



Master's Thesis Computer Science

Characterisation of a high luminance resolution display for psychophysical experiments

Submitted by: Jonas Schmiegel

30.11.2023

First Supervisor

Dr. Uli Wannek

Faculty of Science
Department of Computer Science
Neural Information Processing
University of Tübingen

Second Supervisor

Prof. Felix Wichmann, DPhil

Faculty of Science
Department of Computer Science
Neural Information Processing
University of Tübingen

Schmiegel, Jonas:

*Characterisation of a high luminance resolution display for
psychophysical experiments*

Master's Thesis Computer Science

University of Tübingen

Processing period: 31.05.2023 - 30.11.2023

Abstract

This thesis focuses on the characterization of high luminance resolution displays, specifically those from *VPixx Technologies Inc.*, which are critical for the accurate representation of visual stimuli in various fields and are used for psychophysical experiments in the **NIP** research group. The current calibration process suffers from measurement noise and is time-consuming. This necessitates the exploration of a statistical model to improve accuracy and efficiency.

Investigations reveal the significant influence of temperature on luminance values, emphasizing the need for a warm-up period and highlighting unique temperature dependencies among monitor models. The thesis presents a Gaussian process-based model for predicting and mapping luminance behavior that shows promise while also highlighting areas for improvement, particularly in addressing systematic error and saturation.

The proposed model offers a potential shift in current calibration procedures by suggesting a more streamlined approach. By combining temperature measurements with the model, a more efficient calibration process is envisioned, reducing the need for extensive measurements and expediting calibration.

In conclusion, this thesis provides comprehensive insights into the characterization of high luminance displays and proposes a novel model-driven approach that could revolutionize calibration procedures, enabling more efficient and accurate visual experiments in vision science.

Contents

Abstract	2
1 Introduction	5
1.1 Motivation	5
1.2 Outlook	6
2 Experimental Setup	8
2.1 Display Devices	8
2.1.1 VIEWPixx /3D Lite	8
2.1.2 VIEWPixx	8
2.1.3 Backlight Operating Modes	9
2.2 Measurement Devices	9
2.2.1 Photometer	9
2.2.2 optical transient tecorder (OTR)	9
2.3 Setups	10
2.3.1 Setup I: Absolute full screen luminance	10
2.3.2 Setup II: Amplitude comparison of modes	10
2.3.3 Setup III: Timing behavior of scanning mode	10
3 Current Situation	12
4 Characterization	14
4.1 Luminance as input (L_{in})	14
4.2 Backlight Mode	17
4.2.1 Procedure of the scanning Mode	17
4.2.2 Comparison of the scanning and the normal mode	20
4.3 Temperature	22
4.3.1 Warm-up	22
4.3.2 Effect on luminance characteristic	23
4.3.3 Effect of ambient temperature	25
4.4 Monitor	25
5 Model	29
5.1 Gaussian process (GP)	29
5.2 Gamma	31
5.3 Temperature	32
5.4 Combination of gamma dependency and temperature dependence	34
6 Validating the model with real data	36
6.1 Prediction of measurements	36

6.1.1	Training	36
6.1.2	Prediction	37
6.2	Linearization	38
7	Result and Discussion	41
7.1	Characterization	41
7.2	Model	42
8	Conclusion and Future Work	44
	Bibliography	48
	List of Abbreviations	49
	List of Figures	51
	List of Code	52
	List of Tables	53
	Appendix	54

1 *Introduction*

This chapter provides a overview on the thesis's motivation and offers an outlook on the forthcoming thesis.

1.1 *Motivation*

Display devices play an essential role in both, our daily lives and across various fields of experimental and clinical vision science. In these domains it is very important to ensure an accurate representation of stimuli from those displays. Only with an accurate representation it can be ensured, that research outcomes are reliable and correctly evaluated.

Concrete examples of these requirements can be found, in ophthalmology, visual neuroscience and psychophysics. In ophthalmology, for example, it is essential that visual objects are accurately represented for diagnostic purposes during clinical tests and trials. In visual neuroscience, the photometric properties of displays are important for visual stimuli to studying neural responses. Also, visual psychophysics relies on accurate and precise presentations of visual stimuli, requiring precise onsets, offsets, and interstimulus intervals in experimental paradigms such as rapid serial visual presentation, visual masking, and priming.

A represented visual stimulus not only needs precise control of space¹ and time², but as well in luminance³ or color. In the course of this thesis my focus will be on grayscale images and therefore on the luminance representation of a display.

This thesis is being conducted within the Neural Information Processing (**NIP**) research group led by Prof. Wichmann, which focuses on human perception and combines psychophysical experiments with computational modeling.⁴ The characterization of displays in this area of research can ensure the quality of the results obtained.

The relationship between the intensity of light and how bright it is perceived by a person is non-linear, and this perception also depends on the viewing conditions. Determining the function of brightness perception is therefore a major challenge. In general, however, a change in luminance is perceived more strongly in the dark areas than in the bright areas. The luminance function of the human perception therefore has a compressive nature, as its gradient is greater for darker values and becomes smaller as brightness

¹ This means e.g. the geometric of the display and the pixels.

² This means e.g. the refresh rate of the display. Elze and Tanner [4] and Feng, Pan, and Daly [5] for example, have made investigations on this.

³ Hereby, another important variable, contrast, is also under control.

⁴ For more information, refer to:
<http://wichmannlab.org/>

increases. Cameras and screens divide light into numerical values that are proportional to the amount of light. The resulting value must be quantified to a certain number of bits. If this mapping is made linear, the quantification error would be more visible in the dark areas than in the brighter areas due to the properties of the brightness function. With this linear mapping, the optimum number of bits would therefore be defined by the dark areas. Which would not be very effective. A non-linear transformation is therefore used to mimic the brightness function. One method for this non-linear transformation is the gamma correction, which simply corresponds to a power-law transformation⁵.[2, 22]

As mentioned above, in psychophysical experiments it is important to have as much control as possible over the visual stimuli. The **NIP** research group must therefore be able to characterize the luminance behavior of the monitor before an experiment and adapt it, if necessary, to a desired functional dependency.

Currently, the procedure for determining the luminance behavior is to measure all luminance values before each experiment. These measurements are then used to create a lookup table (**LUT**) with e.g. an inversion of the curve modeling the gamma correction. This method has a number of disadvantages.

Firstly, it does not take into account the fact that the individual measurements of luminance values are made with noise and may therefore show deviations that do not correspond to the true value. Therefore, one of the main objectives of this thesis will be to test if it is possible to use a statistical model⁶ to make the calibration process more resistant to noise.

Furthermore, the current luminance characterization process is time-consuming, often taking up to about 5 hours to measure luminance values for 12-bit resolution.⁷ A significant portion of this time is spent measuring each luminance value individually for the **LUT**. By improving the characterization, I aim to speed up the process in this thesis. For example, a statistical model with a lot of prior knowledge about the behavior of the monitor might need fewer points to find the luminance behavior of the monitor.

1.2 Outlook

The overall goal of this thesis is to improve the characterization of high luminance resolution displays, with a particular focus on the monitors by *VPixx Technologies Inc.* used in the lab of the **NIP** research group led by Prof. Wichmann. The VIEWPixed monitor is notable for its M16 mode, which offers not only the usual 8-bit luminance resolution but up to 16-bit.⁸ This allows to improve e.g. linear or gamma calibration or other stimuli designs. This thesis will focus on the VIEWPixed monitor as an exemplary case for a high luminance resolution display.

To lay the groundwork for this research, the thesis will begin with an overview about the experimental setup⁹ that will be used

⁵ Which looks as follows: $y = x^\gamma$
More on this in Chapter 4.1.

⁶ see Chapter 5

⁷ A log file of this process can be found in Appendix A.

⁸ Chapter 2.1 of the thesis will provide further information about the unique feature of the VIEWPixed monitor.

⁹ see Chapter 2

to characterize the monitors during this thesis. This is followed by an overview of the current calibration process¹⁰ and its limitations. Therefore, it will summarize the preliminary work on existing calibration methods done by David-Elias Künstle¹¹ in the psychopy-pixx library¹², the psychopy library [21], [20] done by Jonathan Österle and the PsychToolbox, supported by Mario Kleiner [15].

This is followed by a characterization of the monitors, showing which parameters influence the behavior of the monitor and thus cause changes in the display of luminance values. In the previous process, it is assumed that only the specified luminance as input (L_{in}) value is the cause of the resulting displayed luminance. Whether this is the case, and what other parameters may influence the displayed luminance, will be investigated in this Chapter 4.

Once this characterization has been completed, the findings will be used to create a statistical model that can be used to map the behavior of the monitor. This model will first be tested on artificial data¹³ and then checked again with real data. During the check, the model is to be tested to predict unknown luminance values and the gamma dependency is to be linearized.¹⁴

If this is successful, a process will be developed as to how the model can be used to replace the current process. At the end of the thesis, the results will be summarized and classified, and an outlook on future developments and research ideas is given.¹⁵

¹⁰ see Chapter 3

¹¹ <https://github.com/dekuenstle>

¹² <https://github.com/wichmann-lab/psychopy-pixx>

¹³ see Chapter 5

¹⁴ see Chapter 6

¹⁵ see Chapter 8 and Chapter 7

2 Experimental Setup

Throughout this thesis, numerous experiments were conducted to enable the characterization of a high luminance resolution display for psychophysical experiments. This chapter provides an overview of the equipment and setups utilized, distinguishing between display and measurement devices outlined in Chapter 2.1 and Chapter 2.2, correspondingly.

2.1 Display Devices

The experiments in this thesis use two monitors by *VPixx Technologies Inc.*¹ as representatives of high luminance resolution displays and characterize them accordingly. The two monitors are namely **VIEWPixx /3D Lite** and **VIEWPixx**.

¹ <https://vpixx.com/>

2.1.1 VIEWPixx /3D Lite

The VIEWPixx /3D Lite monitor (Model No.: VPX-VPX-2004B) provides a pixel resolution of 1920 (horizontal) \times 1080 (vertical) pixels. The display diagonal measures 24 inches (= 60.96 cm), and the pixel pitch lies at 0.2715 mm (horizontal) \times 0.2715 mm (vertical). As a result, the screen size comes to 29.32 cm (horizontal) \times 52.13 cm (vertical).² The VIEWPixx /3D Lite has a resolution of 10-bit per each RGB channel.[14]

The VIEWPixx /3D Lite monitor was acquired 2018 by the **NIP** research group.

² $0.2715 \text{ mm} \cdot 1920 = 521.28 \text{ mm} \approx 52.13 \text{ cm}$
 $0.2715 \text{ mm} \cdot 1080 = 293.22 \text{ mm} \approx 29.32 \text{ cm}$

2.1.2 VIEWPixx

The VIEWPixx monitor (Model No.: VPX-VPX-2001C) provides a pixel resolution of 1920 (horizontal) \times 1200 (vertical) pixels, with a screen diagonal of 22.5 inches (= 57.15 cm). The pixel pitch is 0.252 mm (horizontal) \times 0.252 mm (vertical), resulting in a screen size of 48.38 cm (horizontal) \times 30.24 cm (vertical).³ The resolution of each RGB of the channels of the VIEWPixx is 12-bit.[13]

The VIEWPixx monitor was acquired 2013 by the **NIP** research group.

³ $0.252 \text{ mm} \cdot 1920 = 483.84 \text{ mm} \approx 48.38 \text{ cm}$
 $0.252 \text{ mm} \cdot 1200 = 302.4 \text{ mm} \approx 30.24 \text{ cm}$

2.1.3 Backlight Operating Modes

One of the special features of the monitors by *VPixx Technologies Inc.* are the two different backlight operating modes. During the experiments, the scanning backlight mode and the normal backlight mode⁴ are used. In the scanning backlight mode the monitor can emit a maximum luminance of $100 \frac{cd}{m^2}$ and a gray-to-gray response time of 1 ms. In the normal backlight mode the maximum is increased to $250 \frac{cd}{m^2}$ with a response time of 7 ms. The consequences of these on the properties of the monitor are discussed in Chapter 4.2.

⁴ This mode is also called standard backlight mode are used.[14] Throughout this thesis, we will only differentiate between normal and scanning mode and refer with these terms to them.

2.2 Measurement Devices

For experiments and control measurements of the luminance characteristic and the temporal behavior, the **NIP** research group offered a **Photometer** and a **optical transient tecorder (OTR)**.

2.2.1 Photometer

A photometer⁵ is used to measure the luminance performance (in units of $\frac{cd}{m^2}$) of the monitor.⁶ The photometer is placed at a distance of 30 cm from the display and connected to the PC via a USB interface. Given the viewing angle of $\pm 6.5^\circ$ ⁷, the circular measurement area will have a diameter of $\approx 6.84 \text{ cm}^2$.⁸ Since the photometer is aligned with the center of the screen and the screen size is $27.22 \text{ cm} \times 48.38 \text{ cm}$, the measured area is only part of the actual screen area. The photometer is set to a sampling rate of 150 Hz and an amplification of 10^6 [13, 27, 30].

⁵ UDT Model 211 by *Gamma Scientific Inc.*

⁶ For additional information, see the data sheet: [30].

⁷ Validated through own measurements.

⁸ diameter of measured area:
 $2 \cdot \tan(6.5^\circ) \cdot 30 \text{ cm} \approx 6.84 \text{ cm}$

2.2.2 optical transient tecorder (OTR)

The optical transient tecorder (**OTR**) is capable of “recording, storage and evaluation of temporal variations of [...] luminance”[3, p. 2]. I utilize the **OTR** during this thesis to investigate the procedure of the back panel illumination during the scanning mode in Chapter 4.2.1. The OTR-3 utilized for this thesis was manufacturer by *Display-Messtechnik & Systeme GmbH & Co. KG* and consists of a collimating receiver and an electrical box. The electrical box consists of a processing pipeline of a low-pass filter, an amplifier with variable gain, a 16 bit A/D converter and a controller with a USB interface⁹.[3, p. 2]

The receiver combines a collimating lens and a Si-photodiode which has an adapted spectral sensitivity $V(\lambda)$. The maximum sampling frequency is 100.000 Hz.[3, p. 6]

The OTR-3 software allows measurement control, visualization of the resulting luminance time trace and data management. In this thesis’ research, version 1.1 of the software was utilized.

⁹ USB-2.0

2.3 Setups

During this thesis three experimental setups were developed and will be discussed in the following chapters (Chapter 2.3.1, Chapter 2.3.2 and Chapter 2.3.3), each mention of an experiment is accompanied by a footnote¹⁰ that offers more detailed information about the experiment and its parameters.

Monitor pertains to one of the monitors utilized in this thesis.¹¹

Experiment setup is a reference to one of the three setups used for the experiment, as described in the following sections.

Mode corresponds to the selected backlight mode. Either scanning or normal.¹²

Number of L_{in} specifies how many different L_{in} are considered. More information about L_{in} can be found in Chapter 4.1.

Temperature refers to the internally measured temperature of the monitor throughout the experiment.¹³ The temperature is represented by two methods. The first one is denoted by an \varnothing sign, which signifies the average temperature during the experiment (e.g. $\varnothing 44^\circ\text{C}$). The second method displays a temperature range (e.g. $34 - 45^\circ\text{C}$).

The **Distance** parameter represents the distance from the measurement device to the monitor. It was determined prior to conducting experiments using a digital laser rangefinder¹⁴, with a measuring accuracy of "typically" $\pm 2.0\text{ mm}$.[25, p. 16]

Raw data refers to the original raw data of the experiment. See appendix.

2.3.1 Setup I: Absolute full screen luminance

In this setup, the photometer was positioned approximately 30 cm away from the monitor and aligned with the center of the display. The UDT Model 211 photometer was used. Note that the specific monitor utilized may differ in this particular setup. Please refer to Figure 2.1 for a schematic representation of the setup.

This setup is used to compare the absolute luminance of the full screen in different experiments.

2.3.2 Setup II: Amplitude comparison of modes

This setup is the same as the **Setup I**, but the photometer is replaced by the OTR¹⁵ sensor with a measuring-field diameter of 25 mm. A schematic representation of the setup can be found in Figure 2.2.

This setup is used to compare the amplitude of the different modes.¹⁶

2.3.3 Setup III: Timing behavior of scanning mode

In order to achieve a better vertical resolution, the OTR¹⁶ sensor head was placed in this setup almost direct (less than 0.2 mm) on

¹⁰ Monitor:
Experiment setup:
Mode:
Number of L_{in} :
Temperature:
Distance:
Raw data:

¹¹ Further details regarding the impact of this parameter can be found in Chapter 4.4.

¹² see Chapter 4.2

¹³ see Chapter 4.3

¹⁴ Bosch PLR 25 (3 603 K16 200)

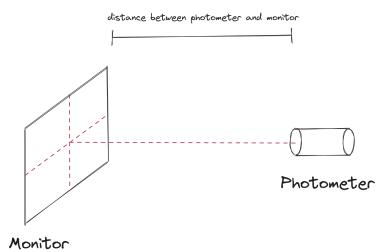


Figure 2.1: Schematic representation of the **Setup I**.

¹⁵ see Chapter 4.2.2

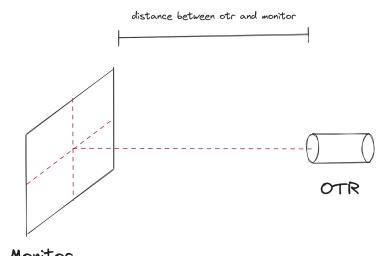


Figure 2.2: Schematic representation of the **Setup II**.

¹⁶ see Chapter 4.2.2

the display surface¹⁷, with a measuring-field diameter of ≈ 2.3 mm. Figure 2.3 provides a schematic representation of the setup.

This setup is used to investigate the timing behavior of the scanning mode.¹⁸

¹⁷ A A4 paper (70 g/m^2) was used to "measure" the distance to the display.

¹⁸ see Chapter 4.2.1

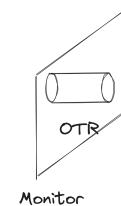


Figure 2.3: Schematic representation of the [Setup III](#).

3 Current Situation

For psychophysical experiments of e.g. human contrast sensitivity, stimuli need to be displayed with a high accuracy and resolution of luminance. The displays by *VPixx Technologies Inc.* are designed (amongst other useful properties) for a luminance resolution of up to 12 bit luminance.¹ Due to their high luminance resolution these devices are widely used by psychophysics studies/labs around the world, e.g. [24], [16] and [17].

The **NIP** research group also has monitors by *VPixx Technologies Inc.* in its psychophysical laboratory and uses them for various psychophysical experiments. A wide variety of psychophysical experiments are carried out on these monitors. For example, [7] compared human and machine visual perception and made about 85,000 psychophysical trials across 90 participants. Or [10], which showed that the ordinal embedding approaches allows obtaining reasonable scaling functions from fewer judgments and also allows estimating multidimensional perceptual scales. These experiments are not always directly related to the high luminance resolution of these monitors. Nevertheless, a wrong display of luminance can suppress the actual effect that one wants to investigate with their experiment. It is therefore necessary for a monitor to be accurately calibrated.

The full resolution standard calibration used so far takes approximately 5 hours. The resulting **LUT** consists of an inverted spline fit to $2^{12} = 4096$ measured values. This calibration routine is provided as `psychopy-pixx` python package² in the **NIP** lab and is used in experiments performed under Psychtoolbox. For experiments performed under `psychopy`³, no suitable calibration routine exists, as [20] evaluated. Therefor an attempt was made by David-Elias Künstle⁴ to implement a (linear) calibration.

The current process of this linearization is that first a measurement of all possible L_{in} values is made. This measurement takes ≈ 5 hours. The measured curve is then taken and inverted. The inversion is carried out in such a way that the L_{out} values are normalized min-max first. Since an inversion only works with monotonically increasing functions, the next step is to check whether the values are monotonically increasing and if this is not the case, the corresponding values are discarded. The two axes are then swapped with each other. To ensure that the new X values are in the same position as before the two axes were swapped, interpo-

¹ see Chapter 2.1

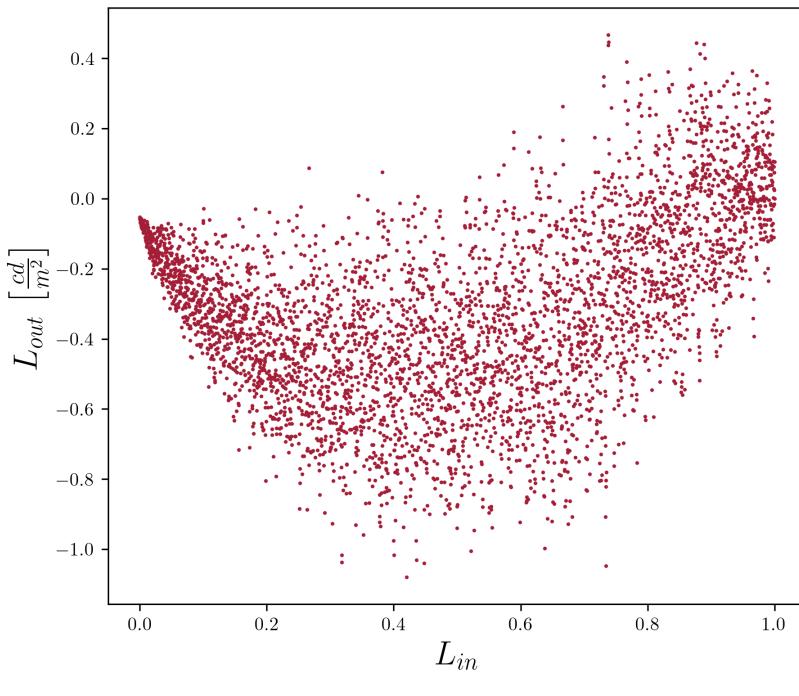
² The package can be found under: <https://pypi.org/project/psychopy-pixx/>
The source code can be found under: <https://github.com/wichmann-lab/psychopy-pixx>

³ <https://www.psychopy.org/>

⁴ <https://github.com/dekuenstle>

lation must be performed between the new Y values. This is done with the method `np.interp()`⁵, which uses splines and offers a one-dimensional linear interpolation for monotonically increasing sample points.⁶ The resulting inverted curve is then saved and applied to the monitor object via a **LUT**. If experiments are now carried out on the monitor, they show a linearized luminance curve.

Figure 3.1 and Figure 3.2 depict the outcome of the linearization process⁷. While Figure 3.2 displays the measured and ideal linearization, Figure 3.1 exhibits the residuals of the measured linearization as compared to the ideal linearization. The residuals have an overall distribution of $\mu \approx -0.32$ and $\sigma \approx 0.26$.



⁵ <https://numpy.org/doc/stable/reference/generated/numpy.interp.html>

⁶ The source code of the described linearization can be found in:
https://github.com/wichmann-lab/psychopy-pixx/blob/main/psychopy_pixx/devices/viewpixx.py starting at line 221.

⁷ Monitor: VIEWPrix /3D Lite

Experiment setup: *Setup I*

Mode: *normal*

Number of L_{in} : 21²

Temperature: 44.125 °C

Distance: 30 cm

Raw data: Appendix F.1

Figure 3.1: Residues of the measured linearized luminance and the ideal linear function, after linear calibration using the current process.

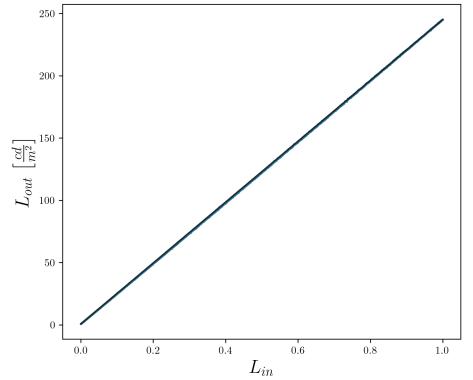


Figure 3.2: Measured and ideal linearization, when using the current process in (almost) perfect alignment. For the Residuals see Figure 3.1.

● ideal linear function

● measured luminance after applying the **LUT**

4 Characterization

This chapter focuses on characterizing the VIEWPixx monitor as a representative of the high luminance resolution displays for psychophysical experiments. Four parameters that significantly affect the displayed luminance value of the monitor are identified during this characterization. While the expected behavior and influence of the first two parameters (input value L_{in} ¹ and backlight mode²) are evident, the nature of the other two parameters (internal temperature³ and monitor model⁴) on the other hand deserve attention.

¹ see Chapter 4.1

² see Chapter 4.2

³ see Chapter 4.3

⁴ see Chapter 4.4

4.1 Luminance as input (L_{in})

Full screen luminance as input (L_{in}) serves for controlling the monitor's displayed luminance value. The value of L_{in} is represented by a value ranging from 0.0 to 1.0 and corresponds to a percentage scale. The predetermined luminance values are transmitted to the three RGB-channels. Code 4.1 demonstrates how this process occurs. However, it should be noted that this code is a reduced and simplified version. The original code used for this implementation can be found in Appendix B.

```
1 | for lum in luminances:  
2 |     rgb = [lum, lum, lum]  
3 |     testPatch.setColor(rgb)
```

Code 4.1: Simplified and reduced code to set the luminance of the monitor.

In Figure 4.1, an experiment⁵ is presented that investigates the relationship between the setting of L_{in} and the corresponding measurement of luminance by a photometer.

In addition to the measured luminance, the figure includes a curve that corresponds to a gamma dependency of 2.2. This gamma dependency is compared due to the fact, that *Vpixx Technologies Inc.* claims in [31] that their monitor has a "native gamma very close to 2.2"[31, p. 102]

A gamma correction of 2.2 for normalized luminance output corresponds to the following mathematical relationship.[22, pp. 257-280]

⁵ Monitor: VIEWPixx /3D Lite

Experiment setup: *Setup I*

Mode: *normal*

Number of L_{in} : 2¹²

Temperature: Ø 35 °C

Distance: 30 cm

Raw data: *Appendix F.2*

$$L_{out} = (L_{in})^\gamma = (L_{in})^{2.2}$$

Since this mathematical relationship only produces values between 0 and 1, in order to make it comparable to the measured

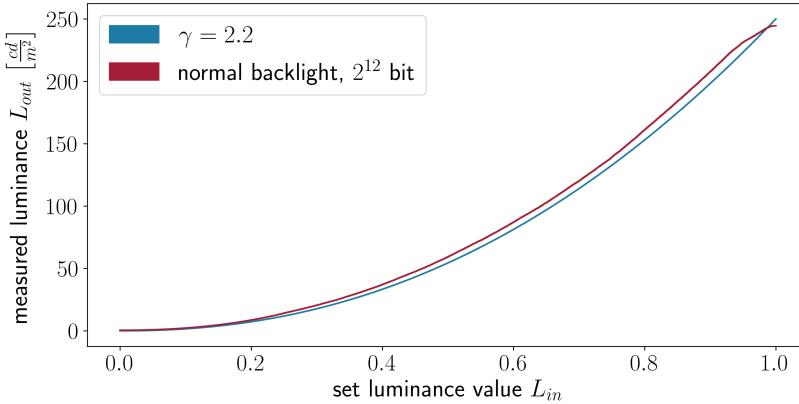


Figure 4.1: Comparison of measured luminance to an ideal gamma dependency corresponding to $\gamma = 2.2$.

luminance curve, it must be scaled as follows.

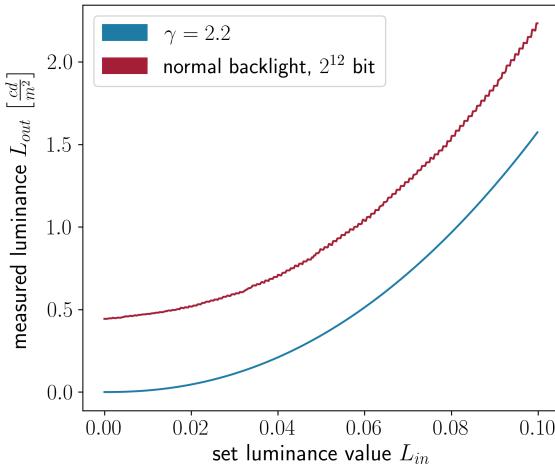
$$L_{out} = L_{max} \cdot L_{in}^{2.2}$$

L_{max} represents the maximum luminance level that the monitor should be capable of. In the normal mode this would be $250 \frac{cd}{m^2}$, according to the data sheet.⁶

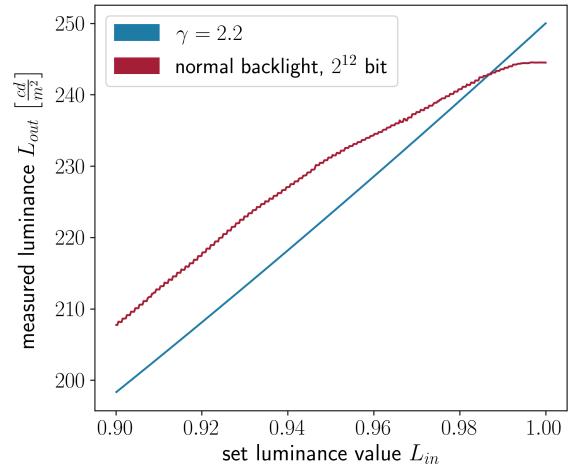
⁶ see Chapter 2.1

Figure 4.1 illustrates that the two curves are distinct, but “very close” to a gamma dependency. But for a high accuracy measurement this means that one must improve the “very close”. This will be done with an appropriate calibration.

In order to get a feeling for the current relationship between the L_{in} and the resulting luminance, the lower 10 percent and the upper 10 percent of the curves are examined more closely in Figure 4.2.



(a) Detailed view of the lower 10 %.



(b) Detailed view of the upper 10 %.

In the detailed views of Figure 4.2, three effects can be observed, that explain the deviation of the two curves.

Figure 4.2: Detailed view of the lower (a) and upper (b) 10 % of the gamma correction of 2.2 and the measured luminance values.

LOWEST LUMINANCE $\neq 0 \frac{cd}{m^2}$: Figure 4.2 (a) illustrates, that when $L_{in} = 0.0$ the measured luminance is $0.44 \frac{cd}{m^2}$, not $0 \frac{cd}{m^2}$. This effect

can be explained by the fact that “closed” pixels of liquid crystal display (**LCD**) does not succeed in blocking the entire backlight which is constantly on.[1, p. 305] Therefore, the lowest possible luminance must be considered in every experiment and can mathematically be described by an offset of (0.44) that is added to the gamma dependency.⁷

HIGHEST LUMINANCE $\neq 250 \frac{cd}{m^2}$: The second effect can be observed in Figure 4.2 (b). The maximum luminance value, set with $L_{in} = 1.0$ is not the expected $250 \frac{cd}{m^2}$ ⁸ but approximately only $245 \frac{cd}{m^2}$. This observation pertains solely to our specific measurement setup. Several factors may have led to the absolute deviation discovered during our experiment. These factors include for example systematic errors in the measuring device or distance between the photometer and monitor, which in turn have altered the angle dependence slightly. However, these factors were not verified as it was also discovered in the course of the investigations for this master thesis that the max luminance value is temperature-dependent.⁹ As it is not possible to display more luminance than the measured max value, a “worst-case-max” must be used. To do this, the multiplication factor in this example must be corrected from 250 to 245.¹⁰

MEASURED LUMINANCE IS SATURATED: Figure 4.2 (b) demonstrates, that the measured luminance does not increase monotonically as intended by the gamma dependency. Instead, the curve of the measured values exhibits an inflection point at approximately 0.93 and approaches saturation at around 0.99.¹¹ To correct this effect, it is necessary to adjust the x-axis of the gamma function as we are not concerned with the points at saturation. This can be achieved by dividing x by the factor of 0.97.¹²

All of these adjustments lead to the subsequent mathematical relationship used to model the gamma dependency of 2.2 on the measured luminance values.

$$L_{out} = 0.44 + 245 \cdot \left(\frac{L_{in}}{0.97} \right)^{2.2}$$

or more generally

$$L_{out} = \alpha + \beta \cdot L_{in}^{2.2}$$

Here α represents the low offset, β represents the combined high offset and the saturation offset.

This correction of the mathematical relationship results into the Figure 4.3.

The newly-formed curve closely resembles the intended curve of $\gamma = 2.2$. However, when analyzing specific segments of the curve such as the middle (approximately between 0.2 and 0.7), discrepancies from the measured curve are still noticeable. Hence,

⁷ This results in a corrected mathematical relationship of $L_{out} = 0.44 + (250 \cdot L_{in})^{2.2}$.

⁸ see Chapter 2.1

⁹ see Chapter 4.3

¹⁰ This results in a corrected mathematical relationship of $L_{out} = 0.44 + (245 \cdot L_{in})^{2.2}$.

¹¹ There is no clear reason or explanation (unlike for the other two effects) for why this saturation effect should occur. It is assumed that VPixx Technologies Inc. is enforcing saturation to guarantee the corresponding maximum luminance.

¹² In principle, there is currently no justification for this factor. However, it is worth noting that this point is situated approximately in between the two identified points, namely the approximate inflection point and the approximate time at which the saturation was observed. It was chosen after a trial and error process to obtain the best fit between the measured curve and the “corrected” gamma curve.

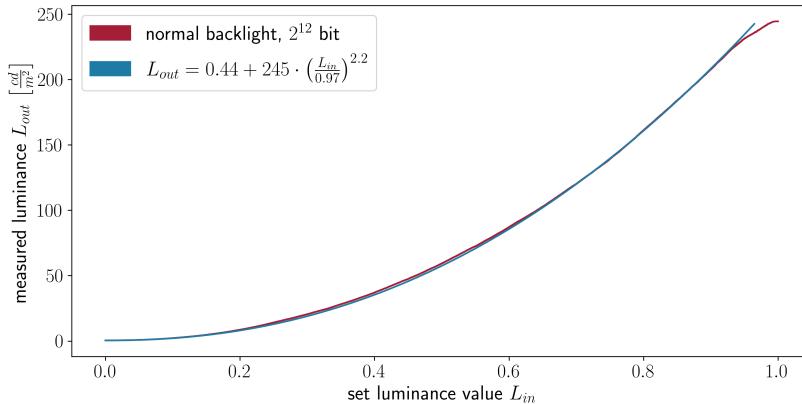


Figure 4.3: Comparison of measured luminance and the “corrected” gamma dependency that correspond to $\gamma = 2.2$.

it can be concluded that the measurements only loosely conform to the gamma dependency of 2.2. This aligns with the imprecise nature of *VPixx Technologies Inc.*’s statement¹³.

¹³ “gamma very close to 2.2”[31, p. 102]

4.2 Backlight Mode

As noted in Chapter 2.1.3, the monitors manufactured by *VPixx Technologies Inc.* offer two distinct backlight modes. The scanning backlight mode and the normal backlight mode.¹⁴ The scanning mode is a distinct feature provided by *VPixx Technologies Inc.* and was a factor in the decision of the NIP research group to use *VPixx Technologies Inc.* monitors.

Nevertheless, the NIP research group utilizes both modes, the normal has the advantage of a brighter display, while the scanning mode has the more accurate timing behavior.¹⁵ Therefore, both modes are pivotal for characterization purposes.

During the characterization, differences and similarities of the two modes were observed. It was discovered that the selected mode is one of four parameters that impact display luminance. Therefore, we will elucidate the precise scanning mode procedure¹⁶ and subsequently compare the two modes¹⁷.

4.2.1 Procedure of the scanning Mode

cathode ray tube (CRT) were the former standard for psychophysical experiments, due to their temporal accuracy.[9, p. 1] *VPixx Technologies Inc.* actively markets its monitors for their scanning mode, which resembles close to the timing behavior of a CRT display¹⁸. Although there is limited information provided in the monitors’ data sheet, about the changes of the backlight, particularly in terms of the exact timing, in the scanning mode. The only information provided is a description which states that “the backlight LEDs [of the monitor] are powered only within a narrow horizontal bar, and [that] the bar is scanned [then] from the top of the display to the bottom”[13, p. 20], during the scanning mode. Additionally, the mode is illustrated through the following Figure 4.4.

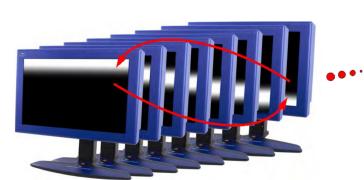
¹⁴ The normal backlight mode is also called standard backlight mode.[14] Throughout this thesis, we will only differentiate between normal and scanning mode and refer with these terms to them.

¹⁵ see Chapter 2.1.3

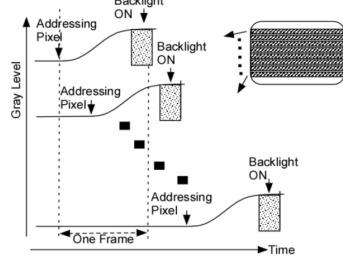
¹⁶ see Chapter 4.2.1.

¹⁷ see Chapter 4.2.2.

¹⁸ see <https://vpixx.com/hardware/crt-replacement/>



(a) Scanning backlight mode (visual example).



(b) Scanning backlight mode (time graph).

Figure 4.4 (a) provided a basic overview of the described scanning mode. Figure 4.4 (b) shows more technically details through a timing graph. This graph supports VPixx Technologies Inc.'s statement that the “scanning [mode] is synchronized to the LCD pixel updates”[13, p. 20]. This has the positive effect, that in the scanning mode much of the liquid crystal rise and fall time is hidden, resulting in an improved display of dynamic visual stimuli.[13, p. 20]

During the work on this thesis, the reason for the difference and exact ration in luminance between the two modes came to interest. Since the provided data sheet gave not enough advice, measurements were conducted about the timing behavior in the scanning mode. The research was aided by an OTR¹⁹ and followed the Setup III outlined in Chapter 2.3.3.

Since the monitor operates²⁰ at a refresh rate of 120 Hz, one cycle of the scanning mode must be completed in $\frac{1}{120}$ s = 8.3 ms. To determine when each horizontal LED bar of the monitor is activated, the OTR was placed at varying heights relative to the monitor and recorded 10 ms. Figure 4.5 provides an approximate overview of the OTR's placements in relation to the different LED bars of the monitor.

Since it was not feasible to synchronize the individual measurements²¹, they had to be slightly adjusted to correspond with their placements and the different phases of the scanning mode. Nevertheless, by properly shifting the individual measurements, Figure 4.6 was achieved, displaying 10 phases (of varying length) within one cycle of the scanning mode.²²

Using this analysis and the (relative) amplitude of each phase, it was possible to obtain an overview (Figure 4.7) of the active LED bars on the monitor during each phase of one cycle of the scanning mode.

As indicated in Figure 4.7 we observed in our analysis that typically only one LED bar is active during one phase, as seen in phase 1 and 3-9. Nevertheless, some phases have two LED bars active simultaneously. These phases are 2 and 10.²³

Figure 4.4: Visual example (a) and time graph (b) of the scanning backlight mode from the manual of the VIEWPixa monitor.[13, pp. 20-21]

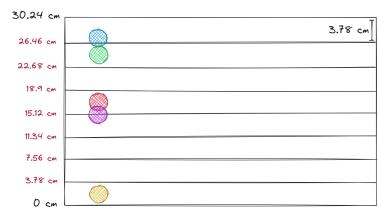


Figure 4.5: Overview of the OTR's placements in relation to the different light emitting diode (LED) bars of the monitor.

¹⁹ see Chapter 2.2.2.

²⁰ In our setup.

²¹ See Appendix C for the individual obtained measurements

²² The individual plots of the measurements can be found in Appendix C.1 to Appendix C.5.

²³ The existence of phases with two active LED bars, like for example phase 10, were verified via a photo captured with an exposure time of 1/4000 s = 0.25 ms (Figure 4.8). Unfortunately, phase 2 could not be captured and confirmed a second time, with this method because the exposure time is 2.5 times longer than the phase duration (0.1 ms).

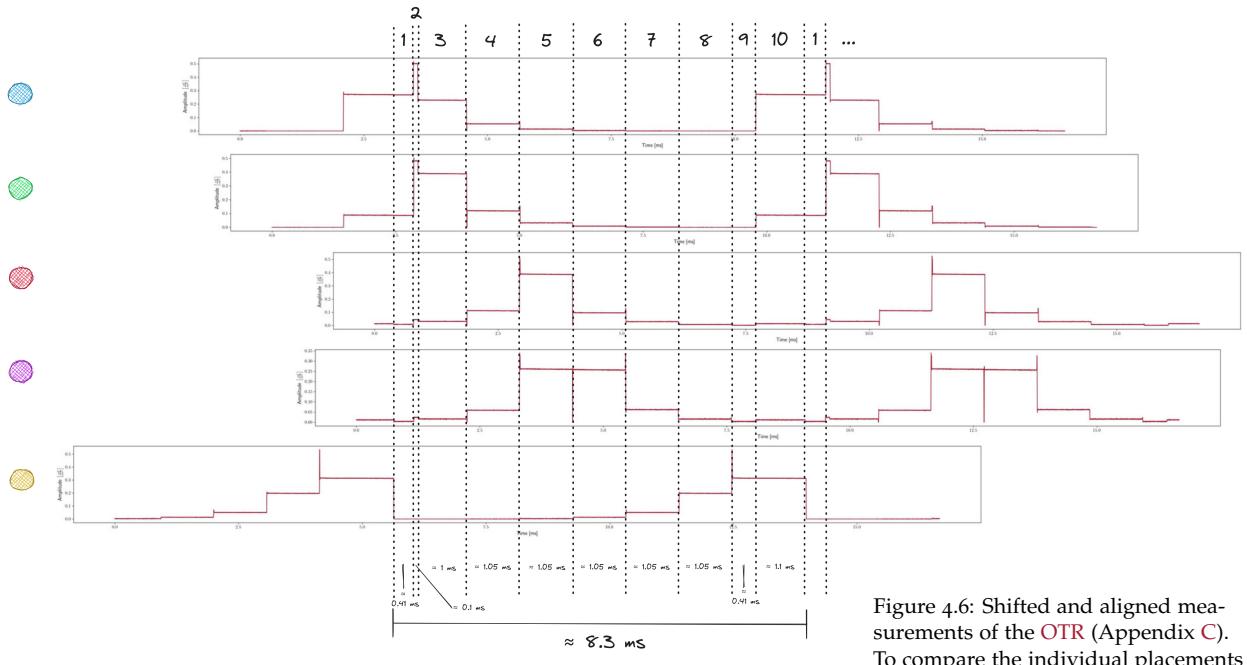


Figure 4.6: Shifted and aligned measurements of the OTR (Appendix C). To compare the individual placements (Figure 4.5) of the measurements.

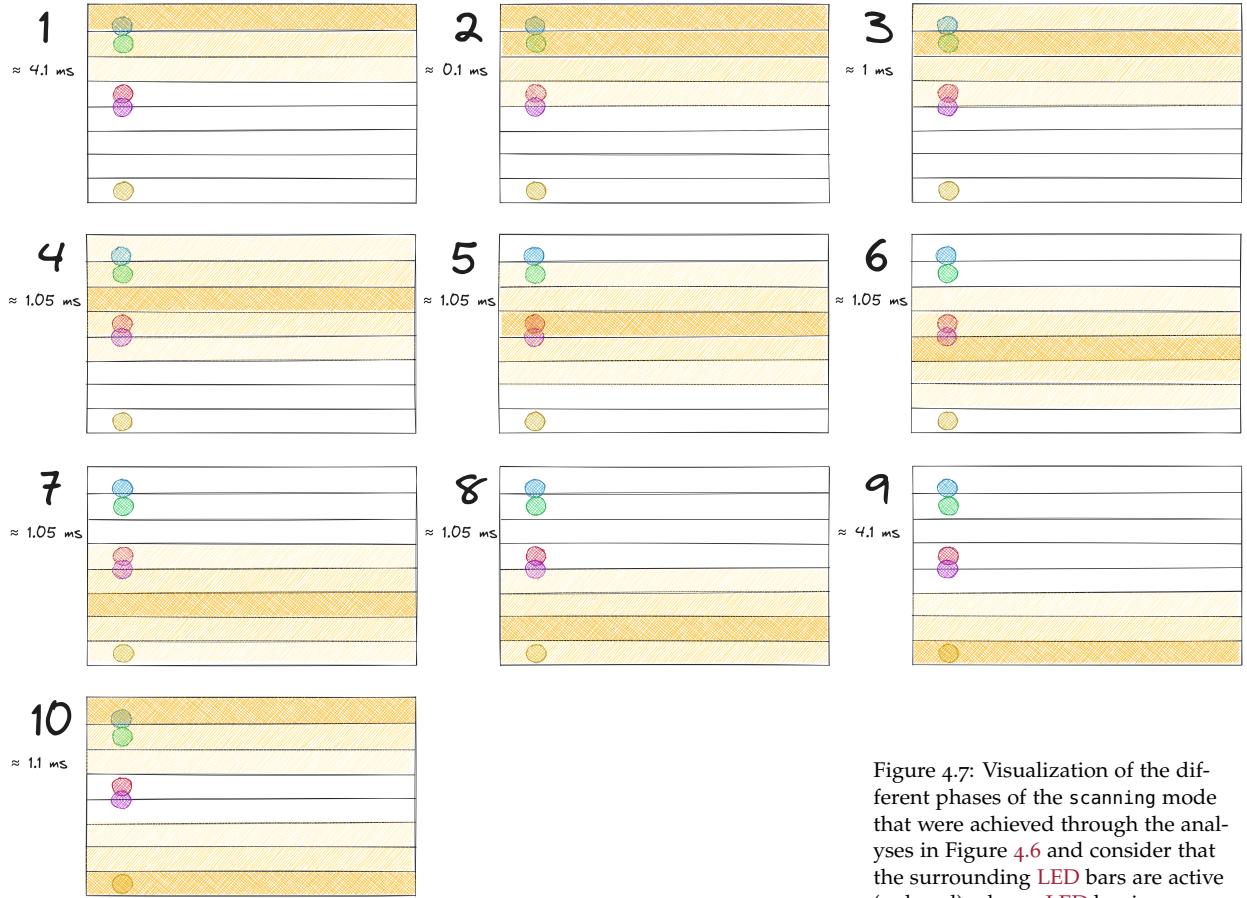


Figure 4.7: Visualization of the different phases of the scanning mode that were achieved through the analyses in Figure 4.6 and consider that the surrounding LED bars are active (reduced) when a LED bar is on.

What is apparent in both the photo²⁴ and the measurements²⁵, is that the visualization from *VPixx Technologies Inc.*²⁶ does not fully depict the lighting. Instead of a single LED bar being active while all others remain completely off, when one LED bar is on, the LED bars surrounding it are also active with decreased brightness. This is confirmed by the clear gradations in the measurements, which indicates when the LED bars adjacent to the targeted LED bar(s) are active. Since the **Setup III** is used, it was assumed that only the targeted LED bar(s) would generate an amplitude. However, this is not the case.²⁷ Instead, the measurement provides information on the activity of neighboring LED bars, which contributes to an overall coherent picture of the phases as shown in the analysis / Figure 4.6.

Therefore, I indicate in Figure 4.7 that the surrounding LED bars also light up (with less intensity) when one LED bar is active.²⁸

4.2.2 Comparison of the scanning and the normal mode

With the knowledge gained from Chapter 4.2.1 and the insight that the normal and the scanning modes might differ only in the number of simultaneously active LED bars, I assumed that the gamma values of the two can be offset. This would be advantageous because the normal mode exhibits much less noise compared to the scanning mode, when measured.

It is evident that there is a notable deviation in the measured values between normal mode and scanning mode. In normal mode, the maximal standard deviation is only $\hat{\sigma} \approx 0.1$, whereas in scanning mode, it is $\hat{\sigma} \approx 28.4$.²⁹ This disparity can be attributed to the device's nature in scanning mode where parts of the backlight frequently turn on and off, unlike in normal mode where the backlight remains constant.³⁰

This fact - that the normal mode produces less noise between individual L_{in} measurements - suggest that the two modes be offset against each other. Fewer measurements per L_{in} would be required, but still allow the use of both modes with a single calibration.

In order to see whether the two modes can be offset against each other, two approaches were made. One approach compared the maximum amplitude of an OTR measurement (see Chapter 4.2.2). The other approach compares two complete curves derived from two experiments (see Chapter 4.2.2).

Comparing the amplitude

The objective of this approach is to check if the scanning and the normal mode can be converted into each other by comparing the amplitude in the OTR of the two modes. An experiment³¹ was conducted in the **Setup II** to measure the amplitude of both modes for a duration of 0.1 s.

While the measurement of the normal mode shows a simple

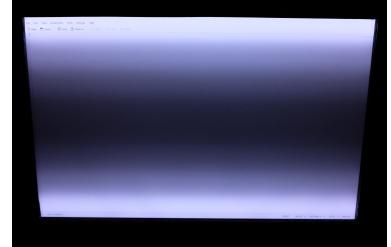


Figure 4.8: Photo, that shows phase 10 (see Figure 4.6). The photo was taken using a Fujifilm X100F and the following settings: ISO = 3200, focal length = 23 mm, f-stop = f/2, exposure time = 1/4000 s

²⁴ see Figure 4.8

²⁵ see Figure 4.6

²⁶ see Figure 4.4 (a)

²⁷ Appendix D gives information of how it was ensured that the observed effects are not optical scattering between the individual LED bars.

²⁸ Figure 3 is only intended to provide a schematic intuition for this. Almost all phases are visible in all measurements (except those at the bottom and top of the monitor) (see analysis and Figure 4.6). This means that almost all bars are active in every phase.

²⁹ Refer to Appendix E for more insights on this.

³⁰ see Chapter 4.2.1

³¹ Monitor: VIEWPixx

Experiment setup: **Setup II**

Mode: scanning and normal

Temperature: 35 °C

Distance: < 0.2 mm

Raw data: Appendix F.5

(noisy) constant ($\approx 0.76 \frac{cd}{m^2}$), the scanning mode oscillate between a maximum ($\approx 0.76 \frac{cd}{m^2}$) and a minimum ($\approx 0.02 \frac{cd}{m^2}$).

To compare the two amplitudes, we calculate their integrals and then find the ratio between them. As these are merely measurements and not functions, we assume that the sum of the measured values approximates the integral. To ensure comparability, both measurements were taken with the same number of measuring points (10,000), resulting in the following calculation.

$$\begin{aligned} \frac{\int_{0}^{10\text{ s}} \text{measurement_scanning}(x) dx}{\int_{0}^{10\text{ s}} \text{measurement_normal}(x) dx} &\approx \\ \frac{\sum_{0}^{10\text{ s}} \text{measurement_scanning}(x)}{\sum_{0}^{10\text{ s}} \text{measurement_normal}(x)} &\approx \frac{2910.03}{7576.88} \\ &\approx 0.38 \end{aligned}$$

This yields a factor of 0.38.

The upcoming analysis³² will reaffirm this factor and demonstrate that the two modes can offset each other by this amount.

Comparing two curves

The motivation of this approach is to examine whether the scanning and the normal modes can be offset against each other by comparing two curves from separate experiments. Two experiments were conducted with the same monitor³³, with identical average temperatures³⁴. The first experiment³⁵ resulted into the normal curve in Figure 4.11. The second experiment³⁶ resulted into the scanning curve in Figure 4.11. Both curves maintained a temperature of 35.125 °C.

By calculating the ratio between the two curves for each measuring point, we obtained an average factor of 0.38. Applying this factor to the normal curve demonstrates, that the normal and scanning modes differ only in the number and process of activated LED bars, not in their gamma behavior. After the “correction”, the both curves are nearly identical (Figure 4.10).

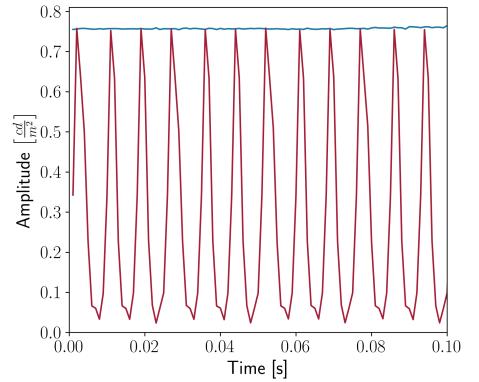
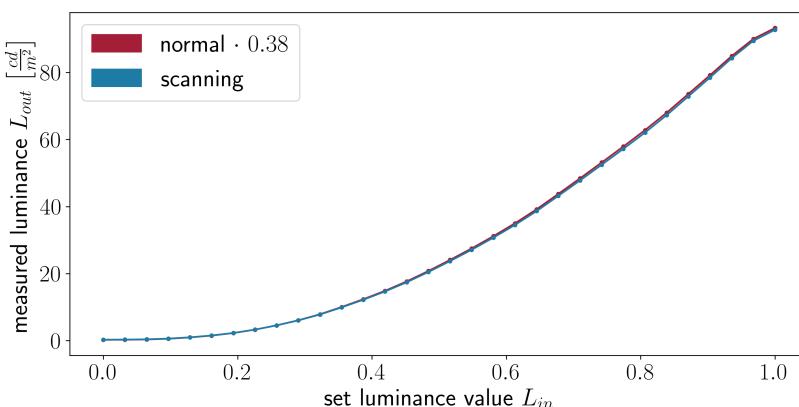


Figure 4.9: The amplitude measured by an OTR during 10 s in the normal (●) and the scanning (●) mode.

³² see Chapter 4.2.2

³³ The significance of this is explained in Chapter 4.4.

³⁴ The significance of this is explained in Chapter 4.3.

³⁵ Monitor: VIEWPixx
Experiment setup: Setup I
Mode: normal
Number of L_{in} : 25
Temperature: 35.125 °C
Distance: 30 cm
Raw data: Appendix F.6

³⁶ Monitor: VIEWPixx
Experiment setup: Setup I
Mode: scanning
Number of L_{in} : 25
Temperature: 35.125 °C
Distance: 30 cm
Raw data: Appendix F.7

Figure 4.10: A luminance curve of the normal mode which is almost exactly the same as the luminance curve of the scanning mode due to the factor 0.38.

Evaluation of the result

The observations from Chapter 4.2.2 and Chapter 4.2.2 indicate that measurements in the two modes can be equalized by a factor of 0.38. This implies that normal and scanning modes differ only in the number of LED bars activated simultaneously. Nonetheless, the factor 0.38 should be used with care. First, it was determined with 1 % accuracy. Second, it applies solely at the same temperature.³⁷ However, this fact should not be taken for granted as all LED bars in normal mode maintain a higher temperature compared to the scanning mode.

4.3 Temperature

The fact that temperature influences the characteristic of luminance presentation is known. In the manual of the monitors by VPixx Technologies Inc. it is especially stated to obey a warm-up period for reliable measurements. However, the extent to which the temperature affects the luminance characteristic could not be determined. To get an overview of this, the warm-up phase is examined in more detail in Chapter 4.3.1. The extent to which the temperature affects the gamma characteristics of the monitor is then examined in the following Chapter 4.3.2.

4.3.1 Warm-up

While VPixx Technologies Inc. claims, that their “innovative LED backlight design eliminates the long warm-up delay required by CRTs and other LCD backlight technologies”[13, p. 5] and their data sheet reports that the “warm-up time is around 20 minutes”[13, p. 10], empirical data from the NIP research group contradicts this claim. Recent empirical findings showed, that one should obey a warm-up period of approximately 1.5 to 2 hours. In order to examine this more deeply and to investigate the systematic dependency on temperature, several warm-up experiments are conducted. One is shown in Figure 4.12³⁸. Both temperatures, the ambient temperature and the monitor temperature, were measured at regular intervals of 2 minutes. The monitor temperature, which is returned directly by the monitor, comprises 8 temperature sensors placed at various points within the monitor housing.³⁹ In the following, we have calculated the average of these sensors. The ambient temperature was measured using an external temperature sensor, placed approximately 20 cm behind the monitor.

Figure 4.12 shows that the temperature consistently rises from the start (13:34) until it reaches a steady level of $\approx 39.9 \text{ }^{\circ}\text{C}$ after 1:49 hour (15:24). During the experiment the ambient temperature increases by approximately $\approx 1 \text{ }^{\circ}\text{C}$.

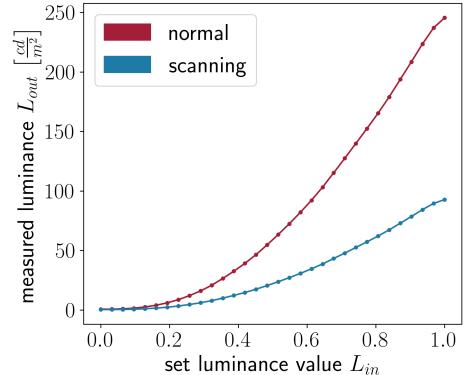


Figure 4.11: Two measured luminance curves that were measured on the same monitor (VIEWPixx) with the same temperature ($35.125 \text{ }^{\circ}\text{C}$).

³⁷ The fact that temperature actually has a major effect on the properties of luminance is explained in more detail in Chapter 4.3.

³⁸ Monitor: VIEWPixx /3D Lite
Experiment setup: only monitor
Mode: scanning
Number of L_{in} : -
Temperature: $\approx 24.9 - 39.9 \text{ }^{\circ}\text{C}$
Distance: -
Raw data: Appendix F.8

³⁹ The 8 sensors have a resolution of $\pm 1 \text{ }^{\circ}\text{C}$. For the sake of simplicity, the average of these sensors will be calculated, resulting in a pseudo resolution of $\frac{1}{8} \text{ }^{\circ}\text{C} = 0.125 \text{ }^{\circ}\text{C}$.

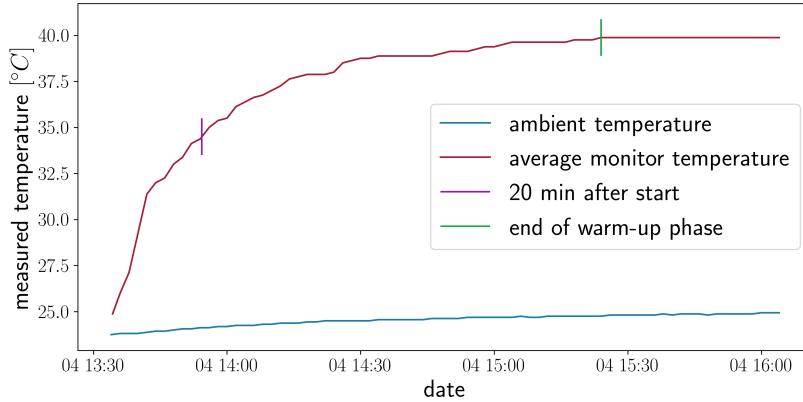


Figure 4.12: Temperature measurement ≈ 2.5 hours beginning at startup of the monitor. The raw data can be found in Appendix F.8.

This data indicated that the warm-up period is about 90 - 120 min which is 4.5 - 6 times longer than the specification in the data sheet.[13, p. 10]

4.3.2 Effect on luminance characteristic

After discovering in Chapter 4.3.1 that the warm-up phase is significantly longer than what was specified by *VPixx Technologies Inc.* and that the monitor's temperature fluctuates rapidly due to environmental changes, we conducted an experiment to investigate how these temperature variations affect luminance. The following outlines our experimental⁴⁰ procedure. The monitor was powered on in normal mode from the off state. During this warm-up phase with increasing temperature from ≈ 30 °C to ≈ 42 °C 50 quick experiments (32 different L_{in} values within ≈ 2.4 min) were performed in order to show directly the temperature dependency of luminance. The resulting values were visualized in Figure 4.13.

From this data, it is evident that temperature has a significant impact on luminance. The luminance curves in Figure 4.13 reveal two effects, which will be further examined below.

REDUCTION OF L_{max} : One impact of temperature increase is the reduction of the maximum luminance (L_{max}). In the coldest measured experiment (≈ 30 °C) L_{max} equals $\approx 257 \frac{cd}{m^2}$, while in the warmest experiment (≈ 42 °C) L_{max} decreased to $\approx 249 \frac{cd}{m^2}$. As LEDs tend to lose their luminosity as temperature rises⁴¹, the decreased backlight intensity may be attributed to this general property.

CHANGE IN THE SHAPE OF THE LUMINANCE CURVE: The second effect is the change shape of the luminance curve. While Figure 4.13 provides an indication of the effect, Figure 4.14 shows the changes in the shape of the curves. This is achieved by subtracting the resulting curve of the coldest measured experiment from others with different temperatures. These changes in the shape are significant and peaks in a maximum deviation of $32 \frac{cd}{m^2}$ at the highest temperature difference.

⁴⁰ Monitor: VIEWPixx /3D Lite
Experiment setup: *Setup I*
Mode: *normal*
Number of L_{in} : 32
Temperature: ≈ 30 - 42 °C
Distance: 30 cm
Raw data: Appendix F.9

⁴¹ Several factors contribute to the decrease in emission intensity of LEDs, such as non-radiative recombination through deep levels, surface recombination, and carrier loss across heterostructure barriers.[26, p. 97]

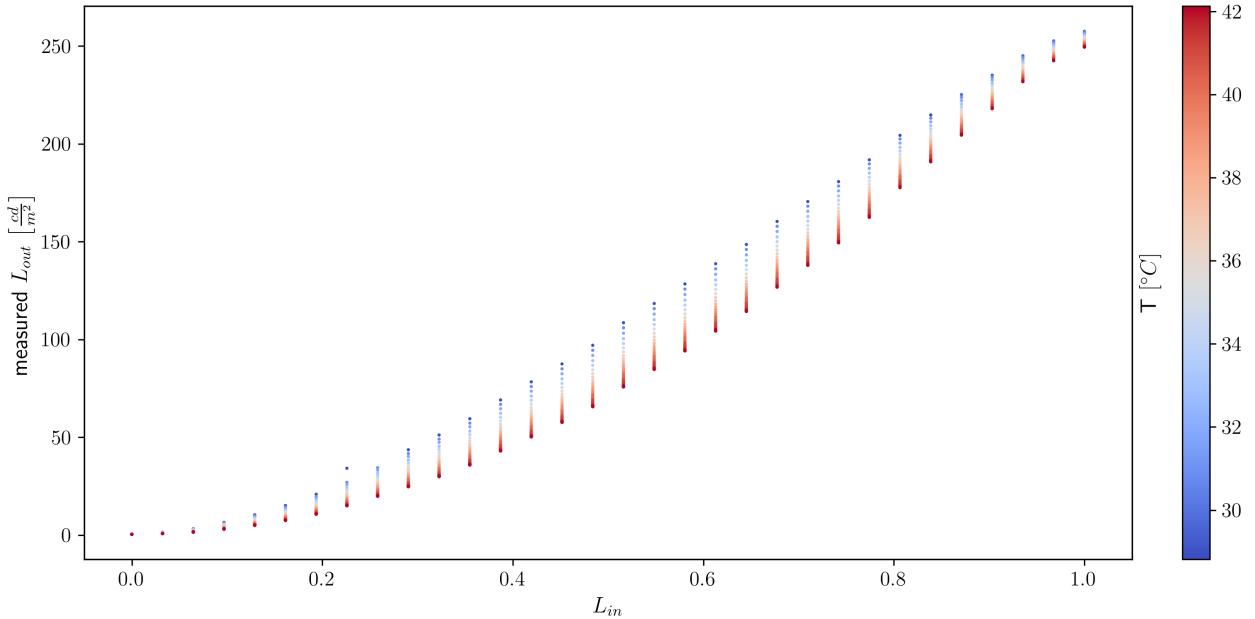


Figure 4.13: Variation of absolute luminance as well as gamma dependency with increasing temperature during warm-up. Internal backlight temperature rises from ≈ 30 °C to 42 °C. The raw data can be found in Appendix F.9.

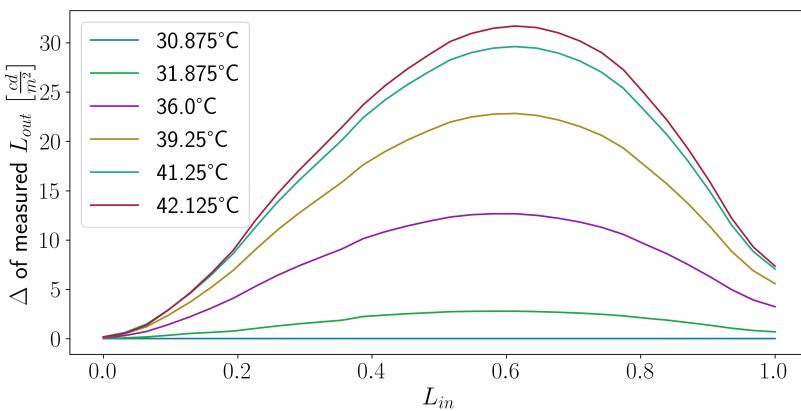
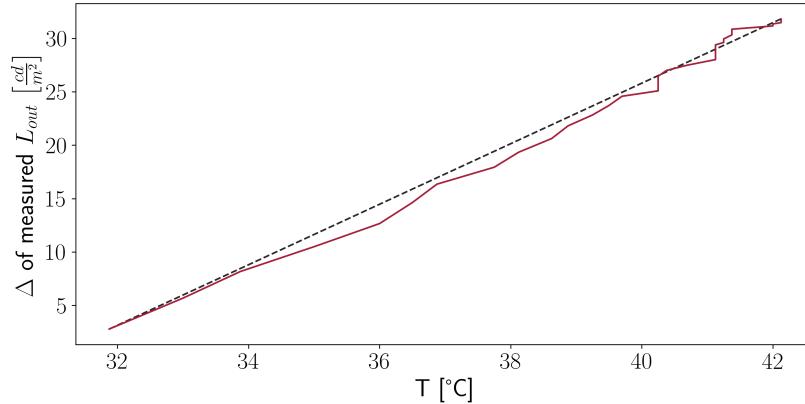


Figure 4.14: Absolute Δ between luminance measurements at different temperatures, using the data from Figure 4.13.

While Figure 4.14 indicated a monotonic increase of luminance dependency on temperature, Figure 4.15⁴² shows this change as even a linear trend.



⁴² In the plot, only the absolute maximum difference of the individual experiments is measured, but it can be estimated that this then applies to the entire curve

Figure 4.15: Maximal absolute Δ between luminance measurements and their corresponding temperatures, using the data from Figure 4.13.

Contrary to the observed reduction of L_{max} , this changes in the shape of the luminance curve, can not be solely attributed to decreases in the emission intensity of LEDs.⁴³ Rather this implies that the liquid crystals of the LCD or their electronic control circuits are likely to be significantly affected by temperature.

These observations show that, temperature is a critical factor in determining the shape of the luminance curve, in addition to L_{in} .

4.3.3 Effect of ambient temperature

The monitor temperature is directly affected by the ambient temperature. The monitor's cooling system consists of 6 small fans that run at a constant speed upon start-up. Figure 4.16 displays the temperature of the monitor situated in the laboratory for several days. The temperature of the monitor exhibits regular changes without being interfered with.⁴⁴ At approximately 20:00, the temperature of the device increases by about 1 °C and subsequently lowers by approximately 9:00 to its initial temperature. This is due to the automatic air conditioning system that turns off at approximately 20:00, resulting in the monitor heats up, together with the ambient temperature in the room. The air conditioning system then turns on again at around 6:00 and requires an additional 3 hours to bring the temperature back down to its initial level.

Currently, we observe a 1 °C deviation, which is quite reasonable. But even then we do observe a change of up to $3 \frac{cd}{m^4}$, as elucidated in Chapter 4.3.2.

4.4 Monitor

Until now, it was presumed that the luminance characteristic of a monitor is only influenced by the parameters L_{in} ⁴⁵, mode⁴⁶ and temperature⁴⁷. However, some experiments have revealed

⁴³ If this were the case, then the curves in Figure 4.14 would have to monotonically increase.

⁴⁴ We have no explanation for the deviations, such as on 19.11 at around 14:00. The hypothesis is that the sensor made an incorrect measurement here.

⁴⁵ see Chapter 4.1

⁴⁶ see Chapter 4.2

⁴⁷ see Chapter 4.3

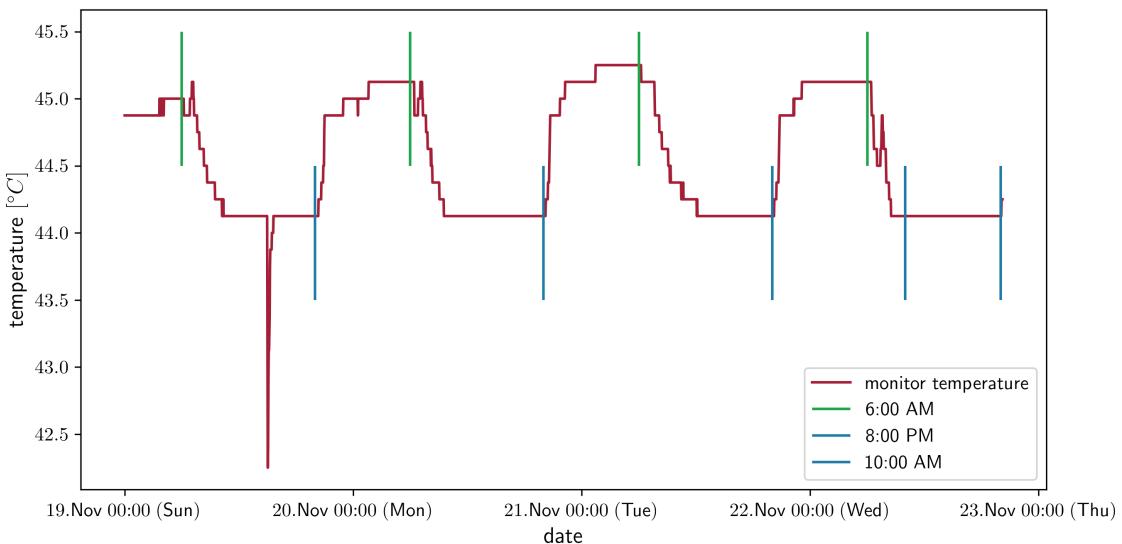


Figure 4.16: Temperature measurements by the monitor in the lab, that show influence of change in room cooling. The raw data can be found in Appendix 4.12.

⁴⁸ see Figure 4.13

that the choice of monitor impacts its temperature behavior and thus the luminance behavior. That is why, in the course of this master's thesis, the temperature analysis⁴⁸ was not only done for one monitor (VIEWPixed /3D Lite), but a second time for the VIEWPixed monitor as well⁴⁹. Both show distinct characteristics resulting from each analysis enabled the development of separate temperature "profiles"⁵⁰ for each monitor, which we will discuss in the following sections.

As can be seen in Figure 4.17 and Figure 4.18, the behavior of the VIEWPixed varies significantly differently with temperature changes in comparison to the . Unlike the VIEWPixed /3D Lite⁵¹, the VIEWPixed does not exhibit a parabolic shaped change in its luminance characteristic with rising temperature.⁵² In fact, the temperature dependence increases with increasing L_{in} . What the two monitors have in common, however, is that the maximum absolute difference between the individual measured curves increases linearly with increasing temperature.⁵³

Due to the differences observed among different monitor models, we conducted a comparative analysis using two VIEWPixed /3D Lite monitors obtained at different times to check if there are also differences between devices of the same model. Specifically, we investigated models with serial numbers 65501-3106 and 65501-3154, acquired in 2013 from NIP and in 2021 from the experimental cognitive science research group⁵⁴ led by Prof. Dr. Volker Franz, respectively. As shown in Figure 4.19, its temperature profile is similar to that of the VIEWPixed /3D Lite monitor with the serial no. 65501-3154⁵⁵.

⁴⁹ Monitor: VIEWPixed

Experiment setup: Setup I

Mode: normal

Number of L_{in} : 32

Temperature: 27.375 - 45.5 °C

Distance: 30 cm

Raw data: Appendix F.10

⁵⁰ We will use this term in the rest of this thesis.

⁵¹ see Chapter 4.3.2

⁵² Compare Figure 4.14 and Figure 4.18 (a).

⁵³ Compare Figure 4.15 and Figure 4.18 (b).

⁵⁴ <https://www.ecogsci.cs.uni-tuebingen.de/index.php>

⁵⁵ You can find the temperature profile of this monitor in Chapter 4.3 as Figure 4.13.

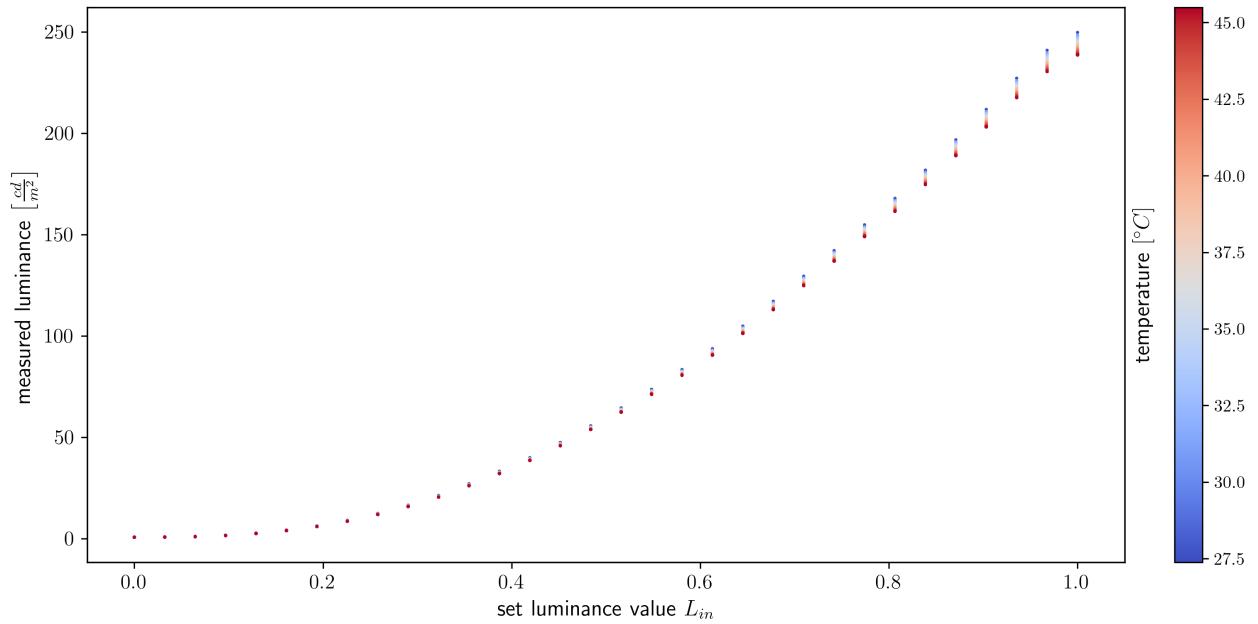
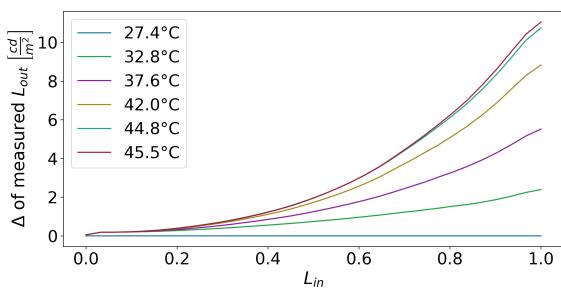
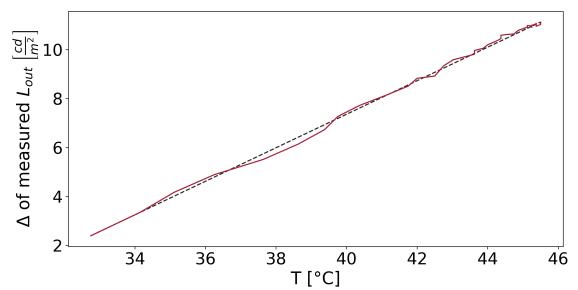


Figure 4.17: Temperature profile of the VIEWPixx (serial no. 60001-0733) monitor. The raw data can be found in Appendix F.10.



(a) Absolute Δ between luminance measurements at different temperatures.



(b) Maximal absolute Δ between luminance measurements and their corresponding temperatures.

Figure 4.18: Absolute Δ between luminance measurements at different temperatures (a) and visualizations Maximal absolute Δ between luminance measurements and their corresponding temperatures (b), using the data from Figure 4.17.

It can therefore assume that different monitor models have the same temperature dependency. Nevertheless, significant differences among different monitor models exist.

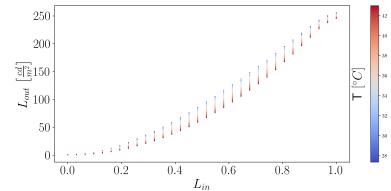


Figure 4.19: Temperature profile of the VIEWPixx /3D Lite (serial no. 65501-3154) monitor. The raw data can be found in Appendix F.11.

5 Model

This chapter focuses on creating a statistical model tailored to capture the characteristics of VIEWPixed monitors¹ as representatives for high luminance resolution displays. The goal is to improve the accuracy and speed of the calibration process through an appropriate model.

As detailed in Chapter 4, we are confronted with a 4-dimensional problem. However, the investigations in Chapter 4.2.2 revealed that the scanning and normal modes can be (approximately) offset of each other. This means, that it is possible to reduce our problem to two relevant dimensions of the dependent parameter: monitor model. The relevant dimensions are L_{in} and temperature.

To solve this reduced two-dimensional problem, Gaussian process (GP)s were selected as the model of choice. Chapter 5.1 gives an overview of the reasons to use a GP and their basic implementation. Subsequent chapters detail how to pre-instruct the model with prior knowledge about the two parameters. Chapter 5.2 presents a method for modelling the gamma dependency and Chapter 5.3 outlines a method to model the temperature dependence. Finally, in Chapter 5.4 both modeling approaches are then combined.

5.1 Gaussian process (GP)

A GP extends the Gaussian probability distribution, governing functions rather than just scalar or vector random variables.[23] Below, advantages and characteristics of GPs are discussed.

PROBABILISTIC MODEL: Noise and deviation are inherent aspects of Gaussian processes, distinguishing them from deterministic models.[23, 8] It is therefore a probabilistic model which encapsulate uncertainty, which is necessary because we must contend with noise and uncertainty in our measurement processes.

FLEXIBILITY: Unlike certain regression methods, such as polynomial regression, GPs possess a unique flexibility. They refrain from exhibiting oscillations or overfitting tendencies commonly associated with polynomial models. Instead, they maintain a smoother behavior while being adaptable to various unknown or impossible to model data patterns. This characteristic empowers GPs to accommodate diverse datasets without becoming excessively complex or

¹ Emphasizing the VIEWPixed/3D Lite due to its notably stronger temperature effect (refer to Chapter 4.4), potentially necessitating a more intricate (and thus more intriguing) model.

overly reliant on specific assumptions, providing robust means to address systematic variations.[23, 8]

HIGH-DIMENSIONAL: A notable feature of Gaussian Processes is their high-dimensional applicability. Theoretically extending their utility to problems with a high number of dimensions. This property makes GPs an attractive choice for modeling complex systems, as they can handle multivariate data and capture intricate relationships across numerous dimensions.[23, 8]

MINIMAL INFORMATION REQUIREMENT is another advantage of Gaussian Processes. These models demonstrate efficiency even with relatively sparse data points, leveraging their probabilistic nature to make informed predictions with a limited amount of information. However, while they can operate with fewer data points compared to certain other methods, an appropriate number of samples is crucial to ensure reliable predictions and account for uncertainties effectively.[23]

As implementation for GPs in this thesis the GPflow framework² for Python is used.[19] As stated by [18, 23], a GP is fully defined by a mean function and a covariance function, also called kernel. GPflow allows for the flexible selection of both functions. Additionally, the framework enables the handling of varying output noise through a likelihood method.[28]

Since the individual L_{in} values are the respective means³, we can assume, by the central limit theorem (CLT), that they follow a Gaussian distribution.[6] Figure E.3 and Figure E.4 suggest that this distribution exhibits increasing noise in L_{out} . To incorporate this behavior, I implemented a function that enables a Gaussian noise with a linear increasing scale.⁴

To model a variety of issues using GPs, you may utilize either the kernel or the mean function. There is no consensus on which of these alternatives is superior. Rasmussen and Williams [23] reduces the mean function to zero and primarily adjusts the GP via the kernel's design, whereas Hwang et al. [12] demonstrates the significance of the mean function and the critical effect of setting it to zero on the hyperparameters. Since I chose to utilize the GPflow framework for my thesis implementation, I follow their interpretation of when mean function and kernel should be used. Their documentation advises employing the mean function when “external factors lead you to anticipate a particular data shape. For instance, if you are modeling a physical system that can be described by closed-form equations from physics.”[29] Through the characterization⁵ we do have a plausible expectation about the shape of the data. The effects of the two parameters (L_{in} and temperature) are therefore mapped below with mean functions. This means that much less consideration is given to the kernel during modelling. A simple squared exponential kernel (SQE)

² <https://www.gpflow.org/>
(see Appendix G for the used version)

³ see Appendix E

⁴ see Code K.9

⁵ see Chapter 4

is utilized as it is the widely accepted default and can be used universal. Code 5.1 shows how it is implemented.

```
1 | kernel = gpflow.kernels.SquaredExponential()
```

Code 5.1: Code to define a **SQE**.

5.2 Gamma

As already shown in Chapter 4.1, we know that the biggest influence on L_{out} comes from L_{in} (as intended) which can be described by a (slightly shifted and stretched) gamma dependency⁶, which has the following generalized form:

$$L_{out} = \alpha + \beta \cdot L_{in}^{\gamma}$$

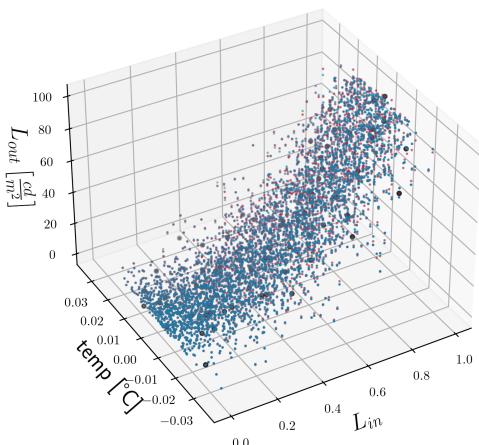
From our previous experiments we know that γ should be ≈ 2.2 , α should be ≈ 0.44 and β should be ≈ 245 .⁷

Since we expect this effect only to be based on L_{in} , we have created a mean function that can handle two-dimensional input data, but only uses the first dimension to fit the generalized gamma dependency to it.⁸

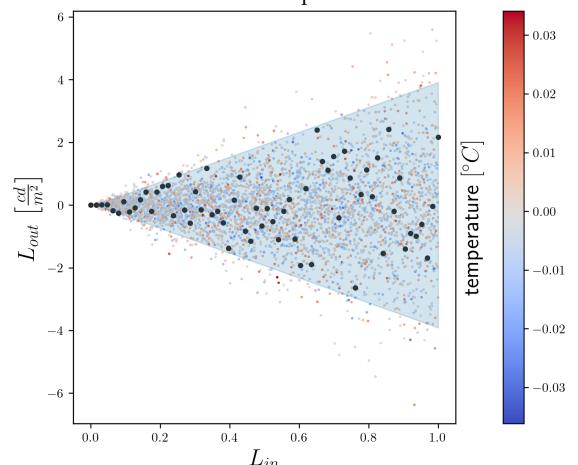
To ensure that a **GP** is capable of finding the underlying gamma dependency by using this mean function, we tested it with artificial data.

The result is visible in Figure 5.1 and Table 5.1.

As the result shows, the **GP** is capable with (just) 64 training points to find the gamma dependency.⁹ Even if the level of noise ($\sigma = 2$) surpasses our expectations in real measurements ($\sigma \lesssim 0.5$).¹⁰



(a) Training, prediction and “real” data points



(b) Residues of predicted points and “real” data points. The residues have $\mu = -0.0765$ and $\sigma = 1.13$.

Figure 5.1: Overview about the results from the **GP** on artificial data representing the gamma dependency.

- “real” data point
 - point that the **GP** got to train with
 - point that the **GP** predicted
 - residues point that represents his temperature with a color map
 - 95% confident interval of the **GP**
- $\sigma = 1.996$

name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.A	Parameter	Identity	True	(1, 1)	float64	[[95.28469]]	
GPR.mean_function.b	Parameter	Identity	True	(1,)	float64	[0.27339]	
GPR.mean_function.gamma	Parameter	Identity	True	()	float64	2.1972	
GPR.kernel.variance	Parameter	Softplus	True	()	float64	158.25159	
GPR.kernel.lengthscales	Parameter	Softplus	True	()	float64	146.48796	
GPR.likelihood.scale.A	Parameter	Identity	True	(1, 1)	float64	[[2.02231]]	
GPR.likelihood.scale.c	Parameter	Identity	True	(1,)	float64	[-0.02797]	

5.3 Temperature

As demonstrated in Chapter 4.3, L_{out} not only depends on L_{in} , but also has a strong temperature dependence.¹¹ Similar to the gamma dependency, this prior knowledge can be incorporated into the model as a mean function.

As Figure 4.14 and Figure 4.15 in Chapter 4.3 suggest, the temperature effect seems to approximately follow a 3-dimensional form, which can be described by the generalized form

$$\Delta_{L_{out}} = \alpha + \beta \cdot t \cdot (L_{in} - L_{in}^2)$$

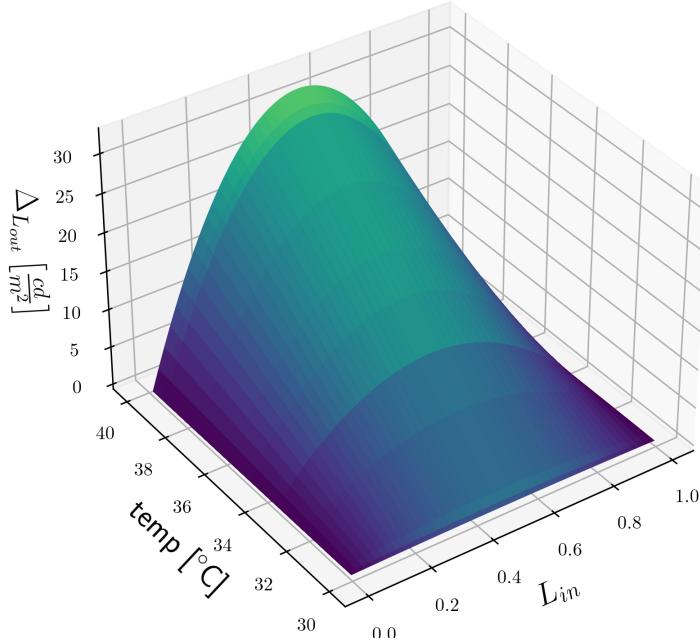


Table 5.1: Optimization result of the hyperparameter by the GP with artificial data representing the gamma dependency.

¹¹ Chapter 4.4 on the other hand demonstrates that the magnitude and nature of this temperature dependence depends on the specific monitor used.

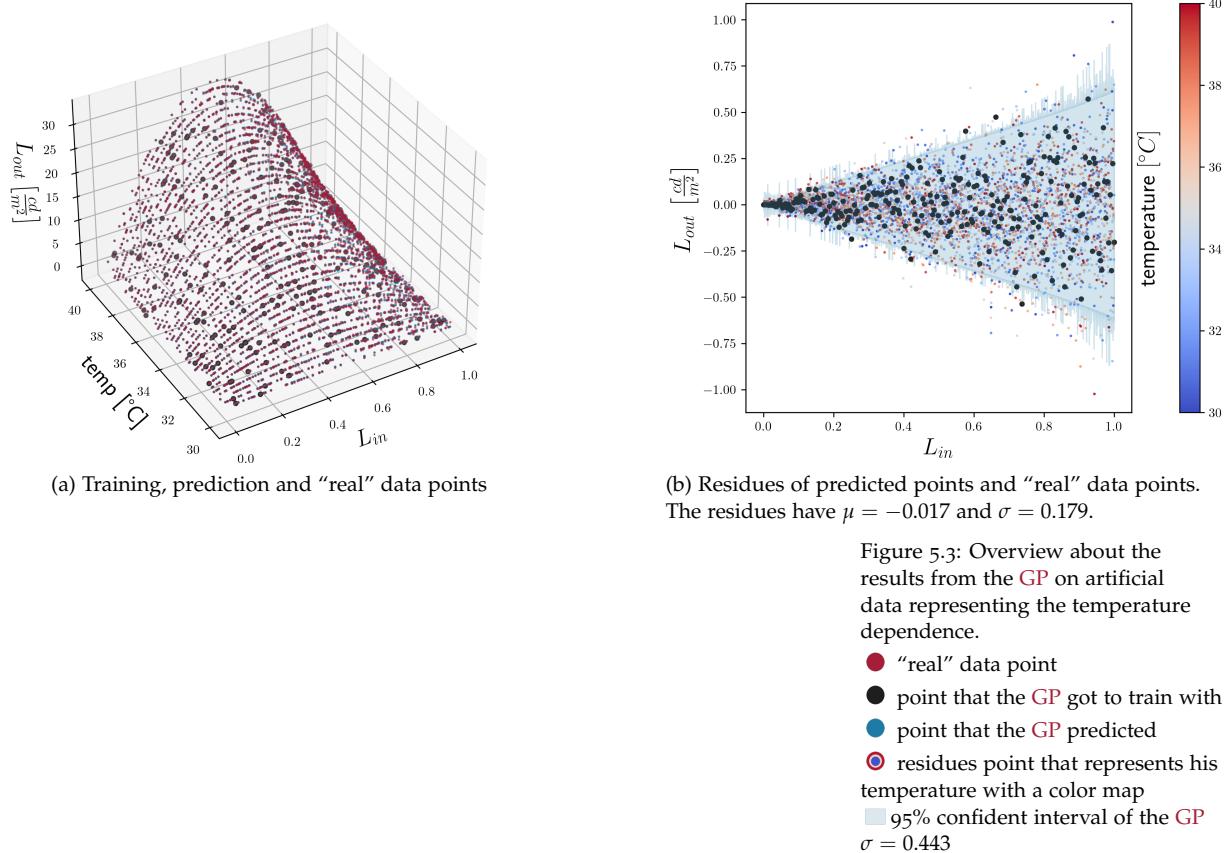
Figure 5.2: Exemplary visualization of $\Delta_{L_{out}} = \alpha + \beta \cdot t \cdot (L_{in} - L_{in}^2)$ with $\alpha = 0$ and $\beta = 130$, so that it approximately corresponds to the observed effect in Figure 4.14 and Figure 4.15.

and is illustrated by Figure 5.2. Which combines a linear increase in the $\Delta_{L_{out}}$ with increasing temperature t and a quadratic behavior with increasing L_{in} . The generalized form enables with α that there is a possible offset in the temperature effect. β determines the slope and the maximum value of the function.

To ensure that a GP is able to find this underlying functionality, we have also created a mean function¹² for it and tested it with artificial data.

The result can be seen in Figure 5.3 and Table 5.2.

¹² The source code for this can be found in Appendix K.2.



name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.c	Parameter	Identity		True	(1,)	float64	[845.25494]
GPR.mean_function.A	Parameter	Identity		True	(1, 1)	float64	[[2.92564]]
GPR.kernel.variance	Parameter	Softplus		True	()	float64	115158.58487
GPR.kernel.lengthscales	Parameter	Softplus		True	()	float64	4.01797
GPR.likelihood.scale.A	Parameter	Identity		True	(1, 1)	float64	[[0.29801]]
GPR.likelihood.scale.c	Parameter	Identity		True	(1,)	float64	[-0.00017]

In contrast to the gamma dependency¹³, this time the GP was not able to completely fit the temperature dependence with only 64 training points.¹⁴ Figure 5.4 shows that although the GP has converged, the residuals still have a strong temperature dependence and thus do not represent the temperature dependence correctly. With 256 training points, however, the GP then arrives at a result that no longer contains any temperature dependency.

Table 5.2: Optimization result of the hyperparameter by the GP with artificial data representing the temperature dependence.

¹³ see Chapter 5.2

¹⁴ The used code to create the fake data and the GP can be found in Appendix H.2.

5.4 Combination of gamma dependency and temperature dependence

To see if a model would be able to cope if both effects are combined, we also created artificial data where both effects persist.¹⁵ As Figure 5.5 shows, the GP is able to find a fit with 1024 data points, which no longer shows any systematic in the residuals and only contains the given noise. Unfortunately, Table 5.3 shows that it does not find the input parameters, but estimates the gamma with ≈ 3.7 .

This indicates that the GP tends to address the entire problem, including the change in the shape of curve from the temperature dependence, using gamma instead of distinguishing between the two effects as originally intended. To confirm this, a third mean function was devised that distinguishes between the two constituents of the mean function utilizing a weight factor α .¹⁶ The following equation provides an overview of the revised generalized mean function.

$$L_{out} = \alpha \cdot (c_1 + \beta \cdot L_{in}^\gamma) + (1 - \alpha) \cdot (c_2 + \delta \cdot t \cdot (L_{in} - L_{in}^2))$$

The alpha value being ≈ 1.0 in the GP result¹⁷ indicates that the mean function's first part (gamma) is the only part used, and the second part is disregarded to match the training data.

However, a control experiment¹⁸ using the same artificial data, comprising both effects and fitted with a GP containing solely a mean function for the gamma dependency¹⁹, showed that this GP produces results with a significant systematic deviation in its errors which can be attributed to temperature variations.

A model that considers both effects appears to yield better outcomes and thus qualifies as a superior model. Although certain temperature effects may be remedied with a modified gamma approach and not with the intended mean function.

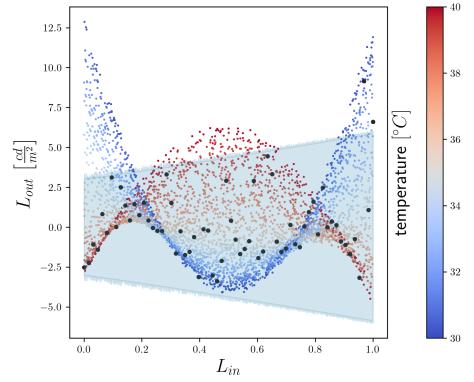


Figure 5.4: Residues of predicted points and “real” data points, when the GP has 64 training points. The residues have $\mu = 0.72$ and $\sigma = 2.71$.

- point that the GP got to train with
 - residues point that represents his temperature with a color map
 - 95% confident interval of the GP
- $\sigma = 3.066$

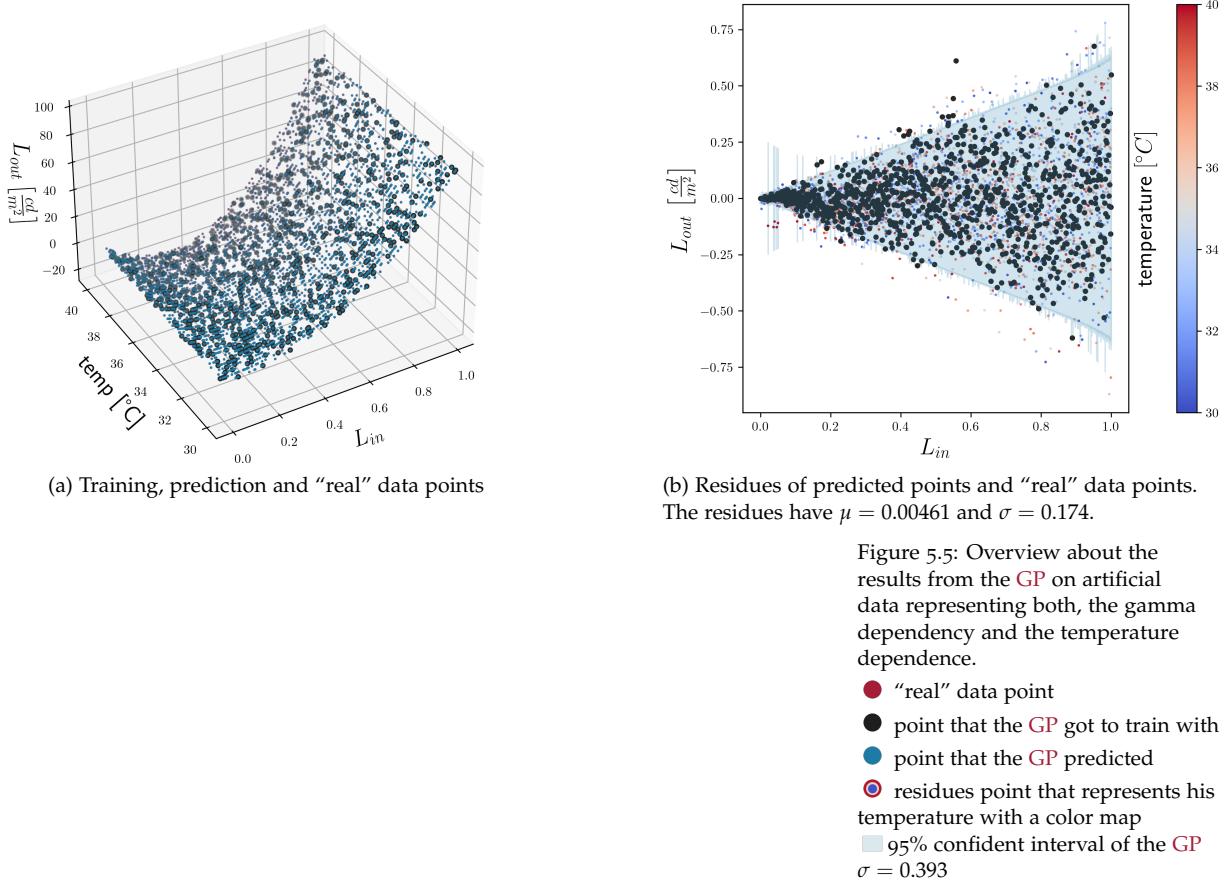
¹⁵ The used code to create the fake data and the GP can be found in Appendix H.3. A comparison of the artificial data from Figure 5.5 and real data from Figure 6.1 shows that the artificial data is somewhat exaggerated and does not correspond 100 % to the real measurements. However, this is not the primary goal of this chapter. It is just to show that the model can handle both effects being mixed. In order for the data to be closer to the real data, its parameters would have to be adjusted accordingly.

¹⁶ The source code for this can be found in Appendix K.3.

¹⁷ The result can be found in Appendix L.1.

¹⁸ The result of this experiment can be found in Appendix L.2.

¹⁹ The source code for this can be found in Appendix K.1.



name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.add_1.A	Parameter	Identity		True	(1, 1)	float64	[[13.00696]]
GPR.mean_function.add_1.b	Parameter	Identity		True	()	float64	0.80135
GPR.mean_function.add_1.gamma	Parameter	Identity		True	()	float64	3.67432
GPR.mean_function.add_2.c	Parameter	Identity		True	()	float64	0.80135
GPR.mean_function.add_2.A	Parameter	Identity		True	(1, 1)	float64	[[-3.29346]]
GPR.kernel.variance	Parameter	Softplus		True	()	float64	1785.22436
GPR.kernel.lengthscales	Parameter	Softplus		True	()	float64	1.20797
GPR.likelihood.scale.A	Parameter	Identity		True	(1, 1)	float64	[[0.29938]]
GPR.likelihood.scale.c	Parameter	Identity		True	()	float64	0.00083

Table 5.3: Optimization result of the hyperparameter by the GP with artificial data representing both, the gamma dependency and the temperature dependence.

6 Validating the model with real data

After developing a model capable of fitting artificial data in Chapter 5, this chapter's objective is to validate the model on real data. In this regard, Chapter 6.1 evaluates the model's predictive capacity for measured values. Subsequently, Chapter 6.2 examines the feasibility of implementing the linearization through the model in order to compare it with the linearization of the previous process.¹

¹ see Chapter 3

6.1 Prediction of measurements

In order to validate the model constructed in Chapter 5, the task of this chapter is to first train the model using actual measured data (6.1.1). The next step is to evaluate the performance by predicting actual measured data (that are unknown to the trained model) and comparing the residuals of the predictions with the actual data (6.1.2).

6.1.1 Training

Acquiring data that covers the complete range of L_{in} and temperature concurrently presents a challenge. As described in Chapter 4.3 and Chapter 4.4, the experiments analyzing the temperature profiles² of individual monitors offer insights into their behavior. Unfortunately, these measurements were only carried out using 32 L_{in} values. A GP model trained on these data would likely identify the temperature dependency, but not the full dependency on L_{in} . To obtain these data, an experiment measuring all 4096 L_{in} values would be necessary. However, since measuring all 4096 L_{in} values take 5 hours and the warm-up phase, which has the greatest temperature spread, lasts "only" 1.5-2 hours³. This means that it is not possible to conduct both experiments simultaneously.

² see Figure 4.13, Figure F.10 and Figure 4.19

Another possibility would be to systematically change the ambient temperature. Unfortunately, the NIP research group does not have complete and precise enough control over the climate system in the lab because it is connected to the building's control system.

As a compromise, we used an experiment⁴ including three test runs to train the model. Figure 6.1 gives an overview of these three test runs. All three were conducted with random measurement order of different L_{in} values. This approach provides an advantage of eliminating any correlation between temperature changes and

³ More information about this can be found in Chapter 4.3.1

⁴ Monitor: VIEWPixx /3D Lite
Experiment setup: Setup I
Mode: scanning
Number of L_{in} : 2¹²
Temperature: $\approx 35 - 39 \text{ }^{\circ}\text{C}$
Distance: 30 cm
Raw data: Appendix F.12

⁵ As already mentioned, the **NIP** research group does not have full control over the temperature in the lab, the different temperature ranges that occurred during this experiment are rather (lucky) coincidences and were not forced.

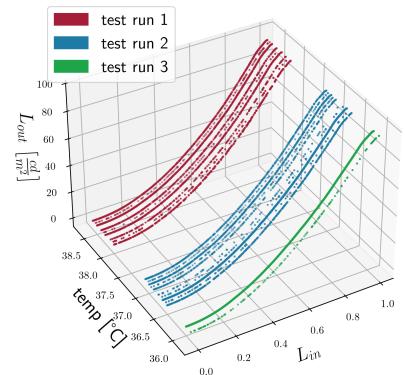
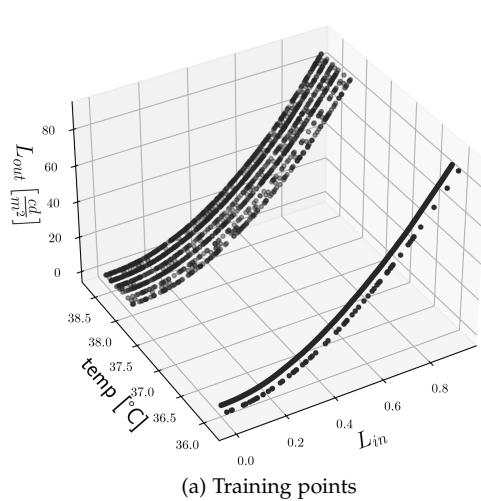
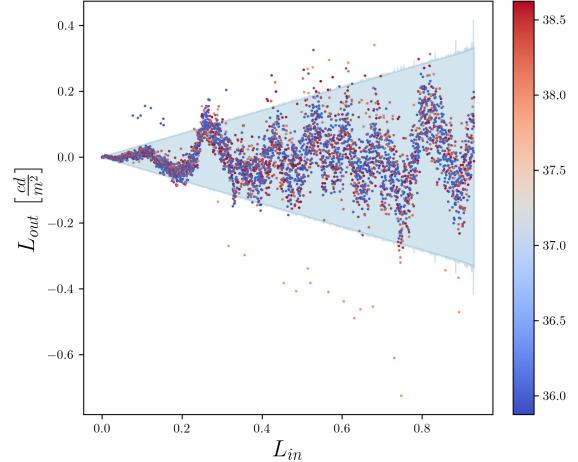


Figure 6.1: Overview of the distribution of the three test runs.



(a) Training points



(b) Residues of predicted points and real data points. The residues have $\mu = 0.0000$ and $\sigma = 0.0841$.

The result shows the fundamental ability of the model to be trained for both artificially generated and real data. The residuals between the mean predictions of the trained model and the real data points no longer show any temperature dependence. What is noticeable, however, is that the residuals show a rest systematic. It can therefore be seen that this does not yet cover another dependency that exists. However, since the σ of the residuals is smaller than we expect from experimental measurements¹⁰, the model was used for the time being.

6.1.2 Prediction

To test how well the model can now predict unknown values, the remaining data from the three test runs was taken. As only half of test runs 1 and 3 were used for the training, it was possible to compare how well the model performs in known temperature ranges (measurements from test runs 1 and 3) and how well it performs in unknown temperature ranges (measurements from test run 2). As the result in Figure 6.3 shows, the model is very good

Figure 6.2: Overview about the results from the **GP** trained with data from the real experiments.

- point that the **GP** got to train with
- residues point that represents his temperature with a color map
- 95% confident interval of the **GP**

⁶ Sparse gaussian process (**SGP**) is a **GP** extension that permits a larger amount of data.[11] However, this approach was not further pursued throughout this thesis. The reason is that in principle, every training data point must be collected first, which is a time-consuming process. This means that a model that is supposed to be suitable for the problem should deal with fewer rather than more training points

⁷ Resulting in "only" 4096 points.

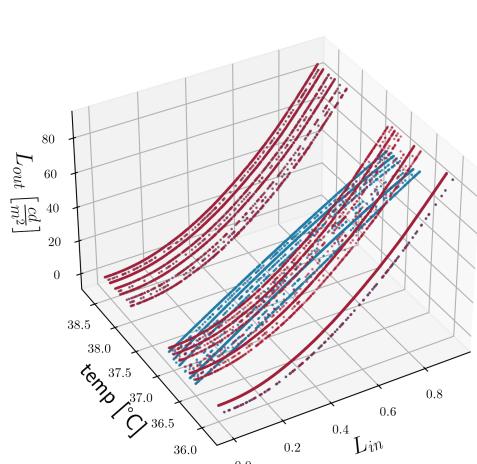
⁸ $2 \cdot 4096 = 8192$

⁹ We have limited our scope to $L_{in} \leq 0.93$ as we can be sure that the saturation effect will not be present in this range. This is necessary because the created model was not designed to cover saturation. More information can be found in Chapter 5.2.

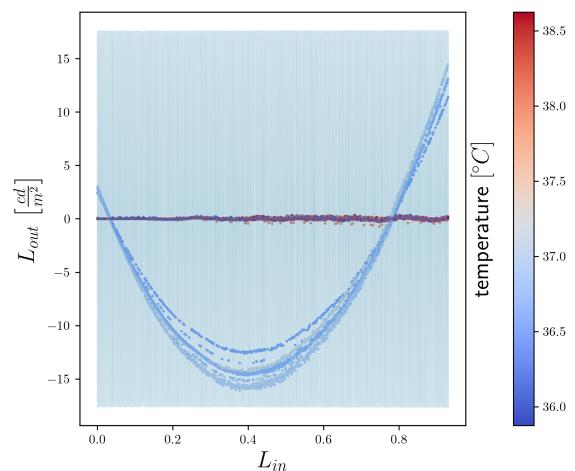
¹⁰ see Table I.1

at predicting data that is in the temperature ranges that are known to the model from the training data ($35 - 36^\circ\text{C}$ and $38 - 39^\circ\text{C}$).

Predictions for data outside these temperature ranges, namely the data of test run 2, deviate strongly from their correct values.



(a) Prediction and real data points from test runs 1 – 3



(b) Residues of predicted points and real data points. The residues have $\mu = -2.26$ and $\sigma = 5.38$.

As Figure 6.3 (a) shows, this deviation is due to the fact that the model attempts to linearize the values in unknown/untrained temperature ranges. This behavior of the model was also visualized again in Figure 6.4. Here a surface was “stretched” over the entire observed value range, displaying a clear visualization of the formation of a linearized “saddle” between the two temperature ranges, that were trained. As a consequence, the data in the range of this “saddle” cannot be predicted accurately.

This means that the model is not able to generalize. An assessment of what this means for the model is discussed in more detail in Chapter 7. Before that, in Chapter 6.2, a linearization is performed in order to be able to evaluate the model further.

6.2 Linearization

In order to validate the model, a linearization is carried out in this chapter. This is to determine whether the model is capable of replacing or extending the previous process described in Chapter 3.

For this two experiments were conducted. The first experiment¹¹ aimed to train a model using recorded measurement data. As was to be expected from the observations in Chapter 6.1, the model fits without residual temperature dependence, but also show a rest systematic. The result of this training is portrayed in Figure 6.5 and Table L.5.

To perform the linearization, the previous process is used as a guide. This means that the mean predictions of the trained model are used and inverted.¹² The resulting inverted curve was applied as a LUT to the monitor object, allowing for the presentation of a

Figure 6.3: Overview about the results from the trained GP when it is predicting unknown data points from test runs 1 – 3.

- real data point
- point that the GP predicted
- residues point that represents his temperature with a color map

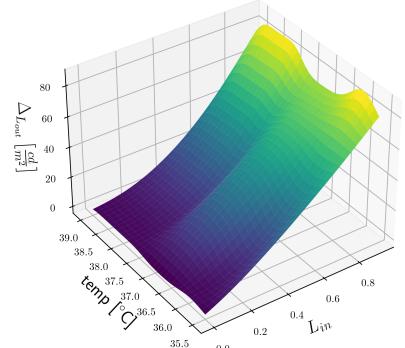
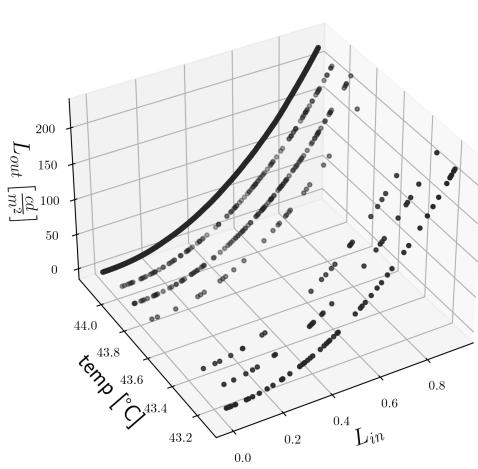


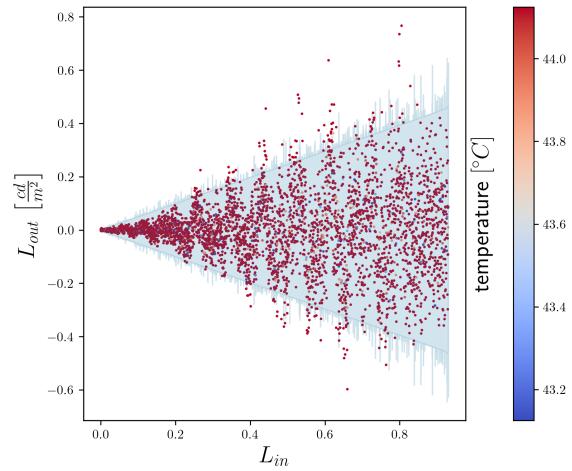
Figure 6.4: Overview of the distribution of the three test runs¹⁰.

¹¹ Monitor: VIEWPixx /3D Lite
Experiment setup: Setup I
Mode: normal
Number of L_{in} : 2¹²
Temperature: $\varnothing 44^\circ\text{C}$
Distance: 30 cm
Raw data: Appendix F.13

¹² For more details on how this inversion is done, see Chapter 3.

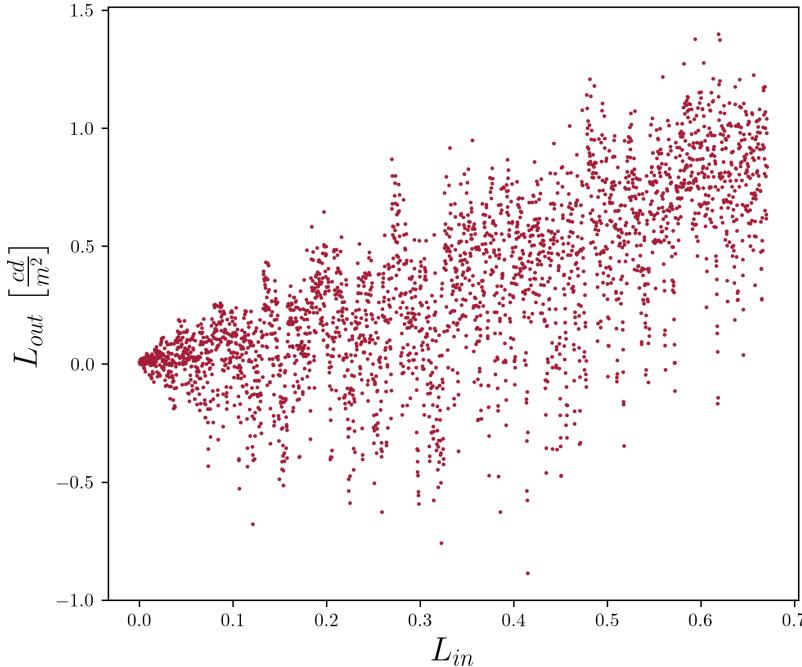


(a) Trainings points for the model to create an inverted LUT



(b) Residues of predicted points and measured points. The residues have $\mu = 0.00$ and $\sigma = 0.138$.

visual stimulus with a linearized luminance behavior. The success of this linearization was evaluated in a subsequent experiment¹³. The residues of the new measured luminance values and the ideal linear function can be viewed in Figure 6.6.¹⁴



As the results indicate, the linearization process was accomplished successfully, with residuals closely matching those of the previous process. This means that $\mu = 0.32$ and $\sigma = 0.37$ of the result are comparable to those of the previous method which were $\mu = -0.32$ and $\sigma = 0.26$.¹⁵ Also, the residuals of the result inherit a systematic, just like the current process. However, this is more linear and not parabolic as in the current process.¹⁶

Figure 6.5: Overview about the results from the GP trained with data from the experiment to enable linearization of visual stimuli.

- point that the GP got to train with
- residues point that represents his temperature with a color map
- 95% confident interval of the GP
 $\sigma = 0.329$

Figure 6.6: Residues of the measured linearized luminance and the ideal linear function, when using the trained model.

¹³ Monitor: VIEWPrix /3D Lite

Experiment setup: Setup I

Mode: normal

Number of L_{in} : 212

Temperature: 44.1 °C

Distance: 30 cm

Raw data: Appendix F.14

¹⁴ As previously discussed in Chapter 5.2 and Chapter 6.1, we have limited our scope to $L_{in} \leq 0.93$, as we can be certain that the saturation effect will not be present in this range. Figure 6.7 provides clear visual representation that this means, that linearization can only be achieved for values up to $L_{in} \leq 0.67$ due to the limitations of the inverted function.

¹⁵ see Chapter 3

¹⁶ Compare Figure 6.6 (new result) with Figure 3.1.

As described in Chapter 3, for mathematical reasons, all points that lead to a measurement curve not being monotonically increasing are discarded before inversion. Since the GP can recognize noise as such, the mean predictions are smoother and more monotonically increasing than in the previous process. Whereas in the previous process, 1473 values¹⁷ are thrown away before the inversion. In contrast, only 343 points have to be discarded when using the model.¹⁸ This is still not perfect, because of course it would be preferable not to have to discard any values, but it shows that even though the model is not perfect¹⁹, we benefit from the properties of the Gaussian process.

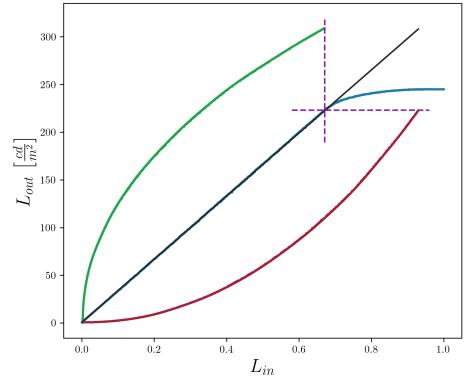


Figure 6.7: Visualization of the functions and the considerable value range of the second experiment.

- ideal linear function
- measured luminance after linearization
- measured curve in first experiment
- inverted curve that is used as a LUT
- visualization of the borders of the considerable value range

¹⁷ For reasons of comparability, we only consider the range of values of $L_{in} \leq 0.93$.

¹⁸ This represents a $\frac{1473 - 343}{1473} \cdot 100 \approx 77\%$ improvement.

¹⁹ see Chapter 6.1.2

7 Result and Discussion

This chapter derives conclusions based on the results of the thesis and organizes them into two main sections: **Characterization**¹ and **Model**². Within each section, the obtained results are listed and discussed.

7.1 Characterization

In Chapter 4 I was able to characterize the monitors³ by *VPixx Technologies Inc.* as a well-known representative of high luminance resolution displays. The following results were obtained:

LONGER WARM-UP PHASE than expected. As shown in Chapter 4.3.1 and visualized through Figure 4.12 the warm-up process of the monitor takes approximate 1.5 – 2 hours. However, an attempt should be made to keep the warm-up phase as long as possible in order to achieve constant temperature. This is because this thesis has also shown that a temperature difference that occurs, results in a change in the displayed luminance values.

TEMPERATURE DIFFERENCES HAVE A SIGNIFICANT INFLUENCE on the change in luminance characteristic. It is difficult to quantify the magnitude of this effect. This is because the absolute amount of change due to temperature depends on 4 parameters. These are L_{in} ⁴, the monitor model⁵, the backlight mode used⁶, and the actual temperature difference⁴. While the monitor model actually determines the shape of the temperature dependence, L_{in} , the mode and the actual temperature difference “only” change the size of the absolute value. It is therefore difficult to make a conclusive statement about how large the temperature effect actually is. Nevertheless, to get an idea of the size of this effect, the effect is quantified below for two scenarios (best- and worst-case). However, the most important result of this analysis is to be aware of this characteristic of these monitors by *VPixx Technologies Inc.*

Worst-case scenario: In this scenario, we use the VIEWPixx /3D Lite with the normal backlight mode and assume a temperature difference of 12 °C (warm-up phase). As Figure 4.14 shows, in this case we have a maximum temperature-dependent deviation of $\approx 32 \frac{cd}{m^2}$ ($\approx 13\%$).

¹ Covering Chapter 4

² Covering Chapter 5 and Chapter 6

³ see Chapter 2.1

⁴ see Figure 4.14

⁵ see Chapter 4.4

⁶ see Appendix J

Best-case scenario: In this scenario we use the VIEWPixx monitor in the scanning backlight mode and assume a temperature difference of only $\approx 1 \text{ }^{\circ}\text{C}$. Which corresponds to the fluctuation in ambient temperature due to the air conditioning in our controlled laboratory.⁷ As we can see from the slope of the (assumed) linear function in Figure 4.18 (b), a temperature difference of $1 \text{ }^{\circ}\text{C}$ means a temperature-dependent deviation of $\approx 0.6 \frac{\text{cd}}{\text{m}^2}$ ($\approx 0.24 \text{ \%}$).

⁷ see Chapter 4.3.3 and Figure 4.16

NORMAL AND SCANNING MODE CAN BE OFFSET against each other. In Chapter 4.2.2, I showed with two analyses that the scanning mode and the normal mode can (roughly) be offset against each other by a factor of 0.38.⁸ This factor should still be treated with care, as it was only determined to 1 % accuracy. Nevertheless, this result was an indication that temperature (which of course differs between scanning and normal mode) is the main domain effect that leads to a change in luminance values.

⁸ At the same temperature.

TEMPERATURE DEPENDENCIES PER VARIOUS MONITOR MODELS.

As indicated in Chapter 4.4, and demonstrated through Figure 4.13 and Figure 4.17, it is necessary to understand these distinctions, not only for creating a calibration model but also for selecting a monitor with appropriate properties, when designing experiments. The reasons for the temperature dependency in the various models remain unclear.

7.2 Model

In Chapter 5, the outcomes from Chapter 4 were utilized to develop a model for mapping and predicting the luminance behavior. The efficiency of this model was then tested with real data in Chapter 6, yielding the following results:

IT IS POSSIBLE TO CREATE A MODEL to model the luminance characteristic of the monitors by VPixx Technologies Inc. In Chapter 5, I showed that it is possible to create a statistical model based on GP that finds previously constructed correlations from artificial data that resemble the real characteristics of the monitors. In Chapter 6.1, this was confirmed⁹ with real data.

⁹ With limitations. The model was able to handle data whose temperature range it knew, but not data outside that range.

A LINEARIZATION IS POSSIBLE with the model and benefits from the properties of the GP. As Chapter 6.2 shows, it is possible to perform a linearization with the statistical model designed in this thesis based on GP. The linearization benefits from the fact that the GP means that fewer points have to be discarded during inversion, as the GP ensures that monotonically increasing points can be recognized as noise and a corresponding mean can be used here.

POSSIBLE IMPROVEMENT. Although the model predicted data with the correct error size in this thesis¹⁰, the data displayed a systematic

¹⁰ Compare the results in Chapter 6.2 and Chapter 3

error. It is hoped that an optimal model will cover such systematic errors resulting in a smaller deviation. Additionally, saturation was not considered in the previous model. A complete model would need to encompass saturation. Also, the model's poor prediction for unknown temperature ranges is an issue that requires improvement. Firstly, the model must make more accurate predictions than what is depicted in Figure 6.3. Secondly, it would be beneficial if the model could be trained with fewer points, as it is challenging to attain full coverage of both the temperature value ranges and L_{in} 's value range.

IDEA FOR A NEW PROCESS. From the sum of the results of this thesis and the experience with the created model, an idea was formed to implement a new calibration process. A sketchy visualization of the new process is shown in Figure 7.1.

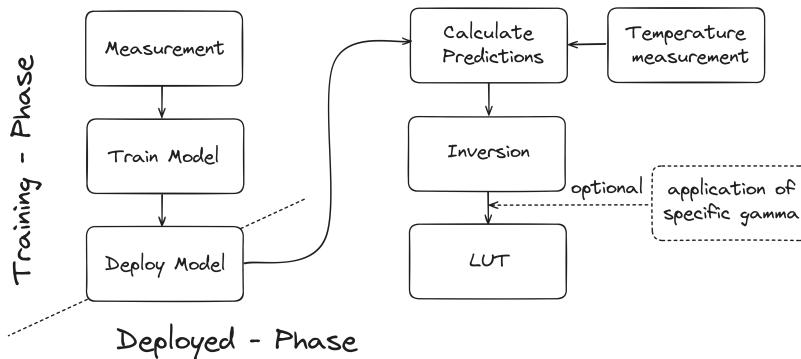


Figure 7.1: Visualized design of a calibration process including L_{in} and temperature as predictor variables.

The process can be divided into two phases, the Training-Phase and the Deployed-Phase. Both phases are separated by a step that involves the deployment of the model. The Training-Phase involves the measurement and subsequent training of the model. It is important that the training data is collected in such a way that it gives the model all the information it needs to cover the full range of possible temperatures in the experiments. Also, the design is based on the assumption that the model is perfect in the sense that it fully captures all properties of the problem. Among other things, this means that the systematic error that was present in the residuals of the previous model¹¹ is no longer present.

In the Deployed-Phase, this model is then used together with a temperature measurement to calculate predictions for the corresponding values of the luminance by the monitor, invert them¹² and then return a LUT that can be applied to the monitor.

This new process has the potential to significantly speed up the existing process.¹³ This is because the monitor only needs to be measured once and the model can then be used in its entirety.

¹¹ see Figure 6.2 (b)

¹² Optionally, the inverted prediction can be further adjusted at this stage, for example to allow a $\gamma = 1.6$ to be displayed.

¹³ see Chapter 3

8 Conclusion and Future Work

The present thesis entitled “Characterization of a high luminance resolution display for psychophysical experiments” has successfully identified four parameters that significantly control the behavior of the high-luminance monitors used. While the effects of L_{in} and the selected mode were expected, the importance of temperature and the differences in behavior between monitors were surprising. An in-depth study of these parameters reduced the scenario, previously assumed to be a 4D problem, to a 2D problem, as scanning and normal mode can be offset against each other, and it is assumed that only one monitor is used at a time.

A statistical model based on a GP was developed for this 2D problem, incorporating prior knowledge from previous characterizations such as gamma and temperature dependence. The developed model was successfully tested on synthetic data and then applied to real data. Despite its applicability, the model showed potential for improvement. A new procedure was derived from the results of this work, which could allow a significant acceleration of the previous procedure.

There is still room for further research and extension in the area of studying high luminance resolution displays. While this thesis focused on grayscale, the study of color may provide additional insight. One hypothesis would be that the temperature dependence varies between different colors, which could indicate that this dependence is influenced by the crystals.

One of the most important future research tasks, already mentioned in Chapter 7.2, is to refine and improve the model developed in this thesis. On the one hand, the identified saturation must be incorporated, but on the other hand, a way must be found to compensate for the residual systematic that can be found in the residuals. A possible further extension could be to create a model that applies to all monitors, thus transforming the 2D problem into a 3D problem. This could potentially improve the calibration process and open up other features of high luminance resolution displays.

To enhance the model further, a measurement process must also be created and implemented to instruct the model about the complete required temperature range, together with all values through L_{in} . One possible strategy could involve a laboratory setting in which the ambient temperature can be directly controlled,

thus enabling full coverage of the temperature range in a regulated manner.

Bibliography

- [1] Munisamy Anandan. *Progress of LED backlights for LCDs*. Tech. rep. in. 2008.
- [2] Marcelo Bertalmío. "Brightness perception and encoding curves." In: *Vision Models for High Dynamic Range and Wide Colour Gamut Imaging*. Elsevier, 2020, pp. 95–129. DOI: [10.1016/b978-0-12-813894-6.00010-7](https://doi.org/10.1016/b978-0-12-813894-6.00010-7).
- [3] Display-Messtechnik & Systeme GmbH & Co. KG. "OTR-3." In: URL: <https://www.display-messtechnik.de/fileadmin/template/main/docs/OTR3-6.pdf> (visited on 09/25/2023).
- [4] Tobias Elze and Thomas G. Tanner. "Temporal Properties of Liquid Crystal Displays: Implications for Vision Science Experiments." In: *PLoS ONE* 7.9 (Sept. 2012). Ed. by Bart Krekelberg, e44048. DOI: [10.1371/journal.pone.0044048](https://doi.org/10.1371/journal.pone.0044048).
- [5] Xiao-Fan Feng, Hao Pan, and Scott Daly. "P-50: Dynamic Gamma: Applications to Improve LCD Temporal Response using Overdrive." In: *SID Symposium Digest of Technical Papers* 36.1 (2005), pp. 462–465. DOI: <https://doi.org/10.1889/1.2036473>. eprint: <https://sid.onlinelibrary.wiley.com/doi/pdf/10.1889/1.2036473>. URL: <https://sid.onlinelibrary.wiley.com/doi/abs/10.1889/1.2036473>.
- [6] Hans Fischer. *A history of the central limit theorem. From classical to modern probability theory*. Sources and studies in the history of mathematics and physical sciences. Includes bibliographical references (p. 363–391) and indexes. New York: Springer, 2011. 402 pp. ISBN: 9780387878577.
- [7] Robert Geirhos, Kantharaju Narayananappa, and Benjamin Mitzkus. "Partial success in closing the gap between human and machine vision." In: 2021.
- [8] Andrew Gelman. *Bayesian data analysis*. Ed. by John B. Carlin et al. Third edition. Texts in statistical science. Includes bibliographical references and indexes. - Description based on online resource; title from title page (viewed May 10, 2017). Boca Raton: CRC Press, 2014. 11 pp. ISBN: 1439898227.
- [9] Masoud Ghodrati, Adam P. Morris, and Nicholas Seow Chiang Price. "The (un)suitability of modern liquid crystal displays (LCDs) for vision research." In: *Frontiers in Psychology* 6 (Mar. 2015). DOI: [10.3389/fpsyg.2015.00303](https://doi.org/10.3389/fpsyg.2015.00303).
- [10] Siavash Haghiri, Felix A. Wichmann, and Ulrike von Luxburg. "Estimation of perceptual scales using ordinal embedding." In: *Journal of Vision* 20.9 (Sept. 2020), pp. 14–14. ISSN: 1534-7362. DOI: [10.1167/jov.20.9.14](https://doi.org/10.1167/jov.20.9.14). eprint: https://arvojournals.org/arvo/content_public/journal/jov/938492/i1534-7362-20-9-14/_1600698630.54938.pdf. URL: <https://doi.org/10.1167/jov.20.9.14>.
- [11] James Hensman, Nicolo Fusi, and Neil D Lawrence. "Gaussian processes for big data." In: *arXiv preprint arXiv:1309.6835* (2013).
- [12] Seung-gyu Hwang et al. "How to use GP: Effects of the mean function and hyperparameter selection on Gaussian Process regression." In: (2022). DOI: [10.48550/ARXIV.2206.15081](https://doi.org/10.48550/ARXIV.2206.15081).
- [13] VPixx Technologies Inc. *User manual VPX-VPC-2001C and VPX-VPX-2000A*. www.vpixx.com.

- [14] VPixx Technologies Inc. *VIEWPixx Full (VPX-VPX-2001C) VIEWPixx Lite (VPX-VPX-2000A) VIEWPixx /3D Lite (VPX-VPX-2004B) VIEWPixx /3D Full (VPX-VPX-2005D) Installation Guide Version 1.2*. www.vpixx.com.
- [15] Mario Kleiner. *Psychtoolbox-3*. Apr. 2021. URL: <https://github.com/Psychtoolbox-3/Psychtoolbox-3>.
- [16] Jonathan WP Kuziek et al. "Real brains in virtual worlds: Validating a novel oddball paradigm in virtual reality." In: *bioRxiv* (2019), p. 749192.
- [17] Bernhard Lang. "Examining Lightness Perception with Photorealistically Rendered Scenes." MA thesis. Eberhard-Karls-Universität Tübingen, 2019.
- [18] David J. C. MacKay. *Information theory, inference, and learning algorithms*. 22nd printing. Cambridge [u.a.]: Cambridge University Press, 2019. 628 pp. ISBN: 9780521642989.
- [19] Alexander G. de G. Matthews et al. "GPflow: A Gaussian process library using TensorFlow." In: *Journal of Machine Learning Research* 18.40 (Apr. 2017), pp. 1–6. URL: <http://jmlr.org/papers/v18/16-537.html>.
- [20] Jonathan Oesterle. "COMPARING GAMMA CALIBRATION METHODS OF PSYCHTOOLBOX AND PSYCHOPY." Feb. 2018.
- [21] Jonathan Peirce et al. "PsychoPy2: Experiments in behavior made easy." In: (Feb. 2019). doi: [10.3758/s13428-018-01193-y](https://doi.org/10.3758/s13428-018-01193-y).
- [22] Charles Poynton. *Digital Video and HDTV Algorithms and Interfaces*. Morgan Kaufmann, p. 736. ISBN: 9781558607927.
- [23] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. 3. print. Adaptive computation and machine learning. Includes bibliographical references and indexes. Cambridge, Mass. [u.a.]: MIT Press, 2008. 248 pp. ISBN: 9780262182539.
- [24] William H Ridder III et al. "Effect of povidone iodine 5% on the cornea, vision, and subjective comfort." In: *Optometry and Vision Science* 94.7 (2017), pp. 732–741.
- [25] Robert Bosch GmbH. "PLR 25." In: 2011. URL: <https://manuall.de/bosch-plr-25-laser-entfernungsmesser/> (visited on 09/26/2023).
- [26] E. Fred Schubert. *Light-Emitting Diodes*. Cambridge University Press, 2003, p. 326. ISBN: 9780521533515.
- [27] *Single/Multi-Channel Radiometer User's Manual*. Gamma Scientific Inc. Aug. 2010.
- [28] The GPflow Contributors. *Gaussian process regression with varying output noise*. URL: https://github.io/GPflow/develop/notebooks/advanced/varying_noise.html (visited on 11/23/2023).
- [29] The GPflow Contributors. *Mean Functions*. URL: https://gpflow.github.io/GPflow/develop/notebooks/getting_started/mean_functions.html (visited on 11/23/2023).
- [30] *UDT Model 211 Photometric Sensor*. Gamma Scientific Inc. 2023. URL: <https://gamma-sci.com/wp-content/uploads/2020/08/UDT-211-Sensor-rev.5.26.20.pdf> (visited on 05/22/2023).
- [31] VPixx Technologies Inc. "Application Guide for VPixx Products Version 1.5." In: 2020.

List of Abbreviations

CLT central limit theorem

CRT cathode ray tube

GP Gaussian process

L_{in} luminance as input

L_{max} maximum luminance

LCD liquid crystal display

LED light emitting diode

LUT lookup table

NIP Neural Information Processing

OTR optical transient tecorder

SGP sparse gaussian process

SQE squared exponential kernel

List of Figures

2.1	Schematic representation of the Setup I	10
2.2	Schematic representation of the Setup II	10
2.3	Schematic representation of the Setup III	11
3.1	Residues of the measured linearized luminance and the ideal linear function, after linear calibration using the current process.	13
3.2	Measured and ideal linearization, when using the current process in (almost) perfect alignment.	13
4.1	Comparison of measured luminance to an ideal gamma dependency corresponding to $\gamma = 2.2$	15
4.2	Detailed view of the lower (a) and upper (b) 10 % of the gamma correction of 2.2 and the measured luminance values.	15
4.3	Comparison of measured luminance and the “corrected” gamma dependency that correspond to $\gamma = 2.2$	17
4.4	Visual example (a) and time graph (b) of the scanning backlight mode from the manual of the VIEWPixx monitor.[13, pp. 20-21]	18
4.5	Overview of the OTR’s placements in relation to the different LED bars of the monitor.	18
4.6	Shifted and aligned measurements of the OTR (Appendix C). To compare the individual placements (Figure 4.5) of the measurements.	19
4.7	Visualization of the different phases of the scanning mode that were achieved through the analyses in Figure 4.6 and consider that the surrounding LED bars are active (reduced) when a LED bar is on.	19
4.8	Photo, that shows phase 10 (see Figure 4.6). The photo was taken using a Fujifilm X100F and the following settings: ISO = 3200, focal length = 23 mm, f-stop = f/2, exposure time = 1/4000 s	20
4.9	The amplitude measured by an OTR during 10 s in the normal (●) and the scanning (●) mode.	21
4.10	A luminance curve of the normal mode which is almost exactly the same as the luminance curve of the scanning mode due to the factor 0.38.	21
4.11	Two measured luminance curves that were measured on the same monitor (VIEWPixx) with the same temperature (35.125 °C).	22
4.12	Temperature measurement \approx 2.5 hours beginning at startup of the monitor. The raw data can be found in Appendix F.8.	23
4.13	Variation of absolute luminance as well as gamma dependency with increasing temperature during warm-up. Internal backlight temperature rises from \approx 30 to 42 °C. The raw data can be found in Appendix F.9.	24
4.14	Absolute Δ between luminance measurements at different temperatures, using the data from Figure 4.13.	24
4.15	Maximal absolute Δ between luminance measurements and their corresponding temperatures, using the data from Figure 4.13.	25

4.16	Temperature measurements by the monitor in the lab, that show influence of change in room cooling. The raw data can be found in Appendix 4.12.	26
4.17	Temperature profile of the VIEWPixx (serial no. 60001-0733) monitor. The raw data can be found in Appendix F.10.	27
4.18	Absolute Δ between luminance measurements at different temperatures (a) and visualizations Maximal absolute Δ between luminance measurements and their corresponding temperatures (b), using the data from Figure 4.17.	27
4.19	Temperature profile of the VIEWPixx /3D Lite (serial no. 65501-3154) monitor. The raw data can be found in Appendix F.11.	28
5.1	Overview about the results from the GP on artificial data representing the gamma dependency.	31
5.2	Exemplary visualization of $\Delta_{L_{out}} = \alpha + \beta \cdot t \cdot (L_{in} - L_{in}^2)$ with $\alpha = 0$ and $\beta = 130$, so that it approximately corresponds to the observed effect in Figure 4.14 and Figure 4.15.	32
5.3	Overview about the results from the GP on artificial data representing the temperature dependence.	33
5.4	Residues of predicted points and “real” data points, when the GP has 64 training points.	34
5.5	Overview about the results from the GP on artificial data representing both, the gamma dependency and the temperature dependence.	35
6.1	Overview of the distribution of the three test runs.	37
6.2	Overview about the results from the GP trained with data from the real experiments.	37
6.3	Overview about the results from the trained GP when it is predicting unknown data points from test runs 1 – 3.	38
6.4	Overview of the distribution of the three test runs ^o .	38
6.5	Overview about the results from the GP trained with data from the experiment to enable linearization of visual stimuli.	39
6.6	Residues of the measured linearized luminance and the ideal linear function, when using the trained model.	39
6.7	Visualization of the functions and the considerable value range of the second experiment.	40
7.1	Visualized design of a calibration process including L_{in} and temperature as predictor variables.	43
D.1	Schematic representation of the double slit orifice. By lining up two slit apertures, scattered light is blocked each time and (nearly) only direct light is able to reach the OTR.	60
D.2	Measurements of the OTR at a placement comparable with  with a double slit orifice between the monitor and the OTR.	60
E.3	Residues of measurements during the normal (a) and the scanning mode (b).	61
E.4	Residues of measurements during the normal mode ($\hat{\sigma} \approx 0.1$).	61
J.5	Absolute Δ between luminance measurements at different temperatures (scanning mode)	72
L.6	Overview about the results from the GP using a mean function with weight factor α on artificial data representing both, the gamma dependency and the temperature dependence.	77
L.7	Overview about the results from the GP using only a mean function for the gamma dependence on artificial data representing both, the gamma dependency and the temperature dependence.	78
L.8	Plot of $L_{out} = L_{in}^{50.86926}$	80

Code

4.1	Simplified and reduced code to set the luminance of the monitor.	14
5.1	Code to define a SQE.	31
H.1	Code generating artificial data with the gamma dependency and creating a GP model.	68
H.2	Code generating artificial data with the temperature dependence and creating a GP model.	68
H.3	Code generating artificial data with both dependencies and creating a GP model.	69
H.4	Code generating artificial data with both dependencies and creating a GP model using the <code>ViewpixxAlpha</code> mean function.	69
H.5	Code generating artificial data with both dependencies and creating a GP model using only the <code>Gamma2DGeneric</code> mean function.	70
K.6	Code of the <code>Gamma2DGeneric</code> mean function that allows to set generic gamma dependency as a mean function in the GPflow Framework [19].	73
K.7	Code of the <code>Temp2DGeneric</code> mean function that allows to set generic temperature dependency as a mean function in the GPflow Framework [19].	74
K.8	Code of the <code>ViewpixxAlpha</code> mean function that allows to set both dependencies and uses a weight factor α in the GPflow Framework [19].	75
K.9	Code to define a likelihood that enables a (with L_{in}) linearly increasing Gaussian noise.	76

List of Tables

5.1	Optimization result of the hyperparameter by the GP with artificial data representing the gamma dependency.	32
5.2	Optimization result of the hyperparameter by the GP with artificial data representing the temperature dependence.	33
5.3	Optimization result of the hyperparameter by the GP with artificial data representing both, the gamma dependency and the temperature dependence.	35
I.1	Overview of σ of experiments on the VIEWPixed / 3D Lite monitor	71
L.2	Optimization result of the hyperparameter by the GP using a mean function with weight factor α on artificial data representing both, the gamma dependency and the temperature dependence.	77
L.3	Optimization result of the hyperparameter by the GP using only a mean function for the gamma dependence on artificial data representing both, the gamma dependency and the temperature dependence.	78
L.4	Optimization result of the hyperparameter by the GP trained with data from real experiments.	79
L.5	Optimization result of the hyperparameter by the GP trained with real data to enable linearization of visual stimuli.	80

Appendix

Contents

A	Log of a calibration process with 12 bit resolution	56
B	Original code sample to set L_{in}	57
C	Measurements of the OTR	58
C.1	Measurements of the OTR, taken at the placement 	58
C.2	Measurements of the OTR, taken at the placement 	58
C.3	Measurements of the OTR, taken at the placement 	58
C.4	Measurements of the OTR, taken at the placement 	58
C.5	Measurements of the OTR, taken at the placement 	59
D	Ensuring there is no optical scattering between LED bars in the OTR measurements	60
E	Noise difference between modes	61
F	Raw data for figures	63
F.1	Raw data of Figure 3.1	63
F.2	Raw data of Figure 4.1	63
F.3	Raw data of Figure E.3 (a)	63
F.4	Raw data of Figure E.3 (b)	63
F.5	Raw data of Figure 4.9	63
F.6	Raw data of the normal curve in Figure 4.11	63
F.7	Raw data of the scanning curve in Figure 4.11	63
F.8	Raw data of Figure 4.12	64
F.9	Raw data of Figure 4.13	64
F.10	Raw data of Figure 4.17	64
F.11	Raw data of Figure 4.19	64
F.12	Raw data for the validation of the model on real measurements Chapter 6	64
F.13	Raw data for training the model to be able to linearize in Chapter 6.2	64
F.14	Raw data for testing the linearization with the model in Chapter 6.2	65
G	Used version of GPflow and dependent packages	66
H	Creation of artificial data	68
H.1	Artificial Data to test the created Gamma2DGeneric mean function	68
H.2	Artificial Data to test the created Temp2DGeneric mean function	68
H.3	Artificial Data to test the combination of both effects	68
H.4	Artificial Data to test the created ViewpixxAlpha mean function	69
H.5	Artificial Data to test the combination of both effects but only uses the Gamma2DGeneric mean function	69
I	Overview of standard deviation of experiments	71
J	Comparison of shape changes in the luminance characteristic with different modes	72
K	Created mean and noise functions for GPs	73
K.1	Mean function to map the gamma dependency	73

K.2	Mean function to map the temperature dependence effect	73
K.3	Mean function to map both dependencies and uses a weight factor α	74
K.4	Function to enable a likelihood with a linear increasing Gaussian noise	76
L	Result of artificial data experiments	77
L.1	Mean function with weight factor α	77
L.2	Only gamma mean function but both effects	78
L.3	Real training data	79
L.4	Trainings data for the Linearization	80

A Log of a calibration process with 12 bit resolution.

```

1 | Tue 30 May 2023 11:01:18 AM CEST
2 | Setup monitor Viewpixx, search for photometer S470 ...
3 | Measure a few black and white screens ...
4 |   1/6 At DAC value 1.000      : 97.28cd/m^2
5 |   2/6 At DAC value 1.000      : 97.28cd/m^2
6 |   3/6 At DAC value 1.000      : 97.27cd/m^2
7 |   4/6 At DAC value 0.000      : 0.28cd/m^2
8 |   5/6 At DAC value 0.000      : 0.28cd/m^2
9 |   6/6 At DAC value 0.000      : 0.28cd/m^2
10 | Measure luminance series ...
11 |   1/4096 At DAC value 0.000    : 0.28cd/m^2
12 |   2/4096 At DAC value 0.000    : 0.28cd/m^2
13 |   3/4096 At DAC value 0.000    : 0.28cd/m^2
14 |   4/4096 At DAC value 0.001    : 0.28cd/m^2
15 |   5/4096 At DAC value 0.001    : 0.28cd/m^2
16 |   6/4096 At DAC value 0.001    : 0.28cd/m^2
17 |   7/4096 At DAC value 0.001    : 0.28cd/m^2
18 |   8/4096 At DAC value 0.002    : 0.29cd/m^2
19 |   9/4096 At DAC value 0.002    : 0.29cd/m^2
20 |   10/4096 At DAC value 0.002   : 0.29cd/m^2
21 |
22 | Time-Estimation:
23 | We needed 00:00:37 until now (10 levels).
24 | This results into 00:00:03 per level.
25 | Estimated ending time: 30.05.23 15:14:54
26 |
27 |   11/4096 At DAC value 0.002    : 0.29cd/m^2
28 |   12/4096 At DAC value 0.003    : 0.29cd/m^2
29 |   13/4096 At DAC value 0.003    : 0.29cd/m^2
30 |   14/4096 At DAC value 0.003    : 0.29cd/m^2
31 |
32 | ...
33 |
34 | Save new monitor calibration ...
35 | Plot measurements ...
36 | Save plot 2023_05_30 15:14_luminance.pdf...
37 | Save plot 2023_05_30 15:14_resolution.pdf...
38 | Done.
39 | Tue 30 May 2023 03:17:10 PM CEST

```

B Original code sample to set L_{in}

Original code sample to set L_{in} . The complete code can be found here: https://github.com/wichmann-lab/psychopy-pixx/blob/main/psychopy_pixx/calibration/calibration.py

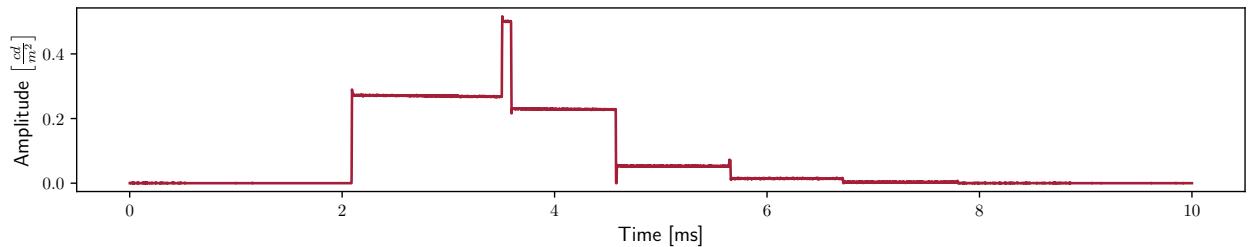
```

10 | def measure_luminances(
11 |     levels,
12 |     window,
13 |     photometer,
14 |     gamma=1.0,
15 |     allGuns=True,
16 |     autoMode='auto',
17 |     random=False,
18 |     stimSize=4,
19 |     n_measures=50):
20 |
21 |     ...
22 |
23 |     if allGuns:
24 |         guns = [0, 1, 2, 3] # gun=0 is the white luminance measure
25 |     else:
26 |         guns = [0]
27 |     # this will hold the measured luminance values
28 |     lumsList = np.zeros((4, len(toTest)))
29 |     # for each gun, for each value run test
30 |     for gun in guns:
31 |         for valN, DACval in enumerate(toTest):
32 |             lum = (DACval * 2) - 1 # from range 0:1 into -1:1
33 |             # only do luminanc=-1 once
34 |             if lum == -1 and gun > 0:
35 |                 continue
36 |             # set the patch color
37 |             if gun > 0:
38 |                 rgb = [-1, -1, -1]
39 |                 rgb[gun - 1] = lum
40 |             else:
41 |                 rgb = [lum, lum, lum]
42 |
43 |             backPatch.draw()
44 |             testPatch.setColor(rgb)
45 |             testPatch.draw()
46 |
47 |     backPatch.draw()
48 |     testPatch.setColor([1, 1, 1])
49 |     testPatch.draw()
50 |
51 |     return lumsList
52 |
53 | 
```

C Measurements of the OTR

The OTR obtained the following records from the experiment.

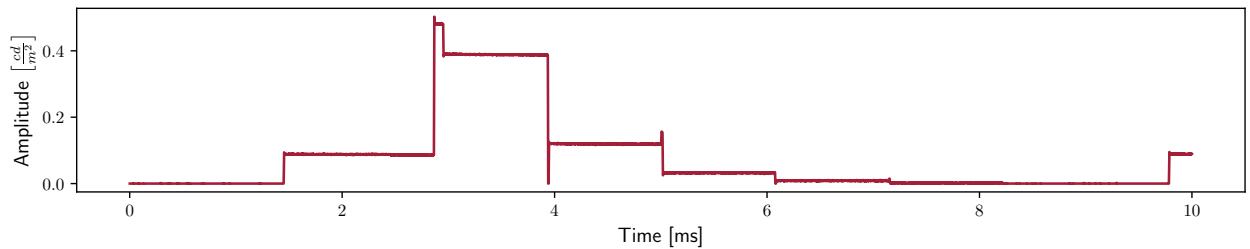
C.1 Measurements of the OTR, taken at the placement ●



The raw data can be accessed through the following link:

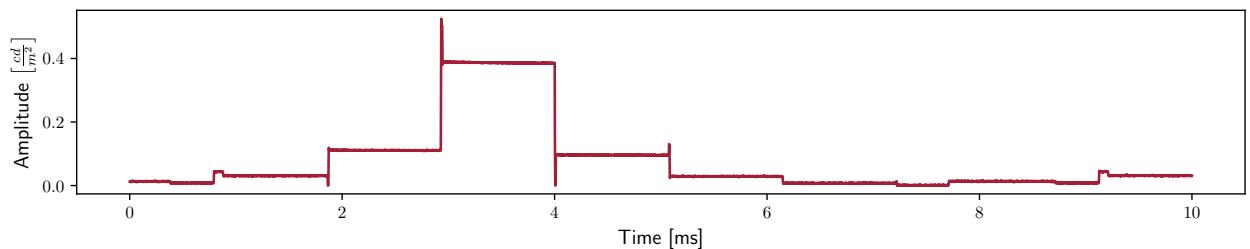
https://github.com/JSchmiegel/MasterThesis/tree/main/data/017otrScanNear/007_1382cm_oo.csv

C.2 Measurements of the OTR, taken at the placement ●



https://github.com/JSchmiegel/MasterThesis/tree/main/data/017otrScanNear/008_1358cm_oo.csv

C.3 Measurements of the OTR, taken at the placement ●

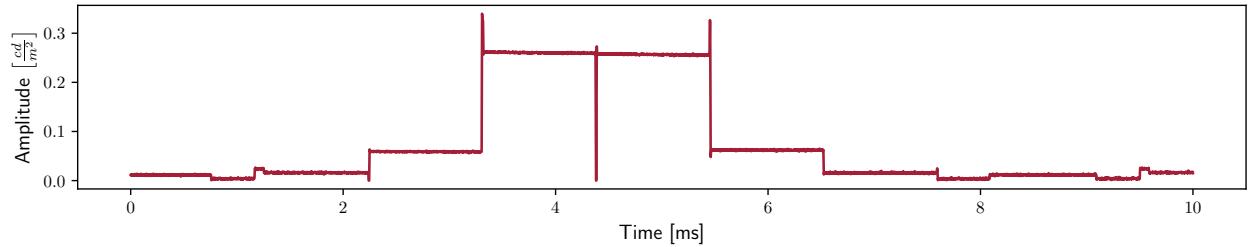


The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/017otrScanNear/009_1277cm_oo.csv

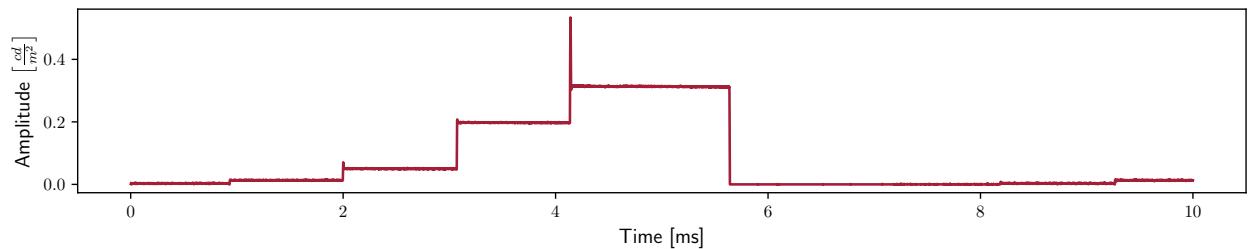
C.4 Measurements of the OTR, taken at the placement ●

The raw data can be accessed through the following link:



https://github.com/JSchmiegel/MasterThesis/tree/main/data/017otrScanNear/010_1265cm_oo.csv

C.5 Measurements of the OTR, taken at the placement ☺



The raw data can be accessed through the following link:
https://github.com/JSchmiegel/MasterThesis/tree/main/data/017otrScanNear/011_1126cm_oo.csv

D Ensuring there is no optical scattering between LED bars in the OTR measurements

To ensure that optical scattering from the individual LED bars is not the cause of these observations, the measurements were repeated by using [Setup III](#) and a double slit orifice positioned between the OTR and the monitor. The double slit orifice allows primarily direct light to reach the OTR, with minimal scattering. Figure D.1 gives an overview how the double slit orifice functions. This resulted in a measurement (Figure D.2) that had reduced noise¹ but did not lead to the elimination of clear gradations in the measurement, thus still making other phases visible.

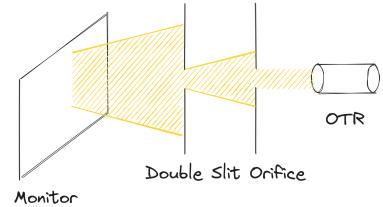
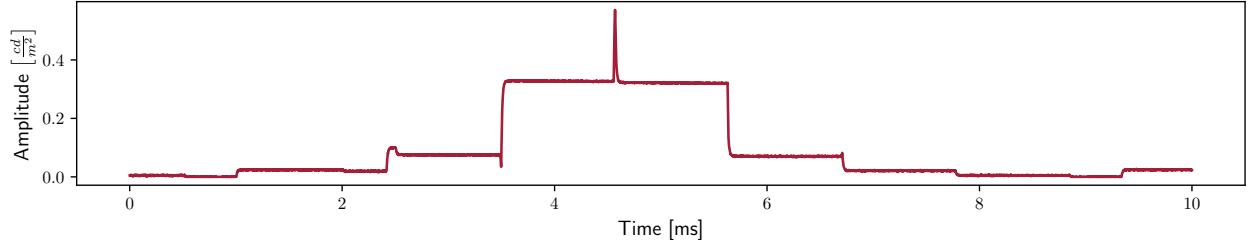


Figure D.1: Schematic representation of the double slit orifice. By lining up two slit apertures, scattered light is blocked each time and (nearly) only direct light is able to reach the OTR.

¹ Compare Figure D.2 and the figure in Appendix C.3 as they were captured at nearly the same height.



The raw data can be accessed through the following link:
https://github.com/JSchmiegel/MasterThesis/tree/main/data/0190TRLichtfalle/004_1284cm.csv

Figure D.2: Measurements of the OTR at a placement comparable with with a double slit orifice between the monitor and the OTR.

E Noise difference between modes

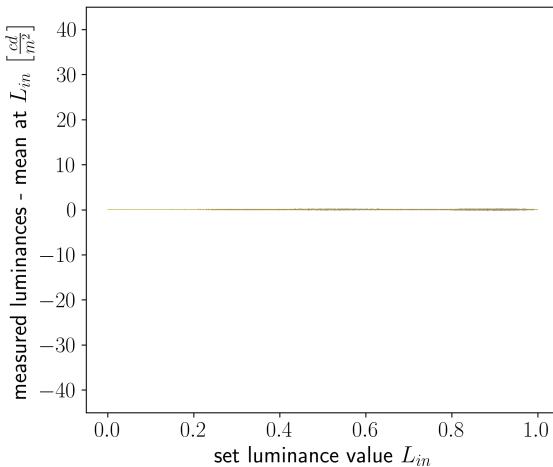
In our experiments we measured 1000 times the luminance at a specific L_{in} value. The reason is that the single measurements of one L_{in} do vary very strong. The primary factor contributing to this variation is the distinct modes utilized, as the magnitude of the variation between different measurements of a single L_{in} value differs significantly when comparing scanning mode to normal mode.² This has been demonstrated by two experiments: one using measurements taken in scanning mode³ and the other using measurements taken in normal mode⁴. Both experiments consisted of 1000 measurements for each L_{in} . Figure 4.9 displays the extent of variation among the 1000 measurements.⁵

³ Monitor: VIEWPixx /3D Lite
Experiment setup: *Setup I*
Mode: *scanning*
Temperature: Ø 46.2 °C
Distance: *Appendix F.4*

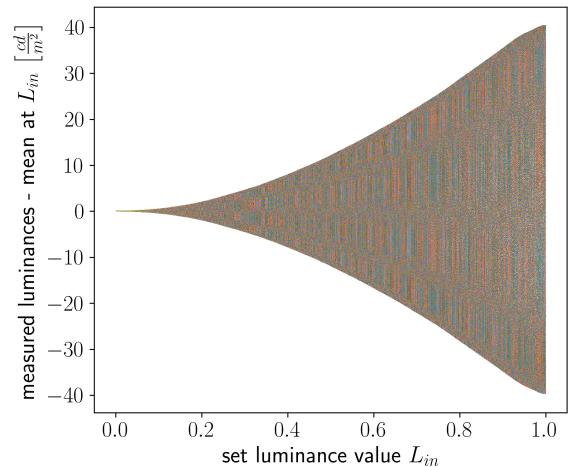
⁴ Monitor: VIEWPixx /3D Lite
Experiment setup: *Setup I*
Mode: *normal*
Temperature: Ø 36.87 °C
Distance: *Appendix F.3*

² More information about the modes can be found in Chapter 4.2

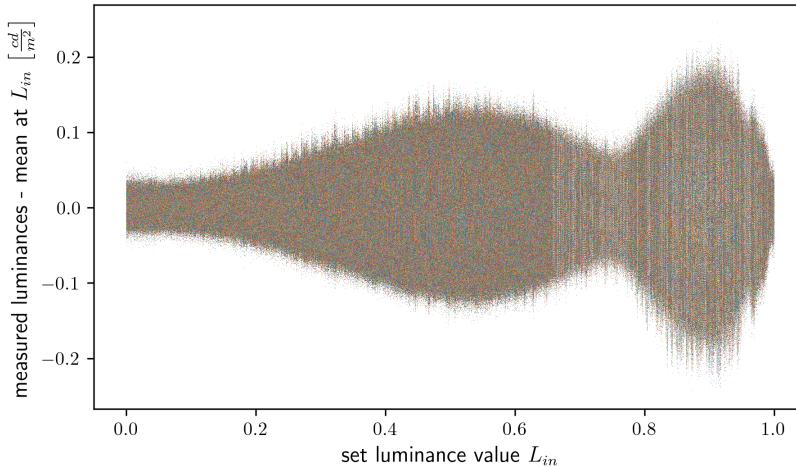
⁵ To compare the two modes, we standardized the Y-axis scaling. Figure E.4 provides an unscaled, zoomed-in view of Figure E.3 (a).



(a) Residues of measurements during the normal mode ($\hat{\sigma} \approx 0.1$).



(b) Residues of measurements during the scanning mode ($\hat{\sigma} \approx 28.4$).



The raw data of the measurement in normal mode can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/>

Figure E.3: Residues of measurements during the normal (a) and the scanning mode (b).

Figure E.4: Residues of measurements during the normal mode ($\hat{\sigma} \approx 0.1$).

008levelCollection02/Test09/allMeasurements2023-07-10_20-30.

csv

The raw data of the measurement in scanning mode can be accessed through the following link:

[https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test09/allMeasurements2023-06-24_09-23.](https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test09/allMeasurements2023-06-24_09-23.csv)

csv

F Raw data for figures

F.1 Raw data of Figure 3.1

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/029linearizationViewpixx3D/Test01/>

F.2 Raw data of Figure 4.1

The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/008levelCollection02/Test07/allMeasurements2023-07-10_10-14.csv

F.3 Raw data of Figure E.3 (a)

The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/008levelCollection02/Test09/allMeasurements2023-07-10_20-30.csv

F.4 Raw data of Figure E.3 (b)

The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test09/allMeasurements2023-06-24_09-23.csv

F.5 Raw data of Figure 4.9

The raw data of 10 s in the normal mode can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/0200TRAmplitude/001normal_10s.csv

The raw data of 10 s in the scanning mode can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/0200TRAmplitude/002scanning_10s.csv

F.6 Raw data of the normal curve in Figure 4.11

The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/015tempDependency1/Test05/allMeasurements2023-08-09_12-51.csv

F.7 Raw data of the scanning curve in Figure 4.11

The raw data can be accessed through the following link:

https://github.com/JSchmiegel/MasterThesis/tree/main/data/015tempDependency2/Test16/allMeasurements2023-08-09_22-32.csv

F.8 Raw data of Figure 4.12

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/temp/>

F.9 Raw data of Figure 4.13

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/014tempDependency1.6/>

F.10 Raw data of Figure 4.17

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/015tempDependency1/>

F.11 Raw data of Figure 4.19

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/026tempDependency/>

F.12 Raw data for the validation of the model on real measurements Chapter 6

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test02/>,

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test05/> and

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/Test08/>

F.13 Raw data for training the model to be able to linearize in Chapter 6.2

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/027lastExperimentPart1/Test01/>

F.14 Raw data for testing the linearization with the model in Chapter 6.2

The raw data can be accessed through the following link:

[https://github.com/JSchmiegel/MasterThesis/tree/main/
data//028lastExperimentPart2_b/Test01/](https://github.com/JSchmiegel/MasterThesis/tree/main/data//028lastExperimentPart2_b/Test01/)

G Used version of GPflow and dependent packages

```

gpflow==2.9.0
|-- check-shapes [required: >=1.0.0, installed: 1.1.1]
|  |-- dropstackframe [required: >=0.1.0, installed: 0.1.0]
|  |-- lark [required: >=1.1.0,<2.0.0, installed: 1.1.8]
|-- Deprecated [required: Any, installed: 1.2.14]
|  |-- wrapt [required: >=1.10,<2, installed: 1.14.1]
|-- muledispatch [required: >=0.6, installed: 1.0.0]
|-- numpy [required: Any, installed: 1.23.5]
|-- packaging [required: Any, installed: 23.1]
|-- scipy [required: Any, installed: 1.10.1]
|  |-- numpy [required: >=1.19.5,<1.27.0, installed: 1.23.5]
|-- setuptools [required: >=41.0.0, installed: 58.1.0]
|-- tabulate [required: Any, installed: 0.9.0]
|-- tensorflow [required: >=2.4.0, installed: 2.14.0]
|  |-- absl-py [required: >=1.0.0, installed: 2.0.0]
|  |-- astunparse [required: >=1.6.0, installed: 1.6.3]
|  |  |-- six [required: >=1.6.1,<2.0, installed: 1.16.0]
|  |  |-- wheel [required: >=0.23.0,<1.0, installed: 0.41.2]
|  |-- flatbuffers [required: >=23.5.26, installed: 23.5.26]
|  |-- gast [required: >=0.2.1,!>0.5.2,!>0.5.1,!>0.5.0, installed: 0.5.4]
|  |-- google-pasta [required: >=0.1.1, installed: 0.2.0]
|  |  |-- six [required: Any, installed: 1.16.0]
|  |-- grpcio [required: >=1.24.3,<2.0, installed: 1.59.0]
|  |-- h5py [required: >=2.9.0, installed: 3.10.0]
|  |  |-- numpy [required: >=1.17.3, installed: 1.23.5]
|  |-- keras [required: >=2.14.0,<2.15, installed: 2.14.0]
|  |-- libclang [required: >=13.0.0, installed: 16.0.6]
|  |-- ml-dtypes [required: ==0.2.0, installed: 0.2.0]
|  |  |-- numpy [required: >1.20, installed: 1.23.5]
|  |-- numpy [required: >=1.23.5, installed: 1.23.5]
|  |-- opt-einsum [required: >=2.3.2, installed: 3.3.0]
|  |  |-- numpy [required: >=1.7, installed: 1.23.5]
|  |-- packaging [required: Any, installed: 23.1]
|  |-- protobuf [required: >=3.20.3,<5.0.0dev,!>4.21.5,!>4.21.4,!>4.21.3]
→ ,!=4.21.2,!>4.21.1,!>4.21.0, installed:
→ 4.24.4
|  |-- setuptools [required: Any, installed: 58.1.0]
|  |-- six [required: >=1.12.0, installed: 1.16.0]
|  |-- tensorboard [required: >=2.14,<2.15, installed: 2.14.1]
|  |  |-- absl-py [required: >=0.4, installed: 2.0.0]
|  |  |-- google-auth [required: >=1.6.3,<3, installed: 2.23.3]
|  |  |  |-- cachetools [required: >=2.0.0,<6.0, installed: 5.3.1]
|  |  |  |-- pyasn1-modules [required: >=0.2.1, installed: 0.3.0]
|  |  |  |  |-- pyasn1 [required: >=0.4.6,<0.6.0, installed: 0.5.0]
|  |  |  |-- rsa [required: >=3.1.4,<5, installed: 4.9]
|  |  |  |  |-- pyasn1 [required: >=0.1.3, installed: 0.5.0]
|  |  |  |-- google-auth-oauthlib [required: >=0.5,<1.1, installed: 1.0.0]
|  |  |  |  |-- google-auth [required: >=2.15.0, installed: 2.23.3]
|  |  |  |  |  |-- cachetools [required: >=2.0.0,<6.0, installed: 5.3.1]
|  |  |  |  |  |-- pyasn1-modules [required: >=0.2.1, installed: 0.3.0]
|  |  |  |  |  |  |-- pyasn1 [required: >=0.4.6,<0.6.0, installed:
→ 0.5.0]
|  |  |  |  |-- rsa [required: >=3.1.4,<5, installed: 4.9]
|  |  |  |  |  |-- pyasn1 [required: >=0.1.3, installed: 0.5.0]
|  |  |  |  |-- requests-oauthlib [required: >=0.7.0, installed: 1.3.1]
|  |  |  |  |  |-- oauthlib [required: >=3.0.0, installed: 3.2.2]
|  |  |  |  |  |-- requests [required: >=2.0.0, installed: 2.31.0]
|  |  |  |  |  |  |-- certifi [required: >=2017.4.17, installed:
→ 2023.7.22]
|  |  |  |  |-- charset-normalizer [required: >=2,<4, installed:
→ 3.3.1]
|  |  |  |  |  |-- idna [required: >=2.5,<4, installed: 3.4]
|  |  |  |  |  |-- urllib3 [required: >=1.21.1,<3, installed: 2.0.7]
|  |  |  |  |  |-- grpcio [required: >=1.48.2, installed: 1.59.0]
|  |  |  |  |  |-- Markdown [required: >=2.6.8, installed: 3.5]
|  |  |  |  |  |  |-- importlib-metadata [required: >=4.4, installed: 6.6.0]

```

```
| | |     |-- zipp [required: >=0.5, installed: 3.15.0]
| | |     |-- numpy [required: >=1.12.0, installed: 1.23.5]
| | |     |-- protobuf [required: >=3.19.6, installed: 4.24.4]
| | |     |-- requests [required: >=2.21.0,<3, installed: 2.31.0]
| | |     |    |-- certifi [required: >=2017.4.17, installed: 2023.7.22]
| | |     |    |-- charset-normalizer [required: >=2,<4, installed: 3.3.1]
| | |     |    |-- idna [required: >=2.5,<4, installed: 3.4]
| | |     |    |-- urllib3 [required: >=1.21.1,<3, installed: 2.0.7]
| | |     |-- setuptools [required: >=41.0.0, installed: 58.1.0]
| | |     |-- six [required: >1.9, installed: 1.16.0]
| | |     |-- tensorboard-data-server [required: >=0.7.0,<0.8.0, installed:
→ 0.7.1]
| | |     |-- werkzeug [required: >=1.0.1, installed: 3.0.0]
| | |         |-- MarkupSafe [required: >=2.1.1, installed: 2.1.3]
| | |-- tensorflow-estimator [required: >=2.14.0,<2.15, installed: 2.14.0]
| | |-- tensorflow-io-gcs-filesystem [required: >=0.23.1, installed:
→ 0.34.0]
| | |-- termcolor [required: >=1.1.0, installed: 2.3.0]
| | |-- typing-extensions [required: >=3.6.6, installed: 4.5.0]
| | |-- wrapt [required: >=1.11.0,<1.15, installed: 1.14.1]
|-- tensorflow-probability [required: >=0.12.0, installed: 0.22.0]
| |-- absl-py [required: Any, installed: 2.0.0]
| |-- cloudpickle [required: >=1.3, installed: 3.0.0]
| |-- decorator [required: Any, installed: 5.1.1]
| |-- dm-tree [required: Any, installed: 0.1.8]
| |-- gast [required: >=0.3.2, installed: 0.5.4]
| |-- numpy [required: >=1.13.3, installed: 1.23.5]
| |-- six [required: >=1.10.0, installed: 1.16.0]
| |-- typing-extensions [required: <4.6.0, installed: 4.5.0]
-- typing-extensions [required: Any, installed: 4.5.0]
pip==23.3.1
```

H Creation of artificial data

H.1 Artificial Data to test the created *Gamma2DGeneric* mean function

```

1 df = pd.DataFrame()
2 X = np.linspace(0.0, 1.0, resolution)
3 df['L_in'] = X
4 df['temps'] = np.zeros(resolution) + np.random.normal(loc=0, scale=0.01,
   ↵ size=resolution)
5 df['L_out'] = 0.24+95*X**2.2 + np.random.normal(loc=0, scale=2,
   ↵ size=resolution)
6
7 kernel = k.SquaredExponential()
8 mean_funct = Gamma2DGeneric(gamma=2.0)
9 likelihood = gpflow.likelihoods.Gaussian()
```

Code H.1: Code generating artificial data with the gamma dependency and creating a [GP](#) model.

H.2 Artificial Data to test the created *Temp2DGeneric* mean function

```

1 df = pd.DataFrame()
2 X = np.linspace(0.0, 1.0, resolution)
3 temps = np.random.permutation((4*np.linspace(30, 40,
   ↵ resolution)).round() / 4)
4 tempmin = temps.min()
5 temprange = temps.max()-tempmin
6 tnorm = (temps-tempmin)/temprange
7 w = [0, 130]
8
9 df['L_in'] = X
10 df['temps'] = temps
11 df['L_out'] = tempponly(df['L_in'], tnorm, w) + np.random.normal(loc=0,
   ↵ scale=0.3, size=resolution)
12
13 kernel = k.SquaredExponential()
14 mean_funct = Temp2DGeneric()
15 likelihood = gpflow.likelihoods.Gaussian()
```

Code H.2: Code generating artificial data with the temperature dependence and creating a [GP](#) model.

H.3 Artificial Data to test the combination of both effects

```

1 df = pd.DataFrame()
2 X = np.linspace(0.0, 1.0, resolution)
3 temps = np.random.permutation((4*np.linspace(30, 40,
4   ↪ resolution)).round() / 4)
5 tempmin = temps.min()
6 temprange = temps.max()-tempmin
7 tnorm = (temps-tempmin)/temprange
8 w = [0, 130]
9
10 df['L_in'] = X
11 df['temps'] = temps
12 df['L_out'] = (0.24 + 95 * X **2.2) - tempponly(df['L_in'], tnorm, w) +
13   ↪ np.random.normal(loc=0, scale=0.3, size=resolution)
14
15 kernel = k.SquaredExponential()
16 mean_funct = Gamma2DGeneric(gamma=2.2) + Temp2DGeneric()
17 likelihood = gpflow.likelihoods.Gaussian()

```

Code H.3: Code generating artificial data with both dependencies and creating a [GP](#) model.

H.4 Artificial Data to test the created *ViewpixxAlpha* mean function

```

1 df = pd.DataFrame()
2 X = np.linspace(0.0, 1.0, resolution)
3 temps = np.random.permutation((4*np.linspace(30, 40,
4   ↪ resolution)).round() / 4)
5 tempmin = temps.min()
6 temprange = temps.max()-tempmin
7 tnorm = (temps-tempmin)/temprange
8 w = [0, 30]
9
10 df['L_in'] = X
11 df['temps'] = temps
12 df['L_out'] = (0.24 + 95 * X **2.2) - tempponly(df['L_in'], tnorm, w) +
13   ↪ np.random.normal(loc=0, scale=0.3, size=resolution)
14
15 kernel = k.SquaredExponential()
16 mean_funct = ViewpixxAlpha(gamma=2.2, alpha=0.9)
17 likelihood = gpflow.likelihoods.Gaussian()

```

Code H.4: Code generating artificial data with both dependencies and creating a [GP](#) model using the *ViewpixxAlpha* mean function.

H.5 Artificial Data to test the combination of both effects but only uses the *Gamma2DGeneric* mean function

```

1 df = pd.DataFrame()
2 X = np.linspace(0.0, 1.0, resolution)
3 temps = np.random.permutation((4*np.linspace(30, 40,
4   ↪ resolution)).round() / 4)
5 tempmin = temps.min()
6 temprange = temps.max()-tempmin
7 tnorm = (temps-tempmin)/temprange
8 w = [0, 130]
9
10 df['L_in'] = X
11 df['temps'] = temps
12 df['L_out'] = (0.24 + 95 * X **2.2) - tempponly(df['L_in'], tnorm, w) +
13   ↪ np.random.normal(loc=0, scale=0.3, size=resolution)
14
15 kernel = k.SquaredExponential(active_dims=[0]) +
16   ↪ k.SquaredExponential(active_dims=[1])
17 mean_funct = Gamma2DGeneric(gamma=2.2)
18 likelihood = gpflow.likelihoods.Gaussian()

```

Code H.5: Code generating artificial data with both dependencies and creating a GP model using only the Gamma2DGeneric mean function.

I Overview of standard deviation of experiments

The table below provides an overview of the standard deviation (σ) of experiments. The σ was calculated by obtaining a moving average⁶ from the measurement data of the experiment, subtracting it, and calculating the magnitude of the resulting residues σ .

⁶ window size = 20

Intern Experiment Name (mode, intensity, order of <i>Lin</i>)	Temp diff [°C]	σ of Experiment $\left[\frac{cd}{m^2}\right]$
004 Test1 (sc, 255, ascending)	0.250	0.123783
004 Test2 (sc, 255, random)	0.625	0.269525
004 Test3 (sc, 255, descending)	1.250	0.119260
004 Test4 (sc, 255, ascending)	0.750	0.116673
004 Test5 (sc, 255, random)	0.125	1.058952
004 Test6 (sc, 255, descending)	1.250	0.116119
004 Test7 (sc, 255, ascending)	1.000	0.122196
004 Test8 (sc, 255, random)	0.625	0.267561
004 Test9 (sc, 255, descending)	0.500	0.115316
008 Test7 (nm, 255, ascending)	0.750	0.359017
008 Test8 (nm, 255, random)	1.750	1.245257
008 Test9 (nm, 255, descending)	0.625	0.376719
010 Test1 (nm, 127, ascending)	2.750	0.137439
010 Test2 (nm, 127, random)	1.250	0.611973
010 Test3 (nm, 127, descending)	0.125	0.134251
012 Test1 (sc, 255, ascending)	8.750	0.113699
012 Test2 (sc, 255, random)	1.000	0.316309
012 Test3 (sc, 255, descending)	1.000	0.119838
012 Test4 (nm, 255, ascending)	5.000	0.347119
012 Test5 (nm, 255, random)	0.125	0.404985
012 Test6 (nm, 255, descending)	1.375	0.358990
012 Test7 (nm, 111, ascending)	0.125	0.116606
012 Test8 (nm, 111, random)	0.500	0.219490
012 Test9 (nm, 111, descending)	0.625	0.112587
average $\bar{\sigma}$	1.3271	0.4347

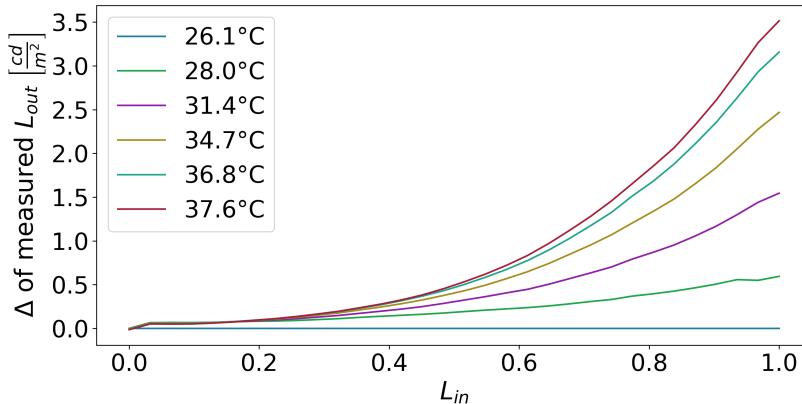
Table I.1: Overview of σ of experiments on the VIEWPixed /3D Lite monitor

The raw data can be accessed through the following links:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/004bigDataCollection/>,
<https://github.com/JSchmiegel/MasterThesis/tree/main/data/008bigDataCollection02/>,
<https://github.com/JSchmiegel/MasterThesis/tree/main/data/010intesityCollection/> and
<https://github.com/JSchmiegel/MasterThesis/tree/main/data/012newFan/>

J Comparison of shape changes in the luminance characteristic with different modes

The experiment carried out for Figure 4.17 was repeated⁷ in the same form in the scanning mode and resulted in the following overview of the deviations (Figure J.5). Comparing Figure 4.14 and Figure J.5 the shape of the change in the luminance characteristic appears to be the same. Only the absolute values differ.



⁷ Monitor: VIEWPixx

Experiment setup: *Setup I*

Mode: *scanning*

Number of L_{in} : 32

Temperature: 26.125 - 37.625 °C

Distance: 30 cm

Raw data: see at the end of this chapter

Figure J.5: Absolute Δ between luminance measurements at different temperatures (scanning mode)

The raw data can be accessed through the following link:

<https://github.com/JSchmiegel/MasterThesis/tree/main/data/015tempDependency2/>

K Created mean and noise functions for GPs

K.1 Mean function to map the gamma dependency

```

1 | class Gamma2DGeneric(gpflow.functions.MeanFunction,
2 |   ↪ gpflow.functions.Function):
3 |   """
4 |   y_i = A * x0_i ** gamma + b
5 |   """
6 |
7 |   @check_shapes(
8 |     "A: [broadcast D, broadcast Q]",
9 |     "b: [broadcast Q]",
10 |   )
11 |   def __init__(self, A: TensorType = None, b: TensorType = None,
12 |   ↪ gamma: TensorType = None) -> None:
13 |     """
14 |     A is a matrix which maps each element of X to Y, b is an additive
15 |     constant and gamma the gamma exponent.
16 |     """
17 |     gpflow.functions.MeanFunction.__init__(self)
18 |     A = np.ones((1, 1), dtype=default_float()) if A is None else A
19 |     b = np.zeros(1, dtype=default_float()) if b is None else b
20 |     gamma = 2.2 if gamma is None else gamma
21 |     self.A = gpflow.Parameter(np.atleast_2d(A))
22 |     self.b = gpflow.Parameter(b)
23 |     self.gamma = gpflow.Parameter(gamma)
24 |
25 |   @inherit_check_shapes
26 |   def __call__(self, X: TensorType) -> tf.Tensor:
27 |     X1 = X[:, 0]
28 |     X1 = tf.reshape(X1, (-1, 1))
29 |     return tf.tensordot(X1**self.gamma, self.A, [[-1], [0]]) + self.b

```

Code K.6: Code of the Gamma2DGeneric mean function that allows to set generic gamma dependency as a mean function in the GPflow Framework [19].

K.2 Mean function to map the temperature dependence effect

```

1  class Temp2DGeneric(gpflow.functions.MeanFunction,
2      ↪ gpflow.functions.Function):
3      """
4          y_i = c + A * (x0_i * x1_i - x0_i**2 * x1_i)
5          """
6
7      @check_shapes(
8          "c: [broadcast Q]",
9          "A: [broadcast D, broadcast Q]",
10     )
11     def __init__(self, c: TensorType = None, A: TensorType = None) ->
12         None:
13             """
14                 A is a matrix which maps each element of X to Y, b is an additive
15                 constant c.
16             """
17
18     gpflow.functions.MeanFunction.__init__(self)
19     c = np.zeros(1, dtype=default_float()) if c is None else c
20     A = np.ones((1, 1), dtype=default_float()) if A is None else A
21     self.c = gpflow.Parameter(c)
22     self.A = gpflow.Parameter(np.atleast_2d(A))
23
24     @inherit_check_shapes
25     def __call__(self, X: TensorType) -> tf.Tensor:
26
27         X0 = X[:, 0]
28         X0 = tf.reshape(X0, (-1, 1))
29         X1 = X[:, 1]
30         X1 = tf.reshape(X1, (-1, 1))
31
32         return self.c + tf.tensordot(X0*X1, self.A, [[-1], [0]]) +
33             tf.tensordot(X0**2*X1, -self.A, [[-1], [0]])

```

Code K-7: Code of the `Temp2DGeneric` mean function that allows to set generic temperature dependency as a mean function in the GPflow Framework [19].

K.3 Mean function to map both dependencies and uses a weight factor α

```

1  class ViewpixxAlpha(gpflow.functions.MeanFunction,
2      ↪ gpflow.functions.Function):
3      """
4          y_i = alpha * (c1 + A * x0_i ** gamma) +
5              (1-alpha) (c2 + B * x0_i * x1_i -B * x0_i**2 * x1_i)
6
7      """
8
9      @check_shapes(
10          "gamma: [broadcast Q]",
11          "alpha: [broadcast Q]",
12          "c1: [broadcast Q]",
13          "c2: [broadcast Q]",
14          "A: [broadcast D, broadcast Q]",
15          "B: [broadcast D, broadcast Q]",
16      )
17      def __init__(self, gamma: TensorType = None, alpha: TensorType =
18          ↪ None, c1: TensorType = None, c2: TensorType = None, A:
19          ↪ TensorType = None, B: TensorType = None) -> None:
20
21          gpflow.functions.MeanFunction.__init__(self)
22          gamma = 2.2 if gamma is None else gamma
23          alpha = np.zeros(1, dtype=default_float()) if alpha is None else
24              ↪ alpha
25          c1 = np.zeros(1, dtype=default_float()) if c1 is None else c1
26          c2 = np.zeros(1, dtype=default_float()) if c2 is None else c2
27          A = np.ones((1, 1), dtype=default_float()) if A is None else A
28          B = np.ones((1, 1), dtype=default_float()) if B is None else B
29          self.gamma = gpflow.Parameter(gamma)
30          self.alpha = gpflow.Parameter(alpha)
31          self.c1 = gpflow.Parameter(c1)
32          self.c2 = gpflow.Parameter(c2)
33          self.A = gpflow.Parameter(np.atleast_2d(A))
34          self.B = gpflow.Parameter(np.atleast_2d(B))
35
36      @inherit_check_shapes
37      def __call__(self, X: TensorType) -> tf.Tensor:
38          X0 = X[:, 0]
39          X0 = tf.reshape(X0, (-1, 1))
40          X1 = X[:, 1]
41          X1 = tf.reshape(X1, (-1, 1))
42
43          partL_in = self.c1 + tf.tensordot(X0**self.gamma, self.A, [[-1],
44              ↪ [0]])
45          partTemp = self.c2 + tf.tensordot(X0*X1, self.B, [[-1], [0]]) +
46              ↪ tf.tensordot(X0**2*X1, -self.B, [[-1], [0]])
47
48          return self.alpha * partL_in + (1-self.alpha) * partTemp

```

Code K.8: Code of the ViewpixxAlpha mean function that allows to set both dependencies and uses a weight factor α in the GPflow Framework [19].

K.4 Function to enable a likelihood with a linear increasing Gaussian noise

```

1 | class LinearNoise(gpflow.functions.MeanFunction,
2 |   ↳ gpflow.functions.Function):
3 |   """
4 |   y_i = c + A * x0_i
5 |   """
6 |
7 |   @check_shapes(
8 |     "c: [broadcast Q]",
9 |     "A: [broadcast D, broadcast Q]",
10|    )
11|   def __init__(self, c: TensorType = None, A: TensorType = None) ->
12|     None:
13|     """
14|     A is a matrix which maps each element of X to Y, b is an additive
15|     constant.
16|     """
17|     gpflow.functions.MeanFunction.__init__(<b>self</b>)
18|     A = np.ones((1, 1), dtype=default_float()) if A is None else A
19|     c = np.zeros(1, dtype=default_float()) if c is None else c
20|     self.A = gpflow.Parameter(np.atleast_2d(A))
21|     self.c = gpflow.Parameter(c)
22|
23|   @inherit_check_shapes
24|   def __call__(self, X: TensorType) -> tf.Tensor:
25|     X0 = X[:, 0]
26|     X0 = tf.reshape(X0, (-1, 1))
27|
28|     return tf.tensordot(X0, self.A, [[-1], [0]]) + self.c
29|
30| likelihood = gpflow.likelihoods.Gaussian(scale=LinearNoise())

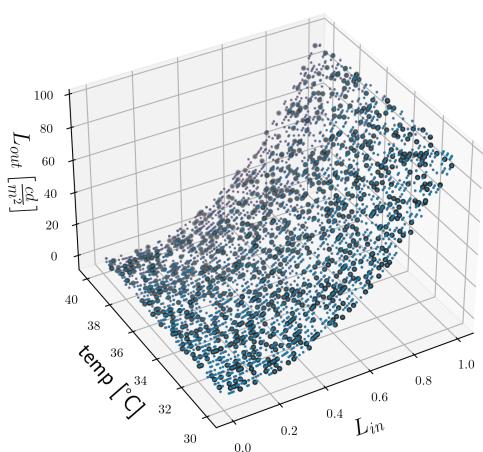
```

Code K.9: Code to define a likelihood that enables a (with *Lin*) linearly increasing Gaussian noise.

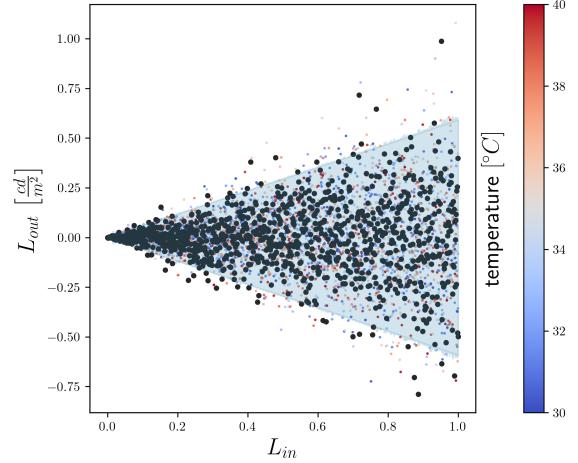
L Result of artificial data experiments

L.1 Mean function with weight factor α

The used code to create the fake data and the GP can be found in Appendix H.4.



(a) Training, prediction and “real” data points



(b) Residues of predicted points and “real” data points.
The residues have $\mu = -0.00122$ and $\sigma = 0.174$.

Figure L.6: Overview about the results from the GP using a mean function with weight factor α on artificial data representing both, the gamma dependency and the temperature dependence.

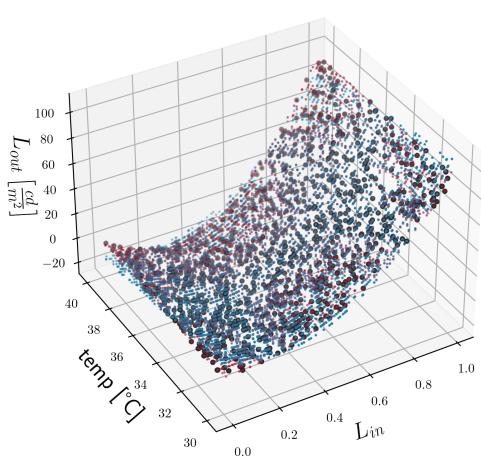
- “real” data point
 - point that the GP got to train with
 - point that the GP predicted
 - residues point that represents his temperature with a color map
 - 95% confident interval of the GP
- $\sigma = 0.312$

name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.gamma	Parameter	Identity	True	()	float64	2.28606	
GPR.mean_function.alpha	Parameter	Identity	True	()	float64	1.00034	
GPR.mean_function.c1	Parameter	Identity	True	(1,)	float64	[-23.82841]	
GPR.mean_function.c2	Parameter	Identity	True	(1,)	float64	[145.50176]	
GPR.mean_function.A	Parameter	Identity	True	(1, 1)	float64	[[69.2259]]	
GPR.mean_function.B	Parameter	Identity	True	(1, 1)	float64	[[3663.8638]]	
GPR.kernel.variance	Parameter	Softplus	True	()	float64	4535.30697	
GPR.kernel.lengthscales	Parameter	Softplus	True	()	float64	4.49107	
GPR.likelihood.scale.A	Parameter	Identity	True	(1, 1)	float64	[[0.29848]]	
GPR.likelihood.scale.c	Parameter	Identity	True	()	float64	0.00063	

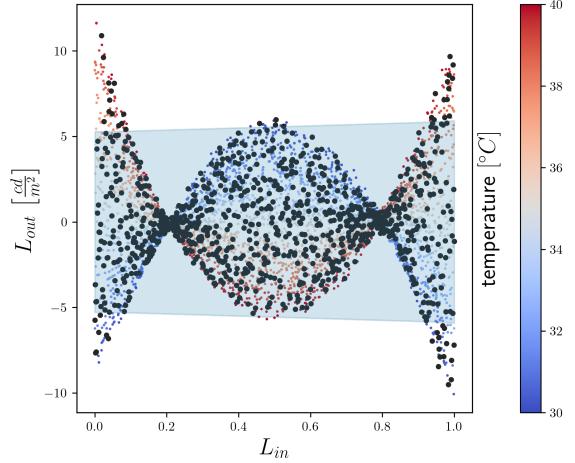
Table L.2: Optimization result of the hyperparameter by the GP using a mean function with weight factor α on artificial data representing both, the gamma dependency and the temperature dependence.

L.2 Only gamma mean function but both effects

The used code to create the fake data and the GP can be found in Appendix H.5.



(a) Training, prediction and “real” data points



(b) Residues of predicted points and “real” data points.
The residues have $\mu = 0.0567$ and $\sigma = 2.81$.

Figure L.7: Overview about the results from the GP using only a mean function for the gamma dependence on artificial data representing both, the gamma dependency and the temperature dependence.

- “real” data point
 - point that the GP got to train with
 - point that the GP predicted
 - residues point that represents his temperature with a color map
 - 95% confident interval of the GP
- $\sigma = 3.025$

name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.A	Parameter	Identity		True	(1, 1)	float64	[[101.6995]]
GPR.mean_function.b	Parameter	Identity		True	(1,)	float64	[-4.14673]
GPR.mean_function.gamma	Parameter	Identity		True	()	float64	2.68172
GPR.kernel.kernels[0].variance	Parameter	Softplus		True	(0)	float64	95.24376
GPR.kernel.kernels[0].lengthscales	Parameter	Softplus		True	(0)	float64	0.44245
GPR.kernel.kernels[1].variance	Parameter	Softplus		True	(0)	float64	578.3492
GPR.kernel.kernels[1].lengthscales	Parameter	Softplus		True	(0)	float64	17.7574
GPR.likelihood.scale.A	Parameter	Identity		True	(1, 1)	float64	[[0.31899]]
GPR.likelihood.scale.c	Parameter	Identity		True	(1,)	float64	[2.67118]

Table L.3: Optimization result of the hyperparameter by the GP using only a mean function for the gamma dependence on artificial data representing both, the gamma dependency and the temperature dependence.

L.3 Real training data

name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.gamma	Parameter	Identity		True	()	float64	1.84789
GPR.mean_function.alpha	Parameter	Identity		True	()	float64	1.03123
GPR.mean_function.c1	Parameter	Identity		True	()	float64	0.14234
GPR.mean_function.c2	Parameter	Identity		True	()	float64	1.00312
GPR.mean_function.A	Parameter	Identity		True	(1, 1)	float64	[[77.58737]]
GPR.mean_function.C	Parameter	Identity		True	(1, 1)	float64	[[0.07319]]
GPR.mean_function.D	Parameter	Identity		True	(1, 1)	float64	[[-82.9574]]
GPR.kernel.variance	Parameter	Softplus		True	()	float64	80.85113
GPR.kernel.lengthscales	Parameter	Softplus		True	()	float64	0.32011
GPR.likelihood.scale.A	Parameter	Identity		True	(1, 1)	float64	[[0.17863]]
GPR.likelihood.scale.c	Parameter	Identity		True	()	float64	0.00066

Table L.4: Optimization result of the hyperparameter by the GP trained with data from real experiments.

L.4 Trainings data for the Linearization

name	class	transform	prior	trainable	shape	dtype	value
GPR.mean_function.gamma	Parameter	Identity		True	()	float64	50.86926
GPR.mean_function.alpha	Parameter	Identity		True	()	float64	0.99971
GPR.mean_function.c1	Parameter	Identity		True	()	float64	4.52828
GPR.mean_function.c2	Parameter	Identity		True	()	float64	1.06937
GPR.mean_function.A	Parameter	Identity		True	(1, 1)	float64	[[112.42374]]
GPR.mean_function.C	Parameter	Identity		True	(1, 1)	float64	[[133.93828]]
GPR.mean_function.D	Parameter	Identity		True	(1, 1)	float64	[[11.70639]]
GPR.kernel.variance	Parameter	Softplus		True	()	float64	371.73787
GPR.kernel.lengthscales	Parameter	Softplus		True	()	float64	0.10012
GPR.likelihood.scale.A	Parameter	Identity		True	(1, 1)	float64	[[0.24974]]
GPR.likelihood.scale.c	Parameter	Identity		True	()	float64	0.00126

One aspect of this optimization result which stands out in particular is the value of $\gamma = 50.86926$. At first glance, such a high value might suggest that the fit is not satisfactory, since in real data, values of γ should only range between 0 and 3. Nonetheless, examining the graph (Figure L.8) of $L_{out} = L_{in}^{50.86926}$ reveals that the function is almost 0 when $L_{in} \leq 0.93$. This implies that the model under consideration may be mapping the diffraction of the luminance exclusively through the part of the mean function that is expected to map the temperature.⁸

Table L.5: Optimization result of the hyperparameter by the GP trained with real data to enable linearization of visual stimuli.

⁸ see Chapter 5.3

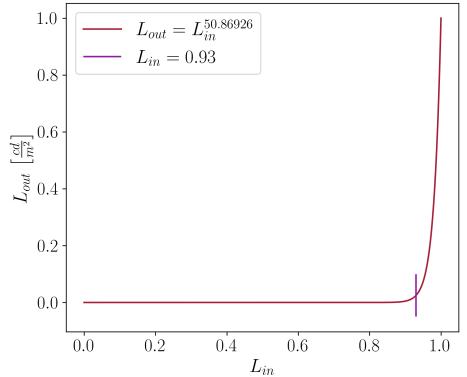


Figure L.8: Plot of $L_{out} = L_{in}^{50.86926}$

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Tübingen, 30.11.2023

Ort, Datum

J. Schägel

Unterschrift