Tutorial: display calibration

marianne.duyck@gmail.com

February 25, 2021

The general goal is to provide some guidelines on how to do present colored stimuli you intend to present and some tools to understand why it's done the way it is. The first step, presented here, is to calibrate your monitor, the second (future guide) will be to understand and juggle between the color spaces we most often use at the lab. Update and improve!

Last edited: February 2020 by marianne.duyck@gmail.com

1 Why

We use monitors to present color-controlled stimuli (color and luminance actually). However each monitor is different (like humans), because of their genetic material - brand and model, and their age - use. On top of that they are usually optimized for human perception. Human perception being non-linear for luminance, in most monitors the visual output of RGB (Red Green Blue) pixels luminance is a non linear function of its voltage input (or to be precise its non-continuous DAC for Digital Analog Converters). What we do by 'calibrating' the monitor is measure the output (luminance and chromaticity) that corresponds to a given input for each of the 3 RGB guns (usually expressed in a value ranging from 0 to 255 if we can display with a precision of 8 bits - corresponding to 2^8, from 0 to 65535 if we can display with a precision of 16 bits (2^{16}), or just from 0 to 1 to simplify some computations along the color spaces way).

The prerequisite of all calculations we need to do to go from one color space to another is that the relationship between input (RGB) and luminance output is linear. To do that we need to get the luminance for different input intensities for the 3 guns. We also need the chromaticity values in one color space (that we can then convert to any other color space).

2 How

2.1 Initial considerations and requirements

- To use that code to calibrate a monitor, you will need Matlab and Psychtoolbox 3 (Psychtoolbox contains function used to display the colors on the screen but also a toolbox that interacts with the PR - in case, a copy of that toolbox is also put in the folder with the code). To calibrate a tablet you can run the calibration on any computer with these 2 requirements and use TeamViewer to display the figure on the tablet from which you'll be taking the measurements. I seem to remember that for windows people in the past had to install drivers for the PR, but I don't think this is necessary anymore (if you realize you have to, that what *prusb.inf* if for). For Mac I didn't need to, and neither for my desktop on Windows 10.
- You want to make sure that the monitor has been on for at least 15 minutes, because it takes time to warm up (obviously you should make sure you do the same before running any participant)
- IMPORTANT. You want to know the monitor or tablet settings for which you are taking the measurements. Often at least contrast and brightness can be controlled from external buttons, sometimes even other settings that affect the colors displayed. You need to write down somewhere the values these variables had when you took your measurements. Of course you shouldn't change these values afterwards when you're using the calibration (or if you do, you have to recalibrate for the new values). Reciprocally, *never change* these parameters for displays other people are using without asking them first.

- You will also want to take the measurements in the same lightning conditions as the one of the experiment (as they will affect the distributions of colors that reach the eye). Meaning if the experiment is done in darkness take the measurements in darkness if its done with the lights on you'll turn the same lights on for the measurements.
- And finally it's best to take the measurements from the same distance to the screen the participant will be and mid-screen level (center of gaze at the level of screen center), because it's how we want the participant to be placed during an actual experiment (and because it's where the colored disks will appear).
- Note also that when the spectrum is measured for a given value of the R, G and B guns, it can be measured once (and that one spectrum is saved in the program) or several times what is saved is the average between the number of measurements. That is a parameter you can find and change in the PR menu. If you stick to one measurement, the whole calibration process - that currently measures 16 values for R, G, B and white (I agree that white is just a sum of the previous 3 but I preferred to include it rather than having to sum the 3 channels) - lasts about 15 minutes.

2.2 Systems/versions on which the code was tested and approved

- Mac OS Catalina+Matlab2016b+Psychtoolbox3 for both 8 bits monitor and tablet

2.3 Where to find the spectrophotometer?

We have 2 of them, their location changes, but I would look in that order on the main table of the lab, on the shelves above all the rigs or in Josh's office. I think we should find a more permanent place for them.

2.4 Code structure and history

Most of the subcode for the different devices (8bit monitor, 16bits with bits++ monitor and tablet) was written by Josh, and Marianne just added a wrapper to it, and an automatic fitting procedure of the gamma curves. Below is a gross schematic.

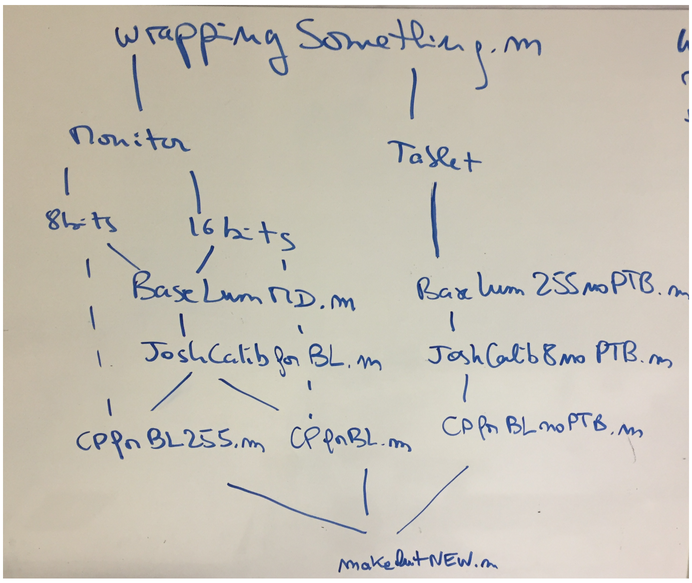


Figure 1: tree of the main code components

2.5 Future updates

If you use the code on systems and versions not listed above, please, update section 2.2. If you realize bug appears with Matlab or Psychtoolbox new versions you should update the code, eventually the guide and replace the old version location. But before you do that, please check your code works for all 3 devices (8bits monitors, tablets and 16bits monitor with the bits++ system). Don't forget that someone else took the time to save you the time to understand and implement the calibration steps so it's only fair you spend a bit of time yourself when arises the necessity to make updates.

2.6 Procedure

2.6.1 General

If you've read the section above, the SR should be at mid-screen level in a steady position at the same distance to the screen the participant will be during the actual experiment these measurements are taken for. It has to be centered around the center of the screen (where the colored stimuli stimuli will appear) and well focused (you will have the opportunity to adjust the focus in a first step of the code). If you don't see anything when looking in the finder, check you removed the cap and that the line on the knob at the top of the SR is horizontal.

Before any of the following steps, the SR has to be connected via usb to the computer getting the results, make sure you have enough cable length. It should also be connected to the current, and turned on.

Open the *wrappingSomething.m* file and change the variable PRport, line 13, to put the name of the port the SR is connected to. To find that port:

- Windows
  Go to Control Panel, then Devices and Printers, you should see the SR and the corresponding port number (eg 'COM3').
- Mac
  Open a terminal window, type **ls /dev/cu***, you should see a list. If you have several options, remove and plug back the SR to figure out which one it is. It should be similar to '/dev/cu.usbmodem14101'.
- Linux (hopefully)
  Open a terminal window, type **ls /dev/ttyACM***, you should see a list. If you have several options, remove and plug back the SR to figure out which one it is. It should be similar to '/dev/ttyACM0'.

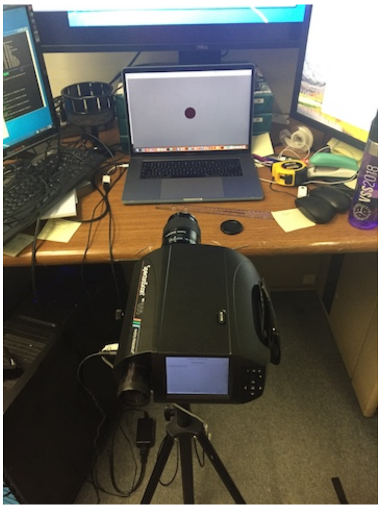
Then run *wrappingSomething.m* from the code folder it's in with all other necessary codes and files. You will be prompted in the command line to enter the name of the device you are calibrating. Once that's done, a new window will pop up asking you to select the device type (monitor or tablet). If you pick the option monitor you will then be asked if it's an 8 bits (eg your laptop) or 16 bits monitor that you want to calibrate (for tablets it will automatically calibrate 8 bits).

2.6.2 Monitor specific

A window displaying a white disc of the same size and position of the following stimuli used to calibrate opens so that you can finalized the position of the SR and its focus. Once you're done, you can press any key and the calibration process will really start (a black disc should appear and the SR screen should be in Remote mode). Then it's automatic. You can go get a coffee in the meantime.

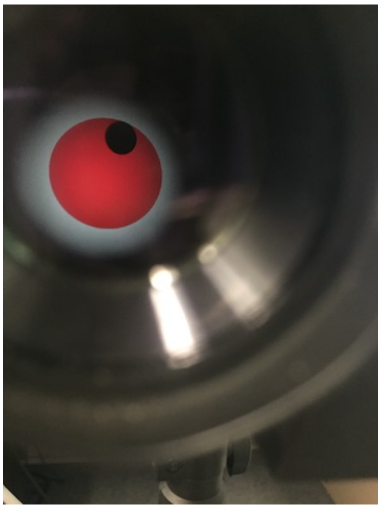
2.6.3 Tablet specific

Because the Android tablets used at the lab cannot have Matlab and psychtoolbox on them, the approach to calibrate a tablet is to use Teamviewer to remote control a computer that will run the code and display a matlab figure with the colors to calibrate. So before calibrating the tablet, you

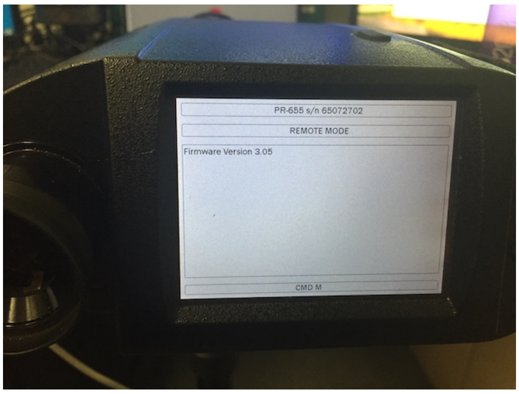


(a) setup to measure

(b) check the knob position



(c) the black dot should be entirely on the disk



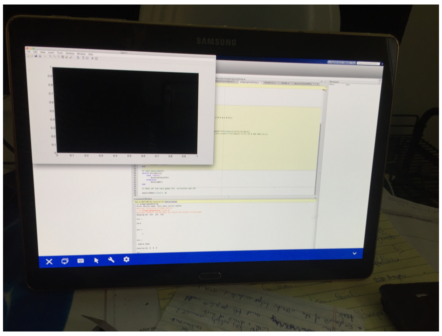
(d) remote mode screen when the SR is measuring

Figure 2: How to take the measurements.

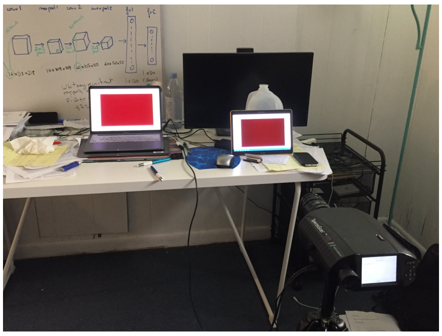
have to install Teamviewer on the tablet, and on the computer that will run the calibration code. Note that because of difference in resolution etc. between tablet and computer that are being used, it's not possible to make sure the elements appear at the center of the screen. Similarly to above, a small figure window will appear, put it at the center of the screen and adjust the focus of the PR so you can see single pixel level clearly. Then press any key, then a bigger window will appear filled with the first color. Then **be quick** (you have a few seconds before the PR takes measurement for that color) move the figure to the center of the screen or put it full screen. Then it's automatic. You can go get a coffee in the meantime.

2.6.4 Troubleshooting

If something bugs for some reason, always turn off the SR (sometimes necessary to also remove the battery) and unplug the usb before restarting. Also if you're having problems and you're not sure where it comes from, it's worth just checking the PR is working with the command `PR655init('usbportid')` followed by `PR655close()`. If you're calibrating a monitor that is the main monitor (and not a secondary one), it might be useful to open a non fullscreen psychtoolbox window so you can see what is happening on the Matlab commandwindow (i.e comment line 41



(a) move or expand the figure to where the PR is measuring



(b) measuring!

Figure 3: Tablet specific setup

and uncomment line 42 of *wrappingSomething.m*, and comment line 13 and uncomment line 14 of *BaseLumMD.m*).

2.6.5 Changes you might want to make

Default number of values measured for each gun is 16 equally spaced steps (line 27 of *wrappingSomething.m*). 16 because then all input values are integers and it doesn't take too long. If the current version of the code, each of the 3 R, G and B guns is measured, and also their sum (i.e white. That last one is not really necessary but I prefer to having it). If you want to remove it, replace the 4, line 17 of *BaseLumMD.m* by 3.

2.7 Outputs

At the end of the calibration process 3 figures will appear: one for each gun showing the measured values and the fits and inverse fits. These are showed for visual inspection so that you see if the fits are good (crucial to have a correct Lookup Table) and if there are not weird measured values. All files and figures should have been saves in a folder in *measurements* with a name that is the devide name you entered and the date.

If everything is ok, you can use the *deviceName_date_LUT.mat* file that contains the Lookup Table to use to linearize the output. The LUT as 3 columns (for respectively R, G and B) and as many lines as values that can be displayed for each gun.

Other files are saved for the sake of it (Figures for each gun fits in different formats, the values of the gamma fit and the values of the inverse gamma fit. I saved these last ones because can be used by the Screen('LoadNormalizedGammaTable' ...) function of the Psychtoolbox.

All these outputs are obtained from the measurements and the *LumValues* structure giving many informations: for each measured value of each gun you have the xyY Cie 1931 coordinates, the xyY Judd-corrected coordinates (they are corrected for the fact that Cie 1931 underestimate sensitivity at short wavelengths) and the full Spectral Power Distribution (SPD). LumValues.red, LumValues.green and LumValues.blue contain data obtained from the respective guns, LumValues.white is for their sum.

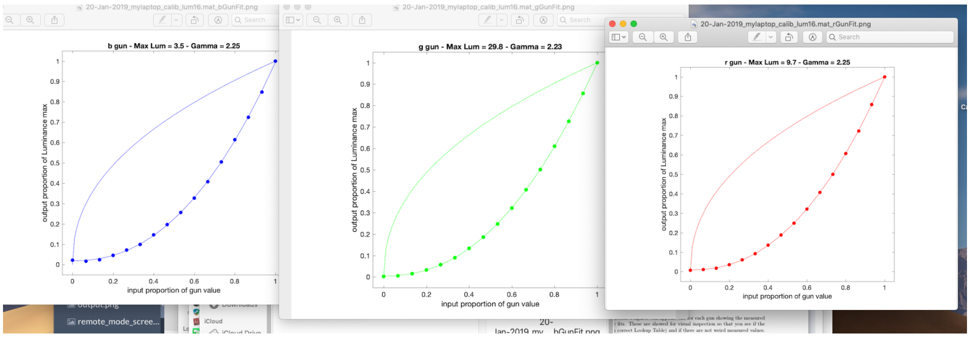


Figure 4: Measured values, gamma fits and inverse gamma fits for each gun

If you want to understand how we get the LUT from the measured values you can read the next section.
An examplar output can be found the measurements in the same root as the code folder.

LumValues | LumValues.red

LumValues.red

Fields	gunValue	xyYcie	xyYJudd	Spectrum
1	0	[0.3109,0.3270,1.4836]	[0.3211,0.3496,1.4904]	101x2 dou...
2	18.2143	[0.3750,0.3308,0.3302]	[0.3830,0.3455,0.3316]	101x2 dou...
3	36.4286	[0.4834,0.3258,0.4909]	[0.4896,0.3362,0.4923]	101x2 dou...
4	54.6429	[0.5623,0.3215,0.7682]	[0.5662,0.3290,0.7695]	101x2 dou...
5	72.8571	[0.6024,0.3200,1.2161]	[0.6042,0.3258,1.2175]	101x2 dou...
6	91.0714	[0.6287,0.3196,1.7585]	[0.6292,0.3245,1.7597]	101x2 dou...
7	109.2857	[0.6437,0.3197,2.4927]	[0.6433,0.3240,2.4938]	101x2 dou...
8	127.5000	[0.6556,0.3196,3.4276]	[0.6544,0.3234,3.4285]	101x2 dou...
9	145.7143	[0.6619,0.3198,4.4883]	[0.6604,0.3233,4.4889]	101x2 dou...
10	163.9286	[0.6672,0.3197,5.7392]	[0.6653,0.3230,5.7396]	101x2 dou...
11	182.1429	[0.6698,0.3194,7.1852]	[0.6678,0.3226,7.1854]	101x2 dou...
12	200.3571	[0.6721,0.3195,8.7658]	[0.6699,0.3226,8.7657]	101x2 dou...
13	218.5714	[0.6743,0.3196,10.6594]	[0.6719,0.3226,10.6588]	101x2 dou...
14	236.7857	[0.6760,0.3193,12.6528]	[0.6735,0.3223,12.6519]	101x2 dou...
15	255	[0.6766,0.3194,15.1815]	[0.6741,0.3224,15.1801]	101x2 dou...
16				

Figure 5: Content of the output structure for the red gun

3 More information about the outputs

3.1 Luminance and linearization

As mentioned before, all color spaces conversion matrices operates if there is a linear relationship between the input RGB values and the luminance, however the relation between RGB input and luminance output is a non linear, and best modeled by a function f:

$$f(x) = L = a + b * x^{\gamma}$$

where x is the normalized gun value ranging from 0 to 1 and L is the normalized luminance for that gun ranging between 0 and 1. As you can see for a given gun increment close to 0.25 the luminance will increase less than for the same gun increment close to 0.75. By linearizing we want to find the new gun value x' we should use in order to have a linear relation between the input and the output luminance (always being on the dashed line of Figure 3(b)). To do that we just have to compute the inverse function f^{-1} of the fit above:

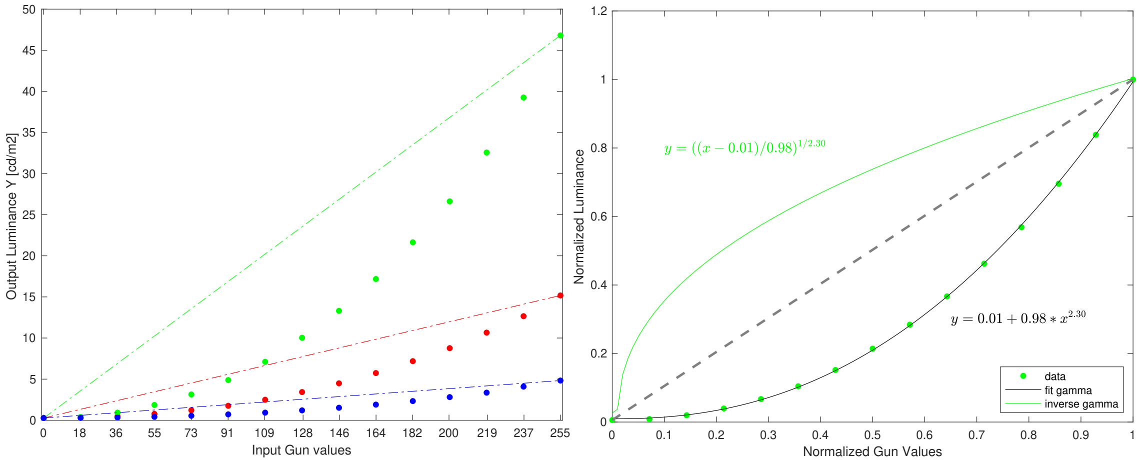
$$f^{-1}(x) = L = (\frac{x-a}{b})^{1/\gamma}$$

Let's take an example. We want to present a green at mid screen luminance level on a 8 bits screen whose measured values are presented Figure 2. If the screen was linear, we would use a gun value of 0.5 (127), but we can see that the luminance would be around 20% instead of the desired 50%. Fitted values give $a = 0.01$, $b = 0.98$ and $\gamma = 2.30$. Following the inverse function above, we get:

$$x' = (\frac{0.5 - 0.01}{0.98})^{1/2.30} = 0.74$$

We can indeed see by looking at the fit that a gun value of 0.74 (189) will yield a luminance of 0.5.

Of course the forget to linearize the 3 guns separately have to be linearized separately.



(a) Measured Values

(b) Fit and inverse fit for the green gun.

Figure 6: (a) Measured luminance values for the 3 guns. We can see how these measurements deviate from linearity for each guns (dashed lines). (b) Usually the luminance is best modeled by a gamma function as exemplified here for the green gun. In order to linearize the relationship between input and output we compute the inverse of the fitted function - represented in green here.

3.2 Chromaticity: from the Spectral Power Distribution to Cie xyY

The SR measure the SPD of each stimulus presented, that is the power at each wavelength (see Figure 3(a)). What we are interested in is getting the coordinates in a color space we know because then we can compute these coordinates for any colorspace via linear transforms. Let's take the example of the Cie 1931 colorspace or CieXYZ because it's an often used one and the real first one...

The only information we need on top of the SPD is the color matching functions (CMFs) for $\bar{x}$, $\bar{y}$ and $\bar{z}$ (these being convenient mathematical constructs derived from measurements obtained by asking participants to adjust the contribution of red, green and blue beams of light to equate a given monochromatic sample - the Wright-Guild measurements in the case of CieXYZ). Also called standard 2° observer CMFs. All that comes from the fact than any colored perceivable by the human eye can be reduced to the sum of red, green and blue monochromatic lights.

3.2.1 from SPD to the tristimulus values XYZ

The tristimulus values X, Y, Z from the SPD and CMFs are related in the following way:

$$X = \int_{\lambda} I(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = \int_{\lambda} I(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = \int_{\lambda} I(\lambda)\bar{z}(\lambda)d\lambda$$

In practice, we just need to multiply the SPD separately by each CMF and to sum the 3 outputs. Figure 5(b) illustrates multiplication of SPD and $\bar{x}$, then we just have to sum the resulting values for all the wavelength.

For this example (red maximum gun value), $X = 0.0471$, $Y = 0.0222$ and $Z = 0.0003$.

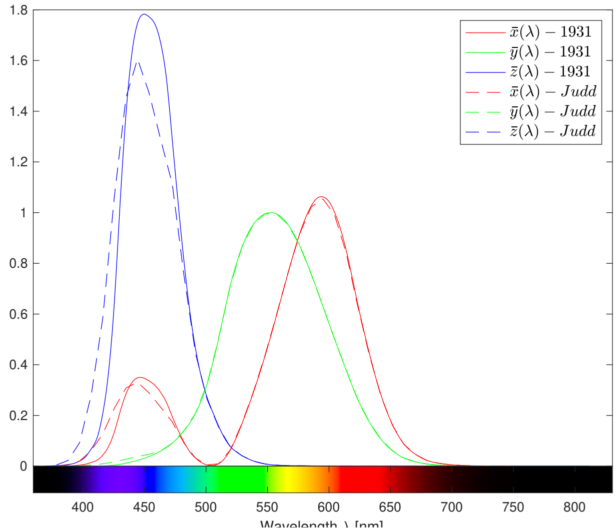
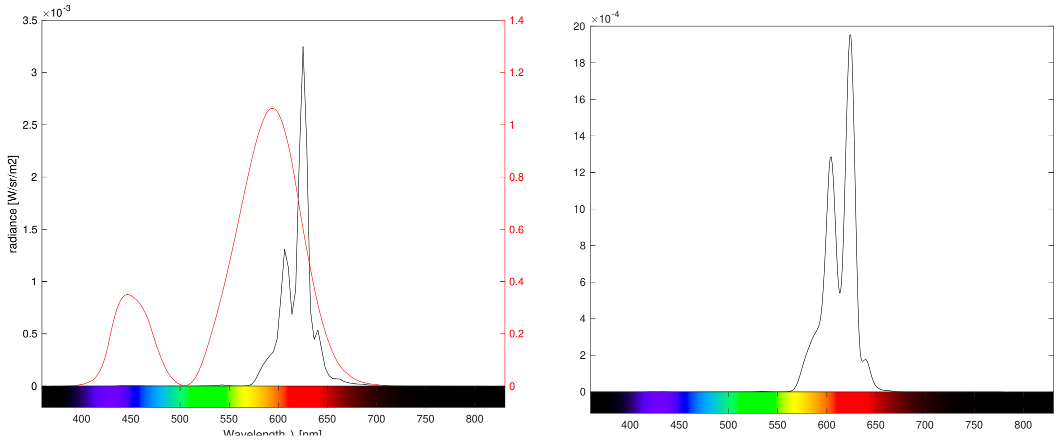


Figure 7: Cie1931 and Judd-corrected CMFs



(a) SPD and CMF for $\bar{x}$

(b) Their multiplication

Figure 8: Illustration from SPD to X

3.2.2 from XYZ to xyY

Then it's very straightforward to go from X, Y, Z to x, y, Y (btw Y always represents the luminance). Note that by convention chromaticity values are in lowercase. The reason why it's done is to easily display the colors on the chromaticity diagram.

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}$$
$$z = 1 - x - y$$
$$Y = Y$$

If we want to have the Y value in cd$/m^2$, i.e in luminance instead of radiance, we need to multiply Y by 683 (683 being the constant of maximal spectral efficacy in lumen/Watt). Thus $Y = 15.2$ cd/m2.

3.2.3 which CMFs to use?

As mentionned before CieXYZ underestimates human sensitivity of the short wavelengths (blue). Several corrections have been proposed since then: the Judd correction (1951), the Judd-Vos correction (1958), Stockman&Sharpe (2005) are the main ones I found. No idea yet which one would be better I didn't dig into that, but Josh's code will give you the Judd 1951 correction and better use this one than the non-corrected one.

Because Josh's code used Cie1931 and the Judd-corrected versions you can find both CMFs in the code as *ciexyz31_1.txt* and *ciexyzj.txt* (and the original files on the CVRL website).

Answers (to questions you might not have had before)

WHY ARE GREEN GUNS ALWAYS BRIGHTER?

As you can see on the example, the maximum luminance of green guns is a lot higher than the maximum luminance achieved by red or blue guns. And it's the case for CRT screens, LCD ones etc... But why would that be, since they receive the same amount of current?

Well, it's because *luminance* is not a physical unit but a photometric one. It depends both on the absolute energy of the spectrum and the sensitivity of the human eye along the spectrum. The unit representing absolute power (of the wavelength(s)) is called *radiance*. Luminance is obtained by weighting that radiance by a factor that represents how sensitive the human eye is at the different wavelengths.

The function that describes human sensitivity to light is called luminosity function or luminous efficiency function $V(\lambda)$ and is simply obtained by asking people to compare perceived brightness of pairs of different colored stimuli. There are different luminosity functions depending on the light levels but at photopic light levels you can see that luminous efficiency peaks around 555 nm corresponding to green. It means that the human eye will get more light from green than blue or red if all three have the same radiance.

In conclusion, luminance of the green gun is higher because our eyes are more sensitive to light in that part of the spectrum than towards red or blue parts of the spectrum.

WHY DON'T COMPUTERS HAVE GREEN TEXT ON BLACK BACKGROUND ANYMORE?[a]

Because a large fraction of adults - me included - have astigmatism and astigmatism (due to asymmetric cornea shape) causes blurry vision. Because at low luminance levels (black screen background) the pupil is more dilated, the light enters the eye on a larger surface and vision is even more affected by the shape of the cornea, causing more blur.

On brighter background (think about that white blank word document - your next paper - you should be working on), the pupil is smaller and the incoming light less affected by the cornea distortions, thus vision is less blurry!

[a]except to read the Matrix

References

[1] C. Abraham. A Beginner's Guide to (CIE) Colorimetry. https://medium.com/hipster-color-science/a-beginners-guide-to-colorimetry-401f1830b65a/, 2016. [Online; accessed 12-July-2018].

[2] Colour and V. R. Laboratory. CMFs. http://www.cvrl.org/, 2018. [Online; accessed 12-July-2018].

[3] D. B. Judd. Report of us secretariat committee on colorimetry and artificial daylight. *Proceedings of the 12th Session of the CIE, 1951*, 1:11, 1951.

[4] K. D. . M. Levoy. Color. http://graphics.stanford.edu/courses/cs178-10/applets//, 2018. [Online; accessed 12-July-2018].

[5] A. Ryer. Light Measurement Handbook. http://apps.usd.edu/coglab/schieber/pdf/handbook.pdf/, 1998. [Online; accessed 15-July-2018].

[6] P. Service. The Wright – Guild Experiments and the Development of the CIE 1931 RGB and XYZ Color Spaces. http://philservice.typepad.com/Wright-Guild_and_CIE_RGB_and_XYZ.pages.pdf/, 2016. [Online; accessed 12-July-2018].

[7] L. T. Sharpe, A. Stockman, W. Jagla, and H. Jägle. A luminous efficiency function, v*(λ), for daylight adaptation. *Journal of Vision*, 5(11):3–3, 2005.

[8] J. J. Vos. Colorimetric and photometric properties of a 2 fundamental observer. *Color Research & Application*, 3(3):125–128, 1978.

[9] S. Westland, C. Ripamonti, and V. Cheung. *Computational colour science using MATLAB*. John Wiley & Sons, 2012.