

## White paper about color correction

# More drama

---

**Many application fields like digital printing industry or the human medicine require a natural display of colors. To illustrate the importance of color fidelity, you can think about a doctor using an endoscope during an operation who wants to remove the right tissue. The purpose of this white paper is to show how you can optimize the color fidelity of a camera, so that it looks as natural as possible on different displays and for human vision.**

This implies some linear and nonlinear operations (e.g. display color space or Gamma viewing LUT) which are normally not necessary or recommended for machine vision algorithms. A standard monitor offers, for example, several display modes like sRGB, "Adobe RGB", Wide Gamut etc., which reproduce the very same color of a camera color differently.

It should also be noted that with modern cameras users can choose for either

- camera based settings and adjustments or
- host based settings and adjustments or
- a combination of both.

Camera (or hardware) based settings are advantageous to achieve highest calculating precision, independent of the transmission bit depth, lowest latency, because all calculations are performed in FPGA on the fly and low CPU load, because the host is not invoked with these tasks. Using, for example, MATRIX VISION's GigE Vision camera mvBlueCOUGAR-X, these camera based settings are

- gamma correction
- negative gain / gain
- look-up table (LUT)
- white balance
- offset
- saturation and color correction

Host (or software/driver) based settings save transmission bandwidth at the expense of accuracy or latency and CPU load. Especially performing gain, offset, and white balance in the camera while outputting RAW data to the host can be recommended.

### Color behaviors

To show the different color behaviors, you have to use a color chart like the GretagMacbeth™ color chart as a starting point:

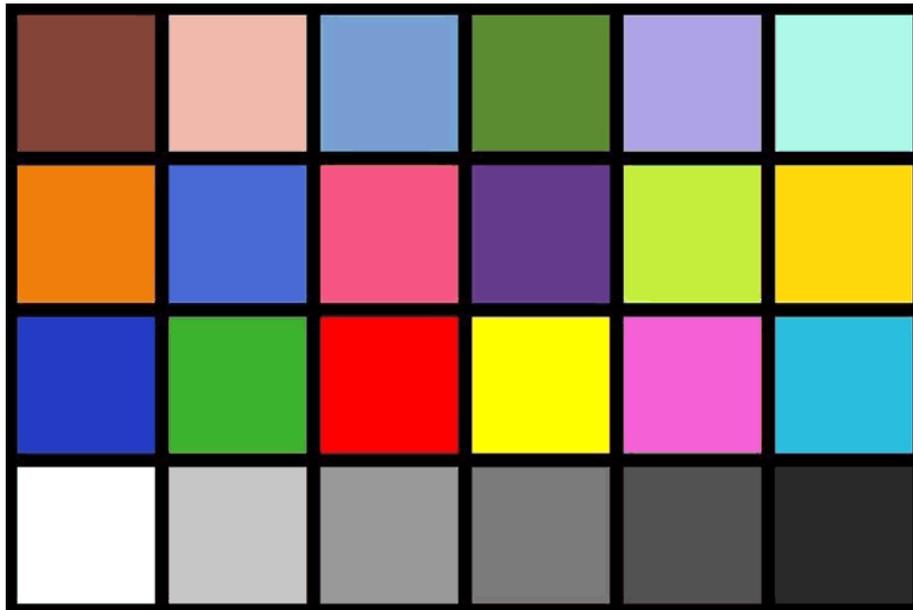


Figure 1 Color chart as a starting point

The color fields are important to compare and adjust the colors while the gray fields are used to adjust the white balance, offset and gain and gamma.

If we take a single shot of the camera without any color optimizations, an image can be like this:

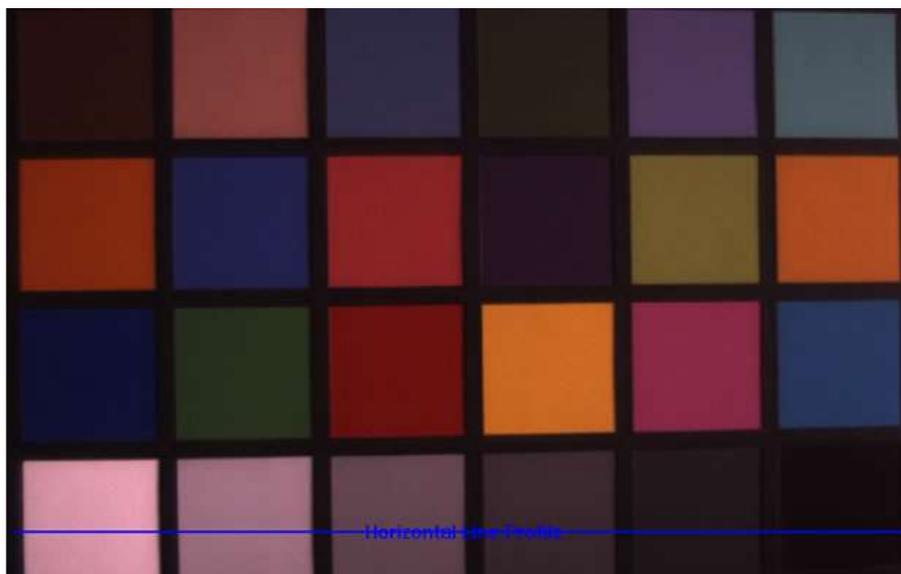


Figure 2 Single snap without any color optimization



Figure 3 Corresponding histogram of the horizontal white to black profile

As you can see,

- saturation is missing,
- white is more light gray,
- black is more dark gray,
- etc.

You have to keep in mind that there are two types of images: the one generated in the camera and the other one displayed on the computer monitor. Up-to-date monitors offer different display modes with different color spaces (e.g. *sRGB*). According to the chosen color space, the display of the colors is different.

For this reason, you have to improve the image data and the display as show in Figure 4:

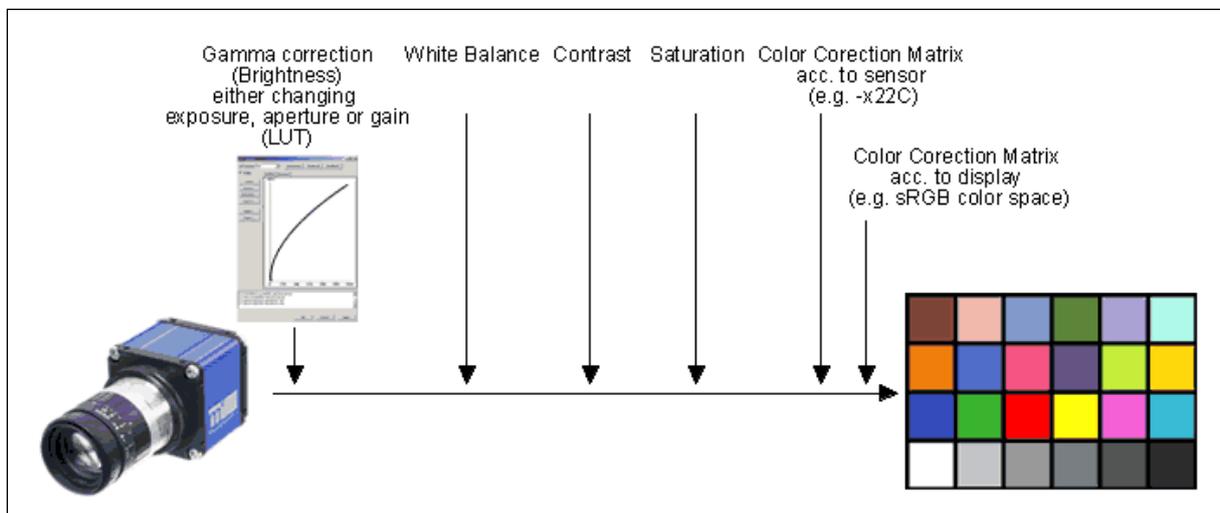


Figure 4 The way to a perfect colored image

To get a perfect colored image you have to do the following steps:

1. Do a Gamma correction (Brightness),
2. make a White balance, and
3. Improve the Contrast.

4. Improve **Saturation**, and
5. Use "**Color Correction Matrices**" for both,
  - a. the sensor and / or
  - b. the monitor.

The following sections will describe the single steps in detail.

### Step 1: Gamma correction

First of all, a **Gamma correction** can be performed to change the image in a way how humans perceive light and color. In the human visual system, the brightness is perceived as the logarithm of the actual light intensity and the visual system interprets the image accordingly. For example, a bright light source which is approximately 6 times brighter than another; the eye will perceive the brighter light approximately 2 times brighter than the darker one.<sup>1</sup> In other words, it follows a power function  $x^{\text{Gamma}}$  (Gamma between 0.3 and 0.5). In contrast, a display device like a LCD monitor works linearly (Gamma = 1). So to perceive this linearity, images have to be encoded with the reciprocal (approx. 2.2) of the Gamma value of the human eye. In former times, CRT monitors did this by default:

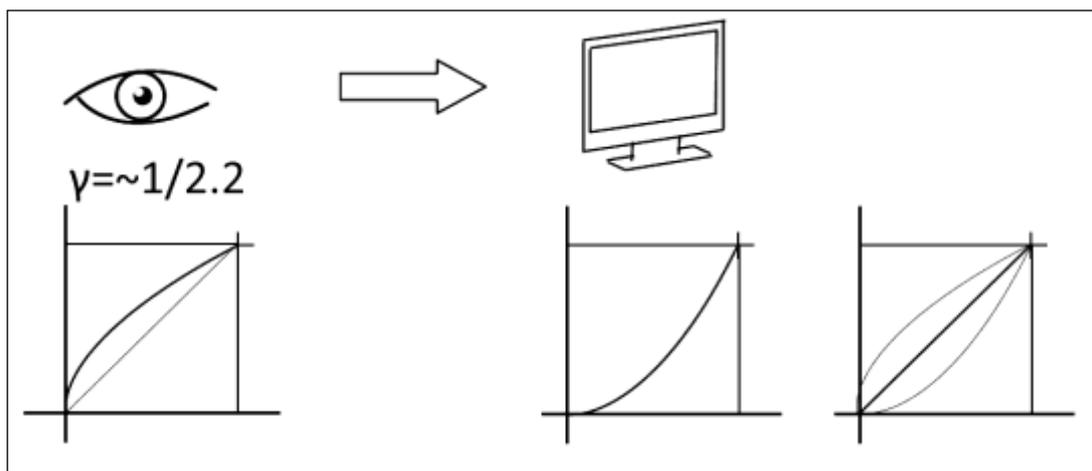


Figure 5 Gamma encoding of a monitor

Today this functionality is built into the monitor itself or in the graphics card and is automatically selected by the color profile (e.g. sRGB). Certain programs do not cross check what other programs do in the pipeline. For this reason it is not trivial to find out the overall GAMMA of the chain (camera->PC->monitor).

A CCD or CMOS camera works linearly as well (Gamma = 1), except when using the HDR mode. As mentioned before, a monitor encodes the image with a Gamma value of 2.2. So to perceive linearity again, you have to do a Gamma correction on the camera with the reciprocal of the Gamma value of the monitor (1/2.2).

<sup>1</sup> [http://charity.cs.uwlax.edu/artofimageprocessing/powerpoint/Chapter\\_02\\_HumanVisualSystem.pptx](http://charity.cs.uwlax.edu/artofimageprocessing/powerpoint/Chapter_02_HumanVisualSystem.pptx)

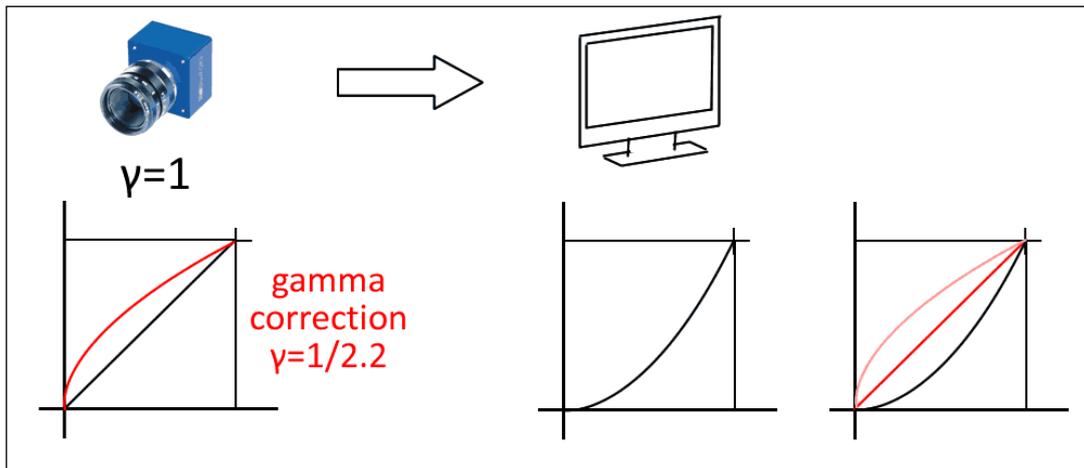


Figure 6 Gamma correction in the camera to get a linear (ideal) Gamma value (red)

A linear camera on a linear display would also show correct images.

Other ways to change the brightness are to adapt either

1. the exposure time,
2. the aperture or
3. the gain.

After gamma correction, the image will look like this:



Figure 7 After gamma correction



Figure 8 Corresponding histogram after gamma correction

## Step 2: White Balance

As you can see in the histogram, the colors red and blue are above green. Using green as a reference, we can optimize the **white balance**. This means that we try to adapt the neutral colors like gray and white to appear neutral in the reproduction. After optimizing white balance, the image will look like this:



Figure 9 After white balance



Figure 10 Corresponding histogram after white balance

### Step 3: Contrast

Still, black is more a darker gray. To optimize the contrast you have to adapt the black level. This means that the darkest part of the image results in a pure black part on the screen. After correcting this, the image will look like this now:

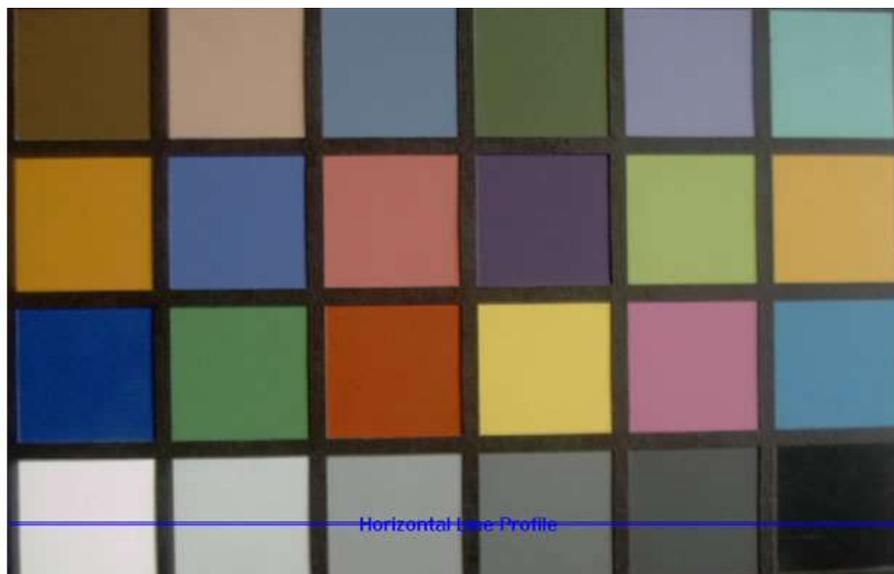


Figure 11 After adapting contrast

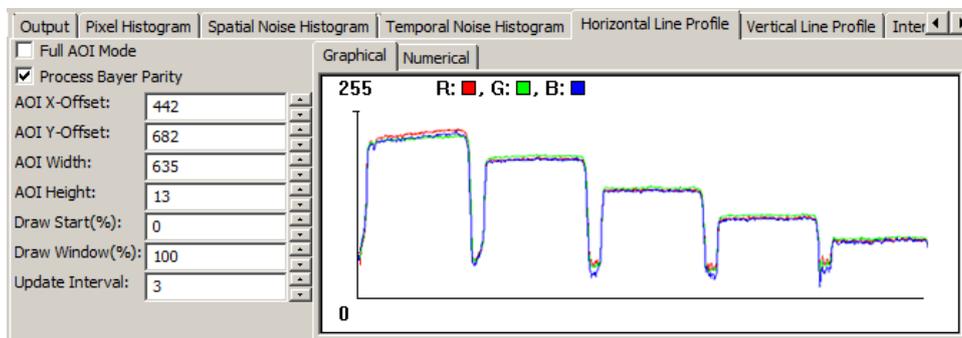


Figure 12 Corresponding histogram after adapting contrast

#### Step 4: Saturation

Still saturation is missing. In common setting tools you have the possibility to increase or decrease the saturation. By increasing the saturation you can remove grayness from the colors.

After the saturation, the image will look like this:

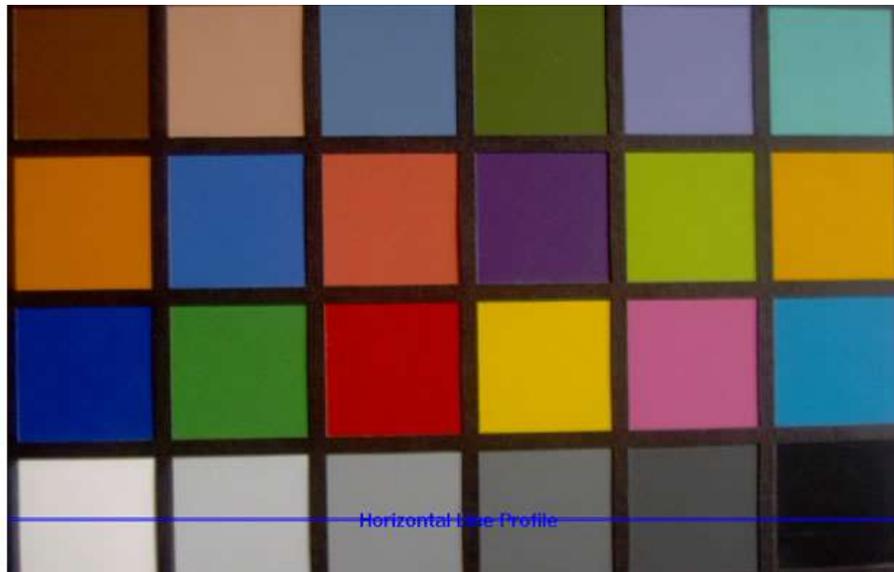


Figure 13 After adapting saturation

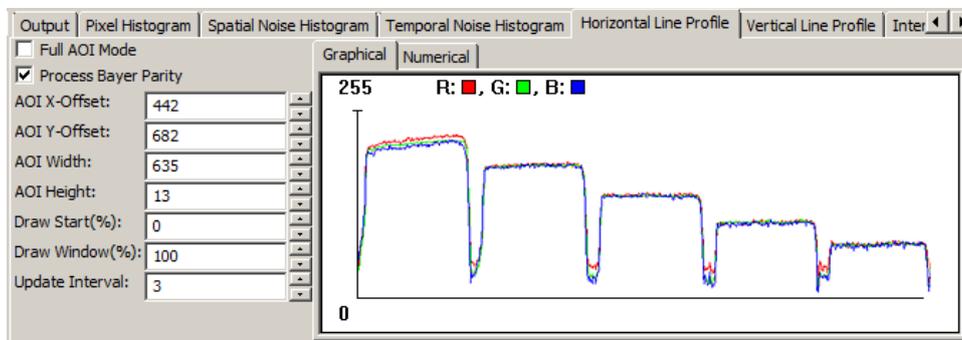


Figure 14 Corresponding histogram after adapting saturation

The result is very close to the original color chart. However, there are still some optimizations possible with regards to the color space of the display and the spectral sensitivity of each sensor compared to the color vision of the human eye - time for color correction matrices.

#### Step 5: Working with Color Correction Matrices (CCM)

Modern cameras offer a property tool where you can normally adapt three separate 3 x 3 color correction matrices:

- A matrix with sensor specific correction coefficients (9 coefficients and 3 offset values for one RGB(in) → RGB(out) transformation).
- A parameter for the color saturation, effective for all image formats (RGB and YUV; see step 4).
- The choice of a display color space.

A result matrix is calculated from these three matrices. This matrix is used in the linear part of the signal path. MATRIX VISION for example measures all sensors of all camera families step by step and determine the spectral characteristic in our measuring lab which could be like the following:

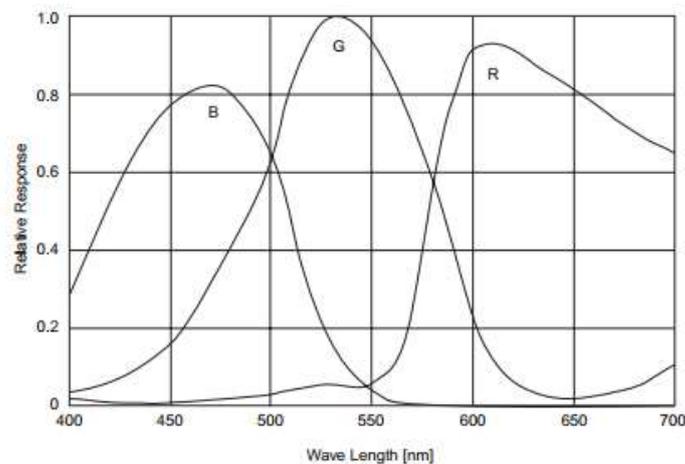


Figure 15 Spectral characteristic of Sony ICX414AQ

Source: <http://www.sony.net/Products/SC-HP/datasheet/01/data/E02403C3Z.pdf>

Compared to the spectral characteristic of the human eye, you will see differences in the blue and red area:

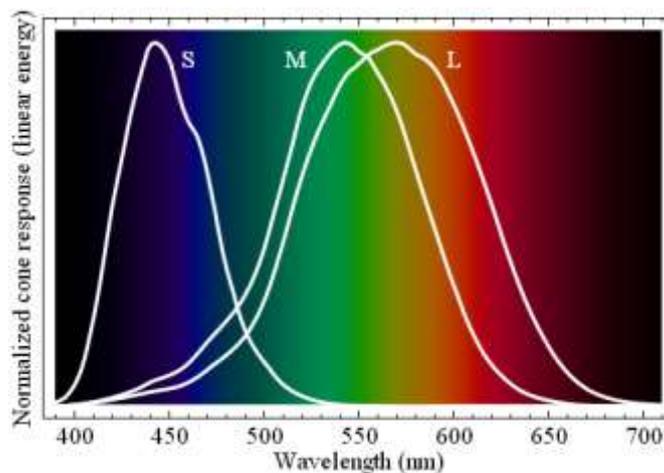


Figure 16 Spectral characteristic of the human eye.

Source: <http://en.wikipedia.org/wiki/File:Cone-fundamentals-with-srgb-spectrum.svg>

A color correction matrix is used to adapt and shift the peaks of the colors. This matrix is called white-point preserving least square (WPPLS) and looks like this:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.65724 & 0.26420 & 0.07857 \\ 0.23601 & 0.92376 & 0.15977 \\ 0.07757 & 0.31273 & 1.23516 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Figure 17 WPPLS color correction of the sensor

In this sample, each color is intensified with all color components (red, green, blue), however, each component is especially intensified by its own color.

The last place where you can optimize the color fidelity is the display. Monitors display color very differently depending on their technology. LCD monitors, for example, offer different color spaces like sRGB, Adobe RGB, or Wide Gamut RGB. The following figure illustrates the different color spaces. The gray area represents the gamut of the colors visible to the human eye:

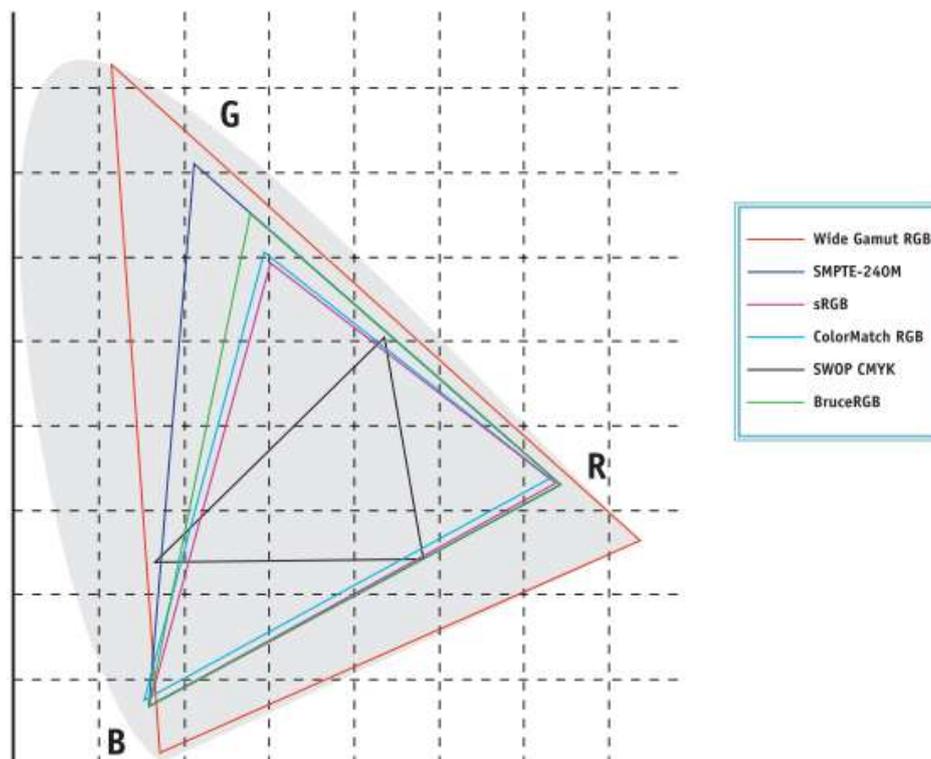


Figure 18 Color spaces  
Source: Adobe Magazine Autumn 1998 page 51

The display matrix transforms the colors from the camera matrix into the desired display. After this color matrices the will look like this:



Figure 19 Final image with corrected colors

### Color correction with modern cameras

State of the art should be that industrial cameras offer host based color correction. However, in modern times cameras like MATRIX VISION's GigE Vision camera mvBlueCOUGAR-X and the USB3 Vision camera feature a FPGA. This FPGA can be used to execute the color correction in the camera directly. This leads to

- highest calculating precision (independent of the transmission bit depth),
- lowest latency (all calculations are performed in FPGA on the fly), and
- low CPU load.

To set the color correction properties, you can use MATRIX VISION's property tool wxPropView. This tool involves a lot of wizards which helps you to achieve your color goals faster – and you will. As you can see in Figure 20, mvBlueCOUGAR-X cameras can reach a color improvement by the factor of 2.

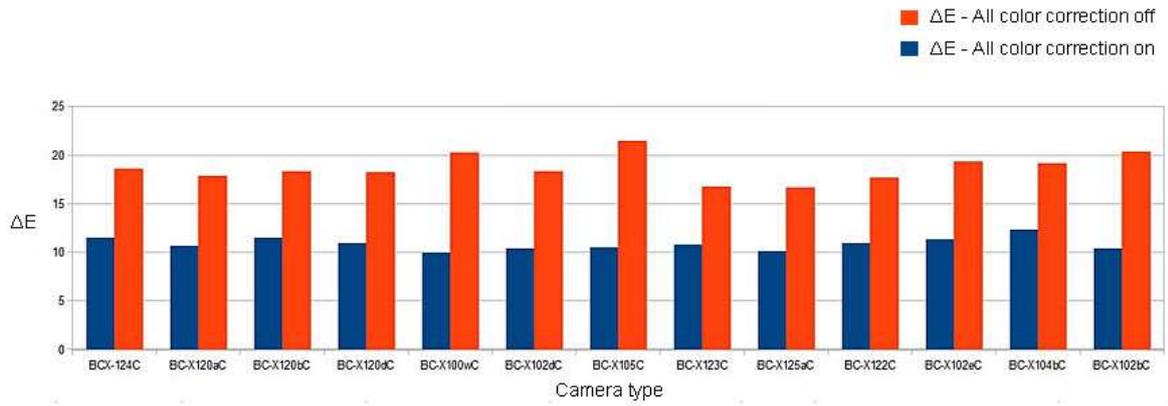


Figure 20 Quantitative comparison of mvBlueCOUGAR-X models

Author: Horst A. Mattfeldt, Senior Consultant MATRIX VISION

<https://www.matrix-vision.com>