Chapter 2

Rendering Techniques in 3D Computer Graphics Based on Changes in the Brightness of the Object Background

Nika Bratuž, Helena Gabrijelčič Tomc and
Dejana Javoršek

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67737

Abstract

Maintaining accurate colour constancy and constant colour appearance are only a few challenges one must conquer in a modern day digital three-dimensional (3D) production. Many different factors influence the reproduction of colour in 3D rendering and one of the most important is certainly rendering engines. In our research, we have studied rendering of colours with three rendering engines (Blender Render, Cycles and Yafaray) of an open source 3D creation suite based on changes in the brightness of the object background from 20 to 80%. In one of these cases, colour of the object was adapted to the lighter background using the colour appearance model CIECAM02. With the analysis of colour differences, lightness and chroma between colours rendered using different rendering engines; we found out that rendering engines differently interpret colour, although the RGB values of colours and scene parameters were the same. Differences were particularly evident when rendering engine Cycles was used. However, Cycles also takes into account the object background. Numerical results of such research provide findings, which relate to the respective environment, and also these certainly demonstrate the successful implementation of the colour appearance model CIECAM02 in the 3D technologies and, in our opinion to other software packages for 3D computer graphics.

Keywords: 3D computer graphics, rendering engines, Blender Render, Cycles, Yafaray, CIECAM02

1. Introduction

The creation of static image in three-dimensional (3D) computer graphic pipeline involves: object modelling, texturing, definition of materials and shading algorithms, illumination, camera setting and rendering. Exact algorithmic description and sampling of light (and consequently

INTECH
open science | open minds

colour) for calculation of final rendering are possible only with the consideration of the basis of radiometry. Radiometry presents the set of mathematic tools, rendering algorithms for description of electromagnetic weaving and light phenomena [1]. The fundamentals of these algorithms are complex and multi-layered; however, for visually accurate 3D CG (i.e. computer generated) imagery, the understanding of the basic reflectance models should be at least understood and implemented in the workflow. In general, the reflection of light can be described with two functions, i.e. BRDF—bi-directional reflectance distribution function and BSSRDF—bi-directional scattering-surface reflectance distribution function [2–4].

The BRDF function was defined by researcher Nicodemus [2] half a century ago and today its application is also well anchored in modern 3D computer graphic solutions. In general terms and as well-known from colourimetry, the mathematical abstraction of BRDF considers parameters of lights source, 3D model with defined textures and materials and the observer (virtual camera), therefore the function could be implemented on all types of 3D object surfaces. In dependence of the angle and the direction of the incident light, the function calculates the radiance value from the 3D object's surface in the observer's direction. Similarly as BRDF, BTDF—bi-directional transmittance distribution function is also defined, calculating the portion and disposition of transmitted light. Based on mathematical foundations of both BRDF and BTDF functions, the BSDF—bi-directional scattering distribution function for the light scattering phenomena on the surfaces and in the materials was also determined [4].

The requirements for achieving photorealistic renderings and description of all optical phenomena in 3D virtual space with different visualization technologies also demanded the development of the function BSSRDF [4]. With BSSRDF, the specific reflectance phenomena in translucent materials with higher portion of light scattering are described. Therefore, the so-called subsurface light transport defines the higher amount of light scattering between the starting point where the light is entering the material and (from the entrance's point of view) very distant exit point. With the implementation of this function, the issues of visualization of natural materials such as skin and wax were solved.

Reflectance models (shading algorithms) as the derivatives of the above-mentioned functions represent the definition of type of interactions between material and light. Regarding their mathematical definition and results of their application on the objects, shading algorithms can be used as: (1) models for diffuse surfaces when they describe the surfaces with partial of total diffuse light reflectance; (2) models for surfaces with specific optical properties (metals, anisotropic materials); and (3) models for specular reflective and transitive surfaces, describing the total and partial specular reflectance and/or transmittance [5–8].

The most basic BRDF function is implemented in Lambert reflectance model, defining entirely diffuse surfaces. This model includes a lot of physical and mathematical simplification; however, it is still adequate for opaque and mat surfaces in CG visualizations [5].

Further, the Phong empirical model was developed with very basic level of consideration of lightning, observer (angle, distance) and normal direction for the calculations of reflected radiance at a surface point. Specular reflection in this specular model is calculated as an exponential function of a cosine function [9].

The further developments in 3D rendering brought new solutions and advanced simulations (mathematical interpretations) of objects during rendering. Models named Oren-Nayar, Torrence-Sparrow, Blinn and models for anisotropic surfaces discuss the object surfaces as an organization of a large number of very small, differently oriented surfaces called microfacet.

With the calculations of light phenomena on a large number of small diffuse surfaces, the Oren-Nayar model [5] describes the partially diffuse material surface (which can include also some specular areas in dependence of incident light). On the contrary, Torrence-Sparrow model [6] was developed for metallic surfaces with specific highlights and shadows. The calculations in this model are performed for a large number of completely specular (metal) surfaces.

Blinn's approach to reflectance calculations involves the mathematical model including exponential averaging of normal vectors' disposition on small surfaces of 3D object. The exponential factors included in Blinn's model determine very rapid changes of normals of smooth object surfaces, while these changes are small for diffuse and relief surfaces [7]. The limitation of Blinn's model is the calculation of symmetrical reflection only, whereas as in nature and in 3D CG imagery many surfaces have asymmetrical light reflections. This has brought to the development of models for specific surfaces. Ashikhmin and Shirley [8] presented the BRDF model for objects with anisotropic surfaces (polished metal, hair and cloth). In this model, the properties of a reflected light in a defined surface point are changed in dependence of the rotation of observation around the defined point.

When considering mathematical description, models for specular and transmissive surfaces are in general simpler as above-mentioned models. The cause of their simplicity can be found in the non-complex interactions between light and material, both in geometrical and physical sense. In these models, both functions BRDF and BTDF calculate specular reflectance of the light rays so that the incident light on the surface is scattered in a specified angle (on totally specular surfaces the angle of incident light rays is identical to the angle of reflected light). When specular transmission occurs, the Sneller law and Fresnel equation are implemented in calculations [10].

Illumination in 3D space can be defined as direct and indirect (any process, which simulates indirect lighting, is also referred as global illumination). The principles of both types of illumination and consequently their equations are different, so that for direct illumination only the illumination directly from light sources is taken into account. In contrast, during the integration of indirect illumination, besides direct light sources, all the objects (background) in the scene are also considered as secondary light sources and different light interactions from all surfaces (materials) are performed and calculated. Rendering engines involve one or more often a set of rendering techniques, which algorithms translate all the data about 3D geometry, textures, materials, illumination and camera (observer) in 2D images.

During last decades, indirect illumination was a subject of various researches [11–15]. Each rendering engine uses its own combination of rendering algorithms and methods and with the implementation of different variations of the functions BRDF, BTDF and BSDF, includes also its own derivation of light transport equation (LTE) [10].

Path tracing was introduced by the researcher Kajiya [11] and is still used in the modern solutions as a version of ''single-direction'' path tracing or bi-directional path tracing [16]. This technique is based on Monte-Carlo equation for light transport. The paths of scattered light rays are generated with the gradual tracing from starting point in the camera and ending point in light sources. The basic parameter of this technique is path sampling that demand a very large number of samples for the quality and accuracy of image generation in one pixel. Rendering times are consequently very consuming. Namely, unsuitable number of samples usually results in rendering ''errors'', i.e. more often noise.

Instant global illumination [13] implements the principle of tracing a lower number of light rays from the light source and re-constructs a defined number of point light sources (so-called virtual light sources) in the positions (points), where the path of rays intersect the scene object and background. In the further procedure, the integrator calculates the radiance of objects surface on the intersecting points, taking into account virtual light sources and laws of indirect illumination.

Beside above-mentioned techniques, the methods of photon mapping and particles tracing are also frequently implemented in the work-flow. These techniques were developed by Jansen [12]. These techniques also use simplifications in calculations with systematic distortions of statistical data during sampling. In rendering procedure, they introduce systematic error into the radiance approximation. The basic idea is to construct a path from lights, where every vertex on the path is treated as sample of illumination. The calculation of optical phenomena (reflection, refraction, transmission, scattering) is performed for every object's surface in the space, considering the optical and colour properties of all objects. The method is proceeded in two phases. In the first phase, the photon map is generated, whereas in the second the variation of ray tracing occurs.

The peak of the development in rendering methods is unbiased rendering techniques. Here, the simplifications and distortions in calculation of final rendered images are minimal [15]. Among these techniques, at least Monte Carlo light transport should be mentioned. The Monte Carlo's method involves bi-directional path tracing of light rays, with the starting point in the observer (camera) and ending point in light source(s). The paths are processed with the modification of paths of rays and with the consideration of indirect illumination also in the parts of the scene, which are excluded from calculations within the other rendering techniques [17].

Even though the general rendering algorithms are known, exact solutions that are implemented in software packages are not open source, neither and apart from Cornell box used for user's testing, there are no standardized method available for objective testing of renderings and visualizations [18]. In fact, considering opinions of the developers, the photorealism in CG imagery was already achieved. However, it must be noted that in some references visual perception of photorealism is actually considered from observers' point of view. Namely, some of the references state that photorealistic accuracy cannot be achieved without perceptual cues that yet remain unsolved [19, 20].

International Commission on Illumination CIE (fr. Commission Internationale de l'Eclairage) established the basis of colourimetry already in the beginning of twentieth century. Nowadays, these bases are also the fundamentals of different derivatives of numerical evaluation of colours [21]. Nevertheless, the accuracy of these fundamentals can be severely discussed [22].

Namely, the results of many researches presented that the perception of colour is not depending only on the observer, stimuli and light source but also on viewing conditions, media where the colour is observed (display, computer display, mobile phones) and specific conditions for each media (overexposure or glare) [23–25]. Besides, the studies demonstrated that cultural context and psychological aspects too have significant influence on the perception of colour [23, 26]. As a result, the application of colourimetry started to spread in different areas and many researches experimented various influences and conditions on colour perception [27–29], including the studies of the colour appearance models [30].

As a definition of the technical committee CIE TC1-34, the colour appearance model is capable to predict perception properties of colour, such as lightness, chroma and hue [31]. Developed by International Commission on Illumination CIE, CIECAM97s was an important step to formation of uniform colour appearance model and is a foundation of actually used CIECAM02 [27, 30, 32, 33]. This simple colour appearance model performs the bi-directional calculations on a large number of data bases and is also, due to its simple structure, practical and applicable on different areas [34, 35]. The model can be used for colour transforms [31], as a connection space in colour management [36, 37], for calculation of colour differences [38, 39], for colour rendering predictions depending on different illumination sources and for definition of metamerism [30, 40]. In our research, the CIECAM02 model was used for calculation of colour transforms during the colour changes of background of a defined object.

In the last decade, the perception of colour and surface properties in 3D generated scenes were analysed with different methods and experiments. The studies of illumination and material influence on renderings revealed that observers perceive the colours that are reproduced in renderings differently as they are predicted by algorithms of various rendering methods [41–43].

Xiao and colleagues [41, 42] demonstrated that by different illumination conditions, there are differences in colour perception between graphical simulations of matte disks and specular spheres. Yang and colleagues [44, 45] presented an expanded study of the correlation between colour perception of surfaces and illumination cues. In these researches, colour constancy was depending on the number of light sources, especially in colour perception of highlights. Meanwhile, the perception of total surface specularity and the perception of the background were discovered not to be so relevant during the observations. In addition, other authors have analysed many aspects of colour constancy and colour perception in 2D and 3D scenes [46–49].

So far, in 3D computer-generated imagery, only preliminary researches about the preservation of uniform colour perception of objects in different observation and illumination conditions were published [50, 51]. Therefore, the review of the references showed that the colour appearance model that would facilitate the prediction of perceptual colour properties as lightness, chroma and hue was still not implemented in 3D virtual space.

In the presented research, we have studied rendering of colours with three rendering engines (Blender Render, Cycles and Yafaray) of an open source 3D creation suite based on changes in the brightness of the object background from 20 to 80%. In one of these cases, colour of the object was adapted to the lighter background using the colour appearance model CIECAM02. One of the main goals of the research was the implementation of the colour appearance model CIECAM02 in the 3D technologies.

2. Experimental

2.1. Methods

2.1.1. Defining test setups

To compare different rendering engines, a simple scene was setup in Blender open source 3D creation suite. Scene was composed from background, object, light source and camera (**Figure 1**). Three rendering engines were used, namely Blender Render, Cycles and Yafaray. Yafaray is an open source Monte Carlo ray tracing engine used for generating realistic images with metropolis ray tracing. Yafaray is used in form of add-on and can generate realistic images using path tracing, bi-directional path tracing and photon mapping. Blender Render is physically non-objective rasterization engine that geometrically projects objects to an image plane without advance optical effects. Cycles is objective, physically unbiased rendering engine that employs path-tracing algorithm. Firstly, a simple grey chart was used to calibrate the scene, since different settings are applied to different rendering engines regarding light intensity. RGB values for each rendering engine were measured to provide repeatability among different rendering engines. Colour management was turned off to achieve accurate RGB values. In Blender software, colour management is not entirely equivalent to colour management used in other professional graphic applications.

Background and object in **Figure 1** are composed of diffuse material with intensity 1 without specular of mirror component. Object was in the shape of a sphere. Diffuse shading model was set to Lambert shader for Blender Render and Yafaray. Cycles only supports BSDF that is composed of Lambert and Oren-Nayar shading model. Camera with automatic settings was set in front of the object at the distance of 10 units and light source was set directly behind the camera. A reflector was set as white light and cone with 120° beam angle of the beam and constant fall-off. Light intensity was changed with rendering engine to achieve repeatability and was 1 for Blender Render, 4000 for Cycles and 14 for Yafaray. Rendering engine settings were as follows: image size was set to 800 × 800 pixels in an 8-bit sRGB colour space, amount

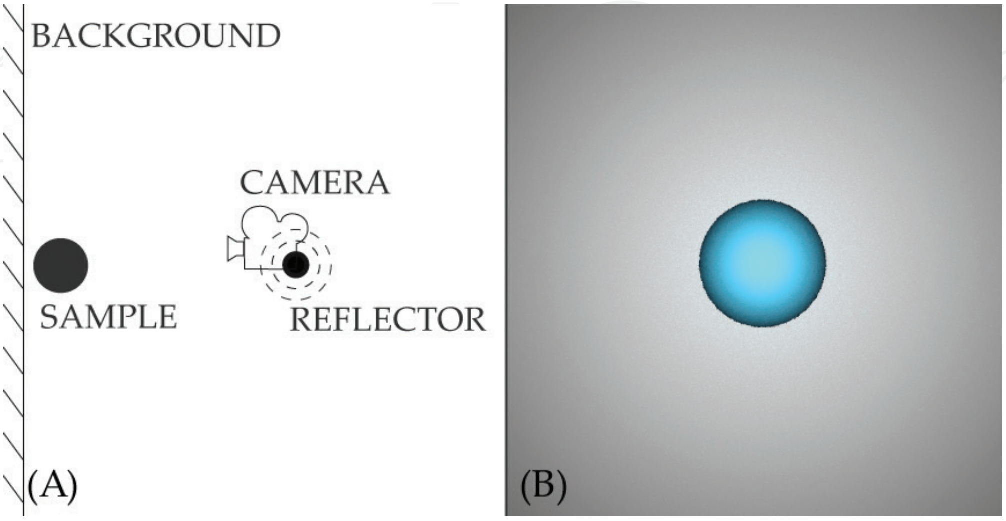

Figure 1. Schematic representation of scene setup in Blender software (A) and rendered image (B).

of anti-aliasing samples per pixel was set to 8 with Gaussian reconstruction filter, all shading options were on, including ray tracing, tile size was set to 64 × 64 units. Rendering was carried out by central processing unit (Intel i7 4770).

Input colours were in the range of RGB = [0, 0, 0] to RGB = [255, 255, 255] with interval of 25.5 units per channel, adding up to 1331 samples. Background was defined as 20 and 80% lightness, meaning RGB20 = [51, 51, 51] for 20% lightness and RGB80 = [204, 204, 204] for 80% lightness. In **Table 1**, important characteristics of each scene element are presented for all rendering engines.

		Blender Render	**Cycles**	**Yafaray**
Object	Colour	1331 colour samples		
	Material	Surface	Diffuse	Glossy-diffuse
	Diffuse	Lambert	BSDF (Lambert in Oren-Nayar)	Lambert
	Amount	1	/	1
	Specular	Off	Off	Off
	Other effects	Off	Off	Off
Background	Colour	RGB20 (20% lightness) and RGB80 (80% lightness)		
	Material	Surface	Diffuse	Glossy-diffuse
	Diffuse shading	Lambert	BSDF (Lambert in Oren-Nayar)	Lambert
	Intensity	1	/	1
	Specular shading	Off	Off	Off
	Other effects	Off	Off	Off
Light source	Lamp	Spot		
	Colour	White		
	Intensity	1	4000	14
	Shape	Cone with 120° beam angle		
Camera	Focal length	35 mm		
	Settings	Auto		
Render settings	Dimensions	800 × 800 pixels		
	Colour space	sRGB		
	Depth	8 bit		
		PNG		
	Colour management	Off	/	/
	Anti-aliasing	Gauss, 8 samples		

Table 1. Rendering engine settings for object, background, light source and camera and general rendering settings.

2.1.2. Colour adaptation with CIECAM02 colour appearance model

Above-mentioned input colours were imported to Blender software as input values RGBi on RGB20 background, as input values RGBi on RGB80 background and as adapted values RGBa on RGB80 background (**Figure 2**). This presents a set of images that will be later used for evaluation. Also, simultaneous contrast can be observed between input colours on RGB20 and RGB80 backgrounds.

Adapted colour was calculated with CIECAM02 colour appearance model as presented in **Figure 3**. From input RGBi values, XYZi values were calculated and used to calculate appearance correlates lightness J, chroma C and hue h, via reverse models adapted XYZa and RGBa were calculated. Following parameters were used in both directions: luminance of the adapting field was set to L_A = 16 cd/m^2 and white point to D65 and surroundings was set to average. Relative luminance of the background was Y_B = 20% for input colours and Y_B = 80% for adapted colours.

2.1.3. Evaluation

Next to input RGBi values and adapted RGBa values, lightest colour RGBs and average colour RGBp on spherical object on rendered image were also obtained. CIELAB values were also

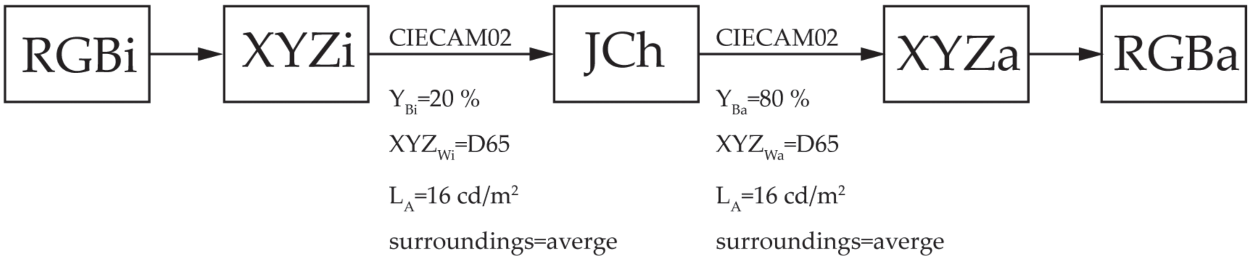

Figure 2. A set of images for colour RGBi = [51, 77, 179] for Cycles rendering engine. From left to right follow input colour RGB on RGB20 background, as input colour RGB on RGB80 background and as adapted RGB colour on RGB80 background.

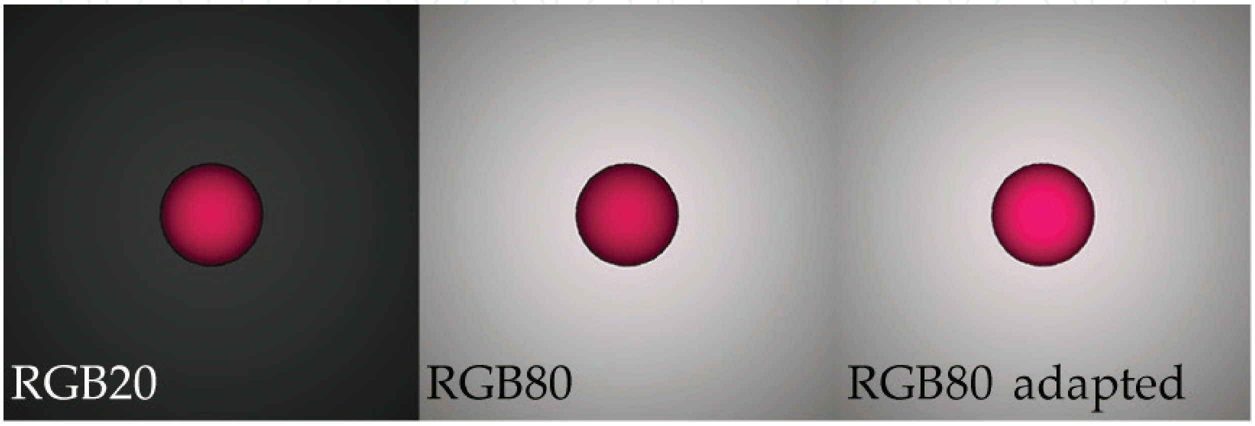

Figure 3. Adaptation workflow, marking 'i' stands for input and marking 'a' for adapted colours.

http://dx.doi.org/10.5772/67737

calculated for each sample of input and adapted RGB values and graphically presented in aforementioned sets. Colour difference ΔE_{00} was calculated between pairs of values, namely between (1) input colour RGBi on RGB20 background and input colour RGBi on RGB80 background and (2) input colour RGBi on RGB20 background and adapted RGBa colour on RGB80 background.

3. Results and discussion

3.1. Colour difference between input and adapted values

Firstly, average colour difference ΔE_{00} between input colour RGBi on RGB20 background and input colour RGBi on RGB80 background (without adaptation) and input colour RGBi on RGB20 background and adapted RGBa colour on RGB80 background was calculated. Results are presented in **Table 2**. It was expected that colour difference would be zero between input colours RGBi on both backgrounds, since input value was actually the same. Despite the fact, there was slight difference between read values for Cycles rendering engine. It can be concluded that in that case, background slightly affects the rendered colour. Average colour difference between input RGBi and adapted RGBa values was greater than zero considering that input value was adapted to different background, thus the colour is slightly changed. Smallest colour difference was obtained for Blender Render, followed by Yafaray. Greatest colour difference was calculated for Cycles.

3.2. Colour difference between input and rendered values

Next, colour difference ΔE_{00} between values that were input into Blender and values that were obtained from rendered images (as lightest colour RGBs and average RGBp colour on spherical object) was calculated for input colours on RGB20 and RGB80 background and adapted colours on RGB80 background. The results are presented in **Table 3**.

It can be noted that colour difference between input and rendered colour on RGB20 and RGB80 remain roughly the same due to the fact that the input colour in the setting was the same between all pairs, which can be deducted from **Table 2**, where colour difference was zero for Blender Render and Yafaray, means background does not affect colour for those two rendering engines.

Rendering engine	**Colour difference ΔE_{00}**	
	20–>80%	**20–>80% CIECAM**
Blender Render (BR)	0	3.28
Cycles (CY)	1.67	6.29
Yafaray (YF)	0	4.42

Table 2. Average colour difference ΔE_{00} between input colour RGBi on RGB20 background and input colour RGBi on RGB80 background (without adaptation) and input colour RGBi on RGB20 background and adapted RGBa colour on RGB80 background.

		Colour difference ΔE_{00}		
		Input RGBi on RGB_{20} background	Input RGBi on RGB80 background	Adapted RGBa on RGB80 background
	Rendering engine	Rendered RGBi on RGB20 background	Rendered RGBi on RGB80 background	Rendered adapted RGBa on RGB80 background
RGBs	Blender Render	3.66	3.66	28.65
	Cycles	10.34	10.85	28.98
	Yafaray	9.11	9.11	34.62
RGBp	Blender Render	0	0	33.11
	Cycles	10.63	10.63	32.83
	Yafaray	9.78	9.87	37.86

Table 3. Average colour difference ΔE_{00} between input and rendered colours for lightest RGBs and average colours RGBp.

Next, a difference between colour difference of lightest RGBs and average RGBp colour on the sphere can be noted. Values vary less than 10% for Cycles and Yafaray, but there is a great difference between lightest and average colour for Blender Render, obviously there is no difference between average RGBp input and rendered colour on both backgrounds. Also, notable difference is present when comparing lightest colour RGBs rendered with Cycles, confirming that Cycles does somehow takes background into account when rendering light colours.

In contrast, colour difference between adapted and rendered adapted colour is high. Colour difference is higher for average colour and lightest colour, which is consistent with non-adapted colours. Values here vary for more than 10%, most for Blender Render, which is quite opposite from non-adapted colours. Presumably, adapted colour is treated differently than non-adapted colour by rendering engines.

3.3. CIELAB evaluation

CIELAB colour values were calculated for all colours and lightness L*, a* and b* co-ordinates of lightest RGBs and average RGBp colours and presented graphically. Following from left to right, input colour RGBi on RGB20 background, input colour RGB on RGB80 background and adapted colour RGBa on RGB80 background are presented in each figure with lightness L* on *y*-axis and sample on *x*-axis.

In **Figures 4** and **5**, a specific grouped pattern can be noted, which is created due to sample selection algorithm, where colours follow in batches from darkest to lightest. Charts for non-adapted colours remain the same but there is increase in lightness of adapted colours, which is a result of adaptation to darker background. This effect can be clearly seen in lower parts of the chart where darker colours resided. In **Figure 5**, samples are set lower on the chart, since lightness L* of average colours is lower, but despite the fact, the effect of adaptation can be seen in darker colours.

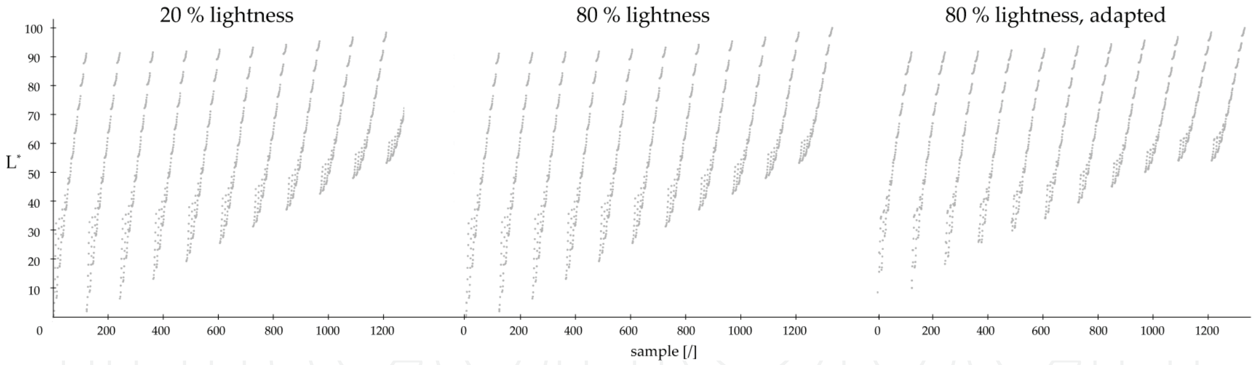

Figure 4. Lightness L* of lightest RGBs colour for each sample for Blender Render.

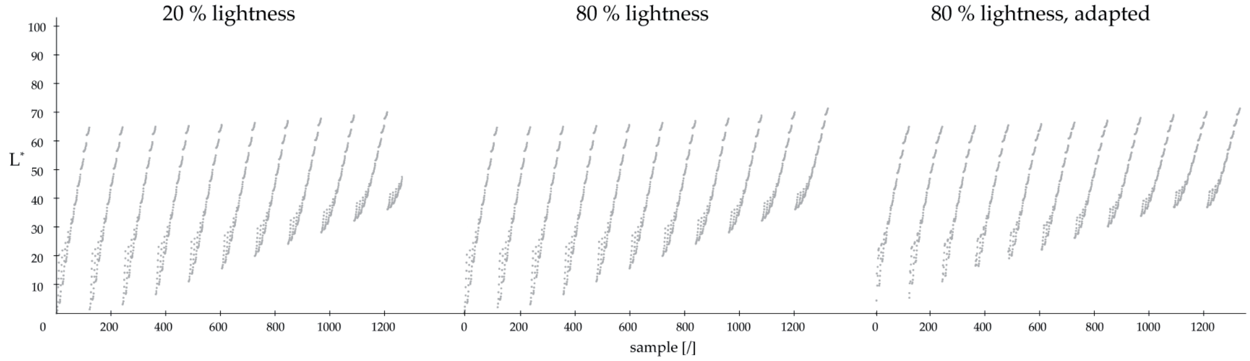

Figure 5. Lightness L* of average RGBp colour for each sample for Blender Render.

In **Figures 6** and **7**, a* and b* co-ordinates for each sample for Blender Render are shown. In **Figure 6**, where a* and b* co-ordinates for lightest RGBs colour are presented, it can be observed that colour space roughly matches sRGB gamut. All renderings took place in sRGB colour space. For adapted colours, there is condensation of samples along lines running from the centre.

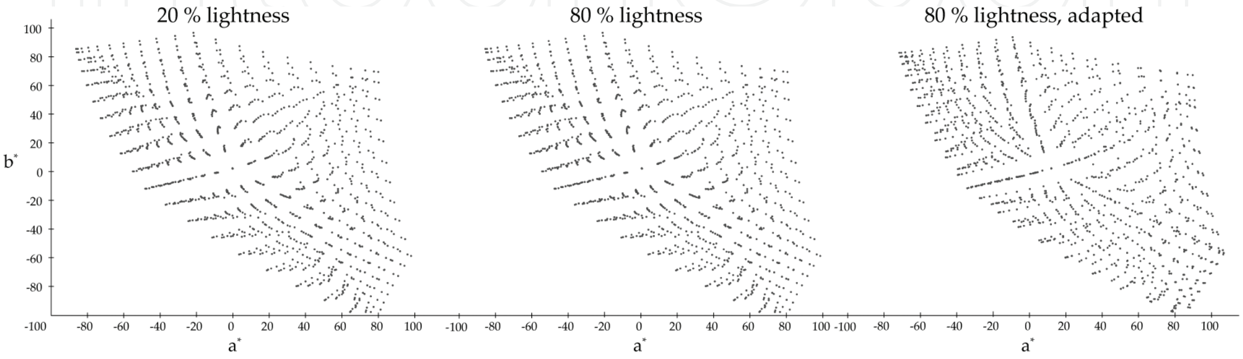

Figure 6. a* and b* co-ordinates of lightest RGBs colour for each sample for Blender Render.

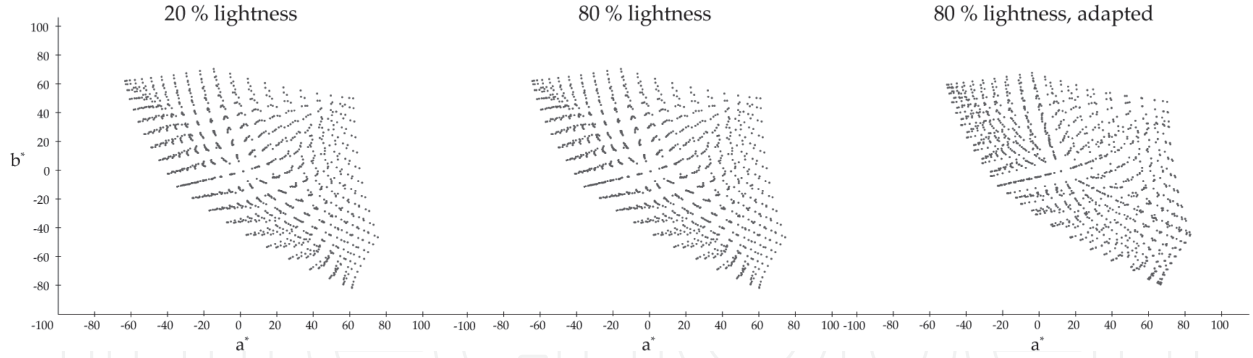

Figure 7. a* and b* co-ordinates of average RGBp colour for each sample for Blender Render.

In **Figure 7**, where a* and b* co-ordinates for average RGBp colour are presented, it can be observed that gamut is smaller than for lightest colours. Again, condensation can be visible for adapted colours.

In **Figure 8**, lightness L* of lightest RGBs colour for each sample rendered with Cycles is shown. In comparison with Blender Render, samples are grouped in parts were lightness is higher, so the arrangement of samples is different. Lightness does not grow constantly as in case of Blender Render. There is a jump when RGB values go over 125. Here, samples are scattered in lighter regions, meanwhile for Blender Render, there is scattering in darker regions. Lightest colours have higher lightness L* in comparison to Blender Render and CIECAM02 makes colours lighter to achieve constant colour appearance.

In **Figure 9**, lightness L* of average RGBp colour for each sample for Cycles is presented. Lightness chart is similar as in case of Blender Render, but the jump in lightness for colour with RGB over 125 is still visible, but not to such extent. The effect of adaptation can still be visible.

Some outstanding phenomena can be observed in **Figure 10**, where a* and b* co-ordinates of lightest RGBs colour for each sample for Cycles are presented. Adapted colours are grouped in a few groups along the lines emerging from the centre. In this case, largest colour difference was obtained in **Table 3**. In **Figure 11**, where a* and b* co-ordinates of average RGBp colour for each sample for Cycles are shown, this anomaly cannot be observed to such extent. Based

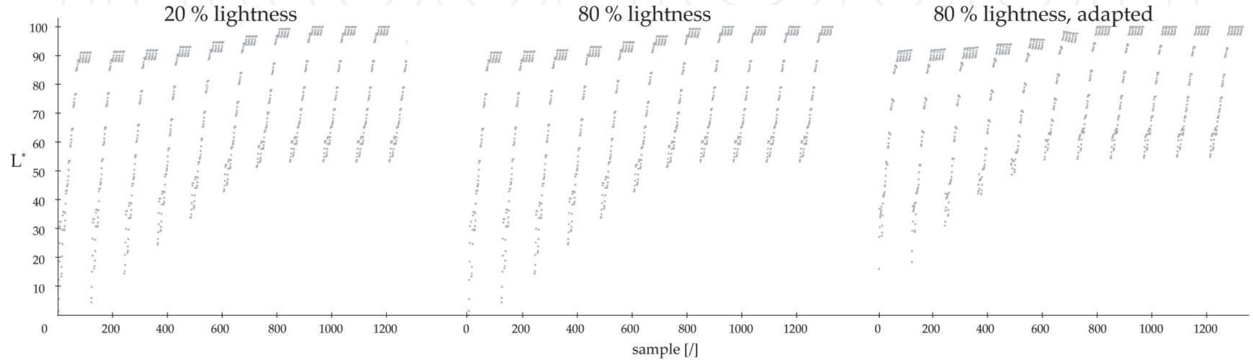

Figure 8. Lightness L* of lightest RGBs colour for each sample for Cycles.

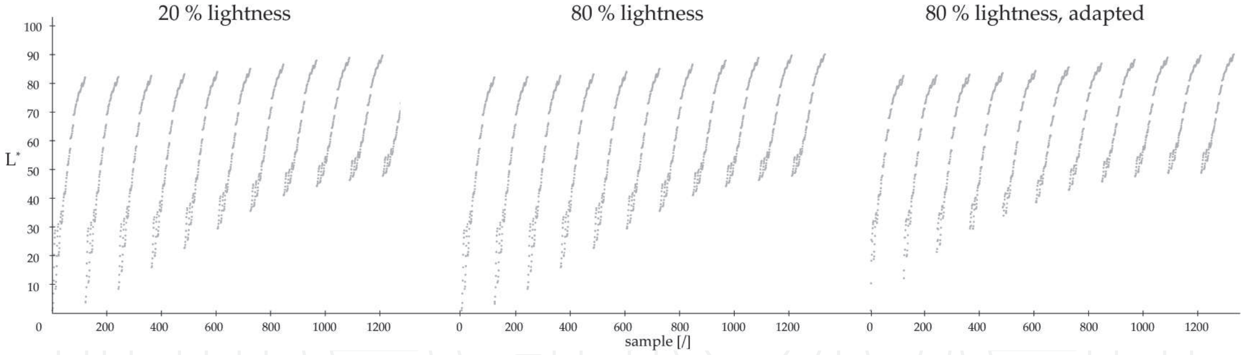

Figure 9. Lightness L* of average RGBp colour for each sample for Cycles.

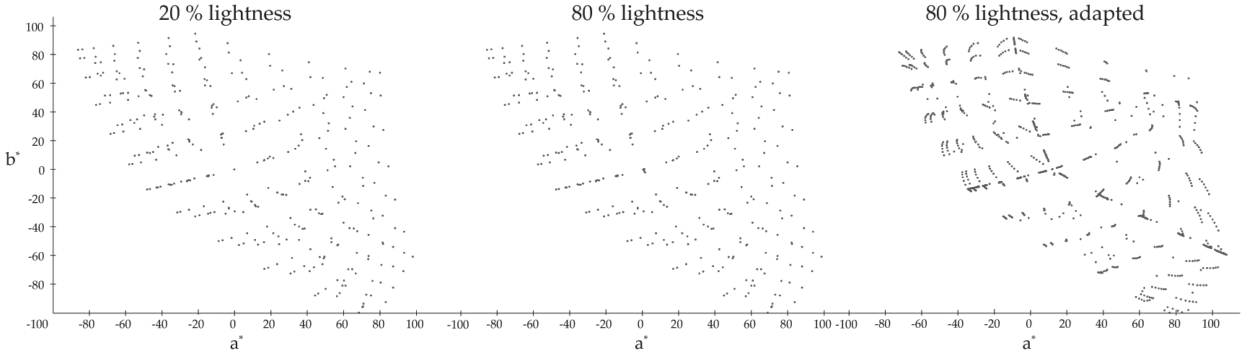

Figure 10. a* and b* co-ordinates of lightest RGBs colour for each sample for Cycles.

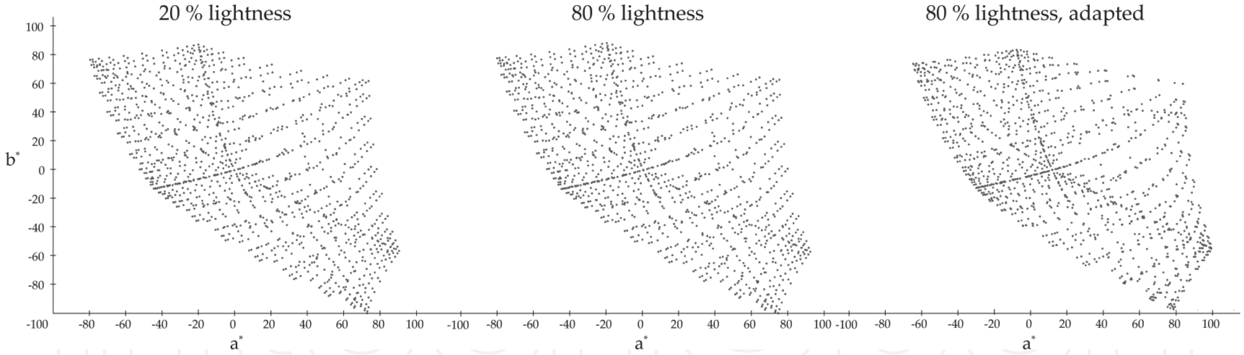

Figure 11. a* and b* co-ordinates of average RGBp colour for each sample for Cycles.

on this observation and colour differences, it can be concluded that lighter colours are treated differently by Cycles than darker colours. Gamut of both charts still resembles that of sRGB colour space.

The results for Yafaray are presented in following figures. In **Figure 12**, lightness L* of lightest RGBs colour for each sample for Yafaray is presented. If compared to Blender Render, there is again more scattering in lighter regions, same as for Cycles. In addition, the jump in lightness for colours with RGB higher than 125 can be visible too. Adaptation effect can also be visible,

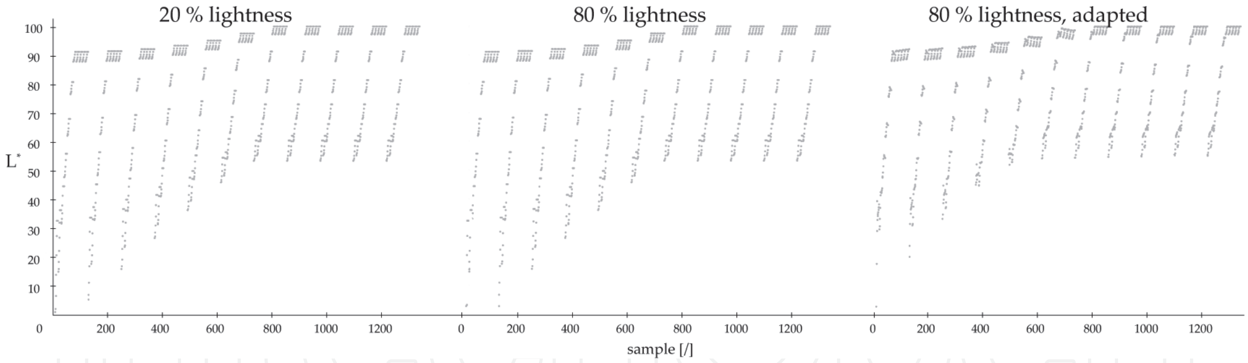

Figure 12. Lightness L* of lightest RGBs colour for each sample for Yafaray.

same as in previous instances. Lightness L* of average RGBp colour obtained for each sample for Yafaray is similar to those of Cycles (**Figure 13**). Again, adaptation affects lightness of colours but not as notable as for lightest colours.

In **Figure 14**, a* and b* co-ordinates of lightest RGBs colour for each sample for Yafaray are presented. Again, gamut matches sRGB colour space. In comparison to Blender Render, samples are condensed into groups for input colours too, meanwhile adapted colours are similar to Cycles, and even though it is not reflected in colour differences. In **Figure 15**, a* and b* co-ordinates of average RGBp colour for each sample for Yafaray are shown. Again, there is difference between input and adapted colours but not as pronounced as in previous case.

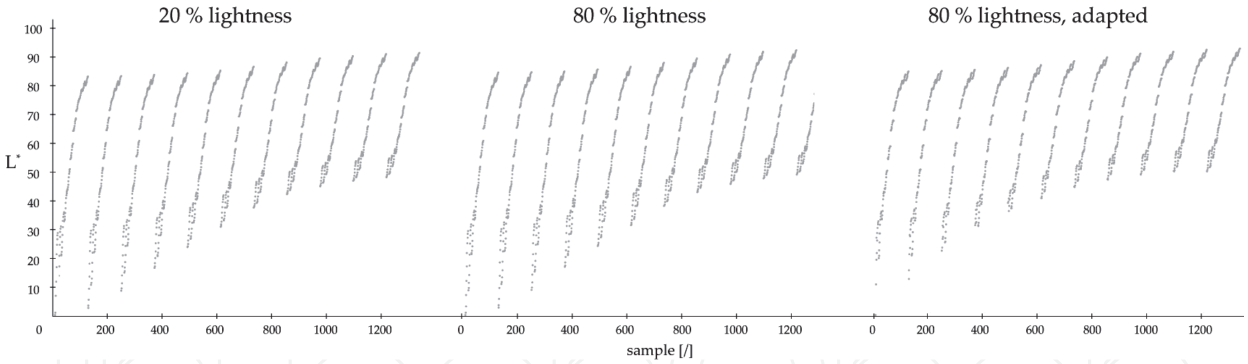

Figure 13. Lightness L* of average RGBp colour for each sample for Yafaray.

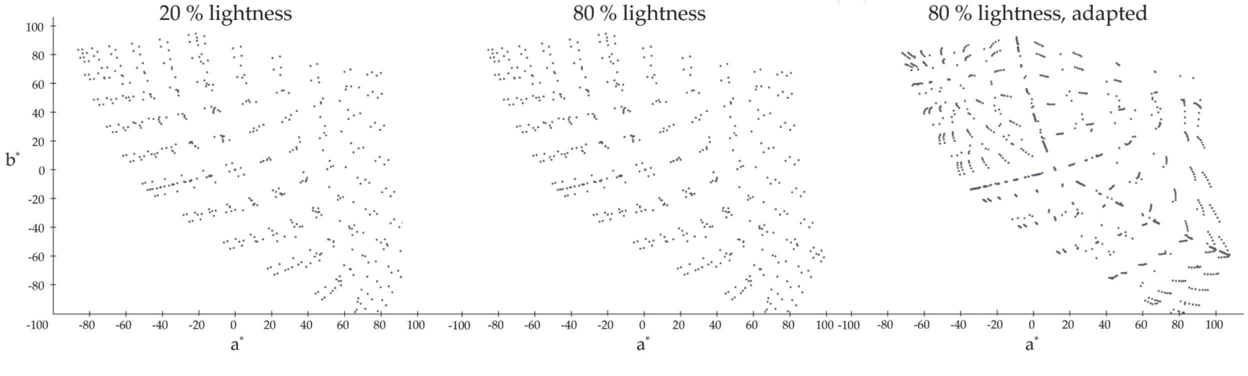

Figure 14. a* and b* co-ordinates of lightest RGBs colour for each sample for Yafaray.

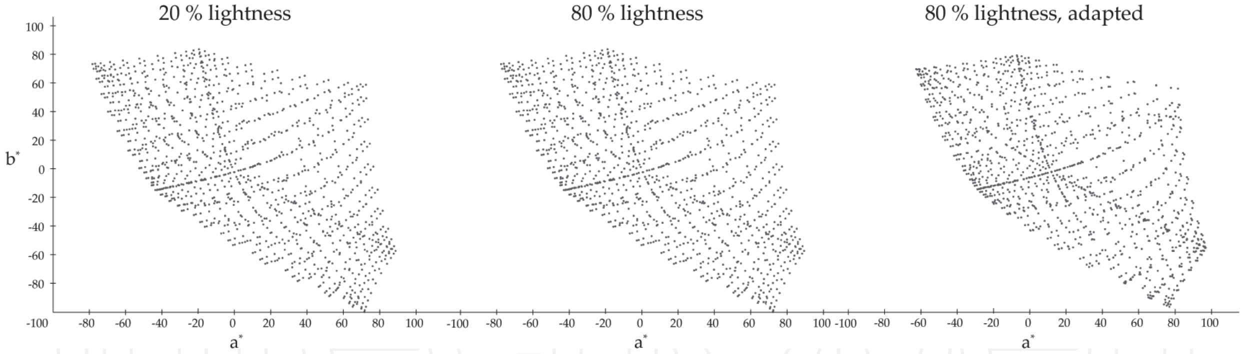

Figure 15. a* and b* co-ordinates of average RGBp colour for each sample for Yafaray.

By comparing CIELAB values, it can be concluded that rendering engines do not treat all colours equally. Even though Cycles and Yafaray do not apply same shading algorithms, yet results are surprisingly similar. It was ascertained that colour differences are largest for Cycles and same fact was confirmed by the analysis of CIELAB values. Moreover, there is a notable influence of the background on rendered colour for Cycles.

3.4. Relationship between L*s and L*p

In **Figures 16–18**, the relationship between lightness of lightest colour L*s in the image (on *x*-axis) and average colour L*p in the image (on *y*-axis) is presented in sets for input colour on RGB20, input colour on RGB80 and adapted colour on RGB80. In **Figure 16**, this relationship is presented for Blender Render. It can be seen that this relationship is quite linear and roughly follows $y = 0.7x$ function. Lightness L*p is lower than L*s, which was expected due to the fact that shading (therefore darkening) is applied on an object. It can be observed that this relationship is uniform for lighter colours; meanwhile grouping can be observed for darker colours. It can also be noted that when adaptation is carried out, colours shift towards lighter colours. This shift is most visible in the range of darker colour with L*s = [0–10]. The same phenomenon was observed previously when analysing CIELAB values.

In **Figure 17**, this relationship is presented for Cycles. It can be observed, that this relationship is not linear, colours are scattered and their position is depending on colour. Again, darker

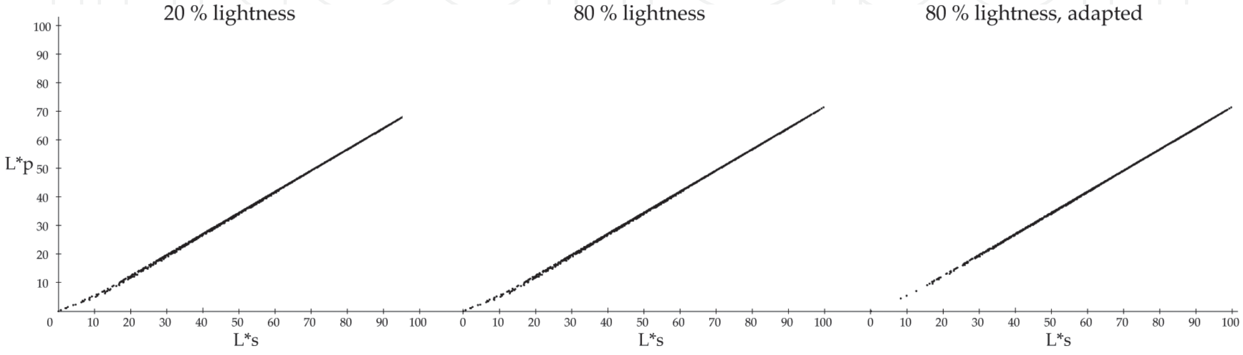

Figure 16. Relationship between lightness of lightest colour L*s in the image and average colour L*p in the image for Blender Render.

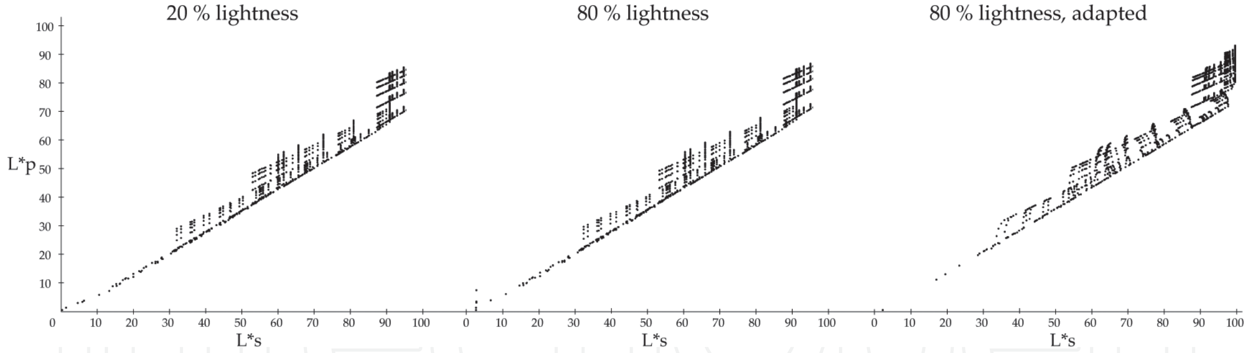

Figure 17. Relationship between lightness of lightest colour L*s in the image and average colour L*p in the image for Cycles.

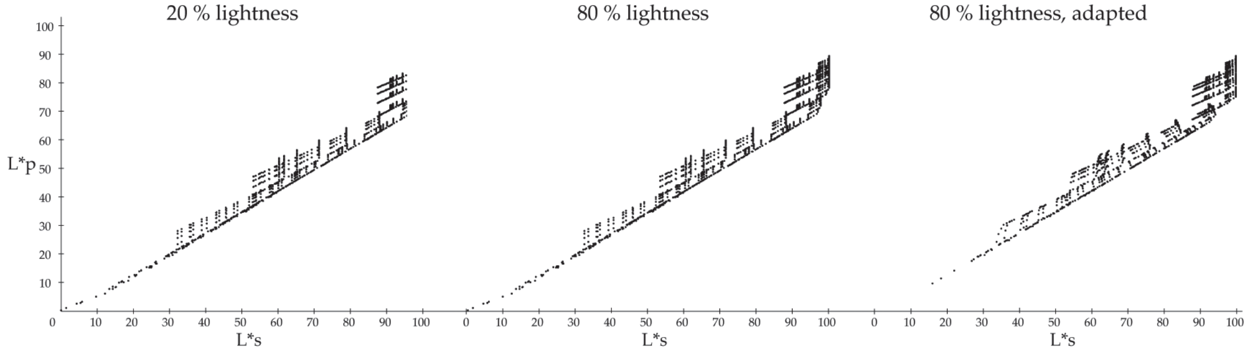

Figure 18. Relationship between lightness of lightest colour L*s in the image and average colour L*p in the image for Yafaray.

colours are grouped, meanwhile lighter colours scatter in groups depending on lightness and is most notable in range L*s = [90–100]. Interestingly, despite contiguous growing of scattering, there is quite uniform range of L*s = [80–90]. For adapted colours, a colour shift is again visible, but in this case it is dependent on lightness. The distance of the shift depends on lightness, this is visible in range of L*s = [30–40]. In this range, lighter colours are shifted more than darker, which is contrary to previous conclusions.

In **Figure 18**, this relationship is presented for Yafaray and it can be observed that results are similar to those for Cycles and are in accordance with CIELAB values analysis. Only difference is visible for lightest colours, they seem to be less scattered as for Cycles. Again, adapted colours shift towards lighter colours with some non-linear shifts, similar to Cycles.

After analysing the relationship between lightness of lightest colour L*s in the image and average colour L*p in the image for all rendering engines, it can be concluded that Blender Render shades colours linearly, meanwhile the shading is more complex in case of Cycles and Yafaray and that it depends on individual colours or colour groups. For those two rendering engines, it was noted that the colours separate into groups. Shading of darker colours inclines towards linear, meanwhile shading pattern of lighter colours cannot be easily defined. A notable difference was perceived between input colours on lighter and darker background. Relationship was the same in case of Blender Render, meaning that it does not consider background. In contrast, the relationship changed for Cycles and Yafaray, meaning that those two rendering engines do consider background and/or surroundings when rendering image. This phenomenon was not observed for Yafaray when analysing CIELAB values.

3.5. Relationship between C*s and C*p

In **Figures 19–21**, the relationship between chroma of lightest colour C*s in the image (on *x*-axis) and average colour C*p in the image (on *y*-axis) is presented in sets for input colour on RGB20, input colour on RGB80 and adapted colour on RGB80. In **Figure 19**, this relationship is presented for Blender Render and it can be observed that similarly to previous section is quite linear with some deviation in regions with lower chroma. Chart is even more uniform in terms of chroma after adaptation.

In **Figure 20**, chroma relationship is presented for Cycles and it can be observed that the relationship is still quite linear for input colours but more spread meaning that Cycles does not treat colours based on chroma. After adaptation, chroma of darker colours decreases and chroma of lighter colours increases, again depending on original colour.

In **Figure 21**, this relationship is presented for Yafaray and again similarities with Cycles can be observed. Likewise Cycles, relationship remains linear for input colours but colours are spread along *y*-axis. In contrast with Cycles, region with lower chroma C*s < 10 is also covered and after adaptation, chroma in darker regions increases.

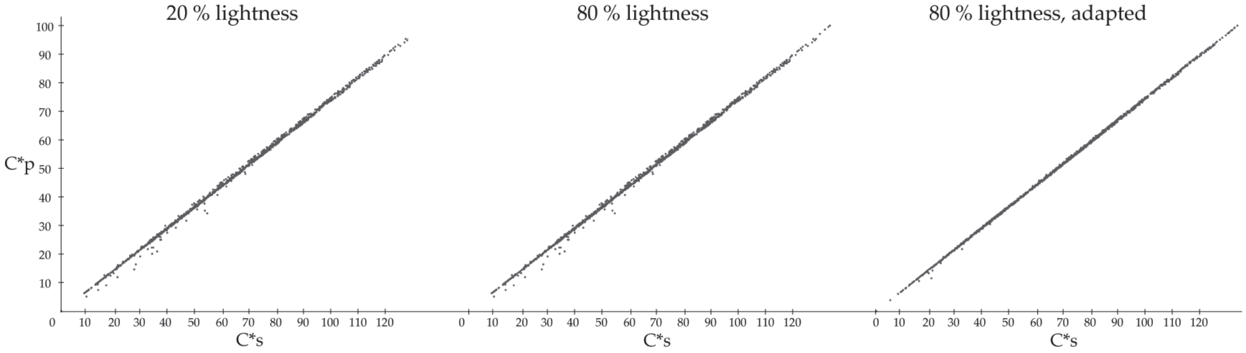

Figure 19. Relationship between chroma of lightest colour C*s and the image and average colour C*p in the image for Blender Render.

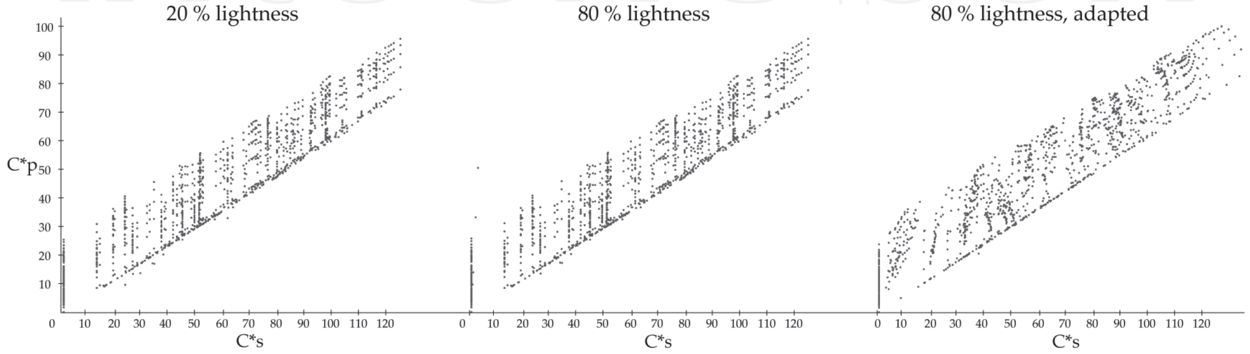

Figure 20. Relationship between chroma of lightest colour C*s and the image and average colour C*p in the image for Cycles.

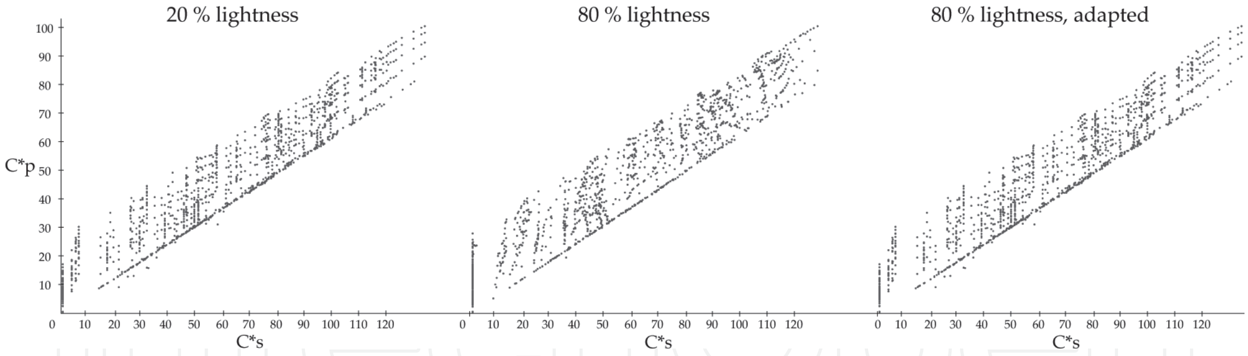

Figure 21. Relationship between chroma of lightest colour C*s and the image and average colour C*p in the image for Yafaray.

Again, Blender Render shades colour uniformly based on chrome unlike Cycles and Yafaray where shading is complex and depending on each colours. In comparison with lightness relationship, colours were not divided in groups but spread more equally. Similarly, difference noted between input colours on lighter and darker background was again visible for Cycles and Yafaray, but not as much as lightness. It can be concluded that chroma affects shading too. Chroma of adapted colours is shifted but not in same directions as lightness and not in same way for all rendering engines.

3.6. Relationship between Rs and L*p, Gs and L*p and Bs and L*p

Finally, relationship between each RGB component of lightest colour (Rs, Gs and Bs on *x*-axis) and lightness of average colour L*p (on *y*-axis) was graphically presented. *z*-axis was introduced to avoid overlapping. Charts are presented in sets for input colour on RGB20, input colour on RGB80 and adapted colour on RGB80.

In **Figure 22**, this relationship is presented for Blender Render. It can be observed that charts are the same for input colour on RGB20 and RGB80 background. The effect of adaptation is visible in terms of lightness and RGB values and that there are some anomalies in darker regions of R and B component which was observed in preliminary research. It can also be observed that CIECAM02 effects lightest colour only minimally.

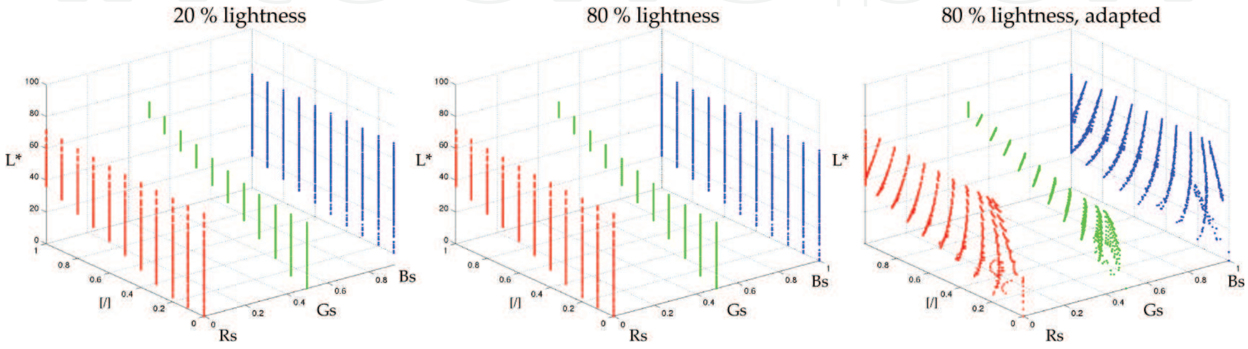

Figure 22. Relationship between Rs and L*p (red), Gs and L*p (green), Bs and L*p (blue) in the image for Blender Render.

In **Figure 23**, this relationship is presented for Cycles and compared to Blender Render, colours are differently grouped and cover wider range of lightness. There is a visible difference for input colour on RGB20 and RGB80 background, same as previous analysis. Again, there is notable difference between input and adapted colours and there are larger changes for darker colours when adapted.

In **Figure 24**, this relationship is presented for Yafaray and results are again similar to those for Cycles, the difference for input colour on RGB20 and RGB80 background is only minimal. Effects of adaptation are again similar to those for Cycles.

From the results, it could be concluded that Blender Render shades colours linearly, meanwhile shading for Cycles and Yafaray is more complex. Both rendering engines consider when rendering colour.

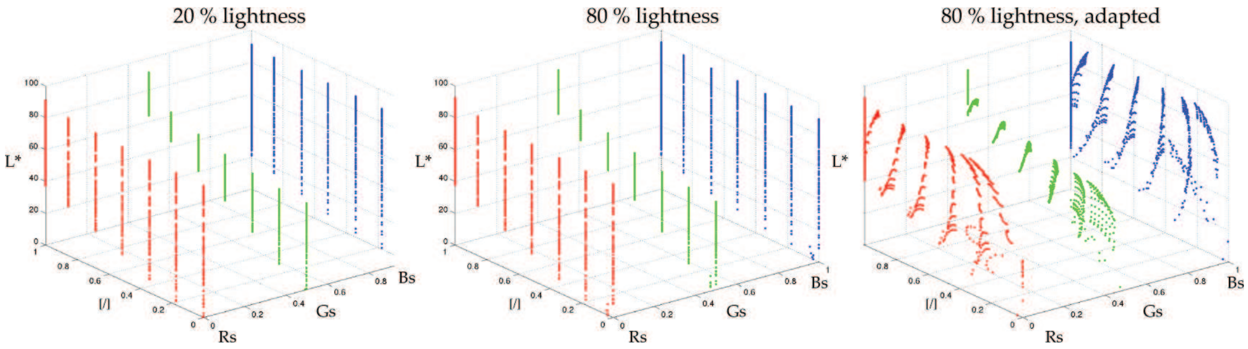

Figure 23. Relationship between Rs and L*p (red), Gs and L*p (green) and Bs and L*p (blue) in the image for Cycles.

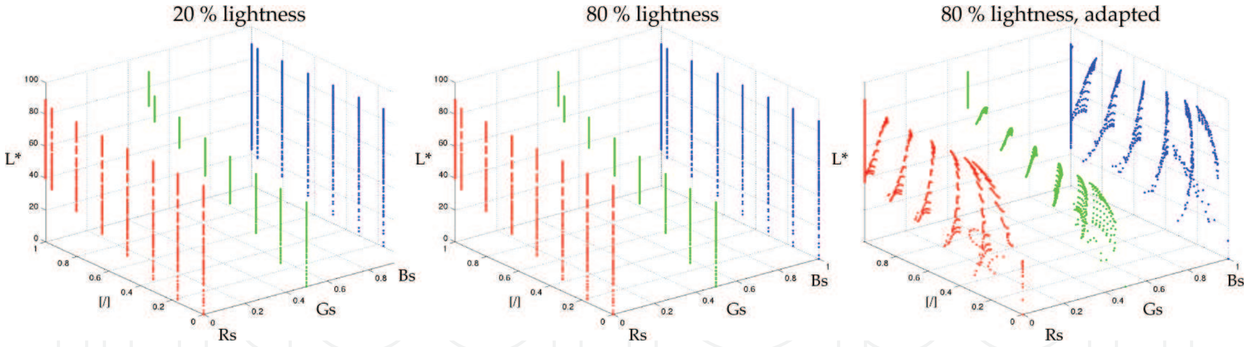

Figure 24. Relationship between Rs and L*p (red), Gs and L*p (green) and Bs and L*p (blue) in the image for Yafaray.

4. Conclusion

The process of producing 2D image from 3D mathematically described that space is very complex, since it depends on many factors that cannot be completely influenced by user.

One of these factors is certainly rendering of colour, and not only in the field of 3D technologies but also in the field of other computer graphics. Despite the progress and the availability of information and knowledge in this field, there is still no clear answer about colour perception, since this phenomenon is, besides physical factors also dependent on some other factors. With the emergence of new complex algorithms for rendering and visualization, the problem of colour reproduction has increased and, despite the established methods there are still no universal solutions to ensure constant colour appearance. Although the colour appearance models, more specifically CIECAM02, have been in use for many years, it has yet not been analysed and explored to ensure constant colour appearance in the field of computer graphics. Demonstration of successful implementation of the model CIECAM02 was possible only empirically, so it was necessary to determine how algorithms interpret colour during simulation and reproduction and also how rendering engines interpret colour on a 2D rendered image after setting the 3D scene.

For that purpose, in our research, we have studied rendering of colours with three rendering engines (Blender Render, Cycles and Yafaray) of an open source software Blender based on changes in the lightness of the object background from 20 to 80%. In one of the cases, colour of the object was adapted to lighter background using the colour appearance model CIECAM02.

With the analysis of colour differences, lightness and chroma between colours rendered using different rendering engines, we found out that rendering engines differently interpret colour, although RGB values of colours and scene parameters were the same. Differences were particularly evident when rendering engine Cycles was used. Results showed that Blender Render treats colour more linearly than other two more advanced rendering engines, Cycles and Yafaray, at least in case of lightness and chroma. In the case of Cycles and Yafaray, colours and change of their properties are considered non-linear, while shading varies depending on the colour and its properties. In addition, we found out that especially Cycles, due to the using of indirect lighting in the colour renderings, also takes into account the object background, since results were different when the background was changed without any adjustments to the colour.

Implementation of the CIECAM02 model in the colour rendering workflow of used rendering engines was successful, since its use in all three cases resulted generally in a better visual match (smaller colour differences) between pairs of stimuli at a controlled change in defined parameters. We have also found out that the quality and quantity of maintaining colour appearance depends on the principle of rendering engines operation and on lightness and chroma of a rendered colour.

The possibilities for further research to in-depth understanding of rendering engines and shading's influence on colour are possible in terms of integration of larger number of objects and backgrounds colours, and also the inclusion of visual assessment with a larger number of observers.

Author details

Nika Bratuž, Helena Gabrijelčič Tomc* and Dejana Javoršek

*Address all correspondence to: helena.gabrijelcic@ntf.uni-lj.si

Department of Textiles, Graphic Arts and Design, Faculty of Natural Sciences and Engineering, Chair of Information and Graphic Arts Technology, University of Ljubljana, Ljubljana, Slovenia

References

[1] McCluney R. Introduction to Radiometry and Photometry. 2nd ed. Norwood: Artech House Publisher; 2014. 240 p.

[2] Nicodemus FE. Directional reflectance and emissivity of an opaque surface. Applied Optics. 1965;**4**(7):757–776. DOI: 10.1364/AO.4.000767

[3] Bartell FO, Dereniak EL, Wolfe WL. The theory and measurement of bidirectional reflectance distribution function (BRDF) and bidirectional transmittance distribution function (BTDF). In: Proceedings of SPIE, Radiation Scattering in Optical Systems, Hunt GH, editor. Huntsville; 1980. pp. 154–160. DOI: 10.1117/12.959611

[4] Donner C, Lawrence J, Ramamoorthi R, Hachisuka T, Jensen HW, Nayar S. An empirical BSSRDF model. ACM Transactions on Graphics (TOG)—Proceedings of ACM SIGGRAPH 2009. 2009;**28**(3):1–10. DOI: 10.1145/1531326.1531336

[5] Nayar M, Oren SK. Generalization of Lambert's reflectance model. Petrovich L, Tanaka K, editors. In SIGGRAPH 94 21st International ACM Conference on Computer Graphics and Interactive Techniques; July 24–29 1994; Orlando, Florida, USA. New York, NY, USA: ACM; 1994. pp. 239–246. DOI: 10.1145/192161.192213

[6] Torrance KE, Sparrow EM. Theory for off-specular reflection from roughened surfaces. Journal of the Optical Society of America. 1967;**57**(9):1105–1114. DOI: 10.1364/JOSA.57.001105

[7] Blinn JF. Models of light reflection for computer synthesized pictures. In: SIGGRAPH '77 Proceedings of the 4th annual conference on Computer graphics and interactive techniques; July 20–22 1977; San Jose, California, USA. New York, NY, USA: ACM; 1977. pp. 192–198. DOI: 10.1145/965141.563893

[8] Ashikhmin M, Shirley P. An anisotropic Phong BRDF model. Journal of Graphics Tools. 2000;**5**(2):25–32. DOI: 10.1080/10867651.2000.10487522

[9] Phong BT. Illumination for computer generated pictures. Communications of the ACM. 1975;**18**(6):311–317. DOI: 10.1145/360825.360839

[10] Pharr M, Humphreys G. Physically based rendering: from theory to implementation. 2nd ed. Burlington: Morgan Kaufman; 2010. 1200 p.

[11] Kajiya J. The Rendering Equation. Wolfe R, editor. In: Seminal Graphics: Pioneering Efforts That Shaped The Field; New York, NY, USA: ACM; 1998.

[12] Jansen HW. A Practical Guide to Global Illumination using Photon Maps. Pueyo X, Schröder P, editors. In: Proceedings of the Eurographics Workshop on Rendering Techniques 96; June 7–19 1996; Porto, Portugal: Springer; pp. 21–30.

[13] Wald I, Kollig T, Benthin C, Keller A, Slusallek P, Debevec P, Gibson S, editors. Interactive global illumination using fast ray tracing. EGRW '02 Proceedings of the 13th Eurographics workshop on Rendering; June 26–28 2002; Pisa: Italy; pp. 15–24.

[14] Kurt M, Edwards D. A survey of BRDF models for computer graphics. ACM SIGGRAPH Computer Graphics—Building Bridges—Science, the Arts & Technology. 2009;**43**(2). DOI: 10.1145/1629216.1629222

[15] Kouhzadi A, Rasti A, Rasti D, Why NK. Unbiased Monte Carlo Rendering – A Comparison Study. International Journal of Scientific Knowledge. 2013;**2**(3):1–10.

[16] Lafortune, EP, Willems YD. Bi-directional Path Tracing. Santo HP, editor. In: Third International Conference on Computational Graphics and Visualization Techniques; December 5–10 1993; Alvor, Algarve, Portugal. Lisbon, Portugal: Association for Computing Machinery, Portuguese ACM Chapter; 1993. pp. 145–153.

[17] Veach E, Guibas LJ. Metropolis light transport. Owen SG, Whitted T, Mones-Hattal B, editors. In Proceedings of the 24th annual conference on Computer graphics and interactive technique; August 3–8 1997; Los Angeles, CA, USA. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 1997. pp. 65–76.

[18] Vidmar Ž, Gabrijelčič Tomc H, Hladnik A. Performance Assessment of Three Rendering Engines in 3D Computer Graphics Software. Acta Graphica. 2014;**25**(3–4):101–114.

[19] Ferwerda JA. Three varieties of realism in Computer graphic. Rogowitz BE, Pappas TN, editors. In: Proc. SPIE 5007, Human Vision and Electronic Imaging VIII, 290; August 20 2003; Santa Clara, CA, USA: SPIE; 2003. DOI: 10.1117/12.473899

[20] McNamara, A. Visual perception in realistic image synthesis. Computer Graphics forum. 2001;**20**(4):211–224. DOI: 10.1111/1467-8659.00550

[21] CIE technical report: Colorimetry. 3st ed. Vienna, Austria: Commission Internationale de l'Eclairage; 2004. 72 p.

[22] Fairman HS, Brill MH. Hemmendinger H. How the CIE 1931 color-matching functions were derived from Wright-Guild data. Color Research and Application. 1997;**22**(1):11–23. DOI: 10.1002/(SICI)1520-6378(199702)22:1<11::AID-COL4>3.0.CO;2–7

[23] Hurlbert A. Colour vision: putting in context. Current Biology. 1996;**6**(11):1381–1384. DOI: 10.1016/S0960-9822(96)00736-1

[24] Winawer J, Witthoft N, Frank MC, Wu L, Wade AR, Boroditsky L. Russian blues reveal effects of language on color discrimination. Proceeding of National Academy of Science of United States of America. 2007;**104**(19):7780–7785. DOI: 10.1073/pnas.0701644104

[25] Hunt RWG. The reproduction of colour. 6th ed. Chichester: John Wiley and sons; 2004. 702 p.

[26] Trstenjak A. Psihologija barv. Ljubljana: Inštitut Antona Trstenjaka za psihologijo, logoterapijo in antropohigieno; 1996. 494 p.

[27] Javoršek A. Preizkus modela barvnega zaznavanja CIECAM97s v standarnih pogojih [thesis]. Ljubljana: University of Ljubljana; 2004. 38 p.

[28] Süsstrunk S E. and Finlayson G D. Evaluating chromatic adaptation transform performance. In: Thirteenth Color Imaging Conference Color Science and Engineering Systems,

Technologies, and Applications; November 7–10 2005; Scottsdale, Arizona, USA. Society for Imaging Science and Technology; 2005. pp. 75–78.

[29] Xiao K, Wuerger S, Fu C, Karatzas D. Unique hue data for colour appearance models. Part II. Color research and application. 2013;**38**(1):22–29. DOI: 10.1002/col.20725

[30] Fairchild M. Color Appearance Models. 3th ed. Chichester: John Wiley and sons; 2013. 385 p.

[31] Bratuž N, Javoršek A, Javoršek, D. Barvne pretvorbe v CIECAM02 in CIELAB. Colour Transforms in CIECAM02 and CIELAB. Tekstilec. 2013;**56**(3):222–229.

[32] Luo MR, Hunt RWG. The structure of the CIE 1997 colour appearance model (CIECAM97s). Color research and application, 1998;**23**(3):138–144. DOI: 10.1002/(SICI)1520-6378(199806)23:3<138::AID-COL5>3.0.CO;2-R

[33] Fairchild M. Status of CIE color appearance models. Chung R, Rodrigues A, editors. In: Proceedings. SPIE 4421, 9th Congress of the International Colour Association, 550; April 2 2001; Rochester, NY, USA: SPIE; 2002. DOI: 10.1117/12.464726

[34] Li C, Luo MR, Hunt RWG, Moroney N, Fairchild MD, Newman T. The performance of CIECAM02. Chung R, Rodrigues A, editors. In: 10th Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications; 2002; Scottsdale, AZ, USA: The International Society for Optics and photonics; 2002.

[35] Brill MH, Mahy M. Visualization of mathematical inconsistencies in CIECAM02. Color research and application, 2012;**38**(3):188–195. DOI: 10.1002/col.20744

[36] Specification ICC.1:2010: Image technology colour management – architecture, profile format, and data structure. Reston, VA, USA: International Color Consortium, 2010. 113 p.

[37] Tastl I, Bhachech M, Moroney N, Holm J. ICC Color Management and CIECAM02 [Internet]. 2005 [Updated: 2005]. Available from: http://www.hpl.hp.com/news/2005/oct-dec/CIC05_CIECAMICC_final.pdf [Accessed: 12. 9, 2016]

[38] Melgosa M, Trémeau A, Cui G. Chapter 3: Colour Difference Evaluation. In: editor, Advanced Color Image Processing and Analysis. New York: Springer Science+Business Media; 2013. pp. 59–78.

[39] Windows Color System [Internet]. 2014 [Updated: 2014]. Available from: https://msdn.microsoft.com/en-us/library/dd372446(v=vs.85).aspx [Accessed: 13. 9. 2016]

[40] Guay R, Shaw MQ. Dealing with Imaginary Color Encodings in CIECAM02 in an ICC Workflow. Chung R, Rodrigues A, editors. In: 13th Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications; 2005; Scottsdale, AZ, USA: The international society for optics and photonics; 2005.

[41] Xiao B, Brainard DH. Surface gloss and color perception of 3D objects. Visual Neuroscience, 2008;**25**(3):371–385. DOI: 10.1017/S0952523808080267

[42] Xiao B, Hurst B, MacIntyre L, Brainard DH. The color constancy of three-dimensional objects. Journal of Vision, 2012;**12**(4):1–15. DOI: 10.1167/12.4.6

[43] Fleming RW, Dror RO, Adelson EH. Real-world illumination and the perception of surface reflectance properties. Journal of Vision, 2003;**3**(5):347–368. DOI: 10.1167/3.5.3

[44] Yang JN, Maloney LT. Illuminant cues in surface color perception: Tests of three candidate cues. Vision Research, 2001;**41**(20):2581–2600. DOI: 10.1016/S0042-6989(01)00143-2

[45] Yang JN, Shevell SK. Surface color perception under two illuminants: The second illuminant reduces color constancy. Journal of Vision, 2003;**3**(5):369–379. DOI: 10.1167/3.5.4

[46] Ruppertsberg AI, Bloj M. Reflecting on a room of one reflectance. Journal of Vision, 2007;**7**(13):1–13. DOI: 10.1167/7.13.12

[47] Ruppertsberg AI, Bloj M, Hurlbert A. Sensitivity to luminance and chromaticity gradients in a complex scene. Journal of Vision, 2008;**8**(9):1–16. DOI: 10.1167/8.9.3

[48] Guzelj A, Hladnik A, Bračko S. Examination of colour emotions on a sample of slovenian female population. Tekstilec, 2016;**59**(4):311–320. DOI: 10.14502/Tekstilec2016.59.311-320

[49] Hedrich M, Bloj M, Ruppertsberg AI. Color constancy improves for real 3D objects. Journal of Vision, 2009;**9**(4):1–16. DOI: 10.1167/9.4.16

[50] Bratuž N, Gabrijelčič Tomc H, Javoršek D. Application of CIECAM02 to 3D scene in Blender. Urbas R, editor. In: 7th Symposium of Information and Graphic Arts Technology; June 5–6 2014; Ljubljana, Slovenia. Ljubljana: University of Ljubljana, Faculty of Natural Sciences and Engineering, Department of Textiles; 2014.

[51] Bratuž N, Jerman T, Javoršek D. Influence of rendering engines on colour reproduction. Novaković D, editor. In: 7th International Symposium on Graphic Engineering and Design; November 13–14; Novi Sad, Serbia. Ljubljana: University of Novi Sad, Faculty of Technical Sciences, Department of Graphic Engineering and Design; 2014.