMS support (i1DisplayPro & i1Pro2)

For very limited budgets and sRGB monitors without hardware calibration, [Color Munki Display](https://www.amazon.com/X-Rite-CMUNDIS-ColorMunki-Display/dp/B0055MBQOM/ref=as_li_ss_tl?ie=UTF8&linkCode=ll1&tag=photolife0c-20&linkId=18d0bd8803247380cf76472963752612) is a cheaper but very accurate option. Keep in mind that it’s a much slower device. A 10min (i1DisplayPro) patch measurement may go up to 40min with the Munki Display and this could be an acceptable time increase for the better price, but the measurement of a huge number of patches done in 30-40 minutes with an i1DisplayPro may go up to several hours with the Munki Display. It is up to you to decide what’s more important, your money or your time.

Sometimes you don’t want a specific CIE XYZ well-known coordinate as your white point calibration target, but some other device’s current white point. A few examples are tablets, paper under normalized light, another monitor…

While an i1DisplayPro is one of the most accurate devices (non-lab grade) to measure current GB-LED wide-gamut monitors just with the help of bundled reference SPD, tablet or paper “reference” white may have an unknown SPD. That device may even have SPD with narrow spikes, so i1Pro2 poor spectral resolution won’t get an accurate measurement either. That means your measurements of paper or tablet reference white come with errors, tiny or big. You may have a superbly accurate device to calibrate your wide-gamut monitor, like the i1DisplayPro, so when you calibrate that monitor to whatever CIE XYZ coordinates, you get a very close match to desired color coordinates with almost no error. But if you set inaccurate coordinates as target, because your devices cannot properly measure that paper or tablet white, you may get a visual mismatch between that reference white and your monitor’s white.

Poor screen uniformity (from your monitor or reference white device) may cause such visual mismatch too. You may get an exact match at the center of the screen (where you measure it for calibration) but if you have some blue, green or magenta tint in other zones of screen, you may see the two devices as a whole very different from each other. Calibration software computations may be inaccurate too, so even with proper equipment the resulting white may be wrong (an after-calibration measurement will diagnose that issue). The most common cause of that white mismatch is the reference white (tablet, paper). Measured coordinates are inaccurate because of your colorimeter or spectrophotometer limitations. Some calibration software acknowledges it, so after or before calibration is done, you can move to calibration monitor’s white point on the a\* and b\* axes (CIE L\*a\*b\*) with the help of that software until you get a visual match. NEC and Eizo software offer such a feature for their high-end wide-gamut monitors.

For current GB-LED backlight technology, a huge mismatch between standard observer and your own visual system (if you do not have a visual disability) is very unlikely to happen (no observer metametic failure), but if you want to match your wide-gamut monitor to a reference white from a device having an SPD with very narrow spikes, you can get a visual mismatch, even with lab grade equipment. Color coordinates of your “own observer” and “standard observer” for that spiky SPD reference may differ significantly. The actual source of observer metametic failure is that reference device whose white you want to “copy”, not your GB-LED monitor. As said before, for current wide-gamut monitors, observer metametic failure is not a real problem, it’s just an issue for other types of light source.

Pasted from <<https://photographylife.com/the-basics-of-monitor-calibration>>