

Matlab Camera Characterization Toolbox v1.0

Documentation

Victor Medina

January 22, 2017

Contents

1	Introduction	1
2	Main interface	1
2.1	Camera characterization	2
2.1.1	Characterization process	2
2.1.2	Overexposed and underexposed photographs	4
3	Display characterization	4
3.0.1	Characterization process	4
4	Image calibration	7
5	Image calibration (batch)	9
6	Known bugs	10

1 Introduction

2 Main interface

The Image tools interface can be opened by calling the *Image_tools* script from the Matlab command line:

```
>> Image_tools
```

This script calls the main user interface for the toolbox, which offers functionality for display and camera characterization, image calibration, and several other utilities. The options available in this menu are (Fig. 1):

- Calibration tools
 - Camera characterization
 - Display characterization
 - Image calibration
 - Image calibration (batch mode)
- Utilities
 - Load image
 - Select pixels
 - Delete selection
 - Average
 - Colorimetry tests

2.1 Camera characterization

The camera characterization option (Fig. 2) allows to estimate a camera's transfer function from a series of corresponding colorimetric and radiometric measurements.

2.1.1 Characterization process

This process is meant to be done with some type of color chart (e.g. the Macbeth color chart). The general process consists of the following steps (marked in red in Fig. 2):

1. **Open a color chart.** The *Open color chart* button (Fig. 2-(1)) allows to load a color chart image (in a new window) to sample the colorimetric values from it. This is typically a photograph of a physical color chart, taken with the same device that we want to characterize under some observation and illumination conditions, with some set of chosen acquisition parameters (focal length, ISO, aperture, and shutter speed).
2. **Select color pixels and read their value.** The *select sample* button (Fig. 2-(2.1)) creates a pixel selection handle in the color chart window (see Fig. 3). This selection handle can be resized and moved around the image with the mouse pointer until it covers all the pixels to be sampled. Once the desired pixels are selected, the *store pixels* button

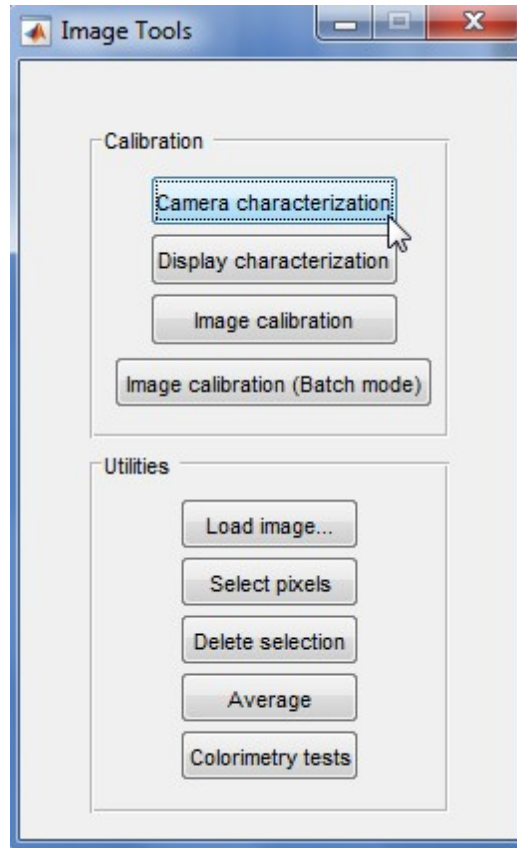


Figure 1: MCCT main menu

(Fig. 2-(2.2)) averages their color values (in the original color space of the image) and adds a new entry in the *sampled colors* table (Fig. 2-(2)).

3. **Fill in the radiance table.** The radiance values should be measured with some radiance measuring device (in our case, a spectroradiometer) under the same observation and illumination conditions used to photograph the color chart¹. These values can be entered in the *Measured radiance* table (Fig. 2-(3)) either by hand, or pasted from some external file such as an Excel data sheet.
4. **Fill in the camera's properties (optional)** Optionally, the camera's model and acquisition parameters can be entered (Fig. 2-(4)) so that they show up in the generated graphs. The main purpose for this option

¹Keep in mind that these values will be the same regardless of the acquisition parameters used to photograph each color chart

is so that graphs can be distinguished from one another and organized accordingly.

5. **Characterize** Once all the necessary data has been introduced, the *characterize* button (Fig. 2-(6)) generates an estimation of both the forward (radiance→color) and inverse (color→radiance) transform matrices for the device. A correlation vector is computed for each transform matrix, indicating the degree of reliability for each channel’s estimation, where 0 is very low and 1 is a perfect correlation. After the estimation process is finished, two plots will pop up (Fig. 4) to illustrate the estimated transform function.

2.1.2 Working with overexposed and underexposed photographs

In some cases, the illumination and acquisition conditions may result in over- or under-exposed photographs. When this happens, the wrongly-exposed pixels should not be used in the characterization because they may bias the estimation process, producing wrong matrices with a lower correlation. We can automatically exclude these pixels by setting a threshold value for overexposure and/or underexposure (Fig. 2-(5)). The corresponding radiance entries will automatically be excluded as well to avoid a mismatch between the number of entries in both tables.

3 Display characterization

The display characterization option estimates the color transform function of a display under some observation and illumination conditions, given a set of corresponding colorimetric inputs and measured radiometric outputs—under those conditions— and a characterization model.

3.0.1 Characterization process

This characterization process has been optimized for liquid-crystal displays (LCD), but can also be used for other types, such as older cathode ray tube (CRT) displays. Typically, a display is characterized by displaying a set of known monochromatic (i.e. only red-, green-, or blue-channel component) color samples on the screen and measuring the radiance emitted on its surface with some radiance-measuring device (in our case, a spectroradiometer). A model describing the response of the display can then be estimated from the correspondence between input and output.

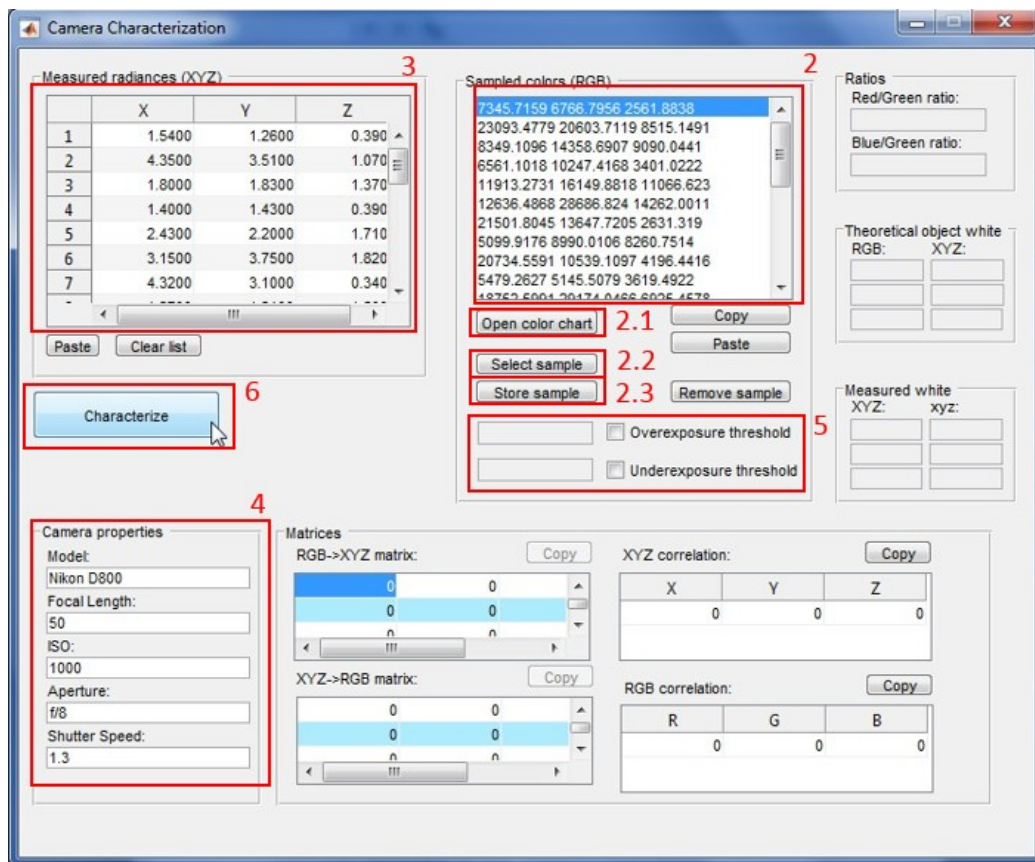


Figure 2: Camera characterization GUI. The numbers in red indicate GUI components referenced in the text.

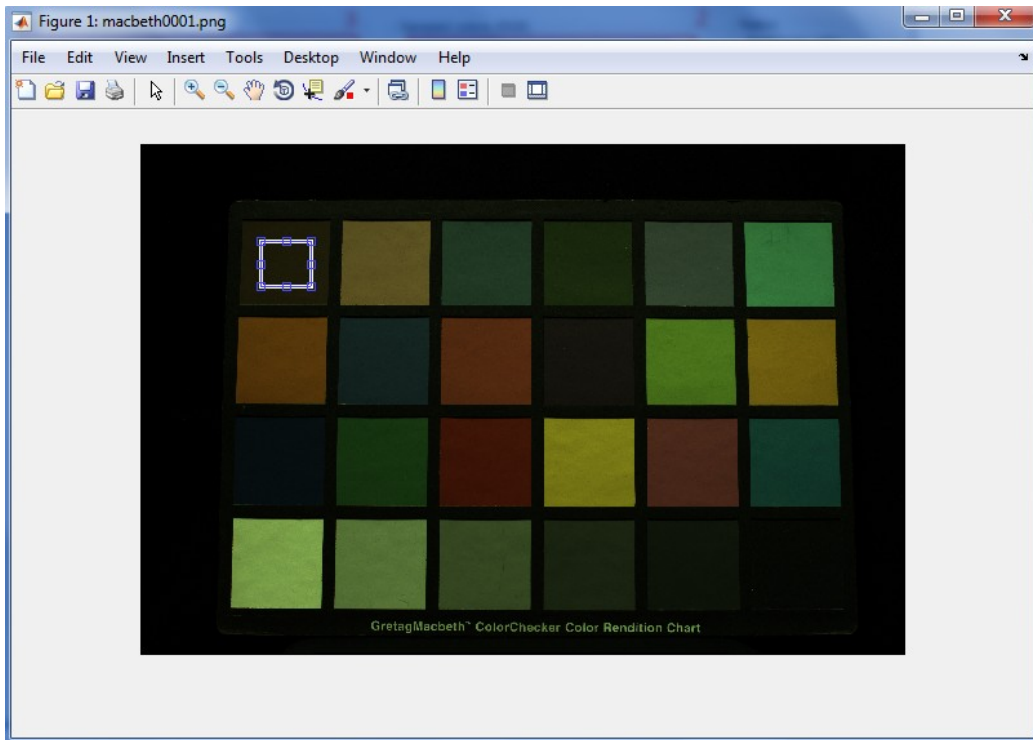


Figure 3: Macbeth color chart. The selection rectangle in the first color target indicate the pixels that will be sampled.

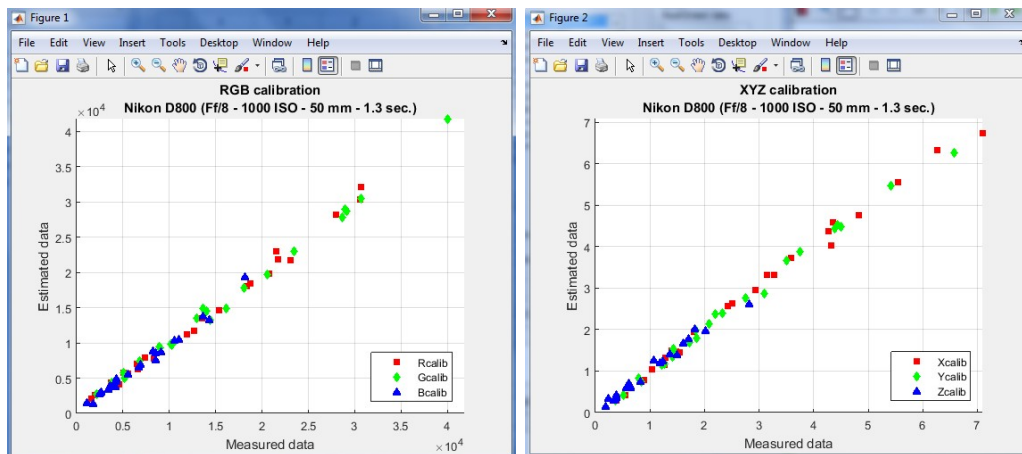


Figure 4: Colorimetric (left) and radiometric (right) estimation graphs which show, on the vertical axis, the value calculated with the estimated matrix, for each measured value shown on the horizontal axis. A straight line indicates a good correlation between real and estimated values.

The process to estimate a characterization model for a display with our tool consists of the following steps (marked in red in Fig. 5):

1. **Fill in the colorimetric input and radiometric output** The input color samples and output radiometric measurements must be entered in the corresponding tables (Fig. 5-(1)) for the red, green, and blue samples. Keep in mind that the same number of samples must be entered in all the tables to avoid a mismatch error.
2. **Select the desired characterization model** Four models are available to choose from (Fig. 5-(2)): Gain-Offset-Gamma (GOG), Gain-Offset-Gamma-Offset (GOGO), Gain-Gamma-Offset (GGO), and a simple display characterization model based only in one gamma parameter. These four models, traditionally used to characterize CRT displays, are also recommended in the literature for LCD characterization, but since they are often not enough to model some characteristics of LCDs, we use them in combination with an additional transform matrix.
3. **Characterize** Once all the required information has been filled in, we can proceed with the estimation of the characterization model parameters and the transform matrices (Fig. 5-(3)). The quality of the estimated model is indicated by the coefficient of determination (R^2) which represents the goodness of fit, between 0 and 1, with 1 being a perfect fit.
4. **Display the characterization graph** The *display graph* button (Fig. 5-(4)) opens up a plot that illustrates the correlation between measured and estimated values.

4 Image calibration

Image calibration is the process that allows to see how a photograph, taken with a camera whose transform matrix is known, will look on a characterized display. This process is performed in two stages: first, the device-dependent RGB pixel values in the photograph—which are expressed in the camera’s RGB color space (Fig. 8-(a))—are transformed into device-independent XYZ pixel values using the camera’s inverse transform matrix (Fig. 8-(b)); then, those XYZ values are transformed back to device-dependent RGB values using the display’s characterization model (Fig. 8-(c)). The result of the first conversion stage gives an approximation of the radiometric information

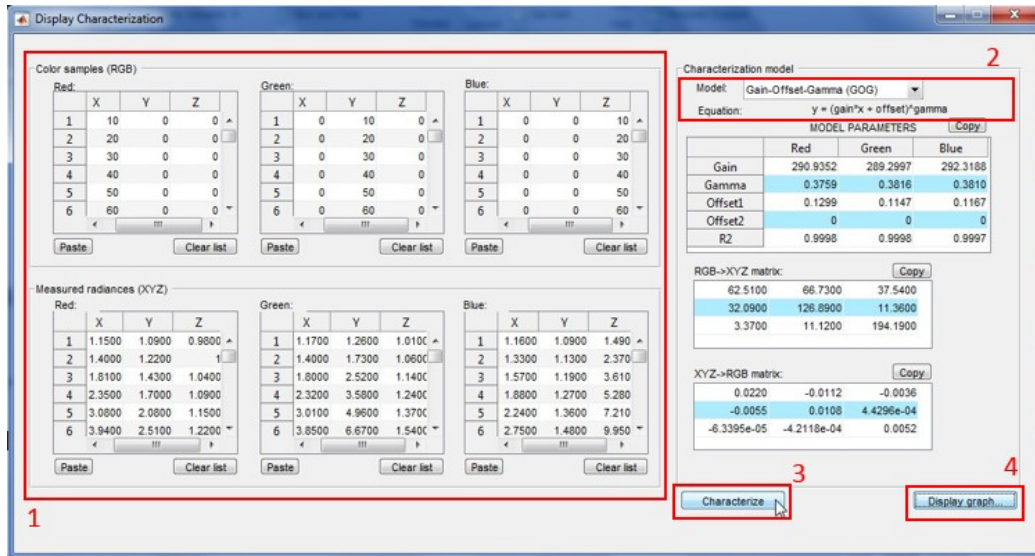


Figure 5: Display characterization GUI. The numbers in red indicate GUI components referenced in the text.

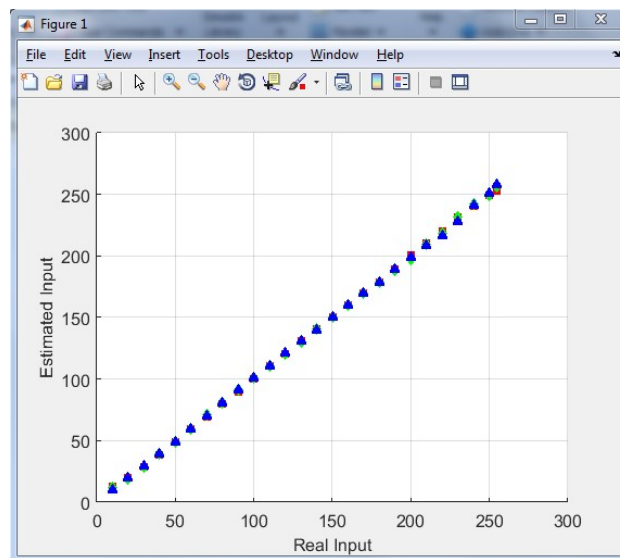


Figure 6: Characterization graph that shows, for a set of measured screen output radiances, the correspondence between real colorimetric input (horizontal axis) and an estimation of the colorimetric input required to produced that same output, using the estimated characterization model (vertical axis). A straight line indicates a good correlation between real and estimated values.

—in the XYZ color space— in the scene where the photograph was taken and the second stage produces the final color calibrated image.

The image calibration tool in MCTB provides this functionality as well as the option to export individual channel images from each step of the process. It may be the case that some photographs have a dynamic range higher than that of the display; this tool also offers the option to set a maximum luminance value for a display, as well as a luminance adaptation factor, so that pixel values that fall outside the display’s luminance dynamic range can be converted to valid values.

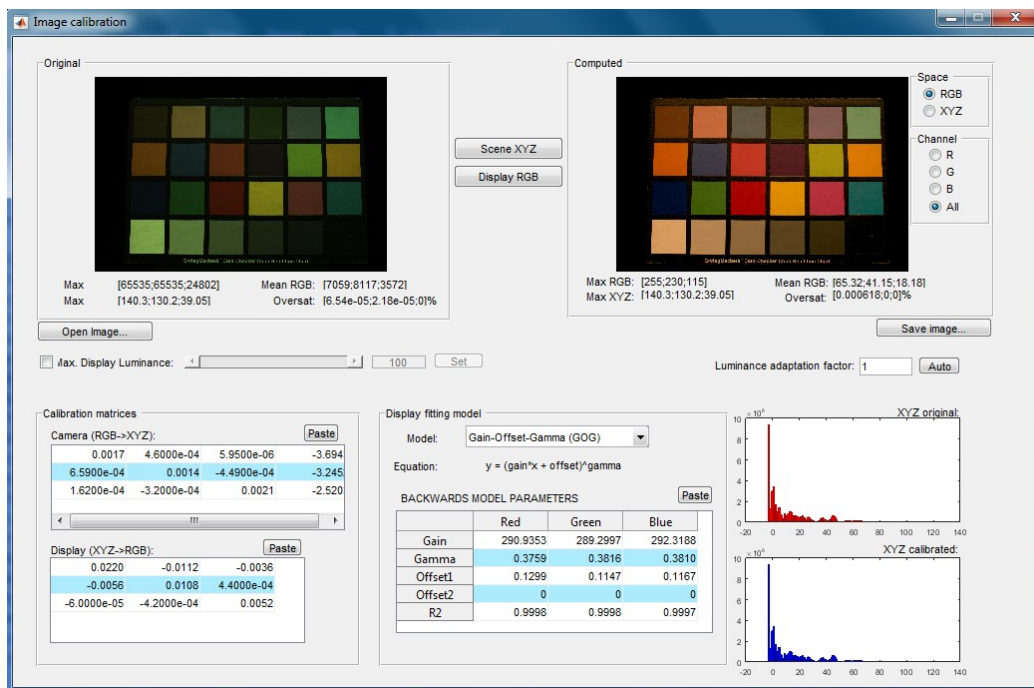


Figure 7: Image calibration GUI.

5 Image calibration (batch)

This tool offers a similar functionality as the non-batch version, except that it allows to load several images at once, creating a calibration queue and saving them automatically. This is useful in cases where one needs to calibrate many images for the same camera and display.

6 Known bugs

The following is a list of known issues in each version of this software:

Version 1.0:

- Common to all:
 - Pasting radiance values from an external file throws a button callback error (Error Message: “Error using reshape”) when numbers use a decimal digit separator other than a point.
- Camera characterization:
 - Clicking on *select sample* creates a new pixel selection handle every time. While this should not be a big issue, it is worth noting that *store sample* will only sample those pixels selected by the last created handle, ignoring any other existing selection handle.

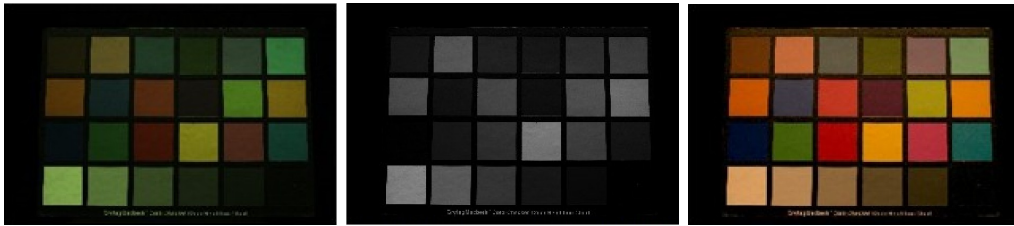


Figure 8: Steps in the image calibration process. From left to right: The original photograph (a) is converted from the camera's RGB color space to a device-independent space (XYZ) (b); then, the image is converted back to RGB using the display's characterization model (c).