Laboration 2

Bildreproduktion och Bildkvalitet
(TNM097)

Device characterization and modeling
Part 2 – Output devices

Preparations

A prerequisite for this lab, focusing on modeling, calibration and characterization of output devices, is Lab 1, focusing on input devices. **It is also necessary and very important that you read the theory part of this document (written in black) prior to the scheduled time for the lab**. You will partly work with the same data as in Lab 1, and you will have good use of your code for evaluating characterization results in terms of ΔE_{ab} color differences. As preparations, you should further read and understand the theory of device calibration and characterization.

Introduction:

The joint Labs 1 & 2 focus on modeling and characterization of a complete color reproduction workflow, including both input and output devices (Fig. 1). In Lab 1, you investigated model-based and empirical characterization approaches for typical input devices (RGB-cameras) – upper part of Fig. 1. In this lab, lower part of Fig. 1, you will work with modeling, calibration and characterization of output devices, using additive color mixing (RGB) for reproducing colors. Much of the data you acquired in Lab 1, will be used here, as input.

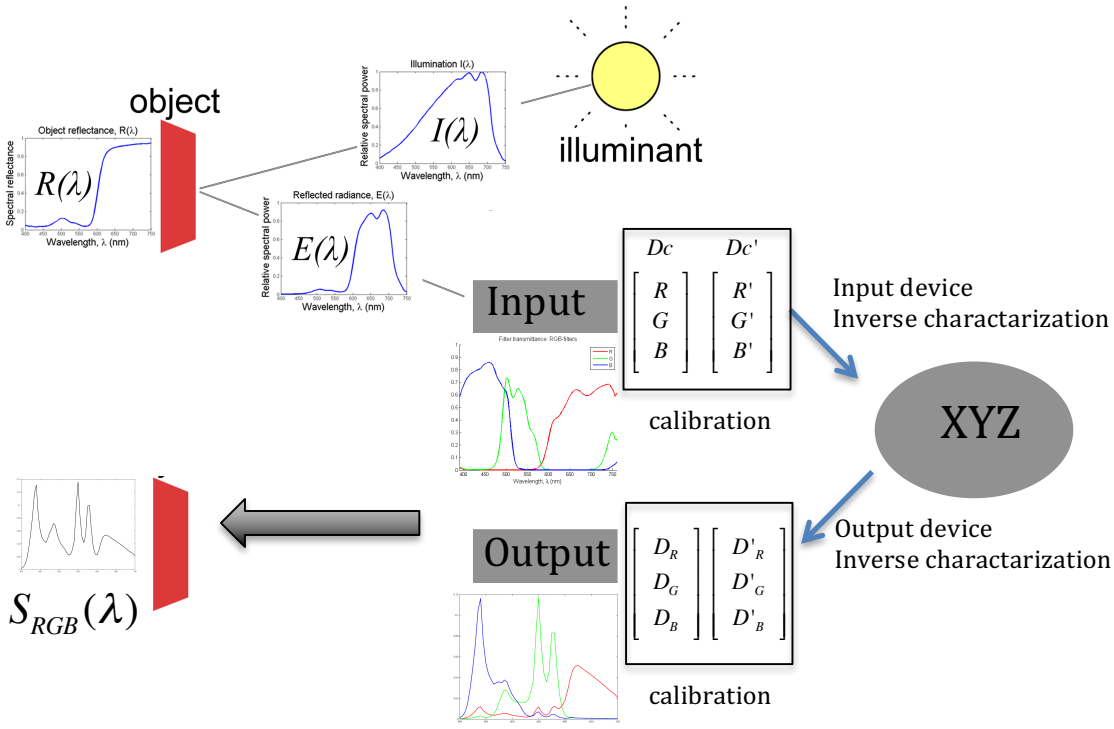

Figure 1. Overview of the complete color reproduction workflow used in Lab 1 & 2, including input- and output- RGB devices.

The data you need is available in the file Lab2.zip, including:

- The file *chips20.mat* contains spectral reflectance data for 20 different surfaces.
- The file *illum.mat* contains spectral data for different light sources.
- The file *xyz.mat* contains the color matching functions (CMFs), used for computing tristimulus values (XYZ) from spectral data.
- The file *TRC_display.mat* contains tone reproduction curves (TRCs) for a display. By loading this file, you acquire three *1 x 101* vectors TRCr, TRCg and TRCb, representing the tone

reproduction curves for R, G and B channels, respectively.
- The file *DLP.mat* contains spectral data for the RGB-primaries of a DLP-projector. *DLP* is a *61 x 3* matrix, in which column 1, 2 and 3 represent the spectral data for the R, G and B primaries, respectively.

Output device calibration and characterization

Figure 2 illustrates the workflow for calibration and characterization of output devices, such as displays and projectors. The calibration process involves setting brightness and contrast to fixed nominal values and establish the relationship between input RGB-values and the resulting display luminance (the relationship between **d** and **d'**). The forward characterization function predicts the output, **c**, (spectral signals or CIEXYZ values) depending on the input **d´** (RGB-values after calibration). On the other hand, the inverse characterization function determines the signal **d'** (RGB-values) that should be sent to the device in order to reproduce a target color, **c** (defined in CIEXYZ).

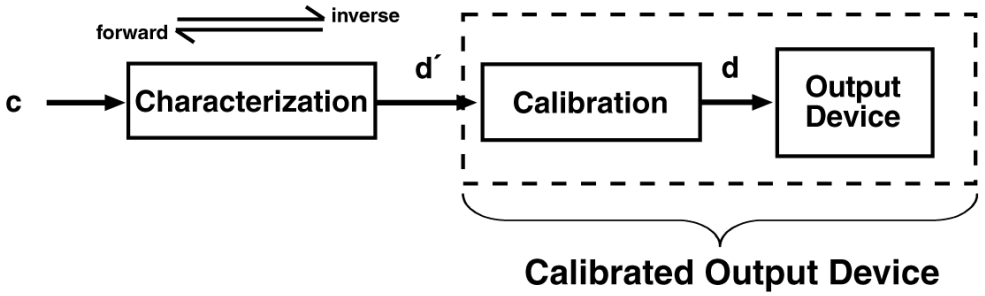

Figure 2. Calibration and characterization workflow for output devices.

For emissive output devices such as displays and projectors, reproducing colors via an additive mixing of red, green and blue (RGB) light, two assumptions can be made that greatly simplify display characterization:

Channel independence – each of the RGB channels operates independently of the others and the contribution of spectral radiance can be separated, as;

$$S_{RGB}(\lambda) = S_R(\lambda) + S_G(\lambda) + S_B(\lambda) \quad (1)$$

Chromaticity constancy – the spectral radiance from a given channel has the same basic shape and is only scaled as a function of the device signal driving the display. This reduces Eq. 1 to:

$$S_{RGB}(\lambda) = D_R' S_{R\max}(\lambda) + D_G' S_{G\max}(\lambda) + D_B' S_{B\max}(\lambda) \quad (2)$$

where $S_{RGB}(\lambda)$ is the resulting emitted spectral radiance (see Fig. 1), $S_{R\max}(\lambda)$ is the radiance emitted from the red channel at maximum intensity and D'_R is the digital input to the display, linearized in the calibration process. Converting spectral radiance to XYZ values, gives us the following relationship between the inputs to a linearized display D'_R, D'_G, D'_B and the resulting tristimulus values (XYZ):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \begin{bmatrix} D'_R \\ D'_G \\ D'_B \end{bmatrix} \text{ or, equivalent: } \mathbf{c} = \mathbf{A}_{CRT}\mathbf{d}' \quad (3)$$

The columns of the matrix $\mathbf{A}_{CRT}$ are the tristimulus values of the primaries R, G and B at maximum intensity and can be obtained by direct measurements from the display.
Note that the signals D'_R, D'_G and D'_B (or **d'**) in Eq. 3 should be linear in luminance, while D_R, D_G and D_B (or **d**) are the raw signals that drive the display. The relation between the linearized values and the raw values are established in the calibration process, which is usually referred to as linearization. In general, tone reproduction-curves (TRCs) that describe the display output for linear RGB-input are derived in the calibration process. For CRT-displays, this relation can be modeled using a single parameter, γ, for each channel (the well known gamma-correction):

$$D_R = D_{\max}\left(\frac{D'_R}{D_{\max}}\right)^{1/\gamma_R} \quad (4)$$

where, D'_R (and, similarly D'_G and D'_B) are linear in luminance, while D_R, D_G and D_B are the raw signals that drive the display. When the gamma correction is incorporated, the CRT will have a tone characteristic that is linear in luminance. Typical values of γ_R, γ_G and γ_B lies between 1.8 and 2.4.

1. Calibration of output devices

1.1.

Let's start by looking at the calibration process for an output device, in this case a display. The file *TRC_display.mat* contains tone-reproductions curves (TRCs) for the R- G- and B- channels of a display. Each TRC contains the measured output from the display to the linear input R=G=B=[0:0.01:1]. Start by looking at the three TRCs plotted against the linear input values [0:0.01:1]. For this display, the RGB-channels have slightly different TRCs. Can you predict how this will effect the reproduction of neutral colors?

.....
.....
.....
.....

1.2.

The image *Ramp_display* is an image of a neutral ramp with R=G=B=[0:0.001:1], displayed by this device, without calibration. Correctly reproduced, this image would be completely neutral (i.e. identical to the image *Ramp_linear*). You can compare the images, using *imshow*. Your task is now to perform linearization of the device, based on the TRCs. Write a simple Matlab-function that compensates an arbitrary RGB-image for the non-linear TRCs of the display. After this linearization, the image *Ramp_display* should look completely neutral (as *Ramp_linear*).

Hint 1: For simplicity, assume an input x=0:0.1:1 and a non-linear output response y, see Fig. 3. As seen in this figure, the output y is not linear with respect to the input x. For a linear output response, for example the input value x=0.34 would have resulted in y=0.34, but as seen in Fig. 3, x=0.34 results in y=0.14. If we want the output to be y=0.34, the input has to be almost x=0.55, follow the arrows in Fig. 3. In order to achieve a linear output response, the input x has to be changed correspondingly, see Hint 2.

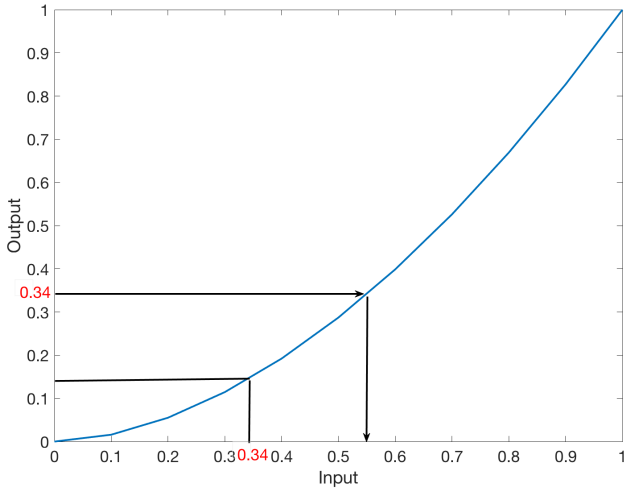

Figure 3. A non-linear response y to x=0:0.1:1.

Hint 2: In the curve in Fig. 3, the output y is only given at 11 points, namely at x=0:0.1:1. If we want the output value at another x-value, for example x=0.34, then we need to use an interpolation. In Matlab you can use the following,

```
>> interp1(x,y,0.34,'pchip')
```

which will return 0.14 for this example. However, as shown in Fig. 3, here we are interested to know the opposite, meaning that what x would result in y=0.34. This means that we need to look at the inverse of the function shown in Fig. 3 and in Matlab it can be done by switching x and y in the above command as,

```
>> interp1(y,x,0.34,'pchip')
```

which will return 0.55.

In the Matlab commands above, the value 0.34 can be replaced by a matrix/vector and the command will return a matrix/vector of the same size.
Notice that in this assignment, the TRCs only contain data for 101 levels, which means that you will need some type of interpolation for intermediate values, as shown in the above hints. Only one-line code is needed for calibration of each channel.

.....
.....
.....
.....
.....

1.3.

By measuring and compensating for TRCs in the calibration process, any display can be linearized (In fact, this type of linearization is often used even for input devices, such as RGB-cameras, and also for printers). However, the data you are working with here is for a CRT-display. This means that the TRCs can be modeled using the parameter γ, and compensated for using gamma correction. Given the gamma values γ_R = 2.1, γ_G = 2.4 and γ_B = 1.8, now compensate the image *Ramp_display* to be linear (neutral), using gamma-correction, according to Eq. 4.

.....
.....
.....
.....
.....

2. Spectral forward model of the output device

The output device you will work with from now on is a DLP-projector (Digital Light Processing), using DMD-technology (Digital Micromirror Device). Since this type of device controls the amount of light for each primary by altering the time, and not the power, the output is linear in luminance. Hence, no linearization functions are now needed, and you can assume that $D'_R = D_R$, $D'_G = D'_G$, $D'_B = D_B$ in these remaining exercises.

2.1.

Let's start by looking at the spectral forward model for the projector, according to Eq. 1. The file *DLP.mat* contains measured spectral radiance for the three RGB channels at maximum intensity, i.e. $S_{R\max}(\lambda)$, $S_{G\max}(\lambda)$ and $S_{B\max}(\lambda)$. Plot the spectra for the projector's primaries against the wavelengths 400:5:700 nm. Comments?

.....
.....
.....
.....

2.2.

By using Eq.2, you can compute the resulting output spectra $S_{RGB}(\lambda)$, for any input RGB-values (D'_R, D'_G D'_B) . Let's try to use the raw camera data from Lab 1 (*RGB_raw*), as input to the projector (i.e. using the signals D_C in Fig. 1). This would correspond to a color reproduction workflow without any color management or calibration, using the device dependent camera-RGB as raw input to the projector.
Recall from Lab 1 that RGB_raw is a *3 x 20* matrix, in which each column contains the RGB-value of each object in *chips20* before calibration.
Convert the resulting spectral radiance, $S_{RGB}(\lambda)$, to XYZ-values, by using the white point for D65 for computing the normalization factor, *k* (the same way as in Lab 1). Evaluate the result by the mean and max ΔE_{ab} color differences for the 20 color samples, between the input values (XYZ_ref) and the corresponding colors reproduced by the projector.

Hint: By using Eq. 2, $S_{RGB}(\lambda)$ for, for example object no. 8, can be calculated by: S_{RGB8}=RGB_raw(1,8)*DLP(:,1)+ RGB_raw(2,8)*DLP(:,2)+ RGB_raw(3,8)*DLP(:,3). Equivalently, this can be calculated by: S_{RGB8}=DLP*RGB_raw(:,8). For all 20 objects, this can be calculated by, S_{RGB}=DLP*RGB_raw, which is a *61x20* matrix.

By using a for loop or utilizing the properties of matrix multiplication, you can find $S_{RGB}(\lambda)$ for all twenty objects, which are then converted to XYZ-values.

.....
.....
.....
.....
.....

2.3.

Now perform the same experiments, using your calibrated camera values (RGB_cal) as input instead. This would correspond to a color reproduction workflow, still without any color management, but with calibration for the input device (the signals D_C' in Fig. 1). Again, evaluate the result by the mean and max ΔE_{ab} color difference for the 20 color samples, between the input values and reproduced colors.

.....
.....
.....
.....
.....

3. Inverse characterization of the output device

To use the projector in a color management workflow, you first need to derive the inverse characterization function, i.e. the 3x3 matrix that determines the input values D'_R, D'_G, D'_B. In Fig. 1 this corresponds to the function that takes your device-independent XYZ-data (from Lab 1) and computes the correct device-dependent RGB-values that should be sent to the projector.

3.1.

Start by deriving the tristimulus values needed for $\mathbf{A}_{CRT}$ in Eq. 3, i.e. the XYZ values for the primaries at maximum intensity (make sure to use the correct normalization factor, k). Notice that in $\mathbf{A}_{CRT}$, column 1, 2 and 3 contain the XYZ-values corresponding to DLP(:,1) (i.e. Red), DLP(:,2) (i.e. Green) and DLP(:,3) (i.e. Blue), respectively.
The 3x3 matrix that determines the device dependent values (D'_R, D'_G, D'_B) for device independent target colors (in XYZ), is then simply given by the inverse of the matrix $\mathbf{A}_{CRT}$. Because of the linearity of additive color reproduction and the assumptions in Eqs 1 and 2, a simple 3x3 matrix can accurately describe the relationship between device-independent XYZ and device–dependent RGB signals.

.....
.....
.....
.....
.....

3.2.

Now, let's simulate the complete color management workflow, including both input- and output-devices. Use your inverse function for the input device to compute XYZ values from the calibrated camera values (your result from 3.4, or 3.5 in Lab 1). Then, use your inverse characterization for the output device (i.e. inv($\mathbf{A}_{CRT}$)) to compute the device dependent input to the projector (D'_R, D'_G, D'_B) from the XYZ values for the 20 patches. Finally, use the forward model to predict the spectral output from the projector (as in 2.2 & 2.3). Make sure that you understand each step (compare to Fig. 1) and the type of data your working with (spectral / device-dependent / device-independent / calibrated / raw)!
Evaluate the result by the mean and max ΔE_{ab} color difference for the 20 samples, between the input values (XYZ_ref) and the reproduced colors on the projector.

.....
.....

.....
.....
.....

3.3.

Let's take a closer look at the device dependent input to the projector (D'_R, D'_G, D'_B) that you computed in 3.2. Do you find anything wrong? Can you explain why this happens?

.....
.....
.....
.....

3.4.

Now lets modify the RGB-values (D'_R, D'_G, D'_B) to lie inside the range [0.1]. Use the modified values in the forward model (as in 3.2) and compare the results. Comments?

.....
.....
.....
.....
.....

3.5.

You are given the function *plot_chrom_sRGB* that can be used to compare the chromaticity coordinates for a display to the coordinates of the sRGB standard (based on CRT-displays). Use this function with your data A_{CRT} to plot the gamut for the projector (in blue) as comparison to sRGB (black). Any comment on the projector gamut? What can you do when a target color is outside the gamut of a device, and cannot be reproduced correctly?

.....
.....
.....
.....
.....

3.6.

Let's take a closer look at one of the color samples (from *chips*) that you have used (preferable one that gives a small ΔE_{ab} color difference). Look at the ΔE_{ab} color difference between the input value (XYZ_ref) and the reproduced color (computed in 3.4). Then plot and compare the spectra for the original color sample (under D65) and the reproduction $S_{RGB}(\lambda)$ for the same color. In Fig. 1 this corresponds to the spectral signal $E(\lambda)$ as input to the camera, and the resulting spectra $S_{RGB}(\lambda)$, produced by the projector. Can you draw any conclusions regarding the accuracy of the reproduction, in terms of spectral similarity vs. perceived color difference ΔE_{ab}?

.....
.....
.....
.....
.....