Mapgains Guide

Author: Kim Miikki
Date: 19.10.2021

1 Introduction

The white balance of the Raspberry Pi camera can be controlled with red and blue channel gains. Increasing the red or blue gain reduces the sensitivity of all other channels. This program is used to measure the effects of red and blue channel gains on RGB channels. The mapgain is a calibration program.

2 System Requirements

Operating System: Raspberry Pi OS or Linux (only analyze mode)

Hardware: Raspberry Pi camera module v.2 or HQ for data acquisition

Python 3 with Matplotlib, Numpy and OpenCV.

3 Calibration and Analysis Method

The RGB mapping plan consists of following parameters: exposure time, red gain range and blue gain range. Fow calibration purpose a gray or white card is also needed. The resolutions of the ranges can be controlled by step parameters, and the range are defined in following way:

Red gain: [start, start+step, …, end], length=(end-start)/step+1
Blue gain: [start, start+step, …, end], length=(end-start)/step+1

All combinations of red and blue gains can be represented with a 2D matrix, where red gains are rows and blue gains columns. Programmatically the red values correspond to the outer loop and the blue values to the inner loop.

Each combination of the red and blue gains are measured by capturing a small picture from the center or ROI area, thereafter mean RGB values are calculated and stored to the measurement matrix.

The first step, data acquisition, requires a Raspberry Pi with a camera. The second step is analysis, and does not require a Raspberry Pi if the program is started in file mode.

R, G and B values are flattened to 1D arrays, their absolute RG, RB and DG distances are calculated, and finally the mean distance of these three distances. The minimum value of the mean distance value in the array is the calibration position. The index of red gain array is calculated with integer division: *position // length of blue array*, and the index of blue array with modulus: *position % length of blue array*. A sample image with minimum mean distance is captured if the program is executed in capture mode.

RGB and gray maps are saved in pickle format for analysis use. The mandatory ranges R and B are saved as CSV files.

If all files creation is enabled, the data will also be saved in clear text and all channels as 2D raster images. Finally, the distance distribution graph and the calibration results are saved.

4 Program Usage

The program will be executed in capture and analysis mode if no arguments are given. Here is a list of the optional arguments:

optional arguments:
  -h, --help show this help message and exit
  -f F   file mode: read rgbmap-YYYYMMDD-hhmm.pkl
  -a   auto, non-interactive mode
  -d   Disable image preview
  -n   Do not create all analysis files

When file mode is selected, image capture is disabled. The red and blue gains ranges CSV files are mandatory, and must be present in the file mode.

The auto mode will capture averything based on source code default values.

If selecting -d, no preview window is displayed during capturing the images.

Argument -n will save only mandatory analysis files, calibration file and distance distribution plot. In capture mode, also a sample image will be captured.

The program working directory is same as current directory. This information is given when the program is started, and thet is also the path where all the results are stored.

The prefixes and extensions for files generated, are listed in the following table. DT is an abbreviation for the date-time string in the format YYYYMMDD-hhmm and represents the analyzis time.

Prefix-DT.extension	**Mode**	**Optional**	**Usage**
rgbmap-DT.pkl	both, mandatory	-	RGB data in binary pickle format
rgbmap-b-gains-DT.csv	both, mandatory	-	Blue gains range values as an array
rgbmap-r-gains-DT.csv	both, mandatory	-	Red gains range values as an array
rgbmap-cal-DT.txt	both	-	Calibration log and results
rgbmap-cal-image-DT.png	capture	-	Calibrated sample image
rgbmap-r-DT.csv	both	Yes	Red channel data
rgbmap-r-im-DT.png	both	Yes	Red channel raster image
rgbmap-g-DT.csv	both	Yes	Green channel data
rgbmap-g-im-DT.png	both	Yes	Green channel raster image
rgbmap-b-DT.csv	both	Yes	Blue channel data
rgbmap-b-im-DT.png	both	Yes	Blue channel raster image
rgbmap-bw-DT.csv	both	Yes	Gray mean data
rgbmap-bw-im-DT.png	both	Yes	Gray mean data raster image
rgbmap-mdist-DT.csv	both	Yes	RGB mean distance array
rgbmap-mdist-DT.png	both	-	RGB mean distance plot

Table 1. Analysis files.

5 Use Cases

Two use cases are presented in this section. The first is a fast calibration and the second a multi-stage calibration. In both cases a white calibration card was used with a flicker-free LED lamp as the main light source.

5.1 Fast Calibration

This two step calibration is based on the RPI camera auto white balance reading and one calibration stage. In Fig. 1 is shown then calibration card and a ROI of it which was selected with *roi-select.py*.

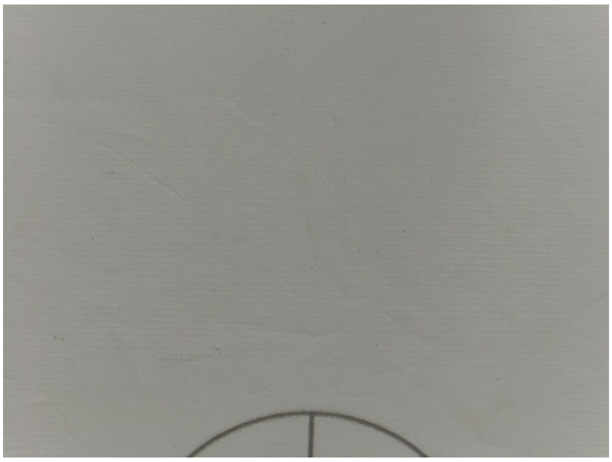

Figure 1. White calibration card: FOV and ROI.

Rough gain values can be acquired by reading the auto white balance gain values with *awb_gains.py*:

```
$ awb_gains.py
Auto white balance gains for Raspberry Pi camera module imx477

Red gain:  3.35 857/256
Blue gain: 1.45 371/256
```

Theese gains are seeds for the red and blue gain ranges. In this case following ranges were selected:

R: 3.2-3.5, step 0.01
B: 1.3-1.5, step 0.01

These values was entered to *mapgains.py*:

```
$ mapgains.py
Mapgains - Map Raspberry Pi gamera gains and calibration
(C) Kim Miikki 2021

ROI file found in current directory: 0.39,0.3036,0.1916,0.2115

Current directory:
/home/pi/python/20211019-mapgains

Select exposure time (250...200000000 µs, Default=20000: <Enter>): 5000

Select red gain min (0.0001...8, Default=1: <Enter>): 3.2
Select red gain max (3.2...8, Default=8: <Enter>): 3.5
Select red gain step (0.0001...0.2, Default=0.2: <Enter>): 0.01

Select blue gain min (0.0001...8, Default=1: <Enter>): 1.3
Select blue gain max (1.3...8, Default=8: <Enter>): 1.5
Select blue gain step (0.0001...0.2, Default=0.2: <Enter>): 0.01
```

```
Capture mode

rgain,bgain: red,green,blue
3.2,1.3: 192.193,198.156,183.968
3.2,1.31: 191.687,197.621,184.676
...
3.5,1.5: 201.658,188.399,198.176

Analyzing data
qt5ct: using qt5ct plugin

Mode: capture and analyze
Time: 2021-10-19 16:13:44.869486

Capture time: 0:03:15.175772
Analyze time: 0:00:07.794050

Calibration
-----------
Red gain : 3.26
Blue gain: 1.42
Minimal distance: 0.077

Red and blue gains
-------------------
R gain min : 3.2
R gain max : 3.5
R gain step: 0.01

B gain min : 1.3
B gain max : 1.5
B gain step: 0.01

Matrix   : 31x21
Position : [6,12]
Index    : 138

Distance distribution
---------------------
N     : 651
Min   : 0.077
Max   : 14.509
Mean  : 6.269
Median: 6.165
Var   : 9.244
```

After some minutes RGB values of all gain combinations were measured and analyzed. The calibration values in this setup were rgain=3.26 and bgain=1.46 with a mean absolute distance of 0.077 of the teoretical calibration distance 0.

Mean distance plot in Figure 2 shows the minimum position of the calibration values, and a verification of a succeded calibration can be seen in the sample picture (Fig. 3) which were automatically captured with the calibration gains.

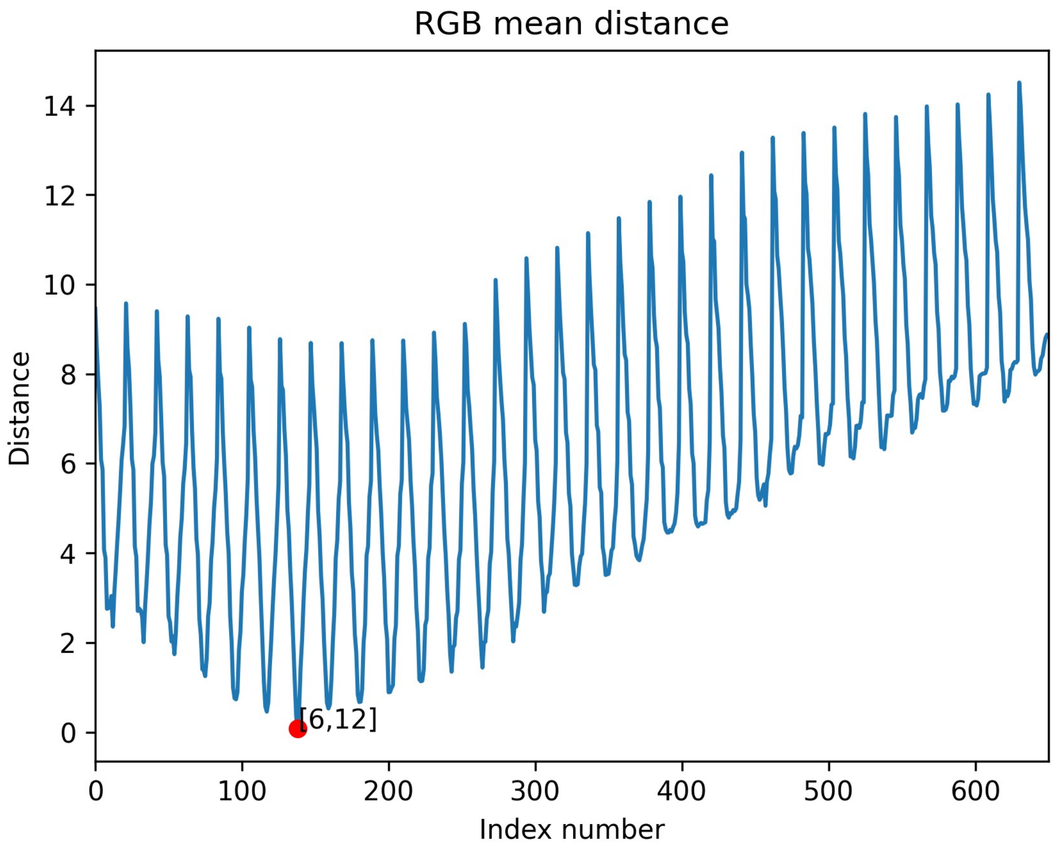

Figure 2. RGB mean distance of the calibration in this section.

Figure 3. Calibration sample image captured when rgain=3.26 and bgain=1.46.

5.2 Multi-stage Calibration

This method is preferred when very accurate gain values are required. After each stage, the steps are divided by 10. First stage is very rough when the ranges are 1-8 for both gains with step of 1.

A calibration ODF a white card gave rgain=3 and bgain=2 with distance of 35.187. The new ranges for stage 2 were set to:

R: 2-4, 0.1
B: 1-3, 0.1

=> rgain=3.2, bgain=1.5, dist=2.224

In stage 3 the ranges were narrowed to these ranges:

R: 3.1-3.3, 0.01
B: 1.4-1.6, 0.01

=> rgain=3.18, bgain=1.47, dist=0.302

Two additional steps were performed. The step was 0.0001 in the last step. The final valibration valus were measured to rgain=3.1733, bgain=1.4670 and dist=0.014.

The gains are shown as a function of performed calibration in Figure 4.

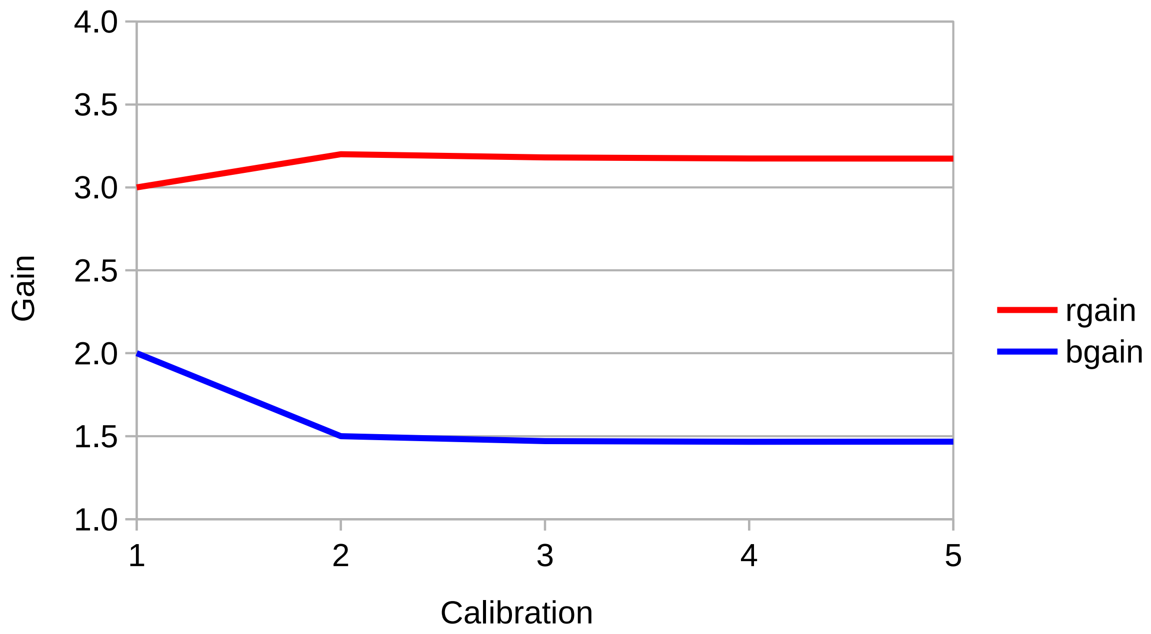

Figure 4. Red and blue gains as a function of calibration.

Better resolution can be obtained by performig a color analysis of the captured calibration images (Fig. 6). This is illustrated in Figure 5.

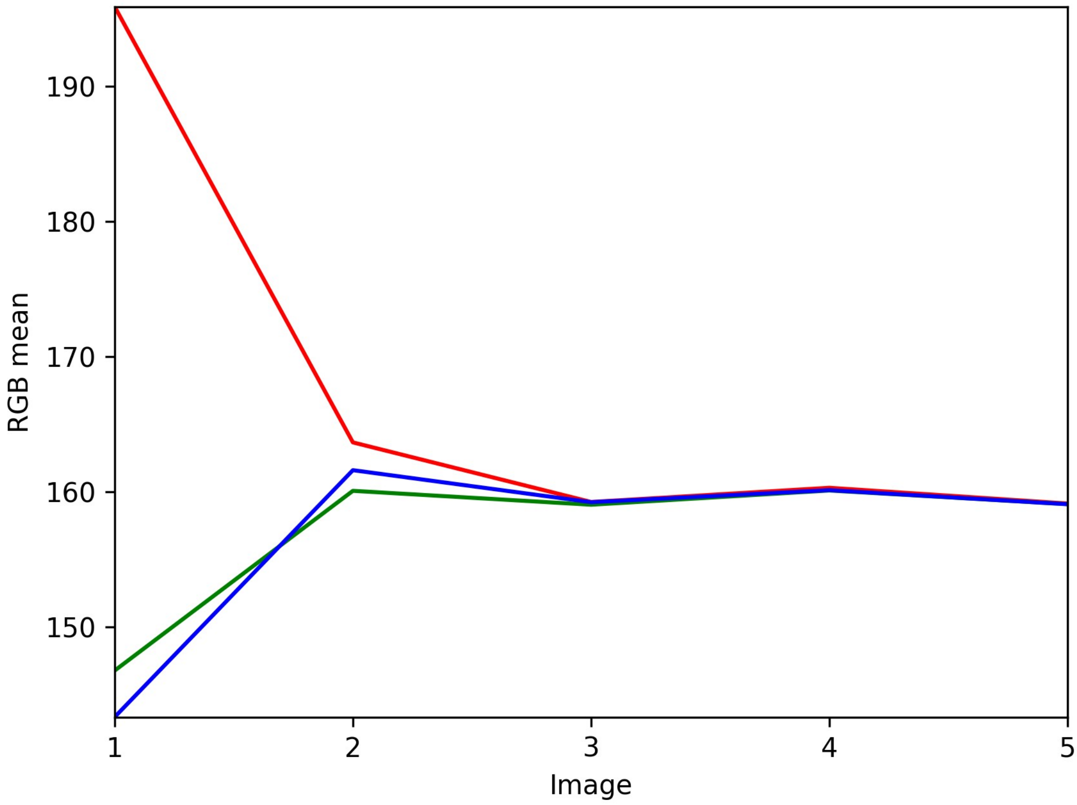

Figure 5. RGB mean values as a function of calibration.

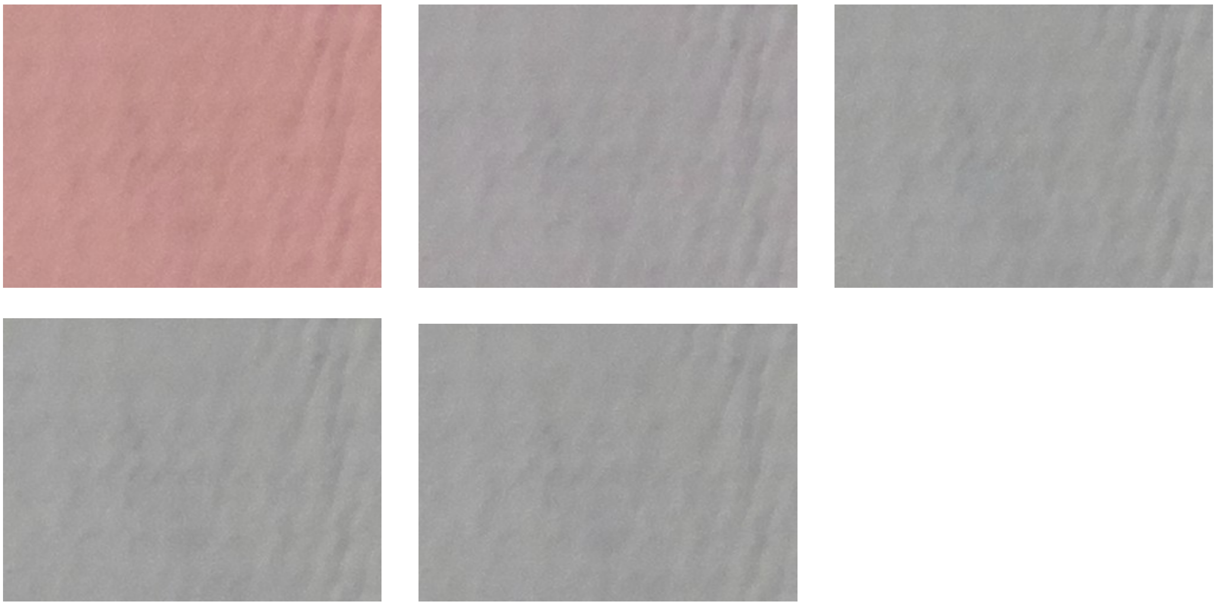

Figure 6. Sample images as a function of calibration.

RGB channel mean values of stage 2 are illustrated as raster images in Figure 7 and stage 5 in Figure 8.

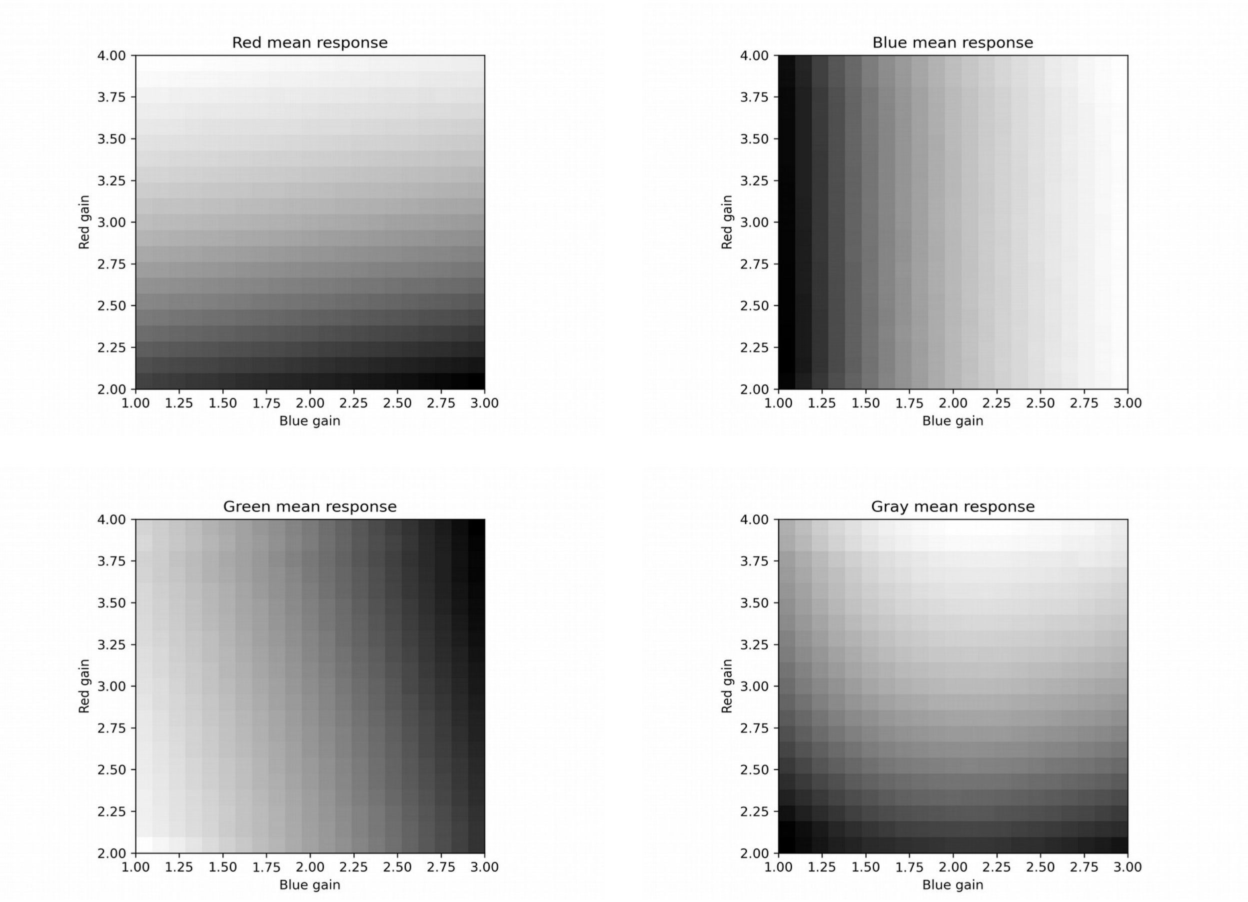

Figure 7. Stage 5 calibration - RGB mean channels.

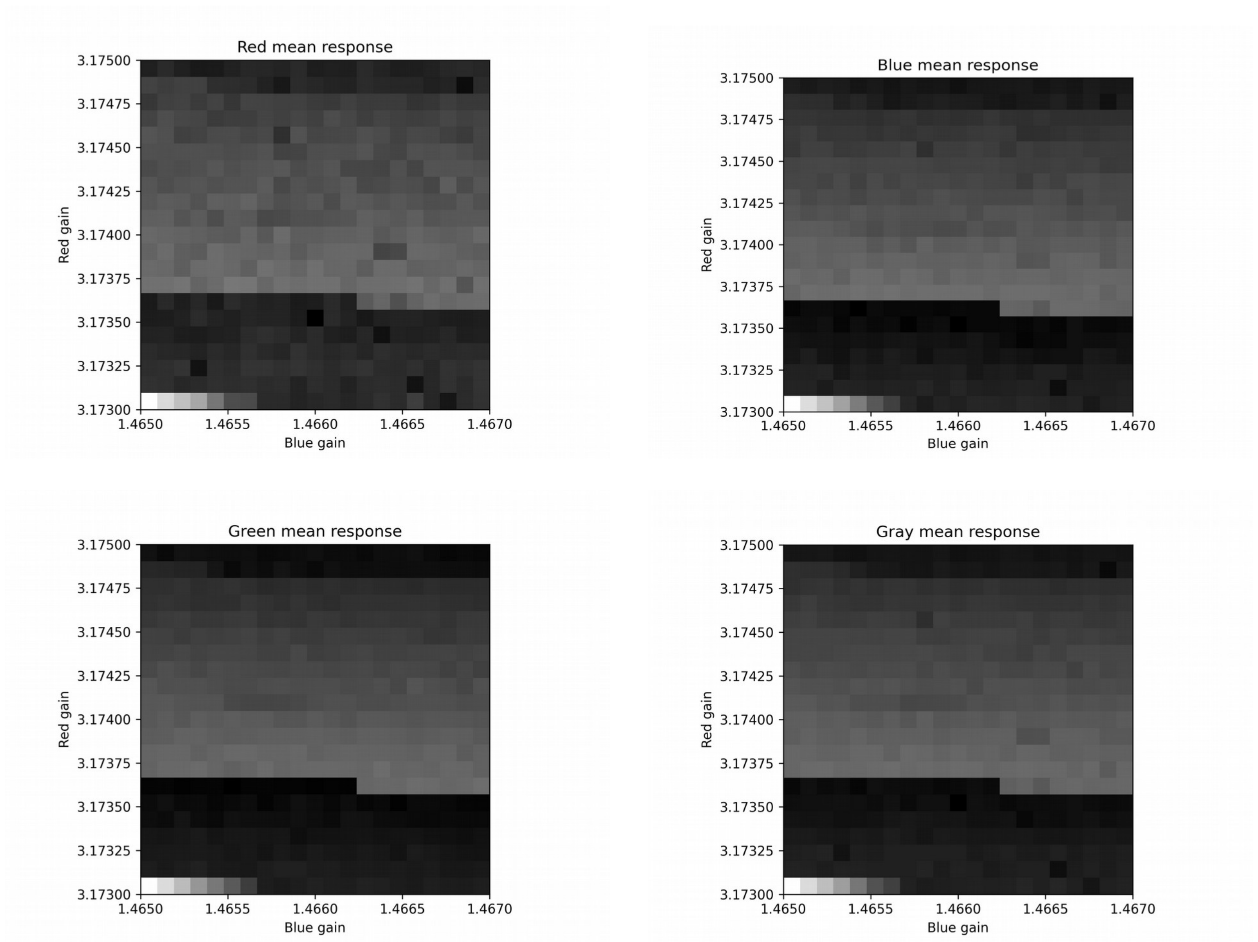

Figure 8. Stage 5 calibration - RGB mean channels.