Introduction

The color (or channel) mixer in the color calibration module prompted several YouTube tutorials and despite the abundance, a lot of people still face issues.

The mixer encompasses three tabs, next to the CAT and others. One for red, green and blue and in each tab three scales with sliders which can move to the left or right. So, in total nine sliders to adjust and uncertainty what to do.

People remain confused as the tutorials take very different approaches. Intuitive approaches exist next to matrix algebra, illustrating an infinite number of solutions to choose from. Others upload short tutorials where one simply moves sliders and leave it at that, suggesting that the background of the module is not relevant to understand one's actions. In pixls.us several threads can be found people asking for guidance and explanations

To summarize: confusion remains and here an attempt to remedy that.

Goal of the color mixer module

As said, the mixer is a part of the color calibration module. In the latter the Chromatic Adaptation Transformation (CAT) is important as it deals with white balancing: no need to go into details on that.

The mixer can adjust red, green and blue colours (= channels). It does so by modifying the strength of the R, G and B input (in an independent manner) and it alters the colours in the photo. If you want to turn the sky into a deeper blue, you need the strengthen the blue channel. In a winter landscape there will be a lack of green tones and you choose to strengthen the green channel. A red tinge can disappear by weakening the red channel etc., etc. The mixer can be viewed upon as a second method for chromatic adaptation, changing colours to your liking. The DarkTable manual lists '*White balance is only a partial chromatic adaptation*' and for a more extensive adaptation you can use the mixer.

So, when you modify the blue channel and leave the other two unchanged, you will create a new colour. RGB (103,58,183 = kind of purple) can be modified into RGB(103,58,75 =reddish brown). So, again, you can correct colours, you don't like or induce strange effects. It is all up to you.

You must realize that the mixer in its default stage is a generic modifier. The entire photo will be adapted. All colours change, even those you like to leave unchanged. As usual, Darktable offers the possibility to insert masks to restrict the actions of the mixer.

You can argue that other modules perform similar tasks: true. As often in DarkTable (DT) different routes can be taken and each deliver similar outcomes (which adds to the confusion). It is up to you what module you (learn to) favour.

The Bayer sensor

To start with the characteristics of the sensor sounds peculiar, but to understand the 'mechanics' of the mixer it is useful.

The individual photosensors capture red, green or blue light, using filters. The sensors, let's say, get charged and the more light, the higher the charge. The software knows which sensors are dedicated to red, green and blue and measures their charges. The photo is subsequently calculated using charge / location data and is shown on screen.

To modify colours you can follow two routes: 1] 'increase' the charge and 2] 'swap' red, green and blue sensors[1]. Increasing the charge via the software means you simply multiply (or divide) a reading by a factor, you choose. Suppose you have a pixel with RGB(120,10,5), the red sensor is charged and the others much less. Increasing the red charge delivers RGB(200,10,5): you went from dark to a lighter red. You can also target (RGB 100,50,10) via appropriate choices. It will clear that many possibilities do exist and let us name this way of working 'in- / decreasing'.

The other route is to 'swap' individual sensors. Of course, physical replacements are impossible, but the software can be instructed to label a green sensor as blue on or a blue one as red. Conceive the blue channel: the red and green sensors can be labelled as blue. The processing of the photo will lead to a stronger blue colour. In this way, the e.g. red channel can be augmented by swapping the blue / green sensors to red. The blue channel by swapping red / green to blue and the green one by swapping red / blue. The above is done when strengthening colours, weakening means swapping to the opposite colours for red, green or blue.

It would be awkward when this swapping would be an all / nothing phenomenon: you can label a smaller / larger proportion of the sensors to be swapped and thus control the increase in colours. Let us name this way of working 'swapping'.

To summarize: the build of the Bayer sensor and the software allows you to change three colour channels independently and use three approaches per channel. Now you know why you see three sliders per channel: one for 'in- / decreasing' and two for 'swapping'.

Maths

In this section an intuitive approach to the underlying maths equations. The tutorials by Nicolas Winspeare deal with the maths in a more fundamental way: these are excellent but can be overwhelming.

It boils down to three equations, one for each channel. Each equation must contain the three terms to modify colours: the 'in- / decreasing' term and two for 'swapping'. Each individual term consists of two building blocks: the input to be changed and the parameter indicating the strength of the change. The outcome of the equations is the new colour.

So we get the following

R_{out} =	$X_{rr} * R_{in}$ +	$X_{rg} * G_{in}$ +	$X_{rb} * B_{in}$
G_{out} =	$X_{gr} * R_{in}$ +	$X_{gg} * G_{in}$ +	$X_{gb} * B_{in}$
B_{out} =	$X_{br} * R_{in}$ +	$X_{bg} * G_{in}$ +	$X_{bb} * B_{in}$

Table 1 linear regression equations for changes in R_{out}, G_{out} and B_{out}. The greyed cells denote the in- / decrease terms, the other the swapping terms

The grey cells denote in- / decreasing' terms, the blank the swapping ones. The X's define the strength of the changes and R_{in}, G_{in} and B_{in} are simply the original (or input) RGB-values. These equations are called linear regression equations by statisticians, the X's are named regression coefficients by them. The equations can be found in the DT manual, too.

Linear regression equations shows interesting characteristics: they are independent (=uncorrelated) and the terms too. That means that you can modify the blue equation, knowing that the red /

1https://discuss.pixls.us/t/how-to-understand-channel-mixing-in-color-calibration/33294 green channel remains unaffected. The next characteristic is that the terms in each equation are also independent: you can use them separately, without affecting the others. Using the mixer thus requires a lot of decisions: 1] which channel to work on and 2] which term to adopt. For these terms: their effects are added, so you may use either one of them or two or three. For the outcome it is irrelevant.

The starting situation is one of '1' for X_{rr}, X_{gg} and X_{bb}, the other coefficients equal '0' and so R_{out} = $R_{in,}$ etc. Nothing has changed and you recognize this in the default slider values.

R_{out} =	1 * R_{in} +	0 * G_{in} +	0 * B_{in}
G_{out} =	0 * R_{in} +	1 * G_{in} +	0 * B_{in}
B_{out} =	0 * R_{in} +	0 * G_{in} +	1 * B_{in}

Table 2 starting situation, indicating non-effective swap terms and R_{out} equals R_{in}, etc.

Moving the sliders to the right increases the coefficient values, to the left they decrease. The range of the coefficient values is limited to -2 and +2.

The swapping coefficients denote the fraction of sensors swapped. 0 for none, 0.5 for 50%.

Where to find the input values? That is easy: use the color picker! You can choose a single pixel or an area and the picker lists the RGB-values for you. It also shows the changes when you start modifying the coefficient values: a very useful option.

An example

I took a winter photo of a swamp nearby. The overall mean RGB of the photo turned to be RGB(123, 122,103), which is rather dull. By shifting the green X_{gg} coefficient to 1.5 I got RGB(123,150, 101), a substantially greener photo. By shifting X_{gr} coefficient to the left I could counteract the action to obtain the unchanged RGB(123, 122,103).

The clever observer immediately noticed that 1.5 * 122 ≠ 150, it should be 183 and the blue value did change a little. The latter was explained by Nicolas Winspeare in his 2nd mixed tutorial[2]: the RGB colour space of the picker deviates somewhat from the working RGB space.

The first phenomenon can only be explained by assuming a non-linear behaviour, which is not surprising as the RGB range is confined to 0 – 255 values: outcomes must fit within this range. Probably the multiplication is adapted when the outcome of the maths lead to high colour values. If one multiplies a high (green) 220 colour value with a high coefficient (e.g. 1.7), the outcome will >>255. This is invalid and probably prevented through ?. So, if outcomes are forced to range between 0 - 255, the classical concept of linear regression equations seems to be invalid. If this is the case, prediction of outcomes by using the equations in e.g. spreadsheets is a hazardous route as the exact multiplier value is a guess. The concept of linear equations, as present in the manual, is then just a nice way to explain the mechanics of the mixer.

Workflow[3]

The mixer needs pivotal choices, but first a warning. The above explanations focused on independent effects in the mixer: red channel changes don't affect the blue channel, a strong prerequisite to obtain valid results. However, to fulfil that requirement the mixer needs the CAT adaptation space to be set to *none (bypass)*. When you stick with the default CAT16 space and shifts the sliders, you will see the RGB values change all over the place. So the advice is to use a 2nd instance and that guar-

2 https://www.youtube.com/watch?v=QX_HItCqDtE
3 https://discuss.pixls.us/t/i-have-no-clue-how-channel-mixer-works/34854/2 for a Todd Prior pdf

antees an automatic change to the *none (bypass)* situation. By doing so, you mix in the RGB working space of the pipeline.

A next thing to realise is that the changes you make, will affect all pixels. When you use the green channel all green pixels will be affected: changes are generic. However, the absolute change will dependent on the number of green pixels in an area. Suppose there is a red area in your photo with a low amount of green in it, but in another area a lot. Multiplying green by a (strong) factor will affect that red area hardly. The green starting amount was low and stays so. An area with a higher amount of green thus will be more affected: the changes you see, depend on the G_{in} values, next to the multiplier magnitude. So from photo to photo the mixer will give you different results.

You need first to decide which area (=colour) is important to you. That is your main goal: a sky needing a colour improvement or grass to look better and let's call that the *primary area*. It is a good idea to use the color picker to measure the RGB values. The next thing is to decide what change you deem necessary: more blue in the sky or more green in the grass? That defines your strategy, which is often a creative decision.

In general,

- In the *R* channel, move sliders to the right and all (=R, G and B) areas show more "red", to the left for more "cyan".
- In the *G* channel, move sliders to the right and all (=R, G and B) areas show more "green", to the left for more "magenta".
- In the *B* channel, move sliders to the right and all (=R, G and B) areas show more "blue", to the left for more "yellow".

Of course you can use two or three channels for to mix colours, like yellow via red / green. The possibilities are endless.

Now how to deal with the three sliders per channel? I feel that starting with the ín- / decreasing sliders is the most efficient way forward. These sliders influence the pixels in a direct and straightforward manner, no fuss and in the color picker readings you can see what has changed. If you do so and are happy with the outcome, that is it: you increased e.g. the amount of blue in the photo to your liking!

The problems start when you are not happy. It is very well possible that certain areas are now too blue to your liking and can you remedy that? Well, to a degree that is possible, but the explanation how to is rather complex. You need not only to focus on the primary area, but also on others. So, use the color picker to measure the RGB-values several other areas. How many: up to you, but select a series of secondary areas (or colours) and make sure that they differ in location from the primary area.

In the table below I illustrate that: one primary area and four secondary ones with their RGB values. The starting value for the primary area was RGB(150,180,222). I decided to move the red channel input R slider to 1.1 and got (RGB (159,181,222).

	areas				
	primary		secondary		
start	150	169	157	244	26
	180	166	157	231	30
	222	93	94	172	25

input R=1.1	159	176	164	252	28
	180	164	156	225	30
	222	92	93	168	25
input G=0.07	159	174	162	249	28
	179	165	156	227	30
	221	92	94	169	25
input B=0.05	159	170	158	246	27
	179	166	157	229	30
	221	93	94	171	25

Table 3: RGB value outcomes for 5 areas after three kinds of adaptations in the red channel

Part of the experiment was to asses whether I could use the input G and B to obtain the same results and you can see that it is the case by choosing resp. 0,07 and 0,05[4]. In other words, for the primary area is doesn't matter in the end which slider you use. (The data indicate that 7% and 5% of the green / blue pixels turned red). As now expected, using input red, the secondary areas change significantly and, again, if you are happy with that is fine. However, using input G and input B elicit very different secondary area outcomes and perhaps better ones. See that the secondary areas after input G changed less and using input B hardly and it is all entirely logical!

Remember that input G and input B swap pixels by a percentage and the higher the original amount of green or blue in your secondary areas, the stronger the changes: it is a simple calculation. In secondary area 1 and 3 (=column 3 and 5 in table 3) the amount of blue differs a lot: low and high. Hence, the 5% change towards red is minor in area 1 and stronger in 3. See the high amounts of green in secondary area 1, 2 and 3: the 7% swap towards red must end in significant changes in these areas.

The take home message is that the swapping sliders offers the possibility to protect secondary areas from changing too much. The workflow one can follow is thus:

1. Select a primary area next to a series of secondary ones and measure the RGB values in all with the color picker.
2. Decide what channel you will use.
3. Use the input red / green / blue sliders first when resp. working in the red / green / blue channel, asses the primary area outcome and measure its RGB value: note it. Happy with the outcome: stop here.
4. Reset the input red / green / blue sliders to '1'.
5. See which colour of the two swapping sliders in the secondary areas differs most from the primary area and use that input to obtain the same results under 2). Rephrased, choose the appropriate percentage of change to get these same results as under 3.

Will it always work: no! The degree of protection depends on the difference between the primary and secondary areas. For green the differences were smaller and so less protection, the blue channel offered better protection for some areas, but not for area 4. When no or small differences are

4 If you would choose input B or G to be -0,07 resp. -0,05 and keep input R at 1,1, you will annihilate the input R 1,1 effects for the primary area

present: no protection via swapping. When the percentage swapping is high, the effects on the secondary areas are higher. So in the end, some luck is needed. Protection of secondary areas needs low percentages of swapping and high differences between primary and secondary area colours. If these prerequisites aren't present: consider masking (or other modules)!