

Spectral Imaging on Gamut Limited Devices

Bachelor Thesis of

Pascal Bies

At the Department of Informatics
Institute for Computer Graphics (IVD)

Reviewer:	Prof. Dr.-Ing. Carsten Dachsbacher
Second reviewer:	Prof. Dr. Hartmut Prautzsch
Advisor:	Dr. Johannes Hanika

Duration: 19. May 2014 – 19. September 2014

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 19. September 2014

.....
(Pascal Bies)

Abstract

Converting a spectral image to color triplet images (e.g. RGB) is classically done by applying the CIE XYZ color matching functions to each spectrum of a pixel and converting the resulting CIE XYZ coordinates with well known transformations into desired color space (i.e. sRGB). Because CIE XYZ color space provides a bigger set of colors (“gamut”) than the desired color space in general, some gamut mapping algorithm has to be applied.

In this thesis, I will examine a set of cameras with various absorption spectra. This can be compared with CIE XYZ color matching functions. Because various absorption spectra give results in different color spaces, a transformation in CIE XYZ color space is required which is approximated linearly with a linear least squares fit.

Finally, the gamuts of the cameras are compared with sRGB and CIE XYZ gamut. This gives criteria to decide, which camera is appropriate for supporting gamut mapping under various conditions.

The first is the luminous efficacy, given by integrating the absorptions spectra of the camera. A camera with small luminous efficacy tends to produce more noise than a camera with high luminous efficacy. The second criterion is the applicability of linear transformation model to the camera color space. On the one hand, the camera gamut should match the target gamut as close as possible, on the other hand, the colors should not be distorted too much, to uphold the impression of the image. Besides this, the shape of the gamut should be convex and not intersect the target gamut.

Contents

1. Introduction	1
1.1. Light	1
1.1.1. Spectrum	2
1.1.2. The Eye	2
1.2. The rendering process	3
1.3. Structure of this thesis	4
2. Color Spaces and Gamuts	5
3. Gamut Mapping	13
3.1. Algorithms	14
3.2. Advanced gamut mapping	18
4. Converting Spectra into sRGB	19
4.1. Standard converter	19
4.2. Camera simulation	20
4.3. Samples	22
4.4. Cameras	22
5. Evaluation	27
5.1. CIE XYZ color matching functions	28
5.2. SONY DXC 930	30
5.3. KODAK DCS 200	33
5.4. Comparison	37
6. Conclusion	39
Bibliography	41
Appendix	43
A. Cameras	43
B. Sample Images	45

1. Introduction

1.1. Light

Light can be seen as electromagnetic wave as well as particles. There are many very interesting effects that base on this dualism but in this thesis, I will focus on one major attribute of an electromagnetic wave: its wavelength λ . Counterintuitively, one light particle (*photon*) has a distinct energy E , given by its frequency ν . The frequency depends on the wavelength and the speed of light.

$$\nu = \frac{c}{\lambda} \quad (1.1)$$

$$E = h \cdot \nu \quad (1.2)$$

Where h denotes Planck's constant and c the speed of light. So increasing the intensity of electromagnetic radiation means increasing the number of photons since the energy of one photon is strictly determined by its wavelength. [YF12]

“The electromagnetic spectrum encompasses electromagnetic waves of all frequencies and wavelength.” [YF12, p. 1054] That is, from high frequency or long-wave gamma rays to low frequency or short-wave radio waves. Since this thesis is about colors, especially the visible part of the spectrum is of interest. Wavelengths of visible electromagnetic range from about 380nm (violet) to 750nm (red) [YF12, p. 1054]

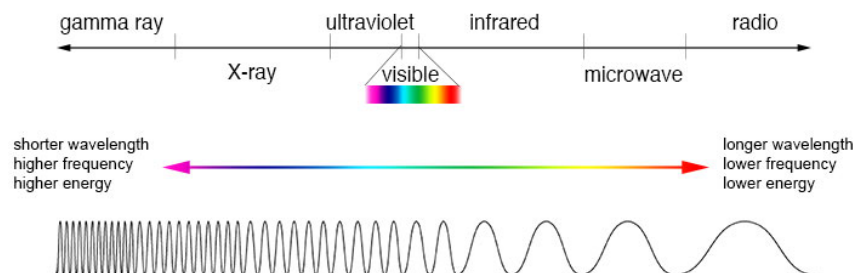


Figure 1.1.: Electromagnetic spectrum. ¹

¹<http://imagine.gsfc.nasa.gov/docs/science/known11/emspectrum.html>

1.1.1. Spectrum

An *emission spectrum* of a radiation source describes, how many photons of a certain wavelength are emitted by this source. In contrast, a *reflectance spectrum* of a surface describes, how many photons are reflected by that surface. Regarding the conservation of energy, the energy of the reflected photons must be lower or at most equal to the energy of the incident photon. That is, the reflectance of a certain wavelength is lower or equal to one.

For example, seeing green grass under sunlight. First of all, the light is emitted by the sun in space with a certain energy distribution $L(\lambda)$. When the photons hit the atmosphere of the earth, they are filtered. For example, ozone filters short-wave (and invisible) photons. Without this gas, those hazardous rays would hit the surface. The atmosphere consists of a mixture of many gases with different spectral transmittances that filter various wavelengths. The part of the spectrum which was not filtered then hits the grass stalk. Here, the reflective spectrum describes how much energy of a certain wavelength becomes reflected. Mathematically, the spectral distribution of the light after reflection is determined by point wise multiplying the filtered emission spectrum by the reflectance spectrum of the grass stalk. The rest of the energy is mostly transformed into heat or, in case of grass, it is used for photosynthesis. Then, the spectrum finally hits the eye. For a more detailed description see [SAM09, pp. 531 ff.].

1.1.2. The Eye

The eye of a human being is a very complex thing. But to understand how a color effect comes up, we only need to know that there are two different kinds of light sensors in the eye: rod and cone cells, where only the cone cells are responsible for color seeing and can be divided furthermore into three groups: long-wave, middle-wave and short-wave cones. Each of this cone cells types is sensitive for certain wavelengths only (see Figure 1.2). [WS82, pp. 90 f.], [SAM09, p. 535]

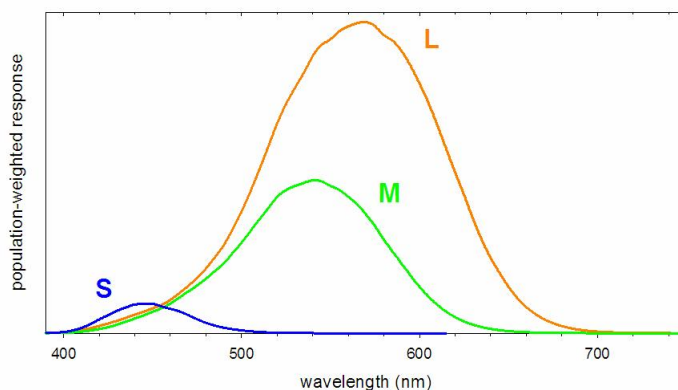


Figure 1.2.: Response curves of the short-wave, middle-wave and long-wave cones of the human eye. ²

When for example 450nm-rays hit the eye, the medium- and long-wave cones do not care but the short-wave cone becomes active and sends a signal to the brain. Since the brain did not receive any signals from the other cones, it interprets this color as “blue”. Analogue, 550nm-rays are recognized only by medium- and long-wave cones. The brain identifies “green” light. But there is no wavelength, that produces a white color impression. To

²<http://www.chm.davidson.edu/vce/coordchem/color.html>

produce “white light”, or any desaturated color, a bunch of rays of different wavelength is needed.

Thus, the color impression can be described by three quantities. This effect is known as trichromacy [BH97], [WS82, pp. 583 ff.]. A ramification is, that significantly different spectra may look exactly the same for an observer. This is known as metamerism. Note, that there is an infinite number of mixtures of wavelengths that produce the same color impression. Two different spectra producing the same color impression are said to be metamers.

For example, see the figure 1.3. Whereas the two spectra differ substantially, the resulting triples match exactly.

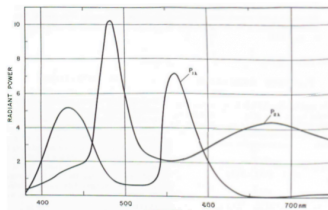


Figure 1.3.: Two emission spectra defining the metameric color stimuli [WS82]

Animals or technical devices may have sensors with different response curves than the human cone cells or even more or less than three types of them. Hence, they may distinguish what looks the same for a human or maybe cannot distinguish what is distinguishably for us. Conceive a gray scale camera which cannot distinguish colors since it has only one “cone” type.

1.2. The rendering process

Rendering an image means to translate a three dimensional scene into a two dimensional image. One render principle is *ray tracing*. “Ray tracing is actually a very simple algorithm; it is based on following the path of a ray of light through a scene as it interacts with and bounces off objects in an environment” [PH10]. With this approach, it is easy to implement effects as illumination with shadows, reflection, transparency, . . . by introducing secondary rays. For example, a camera sends a ray in a specific direction to determine the color of the corresponding pixel. The color of the pixel is then given by the color of the surface the ray hits first. To calculate the color of the surface, further rays might be emitted to detect whether a light source illuminates this point or to acquire the color of an other surface in case of reflectivity. For a detailed description of a ray tracing and computer graphics in general, see [SAM09].

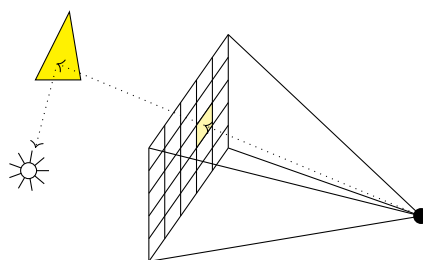


Figure 1.4.: Principle of a ray tracer

Having a white light source and an orange surface, RGB light transport principally multiplies the color of light source and surface color component wise, which gives orange, as

expected.

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \cdot 1 \\ 1 \cdot 1 \\ 1 \cdot 0 \end{pmatrix} \quad (1.3)$$

But with spectral light transport, the spectra are multiplied. The orange might have a reflectance spectrum with solely few small bands, that is, only photons with wavelength in one of those bands are reflected and all other wavelengths are absorbed. It is easy to construct an emission spectrum that produces a white color impression and lacks of just those two bands. The result is a black surface, since no photon emitted by the light source is reflected by the surface.

Beside the metamerism effect, there are some more real world effects that RGB light transport cannot simulate. For example refraction of light due to a change in its transmission medium depends on its wavelength (dispersion) [YF12, pp. 1080 ff.]. This causes the rainbow effect when sunlight runs through a prism. Because the distribution of wavelength is unknown in RGB light transport, the dispersion can only be guessed in this case. Consider a low pressure sodium vapor lamp which emits light in virtually the same color as a dimmed light bulb. Whereas the light of a incandescent bulb contains many wavelengths distributed over the whole visible spectrum, the wavelength emitted by the vapor lamp range from 580nm to 600nm. Therefore, the light of a bulb is separated into a continuous spectrum, while the vapor lamps passes and is little split. See [WS82, pp. 1 – 30] for a set of emission spectra and [WS82, pp. 51 – 63] for reflectance spectra.

Many other effects are wavelength based and can thus be not simulated by RGB light transport but only be estimated.

There are some renderers which use spectral light transport like *pbrt* [PH10] and thus can simulate those effects properly. But there are some problems that come with spectral light transport. For example, dispersion effects can only arise, when rays of different wavelength are calculated separately, because rays with distinct wavelength behave differently. This can increase render time significantly.

One major problem is the incompatibility with RGB images. Those are widely used as textures since they are memory efficient and widely available in contrast to spectral resolved images. But RGB textures cannot be used to calculate reflectances with spectra or to determine spectral emission. The reconstruction of a spectrum from RGB coordinates is principally impossible since information is lost. But there are a number of algorithms (e.g. [Smi99], [Gla89]) that try to estimate a meaningful result.

This field is very interesting but in this thesis, I will focus on the inverse direction. Once the spectral image is rendered, one want to have a RGB image to display it on a computer monitor or TV as well as share it with other people in common image formats. Hence, each pixel of the image must be transformed into RGB coordinates.

1.3. Structure of this thesis

First, I will give an introduction in color spaces and gamuts in general. Then, I want to present some classical techniques for gamut mapping. In the forth chapter there are some camera color matching functions being introduced with a short analysis of their color matching functions and gamuts, which are finally compared and further analyzed in evaluation.

2. Color Spaces and Gamuts

While naming colors is a suitable way in colloquial language (“red”, “lime green”, ...), a more technical way is adequate to describe colors of an image exactly. One way to specify a color is the spectral representation, but handling this is ineffective because it is large in memory and difficult to interpret. Therefore, one introduced *color spaces*: Mathematical spaces where every color has distinct coordinates. One prominent example is the RGB color space. It is determined by the primary valences red, green and blue. For example, the coordinates (0.2, 0.8, 0.2) give a mixture of four-fifth portion of red and a fifth portion red and blue (which is “lime green”). Precisely, the RGB color space is defined used the CIE XYZ color space and in fact [BB09], there are many different definitions of RGB ([SAM09, p. 546]).

A *gamut* is a set of colors inside a color space. For example, a monitor can display a specific set of colors which is called the monitor gamut. Similar, a printer combined with specific paper can produce a specific set of colors and a camera or an eye can see a specific set of colors [Mor08].

Colors that lie outside the gamut of the eye are called *imaginary colors*. Although they do not exist in real world, there is a representation (coordinates), that describe such colors in color space theoretically. Imaginary colors are usefull to describe color spaces such as CIE XYZ.

Gamuts and color spaces are associated because it is often easier to describe a gamut by one particular color space. Nevertheless it is very important to distinguish the terms gamut and color space.

CIE RGB color space

RGB space uses the colors red, green and blue to span a three dimensional color space. Because “red”, “green” and “blue” were undefined so far, CIE standardized the *CIE RGB* color space. Therefore, they chose three light sources, each emitting monochromatic light of wavelength 700 nm (red), 546.1 nm (green) and 435.8 nm (blue). Then test observers tried to match different color effects (B side) with the overlay of the lights (A side) by adjusting their intensities [SB02].

To obtain the \bar{r} , \bar{g} and \bar{b} functions, they illuminated the B side with monochromatic light of wavelength λ . The intensities of the red, green and blue light sources then denoted the value of the \bar{r} , \bar{g} and \bar{b} at λ .

But some greenish color effects could not be reproduced. To match them anyway, they illuminated the B side with the red light too. Therefore, the \bar{r} function is negative in a particular range around 500nm [FBH97]. In fact, it is impossible to mix certain green chromaticities using only the three lights above. Thus, the gamut of any device using solely those primaries does not contain this particular green.

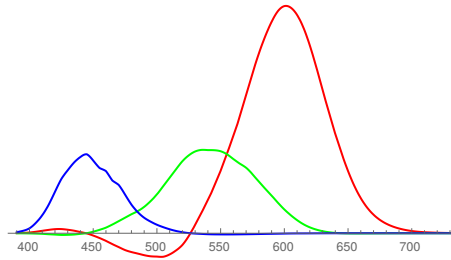


Figure 2.1.: CIE RGB color matching functions [SB55]¹

To obtain CIE RGB coordinates from a spectrum, one must multiply the color matching functions point wise by the spectrum and then integrate the result over the range where the functions are non zero.

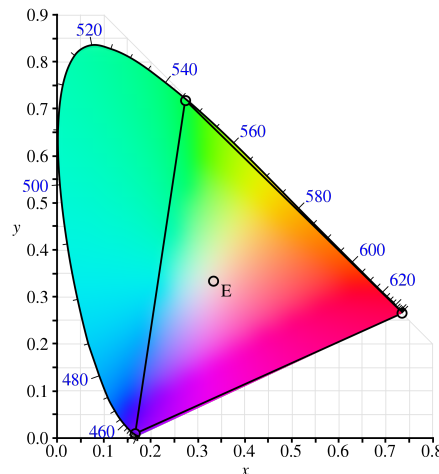


Figure 2.2.: CIE RGB primaries and gamut. Chromaticities inside the triangle can be described with only positive portions of primaries².

CIE XYZ color space

To obtain color matching functions that are entirely positive, CIE defined three imaginary colors X, Y and Z. Now the whole gamut of the human eye can be covered with a (positive) mixture of X, Y and Z, but not all positive mixtures of X, Y and Z lie inside the gamut of the eye (of course, pure X lies outside since it is an imaginary color).

The CIE XYZ gamut is defined to contain all colors a human eye can see. Its shape inside the XYZ color space is an infinite cone with a horseshoe-shaped footprint and its peak at (0, 0, 0) (see figure 2.4).

Converting a spectrum to CIE XYZ coordinates is similar to converting a spectrum to CIE RGB coordinates: It must be multiplied point wise by each of the \bar{x} , \bar{y} and \bar{z} functions, then the result must be integrated over the range where the functions are non zero.

¹Provided by <http://www.cvrl.org/cmfs.htm>

²https://commons.wikimedia.org/wiki/File:CIE1931xy_CIERGB.svg

³Provided by <http://www.cvrl.org/cmfs.htm>

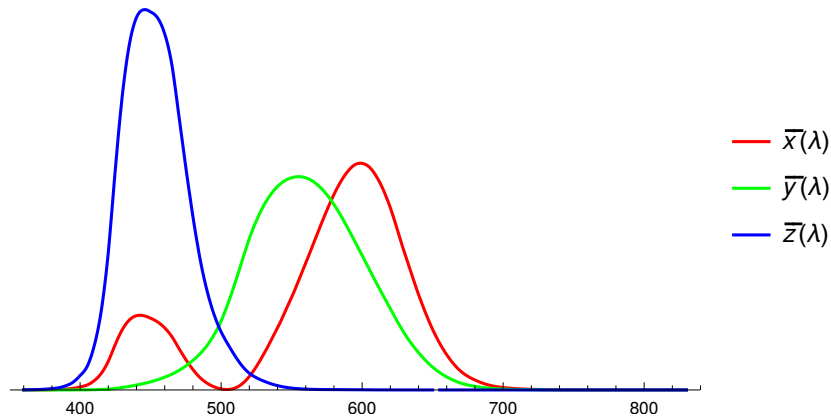


Figure 2.3.: CIE XYZ color matching functions [SSF99]³

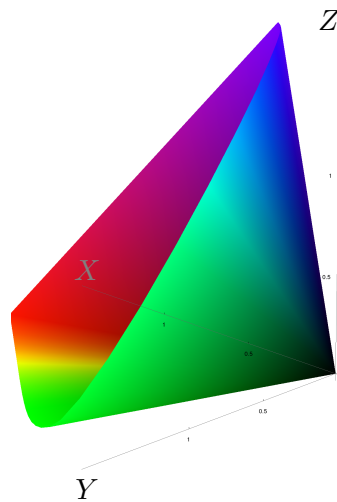


Figure 2.4.: CIE XYZ gamut

xyY color space

Projecting a color onto the plane where X, Y and Z coordinates sum to one leads to the x and y coordinates of the xyY color space.

$$x := \frac{X}{X + Y + Z} \quad (2.1)$$

$$y := \frac{Y}{X + Y + Z} \quad (2.2)$$

This is motivated by the fact, that multiplying X, Y and Z intensities by the same factor does not affect the color effect but only luminance. Since the eye is most sensitive for greenish colors, Y is defined to be the luminance. With the xyY color space, chromaticity and luminance are separated and each luminance and chromaticity can be manipulated without affecting the other.

A little math and the fact that

$$z := \frac{Z}{X + Y + Z} = 1 - x - y \quad (2.3)$$

reveals the inverse formulas

$$X = \frac{Y}{y} \cdot x \quad (2.4)$$

$$Z = \frac{Y}{y} \cdot (1 - x - y) \quad (2.5)$$

with Y value left unchanged.

Since the XYZ gamut cone extends infinite in direction orthogonal to the projection plane, there is no limit for luminance. The xyY color space allows to display the gamut of all XYZ *chromaticities* ignoring luminance in two dimensional space and is thus very descriptive and convenient.

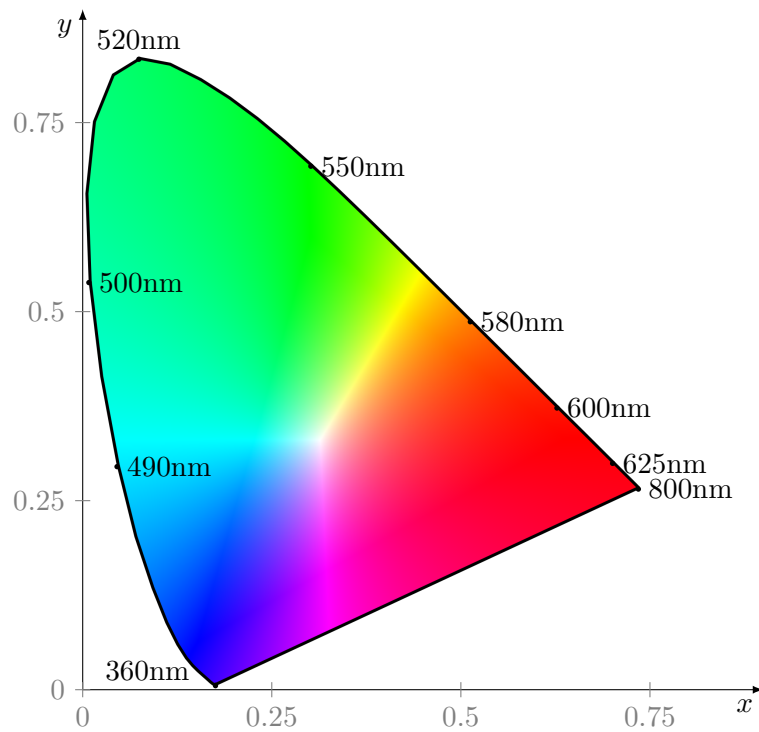


Figure 2.5.: Chromaticity diagram with arbitrary luminance

On the edge of the horseshoe shape, there are the monochromatic chromaticities, that is photons of one pure wavelength. The shape is filled with mixtures of monochromatic light and the straight line at the bottom is known as the line of purples. Chromaticities on that line cannot be produced with monochromatic light. [GM99, pp. 577 f.]

RGB color space

In contrast to CIE RGB color spaces, that covers as many colors as possible, sRGB color space was introduced to match the gamut of typical displays. There are many other RGB standards, such as Adobe RGB, but in this thesis, only sRGB is used.

The sRGB color spaces bases on the *ITU-R Recommendation BT.709*, abbreviated as *REC. 709*. The REC. 709 color space is defined as a linear transformation of CIE XYZ color space with primaries red, green and blue at certain positions in CIE XYZ color space, see figure 2.6. This transformation can be expressed as a matrix M [SAM09, p. 544].

To retrieve sRGB coordinates from REC. 709 coordinates, a gamma correction must be applied. For further information about gamma correction, see [SAM09, p. 544],[BB09].

Transform CIE XYZ coordinates in REC. 709 color space:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.6)$$

where

$$M^{-1} = \begin{pmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{pmatrix} \quad (2.7)$$

and R', G', B' denote the REC. 709 coordinates (before gamma correction).

To obtain XYZ coordinates from sRGB coordinates, gamma correction has to be inverted which leads to R', G', S' values (REC. 709 coordinates) which have to be multiplied by M .

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \cdot \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} \quad (2.8)$$

with

$$M = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1191 & 0.9505 \end{pmatrix}. \quad (2.9)$$

sRGB and REC. 709 gamut are defined to be the set of colors that have coordinates

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} \in [0, 1]^3.$$

in REC. 709 color space. Thus, this gamut is often denoted as the *sRGB cuboid*. It can be calculated easily, see 2.6. Note that gamma correction does not affect the coordinates of the primaries, since components with value zero or one are not affected by gamma correction.

In the xy-chromaticity diagram, the sRGB gamut is depicted by a triangle, spanned by red, green and blue primaries. A chromaticity can be described by barycentric coordinates based on those primaries. That is, chromaticities inside the triangle (i.e. within REC. 709 gamut) have positive barycentric coordinates whereas chromaticities outside the triangle have at least one negative coordinate. This suits well with the observation that multiplying XYZ coordinates (and RGB coordinates as well) only affects luminance. If a XYZ coordinate triple lies outside sRGB gamut because of its chromaticity, than multiplying it with any (positive) factor does not bring it inside since there is still a negative coordinate. On the other hand, if a XYZ coordinate triple lies inside sRGB gamut, multiplying it with any (positive) factor will not bring its chromaticity out of the triangle.

In case that luminance cannot be ignored, the sRGB gamut can be depicted in three dimensional xyY color space too (see figure 2.7). The height of the body denotes its luminance. It shows that the body is the highest at its white point and in greenish regions. Thus gray and green chromaticities can have high luminance inside sRGB gamut, compared to blue or red chromaticities.

	RGB	XYZ	xyY
red	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0.4125 \\ 0.2127 \\ 0.0193 \end{pmatrix}$	$\begin{pmatrix} 0.64 \\ 0.33 \\ 0.2127 \end{pmatrix}$
blue	$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0.3576 \\ 0.7152 \\ 0.1192 \end{pmatrix}$	$\begin{pmatrix} 0.3 \\ 0.6 \\ 0.7152 \end{pmatrix}$
green	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0.1804 \\ 0.0722 \\ 0.9503 \end{pmatrix}$	$\begin{pmatrix} 0.15 \\ 0.06 \\ 0.0722 \end{pmatrix}$
white	$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0.9505 \\ 1 \\ 1.0888 \end{pmatrix}$	$\begin{pmatrix} 0.31272 \\ 0.3290 \\ 1 \end{pmatrix}$

Figure 2.6.: REC. 709 primaries red, green and blue and white point in XYZ and xyY color space.

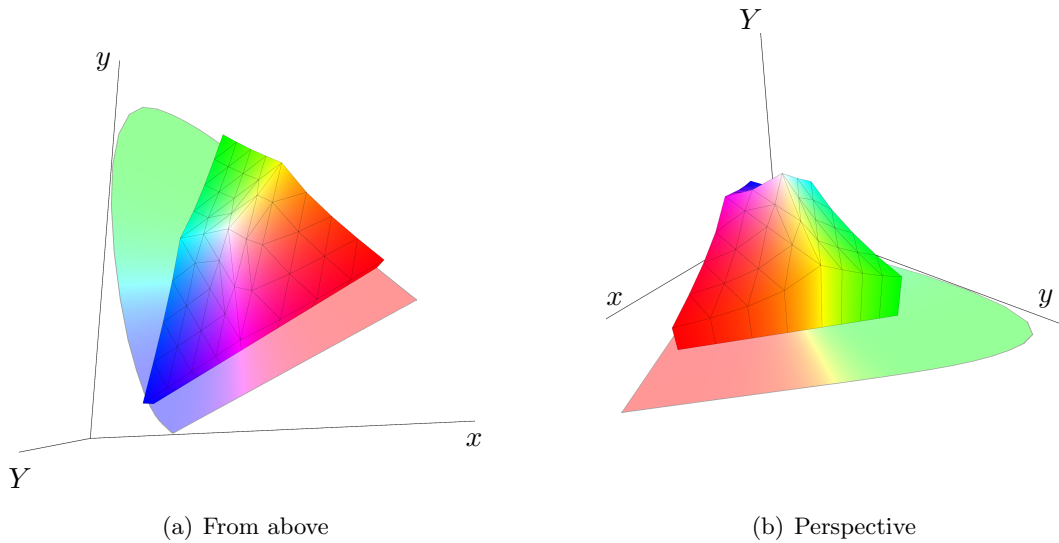


Figure 2.7.: sRGB gamut in xyY color space

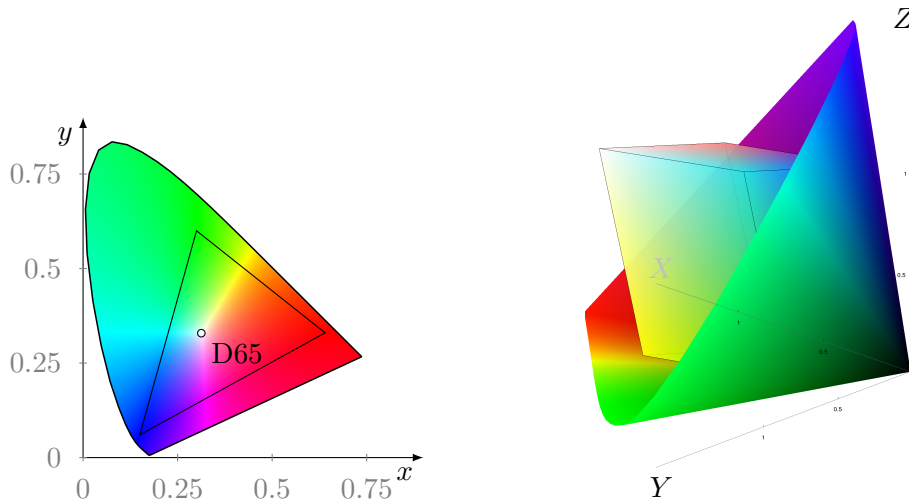


Figure 2.8.: REC. 709 gamut in chromaticity diagram

Figure 2.9.: sRGB gamut in XYZ color space

hsY color space

In some applications it is convenient to split x and y into saturation and hue. Knowing the coordinates of the white point w_x, w_y , this can be easily done in analogy to polar form of complex numbers:

$$\text{hue} = \arctan(y - w_y, x - w_x) \quad (2.10)$$

$$\text{saturation} = \left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} w_x \\ w_y \end{pmatrix} \right\|_2 \quad (2.11)$$

$$x = \text{saturation} \cdot \cos(\text{hue}) + w_x \quad (2.12)$$

$$y = \text{saturation} \cdot \sin(\text{hue}) + w_y \quad (2.13)$$

where \arctan denotes the arctangent function with two arguments and $\|\cdot\|_2$ the two norm. This concept is similar to HSV or HSL color space, but bases on CIE XYZ instead of some RGB color space.

CIELAB and CIELUV

One disadvantage of the XYZ and xyY color space is *non-uniformity*. That means, the euclidean distance of two colors in XYZ or xyY color space does not express their perceived difference. [Mor08, p.28]. Figure 2.10 shows *MacAdam ellipses*, plotted in xy chromaticity diagram (10 times their actual length). The chromaticities on the edge of an ellipse are perceived to have the same distance each to the chromaticity at the center of the ellipse.

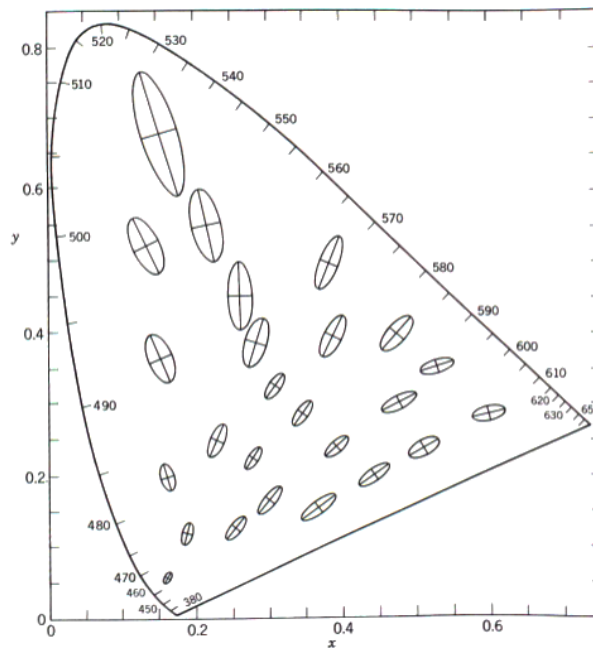


Figure 2.10.: MacAdam ellipses [WS82]

Similar to xyY, CIELAB and CIELUV base on the CIE XYZ color space and separate luminance from chromaticity. In order to achieve an uniform distribution of chromaticities, a nonlinear transformation is applied.

3. Gamut Mapping

Just as one distinguishes color space and gamut, one has to distinguish color space transformation and gamut mapping.

Color space transformation is a mathematically described transformation that is characteristic for a pair of color spaces. For example, transforming a color given in CIE XYZ color space into sRGB color space requires to multiply the XYZ component vector by the well known transformation matrix and apply a gamma correction. But since CIE XYZ gamut is a superset of REC. 709 gamut (i.e. sRGB gamut), the mapped color may lie out of REC. 709 gamut.

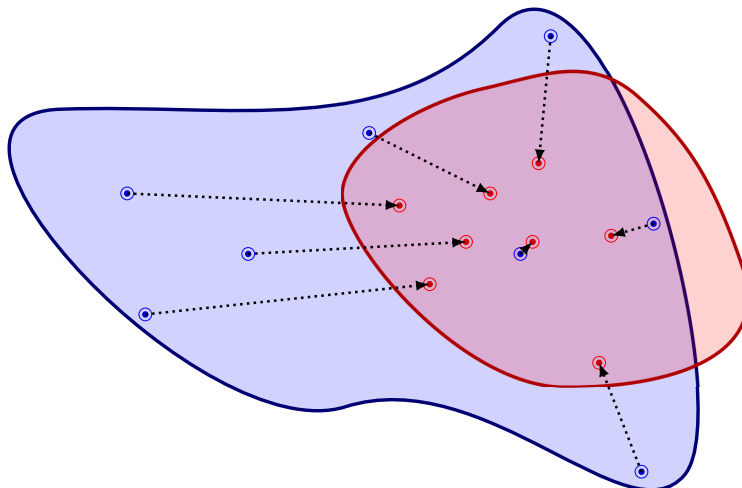


Figure 3.1.: Elements of the blue colored source gamut are mapped to fit into the red target gamut. In this example the gamuts intersect each other.

For example, a typical application for REC. 709 color space is PNG image format. PNG images can express only REC. 709 gamut (i.e. negative values are not designated). [BB09, p. 107]

To avoid this problem, one performs gamut mapping to map all colors of the source gamut into the target gamut. Gamut mapping can happen in arbitrary color space. The choice of a suitable temporary color space for gamut mapping is though very important, since it can simplify the algorithm significantly.

Gamut mapping is not limited to shrink gamuts. It might also be applied to extend gamuts or map between intersecting ones. But in the context of this thesis, the target gamut is virtually always a subset of the source gamut. Thus I will restrict on gamut shrinking in the following.

Also, the following algorithms only handle chromaticity and try to keep luminance invariant. To ensure that luminance of a color is in the REC. 709 color space, it suffices to keep CIE XYZ coordinates lower than the corresponding white point coordinate since the transformation is linear (see 4.1).

Doing sophisticated luminance manipulation is known as “tone mapping” [SAM09, pp. 597 ff]. For further information see also [Ash02], [DMAC03].

3.1. Algorithms

For convenience, I introduce two new symbols

$$\delta_{\text{SRC}} = \delta_{\text{SRC}}(\text{hue}) \quad (3.1)$$

$$\delta_{\text{DST}} = \delta_{\text{DST}}(\text{hue}) \quad (3.2)$$

denoting the distance from white point to the source respectively destination gamut boundary at a specific hue. Note that δ_{SRC} , δ_{DST} is the greatest saturation for a given hue that is inside source gamut resp. target gamut.

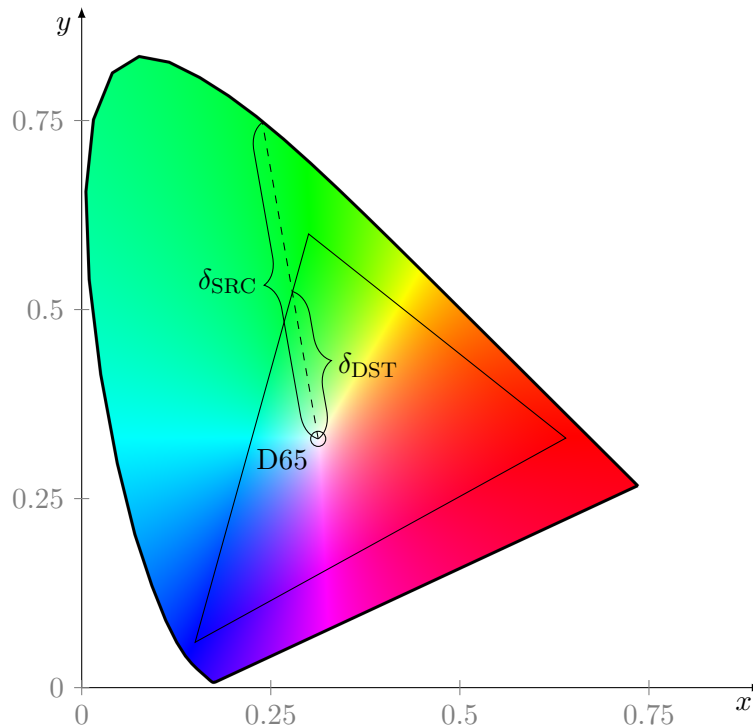


Figure 3.2.: δ_{SRC} and δ_{DST} in chromaticity diagram when source gamut is CIE XYZ gamut and target gamut is REC. 709 gamut.

Clamping components in REC. 709 color space

The easiest way to do gamut mapping is to clamp the coordinates in the target color space to zero. For example, let target space be the REC. 709 color space.

$$R \mapsto \begin{cases} R & R > 0 \\ 0 & \text{else} \end{cases} \quad (3.3)$$

$$G \mapsto \begin{cases} G & G > 0 \\ 0 & \text{else} \end{cases} \quad (3.4)$$

$$B \mapsto \begin{cases} B & B > 0 \\ 0 & \text{else} \end{cases} \quad (3.5)$$

This approach has some disadvantages. The first is, that luminance and hue are non constant. For example, see figure 3.3. A color c , that was a mint green before, is virtually pure green c' after mapping.

Second, the mapping is not smooth. That means, chromaticities lying little outside the REC. 709 gamut were mapped onto the same chromaticities than ones lying way out the gamut. So what was a gradient in original picture might be a region of constant color after gamut mapping and thus a lost of contrast. This leads to banding effects (see chapter 5).

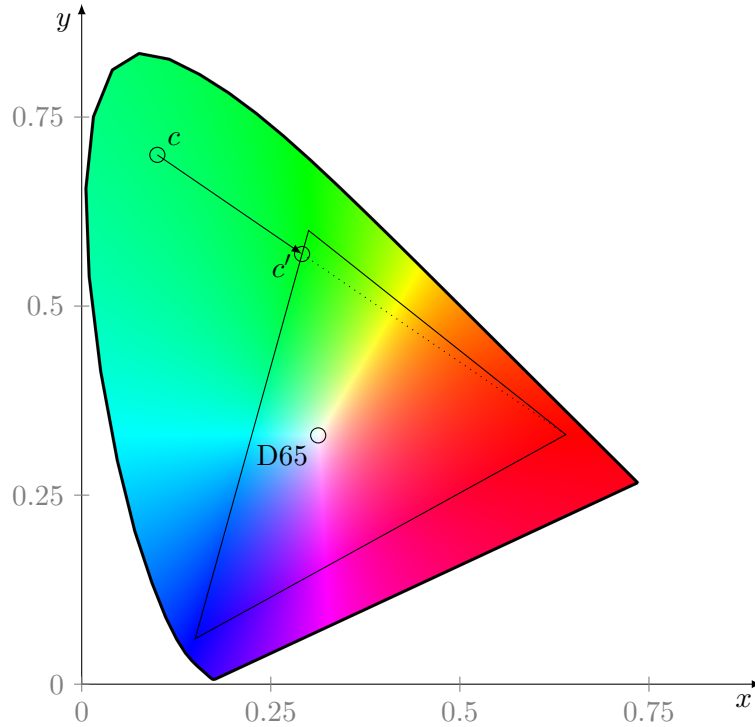


Figure 3.3.: Clamping in REC. 709 color space

Clamp saturation

This algorithm remedies the hue and luminance instability. This is achieved by doing the mapping in hsY color space. If saturation is greater δ_{DST} , it is clamped to this value. Hue and Lightness stay constant. (cf. [Mor08, pp. 98])

$$\text{hue} \mapsto \text{hue} \quad (3.6)$$

$$\text{saturation} \mapsto \begin{cases} \text{saturation} & \text{saturation} < \delta_{\text{DST}} \\ \delta_{\text{DST}} & \text{else} \end{cases} \quad (3.7)$$

$$Y \mapsto Y \quad (3.8)$$

This ensures that the mapped coordinates lie inside or on the edge of the target gamut. That is after transforming into target color space, all components are positive.

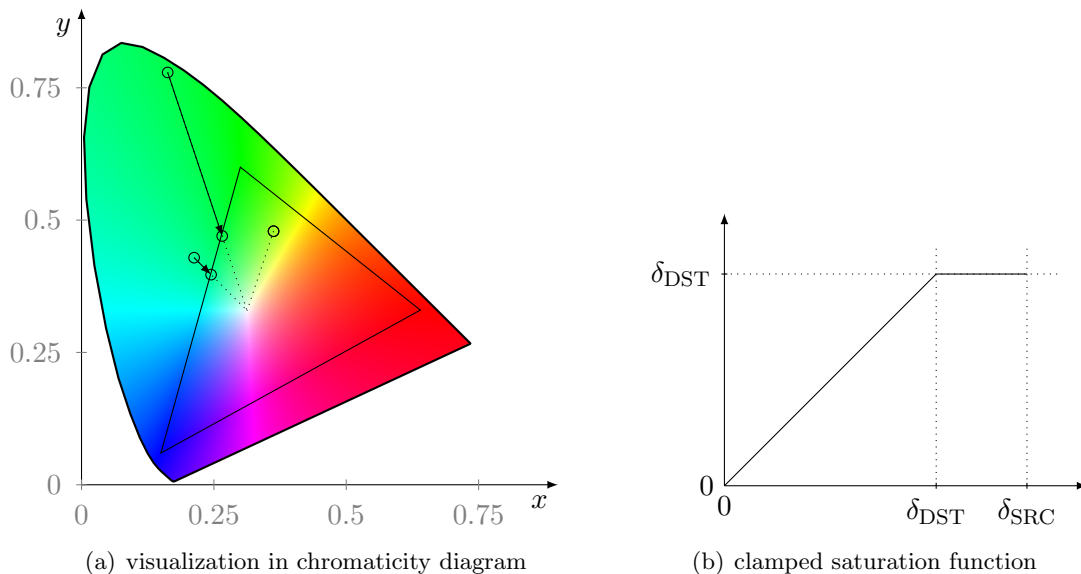


Figure 3.4.: Clamping in RGB color space

Linear mapping

Clamping the saturation solves the luminance and hue problem. But chromaticities little outside the target gamut are clamped to the same chromaticities than those way outside. This problem can be solved by mapping the saturation of chromaticities linear, i.e. chromaticities on the edge of the source gamut are mapped onto the edge of the target gamut, white stays and chromaticities in between are interpolated linearly.

This is achieved by multiplying the saturation with a factor depending on hue. The factor is calculated as the quotient of

$$\sigma(\text{hue}) := \frac{\delta_{\text{DST}}(\text{hue})}{\delta_{\text{SRC}}(\text{hue})} \quad (3.9)$$

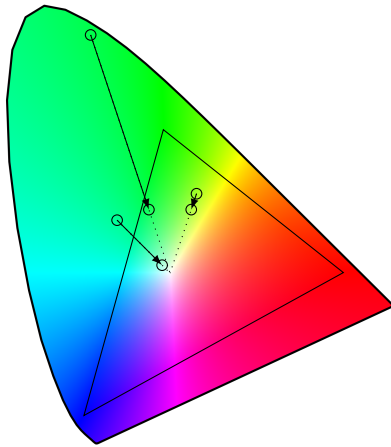
This ensures, that saturation is lower or equal δ_{DST} . And thus all mapped chromaticities lie inside target gamut.

$$\text{hue} \mapsto \text{hue} \quad (3.10)$$

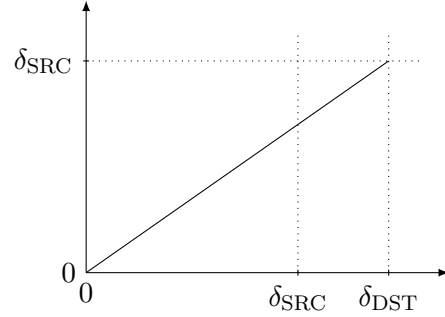
$$\text{saturation} \mapsto \sigma(\text{hue}) \cdot \text{saturation} \quad (3.11)$$

$$Y \mapsto Y \quad (3.12)$$

The disadvantage of this algorithm is, that it assumes to map chromaticities from the *whole* source gamut. A typical image covers only a part of the available gamut. Therefore, colors may be too weak saturated in the result. For example see figure 3.5(a). The leftmost chromaticity is mapped to almost white.



(a) visualization in chromaticity diagram



(b) linear saturation function

Cubic mapping

One way to solve the problem of weak saturated results is to adjust the saturation mapping function

$$f : [0, \delta_{\text{SRC}}] \rightarrow [0, \delta_{\text{DST}}] \quad (3.13)$$

Of course, shades of gray must stay gray, therefore is

$$f(0) = 0.$$

Because the highest saturated chromaticities in source gamut should map onto highest saturated chromaticities in target gamut, it is approved to set

$$f(\delta_{\text{SRC}}) = \delta_{\text{DST}}.$$

Furthermore one might stipulate

$$f'(0) = 1$$

and

$$f'(\delta_{\text{SRC}}) = 0$$

to approximate clamp mapping for low and high saturated chromaticities. The cubic polynomial

$$f(s) = \frac{\delta_{\text{SRC}} - 2\delta_{\text{DST}}}{\delta_{\text{SRC}}^3} \cdot s^3 + \frac{3 \cdot \delta_{\text{SRC}} \cdot \delta_{\text{DST}} - 2 \cdot \delta_{\text{SRC}}^2}{\delta_{\text{SRC}}^3} \cdot s^2 + s \quad (3.14)$$

fulfills these conditions.

Note that f is not monotonous when $\frac{\delta_{\text{SRC}}}{\delta_{\text{DST}}} < \frac{1}{3}$. Therefore, low saturated chromaticities may be mapped on higher saturated chromaticities and vice versa. Also, saturation of some chromaticities can be greater δ_{DST} .

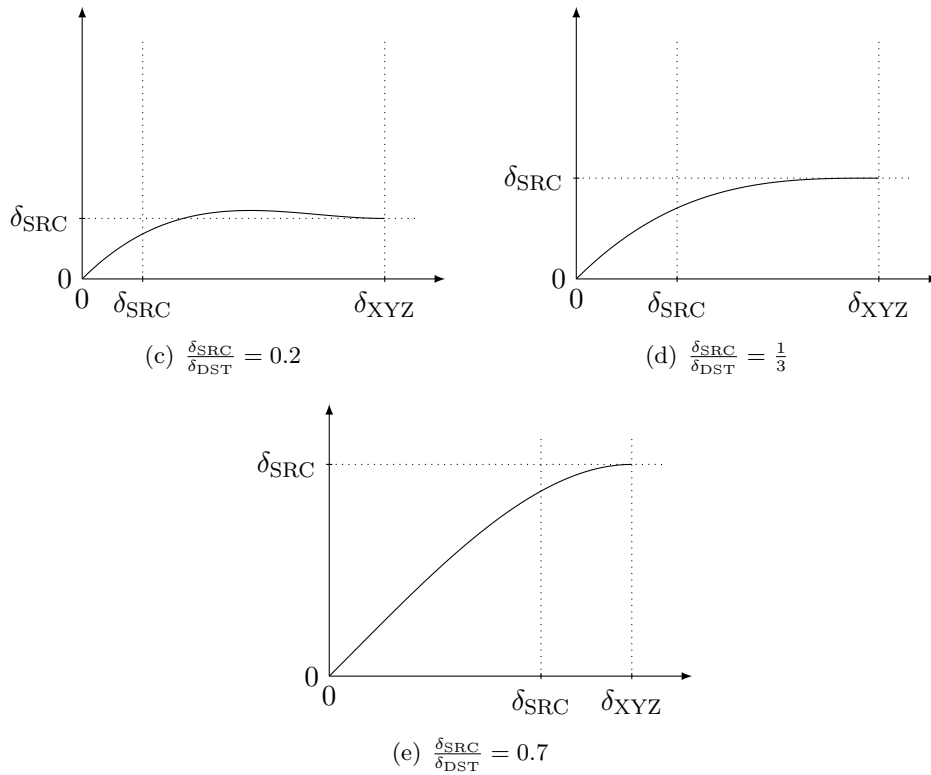


Figure 3.5.: Cubic mapping function with various σ_{SRC}

3.2. Advanced gamut mapping

One disadvantage of the algorithms so far is that they always assume that the chromaticities are distributed over the whole gamut. Since this is not quite the case in typical images, the resulting chromaticities are often not as saturated as they could be.

This issue could be handled by calculating some hull of all chromaticities of an image. But it is not easy to define and calculate a hull, that has suitable properties. For example. Convex hull can be calculated easy, but has problems with outliers and is often not close enough to the original data. For detailed description and evaluation of those gamut mapping algorithms, I refer to [Mor08].

4. Converting Spectra into sRGB

4.1. Standard converter

The standard pipeline to convert a spectrum into REC. 709 values consists of three steps. First, convert the spectrum into CIE XYZ coordinates, using the CIE XYZ color matching functions (see 2). Then, apply some sort of gamut mapping to fit all chromaticities into REC. 709 gamut and then transform it to REC. 709 coordinates, using the transformation matrix. Of course, REC. 709 gamut and color space can be replaced by arbitrary color spaces and gamuts, e.g. Adobe RGB is common. The principle stays the same.

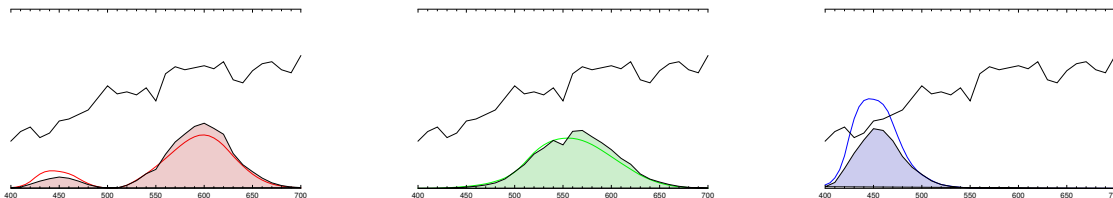


Figure 4.1.: Multiplying the same spectrum by each the \bar{x} , \bar{y} and \bar{z} function obtains three curves. The area under this curves is the X, Y resp. Z value.

The spectral data used here is given by a set of images. That is, the spectrum of one pixel consists of a number of samples in range from 0 to 255. So the absolute physical luminance of a spectrum is arbitrary and only the ratios of the luminances of the pixels are given.

This is no problem since there is no need for absolute luminances in the output image as well. To map all luminances of the spectrum, a normalization is applied so the brightest spectrum will result in the brightest possible pixel with corresponding chromaticity.

Normalization can be quite complex. The whitepoint of sRGB (D65) is at $w_X = 0.95047$, $w_Y = 1$, $w_Z = 1.08883$. The aim of the normalization is, that the components of all XYZ coordinates in one image are equal or less the white point coordinate. This insures, that their intensities are inside REC. 709 gamut [Wat, p. 439].

Coincidentally, the chromaticity must be invariant. Hence the XYZ components must be multiplied by *one* factor for all components. One way to do so, is to divide the components of each pixel by w_X , w_Y respectively w_Z , then finding the maximum m of all pixels component, divide each component by m and then to multiply the components of each pixel with the corresponding white points component again.

To fit the chromaticities in the target gamut, see (chapter 3).

Algorithm 1: Convert spectral image to XYZ image

Data: Color matching functions, Spectral image
Result: XYZ image
 Create an empty image of the same size as the spectral image;
 $max = 0$;
foreach *Pixel in spectral image* **do**
 $X = 0$;
 $Y = 0$;
 $Z = 0$;
 foreach *wavelength λ in spectrum of the pixel* **do**
 $vs =$ value of the spectrum at λ ;
 $vc =$ values of the color matching functions at λ ;
 $X += vs \cdot vc.x$;
 $Y += vs \cdot vc.y$;
 $Z += vs \cdot vc.z$;
 end
 $max = \max\{\frac{X}{w_X}, \frac{Y}{w_Y}, \frac{Z}{w_Z}, max\}$;
 Set pixel value in XYZ image = (X, Y, Z) ;
end
foreach *Pixel in spectral image* **do**
 Set pixel value = $\frac{\text{pixel value}}{max}$; // normalize value
end

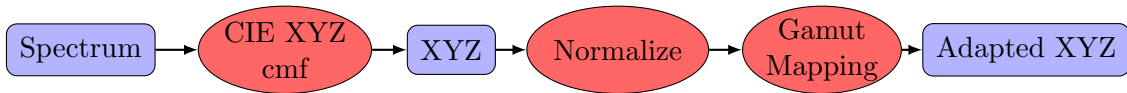


Figure 4.2.: Standard pipeline

4.2. Camera simulation

A camera can be characterized by its color matching functions. That is, similar to the CIE XYZ color matching functions, three curves describing the sensitivity of a sensor dependently from wavelength. Because those color matching functions differ significantly among themselves, their recognizable gamut differs as well.

The idea of simulating cameras is, that some cameras may have gamuts similar to REC. 709 and obviate the need for gamut mapping.

One problem is, that the resulting components are coordinates in an unspecific camera RGB color space and the transformation to standardized color spaces like CIE XYZ is unknown. In the following I assume that this transformation is linear (i.e. it can be expressed with a matrix). To find the matrix, a linear least squares fit is applied, wherefore a number of source samples s_i and target samples t_i is required.

Conveniently, let

$$\hat{t}_i := M \cdot s_i \tag{4.1}$$

The matrix M is characterized as the linear transformation, that minimizes the term

$$\left\| M \cdot \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} - \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} \right\| = \left\| \begin{pmatrix} \hat{t}_1 \\ \vdots \\ \hat{t}_n \end{pmatrix} - \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} \right\| \quad (4.2)$$

in some way. Note that \hat{t}_i as well as t_i and s_i are vectors themselves.

The easiest way for minimizing (4.2) is to minimize the norm of the vector of all components

$$\left\| \begin{pmatrix} \hat{t}_{1x} \\ \hat{t}_{1y} \\ \hat{t}_{1z} \\ \vdots \\ \hat{t}_{nz} \end{pmatrix} - \begin{pmatrix} t_{1x} \\ t_{1y} \\ t_{1z} \\ \vdots \\ t_{nz} \end{pmatrix} \right\| \quad (4.3)$$

with

$$\hat{t}_{ij} = (M \cdot s_i)_j, \quad (4.4)$$

$$= \left(\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \cdot \begin{pmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \end{pmatrix} \right)_j \quad (4.5)$$

$$= (m_{j1} \cdot s_{i1} + m_{j2} \cdot s_{i2} + m_{j3} \cdot s_{i3}) \quad (4.6)$$

$$(4.7)$$

with $1 \leq i \leq n$ and $1 \leq j \leq 3$.

Hence minimizing (4.3) is equivalent to minimizing

$$\xi := \left\| \underbrace{\begin{pmatrix} s_{11} & s_{12} & s_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{11} & s_{12} & s_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{21} & s_{22} & s_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_{21} & s_{22} & s_{23} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{n1} & s_{n2} & s_{n3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{n1} & s_{n2} & s_{n3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_{n1} & s_{n2} & s_{n3} \end{pmatrix}}_{=: \bar{S} \in \mathbb{R}^{3n \times 9}} \cdot \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \\ m_{33} \end{pmatrix}}_{=: \bar{m} \in \mathbb{R}^9} - \underbrace{\begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ t_{22} \\ t_{23} \\ \vdots \\ t_{n1} \\ t_{n2} \\ t_{n3} \end{pmatrix}}_{=: \bar{t} \in \mathbb{R}^{3n}} \right\| \quad (4.8)$$

Since the number of samples n is greater 9 in general, the system of linear equations

$$\bar{S} \cdot \bar{m} = \bar{t} \quad (4.9)$$

is over determined and hence not solvable except of special cases. Although, the error ξ can be minimized by solving the normal equation

$$\bar{S}^T \bar{S} \cdot \bar{m} = \bar{S}^T \bar{t} \quad (4.10)$$

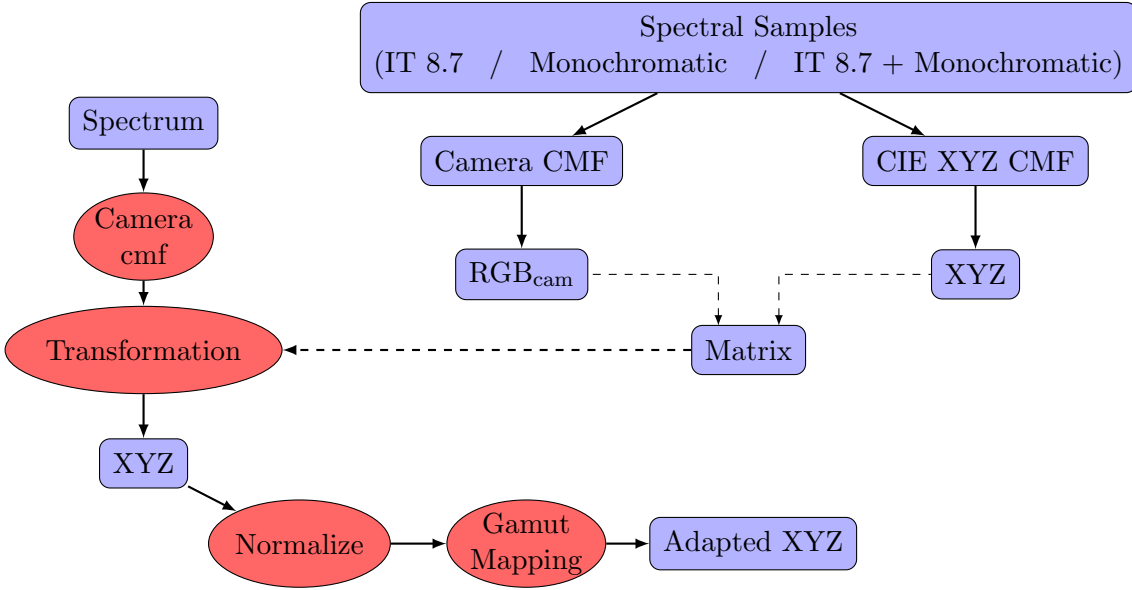


Figure 4.3.: Camera pipeline

Since $\bar{S}^T \bar{S} \in \mathbb{R}^{9 \times 9}$, $\bar{m} \in \mathbb{R}^9$, $\bar{S}^T \bar{t} \in \mathbb{R}^9$ and $\bar{S}^T \bar{S}$ is non-singular, this equation system is always solvable. And the components of \bar{m} denote the elements of M .

Note that the color space transformation takes place in three dimensional space. After that, the resulting gamut is projected on xy-plane. Hence, exchanging the camera color matching functions affects not only chromaticity but also luminance. Empirical analysis of the resulting images exhibits that the linear transformation can compensate this well. Therefore, I will focus on chromaticity only.

4.3. Samples

Obviously, the quality of the matrix depends on the goodness of the samples. Hence they should be carefully chosen. Wolf Faust metered the spectral reflectances of an IT 8.7 calibration target and provides the data on his website [Fau14]. For this thesis, I use the N130501 charge.

The patches of the calibration target provide many different colors distributed over the important regions of XYZ gamut (see 4.4(b)).

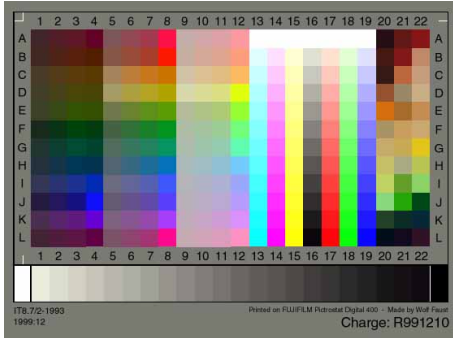
Beside of this, I will use monochromatic chromaticities, because IT 8.7 chromaticities are thinly distributed in high saturated regions, and set of both monochromatic and IT 8.7 samples. This yields three different matrices for each camera.

4.4. Cameras

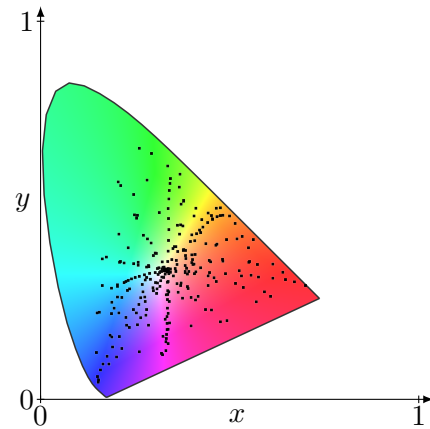
There are twelve sets of color matching functions given by [RKI13]. For sake of clarity, I will confine on three cameras with specific characteristics. The functions are sampled from 400nm to 700nm in 4nm steps.

The integral under the color matching functions gives the luminous efficiency. That is, the bigger the integral of the color matching functions, the less it is susceptible for amplifying noise. Hence, the integrals of the color matching functions provide a numerical quantity to assess the cameras.

Because the linear transform is only approximative correct, the camera gamut may intersect the XYZ gamut. This is an error, because no camera can produce such colors. In fact,



(a) IT 8.7 calibration target



(b) Chromaticities in IT 8.7 calibration target

some cameras might be able to recognize wavelength out of the human visible spectrum, but those “colors” can never be seen by a human, regardless the displaying device. If the camera gamut regions out of CIE XYZ gamut are not too big, this problem can be ignored here. That is, when the camera produces a color outside the CIE XYZ gamut, this color is treated as if it was inside the CIE XYZ gamut.

SONY DXC 930

The color matching functions of the SONY DXC 930 are similar to the CIE XYZ color matching functions with \bar{b} function having a large amplitude at the begin. Second and third function are more separated and third \bar{r} function has only one peak. This results in a gamut similar to CIE XYZ gamut and reflects in the value of the integrals of the functions.

$$\int \bar{r}(\lambda) d\lambda = 4.54964$$

$$\int \bar{g}(\lambda) d\lambda = 6.84325$$

$$\int \bar{b}(\lambda) d\lambda = 17.7768$$

Note that the gamut is not completely inside the horseshoe (see figure 4.5). Using the monochromatic samples, the camera gamut lies almost in the CIE XYZ color space, but REC. 709 color space is not completely included. Using the IT 8.7 samples to determine the transformation matrix, the camera gamut includes sRGB color space but also greater regions out of CIE XYZ gamut. The regions out of CIE XYZ gamut are small, so this is no problem.

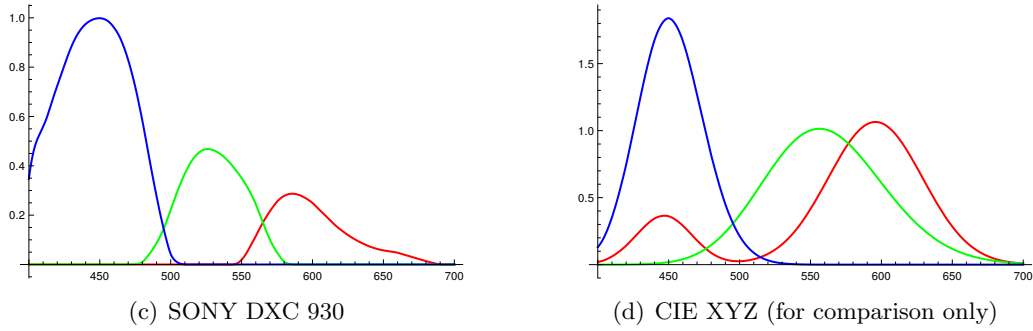


Figure 4.4.: Color matching functions

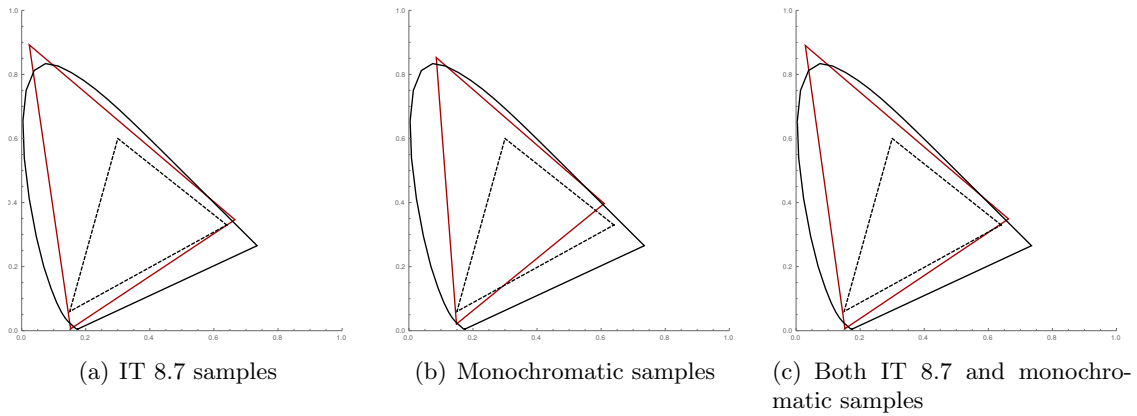


Figure 4.5.: Chromaticity diagram with CIE XYZ gamut (black), REC. 709 gamut (dashed) and SONY DXC 930 gamut (dark red)

KODAK DCS 200

The color matching functions of the KODAK DCS 200 camera differ significantly from the CIE XYZ color matching functions. The \bar{b} function has small amplitude whereas the others have much larger amplitudes. Again, the third function has – unlike CIE XYZ color matching functions – only one peak.

The sensitivity of the blue curve is thus much smaller than sensitivity of red and green curve.

$$\int \bar{r}(\lambda) d\lambda = 11.5798$$

$$\int \bar{g}(\lambda) d\lambda = 19.8605$$

$$\int \bar{b}(\lambda) d\lambda = 3.72188$$

This traits result in a gamut similar to REC. 709 (see figure 4.7). Like the gamut of SONY DXC 930, the KODAK DCS 200 gamut intersects the CIE XYZ gamut and like before, colors outside CIE XYZ gamut are treated as if they were inside. Using the IT 8.7 samples for matrix determination yields in a gamut quite close to REC. 709 gamut. This fact is interesting because it may diminish the importance of gamut mapping, of which more later.

As it is already in 4.4, the linear color space transformation from KODAK DCS 200 color space to CIE XYZ fits quite well. See also chapter 5 for further analyzes and comparisons.

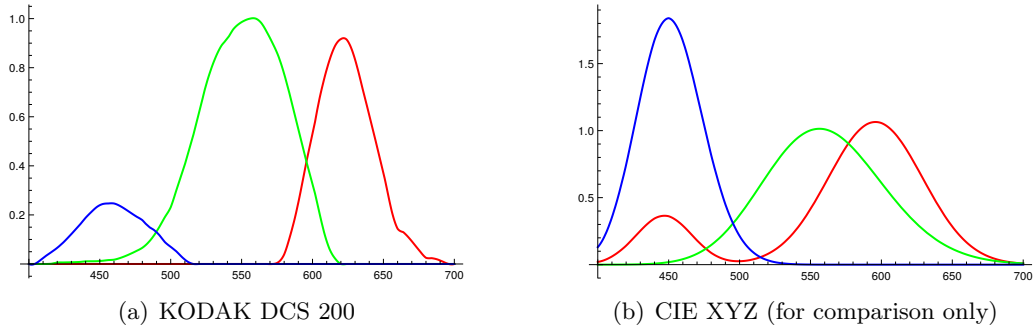


Figure 4.6.: Color matching functions

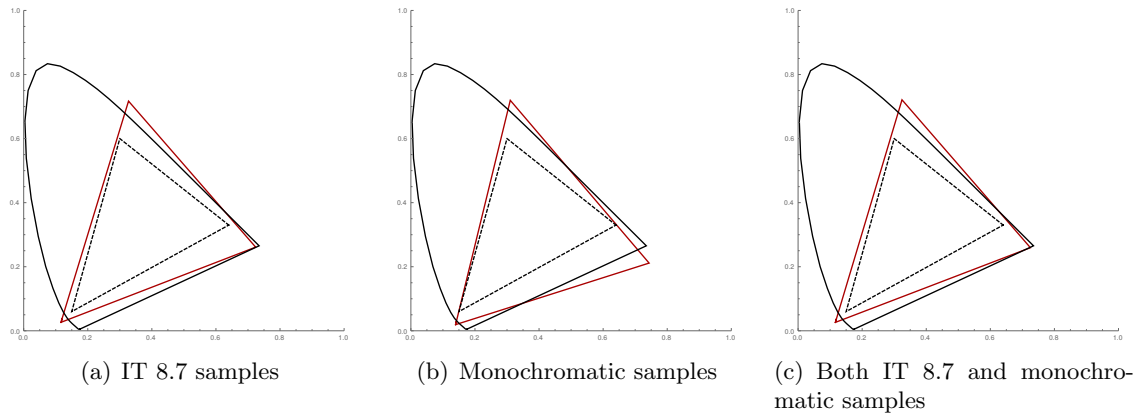


Figure 4.7.: Chromaticity diagram with CIE XYZ gamut (black), REC. 709 gamut (dashed) and KODAK DCS 200 gamut (dark red)

KODAK DCS 460

Unlike the well separated functions before, all three functions are non zero in a large interval about from 450nm to 680nm, that is a significant part of the whole visible spectrum. Especially the \bar{b} function has no out standing peak.

The sensitivity benefits significantly of this, but the transformation from camera RGB to XYZ is not well adaptable by a linear model and hence the gamut boundary displayed in figure 4.9 is distorted.

$$\int \bar{r}(\lambda) d\lambda = 14.9773$$

$$\int \bar{g}(\lambda) d\lambda = 22.5834$$

$$\int \bar{b}(\lambda) d\lambda = 11.4067$$

Note that, although the transformation from camera RGB to CIE XYZ is linear, the transformation from CIE XYZ to xyY is not. Because the latter transformation is a projection, small errors nearby the black point can have a big impact on xy coordinates.

Nevertheless, the non convexity is impractical, since mixtures of monochromatic (i.e. maximal saturated) chromaticities cannot be more saturated than the monochromatic chromaticities itself. With a non convex gamut boundary, linear and cubic gamut mapping does not work anymore since they rely on the fact, that all colors (i.e. all convex combinations of monochromatic colors) lie *inside* the gamut (cf. images in Appendix).

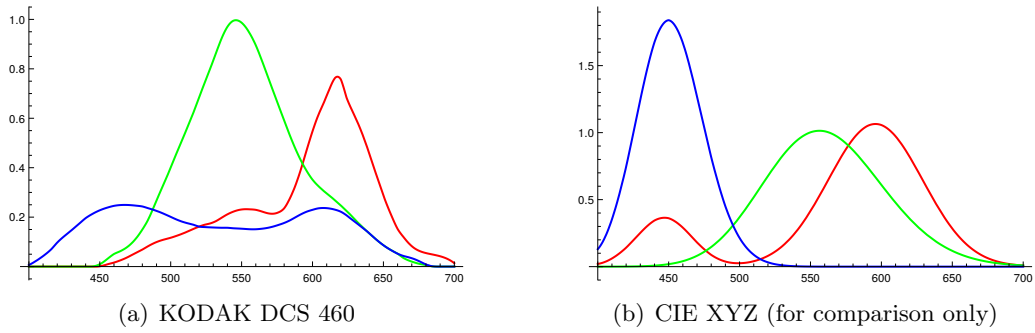


Figure 4.8.: Color matching functions

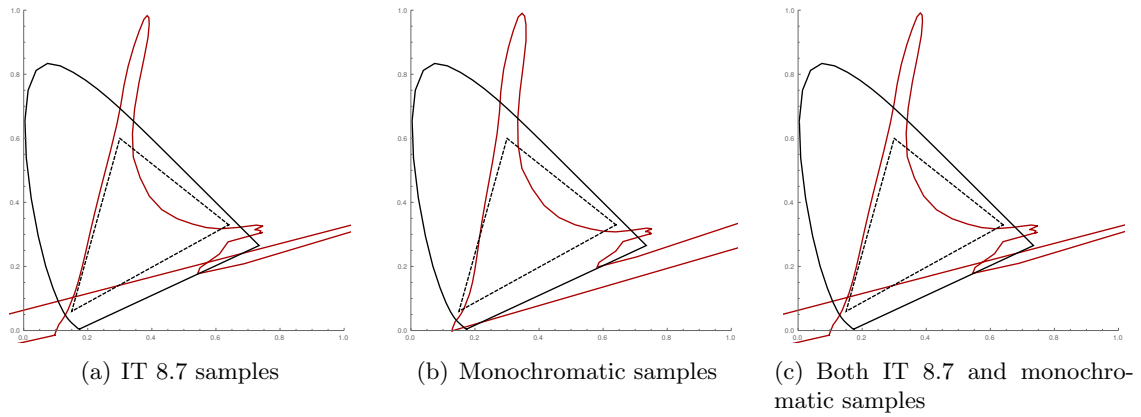


Figure 4.9.: Chromaticity diagram with CIE XYZ gamut (black), REC. 709 gamut (dashed) and KODAK DCS 460 gamut (dark red)

So, without a better transformation model, this camera is useless unfortunately.

5. Evaluation

Using camera color matching functions instead of CIE XYZ color matching functions may operate as gamut mapping at spectral level. Hence I want to compare different cameras and evaluate their suitability for replacing or at least supporting gamut mapping.

There are four kinds of graphics used in the following. The first is the sRGB image as it comes out of the described pipeline. Since the images are saved as PNG, which uses sRGB color spaces (based on REC. 709), solely the colors in REC. 709 gamut are representable. That is, some sort of gamut mapping has to be applied before displaying it.

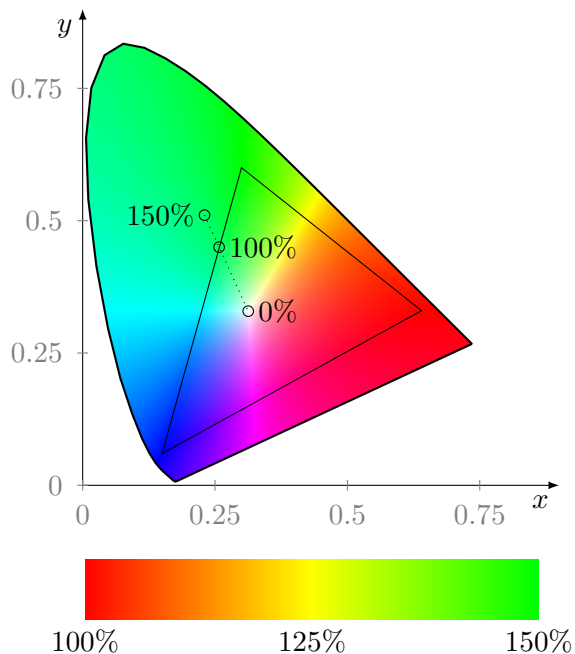


Figure 5.1.: Scala

The second graphics type highlights pixel with chromaticities outside the REC. 709 gamut. In the background there is a gray scale version of the test image for orientation. Pixels with chromaticities outside the REC. 709 triangle are colored: red means this chromaticity is little outside whereas green pixels show chromaticity saturated more than 1.5 times as

much as REC. 709 gamut contains.

Chromaticity plots show the distribution of chromaticities of the image as black dots in xyY color space (Y is omitted). Additionally, XYZ gamut (horseshoe) as well as REC. 709 and camera gamut are depicted. This gives a clue how the chromaticities are mapped.

The vector plots exhibit the movement of chromaticities. That is, the arrow starts at the coordinates of the original chromaticity obtained by applying only CIE XYZ color matching functions and ends at the coordinates of the chromaticity¹ obtained by applying a camera's color matching function. Note that for sake of clarity, the vector plots do not use the chromaticities from the test images but IT 8.7 and monochromatic samples.

The test images are from [YMIN08]. I choose particular the CD image because of its highly saturated gradients. Especially the purple-bluish gradient in a 10 o'clock direction of the CD is vulnerable for banding effects. In some situations, other test images are suitable because of they provide different chromaticities.

Because transformation based on IT 8.7 samples gives virtually the same gamut as transformation based on IT 8.7 together with monochromatic samples, I will limit the comparison to IT 8.7 and monochromatic transformation solely.

Conveniently, a color space transformation based on monochromatic samples is abbreviated as *monochromatic transformation* and a color space transformation based on IT 8.7 samples transformation is abbreviated as *IT 8.7 transformation*.

The tables with statistic data provide both mean and variance of the error. The error is the norm of the difference of reference value provided by applying pure CIE XYZ color matching functions with the result of the actual pipeline. Furthermore, the samples are divided into two groups, the ones with reference values inside REC. 709 gamut and the others with reference value outside.

5.1. CIE XYZ color matching functions

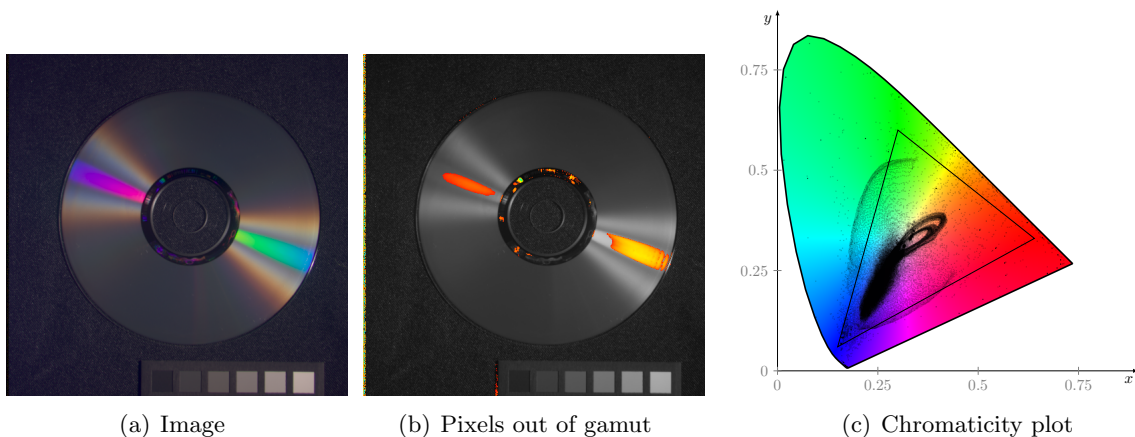


Figure 5.2.: CIE color matching functions with clamping in RGB color space

Figure 5.2 was made with CIE XYZ color matching functions and simple clamping RGB values. At first sight, the image looks fine. But a closer look at the purple-bluish gradient of the CD reveals banding effects. This is a consequence of poor gamut mapping. All pixels inside this artifact are out of gamut (see figure 5.2(b)) and hence at least one component is

¹in CIE xyY color space

negative originally and thus clamped to zero to display the image. So what was a smooth gradient former is a sharp kink now. The green-bluish gradient on the other side of the center of the cd shows similar artifacts.

Note that the artifacts visibility is highly dependent of the displaying device. For example, they are hardly recognizable when viewing perpendicular at a poor notebook display but clearly visible when looking from little above on the same display. On a TV display, the purple artifact looks like a coffee stain with clearly visible edges, depending on the display settings. Of course, banding effects should never be visible, regardless of the display settings or its type. Figure 5.3 shows how editing the curves of an image can increase visibility of artifacts.

In the chromaticity diagram, chromaticities out of REC. 709 gamut lie outside the triangle. That is in particular midway parts of the two “arms”, one reaching from blue over cyan and green to yellowish green and one from blue via purple to reddish purple. The chromaticities on those arms are located at the rainbow effect of the CD image. The secondary and less intensive rainbow effect around the primary effect belongs to the loops right above the white point.

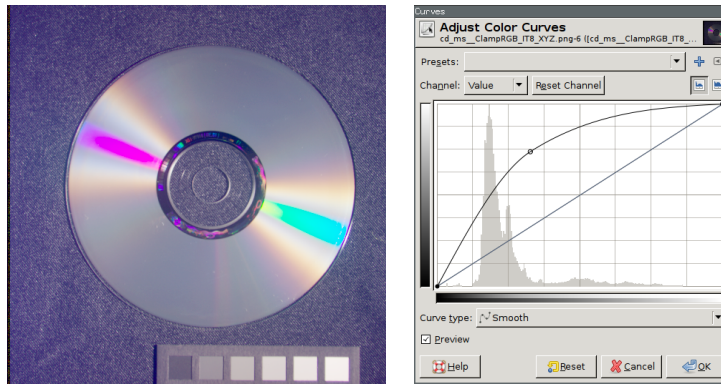


Figure 5.3.: Editing brightness curves highlights the artifacts

Clamping just saturation by constant hue gives virtually the same result with the same artifacts as clamping in RGB color space. Again, editing the image can increase the artifacts visibility enormously.

In contrast, linear and cubic gamut mapping algorithms do not produce artifacts. This is clear since those algorithms avoid sharp kinks in the saturation curve. But the image looks flat. Especially in the green regions, linear gamut mapping produces weak saturated results (see figure 5.4).

	all		within REC. 709		outside REC. 709	
	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2
clamp	0.0997797	0.0978878	0.0	0.00232024	0.269859	0.206678
linear	0.369373	0.14853	0.34177	0.0963378	0.416424	0.228106
cubic	0.180299	0.128024	0.101637	0.0676439	0.314383	0.20092

Table 5.1.: Statistical quantities

The statistics (table 5.1) exposes, that clamping the saturation keeps the error smaller than linear or cubic mapping. Of course, clamping the saturation has no impact on coordinates already in target color space.

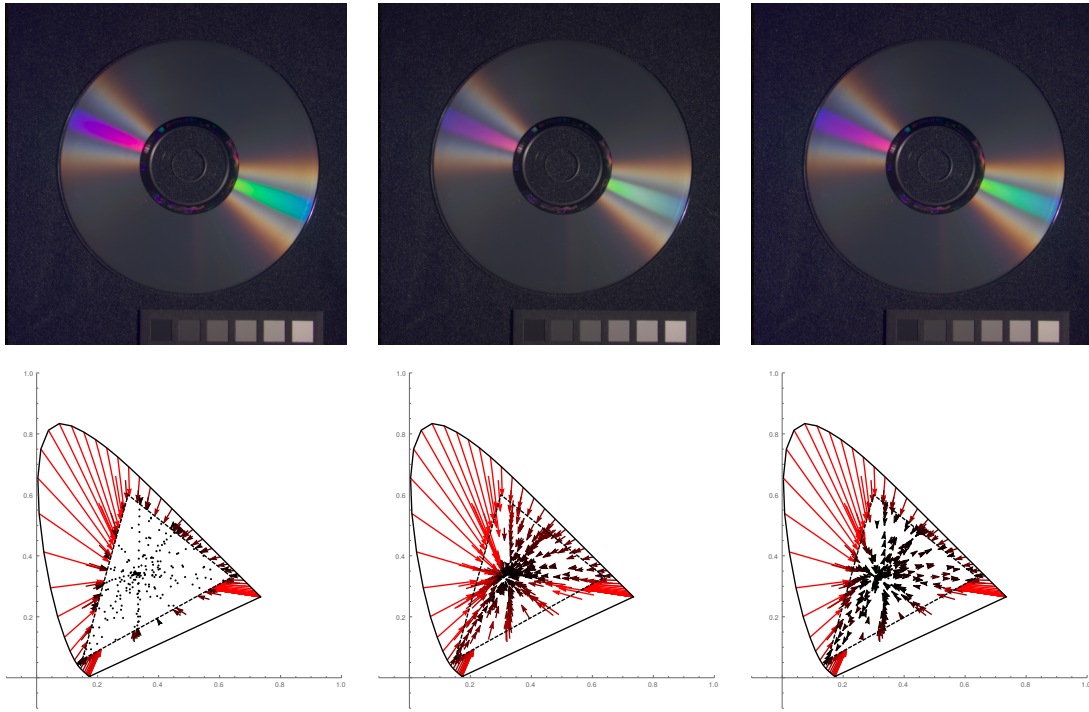


Figure 5.4.: Applying different gamut mapping algorithms and CIE XYZ color matching functions. From left to right: clamp, linear and cubic saturation function.

5.2. SONY DXC 930

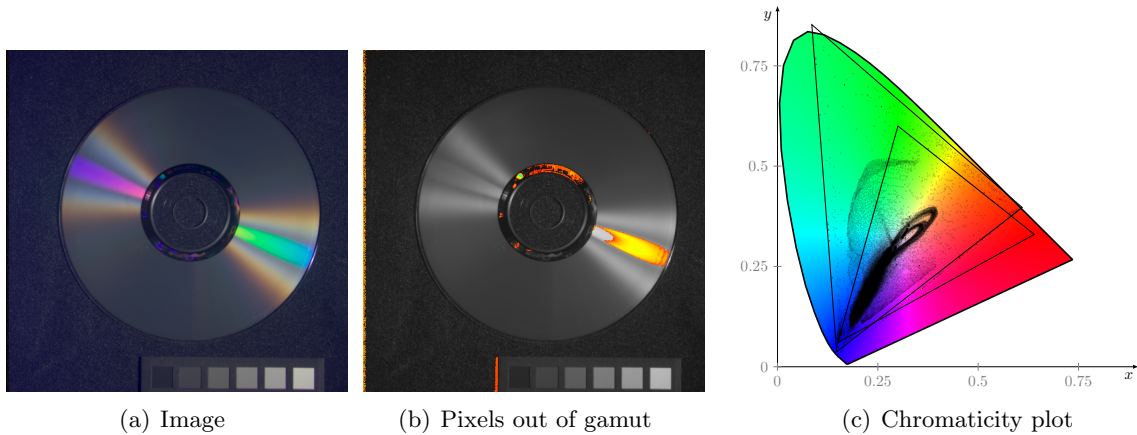


Figure 5.5.: SONY DXC 930 camera, matrix generated with monochromatic samples, clamping in RGB color space

The gamut of the SONY DXC 930 is similar to the CIE XYZ gamut in green regions. Because of the transformation from camera color space to CIE XYZ color space is not exactly linear, the linear approximation varies dependently from which samples were used to determine the matrix.

Using the monochromatic samples, the camera gamut lacks a large region of red in REC. 709 gamut. This is reflected by the fact, that images made with this camera color matching functions and monochromatic transformation have poor red tones, see figure 5.7(a).

Since virtually all purple-bluish chromaticities produced by this camera lie inside the

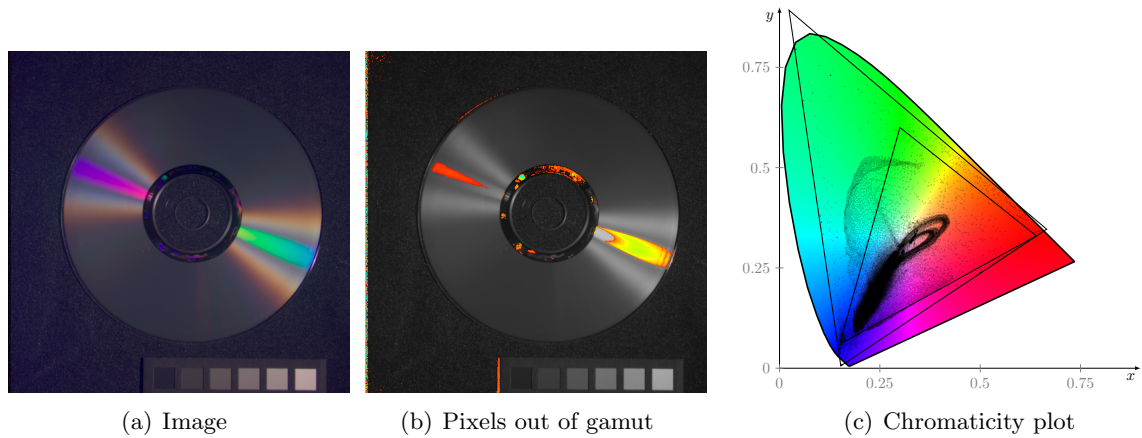


Figure 5.6.: SONY DXC 930 camera, matrix generated with IT8.7 samples, clamping in RGB color space

REC. 709 gamut, there are no artifacts in the purple-bluish gradient of the CD. Still the yellow-green artifacts exist because the camera produces green tones that cannot be represented by REC. 709 devices. This is clearly visible in figure 5.6(b).

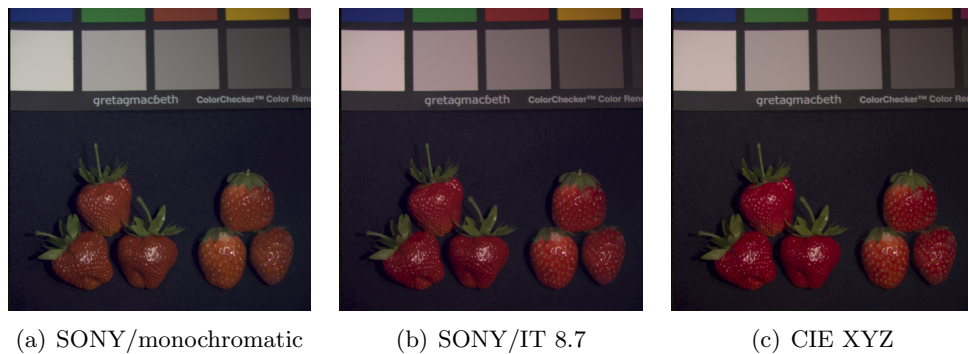


Figure 5.7.: Images produced with SONY DXC 930 and CIE XYZ color matching functions

The IT 8.7 transformation gives a wider gamut than monochromatic transformation. With IT 8.7 transformation, the camera gamut embraces the REC. 709 gamut completely. Thus, the colors of the strawberries in 5.7(b) are much more intensive than in figure 5.7(a). Apart of the good result in red regions, green respectively purple regions of the CD image are worse respectively slightly better than with the original pipeline with CIE XYZ color matching functions. This is, because the gamut is greater for green chromaticities and embraces the REC. 709 gamut closer than CIE XYZ gamut in purple-bluish regions.

The vector plot exhibits the shifting of the chromaticities, that is the error of applying camera color matching functions and transformation compared to applying CIE XYZ color matching functions in xy-plane. In case of monochromatic transformation, especially the red chromaticities are erroneous, whereas the IT 8.7 transformation distorts mostly green chromaticities.

Note that the results of both transforms are further afar from target gamut than with CIE XYZ color matching functions.

Applying saturation clamping, linear or cubic gamut mapping produces similar results than in 5.1. Especially the linear algorithm produces quite low saturated chromaticities

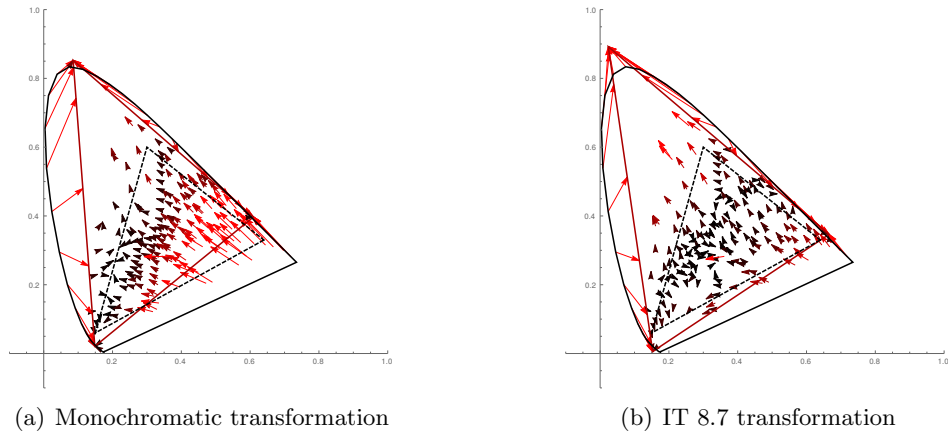


Figure 5.8.: SONY DXC 930 camera color matching functions versus CIE XYZ color matching functions, without additional gamut mapping.

for green. Obviously because the gamut is even wider than the CIE XYZ gamut. In this case, the cubic and even the linear algorithm still produce artifacts in the yellow-green gradient of the CD image (see figure 5.9).

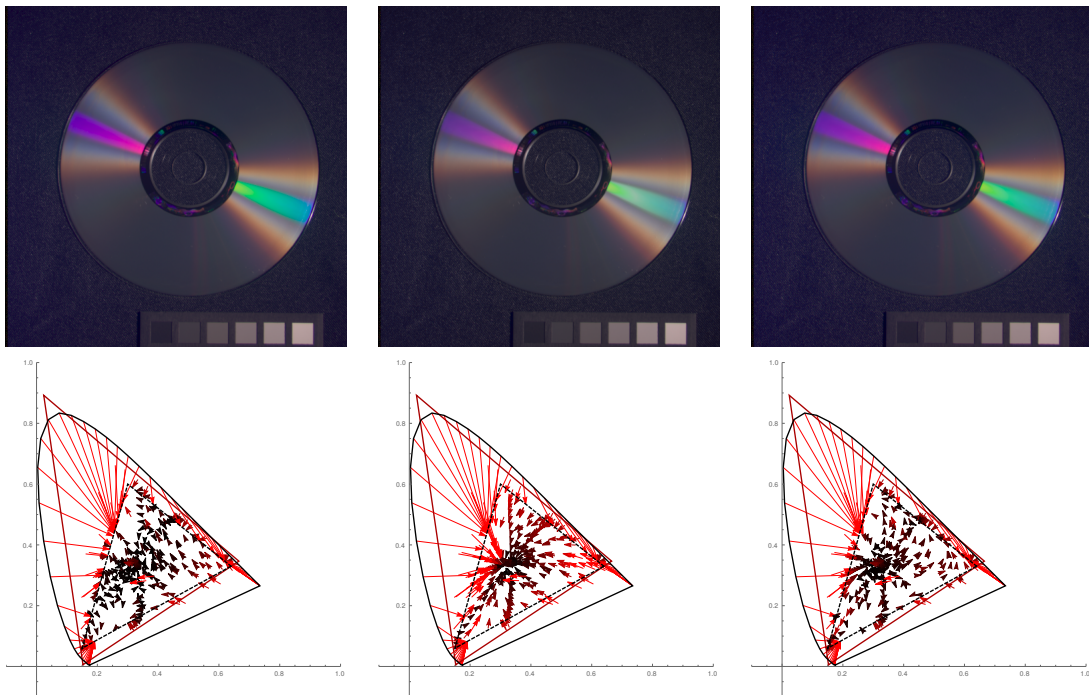


Figure 5.9.: Applying different gamut mapping algorithms. SONY DXC 930 and IT 8.7 transformation. From left to right: clamp, linear and cubic saturation function.

Using the IT 8.7 transformation, the mean error does not increase much, compared to CIE XYZ color matching functions. The mean error of the part that lies outside REC. 709 gamut even decreases significantly. But with the monochromatic transformation, the mean error raises for both groups, within and outside REC. 709 This confirms the empirical impression from the vector plots.

	all		within REC. 709		outside REC. 709	
	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2
IT 8.7 clamp	0.163624	0.0938117	0.0432079	0.000453318	0.393231	0.191458
IT 8.7 linear	0.338689	0.147589	0.256964	0.0891456	0.49452	0.222008
IT 8.7 cubic	0.240748	0.12437	0.149618	0.0653418	0.414512	0.190894
mono. clamp	0.300792	0.112627	0.273085	0.0532969	0.354075	0.192991
mono. linear	0.397256	0.151019	0.378646	0.103952	0.433046	0.225787
mono. cubic	0.361565	0.138967	0.349778	0.105406	0.384232	0.191811

Table 5.2.: Statistical quantities

5.3. KODAK DCS 200

The gamut of the KODAK DCS 200 differs significantly from the CIE XYZ and SONY DCX 930 gamut. Whereas it covers more blue and red, the camera cannot reproduce strong green tones, just like REC. 709. That is, the chromaticities are mostly inside REC. 709 gamut.

With both the monochromatic and IT 8.7 transformation, the chromaticities of the green-bluish gradient is inside the REC. gamut. The chromaticities of the purple-bluish gradient are mostly outside REC. gamut in case of monochromatic transformation or inside in case of IT 8.7 transformation. This reflects in the respective gamuts, see figure 5.10(c) and 5.11(c).

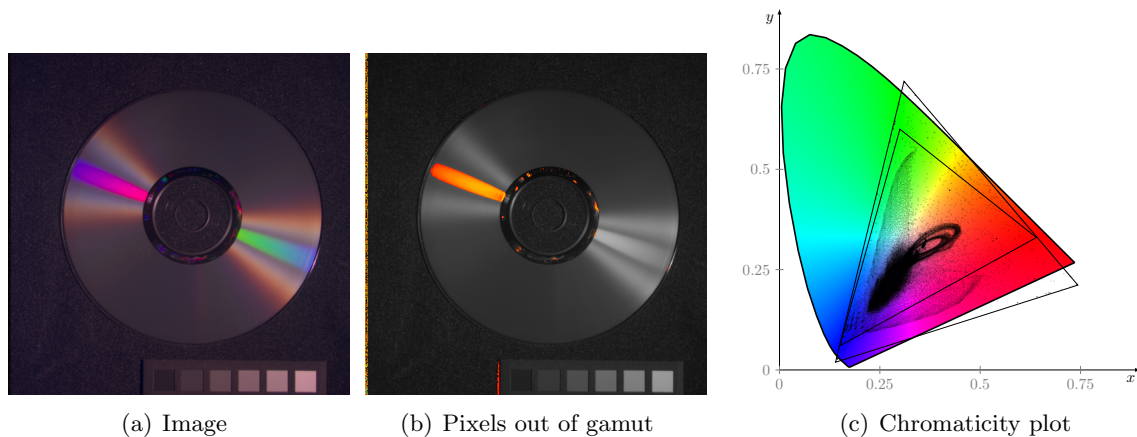


Figure 5.10.: KODAK DCS 200 camera, matrix generated with monochromatic samples, clamping in RGB color space

The KODAK DCS 200 gamut is very closely to the REC. 709 gamut, especially for green chromaticities. The red reaches to the edge of the CIE XYZ gamut in case of IT 8.7 transformation, respectively beyond in case of monochromatic transformation. Both transformation produce gamuts that intersect the CIE XYZ gamut minimal.

The vector plot 5.12(a) reveals big errors of many chromaticities that are shifted towards red. For example see the white of the paint box in figure 5.13(b), which is reddish (cf. figure 5.13(c)).

Unlike before, both linear and cubic gamut mapping produce virtually equal results. Especially green regions are more saturated. This is because the source gamut is much closer to the target gamut, compared with e.g. the CIE XYZ gamut and therefore, δ_{SRC} is much closer to δ_{DST} (see figure 5.14 and figure 5.15).

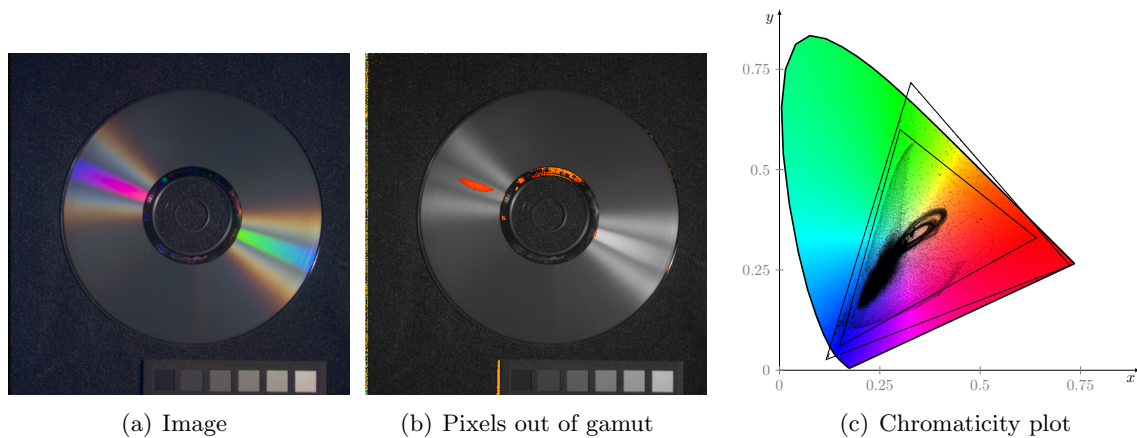


Figure 5.11.: KODAK DCS 200 camera, matrix generated with IT8.7 samples, clamping in RGB color space

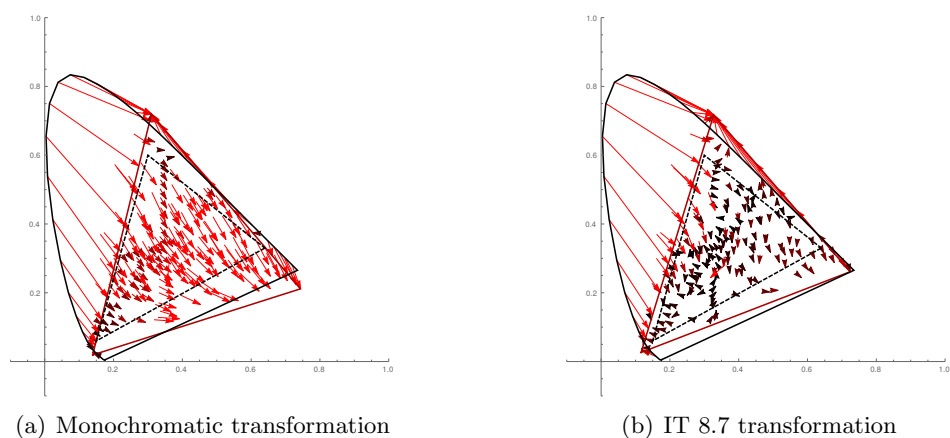


Figure 5.12.: KODAK DCS 200 camera color matching functions versus CIE XYZ color matching functions, without additional gamut mapping.

Just like before with the SONY DXC 930, the mean error does not increase much, compared to CIE XYZ color matching functions. Whereas mean error is bigger than with SONY DXC 930 within sRGB, it is as significantly smaller outside. The mean error outside and inside is comparable with the mean error from CIE XYZ color matching functions, for linear saturation mapping even better.

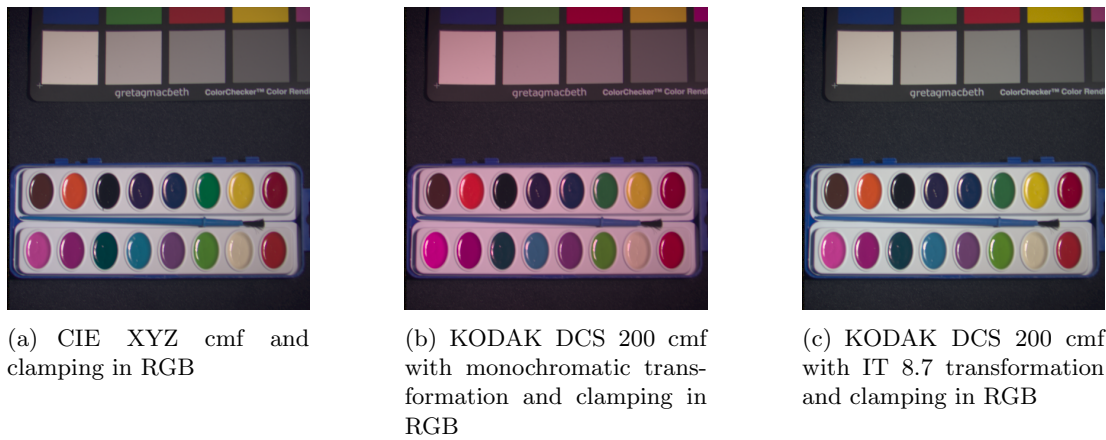


Figure 5.13.: White shift depends on transformation

	all		within REC. 709		outside REC. 709	
	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2
IT 8.7 clamp	0.168187	0.0938325	0.09031	0.00267193	0.317951	0.197125
IT 8.7 linear	0.345003	0.147628	0.317622	0.0928251	0.39766	0.231389
IT 8.7 cubic	0.200016	0.126029	0.130928	0.0656911	0.332877	0.197558
mono. clamp	0.507294	0.211921	0.577093	0.285487	0.374202	0.19182
mono. linear	0.476957	0.166707	0.509655	0.152998	0.414609	0.228393
mono. cubic	0.518335	0.201424	0.583258	0.253386	0.394539	0.191293

Table 5.3.: Statistical quantities

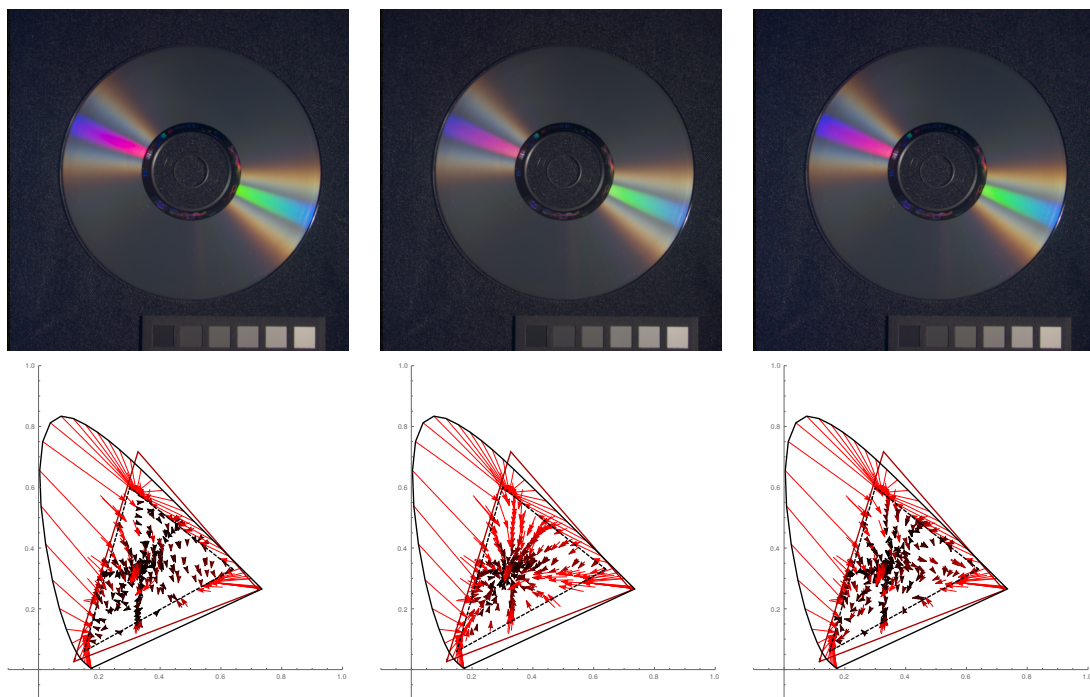


Figure 5.14.: Applying different gamut mapping algorithms. KODAK DCS 200 and IT 8.7 transformation. From left to right: clamp, linear and cubic saturation function.

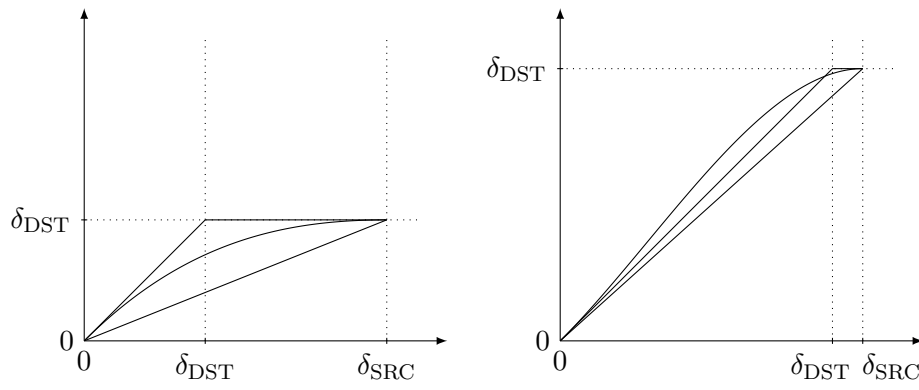


Figure 5.15.: When $\delta_{DST} \rightarrow \delta_{SRC}$, the differences between clamped, linear and cubic functions shrink.

5.4. Comparison

In case of the camera gamut intersects the REC. 709 gamut, one must consider that the gamut mapping is not longer solely gamut shrinking (cf. 3). Therefore special algorithms are required that regard this circumstances, e.g. linear and cubic gamut mapping produce chromaticities outside the target gamut and are not appropriate in this case. This extends for non convex gamuts.

Because linear and cubic gamut mapping rely on convexity of the gamut boundary, they do not work for e.g. KODAK DCS 460 with both monochromatic and IT 8.7 transformation (see figure 5.16 and appendix). Table 5.4 exhibits bigger mean error for virtual all combinations. Especially the badly fitted monochromatic transformation and error susceptible cubic saturation mapping produce large errors. Note the significantly higher variance.

	all		within REC. 709		outside REC. 709	
	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2	\overline{xy}	σ_{xy}^2
IT 8.7 clamp	0.193895	0.094728	0.138944	0.00961872	0.306295	0.199016
IT 8.7 linear	0.436077	0.157073	0.409993	0.112564	0.491445	0.222017
IT 8.7 cubic	1.19801	1.04072	0.469549	0.167698	2.774	5.75807
mono. clamp	0.685278	0.365934	0.807349	0.584364	0.435587	0.193252
mono. linear	0.96886	0.544704	0.896103	0.497645	1.11768	0.610336
mono. cubic	2.36278	4.6274	1.07183	0.915818	5.18282	22.9277

Table 5.4.: Statistical quantities for KODAK DCS 460

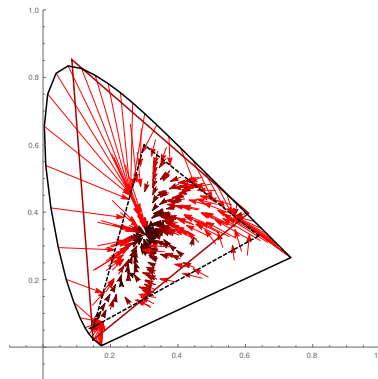


Figure 5.16.: SONY DXC 930 with monochromatic transformation and linear gamut mapping

In contrast, a wider camera gamuts than CIE XYZ gamut do not cause as much problems. But since the camera gamut should be closer to the target gamut than CIE XYZ gamut, the regions out of CIE XYZ gamut should be as small as possible.

If the camera gamut fits closely to the REC. 709 gamut, simple linear and cubic mapping produce very good results.

In general, the mean error and variances do not correlate with the quality of the camera gamut: the mean errors of SONY DXC 930 and KODAK DCS 200 do not differ significantly, neither their variances (in case of IT 8.7 transform). But their gamuts differ and KODAK DCS 200 is much more closely to target gamut.

6. Conclusion

Technical cameras provide a different set of colors (“gamut”) than CIE XYZ color matching functions, because their absorption spectra differ from CIE XYZ color matching functions. To make the camera results usable, a transformation from unknown camera color space to a well known color space is required. As it turned out, a linear transformation to CIE XYZ, obtained by solving a linear least squares fitting problem, gives good results for several cameras and bad results for others. The choice of the fitting samples has a big impact on the quality of the gamut too. Cameras that have well separated color matching functions provide color space that is well transformable to CIE XYZ color space with a linear transformation.

Beside this, one should care about the actual gamut of the camera. Ideally, it is convex and encloses the target gamut as close as possible, but does not intersect it. Then, simple gamut mapping algorithms such as linear saturation mapping obtain very good results. In case of sRGB target gamut, the KODAK DCS 200 camera curves in combination with a transformation based on evenly distributed samples and linear saturation mapping gives best result.

But generally, optimizing the camera color space transformation in a way, the camera gamut matches the target gamut the best, must not give the best results, since the transformation may distort the hue of colors or transform saturation in an unsuitable way.

Although the linear transformation models suffice for several cameras, non-linear transformations or non-linear fitted transformations may enlarge the variety of cameras. Finding such transformations would an interesting direction for further work, as further examining the coherence between color matching functions and the resulting gamut.

Bibliography

- [Ash02] M. Ashikhmin, “A tone mapping algorithm for high contrast images,” in *Proceedings of the 13th Eurographics workshop on Rendering*. Eurographics Association, 2002, pp. 145–156.
- [BB09] W. Burger and M. J. Burge, *Principles of Digital Image Processing*. Springer, 2009.
- [BH97] A. Byrne and D. R. Hilbert, *Readings on color: The science of color*. MIT Press, 1997, vol. 2.
- [DMAC03] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive logarithmic mapping for displaying high contrast scenes,” in *Computer Graphics Forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 419–426.
- [Fau14] W. Faust. (2014) It 8.7 scanner calibration targets. [Online]. Available: <http://www.targets.coloraid.de/>
- [FBH97] H. S. Fairman, M. H. Brill, and H. Hemmendinger, “How the cie 1931 color-matching functions were derived from wright-guild data,” *Color Research & Application*, vol. 22, pp. 11–23, 1997.
- [Gla89] A. S. Glassner, “How to derive a spectrum from an rgb triplet,” *Computer Graphics and Applications, IEEE*, vol. 9, no. 4, pp. 95–99, 1989.
- [GM99] C. Gerthsen and D. Meschede, *Gerthsen Physik*, 20th ed. Springer, 1999.
- [Lin11] B. Lindbloom. (2011). [Online]. Available: http://www.brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html
- [Mor08] J. Morovič, *Color Gamut Mapping*, ser. Imaging Science and Technology, 2008.
- [PH10] M. Pharr and G. Humphreys, *Physically Based Rendering*, 2nd ed. Morgan Kaufman, 2010.
- [RKI13] R. T. T. Rei Kawakami, Hongxun Zhao and K. Ikeuchi. (2013) Camera spectral sensitivity and white balance estimation from sky images. [Online]. Available: <http://www.cvl.iis.u-tokyo.ac.jp/~rei/research/cs/index.html>
- [SAM09] P. Shirley, M. Ashikhmin, and S. Marschner, *Fundamentals of computer graphics*. CRC Press, 2009.
- [SB55] W. Stiles and J. Burch, “Interim report to the commission internationale de l’éclairage, zurich, 1955, on the national physical laboratory’s investigation of colour-matching (1955),” *Journal of Modern Optics*, vol. 2, no. 4, pp. 168–181, 1955.
- [SB02] G. Sharma and R. Bala, *Digital color imaging handbook*. CRC press, 2002.
- [Smi99] B. Smits, “An rgb-to-spectrum conversion for reflectances,” *Journal of Graphics Tools*, vol. 4, no. 4, pp. 11–22, 1999.

- [SSF99] A. Stockman, L. T. Sharpe, and C. Fach, “The spectral sensitivity of the human short-wavelength sensitive cones derived from thresholds and color matches,” *Vision research*, vol. 39, no. 17, pp. 2901–2927, 1999.
- [Wat] A. Watt, *3D Computer Graphics*, 3rd ed.
- [WS82] G. Wyszecki and W. S. Stiles, *Color science*. Wiley New York, 1982, vol. 8.
- [YF12] H. D. Young and R. A. Freedman, *Sears and Zemansky’s university physics with Modern Physics*, F. W. B. Sears and M. W. B. Zemansky, Eds. Addison-Wesley, 2012.
- [YMIN08] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, “Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum,” Tech. Rep., Nov 2008.

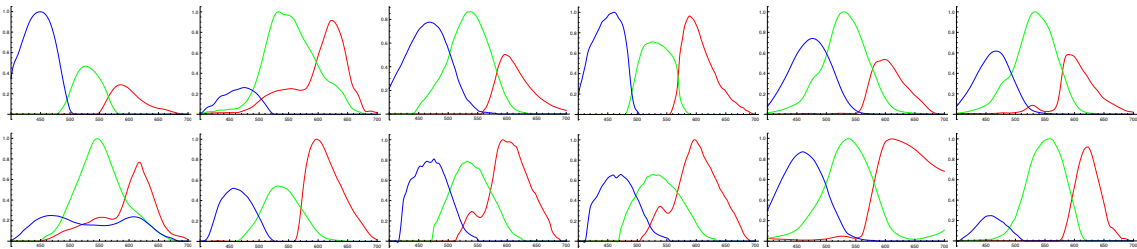
Appendix

A. Cameras

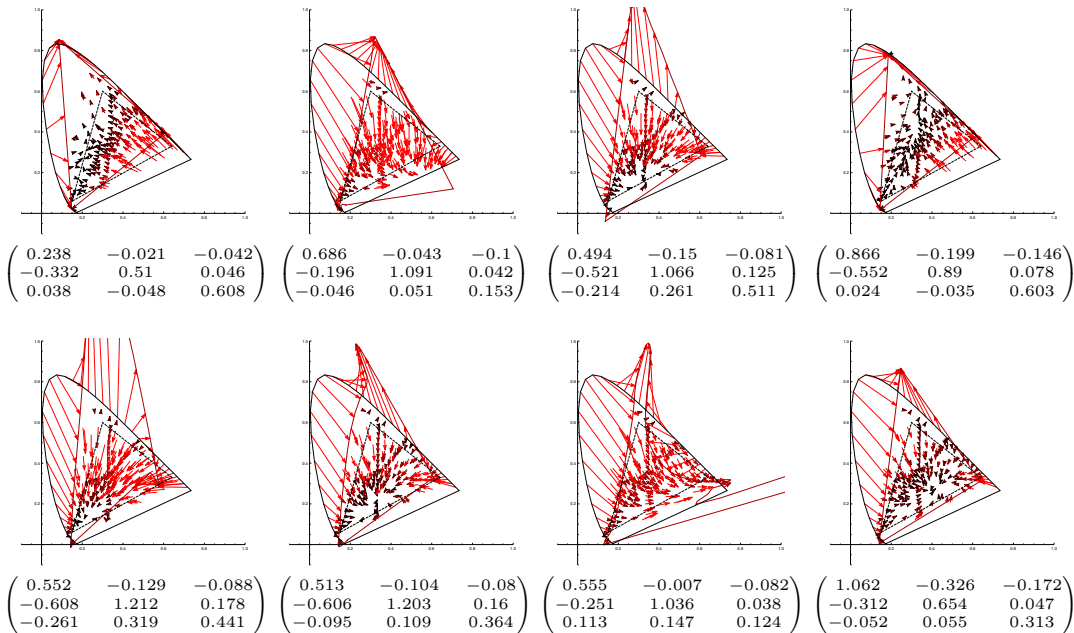
matrix From top left to bottom right:

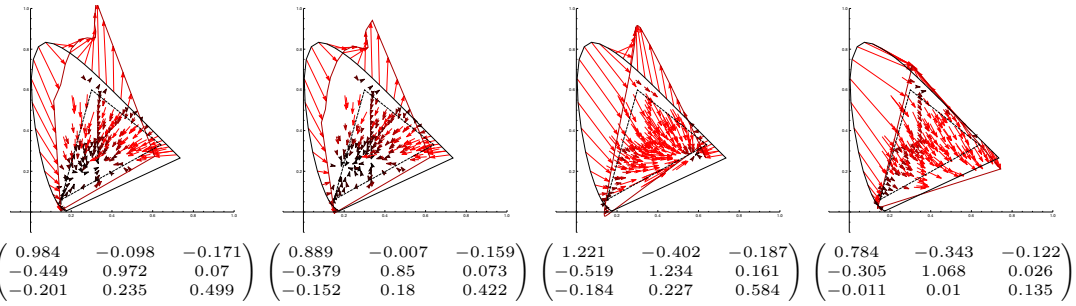
SONY DXC 930, KODAK DCS 420, NIKON D1X, SONY DXC 9000, CANON 10D,
 NIKON D70,
 KODAK DCS 460, CANON 400D, CANON 5D, CANON 5D Mark 2, Ladybug2,
 KODAK DCS 200

Color matching functions

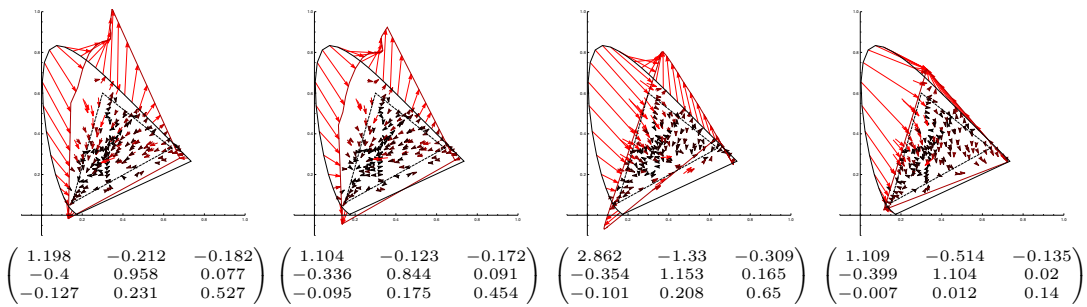
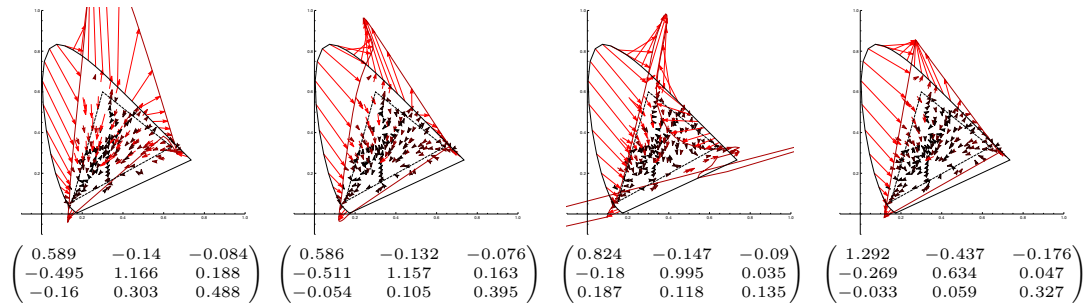
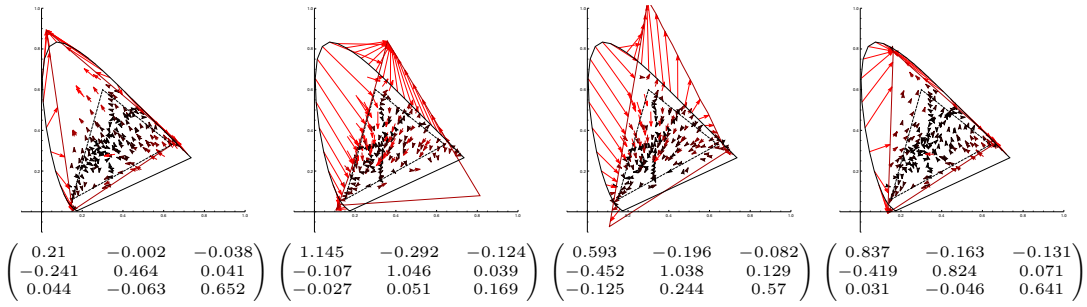


Monochromatic transformation





IT 8.7 transformation

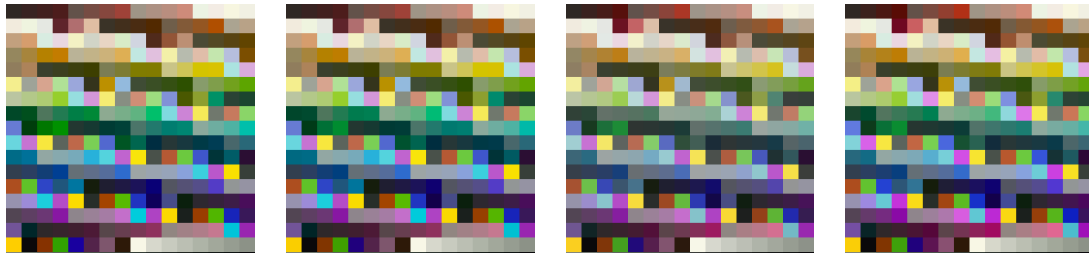


B. Sample Images

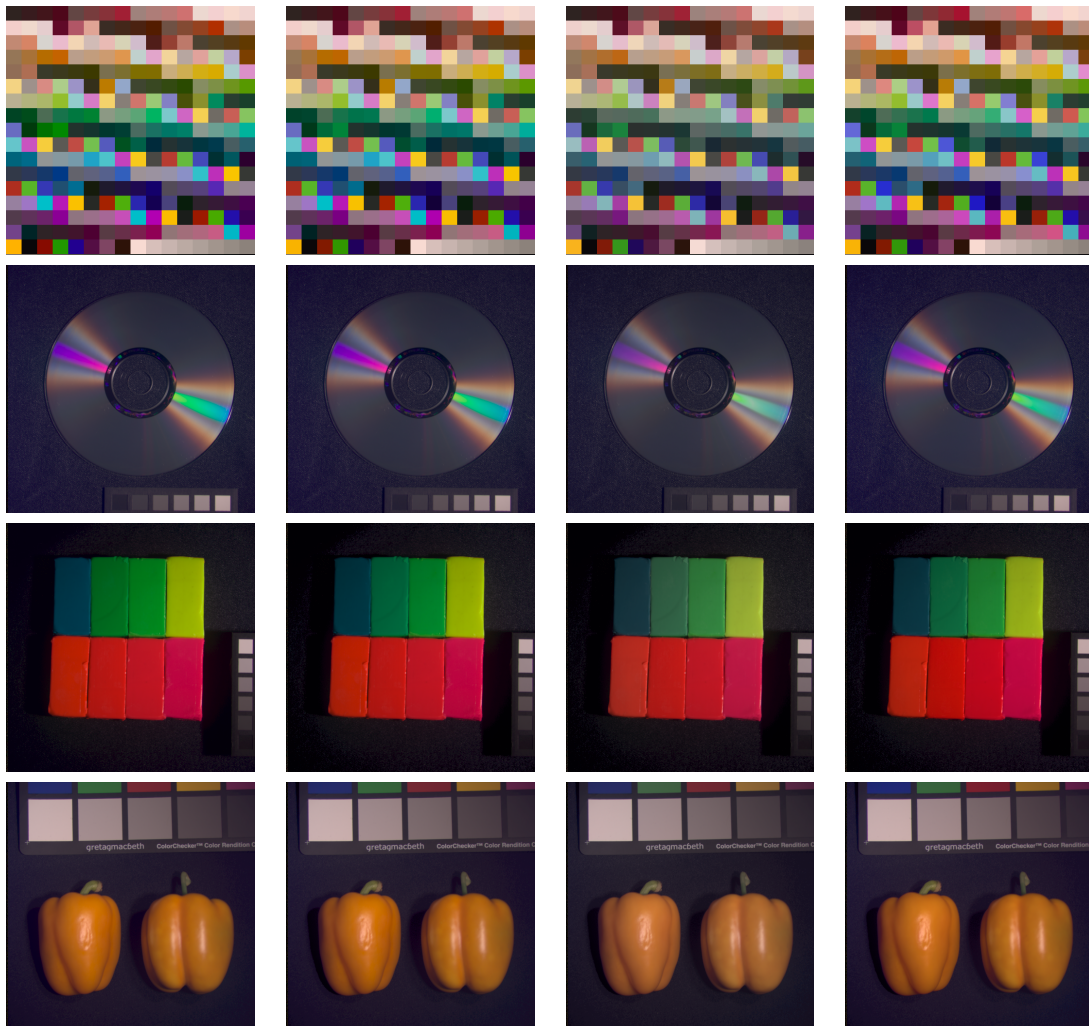
Because monochromatic transformation yields generally worse results than IT 8.7 transformation, only the color checker sample image is provided for both transformations, the other images for IT 8.7 transformation only.

SONY DXC 930

Monochromatic transformation:

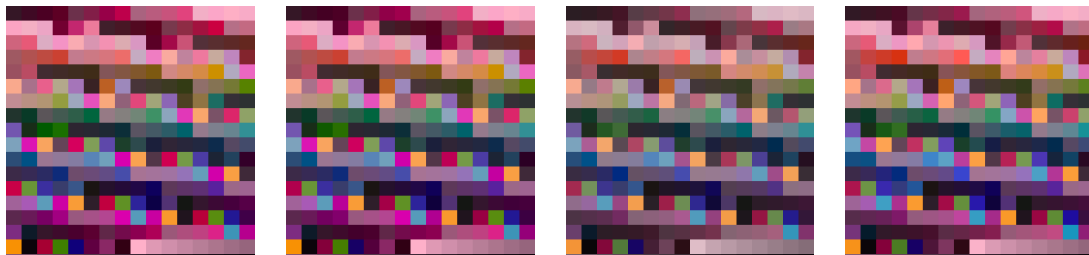


IT 8.7 transformation:

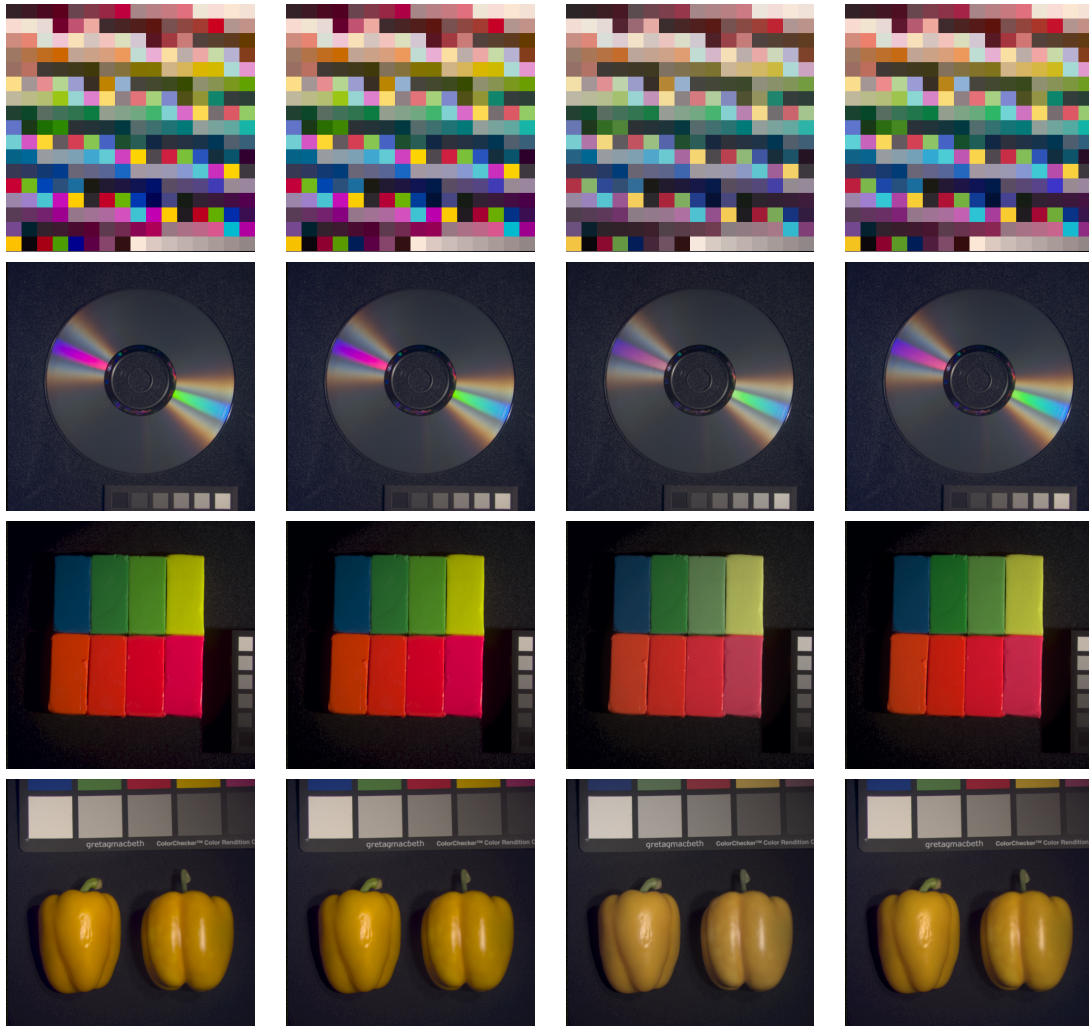


KODAK DCS 420

Monochromatic transformation:

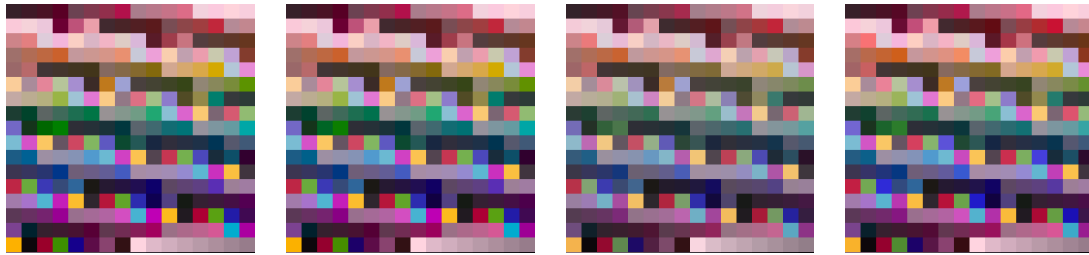


IT 8.7 transformation:

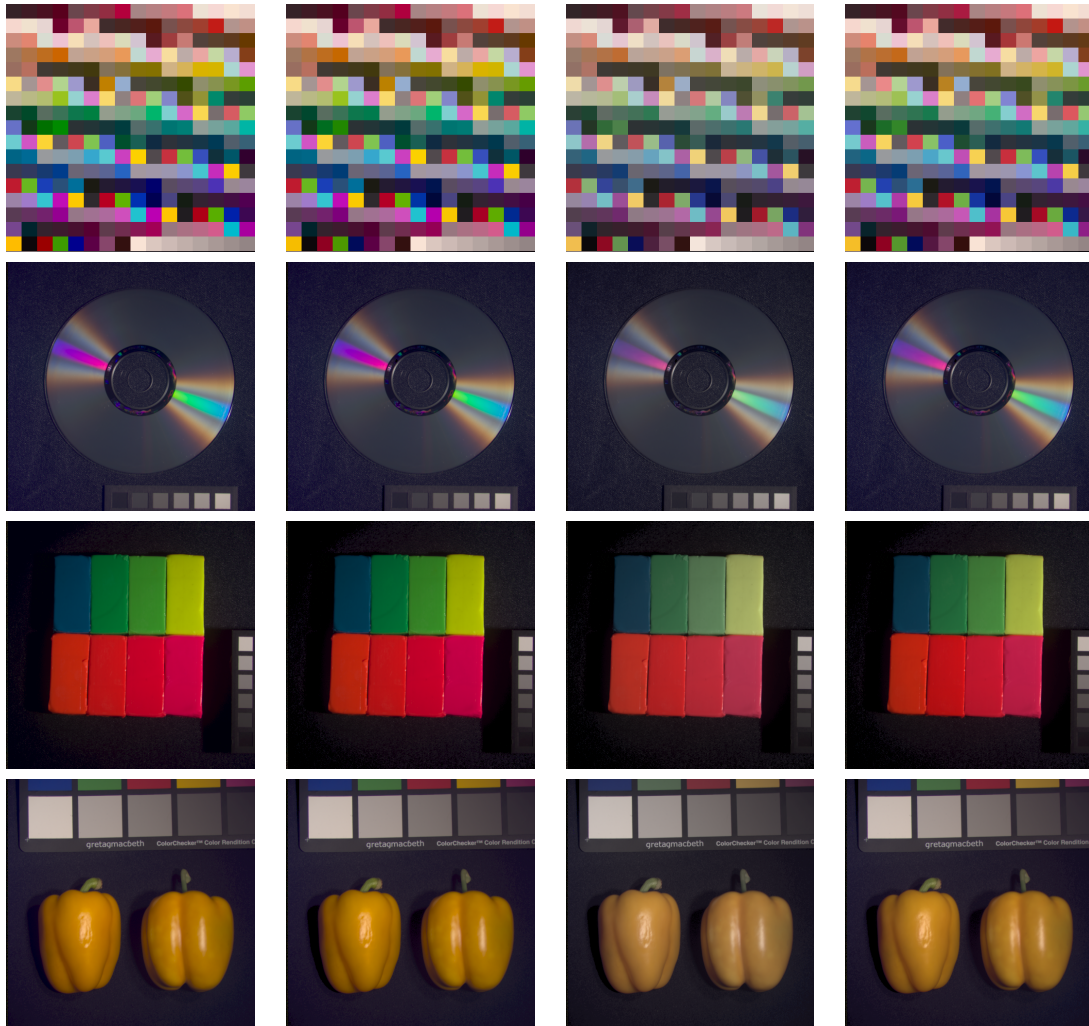


NIKON D1X

Monochromatic transformation:

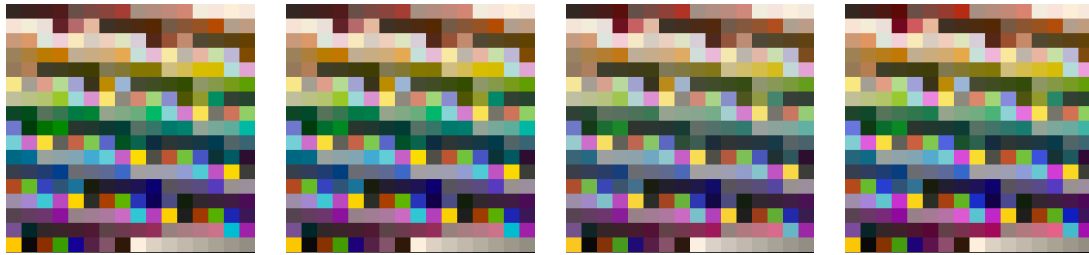


IT 8.7 transformation:

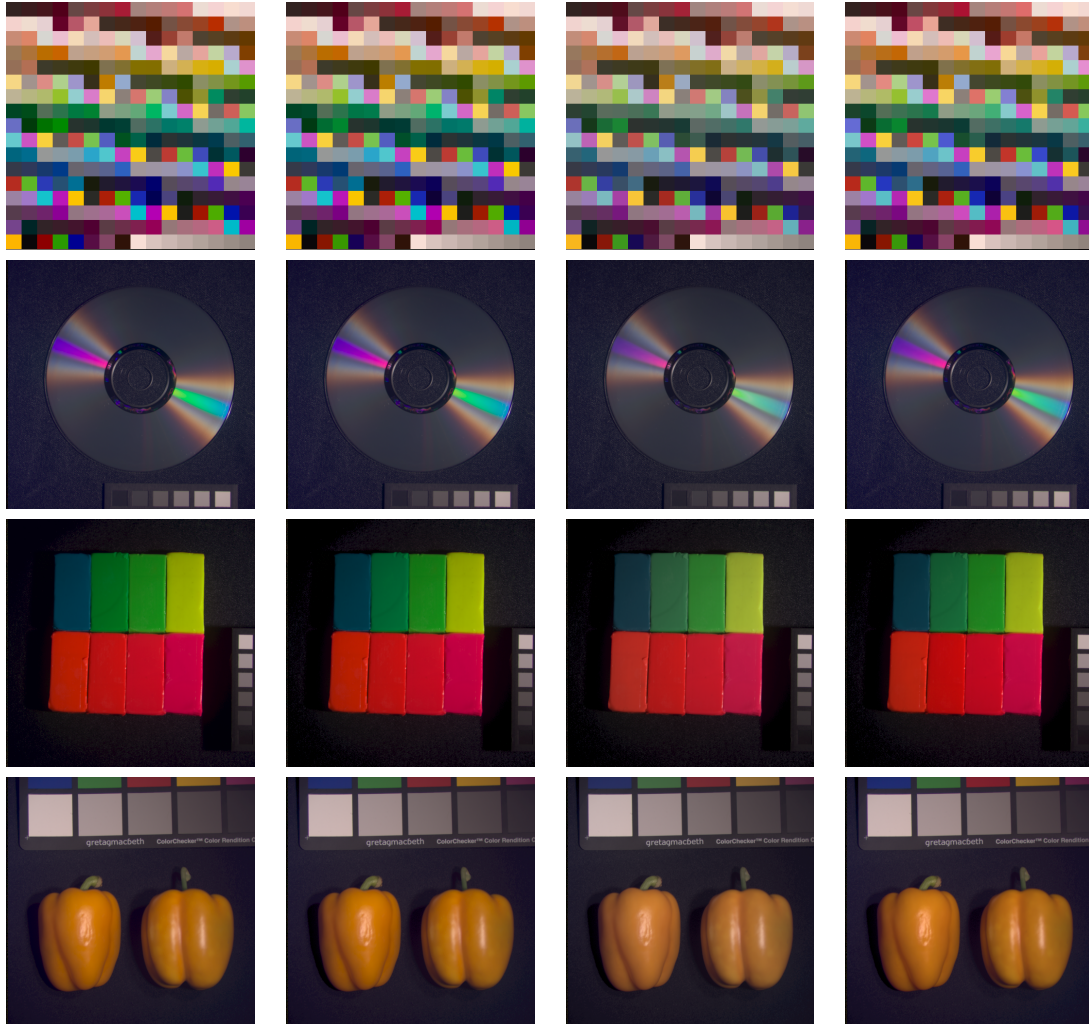


SONY DXC 9000

Monochromatic transformation:

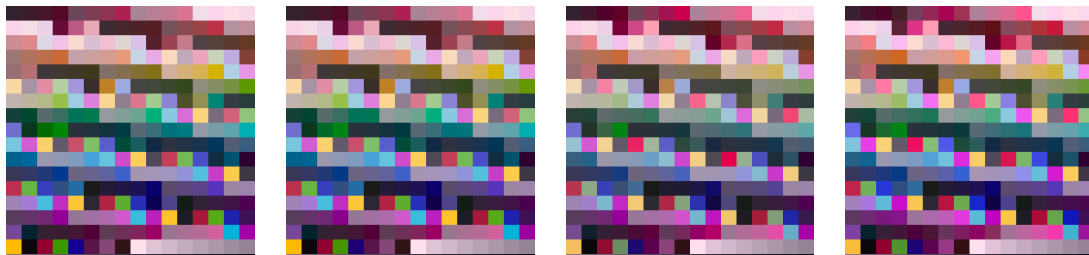


IT 8.7 transformation:

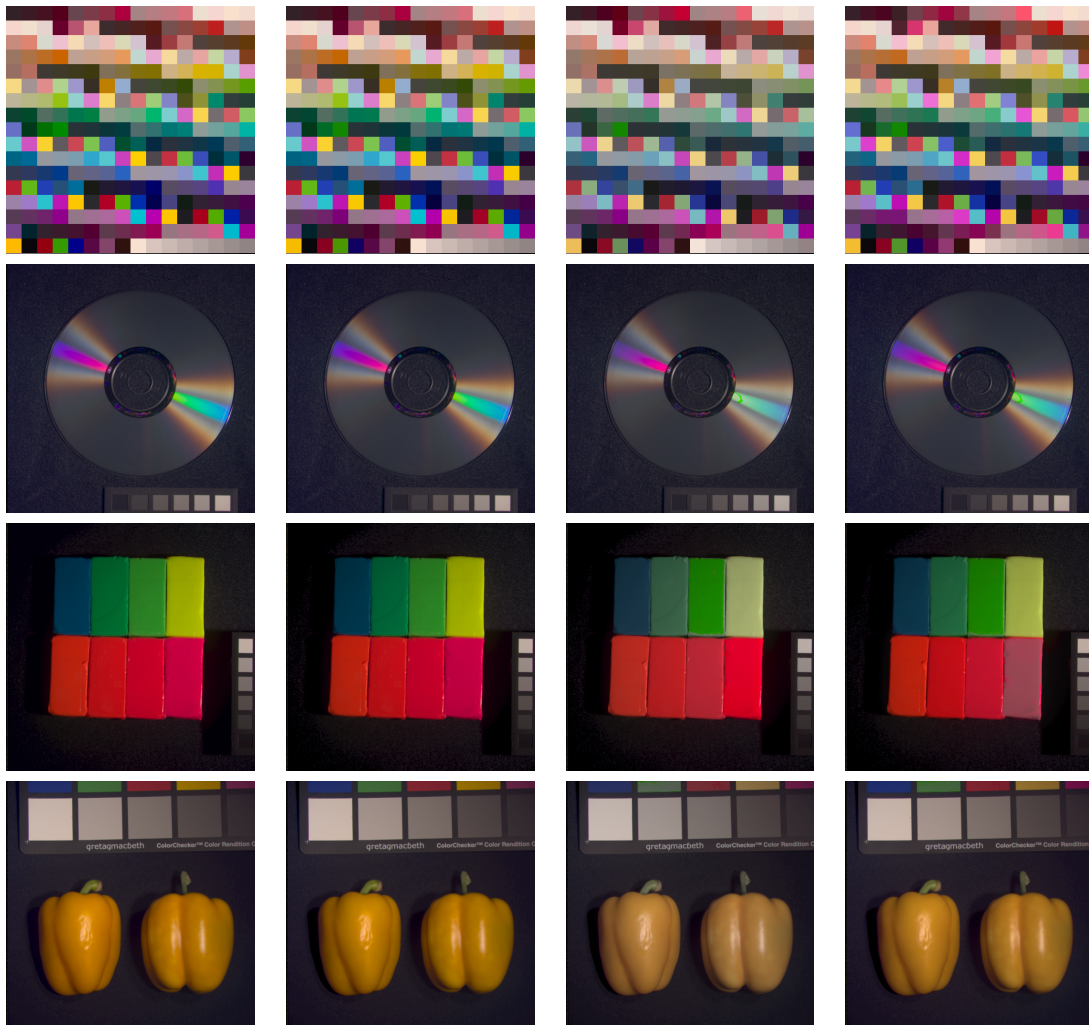


CANON 10D

Monochromatic transformation:

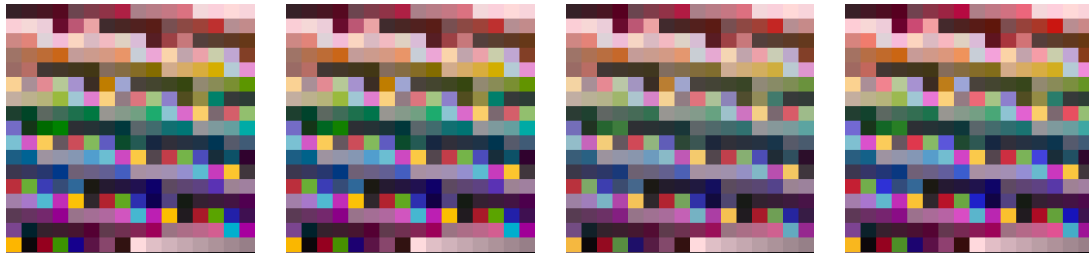


IT 8.7 transformation:

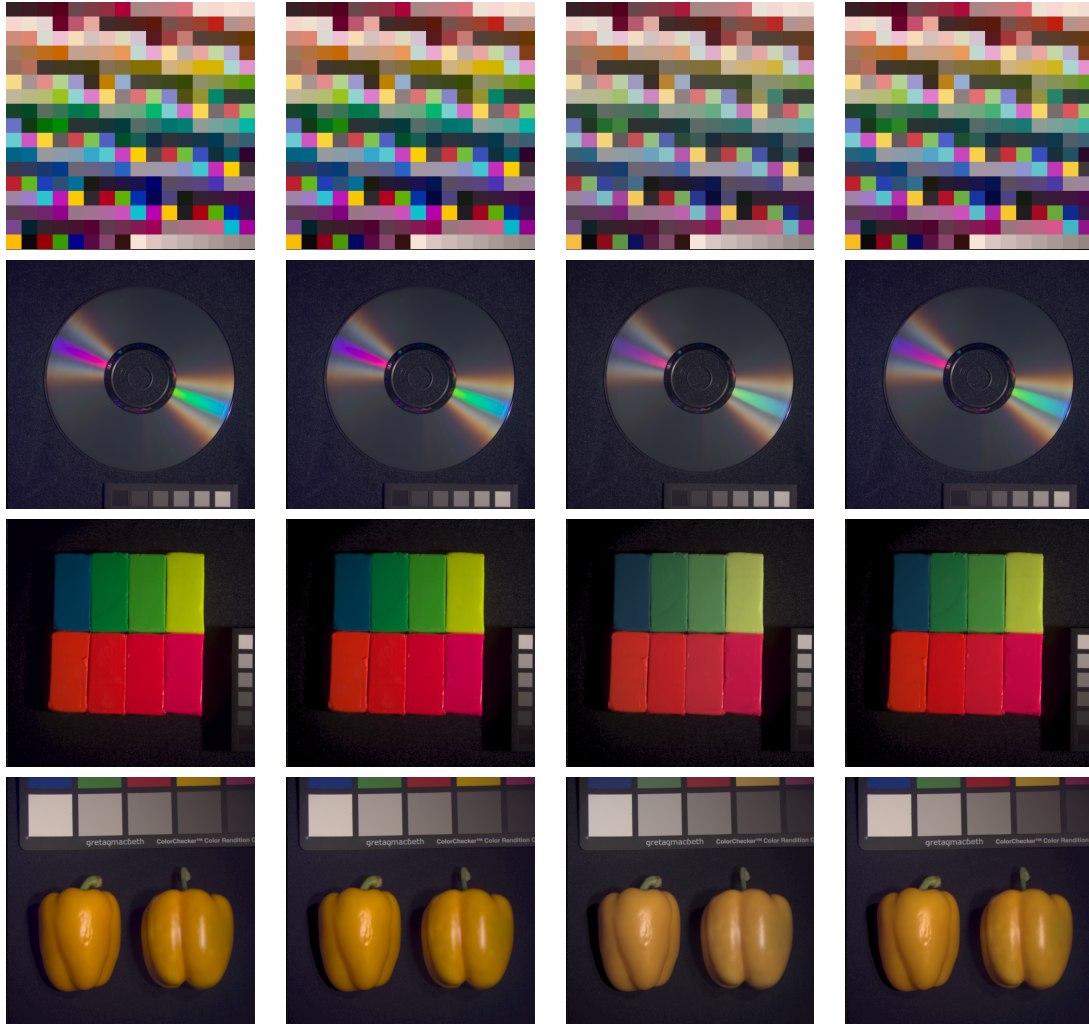


NIKON D70

Monochromatic transformation:

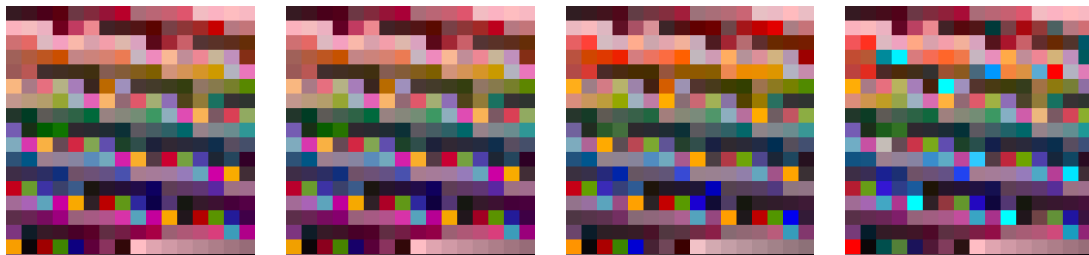


IT 8.7 transformation:

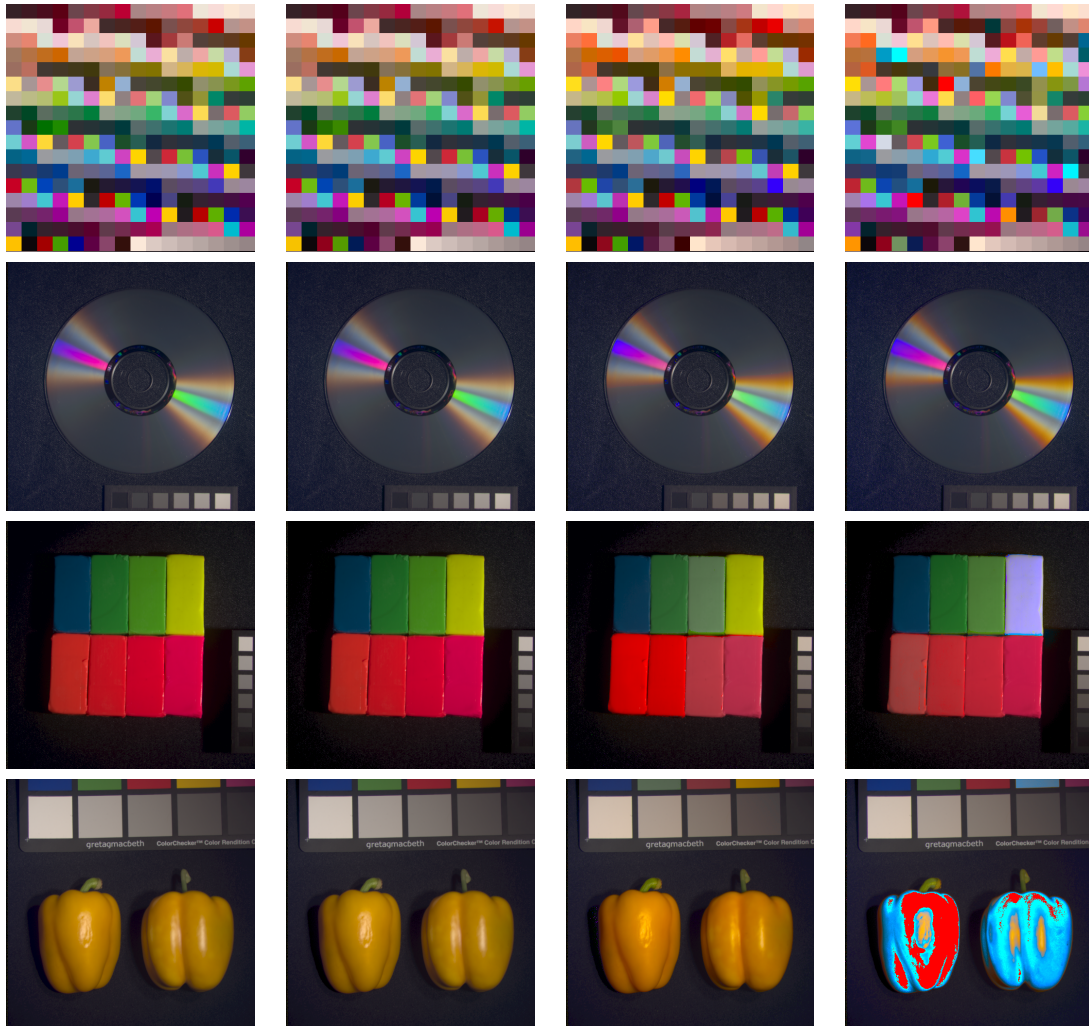


KODAK DCS 460

Monochromatic transformation:

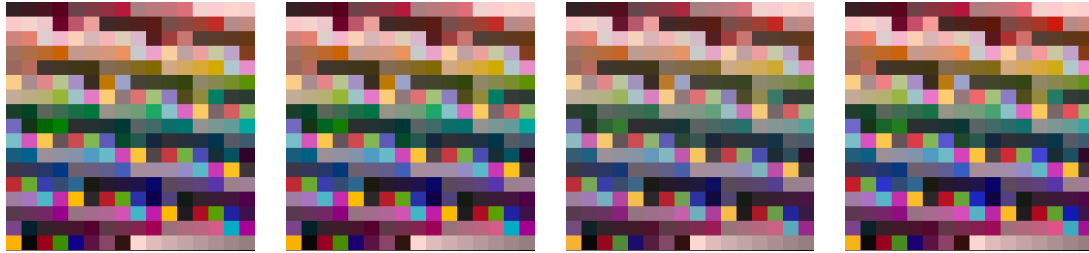


IT 8.7 transformation:

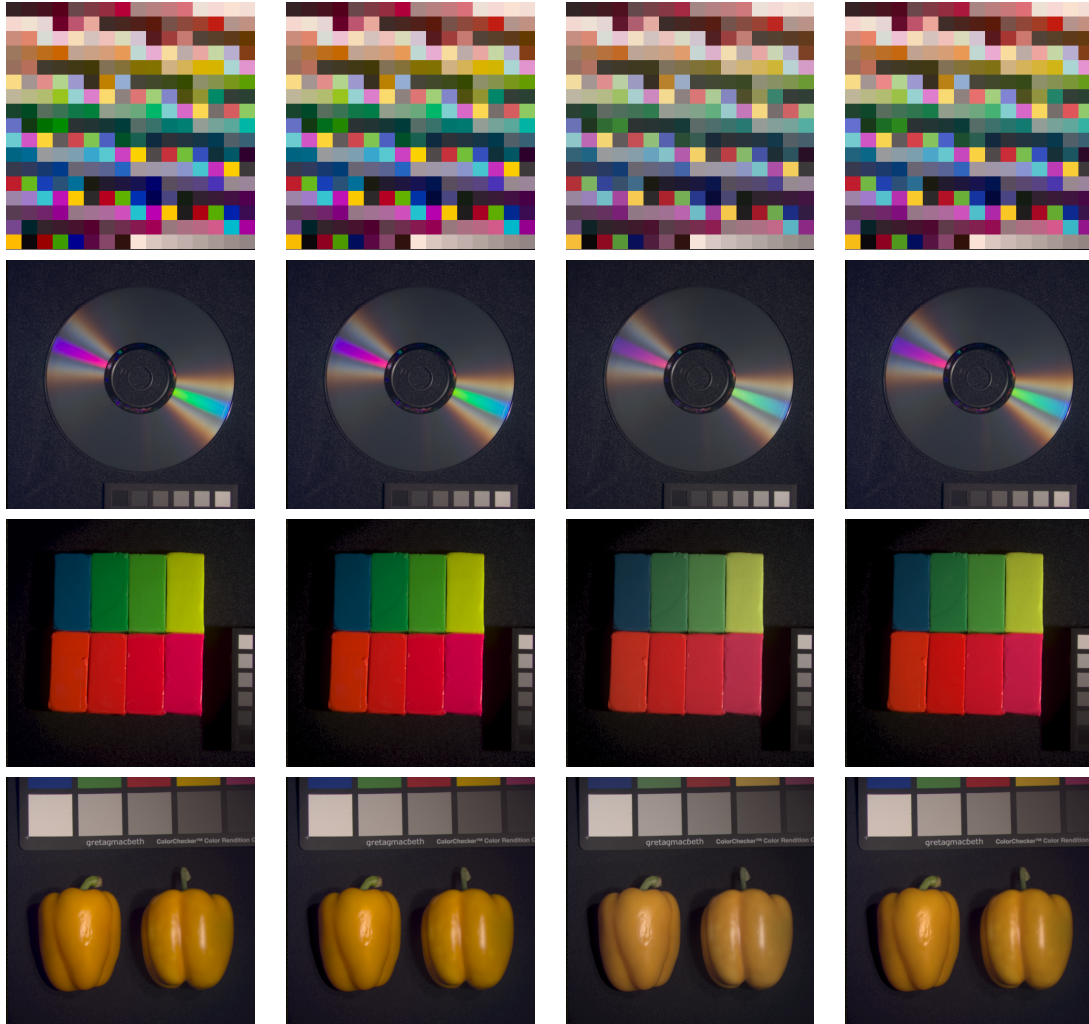


CANON 400D

Monochromatic transformation:

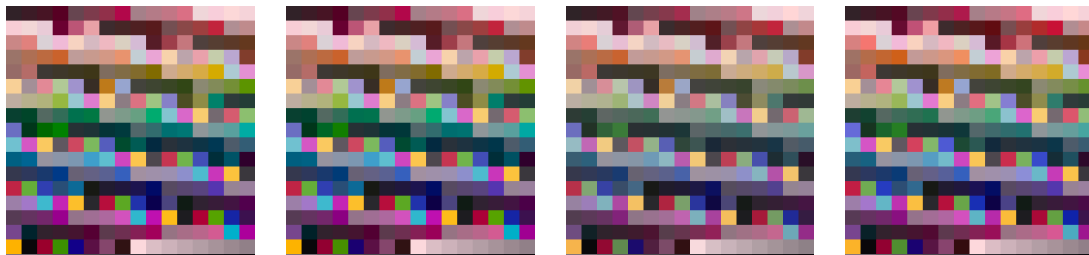


IT 8.7 transformation:

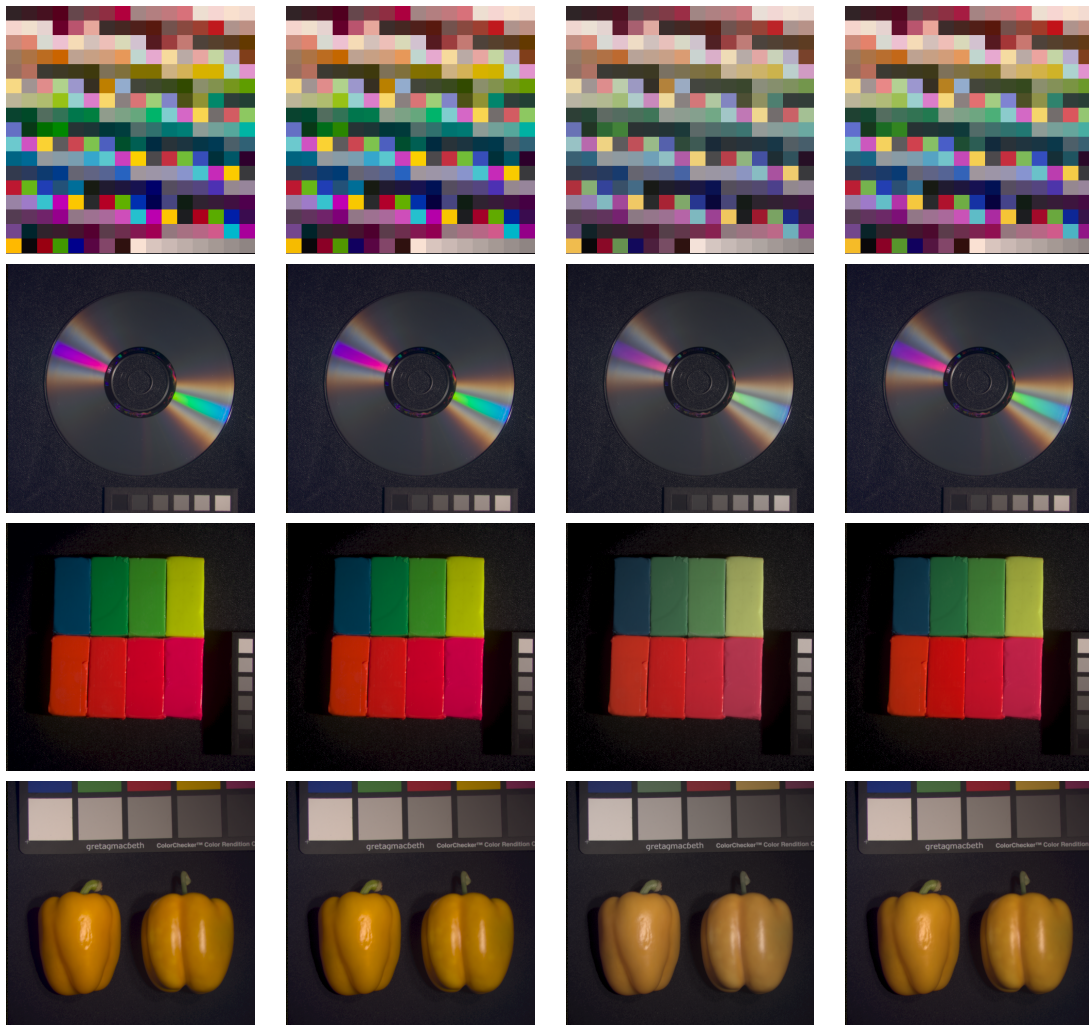


CANON 5D

Monochromatic transformation:

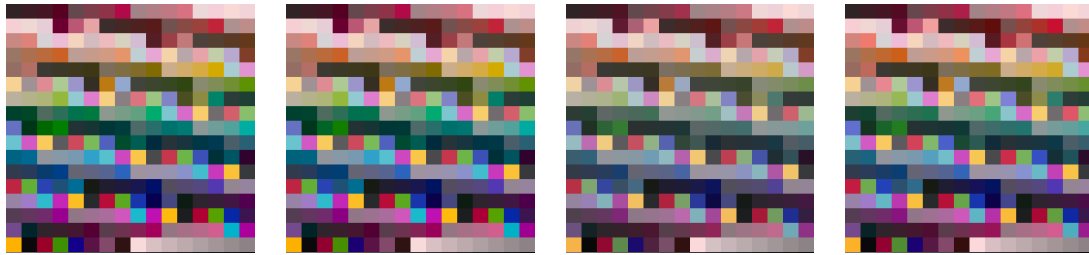


IT 8.7 transformation:

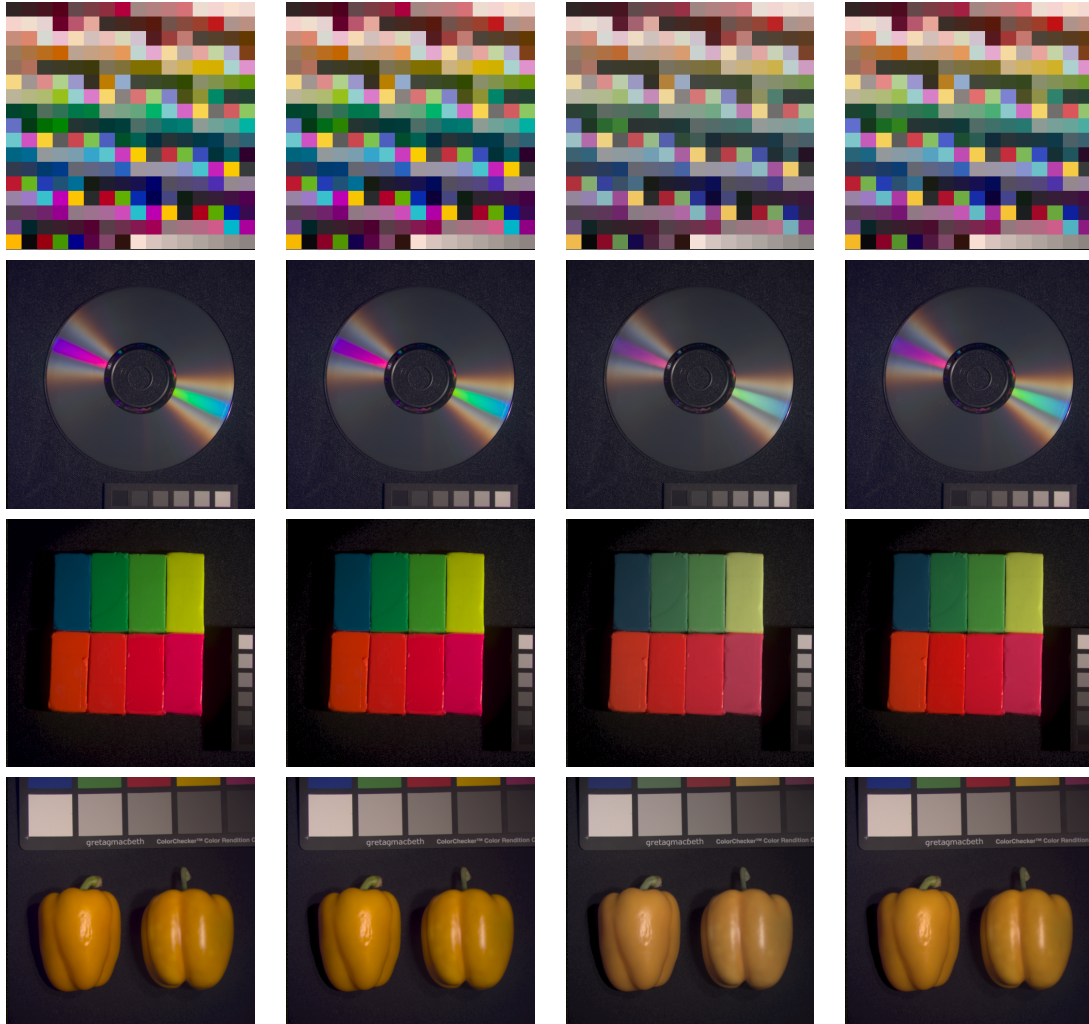


CANON 5D Mark 2

Monochromatic transformation:

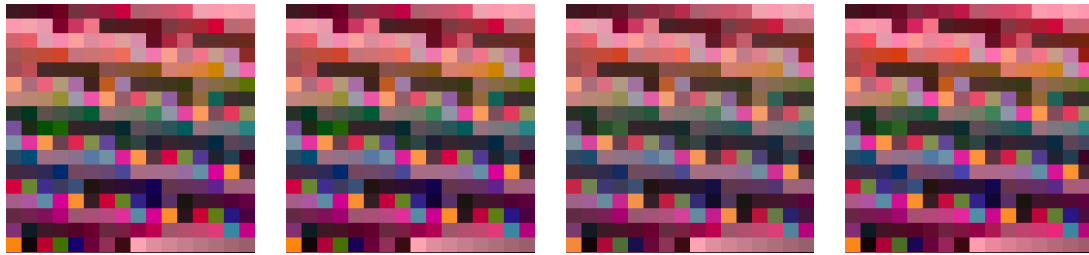


IT 8.7 transformation:

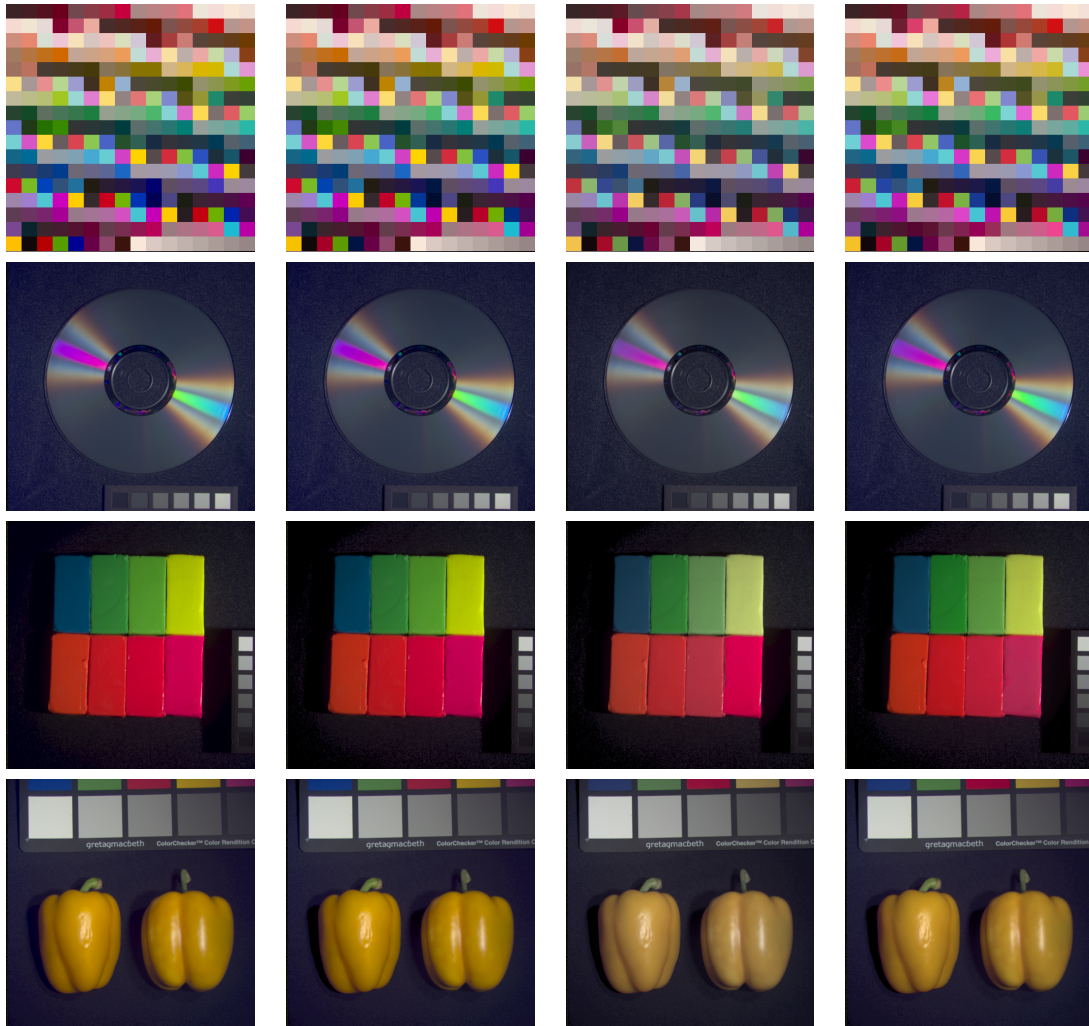


Ladybug2

Monochromatic transformation:

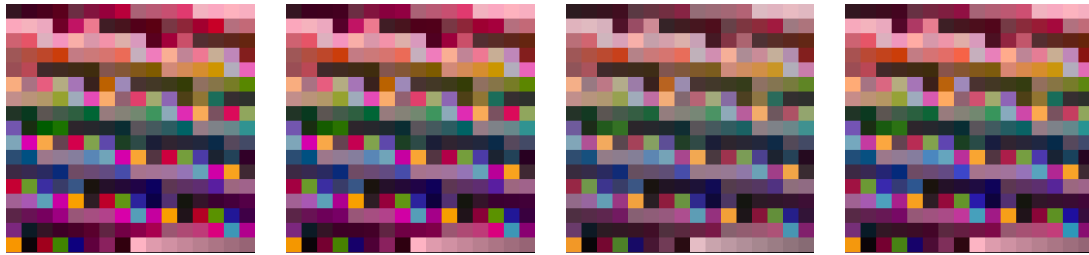


IT 8.7 transformation:

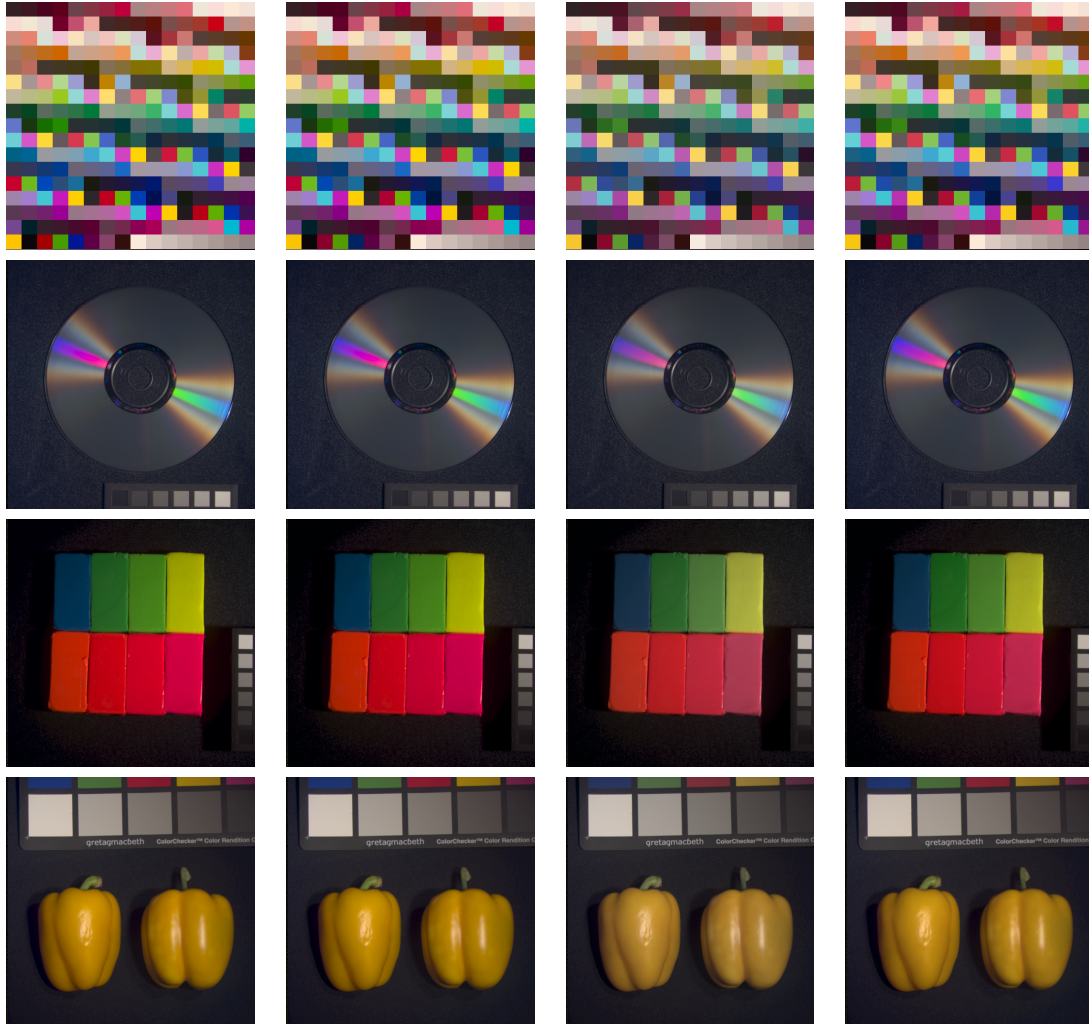


KODAK DCS 200

Monochromatic transformation:



IT 8.7 transformation:



CIE XYZ

