Camera Calibration

Here we introduce a method for calibrating a camera to capture stimuli. We explain the theory behind the required colour conversions and provide example code for implementing these steps.

1. Prerequisites

MATLAB (tested on 2022b) (software)

PsychToolbox-3 (tested on version 3.0.19) (software)

MacBeth Colour Checker (hardware)

Camera that can capture images in RAW format (hardware)

2. Theory

Cameras capture images in RGB coordinates, but as you will have learnt by now, RGB is a device-dependent colour space and thus not directly useful as a description of colour for a colour vision scientist. Thus one must find the matrix N that converts from RGB coordinates to a device independent colour space, such as CIE 1931 XYZ, LMS, etc (see Worksheet XX for more details). Here we use the CIE 1931 XYZ colour space to illustrate how to do these conversions, but one useful exercise for the student would be to replicate what is described in this worksheet for LMS space (think about what material from other worksheets you will need in order to do this).

As we see in the Chromaticity Conversions worksheet, it is relatively straightforward to convert from RGB space to XYZ space if one knows the spectral profile of the RGB primaries (and these behave linearly). It is relatively easy to measure the primaries for a display. But for a camera, its colour space is defined by its RGB sensor spectral sensitivities, and these are not as directly accessible. Thus in order to, solve for the matrix N which converts RGB->XYZ for a camera, one must in effect “estimate” the RGB sensor spectral sensitivities of the camera. This can be done by using a variety of band pass filters in front of the camera to estimate the sensitivities of the sensors, but this is a time-consuming approach that requires additional hardware.

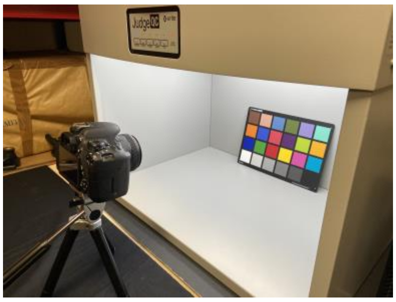
One alternative approach in practice is to find an estimate of the matrix N. This can be done by capturing an RGB image of multiple colour samples with known XYZ values, and finding a least-squares solution for the matrix N that best converts the multiple RGB triplets to XYZ. A standard set of colour samples to use for this are the MacBeth Colour checker squares, but any set of calibrated colour samples can be used.

3. Practical Implementation

To implement this approach one needs to capture a RAW image of the MacBeth Colour Checker (ideally under a calibrated broadband light source such as D65) with the camera to be calibrated, and take spot measurements of the XYZ values of each colour sample. Then one can use the provided MATLAB file: **cameraCalibration.m** to implement the least-squares solution to find the matrix N.

3.1. Camera measurements

The camera must be set up in Manual mode. It must also be set up to save images as RAW files not JPEG. RAW files do not apply any compression and thus likely to be linear RGB values. (Why is this important? How could you verify the linearity if you were so inclined?) A possible layout for using the MacBeth Colour Checker to calibrate a camera is shown in Figure 1.



The camera can be plugged into a computer via a USB cable. This is the preferred method to capture images. When plugged into the PC the remote shooting software should load automatically, or may require installation from online depending on the camera you are using.

Figure 1: Photograph of the way to set up the MacBeth Colour Checker and camera for calibration.

The image taken should clearly contain all the MacBeth Colour Checker samples and nothing else. Once captured the RAW image can be loaded into MATLAB and the RGB values from each sample can be found using the inbuilt MATLAB function colorChecker as below:

```
>> I = imread("myColourCheckerImage.raw")
>> chart = colorChecker(I);
```

3.2. Spot measurements

The XYZ values of each sample must also be measured. This can be done using a spectroradiometer such as the SpectroCAL or PR-670. Instructions on how to use these devices can be found in the Light Measurement Worksheet. The spectroradiometer should be set-up to point at each sample in turn and then an XYZ measurement recorded from that sample. This thus requires a fair amount of manual effort as the spectroradiometer must be moved between each measurement.

3.3. Finding N (i.e. RGB->XYZ)

Once one has the RGB and XYZ values of each sample, one can find the matrix N using the MATLAB mrdivide operator (/). Example code for doing this, using example RGB and example XYZ measurements is provided in the following script:

```
>> cameraCalibration.m
```

4. Discussion

The procedure described above will provide an accurate estimate of XYZ as long as the underlying camera spectral sensitivities are a linear transformation away from the XYZ colour matching functions. This is not the case for many real cameras, so the estimates of XYZ from camera RGB should be regarded as an approximation. A number of papers consider ways to make the estimates more accurate, when there are prior constraints known about the spectra being photographed. One reason that imaging colorimeters, that provide calibrated measurements of XYZ values for each pixel in the image, are expensive is that great care is needed to produce camera RGB sensors whose spectral sensitivities are a linear transformation way from the XYZ colour matching functions.

Camera Calibration