MASTER

A practical device for measuring the luminance distribution

Kruisselbrink, T.W.

Award date:
2016

Link to publication
A Practical Device for Measuring the Luminance Distribution

Kruisselbrink, T.W., Aries, M.B.C., Rosemann, A.L.P.
Eindhoven University of Technology,
Unit Building Physics and Services, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
Corresponding author: T.W. Kruisselbrink@student.tue.nl

Abstract
Automated daylight systems, lighting simulations, and glare analyzes can be optimized using actual luminance distributions. Currently, available luminance distribution measurement devices are often not suitable for this kind of applications. In this study, a practical and autonomous luminance distribution measurement device was developed based on a Raspberry Pi camera system. The luminance distribution was determined using the similarity of the color matching function $\bar{y}$ of the CIE XYZ color space to the luminous efficiency curve of the human eye, representing the luminance, including some corrections. The CIE Y channel was obtained by translating the RGB information of High Dynamic Range images to the CIE XYZ color space. The High Dynamic Range technology was essential to accurately capture the data needed for the luminance distribution because it is, in contrast to ordinary images, able to capture the dynamic range that occurs in the real world. Finally, the measurement results were represented according to the subdivision developed by Tregenza and the entire process, to determine the luminance distribution, was automated in a Python code. Moreover, the results were uploaded to a server, whereby the results were accessible on external devices. Accuracy measurements showed that the luminance distribution measurement device had and acceptable average accuracy of ±15.1% for a measurement range of 5-18.000 cd/m².

Keywords
High Dynamic Range, Raspberry Pi, Autonomous, Measurement device, CIE XYZ, Calibration, Single-board computer

Introduction
Lighting simulation is an efficient way for designing comfortable and sustainable lighting conditions. However, the reliability of the simulation depends on the quality of the input model. An important aspect of the input is the sky luminance distribution. Previous studies have shown that representation of the sky luminance distribution continues to be a challenge (Inanici, 2010; Spasojevi & Mahdavi, 2005). The International Commission on Illumination (CIE) developed 15 generic sky models representing sky luminance distributions for conditions varying from overcast to cloudless skies (Inanici, 2010), based on long-term measurements using sky
scanners. The usability of the expensive sky scanners is limited: it takes a few minutes to measure the hemisphere and it does not allow measurements in high resolution (Kobav & Dumortier, 2007; Wüller & Gabele, 2007). Therefore, the generic CIE models do not represent the actual luminance distribution of the sky for any location and the models are not sensitive to transient luminance variations in different sections of the hemisphere (Spasojević & Mahdavi, 2007). Due to their generic character these models create uncertainties in the lighting simulations.

More and more buildings are applied with automated daylight systems. Relevant and actual sky luminance distributions can increase the performance of daylight systems because both the influence of the neighboring environment and the fast variations of the sky can be included in the input, resulting in an optimized user comfort and energy performance. Currently available sky luminance distribution methods, via sky scanners and cameras with proprietary software, are not suitable for automated daylight systems because of their extremely high price (cameras with proprietary software) and because they cannot handle fast variations of the sky (sky scanners) (Chiou & Huang, 2015).

Electrical lighting and daylight influence the satisfaction and performance of the occupants. Especially daylight can cause visual comfort problems by discomfort glare. Discomfort glare assessment, often indicated with the Daylight Glare Index (DGI), is not straightforward. The variables of discomfort glare: the luminance of the glare source, the size of the glare source, the position of the glare source, and the luminance of the background (Mead & Mosalam, 2016), are not easily measured with the currently available methods (Bellia et al., 2009). With a luminance distribution measurement device, all variables for a glare analysis are available and measured at once.

Previous research has shown that it is possible to measure the sky luminance distribution at once with cheap commercial digital cameras using the Red-Green-Blue (RGB) information captured using High Dynamic Range (HDR) photography (Inanici, 2006, 2010; Sarkar & Mistrick, 2006; Tohsing et al., 2013).

In this research, a method for fast capturing of sky luminance distributions based on a commercially available camera was developed. These real-time sky luminance distributions will offer possibilities to optimize lighting simulations and automated daylight systems, and it provides all needed information for a glare analysis.

The aim of this research was to develop a practical and autonomous camera based luminance measurement device using an inexpensive single-board computer equipped with camera and fisheye lens. In contrast to other measurement devices for the luminance distribution, this method should be cheap, quick, practical and completely automated. An accuracy in the range of ±5% to ±20% was pursued, because it was, in the first case, a practical measurement device. Such a practical and autonomous device could be placed at a certain location in or around
buildings and provide input for daylight simulations, automated daylight system or glare analyzes.

**Method**

As a single-board computer, the Raspberry Pi 2 (model B) was used. The camera functionality was accounted for by the specially developed Raspberry Pi Camera Board version 1.3 with CMOS sensor, which is comparable to the cameras in smartphones. A fisheye lens was used to provide a hemispherical image on the camera sensor. The Walimex Super Fisheye Lens for iPhone was chosen as the most suitable lens for this study. In combination with the camera board, this lens provided an equisolid angle projection with a field of view of 84% of the sky hemisphere. The code, in order to automate the measurement device, was composed in Python 3, one of the programming languages supported by the Raspberry Pi.

**Sky Modeling**

Fisheye lenses have an extremely short focal length and the projection lines that do not pass through the center of the image are strongly bent, resulting in an angle of view up to 180° but with a lower resolution and large distortions at the lens’ periphery (Schneider et al., 2009). Tohsing et al. (2013) suggested a straightforward method to describe the projection image of a fisheye lens by relating the elevation angle to the image radius with a curve fitted equation, a so-called distortion measurement. This equation related every pixel to the elevation as well as to the azimuth angle. It was not desired to get the luminance for every individual pixel because it was considered an ineffective representation. Therefore, the subdivision developed by Tregenza, displayed in Figure 1, was used (Tregenza, 1987).

![Figure 1. Tregenza’s subdivision placed over an image taken with the Raspberry Pi camera system. Due to the fixed focus length only 84% of the hemispherical view was captured, therefore, not all Tregenza samples were captured.](image-url)
Tregenza’s subdivision provides the luminance in a limited amount of samples (145) but with enough resolution to prevent major information losses. A script was developed that determined the average luminance for all pixels covered by one Tregenza sample. Before this was feasible the resolution was determined in which the images were taken. The image dimensions were limited to an aspect ratio of 4:3 since the focus length was not customizable, resulting in an 84% field of view (Figure 1). The applied resolution was based on the optimum between the file size and the accuracy of the Tregenza samples, which was determined by the percentage of sample surface that was covered by whole pixels.

**Input Settings**

In order to determine the luminance based on a photograph, the High Dynamic Range (HDR) imaging technology was required. Standard 8-bits images, with a dynamic range of 1,6 orders of magnitude (Reinhard et al., 2006), were not able to capture the entire dynamic range of the sky hemisphere which can have a dynamic range of 8 order of magnitudes, from $10^3$ to $10^5$ cd/m² (Moeck & Anaokar, 2013). In an HDR image, all pixel values are related to real world luminances; it is scene-related. The most common method to achieve a high dynamic range is the sequential exposure change technique (Cai & Chung, 2011). With this technique, simple digital cameras are used to take low dynamic range (LDR) photographs with sequential exposure settings to cover the desired dynamic range. In order to keep the optical properties constant it is recommended to only change the shutter speed (Cai & Chung, 2011). A measurement setup was designed in order to determine which set of exposures efficiently covered the dynamic range of the sky (Figure 2). A diffuse reflecting target was illuminated with a lamp in a completely dark and black room. The lamp (Arnold and Richter, 220V, 650W) was dimmed (100-260V with Philips Variable Transformer) and placed at multiple positions in order to achieve multiple luminances at the target. The luminance of the target was measured with a Hagner Universal Photometer S2 and simultaneously photographed by the Raspberry Pi with shutter speeds in the range from 17.000¹ to 2 seconds (f/2,9, ISO-100). Based on the under/over saturation empirical equations were determined that described which luminance range the different shutter speeds were able to capture. Based on these equations an exposure sequence of nine images was developed because the HDR image quality moderates at a series of nine exposure values (Cai & Chung, 2011).

The determined exposure sequence was formed into an HDR image by the command-line HDR builder for the Raspberry Pi (HDRgen), originally developed by Greg Ward (n.d), this builder was able to approximate the specific camera response curve using radiometric self-calibration (Inanici, 2006; Mead & Mosalam, 2016; Mitsunaga & Nayar, 1999). The camera response curve was approximated once according to the method described by Reinhard et al. (2006), it was determined for three scenes and averaged into on final response curve, and subsequently, reused for all luminance measurements.
Luminance Calculation

Based on the floating point RGB values of the HDR image the luminance was determined. Floating points are an approximation of real numbers that supports a trade-off between range and accuracy. In order to determine the luminance, the RGB color space was converted to the XYZ color space. An important property of the CIE XYZ color space is that color matching function \( \hat{y}(\lambda) \) is equal to \( V(\lambda) \), the luminous efficiency curve of the human eye for photopic vision, meaning that the Y channel indicates intensity of incoming light weighted by the luminous efficiency curve of the human eye (Wüller & Gabele, 2007), or in other words the luminance. Therefore, the luminance of a pixel was determined by calculating the CIE Y tristimulus value based on the RGB floating point values. The protocol as described by Inanici (2006) was used for translating RGB values to the Y tristimulus value. In this protocol, RGB was translated to XYZ by applying a conversion matrix depending on the primaries and the white point. The primaries are stored in the EXIF data, while the white point, depending on the correlated color temperature, can be extracted from tables (Reinhard et al., 2006) or calculated according to three equations (3.17-3.19) as described by Schanda (2007).

All variables of the conversion matrix except the correlated color temperature were constant. The exact correlated color temperature for each condition was not determined since it is an extensive process. Most studies developing a luminance distribution measurement device assumed a constant correlated color temperature (Inanici, 2006; Roy et al., 1998; Tohsing et al.,
2013; Wüller & Gabele, 2007), mostly illuminant D65, to determine the white point, resulting in errors over 15% in the conversion matrices for correlated color temperatures far from 6.504 K, the correlated color temperature of illuminant D65. Alternatively, the application of three constant correlated color temperatures, each representing a part of the Planckian locus, resulting in three different conversion matrices was used, which limited the error significantly (<6%).

In order to determine the CIE Y tristimulus accurately for all pixels, the vignetting effect was accounted for. The vignetting effect of a lens refers to light fall off at the periphery of the lens (Cai, 2012). Especially fisheye lenses exhibit noticeable light fall off, visible by the gradual darkening towards the corners of the images. In literature, it is noted that some fisheye lenses with some settings exhibit 73% light fall off at the periphery of the lens (Cauwerts et al., 2013). The vignetting effect is a non-linear radial effect along the radius of the lens and is often approximated by a polynomial function. The vignetting effect has a radial symmetric character, whereby the polynomial function can be used to determine the vignetting effect for all pixels of an image (Cauwerts et al., 2013; Inanici & Viswanathan, 2010; Moore et al., 2000).

The vignetting effect was determined in an Ulbricht’s sphere (Ø 700 mm), which was developed in such manner that it has, in theory, a uniform luminance over its surface (±1%) (Moore et al., 2000). For every tenth pixel along the image diameter, the vignetting effect was determined by dividing the luminance, determined with the Raspberry Pi, with the maximum luminance, which was a luminance close to the zenith. This process was repeated multiple times in order to achieve accurate results, limiting measurement errors. Based on the reciprocal of the vignetting effect, the vignetting correction was approximated by an empirical equation along the image radius. With this resulting equation, the vignetting effect for each individual pixel was corrected with a post-process correction filter.

A photometric calibration was the last step in order to accurately extract the luminance from the HDR image. This linear calibration factor related the CIE Y tristimulus to the real photometric quantity luminance, in candela per square meter (cd/m²), and brought the luminance to the correct order of magnitude. The calibration factor was determined for the gray (ρ=0.18) and white (ρ=0.90) samples of the Kodak Gray Cards and was determined under various conditions that were equal to the conditions the measurement device should cover. The samples were placed in front of the camera and were measured with the Hagner Universal Photometer S2 while the CIE Y tristimulus was calculated with the Raspberry Pi. The calibration factor described the ratio between the measured luminance and the calculated CIE Y tristimulus. This calibration process was repeated multiple times in order to avoid a calibration factor that was based on a coincidence. The final calibration factor was the average of all measurements.
Accuracy
The accuracy was measured with an agreement to the method described by Inanici (2006). Two gray Kodak cards were used with the addition of an uncalibrated gray scale and an uncalibrated color scale. Except one gray card all targets were placed in the center of the image, the remaining target was placed close to the periphery of the image to address the potential gradient in accuracy along the radius due to the vignetting effect (Figure 3). The luminance was measured with the Hagner Universal Photometer S2 while the luminance was also determined with the Raspberry Pi. Based on a correlated color temperature measurement, with the Konica Minolta illuminance spectrometer CL-500A, the most suitable constant correlated color temperature was determined and used. The accuracy was indicated by relating the physical measurement to the calculation results of the Raspberry Pi. This process was repeated for multiple scenes under multiple conditions the device should cover.

![Figure 3](image.jpg)

Figure 3. Setup for the accuracy measurements. Gray and colored targets were placed in the center of the image and one gray target was placed at the border. The accuracy was determined by comparing luminance measurements with the calculated luminance, this was repeated for multiple conditions.

Results
In this part, the results of the previously explained test are shown. First, all the settings and equations needed to perform a luminance distribution measurement are displayed, and they are followed by the results of the accuracy measurement using these settings and equations.

Sky Modeling
The optimization between the file size and the accuracy of the Tregenza samples led to an optimal horizontal resolution of 901 pixels and because of the fixed aspect ratio to an optimal vertical resolution of 676 pixels. With this resolution, 95% of a Tregenza sample was represented by whole pixels.

With the distortion measurement, as described by Tohsing et al. (2013), an equation was formed that was able to relate elevation angles to the image radius for the applied fisheye lens for images with a resolution of 901 x 676 pixels. With a coefficient of determination ($R^2$) of 0,9989, this equation was able to accurately determine which pixel represents what part of the photographed scene. The camera projection was described by:

$$ r_i = 624.3 \cdot \sin\left(\frac{\varepsilon_i}{2}\right) $$
With \( r_i \) as the image radius of the pixel and \( \varepsilon_i \) as the polar angle, the reciprocal of the elevation angle. This relation described the projection for this specific lens and resolution. However, it turned out that there was no significant difference between the projection equation of two lenses of the same type. Two similar lenses displayed a maximum absolute difference of 0.13% and a maximum relative difference of 24%. Using the projection equation it was determined which pixels represented all 145 Tregenza samples. The resulting luminance for each Tregenza sample was the average luminance of all corresponding pixels, a python script automated this process with an inaccuracy relative to the camera projection of 0.1%.

Due to the fixed aspect ratio, the camera captured only 84% of the sky hemisphere (Figure 1). Therefore, some of the Tregenza samples were not represented completely, the luminance of these samples was calculated using the only available pixels. Additionally, the samples that were not represented at all were estimated using a linear function that describes the trend of the previous six samples.

**Input Settings**

Based on the relation between the shutter speed and the luminance, as displayed in Figure 4, was determined to have Exposure Values (EV) in the range of 1 to 19.4 EV (maximum shutter speed Raspberry Pi) in steps of 1.8 EV. The exact exposure values slightly differ due to the inaccuracy of the camera device as displayed in Table 1. This sequence guaranteed that, except for the limits, each possible luminance was captured by at least two exposures.

---

**Figure 4.** The relation between luminance and shutter speed. In a measurement, the shutter speed of the Raspberry Pi camera system was related to the luminance based on the saturation of the images (o). This relation was approximated with curve fitted equations for the minimum luminance (blue), median luminance (green) and maximum luminance (red).
Table 1. Exposure sequence in order to make HDR image, including nine exposures values that were conducted by the Raspberry Pi camera system in order to make an accurate High Dynamic Range image for each possible condition. (EV = exposure value)

<table>
<thead>
<tr>
<th>Exposure</th>
<th>Shutter Speed [μs]</th>
<th>EV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>250.000</td>
<td>5.07</td>
</tr>
<tr>
<td>2</td>
<td>76.923</td>
<td>6.77</td>
</tr>
<tr>
<td>3</td>
<td>21.739</td>
<td>8.60</td>
</tr>
<tr>
<td>4</td>
<td>6.211</td>
<td>10.40</td>
</tr>
<tr>
<td>5</td>
<td>1.779</td>
<td>12.21</td>
</tr>
<tr>
<td>6</td>
<td>507</td>
<td>14.02</td>
</tr>
<tr>
<td>7</td>
<td>130</td>
<td>15.98</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>17.83</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>19.42</td>
</tr>
</tbody>
</table>

Tests with this exposure sequence showed that a number of exposures were always over- or under-saturated. For high luminances exposures 1 and 2 turned out to be always completely over-saturated, while for low luminances exposures 8 and 9 were always completely under-saturated. Therefore, the exposure sequence was optimized by leaving out the first or last two exposures depending on the conditions, whereby the quality of the HDR images increased and the influence of transient processes was limited. The most applicable sequence was determined by conducting a base of the exposure sequence (exposure 3-7) and subsequently assessing the 7th exposure on the level of saturation. When an area of exposure 7 was (almost) saturated exposure 8 and 9 were conducted instead of exposure 1 and 2 (Figure 5).

This response curve is specific for this camera. However, measurements with another Raspberry Pi camera board showed that the differences between response curves of two similar camera boards were limited to a maximum absolute difference of 2% and a maximum relative difference, for very low exposures, of 60%.

The HDR images were captured in the OpenEXR (.exr) format using RGB encoding with a depth of 96 bits. This format had sufficient dynamic range (76 orders), files were smaller than the other most used formats (HDR and TIFF), it had a relative step size of 0.1% and it was easy to read with Python using the OpenCV library (Version 2) (Holzer, 2006; Reinhard et al., 2006).
Figure 5. Forming a High Dynamic Range image. Two aspects were needed to form an HDR image: An image sequence and a camera response curve. The image sequence consisted of two parts: The base, this was always captured; and depending on the light intensity images 1 and 2 or images 8 and 9.

Luminance Calculation

Figure 6 shows that for high correlated color temperatures the error in luminance caused by a constant conversion matrix for illuminant D65 can increase to 17.9%. This error was limited by adding a second constant correlated color temperature reference point, that was able to calculate the luminance for high correlated color temperatures more accurately. It turned out that a constant correlated color temperature of 14,000 K was able to limit the maximum error to 7.6%. This error was minimized to 5.4%, by adding a third constant correlated color temperature of 3,000 K. With the addition of two extra constant correlated color temperatures, the maximum error was decreased from approximately 18% to approximately 5%. Basically, the constant of 3,000 K was suitable for luminance calculations indoors (warm white), illuminant D65 for overcast skies (daylight white) and the constant of 14,000 K for blue skies (cool white). The switching point between 3,000 K and D65 was at a correlated color temperature of 6,000 K and the switching point between D65 and 14,000 K was at a correlated color temperature of 8,600 K. For now, the most suitable constant correlated color temperature was selected by the user.
Figure 6. Deviation from luminance caused by constant correlated color temperatures. The conversion matrix to calculate the XYZ color space is dependent on the correlated color temperature, the figure illustrates the deviation that occurs when a constant correlated color temperature is used. \( T_{cp} = 3.000 \text{K} \) (green), \( T_{cp} = 65 \text{ (red)}, T_{cp} = 14.000 \text{K (blue)} \) and the three constant correlated color temperatures combined (black) with switching points 6.014 K and 8.571 K.

The primaries, obtained from the HDR files’ EXIF data, and the white points, calculated according to Schanda (2007), needed to form the color space conversion matrices are displayed in Table 2. Based on the protocol described by Inanici (2006) the conversion matrices were determined (Table 2). The luminance was calculated by extracting the CIE Y channel, leading to a simple equation (L), with calibration factor k and primaries R, G, and B.

Table 2. Variables of conversion matrices to translate RGB to XYZ for constant correlated color temperatures 3.000 K, D65 and 14.000 K. In contrast to the primaries the white points were dependent on the correlated color temperature, resulting in three conversion matrices.
Because the vignetting correction was measured along the entire diameter of the image all data was translated to the first quadrant to be able to extract a function along the image radius. In contrast to previous research, the vignetting filter was not described by a polynomial function. Curve fitting to an exponential function showed a much better match. Robust fitting to a second degree exponential function resulted in the function as described in Figure 7 with an $R^2 = 0.9093$. In order to extract an applicable function outliers were neglected. The outliers at a distance of 240 pixels were caused by an irregularity of the sphere. Some outliers were exhibited at a distance of 450 pixels, the very last pixel of the image, due to darkening caused by the image border. Since the vignetting correction was not perfectly fitted to the measurement data, the approximated function described the measurement data with a deviation of approximately ±10%.

$L_{3000K} = k \cdot (0.2319 \cdot R + 0.7073 \cdot G + 0.0608 \cdot B)$

$L_{D65} = k \cdot (0.2125 \cdot R + 0.7125 \cdot G + 0.0721 \cdot B)$

$L_{14000K} = k \cdot (0.01585 \cdot R + 0.7230 \cdot G + 0.1185 \cdot B)$

Figure 7. Determination and effect of vignetting correction. The black dots represent the correction factors to account for the vignetting effect that was measured in the Ulbricht Sphere, resulting in an approximated curve fitted equation (red). When the images were corrected with this fitted equation the vignetting effect was minimalized (blue dots).

Figure 7 shows that the light intensity, without vignetting filter, at the periphery of the lens was 56% ($1/1.8$) of the intensity in the center. The approximated function accounted for this vignetting effect. The vignetting correction limited the maximum vignetting effect to 14% while the average vignetting was limited to 2.5%. The vignetting effect was not undone completely.
Nevertheless, the reduction of the vignetting effect increased the accuracy close to the periphery significantly.

Calibration measurements showed that the calibration factor differed when the exposure sequence varied. Between different exposure sequences of the same scene the absolute CIE Y tristimulus values differed, while the interrelationship between the CIE Y tristimulus of pixels was rather similar in both sequences. The relative differences should be completely similar, but apparently the presence of saturated exposures, having no useful information, had a small influence on the outcome. Therefore, it was chosen to have the two before mentioned exposure sequences with both their own calibration factor. The calibration factor for exposure sequence 1-7 was 2.338,58 while the calibration factor for exposure sequence 3-9 was 22.902,16.

**Accuracy**

Accuracy measurements took place indoors as well as outdoors. A selection of the results is shown in Figure 8 and 9. Similar accuracy measurements showed similar results for different ranges of luminance. The measurements showed an average error of 10,1% and an error of 8,0% and 12,5% for respectively the gray and colored targets for a range of 5 to 18.000 cd/m². The measurements also showed that the device did not work for very high luminances (e.g. sun or reflections of the sun) due to saturation of the shortest exposure, whereby errors occurred in the HDR assembly resulting in false results.

![Figure 8](image.png)

*Figure 8.* The measured accuracy for indoor conditions with a correlated color temperature of 6.370K. The red dots represent the luminance, plotted on the right axis, measured with the Hagner Universal Photometer while the blue dots were determined with the Raspberry Pi. The black bars represent the relative difference, plotted on the left axis, between the blue and red dots.
It was expected that close to the periphery of the sensor the error would increase because the vignetting correction was not able to account for the vignetting effect completely. The results supported this hypothesis, the errors close to the border of the sensor were significantly higher; for gray targets, it displayed an inaccuracy of 27% compared to 8% in the center. It is expected that this error represented approximately the last 75 pixels along the radius because there the vignetting effect became significant. For the other pixels, the vignetting effect was more limited, it was therefore expected that error for these pixels was similar to the errors at the center of the image.

**Automation**

A Python code was developed that was able to carry out the entire process autonomously. The code is structured as displayed in Figure 10. The entire process is placed inside an infinite loop, so it will run infinitely until the user interrupts it. The code only asks for interaction regarding the constant correlated color temperature. The user selects the most suitable constant correlated color temperature, by an intuitive control panel, based on the user’s knowledge, the specifications of the lighting or a physical measurement (e.g. with Chromameter). This control panel consists of three buttons representing the three constant correlated color temperatures, and three LEDs indicating the selected correlated color temperature. The user is able to switch between the options endlessly, as long as the control panel is enabled (indicated by the red LED). Illuminant D65 is selected as default correlated color temperature. When the luminance...
distribution measurement is started some time is given to select the suitable correlated color temperature. Then, the luminance calculation is started with capturing the most suitable image sequence, when the luminance is too high the process is aborted and after five minutes it will retry. When the exposure sequence is captured successfully, the exposure sequence is formed to an HDR image. Based on the, by the user selected, correlated color temperature the CIE Y is subsequently extracted from the HDR image. Consequently, in order to achieve an accurate luminance, the calibration factor and the vignetting correction are applied. After this, the luminance of each individual pixel is known. This is not an effective representation, therefore, the results are represented more useful by averaging over Tregenza’s subdivision. Finally, the results, a list with the average luminance for each Tregenza sample and a tone mapped HDR image, are uploaded to a server, allowing access to the measurement results from an external computer. When the process is completed the code waits until 5 minutes are passed since the start of the measurement. During this waiting time the user is free to change the correlated color temperature.

Figure 10. Flowchart representing the automated luminance distribution measurement. The straight arrows represent the flow, the blocks the processes, the diamond decisions and the curved arrows represent user input that is acquired at the start of the process and used later in the measurement process.
Discussion

Due to the fixed focus distance the camera was bound to an aspect ratio of 4:3, resulting in some information loss at the top and bottom of the image. However, the information loss was limited because for the part that was not captured the Tregenza samples were approximated based on the neighboring samples.

The established exposure sequence had two variations in order to minimize the number of saturated exposures. It was developed in such a way that the entire range of possible luminances was captured. However, there were plenty of situations where the established sequence was not optimal, because many of the photographed situations had a limited or different luminance range, resulting in more saturated images. The accuracy of the HDR image, and hence the accuracy of the luminance measurement device, would be more accurate when the exposure sequence is based on the situation. Moreover, it turned out that the shortest exposure possible was not able to capture the luminance of the sun and its reflections. The luminance of the sun was multiple orders of magnitude larger than the maximum luminance that could be captured with this exposure sequence. The sequence was formed into an HDR image with the HDR-builder developed by Ward, which provides a range of settings. The settings were assumed constant for all situations, whereby for some conditions the settings were not optimal.

The camera response curve was approximated with the HDR-builder, it was not the exact camera response curve. For low values, the relative difference between the camera response curves of two similar cameras was 60%. Nevertheless, the camera response curves were considered equal since the low absolute difference while the large relative difference occurred only at very low exposures. Only in situations where all information lies in the low exposures, a very dark scene, some significant differences might occur. However, it is not the intended use of this device to measure in low-luminance conditions. Although they displayed a relative difference of 24%, lenses of the same type were also considered equal. These relatively large differences occurred in the center of the image, while the relative difference close to the periphery, where distortion occurs, was negligible. Therefore, the developed code can be used to measure the luminance distribution, with an acceptable accuracy, using another Raspberry Pi, as long as the same type of equipment is used.

The luminance calculation was based on the similarity between the CIE Y channel and the luminous efficiency curve of the human eye for photopic vision. Therefore, the accuracy will decrease when the scene is not well lit because the luminous efficiency curve for mesopic and scotopic light differ from the luminous efficiency curve of photopic vision. Photopic vision changes to mesopic vision at a luminance of approximately 5 cd/m² (Hood & Finkelstein, 1986), therefore the measurement device was not suitable for luminances below 5 cd/m². Moreover, errors occur because three constant correlated color temperatures were used instead of the actual correlated color temperature. Applying three correlated color temperature reference points was a large improvement already limiting the deviation to approximately 5%. However,
since there is user involvement required, it could lead to a slightly higher inaccuracy, when the user does not provide input, constant correlated color temperature D65 was assumed. The highest accuracy will be achieved by determining the actual correlated color temperature and using this to determine the actual conversion matrix for the respective scene.

It seems that in some other studies the vignetting filter was based on one single measurement, resulting in extremely good fits (Cauwerts et al., 2013; Inanici, 2006). This research showed that the vignetting correction should be based on multiple measurements because the vignetting effect displayed differences up to 20% under ‘constant’ conditions. Therefore, the vignetting effect close to the periphery was only limited to 14% by fitting to data achieved by multiple measurements. As a result, the performance of the measurement device is worse for areas with a significant vignetting effect compared to the center of the image. Apparently, the conditions were not entirely constant. This might be due to the relatively low quality of the lens and camera, resulting in more noise. Nevertheless, it is still reasonable to perform multiple measurements in all cases of vignetting effect measurements in order to achieve an optimal vignetting correction.

The calibration factor was determined for white and gray targets, the resulting calibration factors showed a relatively large deviation between white and gray. White and gray are extremes, in most cases the brightness of the measured colors will be somewhere between white and gray. Therefore, the average of white and gray was suitable as the general calibration factor. Every exposure sequence had its own calibration factor. Two different exposure sequences of the same scene showed that the image data was brought into a different domain depending on the exposure sequence while the interrelationship between pixels was independent of the exposure sequence.

The accuracy was determined with the Hagner Universal Photometer S2, which has an accuracy of ±5% itself. This means that the actual accuracy of the luminance distribution measurement device could be 5% higher or lower. So taking in account the inaccuracy of the Hagner Universal Photometer the accuracy of the developed device was: ±15,1%, ±13,0% and ±17,5% for respectively all targets, gray targets, and colored targets.

**Conclusions and Recommendations**

**Conclusions**
The luminance distribution was determined based on the similarity of the CIE Y color matching function and the luminous efficiency curve of the human eye, including some corrections. The CIE Y channel was achieved by translating the RGB information of the High Dynamic Range (HDR) image to the CIE XYZ color space. This was done using three constant conversion matrices each representing a part of the Planckian locus. The High Dynamic Range technology was essential to accurately capture the luminance distribution because it is, in contrast to ordinary images, able
to capture the entire dynamic range that occurs in the real world. Finally, the luminance distribution was represented according to Tregenza’s subdivision.

This process, in order to determine the luminance distribution, was conducted using a Raspberry Pi (with camera board) as a single-board computer which was able to perform all calculations automatically. However, determining the most suitable constant correlated color temperature was not done automatically, this was selected by the user. The device can be considered autonomous because it also functioned (and can still function) without the user’s input. However, this was at the expense of the accuracy because the constant correlated color temperature D65 is assumed while it might not be the most suitable constant correlated color temperature. The best performance was acquired when the user selected the suitable constant correlated color temperature at the start of the measurement and changing this when the conditions had changed. Nevertheless, the device was considered practical. No specific lighting knowledge is needed in order to use the device, almost no interaction with the device is needed. The results are automatically digitalized and uploaded to a server, and the device is fairly accurate with an average accuracy of ±15.1% and an accuracy of ±13.0% and ±17.5% for respectively gray and colored targets. All of this as achieved with an investment of below €100,-.

The device has currently one major limitation. It gives only reliable results within a certain luminance range. Measurements showed that for very high luminances (>18.000 cd/m²) the results were false. Therefore, The measurement range was set to 5-18.000 cd/m² because in this photopic vision range successful measurements were conducted.

**Recommendations for further research**

Despite the fact that the objectives are largely met, some additional improvements can be achieved.

First of all the quality of the HDR image can be further improved. In this study, only two different exposure sequences were used being the optimal sequence for just a limited set of conditions. The quality of the HDR image will be the highest when the exposure sequence is developed specially for each condition, preventing saturated pictures. Meaning that all nine exposures are evenly distributed within the occurring luminance range. An illuminance sensor, for instance, can be very useful for this. The illuminance sensor will indicate the intensity of the incoming light on which, subsequently, the exposure sequence can be based.

Another limitation was that with the current device no luminances over 18.000 cd/m² could be reliably captured. This is because the intensity of the high luminances is, in some cases, so high that the shortest exposure saturated. Because a shorter shutter speed is not possible the intensity of light that falls on the sensor should be decreased. This can be done with a neutral density filter (Stumpfel et al., 2006), a neutral density filter is a gray filter, that can be placed in front of the lens, that decreases the intensity of the incoming light evenly over all wavelengths. This way the current dynamic range can be shifted towards longer exposures and the shorter exposures can be used to capture higher luminances. Disadvantages are that for darker
conditions the exposure time becomes significantly higher, whereby the influence of transient processes increases.

Due to the fixed focus length of the camera board not the entire hemispherical view was captured, in this case, the missing part was approximated. It will be more accurate if the entire hemispherical view is captured. There are two methods available to achieve this. A camera with adjustable focus length can be used. However, in the case of the Raspberry Pi camera, this will mean that the fisheye lens cannot be placed in front of the camera as is done here. Another option could be using two cameras which are rotated 90° compared to each other. This way the entire hemispherical view is captured and fisheye lenses can still be applied. When the entire hemispherical view is captured also the illuminance can be determined.

Also, the accuracy of the luminance calculation can be improved. Three constant correlated color temperatures were applied resulting in an additional luminance error of approximately 5%. This error can be avoided by using the actual correlated color temperature. This can be determined by using three sensors which are responsive to red, green, and blue input respectively, achieved by placing color filters in front of the sensors. With the separate intensities of red, green and blue one is able to determine the exact and actual correlated color temperature. This measure will increase the accuracy as well as the usability because no user input is required anymore.

Finally, some improvements can be made regarding the usability of the measurement device. In order to get insight in the calculation process, or in order to start the process, a screen and keyboard are needed. The usability will improve when no additional equipment is needed, it is, therefore, useful to extend the control panel with a small screen on which the process is displayed; and a button that is able to start and quit the process at any time. The next step in providing a useful representation could be a false color representation instead of Tregenza’s subdivision. A false color representation is able to provide an intuitive and quick understanding of the measurement results.

The luminance distribution measurement device can also be extended with other functions using the camera or the computational power of the Raspberry Pi. Examples could be a security camera, a presence detector, a smart thermostat, and so on.

References

A Practical Device for Measuring the Luminance Distribution


