DOI: 10.22032/dbt.58604

Automated and Repeatable Homogeneity Calculations on Simulated and Measured Luminance Fields

Anil Jean Pekgöz[1], Jannes Buthmann[1]*

[1] Team Licht & Sicht, EDAG Engineering GmbH, Birnbaumstücke 5, 38448 Wolfsburg, Deutschland

[*]Corresponding Author: jannes.buthmann@edag.com

Abstract

In the field of light system engineering, evaluating the performance of the developed device with various tests is as crucial as the development process. The homogeneity performance is one of the tested criteria which plays a significant role for customer satisfaction. To be able to validate the quality of a luminance distribution in a lighting system multiple methods has been developed. These methods which have different use-cases in terms of the evaluation are connected with one similarity - time consuming and with a need of a lot of effort which needs to be spend for the calculation. This issue is caused since the fact that a result with high accuracy is almost impossible to achieve with the conventional methodology of by-hand evaluation. Therefore, an exceptionally good assessment of uniformity for user satisfaction is hard to accomplish. In this project, we developed a program which can perform complex calculations in a semi-automated manner with the highest precision possible, limited by only the finite nature of provided images for the calculations.

Index Terms: Light System Engineering, Homogeneity Analysis, Automated Calculations, Diekmann-Gerloff Method

1 Introduction

In light engineering, many aspects such as physical integrity, production feasibility and light uniformity are tested throughout the development process of a light device. These tests are performed to ensure the user's safety and producibility.

The uniformity of emitted light distribution has a crucial role in the produced light shape and characteristics for interior-light systems are also playing a part in user satisfaction and well-being [1]. Therefore, there have been methods [1-3] and standardization to calculate the uniformity of light distribution. However, the applicability of these standardizations and methods are cumbersome since the non-standard shapes and capturing method. Consequently, uniformity calculations are done mostly by hand.

Furthermore, to save time in those by-hand-calculated results, the arrays are discretized into blocks via downsampling [4] to an array with fewer elements to shorten the calculation time needed, thus lowering the accuracy of the result.

With this in mind, we are proposing a program which can deploy various methods of calculation automatically via automatic field detection [5, 6] and morphological operations [7]. Moreover, since the calculations are based on hardware computation rather than by-hand calculations, the discretization can be reduced to the discrete limitation on a single pixel. Therefore, the calculations do not lack accuracy due to outside factors. Furthermore, the mathematical model implementation allows the user to integrate custom mathematical models which fulfills the user's/customer's requirements on uniformity.

2 Methods

Python has been used as the main programming language since its simple yet powerful nature in the field of image processing. Moreover, the Global Interpreter Lock (GIL) [8] which limits the usable threads has been bypassed for the most demanding process via the Low Level Virtual Machine (LLVM) [9] with parallel computing implementation. Therefore, execution time was almost the same as in other compiled languages.

A total of 7 standardized methods are used in the program 5 of them are standards from light system engineering on cars [1], 1 of them is a standard in display engineering [2,3] and the last one is a custom bundle of smaller methods from an OEM. The methods of light system engineering on cars are depicted in Table 1.

GK_X (Implemented as)	Equation	Description
$GK_1\ (M_1)$	$\log\left(\frac{L_{max}}{L_{min}}\right)$	The method GK_1 calculates the uniformity with the logarithmic human perception in mind [6].
$GK_2\ (M_2)$	$\frac{L_{max}-\bar{L}}{\bar{L}}$	The method GK_2 calculates the difference ratio between the maximum and the mean relative to the mean [1].
$GK_3\ (M_3)$	$\frac{L_{max}-L_{min}}{\bar{L}}$	The method GK_3 is called the *Relative Percentage Difference* (RPD) [10] and represents the ratio between the *range* and the *mean*.
$GK_7\ (M_4)$	$\frac{\sigma_L}{\bar{L}}$	The method GK_7 calculates the dispersion of values around the mean and is called the *Coefficient of Variation* [1, 11].
$GK_8\ (M_5)$	$\sqrt{\sum_{\alpha=x,y}\left(\frac{\Delta S_{\alpha,n}}{\Delta\alpha_n}\right)}$	The method GK_8 calculates the normalized distance between the geometric and photometric centers [1].

Table 1: the standardized methods used in developing light systems for cars. The method names are shown with the standardized name and the name as implemented in the program. The equation and their basic descriptions are also provided.

The most interesting method mentioned in Table 1 is the GK_8 method since it uses more information to calculate the uniformity of the luminance distribution than the other methods.

Another method that is a standard in another field is the H_{mean} method [2,3] where a gradient-like approach was used. This method originates from benchmarking display's light uniformity. However, since some signal functions such as position light, daytime running light or interior lighting features are designed to signal or display a pattern rather than to illuminate areas other than the light source, this method can be used to evaluate those areas. Nevertheless, for the directly illuminating light devices, the light projection taken by a 3D Goniometer can be also used to evaluate the produced light. The equation is depicted at Eq. 1.

$$H_{mean} = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{i^*>i}^{N}\sum_{j^*>j}^{M}\left|\frac{L_{ij}-L_{i^*j^*}}{\frac{\left(r_{ij}-r_{i^*j^*}\right)}{\sqrt{M^2+N^2}}}\right|\right)}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{i^*>i}^{N}\sum_{j^*>j}^{M}\frac{L_{mean}}{\left|\frac{\left(r_{ij}-r_{i^*j^*}\right)}{\sqrt{M^2+N^2}}\right|}\right)} \qquad \text{(Eq. 1)}$$

The main challenge with H_{mean} is the permutative behavior which causes the exponential increase of vectors that must be calculated. Therefore, the general approach is to discretize the image in large areas to decrease the points. Nonetheless, in our approach, we are able to use every pixel to calculate the H_{mean} with the highest accuracy. Table 2 provides the vectors needed to calculate the H_{mean} for different situations.

Image size (% Coverage)	500×500 (50%)	1000×1000 (50%)	1500×1500 (50%)	3000×3000 (50%)
# of Vectors	1.56×10^{10}	2.49×10^{11}	1.26×10^{12}	2.02×10^{13}

Table 2: Needed vectors for calculating the H_{mean} in different situations. The needed vectors are increasing exponentially since the permutative behavior between pixels in the calculations.

Another aspect of the program is the semi-automated area recognition for the calculation. This is provided by the different morphological operations [10] and local labeling algorithms. The local labeling algorithms are used to locate different areas. However, the false-positive fields induced by noise or by complex structure are dealt with a combination of morphological operators.

3 Results and Discussion

As the input, the EDAG's illuminated emblem shown in Figure 1 is used. The size of the luminance array is 2051 by 2449 with a precision of float-64 bit and the coverage in total is 48%. In the image, there are 4 separate areas to be calculated.

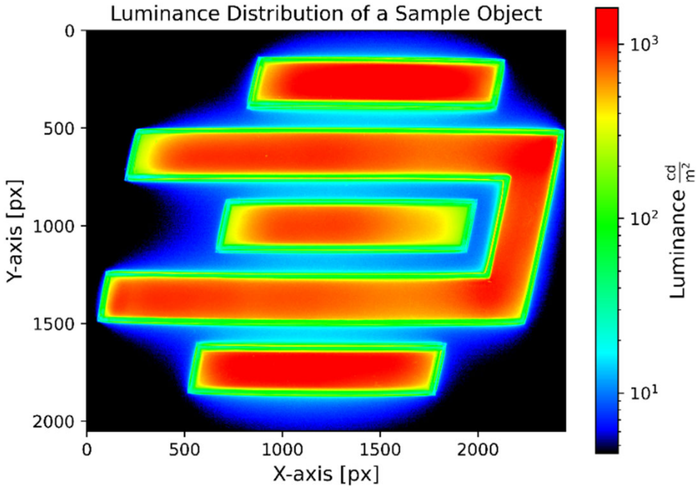

Figure 1: Input luminance image of the EDAG Emblem

The first step is to apply a global noise threshold predefined by the user. The noise-eliminated array is then used to apply the local labeling algorithm mentioned in the *Methods Section*. The resulting image with percentage filtration, where all areas which are smaller than a defined percentage of the total area are eliminated, is depicted in Figure 2.

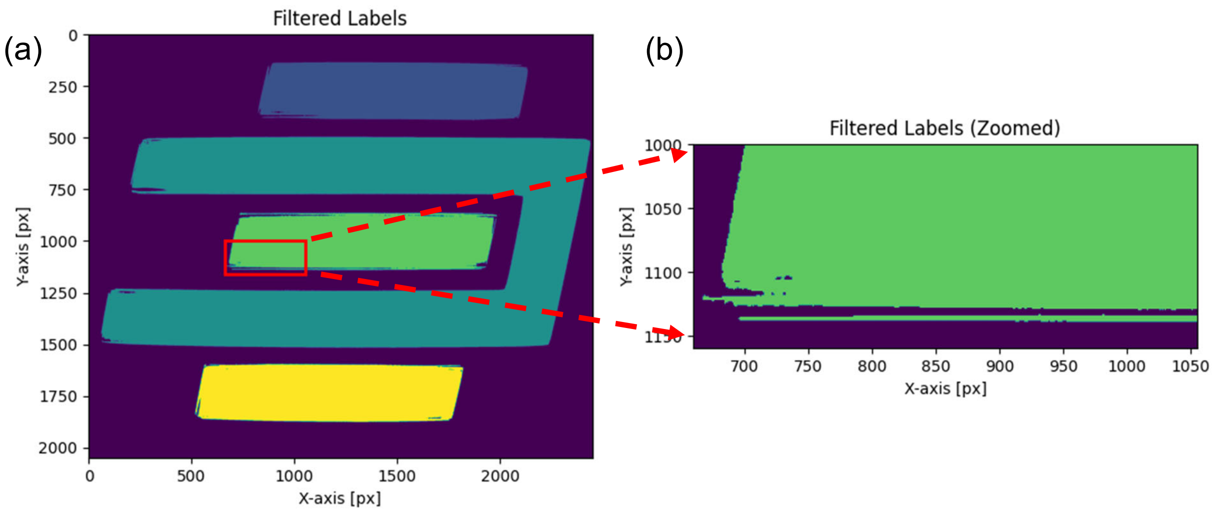

*Figure 2:**A**, The filtered labels. The array now consists of 4 labels which can be addressed with a number and be called whenever a specific test has to be done. **B**, A zoomed portion of the filtered image to show the rough edges which are not optimal for further calculations. The zoomed portion is marked as the red box in Fig.2A.*

The rough edges are fixed with an iterative Morphological Operations (MOs) which modifies the topology of the label till a relatively smoother edge has been archived.

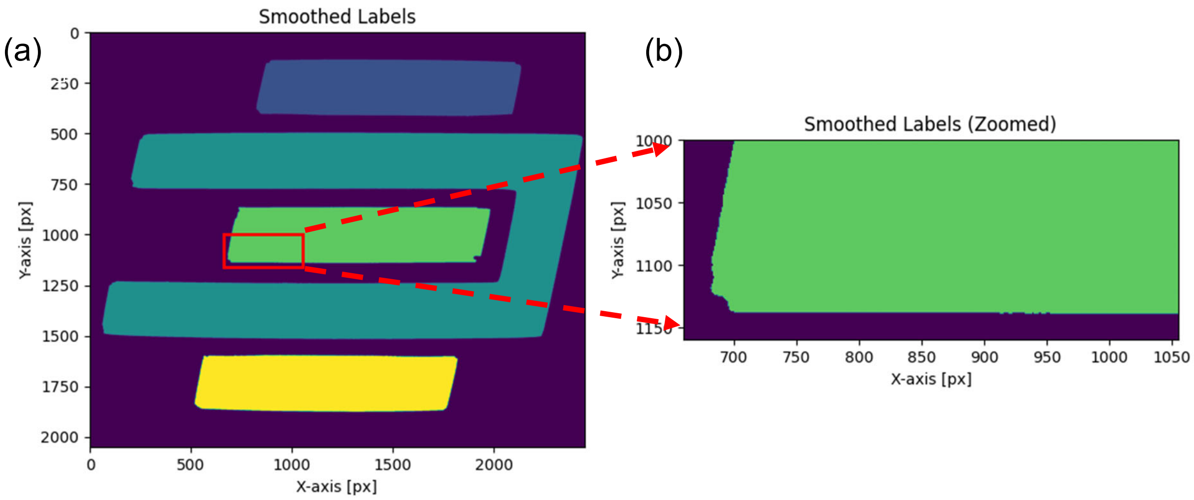

*Figure 3: **A**, The same filtered image with MOs applied. **B**, The same zoomed spot as the Fig.3B to with the MOs applied. The difference in smoothness is visible.*

The resulting labels array is now ready to be used as a map for the luminance array. Methods such as H_{Mean} and GK_x are applied sequentially for each label. The parallel acceleration is used for H_{Mean} since the other methods are mostly iteration-depended.

At the end, the uniformity results are presented as a table where the columns are representing the method and the rows representing the labels. The result for this input image is depicted in Table 3.

Labels (from larger to smaller)	M_1	M_2	M_3	M_4	M_5	H_{Mean}
#2	208.46	1.7	1.23	4952.88	0.02	75.65%
#1	222.81	1.4	5.46	575.91	0.01	68.22%
#3	164.41	1.27	0.73	1035.06	0.03	69.22%
#4	205.85	1.29	0.68	638.7	0.02	71.49%

Table 3: The results calculate for each label by different methods. The rows are sorted from the largest area to the smallest.

The interpretation for each method is quite complex since each method represents a different aspect analytically. Moreover, there are some false-positive results for the GK_x methods for some numerical cases. Therefore, a whole new detailed study on those themes should be conducted to find the best combination of the GK_x methods and what should be *correctly* understood by those values. However, there is a detailed reference table for the H_{Mean} values depicted below.

Uniformity	The visual interpretation of the illuminated surface
100% - 80%	Highly uniform. No brightness change on the surface can be detected by a human observer.
80% - 60%	Uniform. The change in brightness on the surface can be detected but it is not disturbing for the human observer.
60% - 40%	Non-uniform. The change in brightness on the surface is disturbing.
Lower than 40%	Very non-uniform.

Table 4: A reference table for the H_{Mean} value [2, Translated].

Referencing this table, it is possible to make a quick statement from the gotten results. Moreover, it is also possible to reference old revision to see if the values are gotten better by the new revision.

4 References

[1] Gocke, Tim: Objektivierte Homogenitätsbewertung des Erscheinungsbildes automobiler Signalleuchten, 01.01.2013, [online] doi:10.22032/dbt.39795.

[2] Gerloff, Thorsten/Dirk Friedrich/Karsten Diekmann: Die Leuchtdichtehomogenität von Organischen Leuchtdioden (OLEDs), in: AUSGEWÄHLTE THEMEN DER LICHTMESSTECHNIK, Deutsche Lichttechnische Gesellschaft e. V. (LiTG), no. 27, 2015, https://www.litg.de/media/19235.27_Lichtmesstechnik_Download, pp. 16-24

[3] Hupe, Christoph: Homogenitätsbewertung flächiger Lichtquellen und Leuchten, Lux Junior/TU Ilmenau, 2013, [online] https://www.tu-ilmenau.de/fileadmin/Bereiche/MB/lichttechnik/Literatur/2013/Hupe_Luxjunior_2013.pdf.

[4] Digital Signal Processing First, Global Edition: 2nd. ed., Pearson, 01.01.2015

[5] Chaki, Nabendu/Soharab Hossain Shaikh/Khalid Saeed: Exploring Image Binarization Techniques, in: Studies in computational intelligence, Springer Nature, 01.01.2014, [online] doi:10.1007/978-81-322-1907-1.

[6] Wu, Kesheng/Ekow Otoo/Arie Shoshani: Optimizing connected component labeling algorithms, in: Proceedings of SPIE, SPIE, 29.04.2005, [online] doi:10.1117/12.596105.

[7] Soille, Pierre: Morphological Image Analysis, in: Springer eBooks, 01.01.2004, [online] doi:10.1007/978-3-662-05088-0.

[8] Beazley, David: Understanding the Python GIL, PyCon Conference, 2010, [online] http://www.dabeaz.com/.

[9] The LLVM Compiler Infrastructure Project: n.d., [online] https://llvm.org/.

[10] Cole, Tim J/Douglas G. Altman: Statistics Notes: What is a percentage difference?, in: BMJ, BMJ, 16.08.2017, [online] doi:10.1136/bmj.j3663, p. j3663.

[11] Everitt, Brian/Anders Skrondal: The Cambridge Dictionary of Statistics, 4. ed., Cambridge University Press, 2010, [EBook] https://www.cambridge.org/9780521766999.