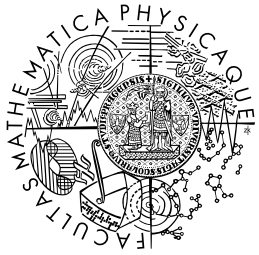


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bc. Lucia Tódová

Constrained Spectral Uplifting

Department of Software and Computer Science Education

Supervisor of the master thesis: doc. Alexander Wilkie, Dr.

Study programme: Computer Science

Study branch: Computer Graphics and Game Development

Prague 2020

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

Dedication.

Title: Constrained Spectral Uplifting

Author: Bc. Lucia Tódová

Department: Department of Software and Computer Science Education

Supervisor: doc. Alexander Wilkie, Dr., Department of Software and Computer Science Education

Abstract: Abstract.

Keywords: key words

Contents

Introduction **2**

1 Color Science **3**
1.1 Light and Color . . . 3
1.2 Color representation . . . 5
1.2.1 Spectral representation . . . 5
1.2.2 Tristimulus representation . . . 6
1.2.3 Color representation in rendering . . . 11

2 Spectral Uplifting **15**
2.1 Uplifting methods . . . 15
2.2 Constrained spectral uplifting . . . 18
2.2.1 Spectral sampling . . . 19

3 Implementation **20**
3.1 Borgtool . . . 20
3.1.1 Optimizer . . . 20
3.1.2 Choice of parameters . . . 20
3.2 Cube constraints . . . 20
3.3 Filling the cube . . . 20

4 Results **22**
4.1 Number of moments . . . 22
4.1.1 Evaluation of parameters . . . 22
4.1.2 Reconstruction results . . . 22
4.2 Spectral uplifting . . . 22
4.2.1 Choice of threshold . . . 22
4.2.2 Sigmoid vs Moment method . . . 22
4.2.3 Constrained input . . . 23

Conclusion **24**

Bibliography **25**

Introduction

1. Color Science

Color science, or colorimetry, concerns itself with human perception of color. It researches the relations between human vision and physical properties of color, and analyzes options for both its capturing and reconstruction.

We begin this chapter by describing the physical properties of light and their subsequent meaning in terms of color. We then provide multiple options for quantifying said color for further possible reconstruction in the digital world(?). Lastly, we show the importance of color representation in modern-day renderers, such as Mitsuba or Corona (add a link).

1.1 Light and Color

The core of human visual perception is electromagnetic radiation, which consists of waves that propagate through space and transmit radiant energy.

An *electromagnetic wave* is characterized by its *amplitude* and *frequency*. Amplitude is defined as the distance between the central axis and either the *crest* (the highest point of the wave) or the *trough* (the lowest point of the wave), while frequency specifies how many wave cycles happen in a second. Together, these properties give rise to the term *wavelength*, denoted λ, which measures the length of the wave — the distance between either two subsequent crests, troughs or any two following spots with the same height.

Every electromagnetic wave can be unambiguously defined by its wavelength. Arranging them according to this criterion creates a classification known as *electromagnetic spectrum* (see fig. 1.1). As the electromagnetic spectrum contains all existing types of electromagnetic radiation, it covers wavelengths in the range from fractions of nanometers to thousands of kilometers. This range is divided into bands to distinguish known types of light; low frequency light such as gamma rays or X-rays; extremely high frequency light such as radio waves.

In this thesis, we will focus on *visible light*, which covers only a mere fraction of the electromagnetic spectrum. Its waves are roughly in the 380-780nm range.

To sum up, electromagnetic waves specify the way in which light travels. To, however, describe the interaction between light and matter, the term *photon* is used. Photons are elementary particles of light moving in a manner specified by their wavelengths, making up electromagnetic radiation. They can be emitted or absorbed by atoms and molecules. During this process, they transfer energy either from the object that emitted them or to the object that absorbed them. This change in energy (denoted E) is proportional to the frequency of the absorbed/emitted photon and can be computed as follows [10]:

$$E = hf = \frac{hc}{\lambda} \tag{1.1}$$

where h is Planck's constant, f is the frequency and c is the speed of light. Therefore, generally speaking, the human eye identifies light when atoms and molecules in the retina absorb photons.

To specify this process, we will first describe the retina. The retina consists of millions of light-sensitive cells, also called *photoreceptors*, which pass a visual

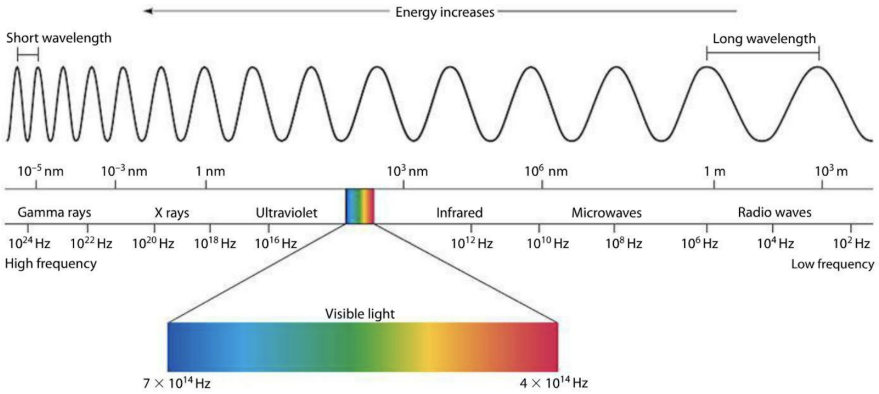


Figure 1.1: An illustration of the electromagnetic spectrum [2]

signal via an optic nerve to the brain, giving the notion of light and color. There are two types of photoreceptors in the human eye — rods and cones.

Rods make up most of the receptor cells (around 91 million according to Purves et al. [27], but other sources state that their number could be as high as 125 million [37]). They are usually located around the boundary of the retina, and are responsible for low light (scotopic) vision. However, they possess very little notion of color, which is also the reason why the human eye has trouble recognizing colors during the night.

Cones are located mainly in the center of the retina and their numbers are a lot lower (from around 4.5 million [27] to 6 million [37]). In contrast to rods, they are active at daylight levels (responsible for photopic vision) and have the notion of color. To be specific, different types of cones differ in their sensitivity to photon energies at concrete wavelengths. The final color is then composed by the brain from the stimulation signals sent by each cone.

The human eye has three types of cones:

- *L-cones*, which are the most responsive to longer wavelengths at around 560nm. When they are stimulated, they correspond to the red color.
- *M-cones*, which are the most sensitive to medium wavelengths at around 530nm and correspond to green color
- *S-cones*, which respond the most to small wavelengths that peak at around 420nm and correspond to blue color

Their relative response to stimulation can be seen in fig. 1.2.

This type of color perception is called *trichromatic*, as it uses three types of receptors to create the whole color space.

The idea behind using three base colors has been adapted in color science to create multiple tristimulus color representations. We will discuss these more thoroughly in the following section.

Up until now, we have been talking about the interaction of light with the human eye. Photons, however, also interact with objects. As established by the relationship defined in eq. (1.1), the energy transferred to an object upon light interaction is dependent on the photon wavelength. This means that objects might absorb some wavelengths and reflect others.

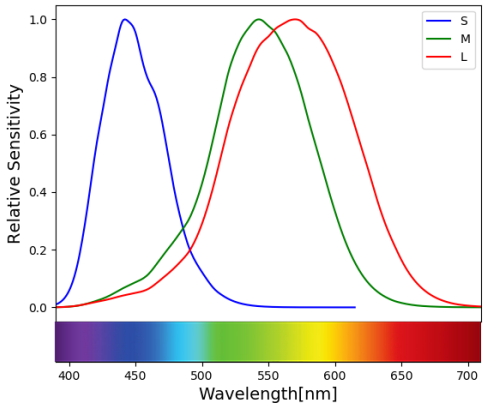


Figure 1.2: Relative sensitivity of S, M and L-cones plotted according to the data measured by Stockman and Sharpe [32].

Object color is defined by the wavelengths it *reflects*. For example, if it reflects all the wavelengths, the resulting color is white, while absorbing all the wavelengths would render the object black. Naturally, human perception of object color is not only dependent on its reflective properties, but also on the lighting of the scene. If the only light present in the scene is red, other wavelengths than red will never hit the object. Therefore, the object might reflect only a subset of wavelengths than it would under white light, which might change the resulting color.

1.2 Color representation

The question of how to discretely represent color has been posed ever since the introduction of the first graphical user interface. For use in computer science, representations are required to be compact, precise, and the operations on colors should be easily executed.

We have already briefly mentioned the tristimulus representation in the previous section. In this section, we will overview its basic properties and describe some of the most popular tristimulus systems. We will also talk about an alternative representation, based primarily on the physical properties of color — spectral representation.

1.2.1 Spectral representation

When defining the color of an object, we must not only specify the wavelengths it reflects, but also the ratio between the incoming energy and the outgoing energy at these wavelengths. The dependence of reflectance on the wavelength is called a *reflectance spectrum*, and is usually a smooth, continuous curve.

Although this definition might be sufficient for reflective surfaces, describing the color emitted by a light source requires the knowledge of the source's power rather than reflectance. For these purposes, *spectral power distribution* (SPD) is used. Generally, SPD is a function describing the relationship between wavelength and any radiometric or photometric quantity (radiant energy, luminance,

luminous flux, irradiance et cetera...). In this thesis, we will use SPD to describe the emissive properties of light sources, and will therefore consider SPD to be a function of wavelength and power.

To compute the color of an object under a light source, one must simply combine the light source's SPD with the reflectance curve of the object, as shown in figure. This way the physical properties of color are preserved and the result is the same as it would be in nature, which, as we will show later, is not always the case with tristimulus representation.

1.2.2 Tristimulus representation

The obvious drawback of spectral representation is the difficulty of its discretization. Another, bigger problem, is caused by the fact that there is an infinite number of possible spectral curves, but only a discrete number of colors perceptible by human eye or even possible to generate by a computer (use word domain?). Representing colors with spectral distribution therefore requires their conversion to a discrete space before arbitrary visualization process.

Tristimulus representation skips the conversion steps and saves the already discretized color as a set of three values. Although the original idea was to simulate the trichromatic perception of human eye (i.e. save values that specify how much have the red, green and blue cones been stimulated), over time, multiple other tristimulus color spaces have been created. They differ mostly in the range of colors they are capable of representing and in their practical use. Following, we provide an overview of some of the most popular ones.

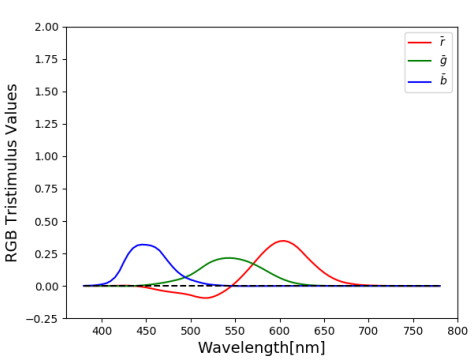
RGB color space

The RGB color space is an additive space employing three primaries — red, green and blue. In other words, if you have three lights with red, green and blue chromacities respectively and you use them to illuminate a single point, you can create any color within the RGB color space solely by changing the lights' intensities.

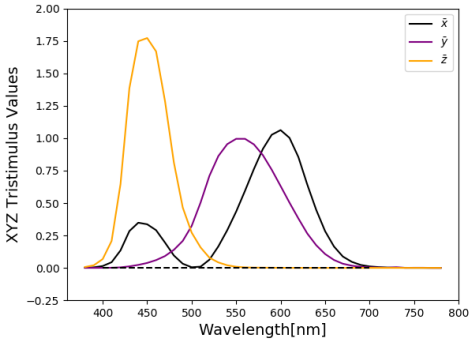
An RGB value can be therefore thought of as a point in a 3-dimensional euclidean space with each of the coordinate axes representing one of the primaries. Specifically, as the light's intensities must be bounded, we can narrow this space down to a cube starting at the base of the coordinate system. Usually, the range for each value is defined within 0 and 255, but a normalized (0,1) range is also used.

Various implementations of the RGB color space exist. They differ in the specifications of the RGB primaries, and therefore in their *color gamut*, which is the subset of colors they are capable of representing. Some examples (named in ascending order with respect to their color gamut) include ISO RGB, sRGB, Adobe RGB, Adobe Wide Gamut RGB and ProPhoto RGB. An illustrative comparison of the sRGB and Adobe RGB gamut in the chromaticity diagram (described thoroughly in section 1.2.2) can be seen in fig. 1.4.

RGB color spaces are commonly used in everyday world, e.g. in LCD and LED displays, digital cameras, scanners and even in computer graphics rendering. Their main downside has, however, been discovered when designing color matching functions [9].



(a) $\overline{r}(\lambda)$, $\overline{g}(\lambda)$ and $\overline{b}(\lambda)$ functions plotted with data by Broadbent [4]



(b) $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$ functions according to their spectral data from Choudhury [6]

Figure 1.3: Color matching functions

A *color matching function* is a function designed to simulate the response of a certain type of cone in the human eye. In 1931, CIE designed a set of three color matching functions that could be used for spectral to RGB conversion [9]. Denoted $\overline{r}(\lambda)$, $\overline{g}(\lambda)$ and $\overline{b}(\lambda)$, they approximate the response of the L, M and S cones respectively. However, as seen in figure fig. 1.3a, the functions may also acquire negative values. This posed a problem at that time due to calculation errors. Therefore, to eliminate these negative portions of functions, CIE designed a new, imaginary color space — the XYZ color space (tu ref?).

XYZ color space

The XYZ color space is a hypothetical color space capable of encompassing all colors perceptible by the human eye. Its color matching functions, $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$, were specifically designed for the purposes of SPD to tristimulus conversion, which is computed using the following equations:

$$\begin{aligned} X &= \int P(\lambda)\overline{x}(\lambda)d\lambda, \\ Y &= \int P(\lambda)\overline{y}(\lambda)d\lambda, \\ Z &= \int P(\lambda)\overline{z}(\lambda)d\lambda, \end{aligned} \tag{1.2}$$

where X, Y and Z are the resulting tristimulus values and $P(\lambda)$ is the spectral power distribution.

Although the X, Y and Z primaries were designed so that the Y primary closely matches luminance and X and Z primaries give color information, they are only imaginary, i.e. they do not correspond to any spectral distribution of wavelengths. This property renders the whole XYZ space imaginary, which means that it cannot be used for visualization purposes. Its main function is to therefore serve as a "middle step" when performing a conversion from SPD to an arbitrary tristimulus space, which eliminates the need for other color matching functions. The conversion from XYZ into a tristimulus space can then be performed by a simple space-specific $3x3$ matrix transformation.

xyY color space

In addition the impossible visualization process, another downside of the XYZ color space is that its values are practically unbounded and do not have any real meaning (such as the RGB triplets have). Therefore, a more intuitive color space has been created, which considers the relative proportions of the X, Y and Z values rather than their unbounded versions — the xyY color space [19]. It is based on the assumption that color can be regarded as a quantity with two properties: *luminance* and *chromaticity*.

First, the following conversion from the X, Y and Z values to their bounded versions, also called *chromaticity coordinates*, is performed [11]:

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} \end{aligned} \tag{1.3}$$

Due to normalization ($x+y+z=1$), $z=1-x-y$, which means that we can drop the term z from the representation as it does not give any additional information about the current color. It also implies that we lost some information during the conversion — we cannot reconstruct the original XYZ triplet using only two values x and y and therefore cannot obtain the initial color. At least one of the original values is needed for this purpose — CIE [7] decided to use the Y component, as it already specifies the luminance.

Plotting the values of the x and y components creates a *chromaticity diagram*, shown in fig. 1.4. Each point of the curved boundary line (which is also called the *spectral locus*) corresponds to a XYZ value that is the result of a monochromatic radiation (i.e. a single-wavelength stimulus). All other chromaticities visible to the standard observer lie within a region bounded by the spectral locus.

L*a*b*

Although the xyY color space is already much more intuitive in terms of human color perception, the differences between individual triplets of the system are not perceptually uniform. The Hunter's Lab color space ref addressed this issue and was designed so that the distance between its two triplets characterized roughly how different they are in chromaticity and luminance. It is based on the Opponent color theory [20], which suggests that the cones in the human eye are linked together in opposing pairs and that the visual system records the *difference* between the stimulation of the pairs rather than the cones' individual responses.

As the Hunter's Lab color space does not achieve perfect uniform spacing of values, CIE $L^*a^*b^*$ color space (CIELAB) has been proposed in an attempt to improve some of its shortcomings and is now more widely used. However, neither of the systems are completely accurate in terms of perceptual uniformity [39].

The three opponent channels used to specify color in the CIE L*a*b* color space are defined as follows [14]:

- L^* — indicates lightness, i.e. the difference between *light* and *dark*. Its

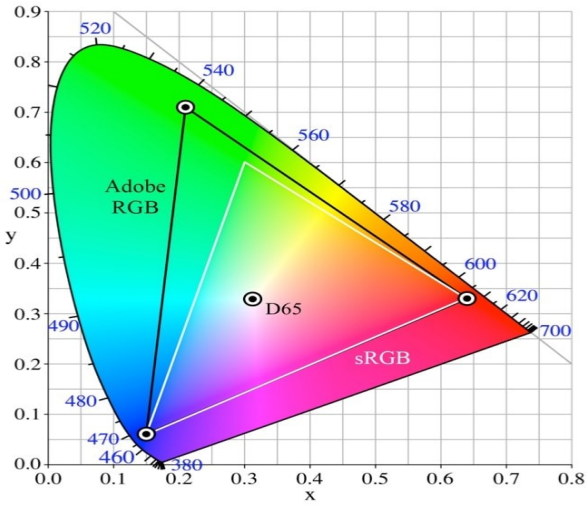


Figure 1.4: An illustrative comparison of the sRGB and Adobe RGB gamut in the chromaticity diagram based on images created by Choi et al. [5]

values range from 0 (yielding black color) to 100 (indicating diffuse white color) [14].

- *a** — defines the difference between *green* and *red*. Positives values of this component indicate the object's color to be more green, while negative values indicate red.

- *b** — defines the difference between *yellow* and *blue*. Positive values indicate the object to be more yellow, while negative values indicate blue.

Neither the range of the a* nor the b* component has any specific numerical limits [14].

The L*a*b* color space is a *reference system* — an abstract, non-intuitive space encompassing all the human perceptible colors. Due to its perceptual uniformity, it is used for color balance corrections by modifying the a* and b* components, and for lightness adjustments by modifying the L* component.

Another advantage and common use of the L*a*b* color space is for computing *color differences*. In 1976, CIE introduced the concept of *Delta E*, which is the measure of change in visual perception of two colors [13]. Denoted ΔE^*_{ab}, it is computed as an Euclidean distance between the two sample points, i.e.:

$$\Delta E_{76} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2}, \tag{1.4}$$

where (L^*_1, a^*_1, b^*_1) and (L^*_2, a^*_2, b^*_2) are the L*a*b* coordinates of the sample points.

However, the sensitivity of the human eye to color differences is not uniform. It is, for example, more sensitive to small color differences in dark blue colors than it is in e.g. light pastel colors. The ΔE^*_{ab} error does not take this ununiformity into account and therefore shows exaggerates differences in light colors while compressing perceptual distances between darker colors. To improve upon these shortcomings, other measuring techniques for computing Delta E have been proposed over the years, such as Delta94 and Delta2000.

Delta94 is computed by modifying the original L*a*b* values of both colors to compensate for perceptual distortions in the color space and computing Euclidean distance from the new modified values. Although the results match the human color difference perception more closely, the Delta94 error metric still lacks some accuracy in the blue-violet region [13].

Delta2000 attempts to remove these inaccuracies. Along with the corrections added to Delta94, Delta2000 adds overall five correctional factors to the original ΔE^*_{ab} — compensation factors for lightness, hue and chroma, compensation for neutral colors and, lastly, a hue rotation term for the problematic blue-violet regions.

From the listed Delta E equations, the Delta2000 error measurements are the most accurate in terms of human color difference perception [13] and, therefore, will also be used in the practical parts of this thesis. However, as the specifics of the Delta2000 equations are out of scope of this thesis, we refer the interested reader to the original article by Sharma et al. [30].

Other color spaces

In addition to the already named tristimulus color spaces, there exist many more used for various purposes. Following, we briefly overview some of them:

- *L*u*v** — Similarly to the CIELAB system, L*u*v* (or CIELUV) aims for perceptual uniformity. As a matter of fact, the $L*$ value is defined in the same manner as in the CIELAB system, while u and v values are evaluated by certain projections of the x and y coordinates of the chromaticity diagram. When comparing their Euclidean error measure, the most important distinction between the two spaces is that while CIELAB generally improves CIELUV in terms of color difference [22], CIELUV does not have as many inaccuracies in the dark regions [29]. Therefore, it is often recommended to use the CIELUV color space for characterization of color displays and CIELAB color space for the characterization of colored surfaces and dyes.

- *HSL* and *HSI* color spaces define color by its *hue*, *saturation* and *lightness* (or *intensity*). They are an alternative representation of the RGB color space and must therefore be defined purely with reference to an RGB space [15]. As their components correlate better with human perception of color than those of the RGB system, they are often used in image processing applications, e.g. for processes such as feature detection (edge detection [35], object recognition) or image segmentation (which can be performed solely with/by? the hue component) [15].

- *CMYK* model is a subtractive color model commonly used in color printing. It is based on RGB's complementary colors — *cyan*, *magenta* and *yellow* respectively. This means that assigning zero values to all components renders white light, and increasing the value of a component specifies how much of the respective color is *subtracted* from the white light. Although the theory states that maximizing CMY values should render perfect black, in reality, the printing inks are not 100% CMY and their combinations cannot produce rich black. For this purpose, a fourth component, *black* (K), is often added, giving rise to the CMYK model.

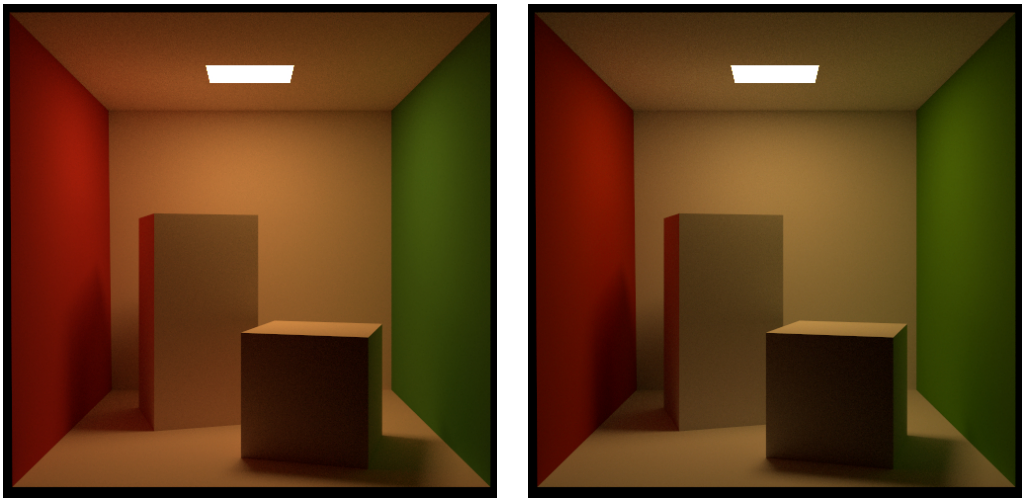


Figure 1.5: Comparison of an RGB-based rendering and spectral-based rendering as presented in the documentation of Mitsuba2 [36]. Left: Spectral reflectance data of all materials is first converted to RGB and the scene is then rendered in the RGB mode, producing an unnaturally saturated image. Right: Scene is rendered directly in the spectral mode, resulting in more realistic colors.

Other color spaces include Munsell color system, RAL, Natural Color System, Pantone Matching System, $\text{CIELCH}_{\text{ab}}$, $\text{CIELCH}_{\text{uv}}$, etc...

1.2.3 Color representation in rendering

Accurate color representation is the core of rendering softwares. Although most of today's renderers support multiple color spaces, we can still divide them into two main categories according to the space used during evaluation of light transfer equations — *tristimulus* and *spectral* renderers.

Tristimulus renderers are usually based on the RGB color space, although they often offer conversions to other tristimulus spaces. Due to the ease of use and simplicity of representation, RGB renderers are more common in commercial rendering software. They provide realistically looking images, often indistinguishable from a photograph, and are more robust, easy to implement and memory efficient.

However, light in real world does not travel as a tristimulus value, but rather as a distribution of wavelengths. As RGB renderers do not possess full-spectral information of materials and light in the scene, they cannot properly simulate the physical properties of the color during e.g. reflections or refractions when ray tracing.

Spectral rendering, on the other hand, uses full-spectral information of all materials and light in the scene during the whole rendering process. Obviously, before visualization occurs, spectral information must be converted into tristimulus (usually RGB) values, but this does not pose a problem as, at the moment of conversion, all the physically-based simulations have already taken place. Therefore, the rendered scene appears more realistic. We demonstrate this difference in 1.5, on a scene already rendered by Mitsuba2 [36].

In addition to rendering reflections and refractions more convincingly, another

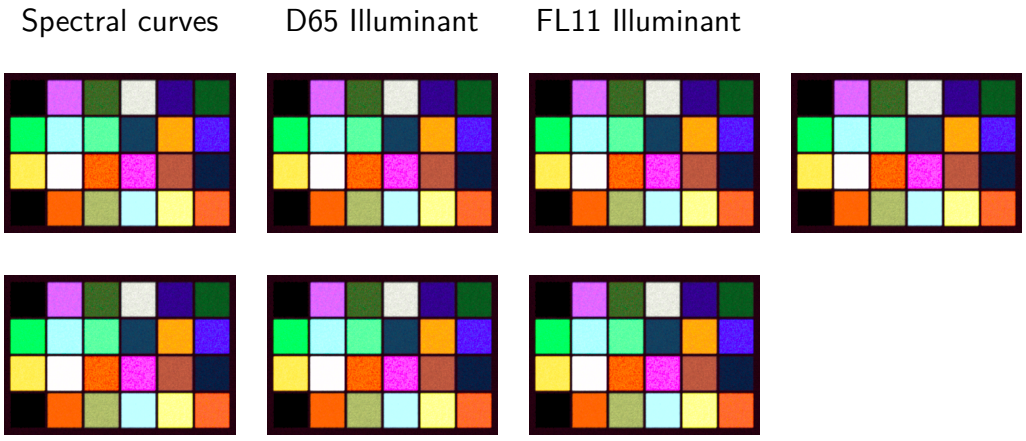


Figure 1.6: The effects of metamerism. Left: Two different spectral reflectance curves, both corresponding to RGB=(0,255,0). Middle: A box in a cornell box rendered with a A Cornell box rendered with

reason for using spectral rendering is its capability of simulating physically based phenomena that arises due to the interaction of color with light. Following, we overview some of the most common ones:

- *Metamerism*

  As already mentioned in section 1.2.2, the human tristimulus perception has a significantly lower domain than the (practically infinite) spectral domain. Therefore, two different spectra can trigger the same cone response in the human eye and appear to have the same color (and, subsequently, to have the same RGB values), giving rise to a phenomenon called *metamerism*. The two spectra evaluating to the same tristimulus values are called *metamers*.

  In real world, metamerism is often perceived when the lighting conditions under which we observe metamers change. An example of this can be seen in 1.6. Although we perceive the color of both objects to be the same under D65 illuminant (daylight), when illuminating the scene with x illuminant, the color changes.

  Obviously, this behavior is irreproducible by an RGB renderer, as it cannot replicate the behavior of spectral reflectance under an illuminant.

- *Fluorescence*

  By definition, fluorescence occurs when light from one excitation wavelength λ_0 is absorbed by an object and is almost immediately re-emitted at a different, usually longer, wavelength λ_1 [12]. Specifically interesting is the fact that the absorbed light can come from outside of the visible spectrum and be re-emitted inside it, which results in unrealistically bright material appearance, perceivable in real world when for example fish, corals, jellyfish or even minerals are illuminated by a UV light.

  RGB renderers attempt to fake this kind of behavior it through custom shaders [41]. As it produces satisfactory results and is immensely easier to implement than physical simulation, physically based fluorescence has re-

ceived small amount of work. Its support can be found in spectral renderers, added for example to ART by Mojzík [24].

- *Iridescence*

  Iridescence, or goniochromism, is a phenomenon occurring when certain surfaces change their color according to the current viewing angle. It arises when the object's physical structure causes interferences between light waves (e.g. inside extremely thin dielectric layers), yielding rich color variations [3]. It can be perceived in nature in certain plants, specific minerals, butterfly wings, peacock's feathers, snakes, but also in man-made products such as oil leaks, soap bubbles or car paints.

  Similarly to fluorescence, iridescent behavior can be "faked" in an RGB renderer [33]. However, research based on physical properties of iridescence has also been conducted. For further information about the current development, we refer the interested reader to the articles by Belcour and Barla [3], Sadeghi and Jensen [28], or Werner et al. [38].

- *Dispersion*

  When light travels from one medium to another (e.g. when light hits glass or water), its direction of travel is changed. This phenomenon is called *refraction* and is closely described by Snell's law, which specifies how the angle of refraction can be computed from the angle of incidence and the *refraction indices* of the two media [8]. However, the refraction index depends not only on the *type* of media, but also on the current *wavelength* [34] — which implies that the resulting direction of photons of different wavelengths might vary.

  Probably the most popular example of this phenomena is white light hitting a dispersive prism. Upon interaction, light is split into a spectrum, creating a "rainbow" effect.

  There have been multiple attempts to simulate physically-based dispersion. We refer the interested reader to articles by Sun et al. [34] or Wilkie et al. [40].

- *Polarization*

  Electromagnetic waves traveling through space are *transverse waves* — their oscillation is perpendicular to their path of propagation. By default, the directions of oscillations are arbitrary for each photon — this type of light is called an *unpolarized light*. Restrictions to the directions of oscillations (also called *polarization*) render *polarized light*. Such phenomenon usually occurs upon light's interaction with certain materials.

  The polarization process contributes to the overall color only in special cases (e.g. when using polarization filters) [41]. Therefore, it receives little attention in implementation of rendering softwares. However, for physical consistencies (and due to the possibility of special scenes) both ART [25] and Mitsuba [36] follow the direction of oscillation during the rendering process.

Other researched phenomena (some of it closely linked to the already mentioned ones) include *phosphorescence*, *bioluminescence*, *dichroism*, *opalescence*, *aventurescence* and many more.

2. Spectral Uplifting

Although spectral rendering has multiple advantages, many renderers do not consider them to compensate for the ease of use and memory efficiency of the RGB representation. Even the physically-based phenomena can be "faked" by a few simple tricks, and therefore, many conventional rendering systems are RGB-based. This implies that most textures and materials created for renderers are RGB-based.

However, spectral renderers are still used both in the research and in the commercial sphere (e.g. ART, Mitsuba, Manuka). In comparison to an RGB-based texture, however, creating spectral textures is much more complicated and usually requires a real-life model whose reflectance spectra can be measured with a spectrometer, which is, in many cases, virtually impossible.

The obvious solution is to convert the already existing RGB models to their spectral variants. We refer to this process as *spectral uplifting*, however, other sources also use the term *spectral upsampling* [16].

By being able to uplift RGB values, we could utilize the RGB textures and materials and therefore eliminate the need for repeatedly creating the textures from scratch by the tedious process of measuring concrete spectral values.

However, converting an RGB value into its respective spectrum poses multiple difficulties. As the relationship between the spectral and RGB domain is not bijective (specifically, infinitely many spectral distributions render the same RGB values), distinctive approaches to the conversion process may render different spectral distributions. Although all of them might be correct in terms of the resulting RGB value, it is possible that none of them would be identical to spectral distribution measured with a spectrometer. This does not cause a problem under standard illuminant with regard to which the RGB values were uplifted. However, as already mentioned in , changing the illuminant causes distinct spectra to behave differently, which consequently results in *metamerism*. Therefore, our uplifted spectra might behave differently than they would in real world. sem dam obrazok ak ho budem mat

We begin this chapter by reviewing the already existing approaches to spectral uplifting. We then talk about a new technique, *constrained spectral uplifting*, which provides means for solving the above mentioned problems.

2.1 Uplifting methods

Although there have been multiple attempts at spectral uplifting, not many meet all the conditions required for a successful and complete conversion (e.g. one method may output reflectance spectra with values outside the (0,1) range, other method might work only for saturated colors etc.).

We base most of this section on an article by Jung et al. [18], as it overviews multiple spectral uplifting techniques. It also proposes a new technique, which is considered to be the current state-of-the-art.

One of the first techniques was proposed by MacAdam [21]. The main goal of his research was to achieve the highest possible brightness for a given color saturation in printing. The uplifting process was only a byproduct of proof of limits

to the brightness of colors, created especially for representing the reflectance of the researched colors (i.e colors of maximum brightness for any given saturation). Although this method is not limited to a specific input, it produces spectra that are box shaped and only consist of rising and falling edges. This type of representation is unsuitable for colors usually found in nature, which tend to have smooth spectra.

Another technique was proposed by Smits [31]. In this case, the uplifting is based on a box basis split into 10 discrete bins, which are derived using an optimization algorithm that accounts for energy conversion and aims for overall smoothness of the spectra. This approach is practically implemented and widely used, as it provides satisfactory results in the sRGB gamut in [17]. However, the uplifted spectra can acquire values above 1 in some cases, which does not satisfy the (0,1) range criterion. Furthermore, conversion of an RGB value to spectra and then back produces slight differences, which are amplified in scenes with multiple reflections. Lastly, this approach becomes unstable when used with wider gamuts, as it was not designed for this purpose.

The goal of the method by Meng et al. [23] is wide-gamut uplifting. It also concentrates on optimizing the uplifting algorithm for spectral smoothness. However, it does not take energy conservation into account, which results in images with colors that have no physical counterpart (i.e. no real material could produce such colors). Meng et al. [23] tries to solve this by introducing a set of scaling methods for mapping the uplifted spectra to valid reflectances. These, however, fail if trying to uplift bright colors.

One of the most recent uplifting techniques has been proposed by Otsu et al. [26]. It is based on the observation that a typical measured reflectance spectrum can be represented with only a few principle components. The method uses clustered principal component analysis (PCA) and, unlike many other approaches, does not assume that spectra must necessarily be smooth. Such a simplification both eliminates the requirement of having a smoothness heuristic and enables the reconstructed spectra to match the actual measured spectra pretty well. This approach, however, also has its downsides. Firstly, the method does not satisfy the (0,1) range criterion, which, if clamped, results in color reproduction errors. Moreover, since there is no interpolation across clusters, similar RGB values might produce very different spectra, which might lead to discontinuities in rendering. However, in multiple cases, this method has been shown to outperform all of the already mentioned ones [17].

A large part of this thesis is based on the work by Jakob and Hanika [17]. We will therefore describe their approach in more detail.

In their article, Jakob and Hanika [17] describe a parametric function space for efficient representation of spectral reflectance curves. They also show how to utilize such a space for the purposes of spectral uplifting.

The main goal was to create a spectral representation that would be both energy-conserving and would have a successful round-trip, i.e. the difference between the original RGB and the RGB obtained by conversion to spectra and back would be as small as possible. A simple analytical model has been created based on the equation specifying the DeltaE error created by round-trips. Spectra in accordance with this model are represented as following:

$$f(\lambda) = S(c_0\lambda^2 + c_1\lambda + c_2), \tag{2.1}$$

Algorithm 1 Spectral uplifting by Jakob and Hanika [17]

create $RGBCube$ with empty RGB:spectra mappings
$unfittedPoints \leftarrow$ a list of all points in $RGBCube$
$centerPoint \leftarrow$ index of the middle of $RGBCube$
▷ $RGBCube[centerPoint].rgb \simeq (0.5, 0.5, 0.5)$
$centerPoint.coefficients \leftarrow (0, 0, 0)$
▷ "guess" the coefficients at $centerPoint$
run the CERES optimizer for $RGBCube[centerPoint]$
remove $RGBCube[centerPoint]$ from $unfittedPoints$
while $unfittedPoints$ is not empty **do**
    for all $point \in unfittedPoints$ **do**
        if $point$ has a neighbor v with defined coefficients **then**
            $point.coefficients \leftarrow v.coefficients$
            run the CERES optimizer for $point$
            if optimization was successfull **then**
                remove $point$ from $unfittedPoints$

where $f(\lambda)$ is the resulting spectrum, S is a simple sigmoid function and c_i are coefficients of a second-order polynomial. Therefore, all spectra in this space are represented by three parameters.

In addition to energy conservation, the resulting spectra do not violate the (0,1) range constraint. They are extremely smooth and simple, which corresponds to many of the spectra typically found in nature. Another great advantage is memory efficiency, which requires only three values. However, representing spectra as specified in 2.1 has its drawback. For example, there is currently no straightforward, well-defined computation of the RGB $\rightarrow$ spectrum conversion in such a domain. To uplift an RGB value, one must keep "guessing" the coefficients until the spectrum evaluates to the desired RGB.

In this specific implementation, the "guessing" process is performed mostly by the CERES solver [1] (note:should I find out how this works?). It requires only an initial guess and a metric according to which it improves the guess (i.e. the DeltaE error originating from round-trips) and requires only a few iterations to converge to 0.

The uplifting process itself works by pre-computing RGB:spectra mappings and storing them in a texture. During rendering, only the required spectra are looked up in the texture.

In algorithm 1, we describe the pseudo-algorithm of the uplifting process, which is very similar to the one used in our implementation.

This means that the cube is optimized from the middle.

Obviously, not all rgb values can be uplifted as such. However, jakob says that we need only 64 dimensions and that, as the sigmoid curves are very similar, the middle values can be interpolated (show algorithm).

Another drawback is the limitation of such a spectral representation to only smooth spectra with no sharp edges. We discuss this specific issue in detail in ref, where we provide arguments as to why we do not use the sigmoid representation later in our thesis.

This technique, in comparison to other existing techniques, has definitely

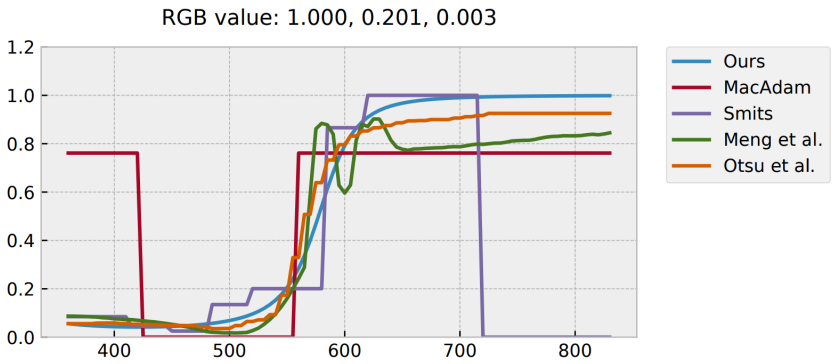


Figure 2.1: Comparison of spectral uplifting techniques as shown by Jakob and Hanika [17]. All spectra were created by uplifting the (1, 0.201, 0.003) RGB value and the results were plotted according to the corresponding techniques. The "Ours" approach, in this case, refers to the approach by [17].

the best error (maybe show image). the results are obviously metameric, show pictures

alisa bases her approach on this thing

2.2 Constrained spectral uplifting

Achieving identity of our uplifted spectra to the real-world spectra is, obviously, impossible. However, uplifting many RGB-based models does not require us to be able to uplift the whole RGB gamut, but only the color spectrum used for the creation of said models. As it is pretty common for the artists in the VFX modeling industry to use specific color atlases when designing textures and materials, the ability to *constrain* the uplifting system with these base colors would be extremely useful in such cases.

In other words, the user would define specific RGB:spectra mappings which would later be used in order to uplift certain RGB triplets. RGB values that would not have a pre-defined mapping would be uplifted by altering the curves of their neighbors that already have a mapping.

We call this process *constrained spectral uplifting*. Although it does not provide as much freedom as other spectral uplifting approaches, it works for our benefit in terms that the results would not be a subject to high metamerism, which is, after all, the goal of this thesis.

Following, we provide a brief overview of the algorithm used for constrained spectral uplifting:

S.1 *RGB cube initialization* — create an RGB cube that will be used as a structure for storing the RGB:spectra mappings. For every RGB triplet in the cube, initialize its respective spectrum to empty.

S.2 *seeding the cube* — store the input spectra at the lattice points with their respective RGB values.

S.3 *sampling of the spectra* — for each spectrum, sample it so it can be represented (and also reconstructed) with a small, constant number of parameters. Save the parameters instead of the spectrum.

S.4 *spectra approximation at other lattice points* — use the already existing parameters in the cube to reconstruct RGB values at lattice points that do not have a defined spectrum yet. For this purpose, an approximation algorithm must be used.

In addition to memory efficiency, sampling spectra is essential for the approximation process mentioned in S.4. We explain the reasoning behind this in ref3. For now, we assume that the spectra must be stored as efficiently as possible, with constant number of parameters.

we base our process on jakob and hanika, but we constrain the input and we also choose a different sampling method.

2.2.1 Spectral sampling

In real world, the spectral reflectance curve is a continuous function obtaining values in range (0,1). As it does not have to be necessarily smooth storing any kind of function in any kind of digital system requires its discretization.

Also, probably here is a good time to mention that we are focusing on the reflectance spectra, however we will also talk about emission spectra in this section solely for research purposes

In many spectral renderers,

Available methods

Trigonometric moment method

A review of the moment method (basically just a review of the paper)

3. Implementation

Mention that we use the trigonometric moment method to extend Borgtool, which is currently used for spectral uplifting. In addition to utilizing the moment method, we are also constraining the input - explain what the constraining means in two sentences.

3.1 Borgtool

Mention how it works, probably a few screenshots, mention the sigmoid method it was already using (reference the available methods subsection in Spectral uplifting). Mention that it was seeding the cube from the middle. Explain the cube "expansion" and that for all the other lattice points, a prior is used from their already fitted neighbor.

Also explain the threshold value - it is really important to set it properly and that it affects performance.

3.1.1 Optimizer

Maybe this doesn't need to be a subsection? Mention how it works, link the ceres optimizer and explain that until now we were using it to fit 3 coefficients.

3.1.2 Choice of parameters

Explain that because of the optimizer, we are actually using 9 moments. Link this to the spectral uplifting section. Emphasize that using complex moments isn't realistic and we need mirroring. Also, the default threshold is 0.1, going below is quite unrealistic - possible but would take a lot of time.

3.2 Cube constraints

We added the option to constraint the input. Explain the color atlases that might be provided, how the cube is seeded with them.

If not specified, the cube is seeded from the middle (use Munsell N5 that was pre-computed). Also explain the relationship of the size of the atlas with cube dimension.

3.3 Filling the cube

Just mention that we are basically doing what has already been done, however the cube is now growing in many directions (multi-threaded) and not only from the middle - would be nice to add progress images. Also, it might be nice to add progress images when seeding only from the middle.

Also emphasize that the optimizer is currently not working ideally and that there are a few issues - probably encounters some unresolved division by zero? - therefore sometimes, there are unexplained gaps. Explain that we are solving

this by trying out various different prior coefficients and that this is definitely an "ugly behavior".

4. Results

4.1 Number of moments

4.1.1 Evaluation of parameters

Add the results from the tests I ran back in April - which combination of number of moments, Warp/NonWarp and Mirror/NonMirror techniques is the most optimal. Also mention that we do not want to use complex moments as it doubles the space needed which will be unusable for the optimizer in the Implementation section.

4.1.2 Reconstruction results

Try the reconstruction on various spectrum values, also add the respective RGB values.

It might be nice to also add the ideal coefficients that we got from the borgtool by optimizing and comparing them to the coefficient that were originally computed (these are different mainly due to rounding errors during the algorithm).

4.2 Spectral uplifting

Not really sure about the subsections here (I still don't understand the texture options).

The idea that currently comes to mind: For each parameter setting, create a record containing the following:

- performance (time)
- average accuracy (targetRGB-latticeRGB)
- maxDifference (the important quantity)
- maybe other stuff (e.g. number of optimizer rounds)

Also mention that it is multi-threaded and performance is not really a priority - the cube has to be created only once and then can be reused as much as the artists need.

Compare these records with regard to:

4.2.1 Choice of threshold

E.g. which threshold is still good enough for the moment method, the performance etc.

4.2.2 Sigmoid vs Moment method

Compare these two approaches (the texture option which I'm not sure about), choose the same threshold for both of them, maybe do this for more thresholds?

Moments	Methods							
	M&W		M&nonW		nMW		nMnW	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
1	35.87	114.04	36.2	113.86	35.92	113.97	36.2	113.86
2	21.3	91.19	26.4	99.44	6.54	42.93	13.24	60.23
3	1.65	15.09	15.43	68.06	2.13	17.51	3.7	17.5
4	0.86	5.6	9.93	55.67	1.13	6.87	0.95	5.3
5	0.54	2.98	4.19	23.53	0.97	4.35	0.48	1.96
6	0.34	2.22	1.19	5.68	0.75	3.37	0.31	0.97
7	0.29	2.17	0.77	2.38	0.55	2.46	0.22	0.99
8	0.28	2.03	0.77	1.86	0.51	1.83	0.22	1.0
9	0.26	1.95	0.62	1.43	0.47	1.67	0.21	1.05
10	0.24	1.81	0.4	1.28	0.43	1.68	0.22	1.04
11	0.24	1.75	0.24	0.96	0.42	1.69	0.22	1.02
12	0.23	1.7	0.22	1.11	0.42	1.69	0.21	1.03
13	0.22	1.64	0.24	1.05	0.4	1.53	0.21	1.05
14	0.22	1.6	0.24	1.0	0.38	1.4	0.22	1.04
15	0.22	1.54	0.22	0.99	0.38	1.37	0.22	1.05
16	0.22	1.46	0.21	1.04	0.34	2.4	0.22	1.05

Table 4.1: this is 1976, not 2000 due to discontinuities when using gradients

4.2.3 Constrained input

Compare maybe the performance of different atlases, or maybe atlas vs. seeding from the middle?

The Results chapter is just a sketch and it may be completely changed, depending on the results and which of them will be interesting.

Conclusion

Bibliography

[1] Sameer Agarwal, Keir Mierle, et al. Ceres solver. 2012.

[2] Anikethan Bekal, Ajit M Hebbale, and MS Srinath. Review on material processing through microwave energy. In *IOP Conference Series: Materials Science and Engineering*, 2018.

[3] Laurent Belcour and Pascal Barla. A practical extension to microfacet theory for the modeling of varying iridescence. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

[4] Arthur D Broadbent. A critical review of the development of the cie1931 rgb color-matching functions. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 29(4):267–272, 2004.

[5] Kyungah Choi, Jeongmin Lee, and Hyeon-Jeong Suk. Context-based presets for lighting setup in residential space. *Applied Ergonomics*, 52:222–231, 01 2016. doi: 10.1016/j.apergo.2015.07.023.

[6] Asim Kumar Roy Choudhury. *Principles of colour and appearance measurement: Object appearance, colour perception and instrumental measurement*. Elsevier, 2014.

[7] CIE. Commission internationale de l'éclairage, 1913. URL `http://cie.co.at/`.

[8] D Drosdoff and A Widom. Snell's law from an elementary particle viewpoint. *American journal of physics*, 73(10):973–975, 2005.

[9] Hugh S Fairman, Michael H Brill, and Henry Hemmendinger. How the cie 1931 color-matching functions were derived from wright-guild data. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 22(1):11–23, 1997.

[10] Lori Gardi. Planck's constant and the nature of light, 05 2018.

[11] TM Goodman. International standards for colour. In *Colour Design*, pages 177–218. Elsevier, 2012.

[12] George G Guilbault. *Practical fluorescence*. CRC Press, 2020.

[13] Martin Habekost. Which color differencing equation should be used. *International Circular of Graphic Education and Research*, 6:20–33, 2013.

[14] CIE HunterLab. L* a* b* color scale. *Applications note, Virginia, USA*, 1996.

[15] Noor A Ibraheem, Mokhtar M Hasan, Rafiqul Z Khan, and Pramod K Mishra. Understanding color models: a review. *ARPN Journal of science and technology*, 2(3):265–275, 2012.

[16] Wenzel Jakob and Johannes Hanika. A low-dimensional function space for efficient spectral upsampling. In *Computer Graphics Forum*, volume 38, pages 147–155. Wiley Online Library, 2019.

[17] Wenzel Jakob and Johannes Hanika. A low-dimensional function space for efficient spectral upsampling. In *Computer Graphics Forum*, volume 38, pages 147–155. Wiley Online Library, 2019.

[18] Alisa Jung, Alexander Wilkie, Johannes Hanika, Wenzel Jakob, and Carsten Dachsbacher. Wide gamut spectral upsampling with fluorescence. In *Computer Graphics Forum*, volume 38, pages 87–96. Wiley Online Library, 2019.

[19] Douglas A Kerr. The cie xyz and xyy color spaces. *Colorimetry*, 1(1):1–16, 2010.

[20] David H Krantz. Color measurement and color theory: Ii. opponent-colors theory. *Journal of Mathematical Psychology*, 12(3):304–327, 1975.

[21] David L MacAdam. The theory of the maximum visual efficiency of colored materials. *JOSA*, 25(8):249–252, 1935.

[22] Manuel Melgosa. Testing cielab-based color-difference formulas. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 25(1):49–55, 2000.

[23] Johannes Meng, Florian Simon, Johannes Hanika, and Carsten Dachsbacher. Physically meaningful rendering using tristimulus colours. In *Computer Graphics Forum*, volume 34, pages 31–40. Wiley Online Library, 2015.

[24] Michal Mojzík. Fluorescence computations in a hero wavelength renderer. 2018.

[25] Computer Graphics Group of Charles University in Prague. Art. URL `https://cgg.mff.cuni.cz/ART/gallery/`.

[26] Hisanari Otsu, Masafumi Yamamoto, and Toshiya Hachisuka. Reproducing spectral reflectances from tristimulus colours. In *Computer Graphics Forum*, volume 37, pages 370–381. Wiley Online Library, 2018.

[27] Dale Purves, G Augustine, D Fitzpatrick, L Katz, A LaMantia, J McNamara, and S Williams. Neuroscience 2nd edition. sunderland (ma) sinauer associates, 2001.

[28] Iman Sadeghi and HW Jensen. A physically based anisotropic iridescence model for rendering morpho butterflies photo-realistically. *Proc. of Iridescence: More Than Meets the Eye (Tempe, Arizona, 2008)*, page 38, 2008.

[29] Gaurav Sharma and Carlos Eduardo Rodríguez-Pardo. The dark side of cielab. In *Color Imaging XVII: Displaying, Processing, Hardcopy, and Applications*, volume 8292, page 82920D. International Society for Optics and Photonics, 2012.

[30] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 30(1):21–30, 2005.

[31] Brian Smits. An rgb-to-spectrum conversion for reflectances. *Journal of Graphics Tools*, 4(4):11–22, 1999.

[32] Andrew Stockman and Lindsay T Sharpe. Cone spectral sensitivities and color matching. *Color vision: From genes to perception*, pages 53–88, 1999.

[33] Yinlong Sun. Rendering biological iridescences with rgb-based renderers. *ACM Transactions on Graphics (TOG)*, 25(1):100–129, 2006.

[34] Yinlong Sun, F David Fracchia, and Mark S Drew. Rendering light dispersion with a composite spectral model. *Diamond*, 2(37.17):0–044, 2000.

[35] Arthur Robert Weeks, Carlos E Felix, and Harley R Myler. Edge detection of color images using the hsl color space. In *Nonlinear Image Processing VI*, volume 2424, pages 291–301. International Society for Optics and Photonics, 1995.

[36] Guillaume Loubet Sébastien Speierer Benoît Ruiz Delio Vicini Wenzel Jakob, Merlin Nimier-David and Tizian Zeltner. Mitsuba2. URL `https://mitsuba2.readthedocs.io/en/latest/src/getting_started/variants.html`.

[37] John Werner. Human colour vision: 1. colour mixture and retino-geniculate processing. 10 2001. doi: 10.1142/9789812811899_0003.

[38] Sebastian Werner, Zdravko Velinov, Wenzel Jakob, and Matthias B Hullin. Scratch iridescence: Wave-optical rendering of diffractive surface structure. *ACM Transactions on Graphics (TOG)*, 36(6):1–14, 2017.

[39] N Whetzel. Measuring color using hunter l, a, b versus cie 1976 l* a* b*. *Application notes). Retrieved from Hunterlab website: https://support. hunterlab. com/hc/enus/articles/204137825-Measuring-Color-using-Hunter-Lab-versus-CIE-1976-Lab-AN-1005b*, 2016.

[40] Alexander Wilkie, Robert F Tobler, and Werner Purgathofer. Raytracing of dispersion effects in transparent materials. 2000.

[41] Alexander Wilkie, Robert F Tobler, and Werner Purgathofer. Combined rendering of polarization and fluorescence effects. In *Rendering Techniques 2001*, pages 197–204. Springer, 2001.