Lux junior 2023

16. Internationales Forum für den lichttechnischen Nachwuchs
Ilmenau, 23. – 25. Juni 2023
DOI (proceedings): 10.22032/dbt.55787

DOI: 10.22032/dbt.58602

Usage of Material Properties of 3D Objects for an Improved Illumination by High-Definition Matrix Headlights

Nathalie Müller[1], Mirko Waldner[1], Alisha Maroke[2] and Torsten Bertram[1]

[1] Institute of Control Theory and Systems Engineering (RST), TU Dortmund University, Germany

nathalie2.mueller@tu-dortmund.de

[2] 3D Mapping Solutions GmbH, Germany

Abstract

The contribution at hand presents a novel algorithm to improve visual perception for humans and machine vision algorithms by virtually adjusting the illumination of high-definition (HD) matrix headlights as a function of different material properties in the environment, such that each material can be illuminated with a different illumination intensity. Since the material properties affect the visual perception of objects, this approach allows the attention of human drivers and other human traffic participants to be focused on selected areas or significantly increase the detection quality for machine vision algorithms in the case of automated driving, while it is possible to save energy at the same time.

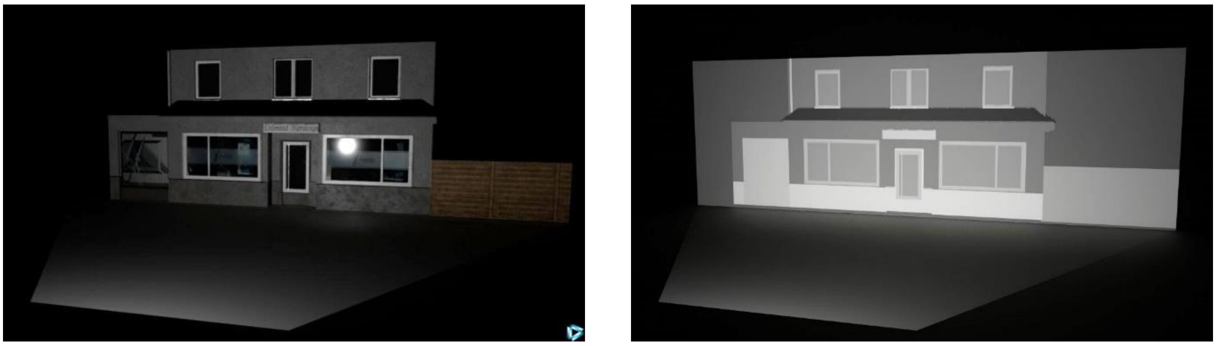
Index Terms: Matrix Headlights, Simulation, Visual Perception, Object Detection

1 Introduction

The visual perception of objects is mainly influenced by their surface properties such as roughness or color, on which the reflection of light depends. If these surface properties are known, the illumination can be optimally adjusted, e. g. to save energy or to improve automated object detection by avoiding overexposed images. This is important for safety-critical applications, e. g. object detection for automated vehicles, as the occurrence of traffic accidents and near accidents can be minimized if there are no incorrectly detected or undetected objects. The same applies to human recognition in manual driving.

The contribution at hand investigates the potential of HD matrix headlights to improve the illumination of the environment with knowledge of the material properties of the illuminated objects for both humans and machine vision algorithms, e.g., YOLOv8 [1], in the case of cameras of automated vehicles. HD matrix headlights consist of a matrix of light pixels that individually illuminate segments of the environment, allowing different materials of objects to be illuminated with different light intensities. This can be used to direct the driver's attention, improve visual perception, or achieve equally good perception while saving energy.

The examinations in this contribution are analyzed virtually in Unreal Engine (UE) 5.1 [2]. Thus, the Physically Based Rendering (PBR) [3] model is used for 3D objects for realistic visualization for evaluation. For the simulation of headlights, headlight models that use projective texture mapping [4] are considered, as this is an already established method for simulating headlights [5]. The basic idea and functionality of the algorithm are shown in Fig. 1, where based on the known materials of an example 3D house and a fence, the illumination can be adjusted individually per material, e.g. wood or glass, to account for different colors or reflectance properties of the individual material.



(a) Illumination of a 3D house in UE

(b) Pure illumination of (a) projected onto a wall

Fig. 1: Example illumination of a scene with one matrix headlight using the presented algorithm for different illumination of other materials. The different materials of the 3D house in (a) are considered by the algorithm, resulting in a texture for the headlight projected on a wall as shown in (b), with different pixel intensities for different materials. Therefore, the shape of the house is also recognizable solely by the adjusted illumination on a wall.

In the next section, the algorithm for setting different pixel intensities of the headlights for different known materials of objects existing in the environment is presented. This is followed by a section on evaluating the impact of the proposed approach on human perception and machine vision algorithms. The contribution concludes with a summary and an outlook.

2 Algorithm for Material Location Detection and Headlight Control

The algorithm to illuminate different materials with different illumination intensities is divided into six main steps, which are listed here shortly to give an overview and will be explained in more detail in the following paragraphs:

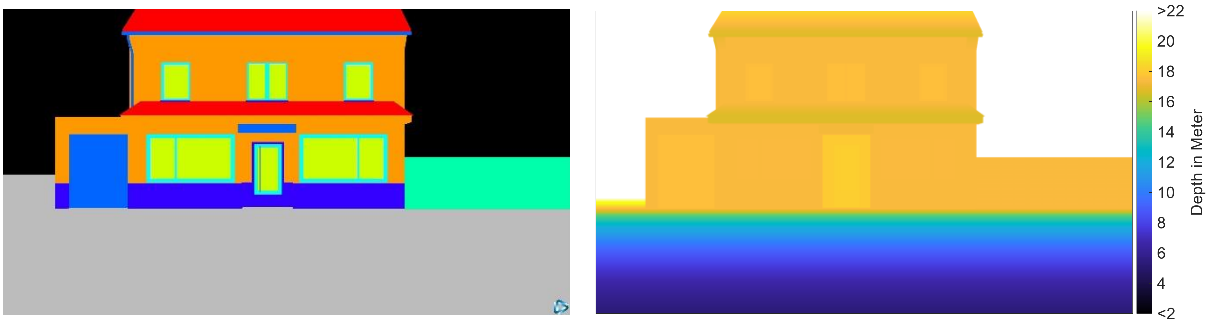
1) Setup Illumination Change: A color coding is set according to different known materials occurring in the scene. Depending on the material, a different normalized illumination intensity is set, ranging from 0 to 1 of its intensity maxima.
2) Getting Distance and Materials: A depth image and a material color-coded picture are captured from the scene.

3) Searching for Defined Materials: The color-coded picture is searched for the known materials from step 1). If a corresponding material is detected in a pixel of the captured image, the pixel is assigned to the material, otherwise, the default or fallback setting is used. A material image is created from this.
4) Calculating Object Coordinates in Headlight Coordinate System: The coordinates of the surface points in world space are calculated from the depth image and the camera position. Afterward, they are transformed into the headlight coordinate system.
5) Calculating Pixel Control Values: The material image from 3) and the corresponding surface points in the headlight coordinate system from 4) are fused to calculate the intensity values of the individual pixels of the HD matrix headlight.
6) Illuminating the World: The intensity values from 5) are set by the HD matrix headlights.

For illustration purposes, the description of the algorithm is supported by an example, which is also used later in the evaluation section. The example includes a scene with a 3D house with a fence at one side, representing a real house in the German city Lippstadt that has been precisely measured and modeled by 3D Mapping Solutions. Fig. 1 shows this example. A pair of parallel HD matrix headlight modules is used for illumination.

As a prerequisite for the presented approach and step 1) of the algorithm, it is necessary to know which different materials are in the scene. It is not important to know the exact appearance, color, or reflectance of each material, but it is important to be able to distinguish the different materials on the different parts of a 3D object. The number of materials can be arbitrary, as long as there are not too many to clearly distinguish between their representation in the database. A possible database is to color-code the materials, so the limitation is the distance of the corresponding colors in the HSV color space. In addition, a tolerance above and below the hue, saturation, and brightness value specified in 1) should be considered, since the color-coded captured image in 2) is taken with a realistically modeled camera and therefore may not provide the same colors as in the scene. For each material, it is possible to set a normalized intensity value with which the different materials are illuminated by the HD matrix headlights. The possible values range from 0 to 1, which are then multiplied by the maximal illumination intensity of the headlight themselves. For example, it is possible to set the normalized illumination intensity of glass to 0.4 and that of wood to 0.6.

In the next step 2), two different virtual images of the scene are taken using a virtual camera with a resolution of $n_\mathrm{x} \times n_\mathrm{y}$ horizontal and vertical pixels. In one of the images, the color coding of the objects' material types in the scene is encoded, and in the other, the depth information in meters from the camera to the objects is encoded. The depth is gained from the Z-buffer of the graphics card and is therefore more precisely the depth on the orthogonal coordinate axis of the image. The captured pictures are shown in Fig. 2a and Fig. 2b, respectively. The information in these images is required for the algorithm to later recognize and distinguish the different materials of the virtual objects and to reconstruct the 3D coordinates of the environment in the headlight system. Direct access to the depth information and the conversion of a realistic visualization of the objects to a color-coded material type library is only possible if a 3D rendering engine is available for the algorithm. If this is not usable, step 2) must be adapted.



(a) Color-coded image for the recognition of materials, each color representing a different one.

(b) Image with depth information of the camera

Fig. 2: Required images for processing material and depth information in the algorithm

In step 3), the color-coded image is searched for the colors of the known materials defined in step 1). As already indicated, the predefined colors are converted from the RGB color space to the HSV color space. In HSV color space, it is more convenient to compare the colors with a tolerance above and below the defined values from step 1) to account for inaccuracies in the image capture. For each material, the hue, saturation, and value of the corresponding color code are compared to each pixel of the captured image of the scene. If the three HSV values of the pixel are within the interval spanned by the tolerance of the material, the pixel is assigned to that material by a corresponding index.

In the end, the indices of the pixels form an overall material image. The default index of the material image is 0, which is set when a color in the image does not represent a color of a known material. An example of such a material image is illustrated in Fig. 3, where the material indices are chosen arbitrarily except for 0.

Fig. 3: Material index image. Each recognized material in the scene is assigned a material index from prior knowledge. For the rest, the unknown material index 0 is assigned.

For step 4) the depth image of step 2) as well as the camera positions $c_{\mathrm{x}}, c_{\mathrm{y}}, c_{\mathrm{z}}$ on the x, y, and z-axis and the rotations $c_{\alpha}, c_{\beta}, c_{\gamma}$ around these axes with respect to the headlight coordinate system are needed. For the transformation of the surface points from the depth image of the camera to the headlight coordinate system, the theorem of intersecting lines is used. In the first step, the length of the z-axis z_{CT} in texture elements (texels) from the camera to the center of the camera texture of the recorded image is calculated, where δ is the field of view angle of the camera

$$z_{\mathrm{CT}} = \frac{\max(n_{\mathrm{x}}, n_{\mathrm{y}})}{2\tan\left(\frac{\delta}{2}\right)}. \tag{1}$$

For each texel of the recorded image texture, the horizontal and vertical texel coordinates $p_{\mathrm{tex,x}}$ and $p_{\mathrm{tex,y}}$, are determined, with the origin at the center of the texture where the z-axis intersects. The z-coordinate of each texel is set to z_{CT}. The corresponding 3D coordinates in the camera system $\boldsymbol{p}_{\mathrm{cam}} \in \mathbb{R}^{3\times1}$ are calculated by scaling the coordinates of each texel $\boldsymbol{p}_{\mathrm{tex}} \in \mathbb{R}^{3\times1}$ containing the elements $p_{\mathrm{tex,x}}$ $p_{\mathrm{tex,y}}$ and z_{CT} as $\boldsymbol{p}_{\mathrm{tex}} = \left(p_{\mathrm{tex,x}}, p_{\mathrm{tex,y}}, z_{\mathrm{CT}}\right)$ by an individual factor s in meters per texel. The scaling s is calculated with the depth image information d in each texel of the depth image

$$s = \frac{d}{z_{\mathrm{CT}}}, \tag{2}$$

$$\boldsymbol{p}_{cam} = s\,\boldsymbol{p}_{\mathrm{tex}}. \tag{3}$$

The affine transformation $\boldsymbol{T} \in \mathbb{R}^{4\times4}$ from the camera coordinate system to the headlight coordinate system

$$\boldsymbol{T} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & c_{\mathrm{x}} \\ r_{21} & r_{22} & r_{23} & c_{\mathrm{y}} \\ r_{31} & r_{32} & r_{33} & c_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

with

$$\begin{aligned} \boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} &= \boldsymbol{R}(c_{\gamma})\boldsymbol{R}(c_{\beta})\boldsymbol{R}(c_{\alpha}) \\ &= \begin{pmatrix} \cos(c_{\gamma}) & -\sin(c_{\gamma}) & 0 \\ \sin(c_{\gamma}) & \cos(c_{\gamma}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(c_{\beta}) & 0 & \sin(c_{\beta}) \\ 0 & 1 & 0 \\ -\sin(c_{\beta}) & 0 & \cos(c_{\beta}) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(c_{\alpha}) & -\sin(c_{\alpha}) \\ 0 & \sin(c_{\alpha}) & \cos(c_{\alpha}) \end{pmatrix}, \end{aligned} \tag{5}$$

is used to obtain the coordinates $\boldsymbol{p}_{\mathrm{HL}} \in \mathbb{R}^{3\times1}$ per texel of the camera texture in the headlight coordinate system by multiplication

$$\begin{pmatrix} \boldsymbol{p}_{\mathrm{HL}} \\ 1 \end{pmatrix} = \boldsymbol{T} \begin{pmatrix} \boldsymbol{p}_{\mathrm{cam}} \\ 1 \end{pmatrix}. \tag{6}$$

Fig. 4 shows the results of the transformation when a normal camera picture without special coloring, e.g. depth information, is mapped onto $\boldsymbol{p}_{\mathrm{HL}}$, with the origin being the middle of the headlight pair.

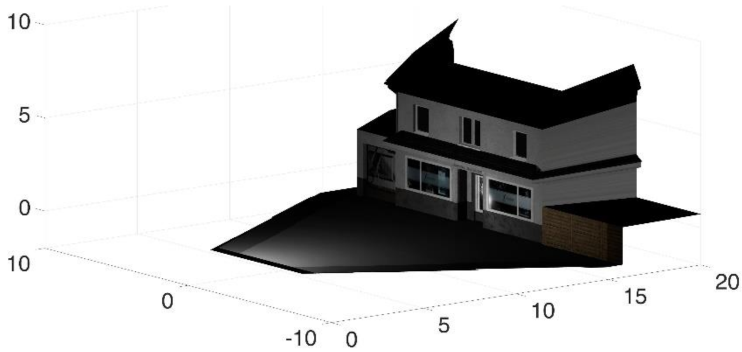


Fig. 4: Reconstructed 3D coordinates with the middle of the headlight pair as origin

In step 5) of the algorithm, the results of steps 3) and 4) are combined to determine the pixel intensity values of the HD matrix headlight. Similar to step 4), the theorem of intersecting lines and the methods of perspective transformation are used for this purpose. This time, the coordinates in the headlamp coordinate system and the texture coordinates of the headlight texture with a z-axis of length z_{HL} in texels in the center are used since a headlight model with projective texture mapping for light simulation is utilized in the rendering engine. Thus, the calculation of the horizontal and vertical coordinates x_{HL} and y_{HL} for the headlight texture from the corresponding world coordinates x_W, y_W, z_{W} with

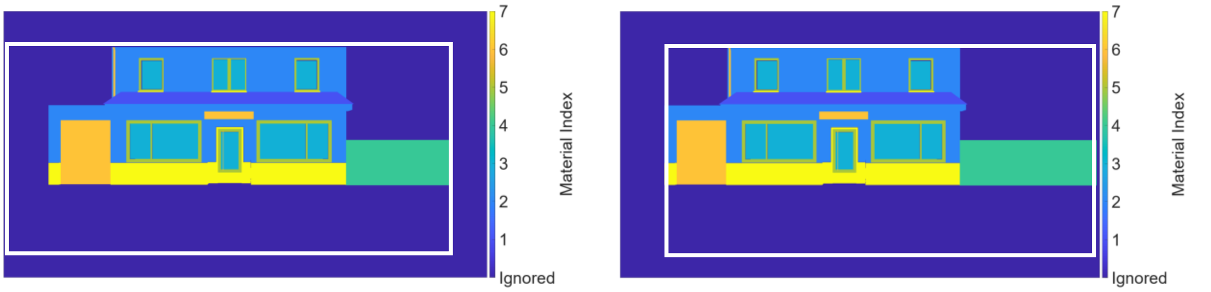
$$z_{HL} = \frac{\max(n_{\mathrm{HL,x}}, n_{\mathrm{HL,y}})}{2\tan(\alpha)}, \tag{7}$$

where the headlight texture has a resolution of $n_{\mathrm{HL,x}} \times n_{\mathrm{HL,y}}$ horizontal and vertical texels and the point light source has an opening angle of α, is

$$x_{\mathrm{HL}} = \frac{z_{\mathrm{HL}}}{z_{\mathrm{W}}}\, x_{\mathrm{W}}, \tag{7}$$

$$y_{\mathrm{HL}} = \frac{z_{\mathrm{HL}}}{z_{\mathrm{W}}}\, y_{\mathrm{W}}. \tag{8}$$

If the x_{HL} and y_{HL} coordinates do not lie within the boundaries of the headlight texture, then the corresponding world coordinates do not fall within the illumination area of the headlight and are therefore ignored. Wherever this condition applies, the corresponding areas of the material image from step 3) are ignored. The same goes for the areas where no known material is detected, as shown in Fig. 5 for each of the two headlight modules.

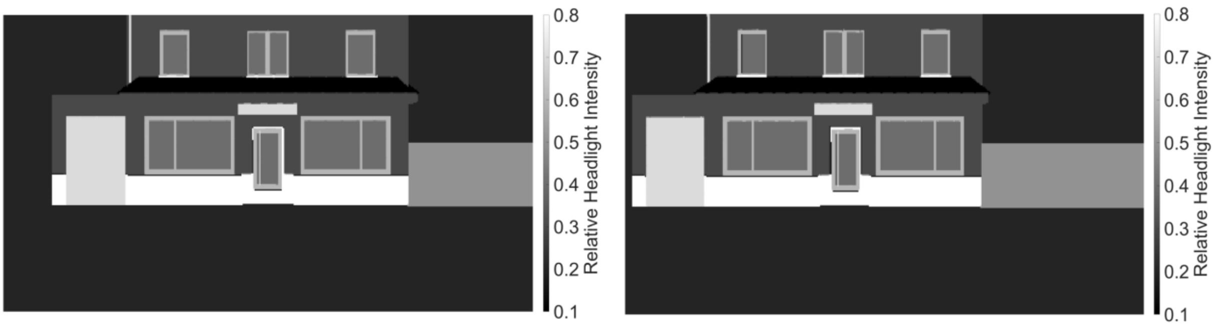


(a) Material image of the left headlight

(b) Material image of the right headlight

Fig. 5: Cropping of material images in relation to the area of possible illumination for each headlight. The areas are shown by a white box.

By comparing the material image with the corresponding x_{HL} and y_{HL} coordinates per headlight, only the x_{HL} and y_{HL} coordinates of the headlight texture that lie within the boundaries of the headlight texture and that illuminate a known material of the environment are left. This creates a list with a pair of the x_{HL} and y_{HL} coordinates of the headlight texture and the corresponding material index that this texel coordinate illuminates. The x_{HL} and y_{HL} coordinates are then transformed to values between one and the number of horizontal and vertical pixels of the matrix headlight, respectively, for indexing. Thus, each texel of the headlight texture can be set to the intensity value of the corresponding material index. If there is more than one coordinate pair belonging to the same texel, the intensity value is calculated by the arithmetic mean of all corresponding coordinate pairs. Therefore, if one pixel of the matrix headlight illuminates two different materials, for example, if a low-resolution headlight is used or it is a borderline pixel, the transition between the different materials is smoother and the materials are not illuminated with a completely different intensity than the intended one. Finally, any texels of the headlight texture that were not set using the above method are set to the default intensity value. The resulting textures for each headlight are shown in Fig. 6, where 0.2 is the default intensity value in this example.



(a) Texture of the left headlight

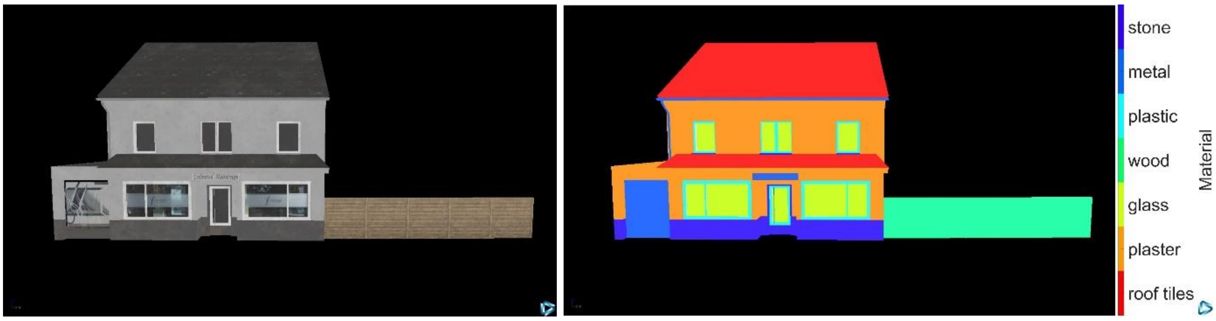
(b) Texture of the right headlight

Fig. 6: Headlight textures with relative pixel intensities to its maxima

As a final step, the resulting headlight texture from step 5) is used as the texture for projective texture mapping for the matrix headlight with a point light source of UE. For more than one headlight, steps 4) and 5) must be repeated with adjusted coordinates and rotations per headlight, which then results in a different headlight texture for step 6). For a pair of two parallel headlights, it is sufficient to perform step 4) once and shift all coordinates for the other headlight according to the distance between the headlights. The previous algorithm can not only be used inside a simulation but can also in principle applied to control real headlights if the depth information and a high-precision HD map of the real environment color-coded by materials are available. Therefore, the headlight texture is an input parameter from which an algorithm must compute the pixel utilization. An optimal approach to do this in real-time is to use the super-sampling method, which we have already published in [6].

3 Evaluation

The presented approach is evaluated virtually in UE using an example scene. The basis of the scene is a multi-story 3D house with a fence at one side, which originates from a highly detailed map from 3D Mapping Solutions of the real German city of Lippstadt. Therefore, the scene represents a real environment. The house and the fence are composed of seven known materials: roof tiles for the roof, glass for the windows, wood for the fence, plastic for the windows frames, metal for signs, the garage door, gutter and door handle, stone for the foundation and plaster for the facade. Fig. 7 shows the unlit scene on the left with the PBR model used by UE for realistic visualization in the simulation, and the color-coded scene on the right. The color coding corresponds to the different materials.



(a) Unlit scene with PBR model

(b) Color coding according to different materials

Fig. 7: Visualization of the base scene in UE

In this contribution, the headlight model with projective texture mapping presented in [7] is used with a pair of HD matrix headlamps. The pixels of each headlight are arranged in a rectangular matrix with a total number of 100,352 and an opening angle of 45° for the directional light source in UE. To analyze the effects on visual perception of automated vehicle cameras, an image is captured from a virtual camera in Unreal. This is evaluated using the latest YOLO neural network, YOLOv8 [1], as a representative of a detection algorithm for computer vision and thus a possible algorithm for the perception of automated vehicles. YOLOv8 was only trained on daytime data.

For evaluation, the relative illumination intensity of the different materials is first varied by the headlights, without objects other than the house and the fence, to observe the visual changes in human perception and the basic functionality of the algorithm. Fig. 8

shows four different illuminations of the scene. The corresponding individual relative intensities for the materials are listed in Table I. The parameters were determined empirically by successive approximation according to the authors' subjective impressions to generate the four examples in Fig. 8. A compromise between brightness and energy efficiency was aimed for. Based on the images, it is shown that depending on the different illumination the effect on the human eye and the impression varies, thus drawing attention to other elements of the scene.


(a) Homogenous pixel intensities


(b) Homogenous appearing illumination


(c) Highlighted windows with strong illumination


(d) Illumination with strong contrast between windows, facade and foundation

Fig. 8: Camera images of the scene in UE with different illumination

TABLE I: Normalized intensity values corresponding to the images of Fig. 8

Material	**Fig. 8a**	**Fig. 8b**	**Fig. 8c**	**Fig. 8d**
Roof Tiles	0.4	0.5	0.1	0.1
Plaster	0.4	0.3	0.3	0.6
Glass	0.4	0.1	0.7	0.1
Wood	0.4	0.3	0.3	0.3
Plastic	0.4	0.1	0.2	0.6
Metal	0.4	0.2	0.2	0.2
Stone	0.4	0.5	0.2	0.2
Default	0.4	0.2	0.2	0.2

For our subjective comparison, fig. 8a displays the scene with a homogeneous light distribution with a total relative intensity of 0.4 for the matrix headlights. Due to the different materials with their different reflective properties and basic colors, some elements of the scene appear brighter and some of them darker, although everything is illuminated with the same light intensity. The attention of the human viewer is drawn to the areas where the greatest contrast is created, e.g., plastic window frames that appear bright compared to windows that appear dark. In contrast, the illumination of the scene shown in Fig. 8b has less prominent areas to the human eye, and the overall image appears more homogenous, so the attention is not drawn to specific regions. In Fig. 8c, there are more reflections on the glass windows, especially the door window, which also changes the perception compared to Fig. 8a because this highlights the glass windows and could potentially blind the human eye. The illumination of the scene in Fig. 8d emphasizes the contrast between the plastic window frames and the glass windows more than when the light intensity is homogenous, while there are fewer reflections. This creates a different perception for the human observer.

The results show that just by modifying the relative illumination intensities of the individual pixels to some extent, the perception of the scene for a human driver and other human traffic participants is different and attention can be directed to different areas. This effect could be used to target attention to safety-related objects that are already part of the scene, or to dynamic objects that are not part of the scene, such as pedestrians in front of the house. For example, a pedestrian might be more visible to human traffic participants if the background appears homogeneous or the contrast to the background is maximized. Conversely, objects irrelevant to traffic can be illuminated in such a way that they are better ignored by the human driver, but not completely out of view. In addition, glare or overexposure of the image can be prevented by selecting different intensity values for highly reflective surfaces such as glass.

To investigate how the proposed algorithm affects the visual perception of cameras of automated vehicles, a person is placed in front of the door in the scene. Then images are taken with a homogeneous intensity distribution as shown in Fig. 9a and two slightly varied illuminations as shown in Fig. 9b and Fig. 9c.

For fig. 9a, all relative intensity values are set to 0.3; for fig. 9b, all relative intensity values except for plaster are set to 0.2 and that for plaster is set to 0.4; and for fig. 9c, all relative intensity values except for plaster are set to 0.3 and that for plaster is set to 0.6. The resulting images of the scene are then analyzed using the YOLOv8 neural network. In this contribution, it is not considered that the person might be blinded by the headlights since in the authors' opinion it does not make sense to apply glare-free high beam to the person and then assess the results with a neural network that is not trained to evaluate such a situation.

(a) Homogenous pixel intensities, no detection

(b) Energy saving illumination

(c) Illumination with higher pixel intensities for plaster

Fig. 9: YOLOv8 detections at different illuminations. Confidence, bounding box and class are shown in blue if detected.

As Fig. 9a shows, the person is not detected by the daytime-trained YOLOv8 neural network with a homogenous intensity distribution. When setting the total illumination except for intensity for the plaster to 0.2 and for the plaster to 0.4, the person is detected with a matching bounding box with a confidence of 67.53%, as illustrated in Fig. 9b in blue. The confidence of Fig. 9c in blue for the detection of YOLOv8 amounts to 78.16%, also with a matching bounding box.

These results demonstrate how it is possible to significantly improve the visual perception of automated vehicles by the proposed algorithm, thus increasing safety at night. Since the number of pixels needed to illuminate the plaster amounts to less than half of all available pixels, as can be seen in Fig. 6, the second option with intensities of 0.2 and 0.4 saves energy while achieving detection with 67.53% confidence compared to no detection. By prioritizing higher confidence over energy consumption, there is also the possibility of further improving confidence, in this case to 78.16%.

The images in Fig. 9 also show that there is a need to further investigate the effects of different illuminations for machine vision algorithms, as they are different from human perception. In the authors' opinion, the detection quality for the human eye does not change as much for the slightly different illuminations as it does for YOLOv8. Since detection quality for machine vision algorithms can be quantified in numbers, such as confidence or intersection over union, rather than for humans where subject studies are required, it is possible to virtually optimize illumination for automated vehicle cameras with a cost function. In this contribution, only exemplary values for the relative pixel intensities were chosen, so while the results are not optimized, they still achieve significant improvement. This also demonstrates the potential of optimization that could yield much better results.

4 Conclusion & Outlook

The contribution at hand proposes a novel algorithm for adjusting the illumination of the environment by HD matrix headlights to improve the detection of traffic-related objects for humans and automated vehicles. The algorithm uses known material properties of the objects in the environment so that the relative pixel intensity per material can be adjusted individually and thus, for example, the amount of reflections reflected from a material can be controlled, affecting the perception of the entire environment. This can influence the glare effect and the main area of attention of the human eye, allowing attention to be directed to traffic-relevant objects. For machine vision algorithms such as YOLOv8, the adapted illumination by the proposed algorithm can make the difference between no detection of a person and detection with a matching bounding box and 78.16% confidence. It is possible to save energy of the matrix headlight while achieving a detection with 67.53% confidence compared to no detection.

The results of manually selected, not optimized relative intensity values show the potential for optimization of illumination, especially for machine vision algorithms, since optimization could be automated without subject studies. This is also emphasized because other initial experiments show that machine vision algorithms are also affected by varying illumination of materials in areas that are not within the range of objects to be detected. Therefore, the next steps include creating a cost function to evaluate the detection quality of machine vision algorithms and building an optimization loop. Future work also includes an automated categorization of arbitrary, unknown virtual objects into materials, so that the algorithm can be used in any virtual environment without prior knowledge.

5 Acknowledgment

All 3D environment objects, e.g. trees and houses, of this contribution, were created by 3D Mappings Solutions. This work is funded by the German Federal Ministry for Economic Affairs and Climate Action as part of the AHEAD project (grant number: 19A21021C), which is a collaboration with Forvia-Hella and 3D Mappings Solutions. The responsibility for the content of this publication lies with the authors.

6 References

[1] Ultralytics (Ed.), “YOLOv8”, available at https://github.com/ultralytics/ultralytics, Visited: 2023-05-08

[2] Epic Games (Ed.), “Unreal Engine 5”, available at https://www.unrealengine.com/en-US/unreal-engine-5, Visited: 2023-05-08

[3] B. Burley, “Physically Based Shading at Disney”, SIGGRAPH, 2012

[4] M. Sega, C. Korobkin, R. van Widenfelt, J. Foran and P. Haeberli, “Fast Shadows and Lighting Effects Using Texture Mapping”, ACM SIGGRAPH Computer Graphics, Volume 26, pp. 249-252, 1992

[5] P. Lecocq, J. Kelada and A. Kemeny, “Interactive Headlight Simulation”, Renault Direction de la Recherche – Research Department, 2004

[6] M. Waldner and T. Bertram: Feedforward Control of HD-Headlights for Automated Driving, 14th International Symposium on Automotive Lighting (ISAL) 2021, 04.-06.04.2022

[7] N. Müller, M. Waldner and T. Bertram, “Virtual Development and Optimization of High-Definition Headlights”, International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2022