Practical Aspects of Spectral Data in Digital Content Production

Course Notes Siggraph 2022

June 6th, 2022

Andrea Weidlich (Organizer)
Nvidia

Chloe LeGendre
Netflix

Carlos Aliaga
Meta Reality Labs

Christophe Hery
Meta Reality Labs

Jean-Marie Aubry
Wētā Digital

Jiří Vorba
Wētā Digital

Daniele Siragusano
FilmLight

Richard Kirk
FilmLight

SIGGRAPH '22 Courses, August 07-11, 2022, Vancouver, BC, Canada ACM 978-1-4503-9362-1/22/08.
10.1145/3532720.3535632.

Abstract

Compared to path tracing, spectral rendering is still often considered to be a niche application used mainly to produce optical wave effects like dispersion or diffraction. And while over the last years more and more people started exploring the potential of spectral image synthesis, it is still widely assumed to be only of importance in high-quality offline applications associated with long render times and high visual fidelity.

While it is certainly true that describing light interactions in a spectral way is a necessity for predictive rendering, its true potential goes far beyond that. Used correctly, not only will it guarantee colour fidelity, but it will also simplify workflows for all sorts of applications.

Wētā Digital's renderer Manuka showed that there is a place for a spectral renderer in a production environment and how workflows can be simplified if the whole pipeline adapts. Picking up from the course last year, we want to continue the discussion we started as we firmly believe that spectral data is the future in content production. The authors feel enthusiastic about more people being aware of the advantages that spectral rendering and spectral workflows bring and share the knowledge we gained over many years. The novel workflows emerged during the adaptation of spectral techniques at a number of large companies are introduced to a wide audience including technical directors, artists and researchers. However, while last year's course concentrated primarily on the algorithmic sides of spectral image synthesis, this year we want to focus on the practical aspects.

We will draw examples from virtual production, digital humans over spectral noise reduction to image grading, therefore showing the usage of spectral data enhancing each and every single part of the image pipeline.

Syllabus

Fifty Shades of Pink (and Why None of Them is in Your Rainbow) (Andrea Weidlich, 25 minutes)

Spectral rendering is still widely considered to be a) slow and b) difficult to implement. Neither is the case. This first section of this course will give an introduction into spectral image synthesis and how spectral workflows can be integrated into a modern production pipeline. We will discusses open research questions and identify the problems which needs to be solved both in realtime as well as offline rendering to make it as practical as conventional tristimulus RGB rendering.

Multispectral Lighting Reproduction for Virtual Production (Chloe LeGendre, 20 minutes)

This talk will present techniques for optimizing the color rendition properties of LED-based lighting reproduction systems. Starting from optimization for individual light sources, these techniques will then be extended to include omnidirectional computational illumination systems, and finally to the LED volumes used in modern virtual production. Although the intuition of thinking of the world as a spectral renderer will be helpful, the proposed techniques require only tristimulus photography rather than spectral measurements.

Estimation of Spectral Biophysical Skin Properties from Captured RGB Albedo for Digital Humans (Carlos Aliaga and Christophe Hery, 30 minutes)

The reliable rendering of digital humans is an important and ongoing research area in content production. In this section we share our experience of how to reconstruct and manipulate the spectral properties of human skin from simple RGB albedo captures. We will show how we could leverage Monte Carlo light simulation over an accurate biophysical human skin layering model parameterised by its most important components, thereby covering a plausible range of human skin colours. The practical complexity of the model allows us to learn the inverse mapping from any albedo to its most probable associated spectral skin properties which will help us to create human avatars in the future.

Sampling and Re-Sampling in Spectral Rendering (Jean-Marie Aubry and Jiří Vorba, 25 minutes)

The wavelength dimension of integration in a spectral renderer necessarily increases the noise of any Monte Carlo technique; it also complicates the implementation of common noise-reducing sampling techniques. In this course, we will show how the non-uniform extension of the "hero wavelength" method interacts with multiple importance sampling and also resampling techniques. This extension is a requirement for spectral guiding which is a technique to optimize importance sampling of the multiscalar wavelength, based on initial estimation of the spectral radiance received at a camera pixel (for path tracing) or of the spectral importance received at a light (for light tracing). It is implemented in Manuka and currently being tested for production.

In the second part, we explore application of ReSTIR (Reservoir-based Spatio-Temporal Importance Resampling) for sampling of direct illumination within spectral renderer. This requires extending the state-of-the-art resampling framework to account for sampling over wavelengths so that previous samples are reused correctly.

Grading Movies using a Smooth Spectral Intermediate (Daniele Siragusano and Richard Kirk, 30 minutes)

Traditional motion picture cameras output RGB pixel values. This makes it impossible to determine the spectral properties of a reflecting surface with metamerism: two colour patches may give the same linear RGB under illuminant A but different RGB under illuminant B. We cannot guess the actual spectrum from a single set of RGB values, but there will be a unique smoothest spectrum that fits the RGB data.

We will explore how we can use this for spectral processing in our grading workflows, knowing the process yields smooth and continuous RGB transforms. We can combine spectral measurements with conventional three-channel workflows, to change illuminants or model optical filters in a physically plausible way.

Organizer

Andrea Weidlich, Nvidia

Andrea Weidlich is a Principal Researcher at Nvidia in the Realtime Rendering Research department. Before joining Nvidia, she worked for Wētā Digital where she designed the material system attached to Wētā's proprietary physically-based renderer, Manuka. Her main research areas are appearance modelling and material prototyping. Andrea holds a Master of Arts in Applied Media from University of Applied Arts, Vienna and a Ph.D. in Computer Science from Vienna University of Technology.

Presenters

Chloe LeGendre, Netflix

Chloe LeGendre is a Senior Research Scientist at Netflix, working in a computer graphics group focused on research at the intersection of machine learning and filmmaking. She earned her Ph.D. in Computer Science at the University of Southern California's Institute for Creative Technologies (USC ICT) in 2019, advised by Paul Debevec. She previously worked as a Senior Software Engineer at Google Research focusing on computational photography and, before her Ph.D., as a Senior Scientist in imaging and augmented reality for the digital incubator division of L'Oréal Research and Innovation.

Carlos Aliaga, Meta Reality Labs Research

Carlos has been a researcher at Meta since 2019, working on physically based inverse and forward rendering of virtual humans. Prior to that, he was a Lead Scientist of SEDDI Labs, a start up based in Madrid that targets the digitalization of fabrics, devising ways of creating virtual replicas of cloth from captures. Carlos got his Ph.D. in the Graphics & Imaging Lab (Zaragoza, Spain), under the supervision of Prof. Dr. Diego Gutiérrez and Dr. Adrian Jarabo. His work focused on the perception of virtual humans in order to explore physically based appearance models of the complex materials involved, like skin or cloth. During his Ph.D., he worked at Walt Disney Animation Studios (2014, 2016) and Technicolor (2016) as an intern, and at Porto's Science Faculty as a research engineer (2012).

Christophe Hery, Meta Reality Labs Research

Christophe Hery joined Reality Labs Research in 2019. Previously, he worked at Pixar, where he held the position of Senior Scientist. After writing new lighting models and rendering methods for Monsters University and The Blue Umbrella, Christophe continues heading the light transport research group in the studio. Christophe's latest work includes Finding Dory, Coco and Toy Story 4. An alumnus of Industrial Light & Magic, Christophe previously served as a research and development lead, supporting the facility's shaders and providing rendering guidance. He was first hired by ILM in 1993 as a senior technical director. During his career at ILM, he received two Technical Achievement Awards from the Academy of Motion Pictures Arts and Sciences.

Jean-Marie Aubry, Wētā Digital

Jean-Marie Aubry is a Senior Researcher at Wētā Digital overseeing several aspects of Manuka, the company's home-grown renderer. Jean-Marie evolved from theoretical mathematics to computer graphics, by the way of physically-based simulation. Before working on Manuka Jean-Marie was a key developer of Wētā's simulation software. His current research interests are in sampling optimization for Monte Carlo rendering.

Jiří Vorba, Wētā Digital

Jiří is a senior researcher and rendering software developer at Wētā Digital. He has received his Ph.D. from Charles University in Prague in 2017. From September 2012 to January 2013, he undertook an internship at Max Planck Institute for Informatics in Saarbrücken under the supervision of Dr. Tobias Ritschel. In 2014, as part of his internship with Wētā Digital, he implemented research results on path guiding achieved during his PhD. into Manuka renderer.

Daniele Siragusano, FilmLight

After his Master of Arts in Electronic Media in 2010 Daniele spent almost five years at CinePostproduction in Munich. As Head of Technology, he was in charge of digital image processing, DI colour management and post-production workflows. In 2014, Daniele joined FilmLight as image engineer, helping Post-production companies worldwide to achieve the best productivity, accuracy and creativity. He is also a principal engineer for developing HDR grading and colour management tools inside Baselight, FilmLight's colour corrector.

Richard Kirk, Filmlight

Richard Kirk developed a colour calibration system for proofing colour pages on film recorders in 1985. Since then, he has worked continuously on colour and imaging technology, with Crosfield Electronics for the printing industry; with Canon for colour photocopiers and desktop devices for Canon; with Framestore-CFC for film post-production, and finally for FilmLight, where he developed the Truelight colour calibration system, the standard for the film previewing and visualization. He holds over 30 patents on colour and imaging techniques, and an AMPAS Scientific and Technical Award (Oscar) for the Truelight system.

Contents

1

Fifty Shades of Pink (and Why None Of Them is in Your Rainbow)

ANDREA WEIDLICH, *Nvidia*

1.1 Introduction

In recent years, spectral rendering and spectral workflows have become more and more common in content creation [Fascione et al. 2018]. There are two main reasons why. Firstly, light transport can be simulated more accurately by tracing wavelengths instead of RGB, which allows for more fidelity when simulating non-linear phenomena like absorption, multi-bounce light transport like scattering or wave-optics phenomena like dispersion or diffraction. Secondly, and more importantly, spectral image synthesis allows reliable colour workflows that is invaluable in content creation. Light and material properties as well as camera responses can be measured. Transformations between spaces can be done easily since the data is in an absolute format and not a value relative to a colour space and a white point. Moreover, only spectral data can capture metamerism, i.e. the effect that one colour can be produced by an infinite number of spectra. Given the small overall performance overhead, the advantages outweigh the disadvantages by far.

This first part of this course will give a general overview of spectral image synthesis as a technology in content creation and will discuss recent research directions important for production pipelines. While some of the content of this chapter has been presented in last year's course, we wanted to provide a summary in a single place to give an overview of spectral image synthesis for members of the audience who are not experts on the subject yet. For an in-depth discussion of the various topics, we would direct the reader to Weidlich et al. [2021].

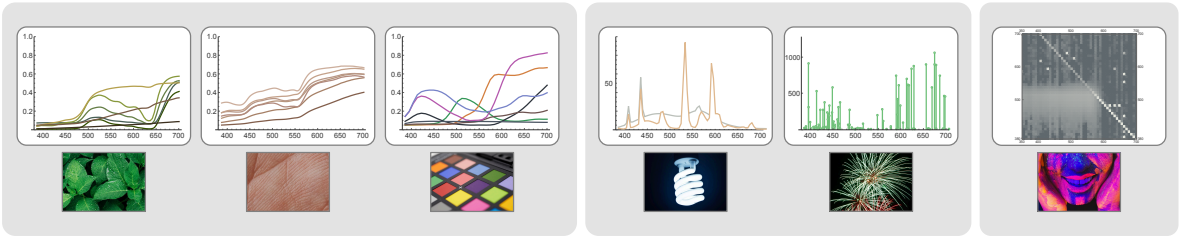
1.2 Spectral Rendering Pipelines

A spectral rendering framework is not that different compared to a traditional RGB system. In a very coarse classification, we can differ between three stages of a renderer: Input, light transport and output. In the following section we will discuss in more detail the influence spectral data has on the respective parts of the pipeline and highlight the peculiarities of a spectral system.

1.2.1 Input Data

1.2.1.1 Spectral Data Types and Storage

The first step in converting a tristimulus renderer to a spectral one is to decide on a storage format to represent colour data. Spectral data can roughly be divided into two classes, smooth spectra and spectra with discontinuities. Smooth spectra can be found in many natural materials whereas spectra of artificial materials, and especially light sources, will often have strong discontinuities. Somewhat orthogonal is the storage of fluorescent data. The data itself is more or less smooth, but a 2D representation is needed where both absorption and re-emission have to be encoded. Examples of different spectra can be seen in figure 1.1.



(a) Spectra of various plants, human skin and Macbeth patches

(b) Illuminant F7 and F12 (left) and Barium (right)

(c) Re-radiation matrix of red paint

Figure 1.1: Different classes of spectra. Natural spectra (a) tend to be smooth whereas artificial spectra often exhibit strong discontinuities (b). Fluorescent responses need to be able to encode both absorption and re-emission (c).

When confronted with the question of how to store a spectrum, several approaches exist. Ideally, a method should be light in memory, expressive and allow fast light transport operations like the multiplication of two spectra.

- **Point-based samples.** The spectral curve is uniformly sampled, and the curve is reconstructed by interpolating between points. Memory footprint and accuracy will depend on the number of sample points.
- **Basis functions.** Data is approximated with several basis functions like e.g. a Fourier basis [Peercy 1993]. Basis functions will be lighter in memory than point-based samples, but they will lack accuracy when it comes to peaky spectra and are more expensive during evaluation.
- **Hybrid.** Hybrid storage solutions split spectral into parts that are stored with different techniques. Sun et al. [1999] split spectra into smooth and peaky segments whereas Hua et al. [2021] use GMMs to represent the absorption of fluorescent data, reducing the 2D re-radiation matrix [Donaldson 1954] commonly used for fluorescent data to a 1D spectrum and a GMM part.
- **Neural Networks.** Little work has been done so far [Huang et al. 2022], especially in the computer graphics community. While the technique seems promising as a storage or input format, it is yet unclear if it is feasible to use it as an internal representation within a rendering framework and more research needs to be done.

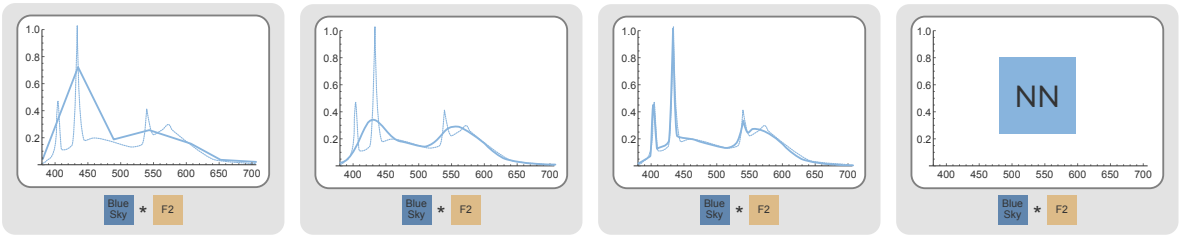


Figure 1.2: Different methods to store spectra. From left to right: Point-based sampling, basis functions, hybrid of point samples and basis functions and neural encoding. In the example, a Macbeth Blue Sky patch is illuminated with a CIE F2 light.

An illustration of different storage methods can be seen in figure 1.2. While all approaches have their advantages and disadvantages, the method most commonly found in modern spectral frameworks is the point-based representation. The number of samples needed to faithfully reconstruct a spectrum will vary from spectrum to spectrum; for smooth curves, 10 or 12 are often sufficient while spectra from e.g. light sources will need a lot more. It should be noted that if a curve is severely under-sampled, best practice is not to sample the curve directly but to use e.g. a Riemann sum which will preserve the area of the curve. What makes this data type so attractive despite its memory overhead and inaccuracies is that interpolation and filtering are straightforward and fast, and light transport operations are simple point-wise operations. Wētā's in-house renderer Manuka for example saves heterogeneous volume data internally as 10 equi-spaced samples [Weidlich et al. 2021].

1.2.1.2 Spectral Data Injection

The crucial next step is to get content from the outside into the framework. If the data is already spectral, it can be used by the renderer directly. But traditionally, most content will be produced in RGB. In this case, RGB data needs to be converted to a spectral representation before it can be used.

- **Constant Data.** The simplest way to inject spectral data is to pass it to the renderer in a tabulated form. Since the spectrum does not vary, we do not have to be concerned about memory. However, the usefulness of this data type is limited because the colour will be uniform. Measured data will most often be used in a constant fashion.
- **Spectral Textures.** A more powerful and content-creation friendlier way of passing data is to store it in a texture format. This comes, however, with a heavy impact on memory and performance since the data need to be either stored on the vertex or the texture needs to be loaded and interpolated per hit.
- **Spectral Uplifting.** The most common way of passing spectral data into a production framework is spectral uplifting. Uplifting describes a process where a spectral response is hallucinated from an RGB input. The advantage of uplifting is that no or little memory overhead is created compared to spectral textures. However, the resulting spectra are almost always a metameric spectrum of the desired response.

Both constant data and spectral textures have their place in content creation. Constant data is most useful for reference objects like grey spheres or Macbeth colour checkers, but

also for light sources and gels which usually won't vary much in colour on a per point basis. Another use-case is homogeneous volumes. And while it is possible to use spectral textures as material parameters, their most useful application is spectral IBLs. However, data is hard to come by and acquisition is time-consuming. Chapter 2 will discuss capturing of illumination for virtual production. Non-proprietary texture formats are rare; lone exceptions are an OpenEXR extension from Fichet et al. [2021] or Mirhashemi [2018] who proposed a multi-stack TIFF format. Proprietary formats like the HSZ format from [Scyllarus 2016] or the one from AxF from [X-Rite 2022] exist, although the latter one is not a texture format but a complete SVBSDF. Furthermore, tools which would allow manipulation of the data, e.g. change in saturation or contrast which is essential in content creation, are not available, which limits their usefulness.

One big disadvantage of using spectral data directly is that tools and pipelines are generally written with RGB data in mind and that the question of how to manipulate spectra is still open. So instead of passing spectral data from the outside, the more practical approach is to rely on RGB data and uplift on the fly. The advantages are that traditional texture filtering and compression methods can be used and artists' workflow don't have to change. Spectral conversion is done on the fly for any colour evaluation. This means a small performance overhead is traded for memory impact. The disadvantage is that the user has limited or no control over the actual spectral shape with generic uplifting methods.

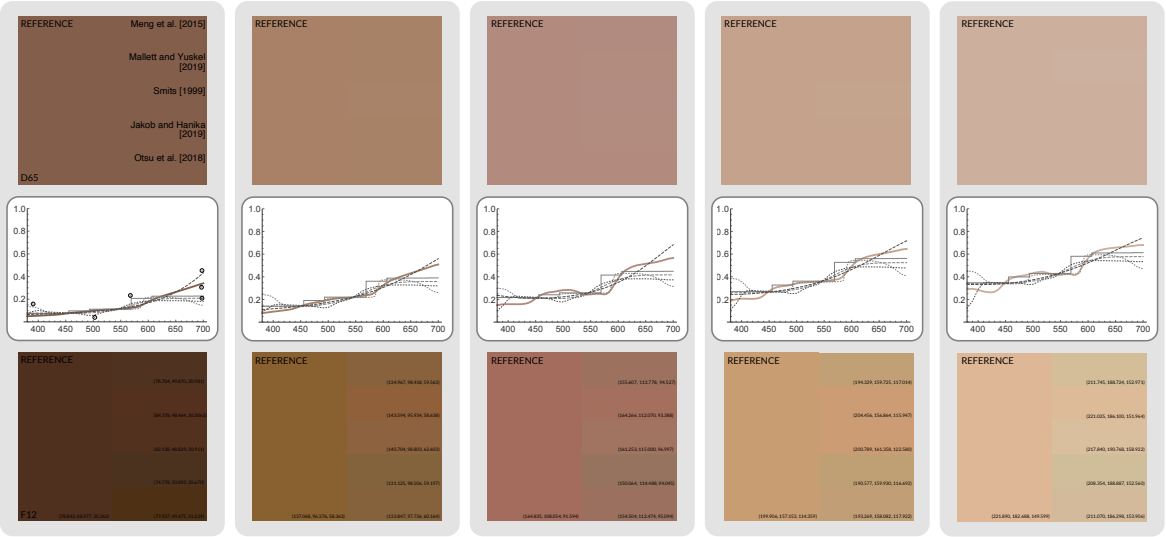


Figure 1.3: Comparison of generic uplifting methods and their generated spectra for various human skin tones to measured spectra Cooksey et al. [2017] under D65 (top row) and CIE F11 illumination (bottom row). From top to bottom in each square: Meng et al. [2015] ①, Mallett and Yuksel [2019] ②, Smits [1999] ③, Jakob and Hanika [2019] ④, Otsu et al. [2018] ⑤. The measured result ⑥ is always to the left. While all of them achieve an almost perfect roundtrip under D65, they do not necessarily generate a spectrum equivalent to the measured version. Note that Otsu et al. [2018] would allow target-based upliftig.

Figure 1.3 compares several production proven generic uplifting approaches. As it can be seen, all of them are more or less metameric under D65 illumination and will produce the same result as a measured skin spectrum. However, the w-shaped dip in the absorption coefficient of haemoglobin [Lister et al. 2012] will cause a spectral response that cannot easily

be replicated by generic algorithms and will result in significant colour shift once illumination is changed. This is unfortunate in scenarios like creation of digital humans where colour accuracy is of highest importance. One solution is to use a target-based method instead [Tódová et al. 2021] [Otsu et al. 2018] which informs the algorithm about the desired shape of the spectrum. They come, however, with memory and performance penalty as well as additional setup complexity from the artist side and have been rarely used in production so far. Chapter 3 will discuss in more detail how a true spectral workflow can enhance content creation and editing.

1.2.1.3 Colour Issues

Regardless of how a spectrum is generated, there is a fundamental problem when it comes to spectral reflectance colour. Since a reflectance colour is bound to be between 0 and 1, there is a limit to how spectrally pure it can become while retaining brightness. So while in theory spectral rendering is able to simulate the entire chromaticity diagram, for a given brightness, reflectance colours are only a sub-space of those. In other words, a reflectance colour cannot be bright and saturated at the same time. A plot of how brightness and saturation are connected can be seen in figure 1.4. While it is possible to generate spectrally pure colours, their intensity goes down the closer they get to the edge of the chromaticity diagram. As an alternative, Jung et al. [2019] and König et al. [2020] proposed to allow reflectance colours to go beyond 1 by including fluorescence, hence allowing more saturated bright reflectance colours. Note that emission can become arbitrarily saturated since its intensity can exceed 1.

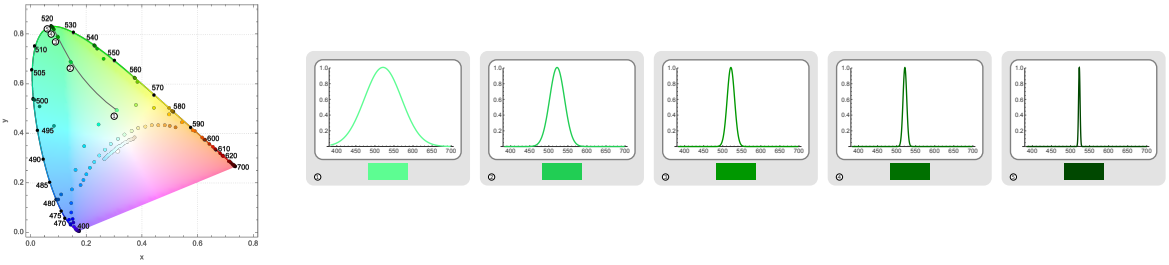


Figure 1.4: Plots of Gaussian spectra with mean ranging from 380nm to 720nm with sigma = (2, 5, 10, 20, 50, 100, 200nm). The more saturated the spectrum is, the darker it will get since it cannot exceed 1.0. Plots of Gaussian spectra at 520nm with sigma = (50, 20, 10, 5, 2nm) can be seen on the right. Note that the absence of samples in the magenta region is the result of using a single Gaussian only. [1]

1.2.2 Rendering

Interestingly enough, while getting spectral data into a renderer requires some effort, the rendering stage of an existing tristimulus renderer can be left comparably unchanged. Early spectral renderers were discretising spectra into several fixed bins and were tracing only these discrete samples, ignoring the rest of the spectrum. Or they would make use of any other data representations discussed in 1.2.1.1. While this approach removes colour noise, it is prone to artefacts like spectral aliasing [Wilkie et al. 2000] and will cause colour shifts, if

[1] Pink is not a pure spectral colour and hence cannot be found in a rainbow. It is in fact a lack of green in an otherwise equal-energy spectrum.

not enough samples are used. A better approach is to Monte Carlo sample the colour domain instead.

1.2.2.1 Single vs. Multi-Wavelength Tracing

In its most primitive form, spectral rendering can be done by stochastically sampling a single wavelength from a given spectral range and trace a ray with just that wavelength. While from an implementation point of view little has to change in the renderer (the traced wavelength needs to be stored on the path and e.g. one component of the existing RGB data struct could be used to store the path's energy), the final image will have more colour noise than the RGB version.

A more advanced approach is to still choose the wavelengths stochastically, but trace multiple wavelengths per path at the same time. A production-proven approach is hero wavelength spectral sampling (HWSS) Wilkie et al. [2014]. The key idea is that as long as paths are not diverging, sampling multiple wavelengths will greatly decrease noise with only a small performance overhead. The samples are chosen with a uniform distribution. The original paper proposed to use 4 wavelengths as the best trade-off between overhead and noise reduction. However, it should be noted, that any number of samples could be used. Using 3 wavelengths allows any existing tristimulus renderer to be easily converted into a spectral renderer supporting HWSS. Figure 1.5 shows how colour noise will go significantly down if multiple wavelengths are traced at the same time.

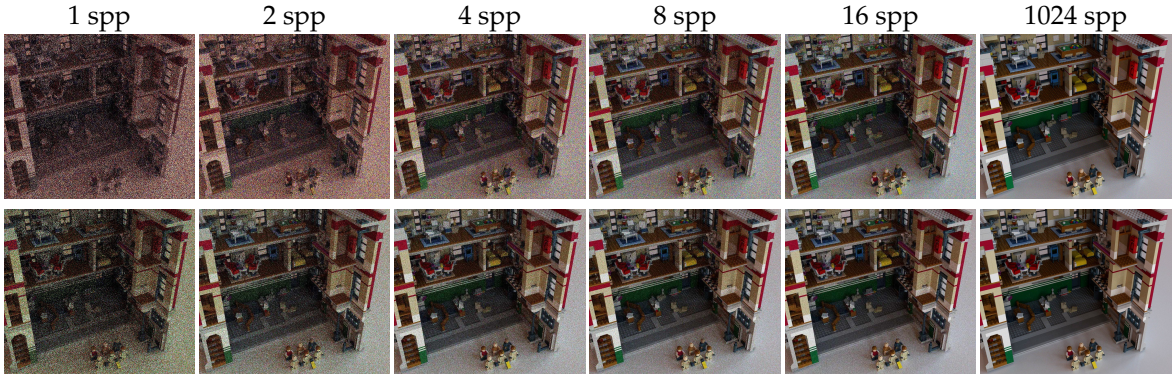


Figure 1.5: Single wavelength (top) and HWSS multi-wavelength rendering (bottom). 3 wavelengths are calculated in parallel. The colour noise is visibly reduced with HWSS with no significant performance overhead.

1.2.2.2 Non-Uniform Wavelength Selection

One problem with the uniform selection of wavelengths in HWSS is that spectra with high energy peaks will not be sampled as efficiently as they could, and consequently such scenes will experience a much higher noise level. Various approaches exist to achieve better wavelength selection, ranging from pre-analysing the image [van de Ruit and Eisemann 2021] to choosing a better wavelength distribution to combine the contributions of multiple importance-sampled wavelengths [West et al. 2020]. Chapter 4 will discuss in more detail how this problem is solved in a production renderer.

1.2.3 Output

Once rendering is done, the spectral data needs to either be stored into a spectral output format or converted to a tristimulus representation. This is done by first going to XYZ or CameraRGB and then to RGB colour space. For movie productions, the XYZ conversion is less interesting than the conversion though camera space since the primary goal is to replicate camera output.

To obtain the camera sensitivity curves CamR | B | G of a camera, it has to be measured and a matrix has to be computed that will convert CameraRGB to RGB. In the spirit of Sun et al. [1999] and Wyman et al. [2013], we converted several interesting camera sensitivity curves from [Digital 2022] used in movie production to analytical functions. All are fitted to the same two exponential functions:

$$CamRGB = a * \exp(-b * x * (\frac{x-c}{d})^i) + e * \exp(-f * x * (\frac{x-g}{h})^j)$$

Please note that these fits are by no means as accurate as using the tabulated values. Nor do we expect them to be faster. They might, however, be useful for someone who wants to experiment with camera profiles due to their simplicity.

- **Red Mysterium X.**

$$C_{Rec709} = M_{SRGB} M_{MysteriumX} C_{CamRGB} = \begin{bmatrix} 2.3576 & -0.0298 & -0.7118 \\ -0.3075 & 2.7325 & -0.4056 \\ 0.0354 & -1.052 & 3.4438 \end{bmatrix} \begin{bmatrix} 0.8688 & 0.775 & 0.1828 \\ 0.28398 & 1.872 & -0.1929 \\ 0.04251 & -0.675 & 3.2105 \end{bmatrix} C_{CamRGB}$$

	a	b	c	d	e	f	g	h	i	j
CamR	0.96483	0.00039	619.502	25.3221	0.01442	0.000047	501.142	45.4519	4	2
CamG	0.09291	0.02769	559.683	346.715	0.50449	0.004775	544.43	65.9919	2	2
CamB	0.02480	0.02588	546.705	253.356	0.42028	56.9374	471.893	8340.07	4	2

- **Canon1DMkIII.**

$$C_{Rec709} = M_{SRGB} M_{Canon1DMkIII} C_{CamRGB} = \begin{bmatrix} 4.8262 & -0.9435 & 0.1134 \\ -0.2585 & 1.620 & -0.535 \\ 0.1079 & -0.442 & 1.8912 \end{bmatrix} \begin{bmatrix} 1.9176 & 0.1105 & 0.1967 \\ 0.8492 & 0.9262 & -0.222 \\ 0.1650 & -0.2451 & 1.7356 \end{bmatrix} C_{CamRGB}$$

	a	b	c	d	e	f	g	h	i	j
CamR	0.08579	0.61987	532.062	210.314	0.520503	0.005352	606.426	79.242	2	2
CamG	0.38979	0.15103	502.205	-55.8344	0.970322	0.004531	535.84	77.918	2	2
CamB	0.7937	0.08549	464.867	25.5899	0.052353	0.004449	535.582	21.591	4	2

- **Canon5DMkII.**

$$C_{Rec709} = M_{SRGB} M_{Canon5DMkII} C_{CamRGB} = \begin{bmatrix} 4.6477 & -1.1862 & 0.2182 \\ -0.4173 & 1.7105 & -0.6984 \\ 0.1375 & -0.51731 & 2.2709 \end{bmatrix} \begin{bmatrix} 1.7926 & 0.02905 & 0.25 \\ 0.6999 & 0.9337 & -0.289 \\ 0.1708 & -0.3107 & 2.079 \end{bmatrix} C_{CamRGB}$$

	a	b	c	d	e	f	g	h	i	j
CamR	0.19591	0.00537	590.737	33.005	0.48923	0.0001	606.542	13.4074	2	2
CamG	0.92157	0.00066	545.648	32.538	0.32964	0.0046	500.735	47.5462	2	2
CamB	0.51269	1.21866	461.553	217.684	0.23489	0.0034	486.44	73.5452	4	4

More information on how to manipulate camera and output data can be found in chapter 5.

1.3 Challenges

Spectral image synthesis and especially spectral rendering are by far not solved. While the light transport part is well understood and many techniques exist to improve colour noise, spectral input and output techniques are less well established. There is still comparably few research about how to improve spectral workflows. For example, most generic uplifting algorithms are based on two assumptions. Firstly, the smoothness-criterium assumes that since natural spectra are often smooth and will not exhibit sharp peak, neither should the generated spectrum. And secondly, the data is bound to be between 0 and 1. While this works well enough for most reflectance data, it causes considerable issues for emission spectra from LEDs or fluorescent lights which can have complex shapes with sharp peaks. Data like absorption or scattering coefficients are unbound. Some parameters like complex IORs are upsampled as RGB colour but are in fact wavelength-dependent data and should be treated as such. Target-based approaches provide some relief, but they don't fix the underlying issue and are too costly to use to a big scale. We lack spectral texture standards and ways to art-direct spectral responses. Similar things are true for the output side. De-noisers will not out of the box work on spectral scenes and will need special training. Without better compression techniques storage of spectral data becomes difficult.

Note that this course does not cover wave optics effects; while of considerable importance in the generation of realistic images, wave tracing is unfortunately often beyond what production renderers can handle. We hope, however, that in the future wave optics research will have an impact on production renders and will become as ubiquitous as path-tracing is now.

References

Catherine Cooksey, David Allen, and Benjamin Tsai. 2017. Reference Data Set of Human Skin Reflectance. https://doi.org/10.6028/jres.122.026

Weta Digital. 2022. *Physlight*. Physlight

R Donaldson. 1954. Spectrophotometry of fluorescent pigments. *British Journal of Applied Physics* 5, 6 (jun 1954), 210–214. https://doi.org/10.1088/0508-3443/5/6/303

Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomas Davidovic, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A Batch-Shading Architecture for Spectral Path Tracing in Movie Production. *ACM Transactions on Graphics* 37 (08 2018), 1–18. https://doi.org/10.1145/3182161

Alban Fichet, Romain Pacanowski, and Alexander Wilkie. 2021. An OpenEXR Layout for Spectral Images. *Journal of Computer Graphics Techniques (JCGT)* 10, 3 (29 September 2021), 1–18. http://jcgt.org/published/0010/03/01/

Qingqin Hua, Alban Fichet, and Alexander Wilkie. 2021. A Compact Representation for Fluorescent Spectral Data. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20211305

Longqian Huang, Ruichen Luo, Xu Liu, and Xiang Hao. 2022. Spectral imaging with deep learning. *Light: Science & Applications* 11 (03 2022), 61. https://doi.org/10.1038/s41377-022-00743-6

Wenzel Jakob and Johannes Hanika. 2019. A Low-Dimensional Function Space for Efficient Spectral Upsampling. *Computer Graphics Forum (Proceedings of Eurographics)* 38, 2 (March 2019).

Alisa Jung, A. Wilkie, J. Hanika, W. Jakob, and C. Dachsbacher. 2019. Wide Gamut Spectral Upsampling with Fluorescence. *Computer Graphics Forum* 38 (07 2019), 87–96. https://doi.org/10.1111/cgf.13773

Lars König, Alisa Jung, and Carsten Dachsbacher. 2020. Improving Spectral Upsampling with Fluorescence. In *Workshop on Material Appearance Modeling*, Reinhard Klein and Holly Rushmeier (Eds.). The Eurographics Association. https://doi.org/10.2312/mam.20201139

Tom Lister, Philip A. Wright, and Paul H. Chappell. 2012. Optical properties of human skin. *Journal of Biomedical Optics* 17, 9 (2012), 1 – 15. https://doi.org/10.1117/1.JBO.17.9.090901

Ian Mallett and Cem Yuksel. 2019. Spectral Primary Decomposition for Rendering with sRGB Reflectance. In *Eurographics Symposium on Rendering - DL-only and Industry Track*, Tamy Boubekeur and Pradeep Sen (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20191216

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

Arash Mirhashemi. 2018. Introducing spectral moment features in analyzing the SpecTex hyperspectral texture database. *Machine Vision and Applications* 29 (04 2018). https://doi.org/10.1007/s00138-017-0892-9

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

Mark S. Peercy. 1993. Linear Color Representations for Full Speed Spectral Rendering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) *(SIGGRAPH '93)*. Association for Computing Machinery, New York, NY, USA, 191–198. https://doi.org/10.1145/166117.166142

Scyllarus. 2016. *HSZ Data Format*. https://scyllarus.data61.csiro.au/data/hsz-data-format/

Brian Smits. 1999. An RGB-to-spectrum conversion for reflectances. *Journal of Graphics Tools* 4, 4 (1999), 11–22.

Yinlong Sun, M.S. Drew, and F.D. Fracchia. 1999. Representing spectral functions by a composite model of smooth and spiky components for efficient full-spectrum photorealism. In *Proceedings Workshop on Photometric Modeling for Computer Vision and Graphics (Cat. No.PR00271)*. 4–11. https://doi.org/10.1109/PMCVG.1999.787742

Lucia Tódová, Alexander Wilkie, and Luca Fascione. 2021. Moment-based Constrained Spectral Uplifting. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20211304

Mark van de Ruit and Elmar Eisemann. 2021. A multi-pass method for accelerated spectral sampling. *Computer Graphics Forum* 40 (2021).

Andrea Weidlich, Alex Forsythe, Scott Dyer, Thomas Mansencal, Johannes Hanika, Alexander Wilkie, Luke Emrose, and Anders Langlands. 2021. Spectral Imaging in Production: Course Notes Siggraph 2021. In *ACM SIGGRAPH 2021 Courses* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 14, 90 pages. https://doi.org/10.1145/3450508.3464582

Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. 2020. Continuous Multiple Importance Sampling. *ACM Trans. Graph.* 39, 4, Article 136 (jul 2020), 12 pages. https://doi.org/10.1145/3386569.3392436

A. Wilkie, S. Nawaz, M. Droske, A. Weidlich, and J. Hanika. 2014. Hero Wavelength Spectral Sampling. In *Proceedings of the 25th Eurographics Symposium on Rendering* (Lyon, France) *(EGSR '14)*. Eurographics Association, Goslar, DEU, 123–131.

Alexander Wilkie, Robert Tobler, and Werner Purgathofer. 2000. Raytracing of Dispersion Effects in Transparent Materials.

Chris Wyman, Peter-Pike Sloan, and Peter Shirley. 2013. Simple Analytic Approximations to the CIE XYZ Color Matching Functions. *Journal of Computer Graphics Techniques (JCGT)* 2, 2 (12 July 2013), 1–11. http://jcgt.org/published/0002/02/01/

X-Rite. 2022. *AxF Data Format*. https://www.xrite.com/appearance-exchange-format-axf

2

Multispectral Lighting Reproduction for Virtual Production

CHLOE LEGENDRE, *Netflix*

2.1 What is Virtual Production?

The umbrella term *Virtual Production* (VP) describes an entire category of nascent, technology-driven film-making techniques leveraging virtual and augmented reality, real-time rendering, and computational display systems. However, in recent years, one technique from this category that has increased in popularity is that of filming actors inside LED stages or "volumes" as shown in Fig. 2.1, surrounding them with photographed imagery or renderings of computer generated scenes (Holben [2020]; Kadner [2021a,c]). In this chapter, when we refer to VP, we are thus referring to this novel LED-based methodology.

2.1.1 Why use Virtual Production LED Stages?

There are three main benefits of filming actors inside a VP LED stage:

1. **Lighting Reproduction.** The LED panels often surround the actors and thus can be used as spatially-varying, highly controllable on-set light sources. When used in this manner, VP can enable *lighting reproduction*, where the actors are surrounded from all directions by individually controllable light sources that can be driven to reproduce the illumination of a given real-world or computer generated scene;

2. **In-Camera Backgrounds.** As LED panels are used to display virtual content behind the actors, you can directly record what is called an *in-camera background*, thereby eliminating some labor-intensive post-production work such as rotoscoping and chromakeying as would be required when filming in a typical greenscreen studio;

3. **Actor Immersion.** As the LED panels surround the actors with a rendering of the virtual scene, they may feel more immersed in this environment, thereby potentially enhancing their performances and providing natural eyelines compared to filming in a greenscreen studio.

In this chapter, our main focus will be on *lighting reproduction* aspect of virtual production, where a thorough view of the world as a *spectral renderer* will motivate our discussion

Figure 2.1: *Dark Bay* LED volume for VP, on the set of the Netflix show *1899*. Image credit: Alex Forge (Netflix).

around color rendition. We will also briefly discuss the color rendition aspects of the *in-camera backgrounds*.

2.2 Background: Lighting Reproduction

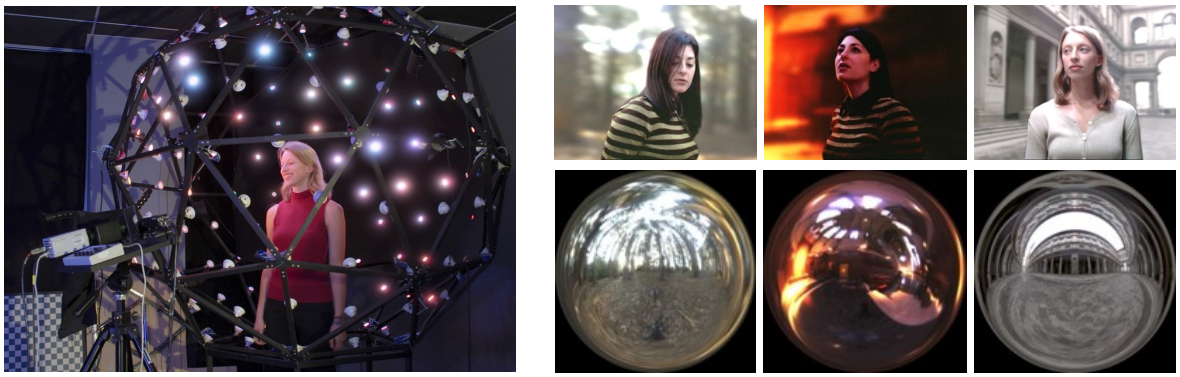


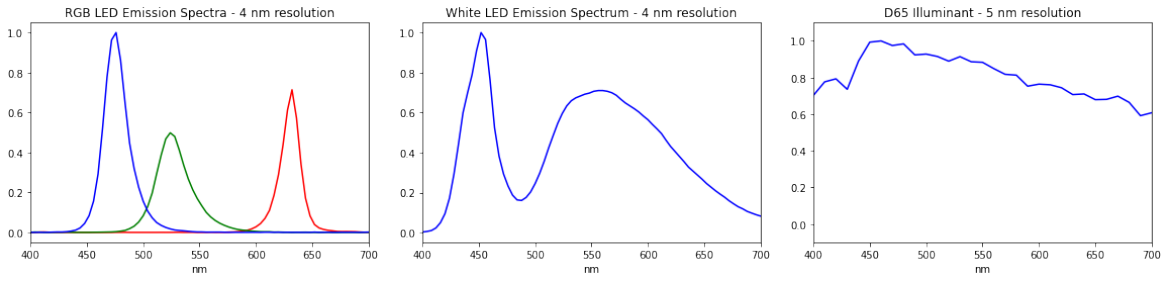
Figure 2.2: Left: Demonstrating image-based lighting reproduction for live-action compositing, where a subject is lit in a studio with RGB LEDs driven to replicate the color and intensity of a scene's measured incident illumination. Right: Three examples of live-action composites generated using this technique, with lighting environments in the bottom row. Images reproduced with permission from Debevec et al. [2002].

2.2.1 Early Examples

Although we now think of VP systems as being comprised of LED panels as in Fig. 2.1, initial systems for lighting reproduction consisted of more sparsely-arranged but still individually-controllable LED light sources affixed on a spherical rig. Debevec et al. [2002] first introduced such a system and the lighting reproduction methodology, surrounding an actor with RGB LED light sources driven computationally to reproduce the illumination of a target real-world or rendered scene (see Fig. 2.2, left). When displaying a record of illumination captured as a high dynamic range image-based lighting (HDR IBL) environment as in Debevec [1998], this technique was able to generally match a subject's appearance in the studio to how they would appear in a real-world scene (see Fig. 2.2, right). Furthermore, computer-generated (CG) HDR IBL environments could be displayed by the light sources of the spherical lighting rig, allowing for seamless composites between actors recorded in the studio and rendered CG imagery. An early instance of extending the spatial resolution of this technique using RGB LED panels was the "Light Box" built to surround and illuminate actors for the film *Gravity (2013)*, detailed in Hamon et al. [2014]. These systems were the initial predecessors to today's ubiquitous LED panel based VP stages.

2.2.2 RGB LEDs in Lighting Reproduction and Their Emission Spectra

Whether using a sparser arrangement of LED light sources or a denser matrix of LED pixels as found in most display panels, the majority of VP stages for lighting reproduction have used three-channel light sources comprised of red, green, and blue LEDs. Since the content to display, typically rendered or photographed HDR IBL environments, is also RGB, this at first glance seems like a logical design choice. However, although the RGB LEDs used in both early systems and today's LED panel based VP stages perform well at displaying light of a wide variety of *colors*, they cannot reproduce light of any target *spectrum*. Unfortunately, by combining different amounts of light produced by red, green, and blue LEDs, you will only be able to produce illumination with relatively "peaky" emission spectra, with distinct gaps between the spectra of each LED channel [see Fig. 2.3 (a)]. In contrast, when trying to reproduce the illumination of real-world scenes, we would ideally want to replicate the emission spectra of real-world illuminants, which are typically relatively broad-spectrum, covering most of the visible light wavelength range (400 - 700nm). For a few examples, see the emission spectra of a typical cool white LED and natural daylight in 2.3 (b, c).



(a) RGB LED (b) cool white LED (c) daylight (D65)

Figure 2.3: Emission spectra of: (a) typical RGB LEDs with a noticeable gap between the red and green peaks; (b) a typical broad-spectrum cool white LED; (c) natural daylight (D65).

2.2.3 Color Rendition Challenges with RGB LEDs

The unusual emission spectra produced by RGB LEDs leads to significant lighting reproduction color rendition errors as compared with color appearance under broad-spectrum, real-world illumination. It's not sufficient for an LED light source to simply appear the same *color* as the intended illuminant, since the world is a "spectral renderer" and thus color appearance is produced by integrating the fully spectral modulation of the light source emission spectra, the camera spectral sensitivity functions, and the material reflectance spectra.

Even if we ensure via a calibration process that the color of the RGB LED produced illumination matches that of the real-world target illuminant, the color appearance of materials with diverse reflectance spectra is unlikely to match that of the real world. As an example, we show in Fig. 2.4 (left) a photograph of a color chart calibration target lit by daylight, outdoors, and the same chart illuminated by an RGB LED system reproducing the correct *color* of illumination in Fig. 2.4 (middle). Finally, we show in Fig. 2.4 (right) the difference between these charts. In the RGB LED lit chart, we observe overly-saturated colors, despite the fact that the neutral gray and white squares match well.

(a) real environment (b) RGB LED lighting (c) sampled from (a, b)

Figure 2.4: (a) A color chart photographed in a real outdoor environment; (b) Photographed using lighting reproduction with RGB LEDs calibrated to match the color balance of the white square; (c) comparison between (a) and (b). Background squares are pixel values sampled from (a) and foreground circles are pixel values sampled from (b). Although the white squares match, the remaining color squares do not. Images reproduced with permission from LeGendre et al. [2022].

2.2.4 Growing Cinematographer Awareness of "Chromatic Chaos"

However, the problem extends beyond color charts and is one of practical concern to cinematographers. Given the recent popularity of LED volume based VP techniques, filmmakers are becoming newly aware of the color rendition challenges that come with using RGB-only lighting systems, as noted and observed by Noah Kadner and Craig Kief in their recent *American Cinematographer* article "Color Fidelity in LED Volumes" (Kadner [2021b]). Most notably, in an RGB LED VP volume, skin tones shift to pink or red, orange materials shift toward red, cyan materials shift toward blue, and yellow materials darken. These color rendition errors result from the spectral gaps between the red, green, and blue LEDs, and from the relatively long wavelength of the red LEDs which illuminates skin where it is relatively more reflective. To quote Weidlich et al. [2021] (last year's predecessor to this course, Sec. 3.3.3.1), "The lack of amber spectral contribution produces a poor [color] rendition of human skin and orange objects lit by the LED wall." For an example of this effect, see Fig. 2.5.

Even before the rise of LED volume filming techniques, The Academy of Motion Pictures Arts and Sciences convened a "Solid State Lighting" working group in part to combat the

(a) indoors, natural light (b) RGB LED panel VP stage

Figure 2.5: (a) A subject illuminated by indoor daytime natural lighting; (b) Reproducing the illumination inside an RGB LED panel VP stage, with noticeable color rendition errors for skin tones and yellow and orange patches on the subject's shirt. The *color* of the illumination matched across both scenes, but the illumination *spectra* differed. Images reproduced with permission from LeGendre et al. [2022].

"chromatic chaos" (Erland [2011]) of solid-state illumination, culminating in the proposal of the Spectral Similarity Index (SSI) to alert cinematographers to potential color rendition challenges when using a particular light source on set (Holm et al. [2016]).

2.2.5 A Prescient Warning

Far before RGB LED panels were used extensively for virtual production, the aforementioned color rendition limitation for lighting reproduction was initially presaged in the "Future Work" section of Debevec et al. [2002] upon introducing the technique:

> "In this work we calibrated the light stage making the common trichromatic approximation (Borges [1991]) to the interaction of incident illumination with reflective surfaces. For illumination and surfaces with complex spectra, such as fluorescent lights and certain varieties of fabric, the material's reflection of reproduced illumination in the light stage could be noticeably different that its actual appearance under the original lighting. This problem could be addressed through multispectral imaging of the incident illumination (Gat and Torrance [1998]), and by illuminating the actor with additional colors of LEDs. Adding yellow and turquoise LEDs as a beginning would serve to round out our illumination's color gamut."

Although it is the *spectrum* and not the *color gamut* that needs "rounding out," this passage nonetheless motivates the methodologies developed in the remainder of this chapter, focusing first on multispectral extensions to lighting reproduction and then on color rendition error amelioration strategies in case only RGB LEDs are available.

2.3 Outline of Remaining Sections

Now that we have presented the color rendition challenges associated with using RGB LEDs for lighting reproduction, we will preview the remaining sections where we attempt to practically overcome this limitation by:

1. Adding spectral channels beyond RGB to a *single* solid state light source and developing strategies for improving color matching using this source, as compared with the baseline approach of matching the *color* of the light produced regardless of the resultant spectrum;

2. Adding spectral channels beyond RGB to an *onmidirectional* lighting reproduction rig, and developing strategies for how to drive a set of multispectral LEDs to best match color rendition without requiring any spectral measurements or capturing fully multispectral HDR IBL environments;

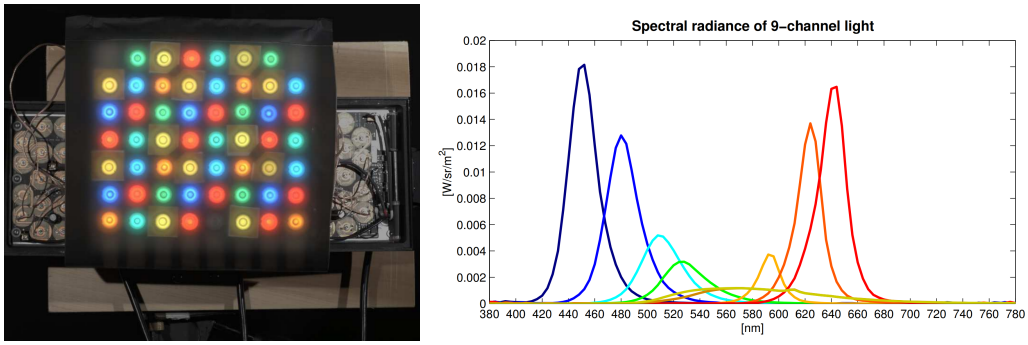
3. Selecting the optimal and minimal set of LEDs of distinct spectra for inclusion in a light source for lighting reproduction; and

4. Optimally calibrating an RGB-only lighting reproduction system, even if you aren't able add spectral channels but want improved color rendition for actors and their costumes (without sacrificing color matching for an in-camera background).

2.4 Beyond RGB: Adding Spectral Channels to a Light Source

2.4.1 A Nine-Channel, Multispectral LED Light Source

Following the guidance of the "Future Work" section of Debevec et al. [2002], Wenger et al. [2003] built a single light source with nine spectral channels to evaluate if color rendition using LED-based light sources could be improved, when trying to reproduce real-world, broad-spectrum lights e.g. daylight, incandescent, and fluorescent sources. This prototype light source is shown in Fig. 2.6(a), and the emission spectra of each of the nine channels is shown in Fig. 2.6(b).



(a) 9-channel LED light (b) measured emission spectra of (a)

Figure 2.6: (a) A nine-channel, multispectral LED light, with six additional channels beyond the typical RGB; (b) Emission spectra measured for each channel. Images reproduced with permission from Wenger et al. [2003].

At the time of publication, sufficiently bright LEDs emitting light in the yellow region of the electromagnetic spectrum between 540 and 600 nm were not available due to limitations of LED technology, so Wenger et al. [2003] employed LEDs characterized as white, royal blue, blue, cyan, green, amber, red-orange and red. In order to fill out the missing yellow part of the spectrum, they placed gel filters in front of the broad-spectrum white LED's optics, only allowing yellow light to be emitted. One white channel was covered with Lee filter #101 and the other channel is with Lee filter #104, both yellow in appearance. We can observe from Fig. 2.6(b) that these filtered sources were both broader spectrum and dimmer as compared with the shorter wavelength blue and longer wavelength red and amber LEDs.

Armed with this multispectral light source, we still, however, need to know how to compute the relative amount of each spectral channel to add together when using the light source to brew up a match for a target illuminant, or how to "drive" each LED in the light source. Towards this goal, Wenger et al. [2003] introduced three different techniques used to compute the relative amounts of each LED spectral channel to combine together to match a target illuminant, which the authors termed (1) Spectral Illuminant Matching, (2) Metameric Illuminant Matching, and (3) Metameric Reflectance Matching. The remaining sections of this chapter build upon these concepts, so we describe them here briefly and compare their performance for color rendition when using a single light source. For a formal analysis of each technique, we refer the reader to Wenger et al. [2003].

2.4.2 Spectral Illuminant Matching (SIM)

This method requires knowledge or measurement of the emission spectra of the different LED spectral channels in the light source along with the emission spectrum of the target illuminant. Given these two spectra, we can solve for an amount to drive each spectral channel to best reproduce the target illuminant spectrum in a least squares sense. One simply minimizes the difference between the LED reproduced spectrum and the target spectrum across each wavelength for which a measurement is available. However, in general, it will not be possible for these spectra to match very well due to the "peaky" LED spectra and the comparatively broad-spectrum nature of real-world light sources.

2.4.3 Metameric Illuminant Matching (MIM)

In this method, we compute an amount of each available spectral channel in the LED source to combine together to match the *color* of the target illuminant as seen by a target observer. Reasonable choices for a target observer include a specific motion picture camera or the human observer. For this method, we can use the recorded/characterized spectral sensitivity functions of the target camera, or, alternatively, rely only on images captured of a white surface illuminated by each LED channel and the target illuminant. "The problem can be set up as a minimization of the sum of the square residuals of the reproduction light color channels observed by the camera system to the target illuminant observed by the camera system" (Wenger et al. [2003]). This process is termed Metameric Illuminant Matching (MIM) because the reproduced illumination is a *metamer* of the target illumination, defined as a perceived color match despite the differing spectral power distributions. For a three-channel RGB LED light source, to reproduce an illuminant using MIM for a tristimulus camera generates three equations for three unknowns (the amount of each spectral channel to combine together), and so we can solve for these values exactly. However, for more than three-channel light

sources, as in the nine-channel light source, we can only solve for a minimum norm solution without further constraints. This method ensures that a white surface illuminated by the reproduced and target illumination will match exactly to the target camera.

2.4.4 Metameric Reflectance Matching (MRM)

In this method, we consider that our reproduced illuminant will be used to light not only *white* materials, but also materials with diverse reflectance spectra (e.g. skin, hair, costumes, and cosmetics). Thus, we compute the amount of each spectral channel in the LED source to combine together to match the colors of *multiple* materials as illuminated by the target light source, as seen by a target observer. While any materials could work, in practice Wenger et al. [2003] used a color calibration chart with 24 spectrally distinct patches, as introduced by McCamy et al. [1976]. "The minimization is set up as the sum of square relative differences between the key reflectance spectra illuminated with the target light spectrum observed by the [target] camera and the same spectra illuminated with the [LED reproduced] light spectra" observed by this same camera (Wenger et al. [2003]). Although Wenger et al. [2003] required spectral measurements of each part of the system, LeGendre et al. [2016] demonstrated that no spectral measurements are required to practice this technique, which we discuss in Sec. 2.5. Note that as multiple spectrally-distinct patches are included in the optimization, it is no longer the case that white surfaces under the target and reproduced illumination must match, unless such a constraint is specifically included in the optimization.

2.4.5 Comparison of Methods

In Fig. 2.7(d), we show an image of a subject illuminated with two real-world illuminants–tungsten from image-left and fluorescent from image-right. Next we show in the upper row, reproductions of the illumination using a light source made of RGB LEDs with each of the three matching techniques described above. Significant color rendition errors are visible, especially when evaluating the subject's denim jacket and skin. In the lower row, each of the lighting reproduction strategies is used with the nine-channel light of Fig. 2.6(a). We can draw two conclusions from these images: (1) color rendition is generally poor with RGB-only illumination compared with added spectral channels and (2) matching the spectrum in a least squares sense, or matching the color of the illuminant, will not lead to optimal color rendition as compared with matching color appearance across diverse reflectance spectra. While these conclusions emerged from tests performed using a single light source rather than a full, omnidirectional lighting rig, they nonetheless should inform our design considerations for LED volume virtual production systems when lighting reproduction is the primary goal.

2.5 Multispectral Lighting Reproduction: Adding Spectral Channels beyond RGB to an Omnidirectional Light Stage

From a hardware perspective, the next method that we describe (LeGendre et al. [2016]) combines the omnidirectional lighting reproduction rig of Debevec et al. [2002] with the multispectral light source idea of Wenger et al. [2003]. However, as foreshadowed, while Wenger et al. [2003] required spectral characterization of each part of the imaging system

Figure 2.7: (d) Subject illuminated from image-left with a tungsten source and image-right with fluorescent source. (a, b, c) SIM, MIM, and MRM lighting reproduction using an RGB LED source. (e, f, g) SIM, MIM, and MRM lighting reproduction using the 9-channel light of Fig. 2.6(a). Images reproduced with permission from Wenger et al. [2003].

(light source emission spectra, material reflectance spectra, and camera spectral sensitivity functions), LeGendre et al. [2016] demonstrated that we can practically use metameric reflectance matching for multispectral lighting reproduction without requiring *any* spectral measurements of any system component. Furthermore, LeGendre et al. [2016] introduced a technique for optimally displaying RGB-only HDR IBL environments in a multispectral lighting reproduction system. This is a fundamentally challenging problem, as it requires solving for how to drive "RGB++" light sources given only RGB lighting information in the form of HDR IBL environments. However, by adding color chart observations to the typical HDR IBL capture, the technique of LeGendre et al. [2016] is able to drive each LED of a lighting reproduction system to optimally reproduce the color rendition properties of any real-world scene.

In Fig. 2.8(a) we show a wide-angle, fish-eye view into the omnidirectional, six-channel multispectral light stage used in LeGendre et al. [2016]. Each of more than 300 lighting directions included light sources made of red, green, blue, cyan, amber, and broad-spectrum white (RGBCAW), with unique emission spectra shown in Fig. 2.9(b). Fig. 2.8(b-e) demonstrates the multispectral lighting reproduction technique of LeGendre et al. [2016] in action.

2.5.1 Method Overview

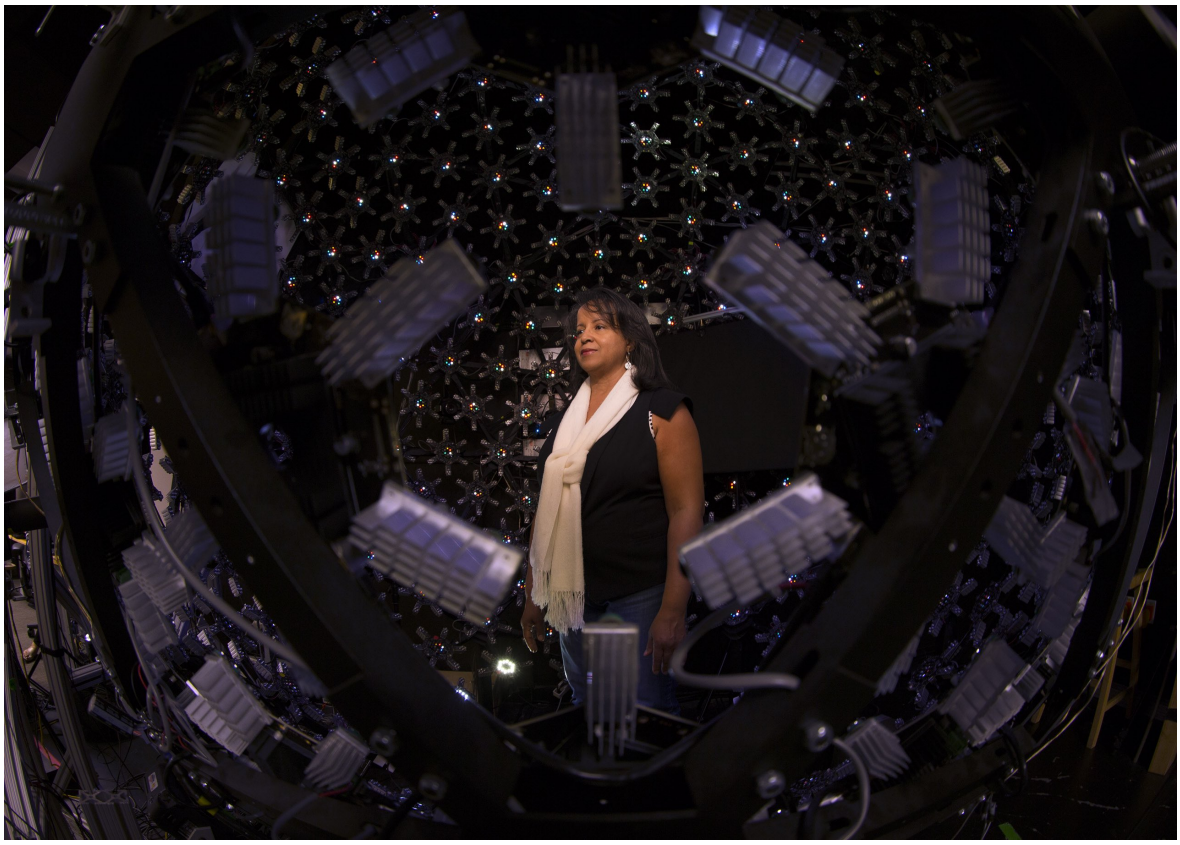
We summarize the technique for multispectral lighting reproduction as in LeGendre et al. [2016] as follows:

1. Photograph a color chart calibration target under each of the different available LEDs (each spectral channel of the light stage).

2. Record scene illumination using panoramic photography for HDR IBL capture plus observations of a color chart facing one or more directions.

3. For each LED light source, estimate the appearance of a *virtual* color chart reflecting its direction of light from the environment.

4. Drive the light source LEDs so that they best illuminate the virtual color chart with the estimated appearance.



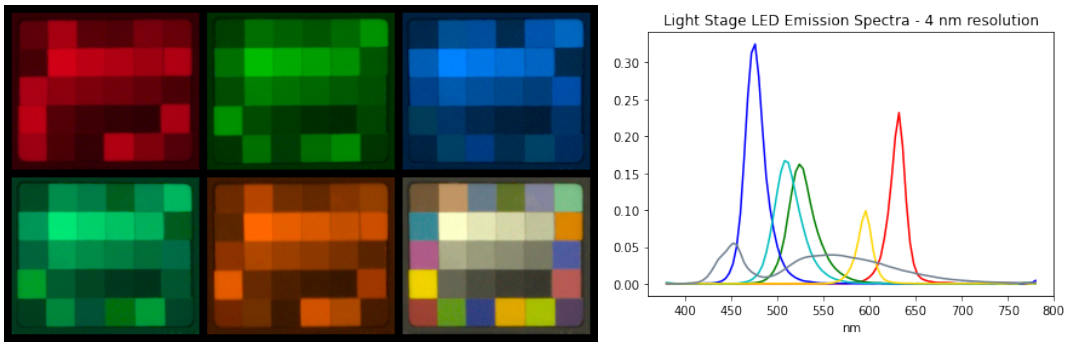
(a) six-channel, multispectral light stage

(b) real lighting (c) LED light stage (d) real lighting (e) LED light stage

Figure 2.8: (a) A subject in the six-channel multispectral light stage, seen through a fish-eye lens. (b, d) The subject photographed in two spectrally-complex real-world environments. (c, e) The subject photographed inside the multispectral light stage reproducing the real-world illumination. Images reproduced with permission from LeGendre et al. [2016].

2.5.2 Multispectral LED Stage Calibration and Matching a Chart Appearance

As a one-time calibration step, we can first photograph the appearance of a color chart under each spectral channel, as shown in Fig. 2.9. Because of the superposition principle for light, we know that the appearance of the color chart placed inside the multispectral light stage reproducing any lighting environment will necessarily be a linear combination of these six images (one for each available spectrum). Thus, without requiring any spectral measurements, we can solve for the amount of each one of these six images to add together to produce the appearance of a color chart lit by a target illuminant, which becomes the weight with which we drive each spectral channel of a particular LED light source. Thus we must solve for the amount of each of these images to add together, and we can do this using a non-negative least squares solution matching a target color chart. We show examples of the resultant matches for three target illuminants in Fig. 2.10, while leaving the formal analysis to LeGendre et al. [2016]. We should note that this is the "metameric reflectance matching" introduced by Wenger et al. [2003], but now without the spectral measurements. These calibration images are the result of modulating the camera spectral sensitivity, LED emission spectra, and color chart reflectance spectra – none of which are needed here in their fully-spectral form.



(a) calibration images (b) measured emission spectra of (a)

Figure 2.9: (a) Images of a color calibration target illuminated by each available spectral channel in the multispectral light stage: red, green, blue, cyan, amber, and broad-spectrum white. (b) The corresponding emission spectra measured for each channel. Images reproduced with permission from LeGendre et al. [2016].

2.5.3 Extending The Approach to All Lighting Directions

The previous section described how to drive a single LED source to match the color rendition properties of a target illuminant, given calibration images of a color chart illuminated by each available spectral channel. Here, we describe how to extend this to an omnidirectional lighting reproduction stage.

LeGendre et al. [2016] introduced a companion technique for multispectral HDR IBL acquisition, where the traditional HDR light probe capture was augmented by photographs of color charts facing in one or more directions, as shown in Fig. 2.11. These relatively diffuse color charts, which each integrate a different full hemisphere of incident illumination, provide a rough record of the color rendition properties of the lighting of the original scene. LeGendre et al. [2016] next hallucinate the appearance of a color chart for each direction for

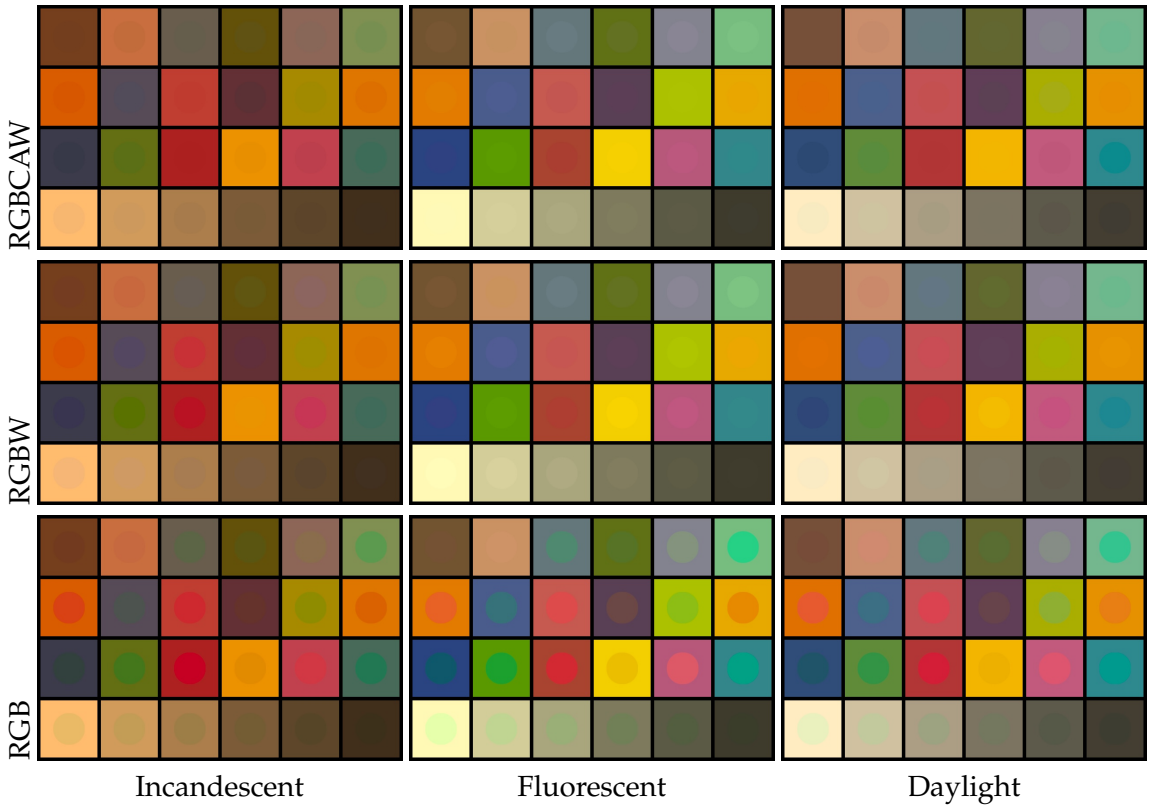


Figure 2.10: Comparison color charts for three target illuminants where the background of each square is the original chart appearance, and the circles (sometimes invisible) show the color chart under the reproduced illumination in the LED sphere using metameric reflectance matching. Six (RGBCAW), four (RGBW), and three (RGB) spectral channels are used. Images reproduced with permission from LeGendre et al. [2016].

which there is a light source in the light stage system, merging together the HDRI observations and the five color chart images. Since the HDRI provides a record of the *color* of the incident illumination for every direction in the environment, it is comparable (up to a scale factor) to the color of a white square of a color chart illuminated by only a tiny solid angle of the lighting environment. Thus, after interpolating an approximate color chart appearance for each direction of the environment from the five chart observations, these interpolated charts can be *tinted* so that each white square matches the color of the corresponding direction in the HDRI. This process ultimately generates a target color chart for each direction for which there is a light in the light stage, and then the non-negative least squares minimization using metameric reflectance matching can be used as before to solve for an amount to drive each spectral channel of each individual light source.

2.5.4 Results

In addition to the images in the bottom row of Fig. 2.8, Fig. 2.12 shows images of a subject in the original lighting environment and then in the reproduction of the environment inside the multispectral light stage, using the omnidirectional metameric reflectance matching approach. For these images, no image-specific color correct was performed. For completeness, Fig. 2.12 shows images of a subject under a reproduction of the illumination in six-channel

(a) shaded environment (b) sunny environment

Figure 2.11: Two lighting environments recorded using traditional HDR IBL capture, plus simultaneous capture of color charts facing in multiple directions. Images reproduced with permission from LeGendre et al. [2016].

light (RGBCAW), four-channel light (RGBW), and RGB-only illumination. As described in Sec. 2.2.3, RGB-only illumination used for lighting reproduction produces overly-saturated and hue-shifted colors. Adding broad-spectrum white LEDs to an RGB light source used for lighting reproduction is already sufficient to improve color rendition as compared with RGB-only lighting. Quantitative results for the color chart follow these trends as well.

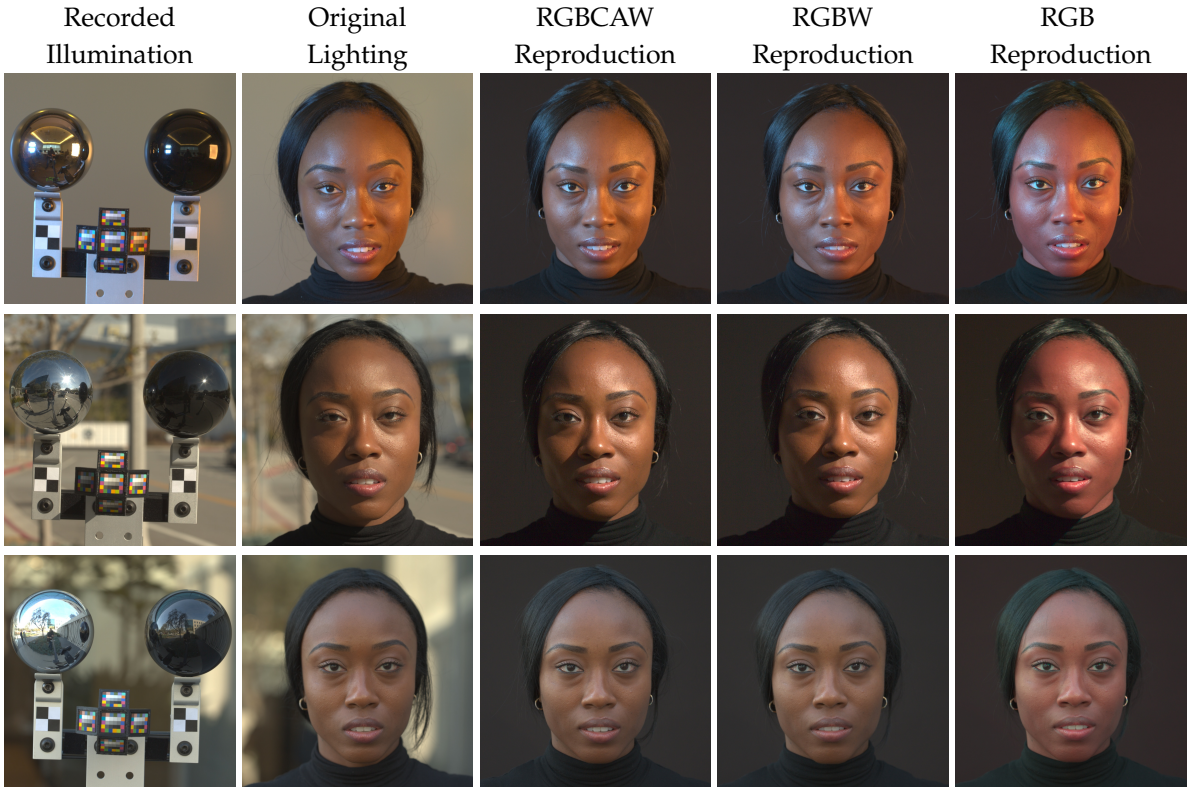


Figure 2.12: A subject in a variety of lighting environments (left columns) with three different lighting reproduction techniques. Generally, RGBCAW and RGBW reproduction produces accurate results, whereas using just RGB LEDs tends to over-saturate colors as expected. Images reproduced with permission from LeGendre et al. [2016].

2.6 Optimal LED Selection for Lighting Reproduction

In Sec. 2.4, we observed improved color rendition when employing a nine-channel light source as compared with using RGB-only lighting. In Sec. 2.5, we also observed improved color rendition when using six-channel (RGBCAW) lighting and even four-channel (RGBW) lighting. While the light stage of LeGendre et al. [2016] was built using cyan, amber, and white LEDs as added spectral channels beyond RGB, these were selected somewhat only with the goal of "filling in the spectral gaps" between the RGB LEDs. In follow-up work, LeGendre et al. [2017] presented a theoretical analysis that affords the selection of a *minimal* set of LEDs of distinct spectra to include in a solid state lighting reproduction system for optimal color rendition, selecting among 11 different LEDs representative of those commercially available. Fig. 2.13 shows the emission spectra of the LEDs under consideration.

LeGendre et al. [2017] demonstrated that the two best LEDs to add to an RGB system to improve color rendition are a broad-spectrum "phosphor-converted" amber (termed PC amber) and a broad-spectrum white LED. Both of these LEDs are manufactured as royal blue LEDs with a yellow-colored phosphor covering the diode to spread out and increase the wavelength of light emitted from each LED unit. The intuition here is that it is indeed important to cover the full visible spectrum (to fill in the RGB spectral gaps), and these broad-spectrum sources best perform this job. Unfortunately, despite many advances in

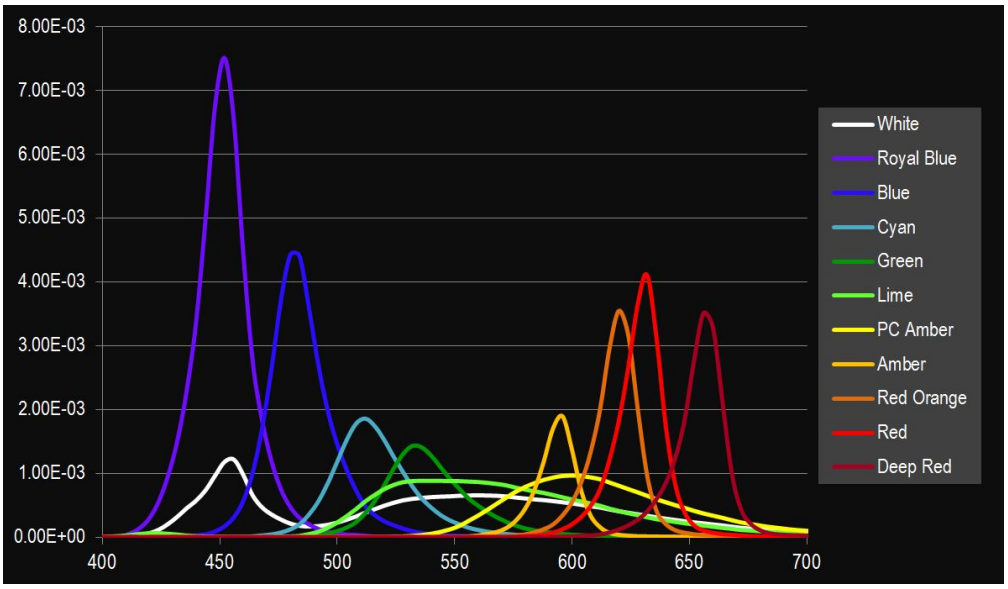


Figure 2.13: The emission spectra of the 11 distinct, commercially-available LEDs under consideration to form a minimal set of LEDs to use in a multispectral lighting reproduction system. Image reproduced with permission from LeGendre et al. [2017].

LED technology since the publication of Wenger et al. [2003], sufficiently bright, narrow band LEDs in the yellow part of the visible spectrum are still unavailable. Thus we'll have to "make do" with a phosphor-converted broader spectrum yellow in our lighting reproduction systems.

In Fig. 2.14, we show theoretical results using metameric reflectance matching for the human observer, with the goal of reproducing the color rendition properties of real-world illuminants including incandescent, daylight, and fluorescent when using all 11 LEDs of distinct spectra, five LEDs (RGBW + PC amber), four LEDs (RGBW), and three LEDs (RGB-only). Since light sources of a lighting reproduction system must frequently reproduce *indirect* illumination, LeGendre et al. [2017] also considered color rendition when reproducing a daylight spectrum modulated by the measured reflectance spectra of grass and sand. From these results, we observe that there are diminishing returns when using more than *five* LEDs of distinct spectra in a lighting reproduction system when reproducing many common real-world illuminants, despite the high-quality results of Wenger et al. [2003] using a nine-channel light. Results are similar when employing metameric illuminant matching or spectral illuminant matching, and they are similar regardless of the observer employed. LeGendre et al. [2017] further considered a thorough database of cameras of differing spectral sensitivities (Jiang et al. [2013]), along with a database of skin reflectance spectra (Uzair et al. [2015]) to move beyond the materials of the color chart.

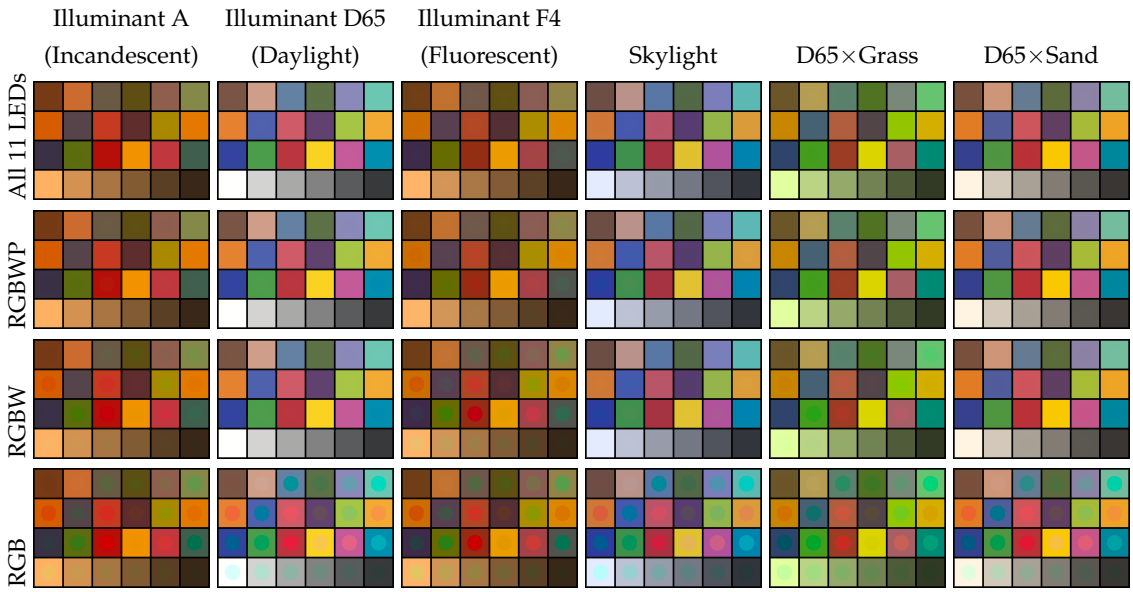


Figure 2.14: Color matching results using Metameric Reflectance Matching for different direct and indirect illuminants for the CIE 2° standard observer, with illumination reproduced using various LED combinations. The background squares represent the ground truth computed color chart appearances under the target illuminants, while the foreground circles represent the computed color chart appearances under LED-reproduced illumination. Row 1 shows the lighting reproduction results using all 11 LEDs of distinct spectra. Row 2 shows results using 5 LEDs only (RGB, White and PC Amber). Row 3 shows results using RGBW only. Row 4 shows the result using RGB only. XYZ tristimulus values are converted to the sRGB color space for display. RGBW lighting produces accurate color rendition for D65, Skylight, and D65 modulated by grass and sand reflectance spectra. Adding PC Amber yields accurate color rendition for illuminants A and F4. Images reproduced with permission from LeGendre et al. [2017].

2.7 Optimal Color Rendition using Only RGB LEDs for Virtual Production

Although we presented in previous sections approaches for ameliorating the color rendition challenges when using RGB LED lighting that added spectra channels to either a single light source (Wenger et al. [2003]) or a full omnidirectional lighting rig (LeGendre et al. [2016]), today's reality is that most virtual production systems are made of RGB-only LED panels originally designed for use as display systems. In this section, we thus present an practical color calibration process for such RGB-only virtual production stages which improves the color rendition in lighting reproduction *without* requiring the addition of new spectral channels. When using LED panel based lighting instruments, in contrast to a sparser arrangements of individual light sources as in Debevec et al. [2002] or LeGendre et al. [2016], we're typically aiming to record an in-camera background in addition to reproducing illumination on the actors and foreground subjects. Thus, we also need to consider the color appearance of these LED pixels being recorded directly by the camera. Finally, although none the the previous approaches considered a post-imaging color correction step, in this section we'll leverage this extra step to get an extra boost in color rendition performance when using only RGB LEDs.

To improve the color rendition of an RGB LED lighting reproduction system without sacrificing the color matching of the in-camera background, LeGendre et al. [2022] proposed a method that optimizes three separate linear color correction transformations for 1) the LED panel pixels visible in the camera's field of view ("in-camera-frustum"), 2) the pixels outside the camera's field of view illuminating the subjects ("out-of-camera-frustum"), and – as a post-process – 3) the pixel values recorded by the studio camera. The result is that footage shot in an RGB LED panel virtual production stage can exhibit more accurate skin tones and costume colors while still reproducing the desired colors of the in-camera background, as in Fig. 2.15. [A note to readers that this section is abridged from LeGendre et al. [2022].]

(a) subjects lit by real outdoor environment

(b) subjects lit by calibrated VP stage (baseline)

(c) subjects lit by calibrated VP stage (new method)

Figure 2.15: (a) Subjects in colorful clothing with a color chart and lighting reference spheres in an outdoor environment. (b) The same subjects in an RGB LED virtual production stage lit by color-matched imagery of the environment, showing color rendition errors in clothing colors and skin tones. (c) The subjects in an RGB LED virtual production stage calibrated using a the technique of LeGendre et al. [2022] to optimize both color rendition accuracy and in-camera background appearance. Images reproduced with permission from LeGendre et al. [2022].

2.7.1 Background: Prior Color Calibration Techniques for RGB LED Virtual Production Stages

Digital imaging technologists working to color calibrate the typical RGB LED panels used in VP stages today aim to make color values in the content to be displayed actually appear those colors to the camera. This calibration process takes into account both the spectral sensitivity of the motion picture camera and the emission spectra of the RGB LEDs, although it typically requires no spectral measurements. One way of ensuring that each color displayed by the

LED panel and then observed by the camera matches the target color of the content is to ensure that the pure red, pure green, and pure blue primaries all match as well. That is to say: red being displayed by the LED wall and recorded by the camera matches the target red of the content, and the same holds for green and blue. In practice this is achieved by photographing with the motion picture camera a patch of each color primary as displayed by the panels. As noted in Debevec et al. [2002], if the primaries match, then all other colors including white must match as well, as all others are simply a linear combination of the three primaries. This primary-based calibration process has been outlined by Unreal Engine (a technology provider for real-time, in-camera visual effects) in the online guide "In-Camera VFX Camera Color Calibration" (unr [[n.d.]]). A variant was also introduced in last year's predecessor to this course (see sec. 3.3.3.1 of Weidlich et al. [2021]). Both of these approaches rely on computing a color transform (usually a 3×3 color matrix to apply to input RGB pixel values in the content to be displayed), with the goal of making the primaries of the panel-displayed content appear the correct colors to the motion picture camera.

We should note that this primary-matching process guarantees that the white point of the content will match the white point displayed by the panels after the 3×3 color transform has been applied, so it is a modified form of "metameric illuminant matching." However, as noted in Sec. 2.4.5, MIM using RGB-only LEDs leads to poor color rendition for lighting reproduction systems. We see the clear result of this poor color rendition in Fig. 2.5, where a subject was filmed in an indoor lighting environment in Fig. 2.5(a) and then in an RGB LED panel based VP system reproducing the illumination in Fig. 2.5(b). The most notable color rendition errors occur for the yellow and orange portions of the subject's shirt, and the subject's facial skin appears too pink. Thus, the above primary-based calibration process is ideal when the goal is simply to use the LED panels as a display system, say, for filming an in-camera background. Difficulties arise, however, when we consider that LED panels in VP are also being used for lighting reproduction where color rendition errors emerge. In this case, we are no longer simply recording the appearance of the LED panels displaying content directly, but now we must consider how these panels *illuminate* actors and their costumes, as we have considered in the previous sections of this course.

2.7.2 Method Overview

Since we know that RGB LEDs produce color rendition errors when simulating broad-spectrum lighting, LeGendre et al. [2022] proposed to solve for an optimal post-correction matrix to improve color matching. Such a post-correction matrix can desaturate overly pink/red skin tones while keeping neutral colors neutral, which cannot be accomplished by altering the content displayed on the LED stage alone. LeGendre et al. [2022] further showed that if the inverse of the post-matrix is applied to the in-camera-frustum area, then the in-camera panel appearance can be matched as well, similar to primary-based, MIM calibration techniques currently practiced in production.

The full technique thus employs three different matrices, as shown in Fig. 2.16:

1. a 3×3 pre-correction matrix $\mathbf{M}$ to be applied to the pixel values of the LED displayed content for the out-of-camera-frustum pixels (the parts of the LED stage predominantly used for illumination, not visible in-camera);

2. a different 3×3 pre-correction matrix **N** to be applied to pixel values of the LED displayed content, for the in-camera-frustum pixels (which appear as the in-camera background);

3. and a third 3×3 post-correction matrix **Q** to be applied to the final imagery captured by the camera, which color corrects both the in-camera background and recorded foreground content together, without requiring any foreground/background separation.

The method assumes that the LED panels as well as the digital cinema camera used throughout the process act as linear systems. For the camera this is typical. For the LED panels, this may require a calibration process where an image series is recorded of the LED panels displaying increasing pixel values, as in Debevec et al. [2002]. The method also assumes that the lighting information to be displayed in the VP stage has been recorded using panoramic, HDR photography such as that of Debevec [1998] and that a companion color chart captured at the panorama's center of projection has been photographed as well.

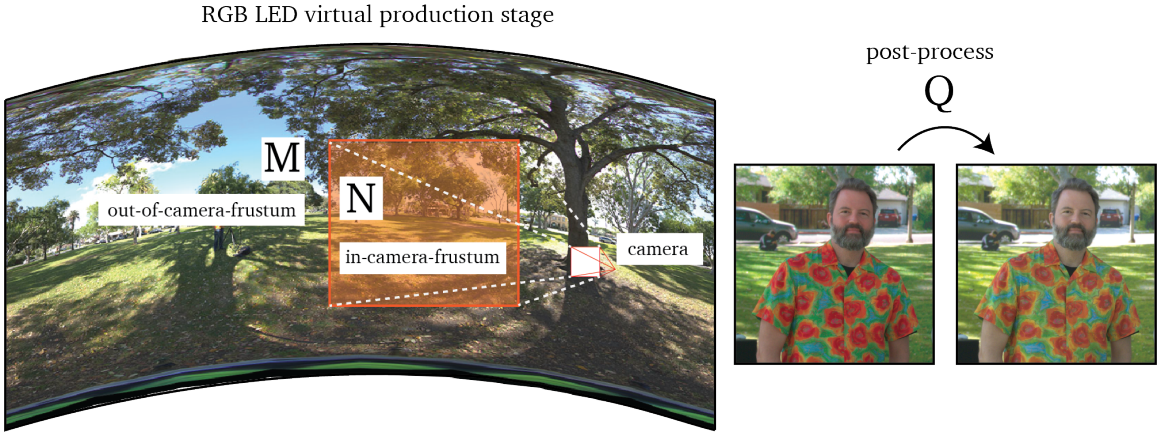


Figure 2.16: A visualization of the method. A 3×3 pre-correction matrix **M** is applied for out-of-camera-frustum content, while a different 3×3 pre-correction matrix **N** is applied for in-camera-frustum content. Finally, a 3×3 post-correction matrix **Q** is applied to the recorded image content. Image reproduced with permission from LeGendre et al. [2022].

2.7.2.1 Solving for M: Primary-based Calibration

This measurement allows LeGendre et al. [2022] to map the target scene's pixel colors to LED panel colors observed by the motion picture camera, ensuring a metameric illuminant match when a scene's HDRI map is displayed. For this calibration, LeGendre et al. [2022] display a patch of pure red, pure green, and pure blue on the in-camera-frustum LED panels and record their appearance to the camera, directly measuring the appearance of each LED channel in the VP stage to the camera. These patches can all be photographed in a single image. This is the same process that is used to generate a primary-calibrated LED panel (Weidlich et al. [2021]; unr [[n.d.]]). These sampled values are concatenated along columns to obtain a 3×3 matrix that LeGendre et al. [2022] call [**SL**], because its elements are the pairwise dot products of the camera's **S**pectral sensitivity functions and the **L**ED emission spectra. [**SL**] has the camera's color channels along rows, and the spectral channels of the

LED volume along columns. Given $[\mathbf{SL}]$, we can solve for $\mathbf{M} = [\mathbf{SL}]^{-1}$. For the full formal analysis, see LeGendre et al. [2022].

2.7.2.2 Solving for post-correcting Q: Color Rendition Calibration

The next goal is to solve for a 3×3 *post-correction* matrix $\mathbf{Q}$ which, when applied to the final image, makes a color chart lit by the VP stage displaying the HDR IBL environment look as close as possible to how it appeared in the real scene. Although we could, in practice, just photograph the chart as illuminated by the HDR IBL displayed by the panels (pre-corrected with $\mathbf{M}$), and solve for $\mathbf{Q}$ from this image directly, ideally we would be able to compute $\mathbf{Q}$ only from calibration imagery, captured once regardless of the number of lighting environments that we wish to display.

Towards this end, LeGendre et al. [2022] proposed photographing calibration imagery of the color chart as illuminated by each channel of the LED volume (the red, green, and blue spectral channels individually), following the procedure outlined by LeGendre et al. [2016] in Sec. 2.5.1. The core insight here is, again, that because of the superposition principle for light, any chart illuminated by the VP stage will resemble a linear combination of these three images. So, if we capture such data, we can simulate the appearance of a color chart illuminated by any given environment, rather than needing to photograph it each time to compute $\mathbf{Q}$. To capture these photographs, LeGendre et al. [2022] turn on a 1m $\times$ 1m square of the LED wall for each spectral channel and place a color chart 1m from the center of the square aimed directly toward the LED wall. The camera is oriented such that its optical axis makes a 45° angle with the surface normal direction of the color chart, leveraging the property that the color chart reflects light diffusely, as in 2.17 (top row). The resulting calibration images captured for a camera are shown in Fig. 2.17 (middle row), and, finally, the pixel values sampled from these images are shown in Fig. 2.17 (bottom row).

Next, we know that our HDR IBL displayed on the VP stage will use a *full sphere* of incident lighting directions rather than the limited lighting directions represented by the 1m $\times$ 1m square. Thus, LeGendre et al. [2022] compute a scale factor β that encodes the difference in the solid angle coverage of a full sphere of even illumination and this calibration setup geometry. And, finally, to compute the appearance of a color chart under a particular HDR IBL, we need to consider how to scale or "tint" the appearance of each chart under a full sphere of each spectral channel – scaling each calibration image based on the amount of red, green, and blue that will be displayed by the VP stage for a given HDR IBL. For this LeGendre et al. [2022] leverage the fact that the image of the color chart under the target illumination that we'd like to reproduce in the VP stage has a white square pixel value $\mathbf{w_{avg}}$ equal up to scale factor to the diffuse convolution of the hemisphere of lighting directions facing the chart. As displayed in the VP stage, this will become $\mathbf{Mw_{avg}}$. So to predict the appearance of a color chart lit by an HDR IBL displayed in a VP volume, LeGendre et al. [2022] simply tint the β scaled calibration images of Fig. 2.17 by $\mathbf{Mw_{avg}}$. From this color chart appearance predicted when the VP stage displays a particular HDR IBL, LeGendre et al. [2022] setup a linear least squares solver to estimate the 3×3 post-correction matrix $\mathbf{Q}$ that optimizes the color rendition when compared with the original chart under the target real-world lighting environment. Again, for the full formal analysis, see LeGendre et al. [2022].

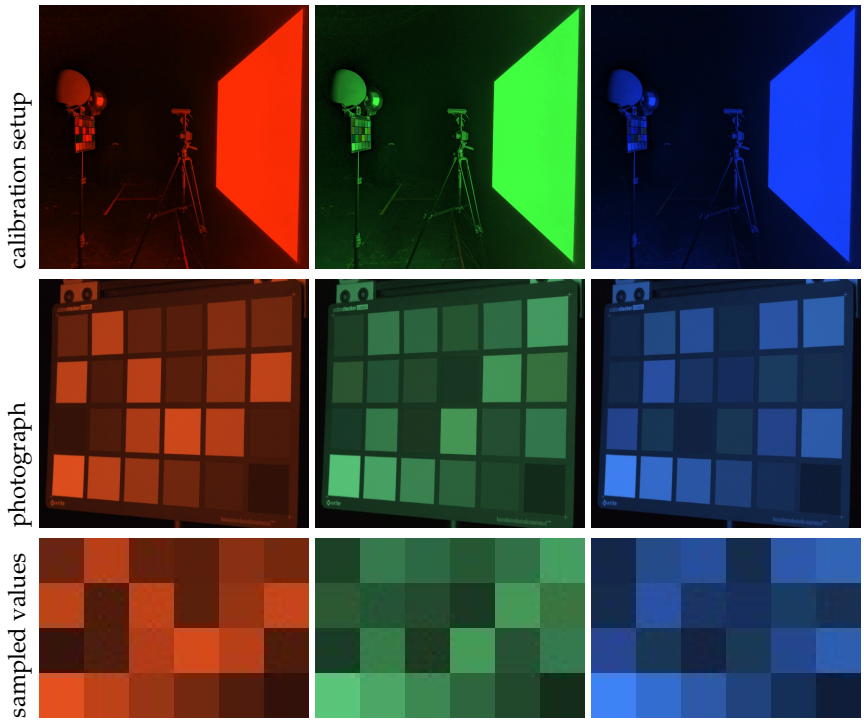


Figure 2.17: Top row: the color rendition calibration setup, photographed with a witness camera. A color chart is placed in front of a 1m × 1m square of illuminated LED panel, with the chart parallel to the panel surface. The motion picture camera is placed 45° off-angle from the surface normal of the chart. Middle row: photographs of the color chart lit by each spectral channel. Bottom row: pixel values sampled from the images of the middle row. Images reproduced with permission from LeGendre et al. [2022].

2.7.2.3 Solving for N: Inverting the effect of Q for the in-camera background

Unfortunately, if we apply the 3×3 post-correction matrix $\mathbf{Q}$ to the whole image, the in-camera background pixels will also be transformed and may no longer appear correct. While foreground detection or rotoscoping could allow the correction to be applied only to the foreground, this would require additional complexity. The key insight here from LeGendre et al. [2022] is that if we know in advance the post-correction matrix $\mathbf{Q}$ that we will ultimately apply to the final image or video sequence, we can apply the *inverse* of this correction to the in-camera-frustum part of the LED panels using $\mathbf{Q}^{-1}$. As the background pixels do not contribute to the lighting on the actors significantly, both the color rendition on the actors and the appearance of the in-camera pixels can be optimized at the same time, with no foreground/background separation required. In total, we can solve for $\mathbf{N} = \mathbf{M}\mathbf{Q}^{-1}$. Please see LeGendre et al. [2022] for the full derivation.

2.7.3 Black Level Subtraction

A current limitation of LED panels used for in-camera backgrounds is that the panels themselves reflect back some of the light which hits them, i.e., they have a non-zero *albedo*. This means that the panels oriented to light the actors also have the unwanted side-effect of illuminating the panels used for displaying the in-camera background. In our experiments, we observed panel albedo values ranging from ∼4-10%. To compensate for this, LeGendre

et al. [2022] adjust the pixel values displayed within the camera frustum with an RGB offset, computed based on a "black level" measurement. See LeGendre et al. [2022] for further details.

2.7.4 Results

LeGendre et al. [2022] captured seven spectrally-diverse lighting environments using a Canon 5D Mark III and a Canon 8-15mm fisheye lens with a 180° field-of-view, using multiple exposure HDR photography. Five of these lighting environments were captured indoors in a living room environment, each with its own distinct type of illumination: (1) broad-spectrum warm white LED lighting, (2) incandescent (tungsten halogen) lighting, (3) daylight shining through windows, (4) RGB LED "white" light, and (5) monochromatic yellow-orange sodium vapor lighting, with a distinct emission spectrum spike at 589 nm. The remaining two lighting environments were captured outdoors at a local park, (6) in the shade and (7) in direct sunlight. In each environment, to capture the color rendition properties of the real-world illumination, LeGendre et al. [2022] additionally photographed two subjects wearing brightly colored patterned clothing, along with a color chart, a diffuse gray sphere, and a mirrored sphere as reference.

In Fig. 2.15, Fig. 2.18, and Fig. 2.19 we show the real-world photographs, then the baseline VP lighting reproduction using **M** only for both in- and out-of-frustum content, and then the full multi-matrix pipeline of LeGendre et al. [2022] using black level subtraction. In each of the images captured within the VP stage, we show images of the target color chart composited into the in-camera background for *display*, along with a color chart *lit* by the out-of-frustum VP stage content, for the purpose of evaluating the color rendition for both real-world materials and those rendered by the panels for the in-camera background. In Fig. 2.18 and Fig. 2.19, we also show sampled color charts where the background squares represent the target color chart pixel values, and the foreground circles represent those sampled from the corresponding charts in the VP stage. For these comparison charts, we scaled the intensity of the target and sampled charts such that the green channel of the white squares matched to enhance visualization.

As anticipated, for the most part, the out-of-frustum *lit* color chart in the VP stage is overly saturated, with hue shifts in yellow and orange materials using the baseline calibration approach with **M** only, with the exception of the RGB LED based white lighting environment. It turns out that the RGB LEDs of the VP stage are quite good at reproducing the illumination of an RGB LED based lighting environment. Indeed, for the RGB LED scene, the post-correction **Q** matrix is close to the identity and does not have a very significant effect.

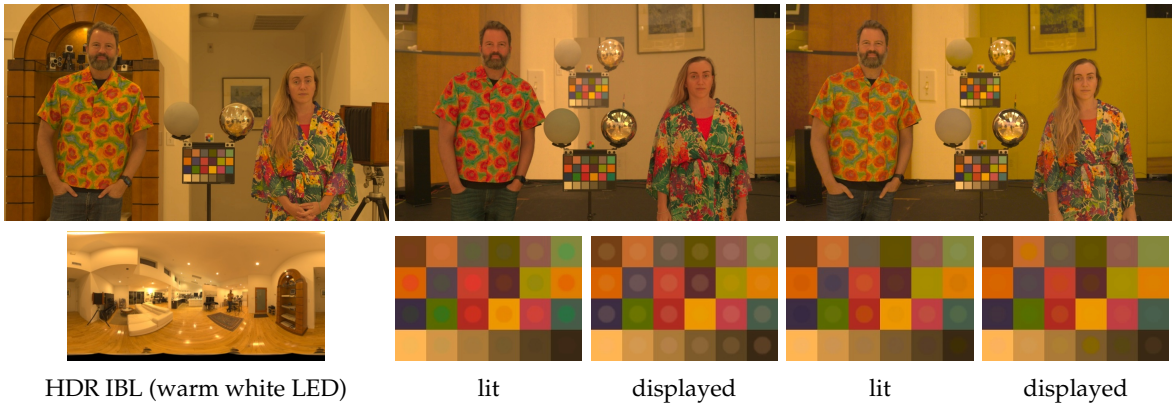
However, there are color rendition challenges for the remaining scenes that include broader spectrum illumination sources. For the remaining scenes, the **Q** matrix is able to desaturate colors as required, leading to improved color rendition for the out-of-frustum *lit* chart. While **Q** desaturates the overall image content, the in-frustum matrix **N** ensures that the in-camera background content is still as close as possible to the target color appearance. However, the black level subtraction is the more significant effect for the in-camera background color rendition due to the in-camera-frustum panels reflecting illumination from the out-of-frustum panels.

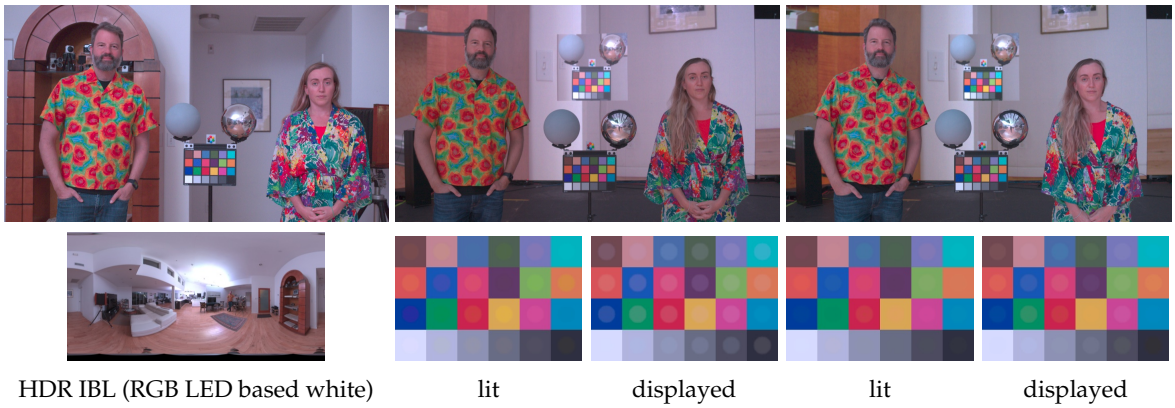
If we compare the qualitative residual errors for the *lit* charts of Fig. 2.18 and Fig. 2.19, we do not, however, see the "dots" disappear into the background squares, as we observed

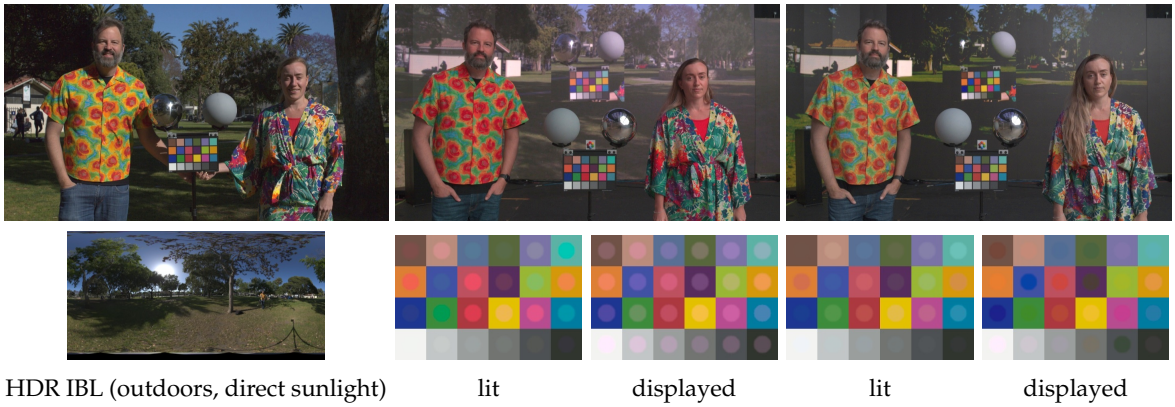
when adding spectral channels to a lighting reproduction system. Nonetheless, the post-correction matrix **Q** provides a powerful tool in our color rendition toolkit when working with RGB-only lighting.

2.8 Conclusion

In this chapter, we demonstrated that the RGB LED based display panels used in today's virtual production volumes yield color rendition challenges when used as light sources for lighting reproduction, despite their widespread popularity and success when used for filming in-camera backgrounds. We presented a series of methods targeted towards mitigating such color rendition challenges including: 1) adding spectral channels to individual light sources, 2) adding spectral channels to an omnidirectional lighting reproduction system, 3) figuring out *which* spectral channels to add to these systems to minimize hardware complexity, and, finally, 4) how to use a post-correction transform to optimally improve color rendition in RGB-only lighting systems while maintaining the color accuracy of an in-camera background, for scenarios where we're unable to round out the system with added LEDs of distinct spectra. RGB-only lighting systems dominate the world of virtual production for now, but nonetheless we hope to have armed all VP practitioners with an appreciation of the spectral aspects of lighting reproduction and their color rendition ramifications.

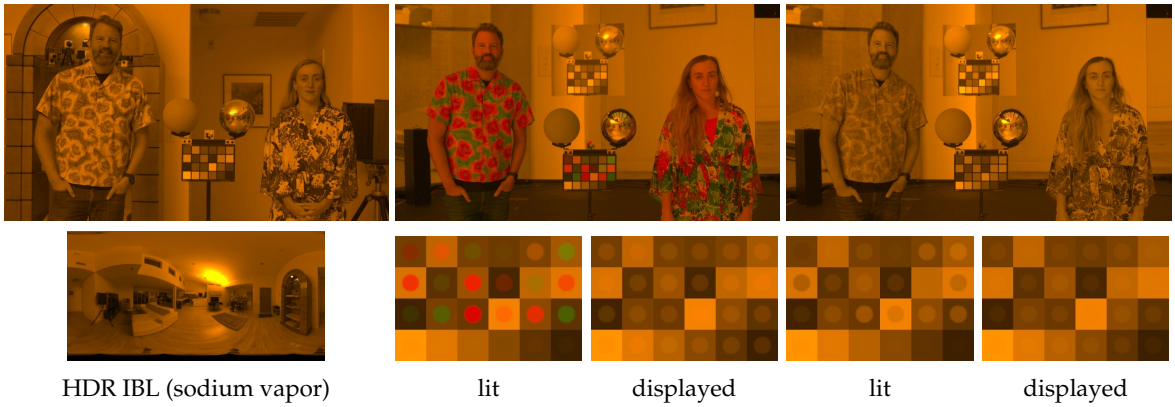


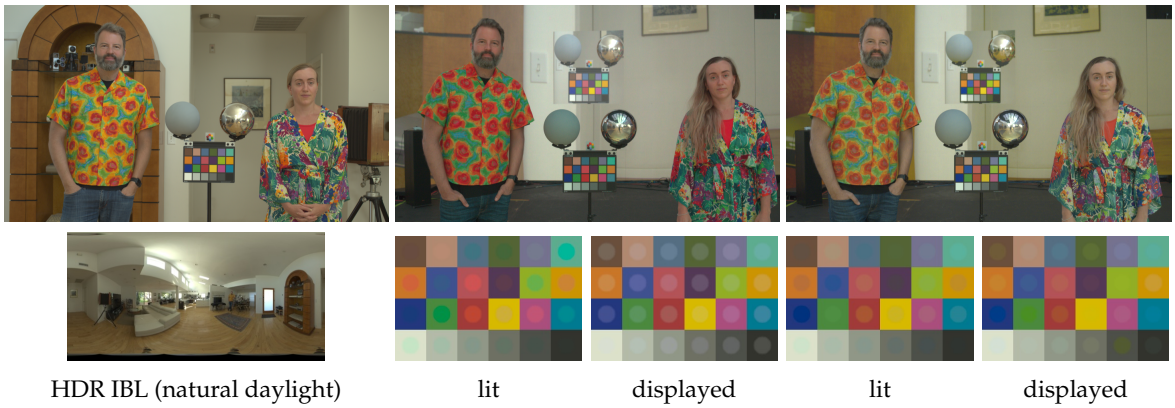


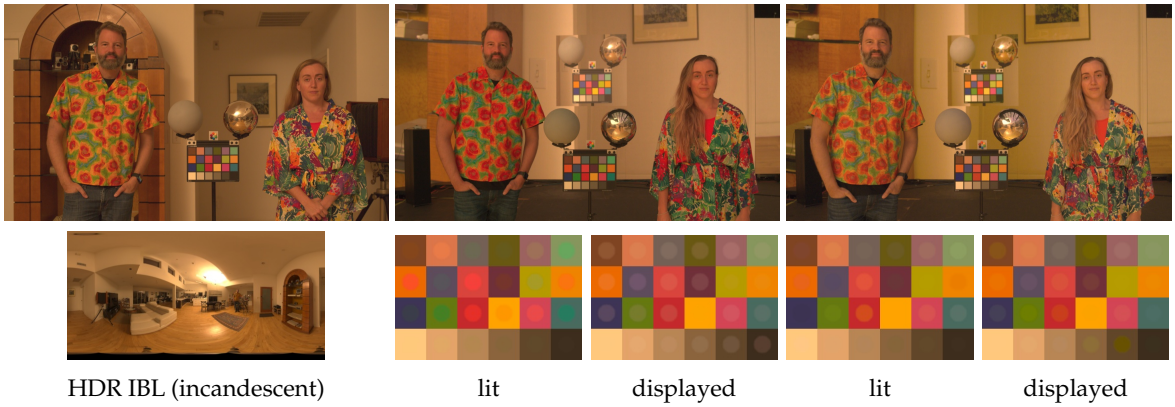


(a) photograph in real environment (b) VP using baseline (c) VP using new approach

Figure 2.18: For three spectrally-diverse lighting environments, we show subjects photographed in the real world (a), lighting reproduction in a VP stage using the baseline approach (b), and using the full approach (c). Compared with the baseline calibration method, the method of LeGendre et al. [2022] enables improved color rendition for the *lit* chart, desaturating skin tones and improving the appearance of orange/yellow materials. The black level subtraction removes the appearance of light bounced off the in-camera background LED panels. Observe the color rendition for the RGB LED based lighting environment (middle rows) is already quite good as we are asking RGB LED panels to reproduce RGB LED based illumination. Images reproduced with permission from LeGendre et al. [2022].







(a) photograph in real environment (b) VP using baseline (c) VP using new approach

Figure 2.19: For three spectrally-diverse lighting environments, we show subjects photographed in the real world (a), lighting reproduction in a VP stage using the baseline approach (b), and using the full approach (c). Compared with the baseline calibration method, the approach of LeGendre et al. [2022] enables improved color rendition for the *lit* chart, desaturating skin tones and improving the appearance of orange/yellow materials. The black level subtraction removes the appearance of light bounced off the in-camera background LED panels. For the sodium vapor lighting environment of the top row, **Q** is able to completely desaturate the RGB-LED color chart in the VP stage to better match the appearance of the nearly monochromatic sodium vapor illumination in the real world. Images reproduced with permission from LeGendre et al. [2022].

References

[n.d.]. In-camera VFX Camera Color Calibration. https://docs.unrealengine.com/4.27/en-US/WorkingWithMedia/IntegratingMedia/InCameraVFX/InCameraVFXCameraCalibration/

Carlos F Borges. 1991. Trichromatic approximation for computer graphics illumination models. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques (SIGGRAPH '91)*. 101–104.

Paul Debevec. 1998. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 189–198. https://doi.org/10.1145/280814.280864

Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. 2002. A Lighting Reproduction Approach to Live-Action Compositing. *ACM Trans. Graph.* 21, 3 (Jul 2002), 547–556. https://doi.org/10.1145/566654.566614

Jonathan Erland. 2011. Chromatic Chaos: Implications of Newly Introduced Forms of Stagelight. National Association of Broadcasters (NAB) Conference.

Nahum Gat and CA Torrance. 1998. Real-time multi-and hyper-spectral imaging for remote sensing and machine vision: an overview. In *Proc. 1998 ASAE Annual International Mtg*. Citeseer.

Pierre-Loïc Hamon, James Harmer, Stuart Penn, and Nicolas Scapel. 2014. Gravity: Motion control and face integration. In *ACM SIGGRAPH 2014 Talks*. 1–1.

Jay Holben. 2020. The Mandalorian: This Is the Way. https://ascmag.com/articles/the-mandalorian

Jack Holm, Tom Maier, Paul Debevec, Chloe LeGendre, Joshua Pines, Jonathan Erland, George Joblove, Scott Dyer, Blake Sloan, Joe di Gennaro, et al. 2016. A cinematographic spectral similarity index. In *SMPTE 2016 Annual Technical Conference and Exhibition*. SMPTE, 1–36.

Jun Jiang, Dengyu Liu, Jinwei Gu, and Sabine Süsstrunk. 2013. What is the space of spectral sensitivity functions for digital color cameras?. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 168–179.

Noah Kadner. 2021a. 1899 Wraps Innovative Virtual Production. https://ascmag.com/articles/1899-wraps-virtual-production

Noah Kadner. 2021b. Color Fidelity in LED Volumes. https://ascmag.com/articles/color-fidelity-in-led-volumes

Noah Kadner. 2021c. On The Walls: Virtual Production for Series Shooting. https://ascmag.com/articles/on-the-walls

Chloe LeGendre, Lukas Lepicovsky, and Paul Debevec. 2022. Jointly Optimizing Color Rendition and In-Camera Backgrounds in an RGB Virtual Production Stage. *(arXiv.org)* (2022).

Chloe LeGendre, Xueming Yu, and Paul Debevec. 2017. Optimal LED selection for multispectral lighting reproduction. *Electronic Imaging* 2017, 8 (2017), 25–32.

Chloe LeGendre, Xueming Yu, Dai Liu, Jay Busch, Andrew Jones, Sumanta Pattanaik, and Paul Debevec. 2016. Practical multispectral lighting reproduction. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Calvin S McCamy, Harold Marcus, James G Davidson, et al. 1976. A color-rendition chart. *J. App. Photog. Eng* 2, 3 (1976), 95–99.

Muhammad Uzair, Arif Mahmood, Faisal Shafait, Christian Nansen, and Ajmal Mian. 2015. Is spectral reflectance of the face a reliable biometric? *Optics express* 23, 12 (2015), 15160–15173.

Andrea Weidlich, Alex Forsythe, Scott Dyer, Thomas Mansencal, Johannes Hanika, Alexander Wilkie, Luke Emrose, and Anders Langlands. 2021. Spectral imaging in production: course notes Siggraph 2021. In *ACM SIGGRAPH 2021 Courses*. 1–90.

Andreas Wenger, Tim Hawkins, and Paul Debevec. 2003. Optimizing color matching in a lighting reproduction system for complex subject and illuminant spectra. In *Rendering Techniques*. Citeseer, 249–259.

3

Estimation of Spectral Biophysical Skin Properties from Captured RGB Albedo for Digital Humans

CARLOS ALIAGA AND CHRISTOPHE HERY, *Meta Reality Labs Research*

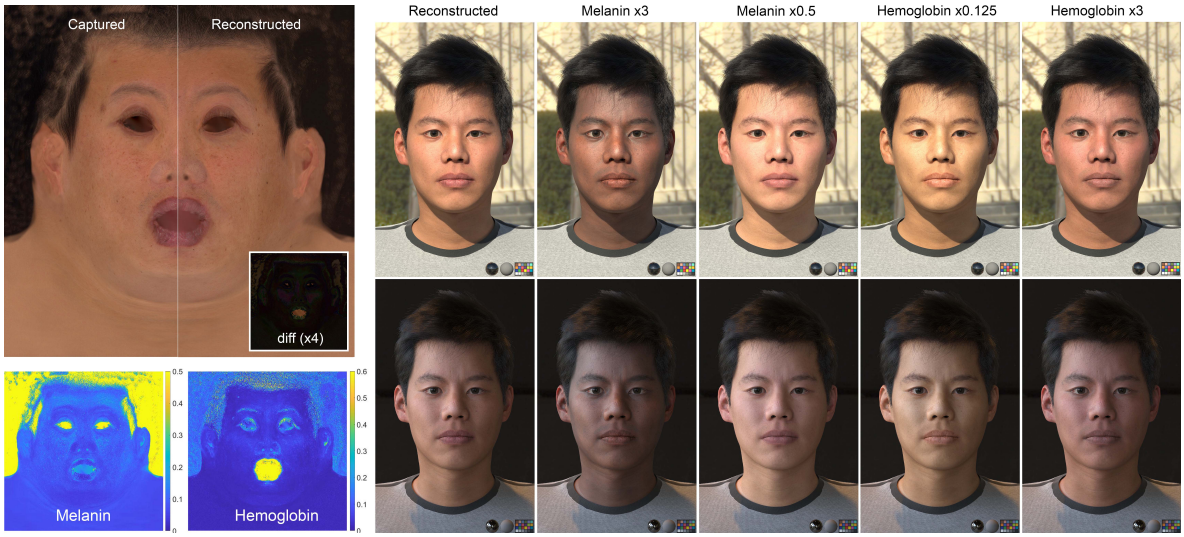


Figure 3.1: **Left**: Example estimation of the skin main chromophores (melanin and hemoglobin) from a captured RGB albedo. **Right:** The recovered skin parameter maps enable editing the skin appearance in a biophysically-constrained space. All images have been rendered using Meta's in-house physically-based path tracer, under different environment lighting.

The appearance of skin has been object of study in different fields, spanning from computer graphics and vision, to medicine or cosmetics, with numerous applications including photorealistic rendering, face recognition, dermatology diagnosis, photo-therapy or make up development. Within computer graphics, accurately modeling skin has received significant attention, given its importance on the realistic depiction of virtual humans. In particular, a wide variety of works have proposed different methods for modeling and rendering the complex translucent multilayered appearance of skin. This appearance is based on the interactions between light and the so-called chromophores, that determine the color of the skin

by selectively absorbing and scattering the incident electromagnetic spectrum. Throughout this chapter, we describe practical methods for modeling and capturing the spatially-varying color of skin based on spectral data. These methods build upon a biophysically-based manifold, that enables accurate estimation of the core components responsible for the colloration of human skin. Such manifold is a robust tool for capturing and editing the spectral skin diffuse reflectance (its color) in a physically-meaningful form, even from RGB captures, as in the example shown in Figure 3.1.

3.1 The Appearance of Skin

The human visual system is particularly sensitive at perceiving human faces; any subtle change on the skin appearance conveys rich information of a person's background and health, his or her habits, or the physical state at a particular moment. Such fine tuning of the human brain to faces makes them extremely challenging to render, requiring the upmost realism to avoid falling into the so-called *uncanny valley*.

However, the appearance of skin is the result of the combined effect of the optical properties of its internal structures and molecular components such as fibers or cells, as well as their intricate interaction with light (see Figure 3.2, right): When a beam of light hits the skin surface at normal incidence around 5% of its radiance is reflected by the surface, producing the achromatic Fresnel-based component of the skin reflectance. The remaining (95%) fraction of light penetrates inside the tissue and undergoes a series of scattering and absorption events until it emerges back on the surface, potentially in a different point than the point of entry. This process is generally called *subsurface scattering* (SSS) [**?**], is common in all translucent appearances, and is mainly responsible of the *diffuse surface albedo* of the skin, i.e. its color.

The physical process of subsurface scattering can be modeled using the *radiative transfer theory* [**?**], that statistically characterizes the amount of absorption and scattering events using the spectrally-resolved scattering and absorption coefficients $\mu_s(\lambda)$ and $\mu_a(\lambda)$ [m^{-1}]. The scattering coefficient $\mu_s(\lambda)$ represents the inverse of the average distance a photon can travel inside the medium without suffering a scattering event. The angular scattering distribution is represented by its phase function $f_r(\theta, \lambda)$ [sr^{-1}], which depends on the angle between incoming and outgoing directions θ; note that in the case of anisotropic scattering, it is common to characterize the medium using the so-called *reduced scattering coefficient* $\mu_s'(\lambda) = \mu_s(\lambda)\,(1-g)$ [m^{-1}], with $g(\lambda) = 2\pi \int_0^{\pi} f_r(\theta, \lambda)\cos(\theta)\sin(\theta)d\theta$. The absorption coefficient $\mu_a(\lambda)$, on the other hand, statistically describes how much light is lost due to absorption, and it is determined by the pigments in the skin (chromophores), chemicals that dissipate light in a wavelength-dependent manner, thus defining the color of the skin. In the following, we outline how such components are distributed within the skin, and how we model them.

The structure of skin Figure 3.2 (left) shows an illustration of a cross section of the structure of human skin. Human skin presents a multilayered structure, broadly composed of two main layers: epidermis and dermis. Depending on the body location, we can find a third, deeper layer (the hypodermis), mostly consisting of lipids. On top of the skin, there is generally a thin layer of sebum, responsible for part of the specular reflection of the skin.

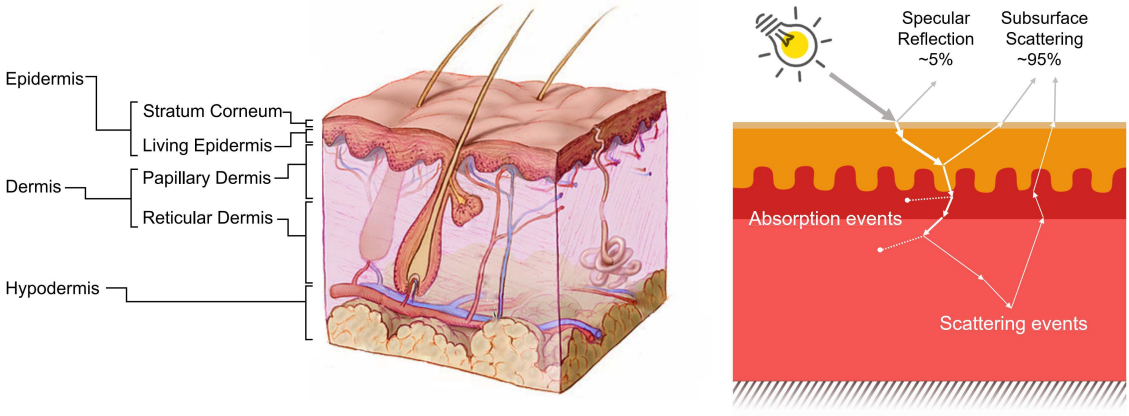


Figure 3.2: **Left:** Cross section of the human skin, showing the different layers that compose it (credits: Don Bliss, National Institutes of Health, US). **Right:** Diagram of the light interactions with skin. About 95% of the incident visible radiation penetrates inside the skin, and interacts with its components via absorption and scattering events. Light paths emerging back from the surface are responsible of the color of the skin.

The *epidermis* is the outermost layer of the skin, and can be divided in two sub-layers: the *stratum corneum*, which is composed of dead cells and keratin, and the living epidermis, which contains living cells like melanosomes responsible of synthesizing melanin. Melanin is the most important chromophore affecting the final coloration of the skin. The living epidermis can also contain carotenoids, with lesser impact on skin appearance, but sometimes noticeable depending on the diet, giving an extra orange tone to the skin.

The *dermis* is deeper inside the skin, and it is significantly thicker than the epidermis. Structurally, it includes blood vessels and most capillary veins. It can be subdivided in two sub-layers: the papillary dermis, which interfaces with the epidermis through the dermoepidermal junction, and the deeper reticular dermis, which contains bundles of collagen, fibers and veins. The main chromophore in the whole dermis is the blood's heamoglobin, which can be either oxygenated or deoxigenated and which gives its reddish color to the human skin. Other absorbers present in the epidermis are carotene, bilirubin, and water, but they have a lesser effect on the final appearance.

3.2 Modeling and Editing Skin Appearance: Why Spectral?

In production settings, in both the visual effects (VFX) and animation industries, defining the desired look for a character's skin typically requires the intervention of skilled artists. Obtaining the final color, translucency and specular reflectance involves fine tuning a large number of (textured) shading parameters. The diffuse reflectance resulting from SSS (the surface albedo) is usually described by a texture RGB image, either captured or authored by hand. Then, any required editing is performed through directly modifying such texture and re-rendering. Unfortunately, this means that non-physically plausible colors can be created, which not only makes the results potentially less realistic, but also might present problems of color constancy when observing the skin under new illumination setups. Or said otherwise, the captured or authored properties might not explain the actual process involving skin coloring.

Instead, our goal is to restrict the appearance of the skin (captured or manually authored) to physically-plausible colors based on the biophysical properties of skin. For that, we create a general physically-constrained manifold for skin albedo, where the wide variety of skin tones can be represented, beyond the relatively simple melanin-based space proposed by Fitzpatrick [1988]. This space is defined by the concentration of the different chromophores present in the skin, in the ranges found in medical literature. The goal is then to find a space that allows an as-bijective-as-possible mapping between the concentration of chromophores inside the skin and its surface albedo. As we show in Section 3.9, such space has several potential usages, like intuitive appearance editing through meaningful physical parameters, allowing to manipulate the spatial distributions of chromophores to e.g. vary features like moles and capillary veins; or as a physical prior to constraint optimizations in inverse rendering problems.

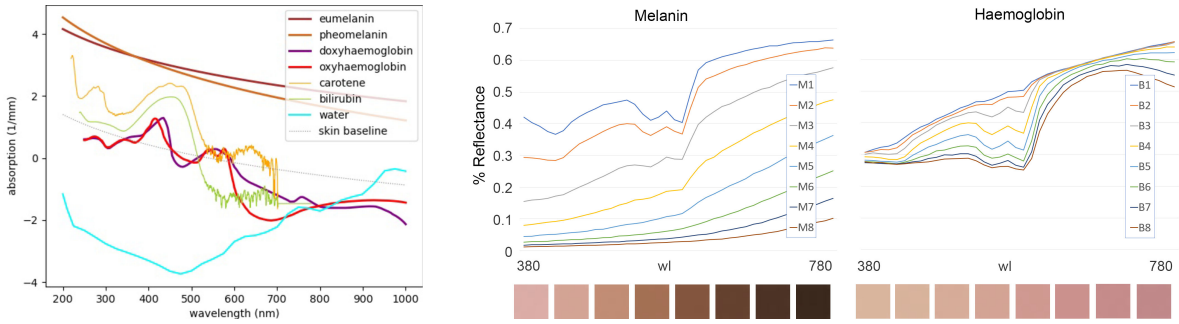


Figure 3.3: **Left:** Spectral absorption of main skin chromophores in log scale. Each of them present a particular spectral signature that allows estimation through spectroscopy. **Middle & Right:** Varying the concentration of melanin (middle) and haemoglobin (right) have a non linear effect on the resulting skin spectra, as shown in the spectral plots (top), and the corresponding skin tones in RGB (bottom). In the middle, melanin concentration is cubicly sampled, with Mn = (0.2, 0.7, 2.3, 5.4, 10.6, 18.2, 29.9, 43.1)%, while haemoglobin concentration is fixed at 2%. On the right, Haemoglobin concentration is quarticly sampled with Bn = (0.1, 0.2, 0.7, 2, 4.5, 9.6, 17.7, 30)%, while melanin concentration is fixed at 0.8%.

In order to build our space, we rely on the many previous works devoted to skin imaging Krishnaswamy and Baranoski [2004]; Baranoski and Krishnaswamy [2010]; Chen et al. [2015]; Iglesias-Guitian et al. [2015]; Alotaibi and Smith [2017]; Zherebtsov et al. [2019]; Gevaux et al. [2019, 2021]; Gitlina et al. [2020]. Skin reflectance spectra has been used to study its physical structure and chemical content, and reflectance spectroscopy[1] is a widespread technique for obtaining dermal information Zonios et al. [2001]: Measurements of the skin reflectance are obtained by illuminating the sample using a light source with calibrated spectral power distribution, and captured with a multiband digital imaging device (a spectrophotometer). The measured spectra can be then used to estimate distributions of skin chromophores. In other words, the goal is to explain the spectral reflectance through a set of basis functions: The spectral signatures of the chromophores non-linearly weighted by their concentration. Such concentrations vary in ranges measured in tissue optics and dermathology. Chromophore estimation techniques are well-known in tissue optics and medical imaging, and several commercial technologies have been developed, such DermaSpectrometer®,

[1] Spectroscopy is the study of the interaction of matter with electromagnetic radiation as a function of the frequency of the radiation

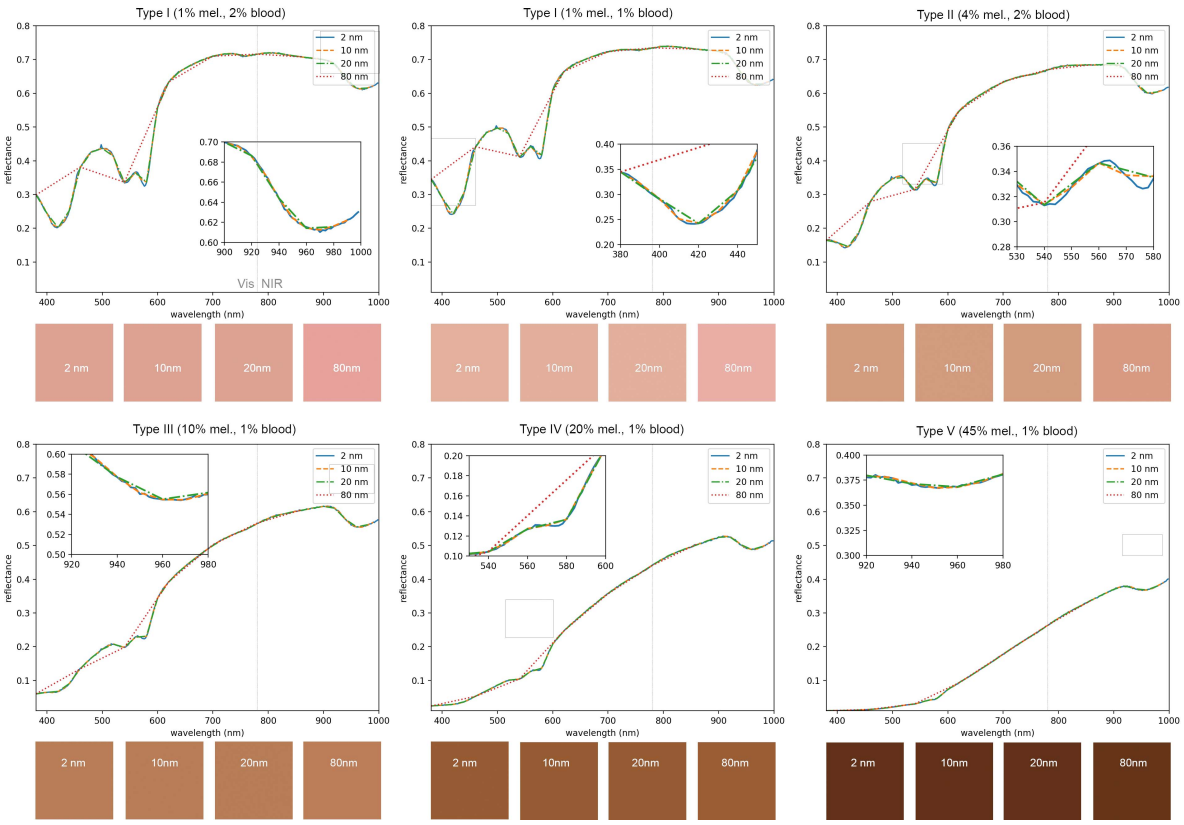


Figure 3.4: Spectral reflectances (top plots) and RGB albedos (bottom color squares) for 6 different skin types of increasing melanin concentration. Each skin has been computed using increasing spectral resolutions: 2nm (200 samples), 10nm (40 s), 20nm (20 s), and 80nm (5 s).

Mexameter®, Chromameter®, EMM-01, SIAScopy®, TIVI®, or many brands of over-the-counter fingertip oximeters that measure the oxygen saturation in blood.

Chromophore spectral signature The main assumption from reflectance spectroscopy is that each component in the tissue have a characteristic spectral signature that can be used to obtain their concentration from measurements. This is true, for the main components of the skin, as shown in Figure 3.3, left. These spectral curves determine the spectral absorption of each component (details in Section 3.3.2), which can be obtained either via spectral measurements or empirical fits [Donner and Jensen 2006].

As light suffers scattering and absorption events inside the tissue, that spectral signature manifest on the aggregated emerging spectrally-resolved reflectance, in a strongly non-linear manner, as shown in Figure 3.3, middle and right. In these plots, the bulk concentration of melanin (middle) and haemoglobin (right) increases linearly for each line, resulting in notorious changes on the spectral reflectance of the skin (the bottom color squares represent the RGB skin reflectance obtained as the concentration of melanin or haemoglobin increase). Note that such spectra are computed using the framework described in Sections 3.3 and 3.4.

Spectral resolution Skin reflectance is the result of a complex non-linear combination of its structure (e.g.: epidermal thickness) and the concentration of chomophores, that absorb

light spectrally, with potentially non-smooth spectral absorption (see e.g. oxyhaemoglobin in Figure 3.3, left). Thus, for accurate estimation of the appearance based on accurate absorption spectra from chromophores, a reasonably high-resolution sampling of the spectrum is needed.

Figure 3.4 illustrates the effect of spectral resolution on the final simulated appearance from the chromophores' spectral data, for different types of skin: Using a resolution of 10 nanometers (totalling 40 spectral bands) offers a good balance between error and computational time, showing similar results to $\times 5$ more spectral bands. We can also see that, even with visually similar appearances in RGB, the spectral reference presents strong differences between different types of skin. For light skin, the spectral reflectance presents a characteristic W-shape, that progressively flattens as melanin concentration increases. On the other hand, dark skin presents a flatter spectral profile and lower amplitude, which might result on larger visual errors even for small deviation on the parameters. This spectral footprint can be of enormous value for classifying the type of skin and its components more accurately.

3.3 Making a Practical Model for Skin Appearance

As shown above, the skin is a multilayered model or roughly parallel layers, each of them with different concentrations of chromophores as well as slightly different indices or refraction. Unfortunately, explicitly modeling all the layers complexity, even if we stochastically model the different components of the skin via radiative transfer theory, it is too expensive for our goal.

Building our physically-based reflectance manifold requires hundreds of thousands of light transport simulations, see Section 3.7, mapping the different optical parameters describing each individual type of skin with the final albedo data. For that, we perform several simplifications of the model, including a layer reduction, as well as a simplification of the layer interfaces. Also, this avoids the need to model certain properties of the skin that are difficult or impossible to measure in reality and have minimal effects in the reflectance after the many bounces light undergoes inside the skin, like the inner interfaces roughness.

In addition to reduce simulation times, we aim to reduce the number of parameters defining the model to a minimum expressive set. This has two main benefits: First, the edits are intuitive and not too cumbersome, and inverse rendering can find reasonable parameters avoiding extreme metamerism. In the following, we describe the skin model we use for developing our manifold.

3.3.1 Simplifying the Structure

In our model, we decided to reduce the number of layers form five (see Figure 3.2) to two main layers, epidermis and dermis. This assumption has been proven to be adequate in the past [Meglinski and Matcher 2002; Donner and Jensen 2006] and fits our purpose, though it might introduce a slight error in the overall translucency, as shown by Iglesias-Guitian et al. [2015].

To build the two-layers model, we simplify the epidermis by considering only the optical properties of the living epidermis, ignoring the *stratum corneum*. This is reasonable, since the *stratum corneum* has a minimal effect on the skin albedo due to low chromophore concentration and small thickness (5-20μm, depending on the body location [Czekalla et al. 2019]). The

main effect of the *stratum corneum* is on the specular component, which we model by using standard rough microfacet models [Walter et al. 2007]. The thickness of the whole epidermis is a parameter we vary (from 10 to 350 μm) to create our space (see Section 3.5 and Table 3.2).

The optical parameters of the dermis, on the other hand, are computed by weighted averaging the absorption and reduced scattering properties of the reticular and papillary dermis. We model it as a semin-infinite medium, omitting the subdermal layers, that vary from specimen and part of the body (e.g. adipose layers, muscles, cartilage...). We empirically found that a finite dermal thickness has a minimal effect over the resulting skin albedo. Also note that it is next to impossible to infer the exact local shape of these interfaces in vivo, except maybe via expensive MRI scanning.

3.3.2 Optical Properties

Following well-known multi-layered tissue optics models [Jacques 2013], we describe the optical properties of each layer by its spectral absorption $\mu_a(\lambda)$ and reduced scattering $\mu'_s(\lambda)$ coefficients, which can be directly used in any volumetric light transport simulator implementing the RTT. We compute both coefficients for each layer i, based on the chromophores concentration of each layer, as well as the spectral response of each of these chromophores. Importantly, we assume perfect decorrelation of such components.

Absorption The absorption coefficient $\mu_{a_i}(\lambda)$ of each layer i is the result of the additive contribution of each chromophore absorption $\mu_{a_c}(\lambda)$ present in each layer:

$$\mu_{a_i}(\lambda) = \sum_{c \in C_i} \mu_{a_c}(\lambda) = \sum_{c \in C_i} \frac{V_c p_c \varepsilon_c(\lambda)}{w_c} \tag{3.1}$$

where C_i is the set of chromophores in layer i; V_c is the volume fraction of the substance containing chromophore c [unitless]; p_c is the chromophore concentration [g l^{-3}]; ε_c is the molar extinction of the chromophore [m^2 mol^{-1} nm^{-1}]; and w_c is its molar weight [g mol^{-1}]. Table 3.1 shows plausible ranges for each value.

Following previous biophysical approaches [Krishnaswamy and Baranoski 2004; Iglesias-Guitian et al. 2015; Chen et al. 2015; Donner and Jensen 2006], the dominant chromophore in the epidermis is the melanin, characterized by the melanosomes volume fraction V_m, and the two different types of melanin (eumelanin $\mu_{a_{eu}}(\lambda)$, which is a broad absorber that tends to darken the skin, and pheomelanin $\mu_{a_{ph}}(\lambda)$), which absorbs more shorter wavelengths, resulting in a redish tint in e.g. the lips. The relative ration between eumelanin and pheomelanin is specified by the melanin type ratio φ_m, which varies through skin types.

For the dermis, the main chromophore is the haemoglobin present in blood in the dermis, with concentration V_b. There are two types of haemoglobin: oxygenated haemoglobin $\mu_{a_{hbO2}}(\lambda)$, responsible for the saturated reddish tint; and deoxygenated haemoglobin $\mu_{a_{hb}}(\lambda)$, responsible for a desaturated purple color. The relative ration between them is controled by the haemoglobin type ratio φ_h.

Other chromophores included in the model are the beta-carotene $\mu_{a_{\beta-c}}(\lambda)$ (both epidermis and dermis), and the bilirubin $\mu_{a_{bil}}(\lambda)$ in the dermis, contained in the blood and therefore more dominant in the dermis. Additionally, similar to previous models, we include a baseline of skin absorption in both epidermis and dermis, computed as $\mu_{a_{base}} = 7.84 \times 10^8 \, \lambda^{-3.255}(\lambda)$ [Saidi 1992] (λ in nanometers).

Parameter	Description	Value
$\mu_{a_{hbO2}}$	Oxy-Haemoglobin absorption	2.303 $\frac{p_{hb}\varepsilon_{hbO2}}{w_{hb}}$
$\mu_{a_{hb}}$	Deoxy-Haemoglobin absorption	2.303 $\frac{p_{hb}\varepsilon_{hb}}{w_{hb}}$
ε_{hbO2}	Oxy-Haemoglobin Extinction	Jacques [2013]
ε_{hb}	Deoxy-Haemoglobin Extinction	Jacques [2013]
p_{hb}	Haemoglobin Concentration	150
w_{hb}	Molar weight of Haemoglobin	64500
$\mu_{a_{eu}}$	Eumelanin absorption	$6.6x10^{11}\ \lambda^{-3.33}$
$\mu_{a_{pheo}}$	Pheomelanin absorption	$2.9x10^{15}\ \lambda^{-4.75}$
p_{bil}	Bilirubin Concentration	0.05
w_{bil}	Molar weight of bilirubin	584.66
$p_{\beta-c_e}$	β-carotene Concentration (Epidermis)	$2.1x10^{-4}$
$p_{\beta-c_d}$	β-carotene Concentration (Dermis)	$7x10^{-5}$
$w_{\beta-c}$	Molar weight of β-carotene	536.8726

Table 3.1: Chromophore specifications. The absorption coefficient is defined in cm^{-1}; the extinction coefficient ε_c in $\frac{cm^{-1}}{moles/liter}$; the concentration of the chromophore p_c in g/L; and the molar weight w_c in g/mol. The absorption of melanins $\mu_{a_{eu}}$ and $\mu_{a_{pheo}}$ is defined through a fit from Donner and Jensen [2006] to the measurements from Jacques and McAuliffe [1991] and Sarna and Swartz [2006] respectively. The 2.303 coefficient in $\mu_{a_{hb}}$ comes from deriving a factor of $ln\,(0)$, since ε has been historically recorded in such base 10 nomenclature from measurements of old spectrometers in literature. Finally, oxy and deoxy haemoglobin extinction can be found tabulated in Prahl and Jacques [[n.d.]].

Following Equation (3.1), the bulk spectral absorption of epidermis and dermis, $\mu_{a_e}(\lambda)$ and $\mu_{a_d}(\lambda)$ respectively, are defined as (we omit the λ-dependency for clarity)

$$\mu_{a_e} = V_m \left(\varphi_m \mu_{a_{eu}} + (1-\varphi_m)\,\mu_{a_{ph}}\right) + (1-V_m)\left(\mu_{a_{\beta-c}} + \mu_{a_{base}}\right), \tag{3.2}$$

$$\mu_{a_d} = V_b \left(\varphi_h \mu_{a_{hb}} + (1-\varphi_h)\mu_{a_{hbO2}} + \mu_{a_{bil}} + \mu_{a_{\beta-c}}\right) + (1-V_b)\mu_{a_{base}}. \tag{3.3}$$

Scattering. The reduced scattering coefficient $\mu_s'(\lambda)$ is modeled through a function of wavelength, fitted by Jacques [2013], which is a generic formula that applies on a wide range of human tissues as

$$\mu_s'(\lambda) = a\left(f_{Ray}\left(\frac{\lambda}{\lambda_r}\right)^{-4} + (1-f_{Ray})\left(\frac{\lambda}{\lambda_r}\right)^{-b_{Mie}}\right), \tag{3.4}$$

where the wavelength λ is normalized by a reference wavelength $\lambda_r = 500$ nm, a is a fitted scaling factor, f_{Ray} is the relative contribution of Rayleigh scattering, and b_{Mie} characterizes the wavelength dependence of the Mie scattering component. For both the dermis and epidermis, we use the fitted coefficients for human skin reported by Bashkatov et al. [2011], with a = 36.4, f_{Ray} = 0.48, and b_{Mie} = 0.22. In addition, we include the spectral dependency of the anisotropy factor $g(\lambda)$ measured by Van Gemert et al. [1989], which ranges from $g(380\text{nm}) = 0.73$ to $g(780\text{nm}) = 0.84$ at 780 nm. It is the same for both dermis and epidermis, and can be approximated as $g_{epidermis}(\lambda) \approx g_{dermis}(\lambda) \approx 0.62 + \lambda\ 0.29 \times 10^{-3}$ (λ in nanometers).

Index of refraction To improve convergence and sampling in simulation time, we remove the refractive interfaces between the two layers, so that both epidermis and dermis have an index of refraction (IOR) of 1.4. This IOR is the result from the weighted (by thickness) sum

Parameter	Description	Epidermis	Dermis
V_m	Melanin Volume Fraction	[0.001, 1]	-
V_b	Blood Volume Fraction	-	[0.001, 1]
t	Thickness [μm]	[10, 250]	2100
φ_m	Ratio of melanin types	[0.001, 1]	-
φ_h	Ratio of haemoglobin types	-	[0.001, 1]

Table 3.2: Ranges for the skin properties of our 5D albedo space.

of the IOR of each corresponding sub layers: *stratum corneum* (1.53), living epidermis (1.34), papillary dermis (1.39) and reticular dermis (1.395).

3.4 Light Transports in the Layered Model

We focus on resolving the light transports through the stacking of the different layers of the skin structure (Section 3.3.1). We conduct a random walk in 2D, assuming symmetry in the azimuthal planes of each layer. The interface between the epidermis and dermis only considers the change in scattering and absorption parameters, since the index of refraction is measured to be mostly the same in both layers Lister et al. [2012].

For each skin tone, we run spectral simulations for the wavelengths comprised in the visible range, between 380 to 780 nm. We empirically found that steps of 10 nm is enough to produce stable, noise free albedos. For each wavelength, a million photons are launched, simulating the 2D random walk over the two-layered, semi infinite medium, to produce diffuse albedos (by opposition to the rendering step, where we make use of such uv-mapped albedos to conduct true 3D random walks, as outlined in Section 3.8). We treat the skin as an homogeneous medium with exponential decay, since the more recent statistical theories attempting to account for spatial correlations in some materials Wrenninge et al. [2017]; Jarabo et al. [2018]; Bitterli et al. [2018]; d' Eon [2019] do not have corresponding measured data of particle distributions for tissue. We use the multiple scattering parameters listed previously, including the wavelength dependent anisotropy factor g from the Henyey-Greenstein Henyey and Greenstein [1941] phase function, expressed in its 2D form d'Eon [2021]. Note that the simulation starts once the photon has crossed the outermost interface and scatters diffusely into the tissue, namely choosing the initial direction from a Lambertian distribution around the inverse surface normal, in agreement with what we later use at rendering time (Section 3.8).

3.5 Albedo Manifold: Parametrization and Sampling

Our biophysically-based albedo space is created by varying the skin properties in the ranges listed in Table 3.2. Note that we consider epidermal thickness, which proved to be critical for parameter estimation Zherebtsov et al. [2019]. Accounting for this parameter over the face, along with varying values in melanin concentrations, helps us achieve local dark zones, such as moles, and generalizes over any skin type. We also allow melanin and haemoglobin to go beyond the usual values for human adults measured in the literature Meglinski and Matcher [2002], in order to automatically handle outliers found in the face, such as the lips, which exhibit very thin epidermis and higher blood concentration, or other cases like underlying

veins and capillary veins, or areas with abnormal melanin concentration like freckles or spots. This range expansion is also reasonable for the oxygenation level, since it can vary a lot depending on the physical state of the person, and for the melanin type ratio, where there is little agreement in the available measured data.

Thus, some parameters like bilirubin and β-carotene concentrations, remain fixed to common values of human skin measurements found in literature. The remaining 5D parameter space is sampled as follows: melanin and haemoglobin are respectively selected cubicly ($\sqrt[3]{V_m}$) and quarticly ($\sqrt[4]{V_b}$), to better adjust to their non linear effect on the albedo (see Figure 3.3); epidermal thickness, melanin type ratio and haemoglobin type ratio all are treated uniformly.

3.6 Mapping Albedos to Skin Properties

Our biophysical model defines the forward mapping from the skin parameters to the skin albedo. For the inverse process, we need to characterize the mapping from the skin albedo back to skin parameters. The non-bijective nature of such mapping, where many combinations of skin properties can lead to the same albedo, makes this task challenging.

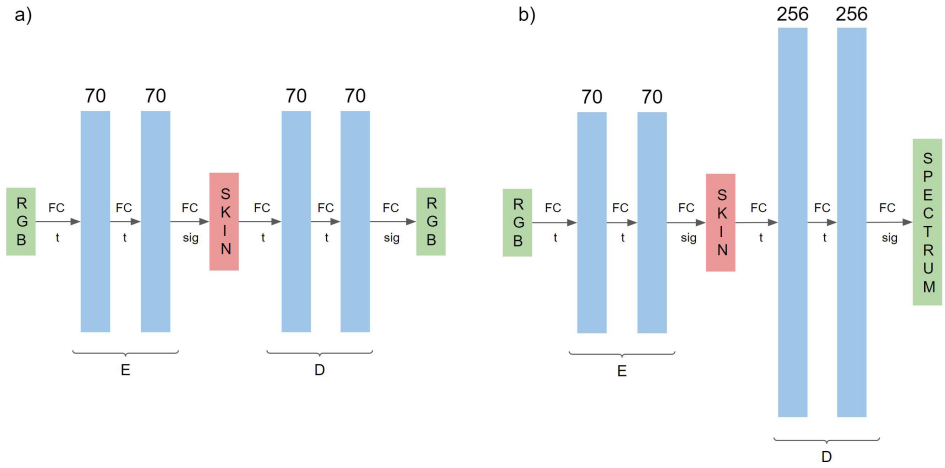


Figure 3.5: **Left:** encoder-decoder network to map albedos in RGB to skin parameters and the reverse. **Right:** exploiting spectral footprints in the decoder network to impose harder constrains in training.

We opted to train an encoder-decoder network to recover smooth skin parameter maps. The encoder maps a 3D skin albedo to a 5D skin parameter vector, while the decoder does the inverse. Figure 3.5 shows two of the several tested options for learning the mapping. The loss functions for the a) and b) networks are defined as follows:

$$\mathcal{L}_{rgb} = \mathcal{L}_p + L_1(\mathbf{D}(skin), rgb) + L_1(\mathbf{D}(\mathbf{E}(rgb)), rgb) \tag{3.5}$$

$$\mathcal{L}_{spectral} = \mathcal{L}_p + SAM(\mathbf{D}(skin), rgb) + L_1(\mathbf{downsample}(\mathbf{D}(\mathbf{E}(rgb))), rgb) \tag{3.6}$$

Both networks share the same skin parameter loss in the encoder **E**, defined as $\mathcal{L}_p = MSE(\mathbf{E}(rgb), skin)$. However, the loss for the decoder **D** and the full cycle loss **D**-**E** differ. For a) it is the L_1 loss of predicted vs ground truth rgb albedos. Instead, the network b) employs the spectral albedos in different ways. The full cycle downsamples the spectrum

to its corresponding rgb (see details in Section 3.7), and compares against the rgb albedo ground truth.

Figure 3.6: From top to bottom, captured albedos and reconstructions with our neural mapping RGB-skin-RGB (2nd row) and RGB-skin-spectrum (3rd row), for five subjects of different skin type (x4 absolute errors as insets).

The decoder **D** compares directly the predicted spectra against the ground truth spectral albedo, through *Spectral Angle Mapper* (SAM), which is a distance metric that measures the similarity of two given spectra, namely the angle between two n-D vectors. It is a well known metric often used in spectral sensing problems Dennison et al. [2004], like spectral unmixing Keshava and Mustard [2002] where a given spectrum is interpreted as the aggregation of several endmembers (materials with known spectral footprints). It has the advantage of being agnostic to the amplitude, thus only focuses on the spectral curve, i.e. the hue. In our tests, imposing such constraint only in the decoder part of the network consistently reduced the reconstruction errors. See Section 3.8 for further discussion.

3.7 Technical Details

Spectral Downsampling. The albedo space is generated through spectral computations, but input albedos are in RGB space. To downsample the multi-band spectral values into RGB, we use an existing integration approach Mallett and Yuksel [2019] (note that we could rely on other concurrent works Jakob and Hanika [2019] to this end). Most color spaces will work here (we use sRGB), considering that diffuse albedos have rather limited gamuts and dynamic ranges.

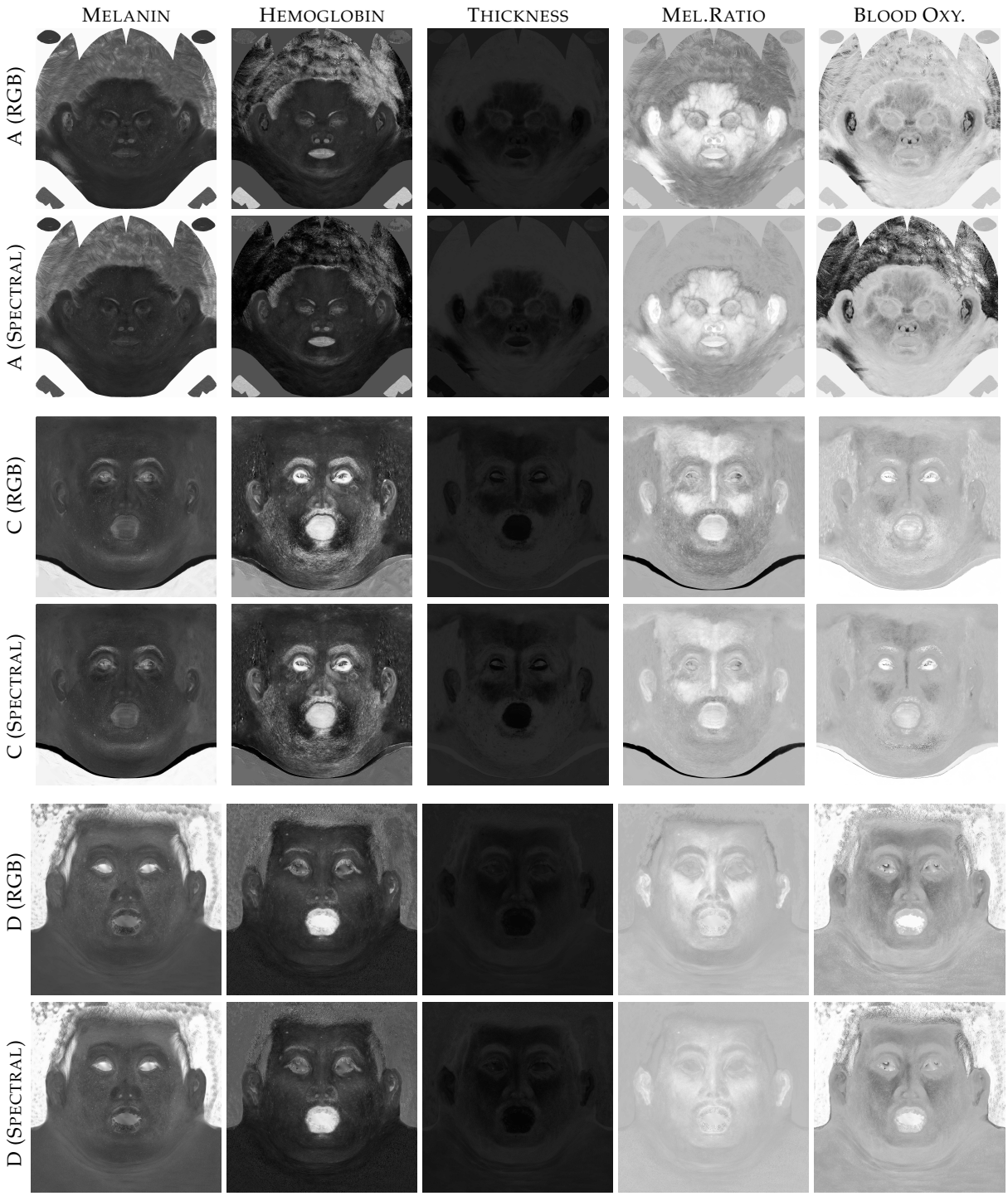


Figure 3.7: *Estimated skin parameters*. The overall melanin concentration is coherent to the skin type of each subject. Note how melanin spots are nicely isolated from small capillars, veins or reddish imperfections. Also, the color of the lips is mostly due to the high concentration of blood and the relatively thin epidermis in all cases.

Using Albedo Maps in Rendering. The final 3D lit geometries of our virtual faces are rendered using our own skin materials inside *Blender Cycles* Community [2020]. Aside from

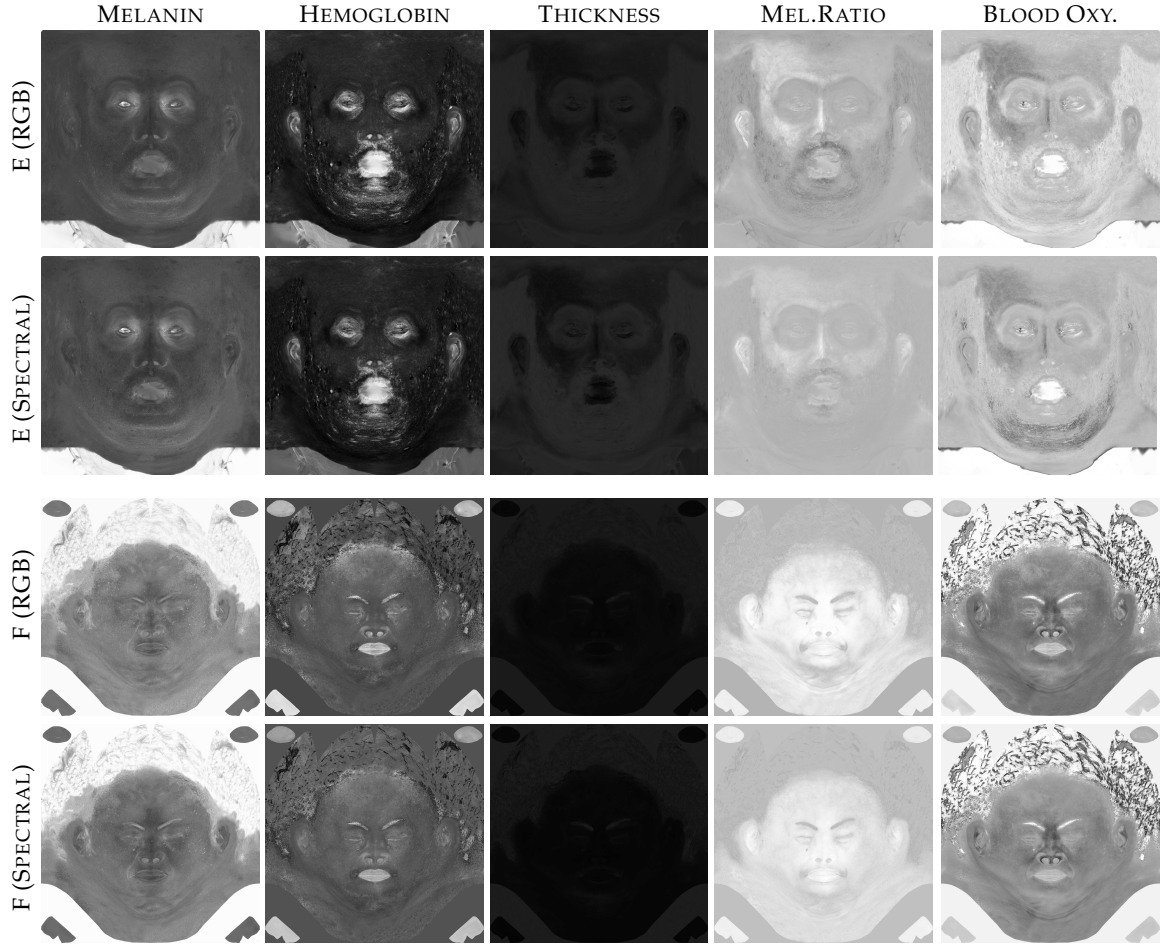


Figure 3.8: *Estimated skin parameters (II)*. For the extra subjects E and F, the results remain consistent.

the specular component of the skin, which we represent as a double lobe GGX Walter et al. [2007], we follow, in the spirit of a state of the art technique in production Wrenninge et al. [2017], a 3D random walk subsurface scattering solution that relies on a numerical albedo inversion around the mean free path and accounting for the anisotropy factor g. At this stage, we simplify the model to be single layered, dermis and epidermis combined, as a semi infinite medium. Obviously multi-layered models could be employed, for instance directly consuming the chromophores estimations, but these are out of scope of this paper. See Figures 3.1 and 3.10.

3.8 Estimating Skin Properties

We perform estimations and manipulations of skin parameters over several skin types covering the Fitzpatrick scale Fitzpatrick [1988]. In Figure 3.6 we show the reconstructed albedos using the RGB and Spectral solutions, and the corresponding estimated skin properties in Figures 3.7 and 3.8. The most interesting insight of the RGB vs spectral solutions is the improved recovery of the original skin tones, with less shifts in hue through the spectral mode (non skin areas, e.g. hair and beard, seem to be better preserved as well). This points to

the spectral metrics and the SAM loss as the right directions for more robustness. For more details about the training, please refer to our paper Aliaga et al. [2022].

As for the estimated parameters in Figures 3.7 and 3.8, the estimated values from RGB and spectrum solutions remain in similar ranges. However, melanin maps exhibit more contrast in the spectral version, better differing the spots vs normal skin. Also, there is less leaking of hemoglobin concentration into the melanin map (see ears), and the hemoglobin concentration maps are less noisy overall. The extra dimensions, melanin ratio and oxygenation level, do not need to reach such extreme values as in the RGB solution and yet produce better albedo reconstructions in the spectral mode.

3.9 Editing Skin Properties

We show how we can manipulate directly in this space of inferred skin properties, scaling some of them up or down in an intuitive and predictable manner. We run the neural decoder on these modified quantities to reconstruct biophysical albedos, and finally render them on 3D faces. For skin types ranging from I to V, we perform large edits in haemoglobin and melanin content, with details explained in Figure 3.10. Note the edits are naive in order to cover similar ranges for all skin types. Figure 3.9 shows edits over the rest of the parameters of the model. The level of blood oxygenation translates into paler or more saturated skin colors. The thinning of the epidermis, which typically occurs with aging, translates into a more translucent look, revealing the heterogeneities of the underlying layers (e.g. capillary and veins), while a thicker epidermis results in a more opaque and rough appearance. Last, we vary several components to simulate tanning and flushing. While it is hard to fully validate the correctness of the recovered skin properties, we find the parametrization adequate to produce plausible human skin albedos.

Figure 3.9: *Examples of parameter manipulation*. From left to right, a) original albedo, b) fully deoxygenated and c) oxygenated blood, epidermal thinning d) and thickening e) (minimum and maximum respectively), f) tanning (40% melanin increase, full pheomelanin), g) flushing (70% increase blood, fully oxygenated), and h) simulated vitiligo through edited melanin concentration.

3.10 Conclusion

We have presented a solution used at Meta to recover and robustly manipulate the biophysical properties of human skin from RGB albedo captures. Lifting from RGB to spectrums is not only necessary for accurately computing the diffuse albedo of the skin, but this can also be utilized at other stages of the inversion framework for constraining the optimization space, leading to better preservation of the original skin tones while estimating meaningful

Figure 3.10: Rendering results under our most neutral lighting scenario for edited skin parameters inferred by our model over different skin types (classified according to the Fitzpatrick scale). From left to right, original, followed by different manipulations per subject, three in blood concentration and three in melanin concentration. Note that these are straight algebraic edits on the recovered skin components, with no additional artistic tweaks or touch-ups involved. Though they demonstrate that our skin model describes an expressive latent space, these naive edits can result sometimes in semi non natural skins, and we suggest maybe more complex edits, some of them shown in Figure 3.9.

skin properties. As future avenues of work, we want to include variations of other extra chromophores that change more temporally, like the beta-carotene, bilirubin or any other dietary or health conditions. We also want to further validate the current method with hyper spectral captures in an end to end, *spectrum - skin properties - spectrum* learned mapping.

References

Carlos Aliaga, Christophe Hery, and Mengqi Xia. 2022. Estimation of Spectral Biophysical Skin Properties from Captured RGB Albedo.

Sarah Alotaibi and William AP Smith. 2017. A biophysical 3D morphable model of face appearance. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 824–832.

Gladimir VG Baranoski and Aravind Krishnaswamy. 2010. *Light and skin interactions: simulations for computer graphics applications*. Morgan Kaufmann.

Alexey N Bashkatov, Elina A Genina, and Valery V Tuchin. 2011. Optical properties of skin, subcutaneous, and muscle tissues: a review. *Journal of Innovative Optical Health Sciences* 4, 01 (2011), 9–38.

Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. 2018. A Radiative Transfer Framework for Non-Exponential Media. *ACM Trans. Graph.* 37, 6, Article 225 (Dec. 2018), 17 pages.

Tenn F Chen, Gladimir VG Baranoski, Bradley W Kimmel, and Erik Miranda. 2015. Hyperspectral modeling of skin appearance. *ACM Transactions on Graphics (TOG)* 34, 3 (2015), 1–14.

Blender Online Community. 2020. Blender: a 3D modelling and rendering package. (2020). http://www.blender.org

Carolin Czekalla, Karl Heinz Schönborn, Jürgen Lademann, and Martina C Meinke. 2019. Noninvasive determination of epidermal and stratum corneum thickness in vivo using two-photon microscopy and optical coherence tomography: Impact of body area, age, and gender. *Skin pharmacology and physiology* 32, 3 (2019), 142–150.

Philip E Dennison, Kerry Q Halligan, and Dar A Roberts. 2004. A comparison of error metrics and constraints for multiple endmember spectral mixture analysis and spectral angle mapper. *Remote Sensing of Environment* 93, 3 (2004), 359–367.

Craig Donner and Henrik Wann Jensen. 2006. A Spectral BSSRDF for Shading Human Skin. *Rendering techniques* 2006 (2006), 409–418.

Eugene d' Eon. 2019. A Reciprocal Formulation of Nonexponential Radiative Transfer. 2: Monte Carlo Estimation and Diffusion Approximation. *Journal of Computational and Theoretical Transport* 48, 6 (Sep 2019), 201–262.

Eugene d'Eon. 2021. A Hitchhiker's Guide to Multiple Scattering. (2021).

Thomas B Fitzpatrick. 1988. The validity and practicality of sun-reactive skin types I through VI. *Archives of dermatology* 124, 6 (1988), 869–871.

Lou Gevaux, Cyprien Adnet, Pierre Séroul, Raphael Clerc, Alain Trémeau, Jean Luc Perrot, and Mathieu Hébert. 2019. Three-dimensional maps of human skin properties on full face with shadows using 3-D hyperspectral imaging. *Journal of biomedical optics* 24, 6 (2019), 066002.

Lou Gevaux, Jordan Gierschendorf, Juliette Rengot, Marie Cherel, Pierre Séroul, Alex Nkengne, Julie Robic, Alain Trémeau, and Mathieu Hébert. 2021. Real-time skin chromophore estimation from hyperspectral images using a neural network. *Skin Research and Technology* 27, 2 (2021), 163–177.

Yuliya Gitlina, Giuseppe Claudio Guarnera, Daljit Singh Dhillon, Jan Hansen, Alexandros Lattas, Dinesh Pai, and Abhijeet Ghosh. 2020. Practical Measurement and Reconstruction of Spectral Skin Reflectance. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 75–89.

Louis G Henyey and Jesse L Greenstein. 1941. Diffuse radiation in the galaxy. *The Astrophysical Journal* 93 (1941), 70–83.

Jose A. Iglesias-Guitian, Carlos Aliaga, Adrian Jarabo, and Diego Gutierrez. 2015. A Biophysically-Based Model of the Optical Properties of Skin Aging. *Computer Graphics Forum (EUROGRAPHICS 2015)* 34, 2 (2015).

Steven L Jacques. 2013. Optical properties of biological tissues: a review. *Physics in Medicine & Biology* 58, 11 (2013), R37.

Steven L Jacques and Daniel J McAuliffe. 1991. The melanosome: threshold temperature for explosive vaporization and internal absorption coefficient during pulsed laser irradiation. *Photochemistry and photobiology* 53, 6 (1991), 769–775.

Wenzel Jakob and Johannes Hanika. 2019. A Low-Dimensional Function Space for Efficient Spectral Upsampling. *Computer Graphics Forum (Proceedings of Eurographics)* 38, 2 (March 2019).

Adrian Jarabo, Carlos Aliaga, and Diego Gutierrez. 2018. A radiative transfer framework for spatially-correlated materials. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Nirmal Keshava and John F Mustard. 2002. Spectral unmixing. *IEEE signal processing magazine* 19, 1 (2002), 44–57.

Aravind Krishnaswamy and Gladimir VG Baranoski. 2004. A biophysically-based spectral model of light interaction with human skin. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 331–340.

Tom Lister, Philip A. Wright, and Paul H. Chappell. 2012. Optical properties of human skin. *Journal of Biomedical Optics* 17, 9 (2012), 1 – 15.

Ian Mallett and Cem Yuksel. 2019. Spectral Primary Decomposition for Rendering with RGB Reflectance. In *Eurographics Symposium on Rendering (EGSR 2019)* (Strasbourg, France). The Eurographics Association.

Igor V Meglinski and Stephen J Matcher. 2002. Quantitative assessment of skin layers absorption and skin reflectance spectra simulation in the visible and near-infrared spectral regions. *Physiological measurement* 23, 4 (2002), 741.

Scott Prahl and Steven L Jacques. [n.d.]. Tabulated Molar Extinction Coefficient for Hemoglobin in Water. https://omlc.org/spectra/hemoglobin/summary.html.

Iyad Salam Saidi. 1992. *Transcutaneous optical measurement of hyperbilirubinemia in neonates*. Ph.D. Dissertation.

Tadeusz Sarna and Harold A Swartz. 2006. The physical properties of melanins. *The pigmentary system: physiology and pathophysiology* (2006), 311–341.

MJC Van Gemert, Steven L Jacques, HJCM Sterenborg, and WM Star. 1989. Skin optics. *IEEE Transactions on biomedical engineering* 36, 12 (1989), 1146–1154.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) *(EGSR'07)*. Eurographics Association, Goslar, DEU, 195–206.

Magnus Wrenninge, Ryusuke Villemin, and Christophe Hery. 2017. *Path traced subsurface scattering using anisotropic phase functions and non-exponential free flights*. Technical Report. Technical Memo.

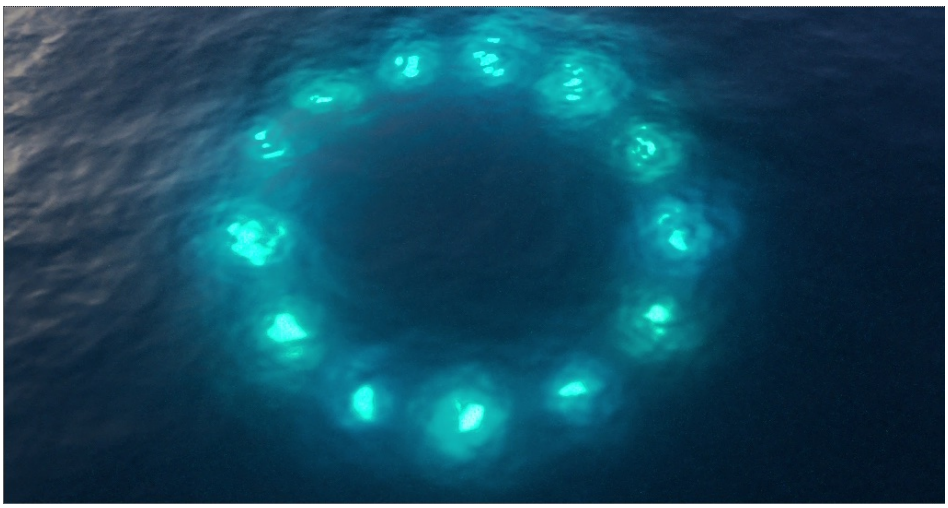
Evgeny Zherebtsov, Viktor Dremin, Alexey Popov, Alexander Doronin, Daria Kurakina, Mikhail Kirillin, Igor Meglinski, and Alexander Bykov. 2019. Hyperspectral imaging of human skin aided by artificial neural networks. *Biomedical optics express* 10, 7 (2019), 3545–3559.

George Zonios, Julie Bykowski, and Nikiforos Kollias. 2001. Skin melanin, hemoglobin, and light scattering properties can be quantitatively assessed in vivo using diffuse reflectance spectroscopy. *Journal of Investigative Dermatology* 117, 6 (2001), 1452–1457.

4

Sampling and Re-Sampling in Spectral Rendering

JEAN-MARIE AUBRY AND JIŘÍ VORBA, *Wētā Digital*



4.1 Introduction

A spectral renderer such as Wētā Digital's Manuka simultaneously integrates the radiance in multiple different wavelengths per path, thus efficiently exploiting SIMD architectures. Emissions and materials have a physically-based spectral models. Manuka is also a multi-technique ray-tracing imager, which employs Multiple Importance Sampling (MIS) to minimise sampling noise in each pixel. A denoising step completes the rendering process.

Advantages over RGB rendering are multiple in terms of photorealism and creative power; they have been amply covered in the previous edition of this course. Here we only recall the basics.

4.1.1 Spectral Techniques

In a camera, a pixel sensor (or a photoreceptor cell) integrates its response to incoming radiance not just in area but also in wavelength. This spectral radiance is the result of emitted radiance from a light source, that has bounced around in a scene before reaching the camera.

Figure 4.1: Spectral rendering

In order to simulate this physical process in a computer-generated image, forward path-tracing Quasi-Monte Carlo (QMC) samples are generated in a space that includes wavelength as a dimension. We will focus mostly on that scalar variable λ in this document but of course integration occurs on the other dimensions x as well (sensor position, lens position, time, path vertices...).

Because the path originates at the camera, in the light transport simulation the choice of the wavelength is bound to that emitter. To be optimal, that choice should thus reflect the response of the pixel (its "spectral importance"), as well as the incoming radiance.

A different light transport simulation technique could be of the light-tracing variety, for which the path originates at a light source. This situation is dual from the previous one: the choice of the wavelength should now depend on the emissive spectral radiance, but also of the spectral importance that has arrived there.

One important choice is the definition of the optimization criterion. It is an observed fact that perception of noise is mostly driven by camera luminance. Thus, the sampling techniques should aim to minimize the variance of that luminance. That is the goal of importance sampling.

4.1.2 Hero Wavelength Sampling

Since Wilkie et al. [2014], we know how to sample a single (hero) wavelength to drive the light path, while computing the radiance simultaneously in multiple (hero + supplementary) wavelengths, without bias. In theory, almost any probability distribution could be used to sample the hero wavelength; in practice however, only the uniform distribution is used for rendering. That is because noise performance has been observed to degrade sharply when moving from single to multiple wavelenghts, except when using the uniform sampling.

The reason for this is the following: according to Wilkie et al. [2014], supplementary wavelengths are translated (with periodisation over the range Λ of visible wavelenghts) of the hero wavelength by constant shifts multiples of $|\Lambda| / C$, C being the number of channels. This ensures stratification of the wavelengths across the spectrum, thus much reduced colour

noise. Note that if the hero wavelength is sampled uniformly, then so is each supplementary wavelength.

If one were to sample the hero wavelength with a better importance sampling than uniform, then the supplementary wavelengths would have distributions translated by the same shifts, which is a sure recipe for inefficient sampling (one of the wavelengths modes is bound to coincide with a region of low probability for the hero's importance). Thus, it is necessary to find a better spreading of the supplementary wavelengths. The solution exposed in § 4.2.1 is the most natural and the one that leads to the simplest form for the MIS coefficients. The basic idea can be traced back to West et al. [2020], however our development is different and does not require the notion of continuous multiple importance sampling. Estimating the wavelength importance from the first QMC samples is the next logical step and was published first by van de Ruit and Eisemann [2021].

4.1.3 RIS, ReSTIR, GRIS

We give here a very brief overview of the latest work on re-sampled importance sampling (RIS) Talbot et al. [2005] and related algorithms while our extension of spectral re-sampling can be found in Sec. 4.3. Note that in the following, we sometimes vaguely refer to re-sampling or RIS while we actually formulate our spectral re-sampling based estimator within the very recent GRIS (Generalized Resampled Importance Sampling) Lin et al. [2022] framework. To construct a RIS sampler, one needs to propose a set of candidate samples, compute a re-sampling weight for each of them as a ratio between the target function (in general un-normalized desired sampling density) and the actual density of proposing the sample, and subsequently re-sample a result proportionally to the re-sampling weights. With growing number of candidates, the distribution of re-sampled results asymptotically converges to a distribution proportional to the target function.

Relatively recently, Bitterli et al. [2020] turned attention of computer graphics community to RIS once again. They introduced reservoir sampling into computer graphics which enables progressive re-sampling from arbitrary number of candidate samples while keeping a small constant memory footprint. Moreover, they proposed reusing candidates from temporal and spatial domains to further increase the number of candidates which significantly increases the power of RIS. They have shown significant gain in convergence when they applied their so called ReSTIR improvements as described above to sampling direct illumination in the context of real-time rendering with a budget of one sample path per pixel. Ouyang et al. [2021]; Lin et al. [2021] extended this idea for sampling full light paths while employing reconnections for path reusing. Other works Boissé [2021]; Boksansky et al. [2021] introduced ReSTIR for world-space sample reuse. Finally, Lin et al. [2022] build further upon reconnecting full paths and improve the results mainly for specular light paths. Moreover, they provide solid framework for formulating re-sampling based estimators and thoroughly study convergence properties of the algorithm. To our best knowledge, no previous works does not provide details on application of re-sampling in spectral renderer.

4.2 Spectral Guiding

4.2.1 Non-Uniform Wavelength Sampling

In the following, we propose a different way of obtaining a range of supplementary wavelenghts, each following the *same* distribution as the hero wavelength and with similar stratification properties as the uniform translation.

4.2.1.1 Probability-Stratified Mapping

Let f be a probability density with respect to the Lebesgue measure, non-vanishing on a range $\Lambda := [\lambda_{\min}, \lambda_{\max}]$. Let $F : \lambda \mapsto \int_{\lambda_{\min}}^{\lambda} f(\tau)d\tau$ be the corresponding cumulative distribution function. For $k \in \mathbb{Z}$ define

$$\varphi_k(\lambda) := F^{-1}\left(\left[F(\lambda) + \frac{k}{C}\right]\right) \tag{4.1}$$

where the bracket above stands for periodisation in $[0, 1]$.

Observe that $\varphi_{k+C} = \varphi_k$, $\varphi_0 = \mathrm{Id}$ and $\varphi_k \varphi_j^{-1} = \varphi_{k-j}$ (one could say that $k \mapsto \varphi_k$ is a group homomorphism from $\mathbb{Z}/C\mathbb{Z}$). Furthermore, $\forall \lambda, k$, $\mathbb{P}_f([\varphi_{k+1}(\lambda) - \varphi_k(\lambda)]) = \frac{1}{C}$ (Fig. 4.2).

If f is continuous, then

$$\frac{d\varphi_k}{d\lambda}(\lambda) = f(\lambda)\left(F^{-1}\right)'\left(\left[F(\lambda) + \frac{k}{C}\right]\right) = \frac{f(\lambda)}{f(\varphi_k(\lambda))}$$

because for any $\xi \in [0, 1]$, $\left(F^{-1}\right)'(\xi) = \left(f\left(F^{-1}(\xi)\right)\right)^{-1}$. Symmetrically,

$$\frac{d\varphi_k^{-1}}{d\lambda}(\varphi_k(\lambda)) = \frac{f(\varphi_k(\lambda))}{f(\lambda)}$$

Now let $p_{\lambda,0}$ be another pdf, with which the wavelength λ^0 is sampled. Using φ_k above, we map the hero wavelength λ^0 to the k-th technique's wavelength $\lambda^k := \varphi_k(\lambda^0)$. The pdf of the latter is related to that of the former by

$$\begin{aligned} p_{\lambda,k}(\lambda^k) &= p_{\lambda,0}(\varphi_k^{-1}(\lambda^k))\frac{d\varphi_k^{-1}}{d\lambda}(\lambda^k) \\ &= p_{\lambda,0}(\lambda^0)\frac{f(\lambda^k)}{f(\lambda^0)} \end{aligned} \tag{4.2}$$

4.2.1.2 Equiprobable Mapping

In Wilkie et al. [2014], only uniform translation maps of the type $r_k : \lambda \mapsto \left[\lambda + \frac{k}{C}(\lambda_{\max} - \lambda_{\min})\right]$ were considered (here the bracket meaning periodisation in Λ). These are of the form (4.1) if f is taken to be the uniform density on Λ, in which case the ratio above is always equal to 1:

$$p_{\lambda,k}(\lambda^k) = p_{\lambda,0}(\lambda^0) \tag{4.3}$$

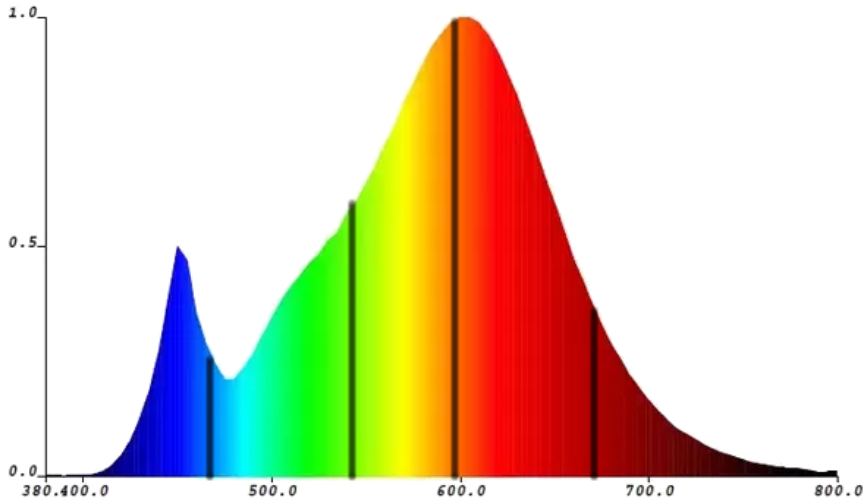


Figure 4.2: Stratification in probability: the area between two consecutive (periodized) wavelengths is constant ($\frac{1}{C} = \frac{1}{4}$)

The pdf for that technique's wavelength was thus obtained by (periodised) translation of the hero wavelength's pdf.

However, especially for non-uniform hero wavelength sampling, there may be better choices for φ_k. One simple possibility is to take $f := p_{\lambda,0}$ (*equiprobable mapping*). Consequently,

$$p_{\lambda,k}(\lambda^k) = p_{\lambda,0}(\lambda^k) \tag{4.4}$$

In contrast[1] to (4.3), the probability density for the technique's wavelength is the same (untranslated) as that of the hero wavelength.

4.2.1.3 Non-Uniform Hero Wavelength Sampling

Now, in addition, we consider that a path X is also sampled (the combination of sampling wavelengths then sampling X forms a *sampling technique*). In order to be able to vectorize computations, we assume as in Wilkie et al. [2014] that the sampling of X depends only on the hero wavelength λ^0. In other words, we have a conditional pdf $p_X(X|\lambda^0)$.

Let us call p_0 the probability density of the technique that consists in: first sampling λ^0 according to pdf $p_{\lambda,0}$, then sampling X (conditionally to λ^0) according to pdf $p_X(X|\lambda^0)$. By this description we have

$$p_0(X, \lambda^0) = p_X(X|\lambda^0)p_{\lambda,0}(\lambda^0)$$

Now imagine that we transform the result of such a sampling by applying φ_k to the wavelength component λ^0, and again write $\lambda^k := \varphi_k(\lambda^0)$ (importantly, the path X doesn't change). We call this the k-th wavelength sampling technique and note (X, λ^k) its result.

To compute the probability density of that new technique, we transport the initial pdf by the mapping $\Phi_k : (X, \lambda^0) \mapsto (X, \lambda^k)$. Since X is untouched, the Jacobian determinant of Φ_k

[1] The formulas coincide if and only if $p_{\lambda,0}$ was chosen to be uniform.

is the same as that of φ_k:

$$\det\left(\frac{d\Phi_k}{d(X,\lambda)}(X,\lambda^0)\right) = \frac{d\varphi_k}{d\lambda}(\lambda^0) = \frac{f(\lambda^0)}{f(\lambda^k)}$$

Then, in similar fashion to (4.2),

$$\begin{aligned} p_k(X,\lambda^k) &= p_0(X,\lambda^0)\det\left(\frac{d\Phi_k}{d(X,\lambda)}(X,\lambda^0)\right)^{-1} \\ &= p_X(X|\lambda^0)p_{\lambda,0}(\lambda^0)\frac{f(\lambda^k)}{f(\lambda^0)} \end{aligned}$$

If we use equiprobable mapping $f := p_{\lambda,0}$, then the combined probability density for this technique becomes, as in (4.4),

$$p_k(X,\lambda^k) = p_X(X|\lambda^0)p_{\lambda,0}(\lambda^k) \tag{4.5}$$

4.2.2 Multiple Importance Sampling

We now assume that equiprobable mapping is adopted to generate new wavelengths whenever a hero wavelength and its path are sampled.

4.2.2.1 Single Path Technique

Here we consider MIS between C wavelength techniques, all based on the same hero path but with different shifts $\varphi_i, i \in \mathbb{Z}/C\mathbb{Z}$. To compute the MIS weight of path (X,λ^k) produced by technique k, we need not only (4.5) but also the probability density, at the *same* path, of an other sampling technique (say, technique number i). A little manipulation is required for this.

Take (4.5) but make λ^0 an explicit function of λ^k:

$$p_k(X,\lambda^k) = p_X(X|\varphi_k^{-1}(\lambda^k))p_{\lambda,0}(\lambda^k)$$

Now the name λ^k doesn't play any specific role anymore, we can change it (temporarily) to λ

$$p_k(X,\lambda) = p_X(X|\varphi_k^{-1}(\lambda))p_{\lambda,0}(\lambda)$$

and since this is true for any $k \in \mathbb{Z}/C\mathbb{Z}$, we can also name the technique index i instead of k:

$$p_i(X,\lambda) = p_X(X|\varphi_i^{-1}(\lambda))p_{\lambda,0}(\lambda)$$

Now let's call the variable λ^k again:

$$p_i(X,\lambda^k) = p_X(X|\varphi_i^{-1}(\lambda^k))p_{\lambda,0}(\lambda^k) \tag{4.6}$$

From (4.6) we deduce (for example) the balanced heuristic MIS weights

$$\begin{aligned} w^k(X,\lambda^k) &= \frac{p_k(X,\lambda^k)}{\sum_{i=0}^{C-1} p_i(X,\lambda^k)} \\ &= \frac{p_X(X|\varphi_k^{-1}(\lambda^k))p_{\lambda,0}(\lambda^k)}{\sum_{i=0}^{C-1} p_X(X|\varphi_i^{-1}(\lambda^k))p_{\lambda,0}(\lambda^k)} \\ &= \frac{p_X(X|\lambda^0)}{\sum_{i=0}^{C-1} p_X(X|\lambda^{k-i})} \end{aligned}$$

or even simpler, by re-indexing the sum,

$$w^k(X,\lambda^k) = \frac{p_X(X|\lambda^0)}{\sum_{i=0}^{C-1} p_X(X|\lambda^i)} \tag{4.7}$$

Here, $p_X(X|\lambda^i)$ is to be read as the conditional pdf for X *as if* λ^i were a hero wavelength (in fact, it is the hero wavelength that would have generated λ^k in technique $k-i$). Note that the weight is the same for all wavelength techniques.

This should be compared to (10) in Wilkie et al. [2014].[2]

4.2.2.2 Multiple Path Techniques

We now consider $N \times C$ techniques: N different ways of sampling the path, times C options for sampling the wavelength. We index the path techniques with $l = 0,\dots,N-1$ and the wavelength techniques with $k = 0,\dots,C-1$. Again we impose the "hero wavelength" restriction, that when sampling any of the N paths (say path l), wavelength dependency can only be on $\lambda^{l,0}$ (no matter how the other wavelengths $\lambda^{l,k}$ are obtained from it).

Note that the wavelength mappings $\varphi_{l,k} : \lambda^{l,0} \mapsto \lambda^{l,k}$ can be different per path-technique; for instance if we use equiprobable mappings, the pdf $p_{\lambda,l}$ obviously depends on l. The resulting wavelengths $\lambda^{l,k}$ could be all different.

In balanced heuristic, the weight for technique (l,k) at sample $(X^l,\lambda^{l,k})$ is

$$\begin{aligned} w^{l,k}(X^l,\lambda^{l,k}) &= \frac{p_{X,l}(X^l|\varphi_{l,k}^{-1}(\lambda^{l,k}))p_{\lambda,l}(\lambda^{l,k})}{\sum_{j,i} p_{X,j}(X^l|\varphi_{j,i}^{-1}(\lambda^{l,k}))p_{\lambda,j}(\lambda^{l,k})} \\ &= \frac{p_{X,l}(X^l|\lambda^{l,0})p_{\lambda,l}(\lambda^{l,k})}{\sum_{j,i} p_{X,j}(X^l|\varphi_{j,i}^{-1}(\lambda^{l,k}))p_{\lambda,j}(\lambda^{l,k})} \qquad (4.8) \\ &= \frac{p_{X,l}(X^l|\lambda^{l,0})}{\sum_i p_{X,l}(X^l|\lambda^{l,i}) + \sum_{j\neq l} \frac{p_{\lambda,j}(\lambda^{l,k})}{p_{\lambda,l}(\lambda^{l,k})} \sum_i p_{X,j}(X^l|\varphi_{j,i}^{-1}(\lambda^{l,k}))} \qquad (4.9) \end{aligned}$$

after re-indexing the first sum in i. Unfortunately, there is no such re-indexing possible for the j-dependent sums in i. Unlike in (4.7), the wavelength pdf does not simplify out; also in general, unless $j = l$, $\varphi_{j,i}^{-1}(\lambda^{l,k})$ will not identify with one of the other $\lambda^{x,y}$. In form (4.9), the term with technique $j = l$ has been isolated in the denominator to emphasize the resemblance with (4.7).

4.2.2.3 Case of Common Wavelength Distribution

The situation simplifies considerably if all path-techniques use the same wavelength sampling distribution, $p_{\lambda,j} = p_\lambda$ implying $\varphi_{j,i} = \varphi_i$ for all i. In that particular case, after re-indexing and simplification, the balanced heuristic MIS weight boils down to

$$w^{l,k}(X^l,\lambda^{l,k}) = \frac{p_{X,l}(X^l|\lambda^{l,0})}{\sum_{j,i} p_{X,j}(X^l|\lambda^{l,i})} \tag{4.10}$$

[2] With our notation, that would read $w^k(X,\lambda^k) = \frac{p_X(X|\lambda^0)p_{\lambda,0}(\lambda^0)}{\sum_{i=0}^{C-1} p_X(X|\lambda^i)p_{\lambda,0}(\lambda^i)}$.

4.2.3 Wavelength Importance

The quantity whose variance we wish to minimize is the scalar product over λ of the camera luminance curve with the incoming radiance.

$$I = \int y(\lambda)F(\lambda)d\lambda \tag{4.11}$$

In general, in Monte Carlo rendering, integrated quantities are non-negative and in that case, the variance-minimizing distributions is just the integrand normalized (then the variance is zero). In some particular cases however, the integrand could be negative in places. For reasons beyond the scope of this course, this can happen to the camera luminance curve y that is used in the rendering computation. It well-known but still remarkable fact that in this case, the variance-minimizing distribution is the *absolute value* of the integrand normalized.

Theorem. *The Monte Carlo estimator $\hat{I} = \frac{g(X)}{p_X(X)}$ of $I = \int g(x)dx$ has a variance*

$$\operatorname{var}(\hat{I}) \geq \left(\int \left|g(x)\right| dx\right)^2 - \left(\int g(x)dx\right)^2 = \operatorname{var}(\hat{I}_0)$$

where $\hat{I}_0$ is the Monte Carlo estimator using probability density $\frac{\left|g(x)\right|}{\int\left|g(x)\right|dx}$.

Proof. First note that

$$\begin{aligned}\operatorname{var} I_0 &= \int \frac{g(x)^2}{\frac{\left|g(x)\right|}{\int\left|g(x)\right|dx}}dx - \left(\int g(x)dx\right)^2 \\ &= \left(\int \left|g(x)\right| dx\right)^2 - \left(\int g(x)dx\right)^2\end{aligned}$$

Now we want to bound from below $\operatorname{var}(I) = \int \frac{g(x)^2}{p_X(x)} - \left(\int g(x)p_X(x)dx\right)^2$.

Observe that

$$\left(\int \left|g(x)\right| dx\right)^2 = \left(\int \frac{\left|g(x)\right|}{\sqrt{p_X(x)}}\sqrt{p_X(x)}dx\right)^2$$

by Cauchy-Schwartz' inequality,

$$\begin{aligned}&\leq \int \frac{g(x)^2}{p_X(x)} \int p_X(x)dx \\ &\leq \int \frac{g(x)^2}{p_X(x)}\end{aligned}$$

Subtracting $\left(\int g(x)dx\right)^2$ to both sides yields: $\operatorname{var}(\hat{I}_0) \leq \operatorname{var}(\hat{I})$. □

As an application, knowing that $F \geq 0$ the optimal importance distribution for estimating (4.11) has density

$$p_{\text{opt}}(\lambda) = \frac{\left|y(\lambda)\right| F(\lambda)}{\int \left|y(\lambda)\right| F(\lambda)d\lambda} \tag{4.12}$$

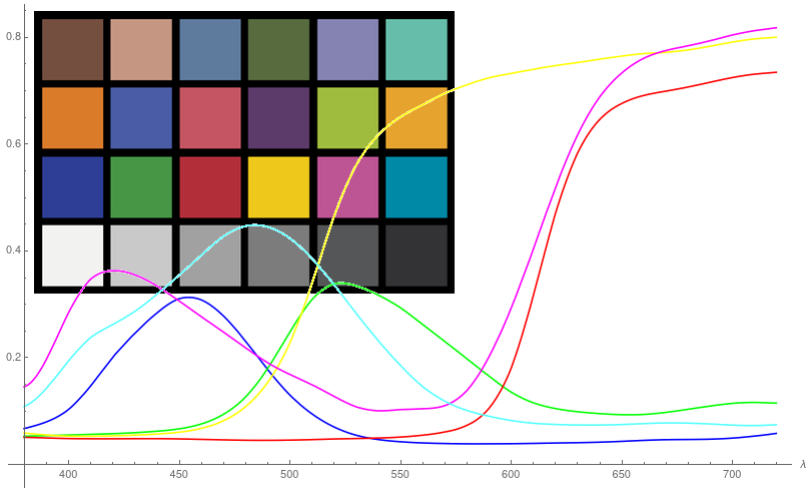


Figure 4.3: Macbeth chart and its second row's reflectance spectra

4.2.4 Histogram Estimation

To estimate p_{opt} in (4.12), we gather for each pixel a number of samples of that quantity. Following van de Ruit and Eisemann [2021], we apply a simple bilateral filtering while splatting and accumulate those splatting weights in a fixed-bins histograms (one per pixel). In test renders of natural scenes, where the wavelength distribution tends to be naturally smooth, a relatively low resolution for that histogram was found to be quite sufficient, with the advantage of sparing memory and obtaining a robust estimator of the distribution even with relatively few samples (the initial wavelengths must be sampled using a fixed wavelength distribution, for instance uniform).

4.2.5 Some Remarks

Correctness of the implementation can be asserted in multiple manners.

First, to ensure that wavelength importance is correctly estimated we render a reference material (for which the spectral reflectance is known) such as a Macbeth chart (Fig. 4.3), under some prescribed illuminant and using some camera's sensitivity curves and compare theoretical (4.12) to the renderer's estimation (Fig. 4.4).

Second, we remark that in (4.9), any error or inaccuracy in the computation of the MIS weights may be revealed in the render in different manners.

Across wavelengths, the techniques conditioned to the choice of λ are obviously biased (unless the render is truly monochromatic). However, with the proper combination weights, their sum (within any given path technique) is unbiased. Thus any error or imprecision in the computation of these weights will appear as a noticeable color bias.

On the other hand, between path techniques, each of them being unbiased, imprecise weights will not result in a bias but they may be sub-optimal in terms of variance reduction.

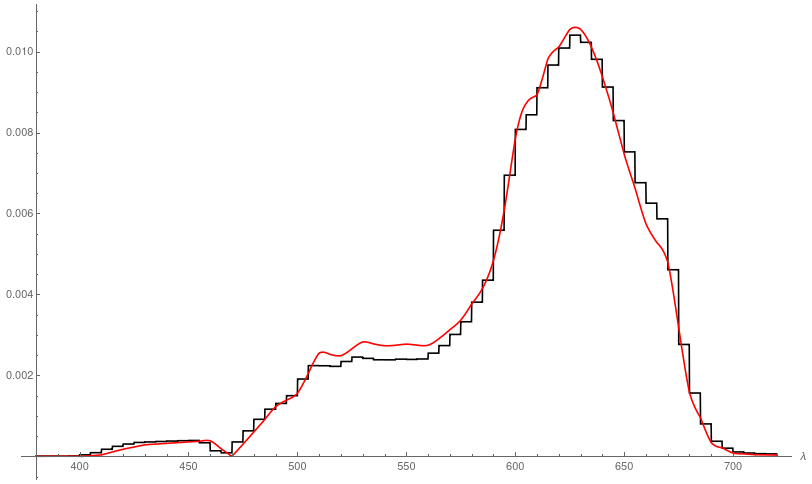


Figure 4.4: Piecewise-constant estimation of $p_{\text{opt}}(\lambda)$ for the red Macbeth patch

4.3 Spectral Re-Sampling

We share details of our experience with re-sampling Bitterli et al. [2020]; Lin et al. [2022] in the context of offline *spectral* rendering. Recently, re-sampling was studied for purposes of real-time graphics and the very latest work Lin et al. [2022] also suggested its application for offline rendering. However, previous works do not aim at its application to rendering high-fidelity images with a spectral renderer. We experimented with re-sampling of direct illumination, however, we believe that extension to full path space would be straightforward. Brief overview of related work was given in Sec. 4.1.3.

4.3.1 Direct Illumination

We integrate direct illumination L_o reflected from point $\mathbf{z}$ towards direction ω_o

$$L_o(\mathbf{z}, \omega_o) = \int_{\mathcal{M}} \rho(\mathbf{z}, \omega_o, \mathbf{z_e}) L_e(\mathbf{z}, \mathbf{z_e}) G(\mathbf{z}, \mathbf{z_e}) T(\mathbf{z}, \mathbf{z_e}) \mathrm{d}\mathbf{z_e}, \tag{4.13}$$

which is a product of BSDF ρ, emitted radiance from point $\mathbf{z_e}$ towards $\mathbf{z}$ and geometric term G and transmittance T between the respective points. To that end, we utilize two sets of samples. One is given by all emissive points $\mathbf{z}_\mathbf{e}^{\text{uni}}$ given by random collisions of reflected rays with emissive geometry or environment maps and their pdf is known. The other set comprises explicitly sampled next-event estimation points $\mathbf{z}_\mathbf{e}^{\text{nee}}$ which are produced by RIS (Resampled Importance Sampling) and have tractable unbiased contribution weights converging to their pdf reciprocal Lin et al. [2022]. We aim at combining both of these sets through multiple importance sampling (MIS). This is different from most of the previous works proposing one major re-sampling estimator where $\mathbf{z}_\mathbf{e}^{\text{uni}}$ and $\mathbf{z}_\mathbf{e}^{\text{nee}}$ samples would be used as candidates for re-sampling and would be weighted through MIS within RIS.

However, if we perceive our problem through prism of GRIS, we are looking at combination of two estimators with unbiased contribution weights while $\mathbf{z}_\mathbf{e}^{\text{uni}}$ samples do not involve re-sampling. Our major motivation for avoiding having one re-sampling estimator

above space of paths is exploration of lightweight utilization of re-sampling within existing offline production renderers without need of rewriting their integration core.

To utilize re-sampling with recently proposed sample reuse Bitterli et al. [2020] in Manuka, we also need to accommodate integration over the spectral domain.

4.3.2 Spectral Re-Sampling of Next-Event Estimation (NEE)

We first describe our way of generating $\mathbf{z}_\mathbf{e}^{\text{nee}}$ samples. This involves re-sampling a fresh sample from a set of candidate samples, then temporal and finally spatial re-sampling steps.

Fresh samples. We generate M candidate samples $\mathbf{x_i}$ on light each with pdf $\mathrm{p}(\mathbf{x_i})$ - this can be uniform light sampling or even proposed by a fast light hierarchy. We merge those samples into a reservoir r, effectively re-sampling a fresh sample $\mathbf{y_i}$. For this importance re-sampling, we determine the RIS weight $\mathrm{w}(\mathbf{x_i})$ of each fresh sample $\mathbf{x_i}$

$$\mathrm{w}(\mathbf{x_i}) = \frac{1}{M}\frac{\hat{\mathrm{p}}(\mathbf{x_i}|\mathbf{z_i}, \lambda^0)}{\mathrm{p}(\mathbf{x_i})} \tag{4.14}$$

simply as averaged ratios of target function for given candidate sample $\hat{\mathrm{p}}$ and probability density p of generating the candidate sample. Note that we do not employ spectral dependent sampling of the candidate samples. The target function

$$\hat{\mathrm{p}}(\mathbf{x_i}|\mathbf{z_i}, \lambda^0) = L_e(\mathbf{z_i}, \mathbf{x_i}|\lambda^0)G(\mathbf{z_i}, \mathbf{x_i}) \tag{4.15}$$

is evaluated at the hero-wavelength chosen for sampling the current path. We use $\mathbf{z_i}$ instead of $\mathbf{z}$ to explicitly express the domain in which we have sampled the fresh sample $\mathbf{y_i}$. In practice we generate only one or two candidate samples $\mathbf{x_i}$ to keep the total time per one sampled path reasonably low.

Temporal Re-Sampling. Instead of reusing samples across multiple frames, we reuse them over progressions which is akin to the temporal reuse in the context of real-time rendering Bitterli et al. [2020]. Here, one progression corresponds to tracing one path through each pixel while we typically need hundreds of progressions to converge light transport within one frame. After this temporal re-sampling step, we also reuse samples from neighbouring pixels through spatial re-sampling. We also retain previous samples in reservoirs associated with each pixel while reservoir $\mathrm{r}(\mathbf{z})$ corresponds to a pixel with a footprint containing the position $\mathbf{z}$.

Specifically, we start with temporal re-sampling so first, we merge the reservoir $\mathrm{r}(\mathbf{z_i}) \equiv \mathbf{y_j}$ which is given by the position of receiver $\mathbf{z_i}$ and holds the previous sample $\mathbf{y_j}$ into our current reservoir $\mathrm{r} \equiv \mathbf{y_i}$. Note that our time domain is limited to shutter open - shutter close interval due to motion blur. We determine the re-sampling weight of our fresh sample $\mathbf{y_i}$

$$\mathrm{w}(\mathbf{y_i}) = \mathrm{m}(\mathbf{y_i})\hat{\mathrm{p}}(\mathbf{y_i}|\mathbf{z_i}, \lambda_i^0)\mathrm{W}(\mathbf{y_i}) \tag{4.16}$$

in the spirit of GRIS Lin et al. [2022] where m is RIS MIS weight and W is the unbiased contribution weight corresponding to the sample $\mathbf{y_i}$. Note that we do not have any Jacobian in the equation because $\mathbf{y_i}$ is the fresh sample and was already sampled in the target domain $(\mathbf{z_i}, \lambda_i^0)$ given by the light receiver $\mathbf{z_i}$ and the current hero-wavelength λ_i^0. Note also that we use subscript i with the hero-wavelength λ^0 to distinguish between hero-wavelengths of different reservoirs which are sampled independently across pixels and rendering progressions.

In fact, the sampling domain is also given by normal at $\mathbf{z_i}$ and shutter time, which we need to respect in our implementation. However, to keep the notation concise, we stress here only the wavelength and receiver position explicitly.

In practice, Eq. (4.16) further simplifies to multiplication of the MIS weight m with sum of re-sampling weights used for generating the fresh sample $\mathbf{y_i}$ because its unbiased contribution weight

$$\mathrm{W}(\mathbf{y_i}) = \frac{\sum_k^M \mathrm{w}(\mathbf{x_k})}{\hat{\mathrm{p}}(\mathbf{y_i}|\mathbf{z_i}, \lambda_i^0)} \tag{4.17}$$

is equal to ratio of the re-sampling weights sum and target pdf of the fresh sample.

To acknowledge the fact, that sample $\mathbf{y_i}$ could be sampled from other sampling domain $(\mathbf{z_j}, \lambda_j^0)$, we choose to weigh it appropriately by generalized Talbot re-sampling MIS weight Lin et al. [2022]

$$\mathrm{m}(\mathbf{y_i}) = \frac{\hat{\mathrm{p}}(\mathbf{y_i}|\mathbf{z_i}, \lambda_i^0)}{\hat{\mathrm{p}}(\mathbf{y_i}|\mathbf{z_i}, \lambda_i^0) + \hat{\mathrm{p}}(\mathbf{y_i}|\mathbf{z_j}, \lambda_j^0) \left| \frac{\partial T_i^{-1}}{\partial \mathbf{z_i}, \lambda_i^0} \right|}, \tag{4.18}$$

where $\hat{\mathrm{p}}$ are target functions corresponding to the two possible sampling domains for sampling $\mathbf{y_i}$. Here, $T_i(\mathbf{z_j}, \lambda_j^0)$ is the shift mapping into the domain used for sampling our fresh sample $\mathbf{y_i}$. In our experimental implementation, we simply set the Jacobian of the shift mapping due to change of the hero-wavelength to 1.

Similarly, we need to define a re-sampling weight

$$\mathrm{w}(\mathbf{y_j}) = \mathrm{m}(\mathbf{y_j})\hat{\mathrm{p}}(\mathbf{y_j}|T_i(\mathbf{z_j}, \lambda_j^0)) \left| \frac{\partial T_i}{\partial \mathbf{z_j}, \lambda_j^0} \right| \mathrm{W}(\mathbf{y_j}) \tag{4.19}$$

for the previous sample $\mathbf{y_j}$ originally sampled from the domain $(\mathbf{z_j}, \lambda_j^0)$, but this time we reuse it in the current domain $(\mathbf{z_i}, \lambda_i^0)$. The corresponding unbiased contribution weight is analogous to Eq. (4.17) but the sum of previous re-sampling weights is divided by the target function $\hat{\mathrm{p}}(\mathbf{y_j}|\mathbf{z_j}, \lambda_j^0)$ based on the domain $(\mathbf{z_j}, \lambda_j^0)$.

Finally, we define the re-sampling MIS weight

$$\mathrm{m}(\mathbf{y_j}) = \frac{\hat{\mathrm{p}}(\mathbf{y_j}|\mathbf{z_j}, \lambda_j^0)}{\hat{\mathrm{p}}(\mathbf{y_j}|\mathbf{z_j}, \lambda_j^0) + \hat{\mathrm{p}}(\mathbf{y_j}|T_i(\mathbf{z_j}, \lambda_j^0)) \left| \frac{\partial T_i}{\partial \mathbf{z_j}, \lambda_j^0} \right|}, \tag{4.20}$$

for the previous sample $\mathbf{y_j}$.

Spatial Re-Sampling. Next, we randomly choose three neighbour reservoirs (pixels) from vicinity of the reservoir $\mathrm{r}(\mathbf{z_i})$. We merge them one-by-one with the current reservoir r where we already have the result of our temporal re-sampling. Finally, we replace the reservoir with previous sample $\mathrm{r}(\mathbf{z_i})$ with the current reservoir r and we take it as our new $\mathbf{z_e^{nee}} \equiv \mathrm{r}(\mathbf{z_i})$ sample. Note that technically, except for chosen sample $\mathbf{z_e^{nee}}$, we also need to store all the data related to the current sampling domain such as the hero-wavelength, position, normal, and time so that we can use the reservoir in the subsequent progression.

4.3.3 Weighing NEE Contributions

We estimate the final contribution as $f\mathrm{W}$, where $f = \rho L_e G T$ is product of BSDF, emitted radiance, geometry and transmittance terms. W is the unbiased contribution weight associated with the sample $\mathbf{z}_\mathbf{e}^{\mathrm{nee}}$ chosen by the process described in the previous section. It is important to note, that f is a vector function giving C contributions at wavelengths λ^k. The dependency of wavelengths λ^k on hero-wavelength λ^0 is given by Eq. (4.2).

We now consider that re-sampled NEE is now the only path sampling technique we have. Because we also have C wavelength sampling techniques, we need to weigh the resulting contributions appropriately by MIS contribution weight w. For simplicity, we assume *equiprobable mapping* of wavelengths [3] (see Sec. 4.2.1.2) which gives us Eq. (4.7)

$$w^k(X, \lambda^k) = \frac{p_X(X|\lambda^0)}{\sum_{j=0}^{C-1} p_X(X|\lambda^j)} \tag{4.21}$$

for weighing contributions of path X. Lets assume that the path probability $p_X = p(X_{\mathbf{z_i}}|\lambda^k)p(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)$ is product of probability $p(X_{\mathbf{z_i}}|\mathbf{z_i}, \lambda^k)$ of path X up to the receiver vertex at $\mathbf{z_i}$ and unknown probability $p(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)$ of sampling $\mathbf{z}_\mathbf{e}^{\mathrm{nee}}$ given the wavelength λ^k and receiver $\mathbf{z_i}$.

We don't know the probability $p(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)$, however we can approximate its reciprocal using the *unbiased contribution weight* W as proposed by Lin et al. [2022]; Bitterli et al. [2020]. Current reservoir $\mathrm{r}(\mathbf{z_i})$ gives us the unbiased contribution weight $\mathrm{W}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda_i^0)$ for sample at current hero-wavelength $\lambda^0 \equiv \lambda_i^0$. Using that weight, we can arrive at

$$p(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k) = \frac{1}{\mathrm{W}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda_i^0)\dfrac{\hat{\mathrm{p}}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda_i^0)}{\hat{\mathrm{p}}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)}} \tag{4.22}$$

$$= \frac{\hat{\mathrm{p}}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)}{\mathrm{W}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda_i^0)\hat{\mathrm{p}}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda_i^0)} \tag{4.23}$$

$$= \frac{\hat{\mathrm{p}}(\mathbf{z}_\mathbf{e}^{\mathrm{nee}}|\mathbf{z_i}, \lambda^k)}{\sum_q^M \mathrm{w}(\mathbf{x_q})}, \tag{4.24}$$

the unbiased approximation of the unknown sampling density by multiplying the weight by ratio of target function $\hat{\mathrm{p}}$ at the hero-wavelength λ_i^0 and wavelength λ^k in Eq. (4.22). Note that the denominator in the Eq. (4.24) is the sum of RIS weights which gets updated every progression, is not dependent on the wavelength, and serves as an unbiased estimate of the target function normalization factor.

It is interesting to note, that while in hero-wavelength sampling with classical MC estimator without re-sampling, we could make a sub-optimal choice and set the MIS contribution weight from Eq. (4.21) simply to constant C^{-1} where C is the number of wavelength sampling techniques (lanes in SIMD vector), we cannot do this in general for re-sampled estimator. We need to acknowledge the fact, that target function $\hat{\mathrm{p}}$ can be zero for some wavelengths and thus $\mathbf{z}_\mathbf{e}^{\mathrm{nee}}$ could not be sampled with such technique. Otherwise our estimator would converge to an incorrect result.

[3] For uniform density based mapping, we would need to multiply both nominator and denominator by appropriate $p_{\lambda,0}$ terms.

However, in classical MC estimator without re-sampling, we continue sampling the path if contribution at the hero-wavelength is zero as long as other lanes are non-zero. In other words, because pdfs are independent from contributions, sampling zero contribution in hero-wavelength still allows non-zero contribution in other wavelength channels. This is obviously not true for re-sampling with MIS weight using Eq. (4.24) as our pdf approximate.

4.3.4 Weighing Unidirectional and NEE Contributions

When we want to splat contributions from the samples $\mathbf{z_e^{uni}}$ given by random collisions of paths with light sources (aka. *unidirectional path sampling*) together with re-sampled next-event estimation samples $\mathbf{z_e^{nee}}$, we need to recognize existence of multiple path sampling techniques. To that end, we use MIS contribution weight given by Eq. (4.10)

$$w^{l,k}(X^l, \lambda^{l,k}) = \frac{p_{X,l}(X^l|\lambda^{l,0})}{\sum_j^{C-1} p_{X,\text{nee}}(X^l|\lambda^{l,j}) + \sum_j^{C-1} p_{X,\text{uni}}(X^l|\lambda^{l,j})}, \tag{4.25}$$

where index l stands for the path sampling technique, $p_{X,\text{nee}}$ is the probability of sampling path X using next-event estimation, $p_{X,\text{uni}}$ is the probability of sampling path X with unidirectional sampling, and index j iterates over C wavelength sampling techniques. Computing term $p_{X,\text{uni}}$ is straightforward as it does not differ from standard MC estimation without re-sampling. To determine term $p_{X,\text{nee}} = p(X_{\mathbf{z_i}}|\lambda^{l,k})p(\mathbf{z_e}^l|\mathbf{z_i}^l, \lambda^{l,k})$, we again express it as a product of probabilities $p(X_{\mathbf{z_i}}|\lambda^{l,k})$ of sampling the path up to the receiver vertex at position $\mathbf{z_i}$ and probability

$$p(\mathbf{z_e}^l|\mathbf{z_i}^l, \lambda^{l,k}) = \frac{\hat{\mathrm{p}}(\mathbf{z_e}^l|\mathbf{z_i}^l, \lambda^{l,k})}{\sum_q^M \mathrm{w}(\mathbf{x_q})}, \tag{4.26}$$

of sampling the last emissive path vertex at $\mathbf{z_e}^l$ given the receiver at $\mathbf{z_i}^l$ and wavelength $\lambda^{l,k}$. This is identical to Eq. (4.24). Note that the upper script l is used with points $\mathbf{z_e}$ and $\mathbf{z_i}$ to depict that they could have been sampled with either of the path sampling techniques. The sum in the denominator is the statistic stored with the reservoir $\mathrm{r}(\mathbf{z_i})$.

From a practical point of view, it is important to realize that such a formulation of contribution MIS weight is stochastic, that is the weight itself is a random variable exactly because of the denominator in Eq. (4.26). In early progressions when there are a few or no non-zero samples in the reservoir, the denominator can suffer from high variance. This can lead to unidirectional sampling receiving very high weight (or even weight equal to 1) which, in turn, can project the high variance into the combined estimator. For this reason, it might be useful to "warm-up" the estimator by executing first progressions without generating $\mathbf{z_e^{uni}}$ samples or alternatively without splatting any contributions at all.

4.3.5 Discussion

We did not include bsdf in the target function nor transmittance for simplicity as we mainly wanted to point out the spectral aspect of re-sampling. However, extension of target function by bsdf should be relatively straightforward. Including transmittance (or at least visibility) will be important for truly unbiased estimates in general. We also think that an extension to re-sampling of full paths in the spirit of GRIS and similar works while treating spectral

domain in the proposed way should be feasible. Note that this part of the course is report on our current work in progress, sharing lessons we learned, and thus our prototype described here has not been thoroughly evaluated on production scenes yet.

References

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). https://doi.org/10/gg8xc7

Guillaume Boissé. 2021. *World-space spatiotemporal reservoir reuse for ray-traced Global illumination*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3478512.3488613

Jakub Boksansky, Paula Jukarainen, and Chris Wyman. 2021. *Rendering Many Lights with Grid-Based Reservoirs*. Apress, Berkeley, CA, 351–365. https://doi.org/10.1007/978-1-4842-7185-8_23

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41, 4, Article 75 (07 2022), 23 pages. https://doi.org/10.1145/3528223.3530158

Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast Volume Rendering with Spatiotemporal Reservoir Resampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2021)* 40, 6, Article 279 (dec 2021), 18 pages. https://doi.org/10.1145/3478513.3480499

Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. *Computer Graphics Forum* 40 (12 2021), 17–29. https://doi.org/10.1111/cgf.14378

Justin Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. *Eurographics Symposium on Rendering*, 139–146. https://doi.org/10.2312/EGWR/EGSR05/139-146

M. van de Ruit and E. Eisemann. 2021. A multi-pass method for accelerated spectral sampling. *Computer Graphics Forum* 40, 7 (2021), 141–148. https://doi.org/10.1111/cgf.14408 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14408

Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. 2020. Continuous Multiple Importance Sampling. *ACM Trans. Graph.* 39, 4, Article 136 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392436

A Wilkie, S Nawaz, Marc Droske, A Weidlich, and J Hanika. 2014. Hero Wavelength Spectral Sampling. *Computer Graphics Forum* 33 (07 2014).

5

Grading Movies Using a Smooth Spectral Intermediate

RICHARD KIRK, *Scientist FilmLight* DANIELE SIRAGUSANO, *Image Engineer FilmLight*

5.1 Introduction

Some cameras can record a spectrum for each pixel. These are rare and expensive devices. They need a lot of light to work, as the light from each pixel is divided into many spectral bands. A complete spectrum workflow means serious compromises in resolution and frame rate, the lenses you can use, how you light a set, as well as resulting in larger files. This is not a good trade-off for motion pictures: you give up too much and get too little in return. Motion picture cameras will continue to capture RGB values.

Each reflecting object has its own reflection spectrum. We cannot guess this whole spectrum from a single set of RGB values, but there will be a unique smoothest spectrum that fits the RGB values. A set of smooth spectra can give us smooth and continuous RGB transforms, allowing us to combine spectral measurements with conventional three-channel workflows.

The techniques described in this lecture are no theoretical exercise but form the basis of transforms and looks used in high-end motion picture production on a daily basis around the globe.

5.2 Spectral Intermediate Pipeline

The idea is simple:

Let's use spectral calculation to derive an *RGB to RGB transform*.

Suppose motion pictures continue to use RGB cameras. They also use RGB displays. If they use spectra at all, they will use them in a *spectral intermediate pipeline (SIP)* connecting their RGB input and an RGB output.

A general *spectral intermediate pipeline* could combine spatial and colour operations, and model light fluxes within a scene. A special case of the *spectral intermediate pipeline* is a pure RGB to RGB colour transform. This fits in well with traditional motion picture workflows. They do not have to apply the transform to the whole image: they can use a matte to apply a colour transform only to some region within the shot that was lit differently. From now on,

we shall refer to a spectral intermediate pipeline that is a pure RGB to RGB colour transform as a *spectral intermediate pipeline*.

The first step is to allocate a reference spectrum to each RGB input colour.

We could interpolate from existing spectral data sets shown in Figure 5.1. We do not know the actual reflection spectrum for the object in our scene, so we guess that anything that is lemon yellow has the reflection spectrum of a lemon; anything that is tomato red has the reflection spectrum of a tomato; and so on. The spectral reflectance from our reference scene now follows two spectral calculation pipelines, called Input and Output Pipeline. Both pipelines are identical in their setup (Figure 5.2). We define a light source, a spectral filter, and an observer.

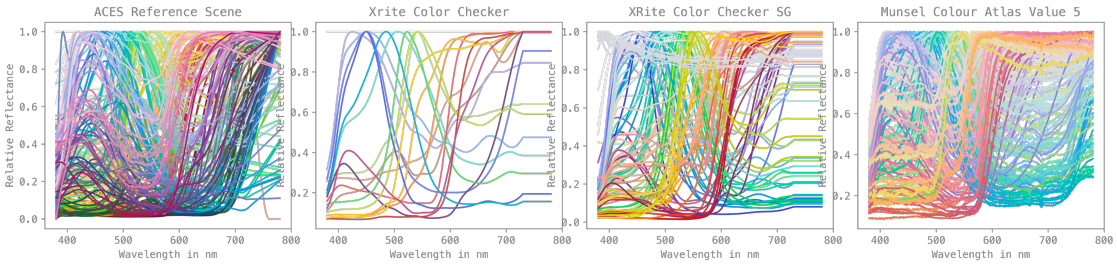


Figure 5.1: Four different reference scenes plotted as their spectral reflectance.

We can think of the reference scene as a proxy description of our scene. Sure, the spectral composition of the original scene might differ a lot. Still, maybe the proxy scene is close enough for certain use cases. The spectral reflectance from our reference scene now follows two spectral calculation pipelines, called Input and Output Pipeline. Both pipelines are identical in their setup. See Figure 5.2. We define a light source, a spectral filter, and an observer for each pipeline. Then we multiply the reference scene with the light, filter and observer, respectively. This operation produces an RGB tuple for each spectral sample in the reference scene.

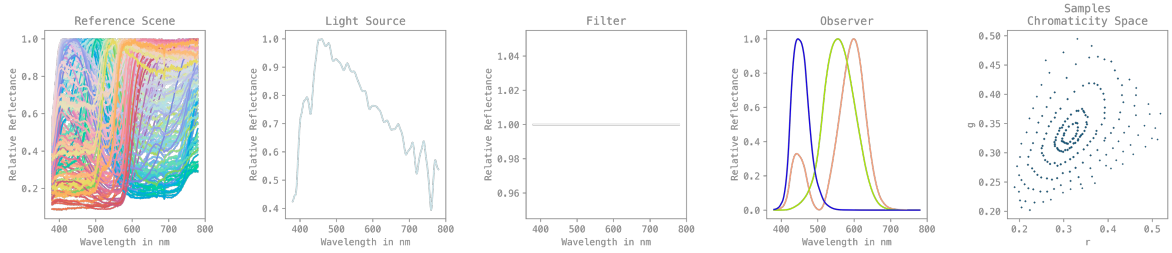
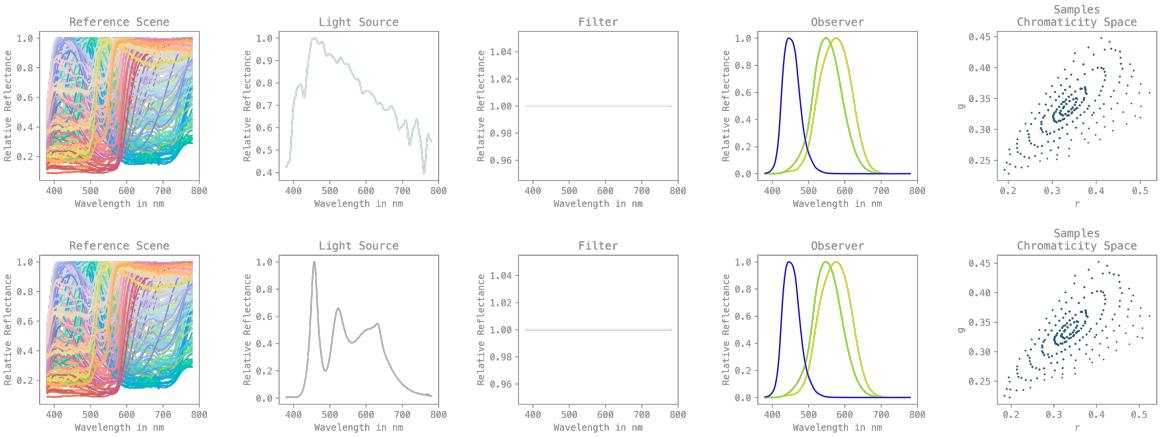


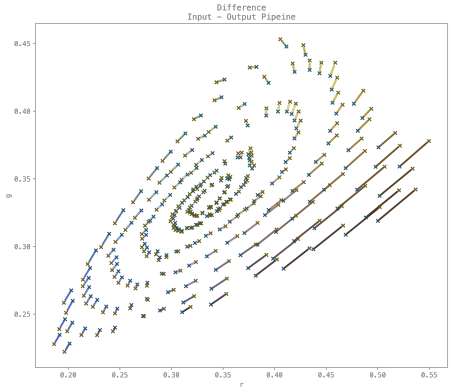
Figure 5.2: A single spectral pipeline for a given reference scene.

Because we calculate two pipelines, we will have two sets of RGB tuples. If both pipelines have identical settings, both sets of RGB tuples are identical too. The input pipeline should model the actual scene as closely as possible. The output pipeline then describes the desired output state. If we modify a single property in one of the two pipelines, the two RGB tuples

are different. See Figure 5.3. We will use these two sets of RGB samples to generate our non-linear *spectral intermediate transform*. See Figure 5.4. An overview of the proposed pipeline can be seen in Figure 5.5.



(a) Spectral Intermediate Dual Pipeline



(b) Resulting input and output RGB samples shown as a vector field in chromaticity spaces

Figure 5.3: Input and output pipeline and the resulting difference in chromaticity: All settings are identical besides the light source.

5.3 Input RGB State

Before applying the transformation to an image, our image needs to meet a few criteria.

Image State Modern cameras can record scene-referred data. If the dark levels are correctly set up and no non-linear transformations have been applied, the camera RGB values should

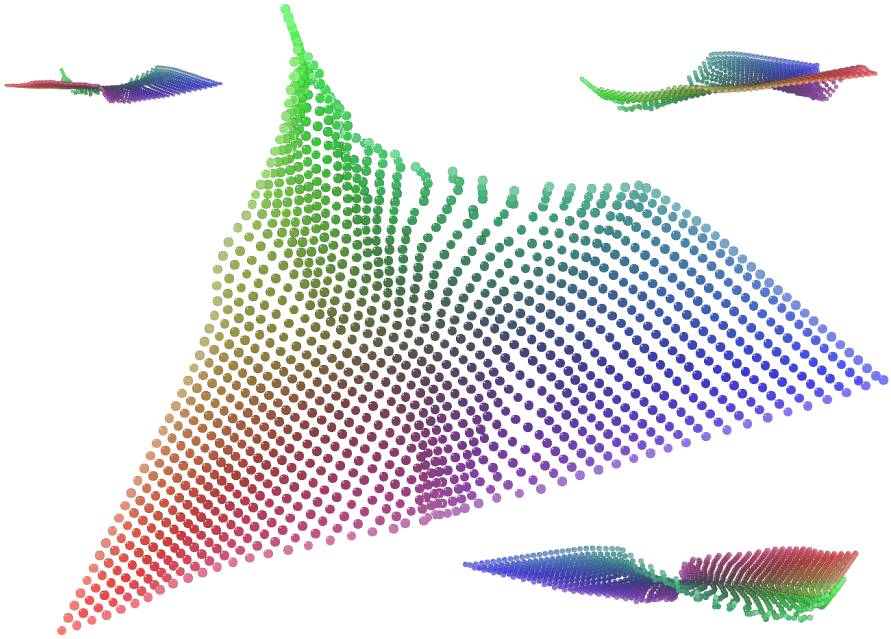


Figure 5.4: Spectral intermediate transform.

correspond to light levels hitting the camera sensor. This image state is crucial to marry the spectral calculation and the image data.

Linear Light The spectral calculation is applied in a scene-linear light domain. Professional cinema cameras often apply a per-channel pseudo log encoding, which we must undo before applying the *spectral intermediate transform*.

Chromatic Adaptation Cameras do not work well when one channel is much brighter than another. They give the best results when the average red, green, and blue signals are roughly the same. We could add a filter to balance the light levels, but the usual solution is for the camera to balance the gain of the RGB channels. We can model this in our spectral intermediate pipeline as a chromatic adaptation of the observer. However, often additional chromatic adaptation transforms are applied to camera data. Some camera manufacturers use a colour temperature-based correction matrix so that the RGB signal looks right on a monitor with a D65 white. Additional matrices might be applied when an image is imported into a colour management workflow with a different reference white point. These concatenations of matrices are challenging to undo or re-simulate in practice. Remember, we need to model the input state of the RGB data as closely as possible to produce a realistic end result.

Spectral Sensitivity An ideal camera matches the human eye, so it sees what we see. Real cameras do not match our eyes, but this is not a problem if we know the spectral sensitivity of the camera. Knowing this, we can reconstruct the RGB from the camera and our LMS or XYZ at our eye for each spectrum, and correct. In practice, camera spectral sensitivities are

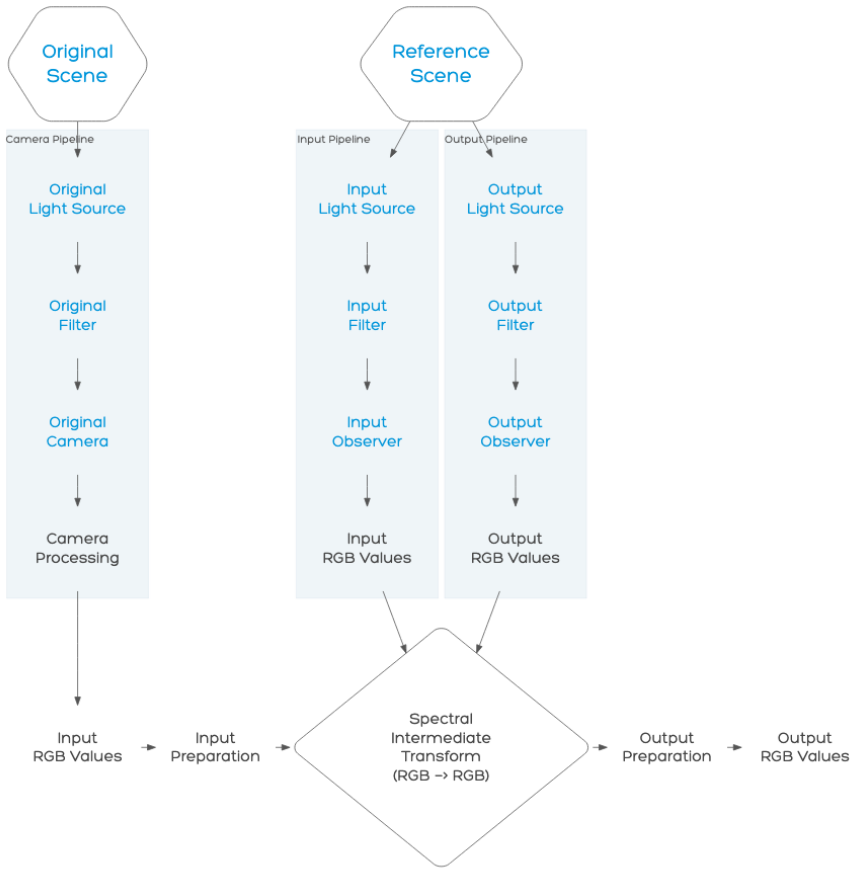


Figure 5.5: Spectral Intermediate Pipeline: Spectral processing flows orthogonal to RGB pipeline. Blue items operate in spectral domain. Grey items operate in RGB domain.

rarely known, hard to measure, and will depend on the lenses and filters used. We have characterised the lighting, camera sensitivity, and RGB processing pipeline for these experiments as best we can. We were more thorough than the motion picture industry usually is.

5.4 Reference Scene

The quality of our transform will depend on the choice of spectra in our reference scene.

Accuracy We would like a spectrum for each object the camera sees. This is not realistic, and we must settle for less. We can only have one spectrum for each input RGB if our transform output is to be unique. We might measure the spectral reflectance of each and every object in each scene we capture, but that is a lot of work.

Robustness In professional motion picture pipelines, robustness is the most critical property. Suppose we are transforming an image of a red car. The red should match in the

highlights and the shadows. If we measured many red objects, we might see coloured bands where the red went from, say, a chilli pepper red to a tomato red. This would look wrong, but it would also put a band into the image data which would be very hard to take out again. It is better to ignore the difference between these two reds, even if it is less accurate, and use a smoother transform.

Exposure Invariance Another valuable property we would like to achieve in our *spectral intermediate transform* is exposure invariance:

$$SIT\left(rgb\right) \cdot m = SIT\left(rgb \cdot m\right) \tag{5.1}$$

An exposure invariant transform simplifies the fitting process. Further exposure invariance allows us to cover an unbounded dynamic range, which is vital for dealing with scene-referred image data. It imposes a useful limitation on the reference scene: Our reference scene can only have a single sample for each chromaticity. If the reference scene contains multiple samples for a single or nearby chromaticity, one must weigh their contribution. This is a non-trivial task.

Gamut Coverage Every point in the colour space should lie close to some measured spectrum. If there are large missing regions in our colour gamut, then the interpolation may not be accurate. This is not easy: bright pinks or cyans are rare in nature. We also need to be careful close to the spectral locus where reflection colours are dim, particularly if we switch from reflection colours to light sources.

5.5 Smooth Spectra

So we want a set of reflectance spectra that are:

- realistic: spectra like real objects
- unique: one sample per chromaticity
- smooth: similar colours should have similar spectra
- uniform: no gaps in the gamut

If we look back at the collections of spectra in Figure 5.1, we can see some common patterns. There are smooth peaks, rounded edges, and flat regions. Instead of measuring lots of colours, can we generate a set of spectra with the properties we want?

There is one promising solution: for each colour as measured by camera RGB, or human vision XYZ or LMS, we can solve for the smoothest spectrum: the one that fits our measurements but also has the smallest sum of differences between adjacent wavebands. We do this via *constrained optimisation* using *Lagrange multipliers*. This is a standard technique, so we do not describe the mathematics in detail. Instead, we shall give a worked example. We start with the magenta patch on a small X-Rite Colorchecker chart. We measured the XYZ values for this patch.

$$\begin{pmatrix} X_M \\ Y_M \\ Z_M \end{pmatrix} = \begin{pmatrix} 0.311 \\ 0.197 \\ 0.284 \end{pmatrix} \tag{5.2}$$

We will calculate a reflectance spectrum with seven bands from R400 at 400nm to R700 at 700nm in steps of 50nm. A typical interpolation might have 201 bands going in steps of 2nm. We calculate the XYZ weights for the illuminant for each of these wavebands. We multiply the CIE 1931 XYZ weights by the illuminant light level for each waveband. The weights for the middle waveband are:

$$\begin{pmatrix} X_{550} \\ Y_{550} \\ Z_{550} \end{pmatrix} = \begin{pmatrix} 0.214 \\ 0.476 \\ 0.004 \end{pmatrix} \tag{5.3}$$

We then need to solve the matrix equation 5.4

$$\begin{pmatrix} 1 & \varepsilon-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.008 & 0.000 & 0.031 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0.183 & 0.020 & 0.835 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0.003 & 0.165 & 0.124 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0.214 & 0.476 & 0.004 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0.479 & 0.275 & 0.000 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0.115 & 0.042 & 0.000 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0.004 & 0.001 & 0.000 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon-1 & 1 & 0 & 0 & 0 \\ 0 & 0.008 & 0.183 & 0.003 & 0.214 & 0.479 & 0.115 & 0.004 & 0 & 0 & 0 & 0 \\ 0 & 0.000 & 0.020 & 0.165 & 0.476 & 0.275 & 0.042 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0.031 & 0.835 & 0.124 & 0.004 & 0.000 & 0.000 & 0.000 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} R350 \\ R400 \\ R450 \\ R500 \\ R550 \\ R600 \\ R650 \\ R700 \\ R750 \\ Rx \\ Ry \\ Rz \end{pmatrix} = \begin{pmatrix} 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.311 \\ 0.197 \\ 0.284 \end{pmatrix} \tag{5.4}$$

The XYZ weights are the non-zero values in the last three columns and the last three rows. The red 550nm line shows the 550nm values we listed earlier. Multiplying the last three rows by the column of reflection values R gives the XYZ values for the magenta patch: the last three values on the right-hand side. The lines with -1 2 -1 weights should get the reflectance values R400..R700 as close to the average values on either side without disturbing the XYZ totals. There are two extra rows corresponding to 350nm and 750nm. These wavelengths have no weight, so they do not affect the XYZ sums. These rows should make the spectrum smooth at either end. We want to solve for R400..R700. We do this by inverting the matrix. If we used 2nm intervals, we would be inverting a 201x201 matrix. However, there are a lot of zero values, so the inversion is usually quick and stable. The ε (epsilon) value is small: typically 1e-6. If ε is zero, the matrix cannot be inverted, so we use slightly different weights, but the result is almost identical. We ignore R350, R750, Rx, Ry and Rz.

5.5.1 Boundary conditions

Figure 5.6 shows the reflectance spectra for our magenta patch. The first line shows the actual patch reflectances as measured by a spectrometer. The graph is noisy on the left because the light source had little short-wavelength light. The smooth spectrum does not match particularly well, but this is not surprising. Our 7-point spectrum with 50 nm steps comes close to the smooth calculation with 2nm steps. Reflectance values should lie between 0.0 and 1.0.

These values are all between 0.0 and 1.0, but there is nothing in the calculation to force this to be so. The smooth spectrum can overshoot if you have a sharp step in reflectance.

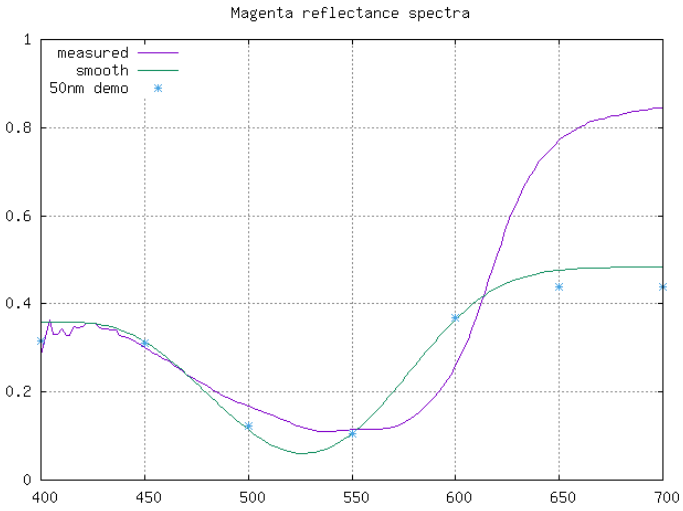


Figure 5.6: Measured sample vs smooth spectra.

Suppose the 550 nm value had come out negative. We return to the matrix and replace the 550nm line with one that forces R550 to zero. We replace the diagonal value with 1.0 and replace the other highlighted values with 0.0. If we have several negative values, we pick the most negative one and force that to zero, then repeat until all the values are positive or there is no solution. If R550 had been greater than 1.0, and we want to clip it, we do the same as before and change the value on the right to 1.0. We do not often fit reflectance to 1.0. We do not necessarily know the illumination at a pixel or even whether the object is emitting its own light. Instead of having a 3D set of smooth spectra for all colours and brightnesses, we often fit the colour and scale the brightness. We repeat these steps for any chromaticity within the spectral locus to generate the complete set. Figure 5.7 shows four standard datasets based on measured data and the smooth spectra set in comparison.

5.6 Illuminant transfer

We can use a *spectral intermediate transform* to simulate a change of illuminant. We set up the input pipeline to match the setup in the actual scene and change the illuminant spectrum in the output pipeline.

In professional motion picture photography, the illuminant is chosen with great care. The director of photography will most often choose the illuminant which produces the intended look for a particular scene. Normally this involves a lot of experience and test shoots before principal photography. A normal chromatic adaptation transform performed in the camera should, therefore, not change the characteristics of the illuminant. We can clearly demonstrate this if we shoot the same scene with different light sources and use standard chromatic adaptation to balance spectrally unselective objects. Figure 5.8 shows colour tem-

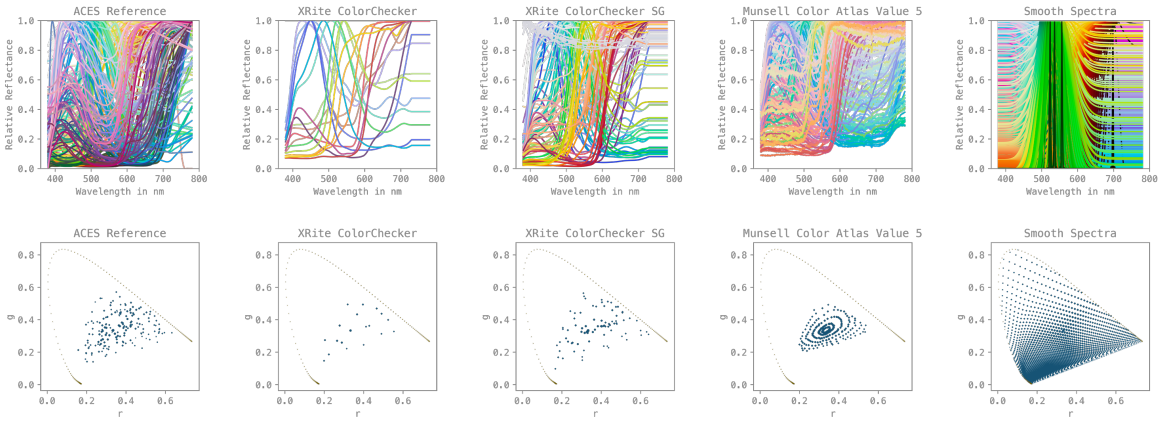


Figure 5.7: Different reference scenes and their resulting samples in CIE 1931 xy.

perature bracketing without white balancing in the camera. The light fixture in use has multiple LEDs. Figure 5.9 shows the same series after typical chromatic adaptation.

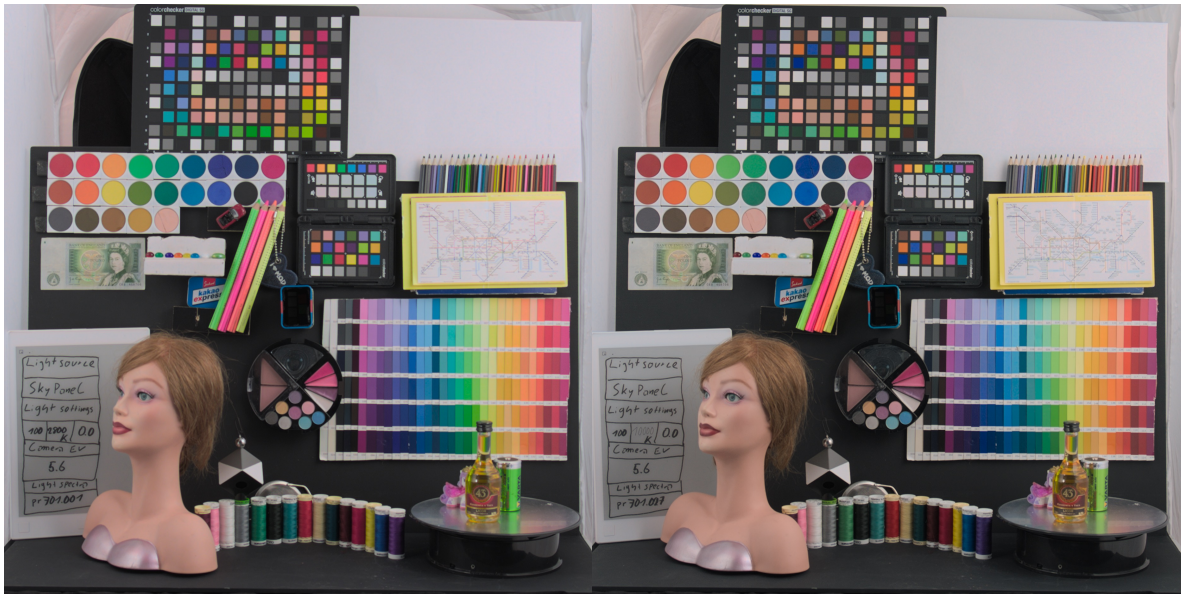
Figure 5.8: Colour temperature bracketing. The last two images are lit with the white LED only and RGB only, respectively.

Figure 5.9: Colour temperature bracketing. The last two images are lit with the white LED only and RGB only, respectively. All shots are balanced via a chromatic adaption transform using a 3x3 matrix.

At first glance, we get a good colour match. Still, we can find quite noticeable differences in colour rendition under more careful inspection. Notice the differences in rendition in the red, orange and purple water colours in Figure 5.10. We use the same light fixture to produce different illuminations in this example. But the different illuminations are composed from the same LED light sources within the light fixture. Hence the spectral difference between a 2800K and a 10000K illumination is not large. So the look difference is also small after the chromatic adaption. We would get a much stronger result if we used two light fixtures with different spectral emissions. To amplify this effect, we illuminated the same scene again, but this time only with the r,g,b LED lights (Figure 5.11(b)). We compare this to the 2800 Kelvin lit scene (Figure 5.11(a)). We can clearly see a great rendering difference in almost every spectrally selective object. Most notably, saturated colours are rendered much brighter and maybe even too bright for their saturation, giving especially red saturated colours a fluo-

rescent or brilliant look. This example also illustrates the challenges in virtual production where parts of the final illumination have a similar spectral composition.

As demonstrated, the illuminant is a powerful tool in our toolbox. In professional motion picture pipelines, we expect that the look of the illuminant is carefully chosen and carries the vision of the director of photography.



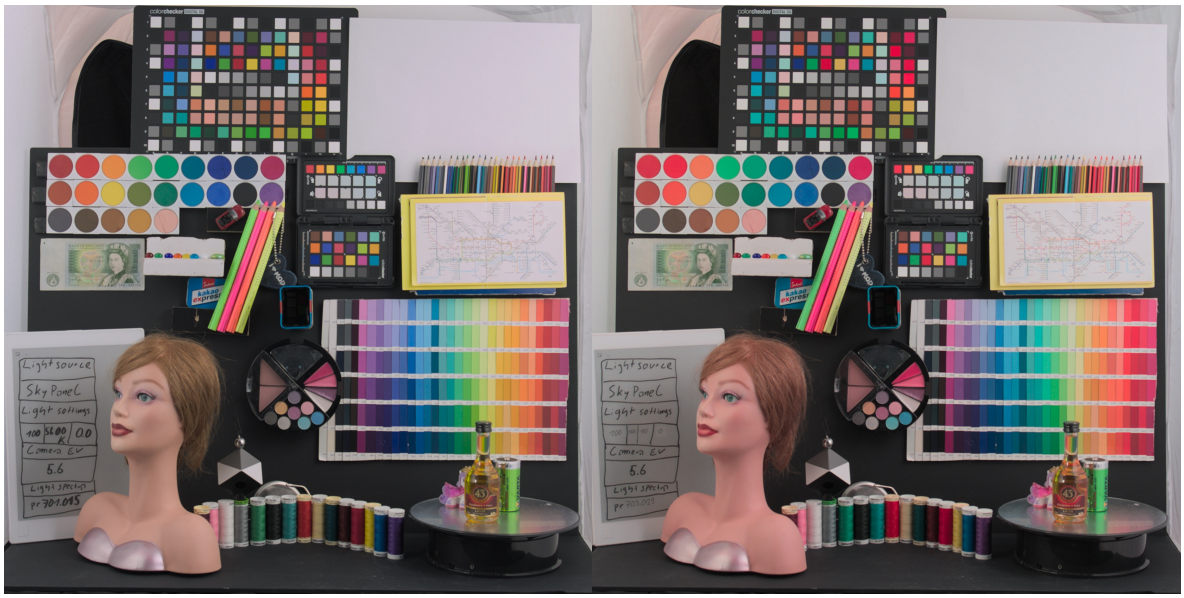
(a) Test scene lit with 2800K Shot (b) Test scene lit with 10000K Shot

Figure 5.10: 2800K and 10000K illuminated scene. Notice the differences in the red, orange and purple water colours.

Qualitative evaluation of illuminant transfer Sometimes though, there are exceptions to the *conservation of the illuminant look*. Sometimes it would be desirable to shoot with a certain illuminant from a practical standpoint while desiring the look of another one. Maybe it is more practical to use a certain light source because of power consumption or other features while still preferring the look of an unpractical or energy-consuming light source (e.g., shooting with LED lights while preferring the look of a tungsten light source). Another example would be simulating exotic illuminants. Extremely spiky illuminants like *high-pressure sodium lights* can produce unpredictable results on many digital cameras. It would be safe to shoot an image with a broader light source and, later in post-production, apply a *spectral intermediate transform* to mimic the look of *high-pressure sodium lights*. Figure 5.12 shows the *spectral intermediate pipeline* set up for such a use case.

The input pipeline is:

- scene: smooth spectra
- illuminant: D65
- filter: unity



(a) Test scene lit with 5600K (b) Test scene lit with RGB

Figure 5.11: 5600K and RGB only illuminated scene balanced with the camera SDK white balance. Notice the differences in almost all colours.

- Observer: von Kries LMS

The output pipeline would be:

- scene: smooth spectra
- illuminant: *high-pressure sodium lights*
- filter: unity
- Observer: von Kries LMS

In this setup, there are a few unsharp choices. Suppose we are building an *illuminant transfer* for a particular project. We will choose an input illuminant that comes close to the illuminant the director of photography uses on set. Most certainly, a night shot will not have a daylight illuminant. Also, we need to assume that the spectral sensitivities of the camera in use are not known. So we need to base our pipeline around another observer. By using the *von Kries observer*, we need to trust the camera manufacturer's camera profile for the conversion to the *CIE system*. We know this transform is not exact as no motion picture camera is actually colourimetric. We could have used the *XYZ Observer* instead of the *von Kries Observer*. But using the *von Kries observer* allows us to include the *partial adaptation to the illuminant* within the *spectral intermediate pipeline* as shown in Figure 5.13(c) and (d). Of course, we need to convert the image data to *linear von Kries LMS* before applying the *spectral intermediate transform*. This would be one example of *Input Preparation* shown in Figure 5.5. Figure 5.13 shows the *spectral intermediate transform* applied to an actual image. We are not aiming for an exact match between two scenes but a qualitative simulation of the look of an illuminant.

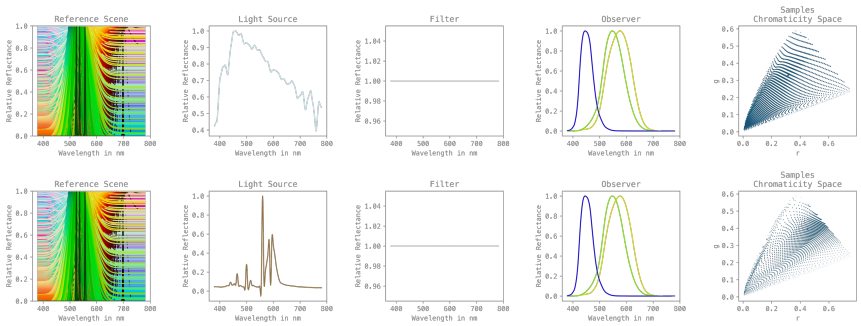


Figure 5.12: *Spectral Intermediate Pipeline* setup for illuminant transfer from D65 to high-pressure sodium light.

(a) Original Scene (b) Sodium Look (c) Sodium Look with 25% adaptation (d) Sodium Look with 50 % adaptation

Figure 5.13: Illuminant transfer series for high-pressure sodium lights with various chromatic adaptation factors.

Quantitative evaluation of illuminant transfer The sodium light example shown in Figure 5.13 is quantitative because we do not know what the scene would have looked like under high-pressure sodium lights. Many other attributes would have changed, like the light diffusion, and the captured images would have been much noisier. Let us examine if we can match one illuminant to the other with an objective goal to further test this technology. For this, we use the same test scene previously used. Let's try to match the 2800K, 10000K, *White only*, and *RGB only* shots to the 5600K shot. For all shots, we have the illuminant spectrum, the camera's spectral sensitivities, and access to the camera RGB linear light data. We performed this exercise twice. The first time we used the X-Rite Colorchecker SG reference scene. This might be very close to the scene spectra in the scene (we have an X-Rite Colorchecker SG chart in the upper part of the image). This should work very well. Figure

5.14 shows the result of the series. All shots besides the RGB only (5.14(e)) have very similar colours. We can also see a great improvement for (5.14(e)) especially if we compare how differently the illuminant renders the colours in Figure 5.11(b), but we can still see colour differences. Figure 5.15 shows the results of the same experiment but using smooth spectra as a reference scene. We can see the match is also OK. But we see larger differences, especially in the red and purple areas. So for this particular scene, one might conclude that the X-Rite Colorchecker scene spectra might be more accurate. But let's examine the robustness of the two *spectral intermediate transforms*. We can visualise the smoothness of a transform by adding noise to the source image and looking at the resulting *uv chromaticity diagram*, see Figure 5.16. We can clearly see folds in the Colorchecker SG based transform 5.16(a). In comparison, the *uv chromaticity diagram* for the smooth spectra transform looks very smooth, as expected (Figure 5.16(b)).

We trade-off colour accuracy using smooth spectra but gain robustness through smooth results. We believe this is wise for several reasons:

- We get hardly any spectral data from real productions.
- Collecting spectral data is difficult and can be imprecise.
- Spectral measurements are not representative over different shots or scenes.
- Other tools are much faster and more flexible for explicit shot matching.

The proposed *smooth spectra based illuminant transfer* gets most things right and puts the shot into the right ballpark while guaranteeing a smooth *spectral intermediate transform*. We do not expect to predict an exact match. Instead, the resulting *spectral intermediate transform* can be generally applied to many scenes and still convey the look of an illuminant. This makes the resulting transform useful in modern high-quality motion picture pipelines.

(a) 2800K (b) 5600K (c) 10000K (d) White Only (e) RGB Only

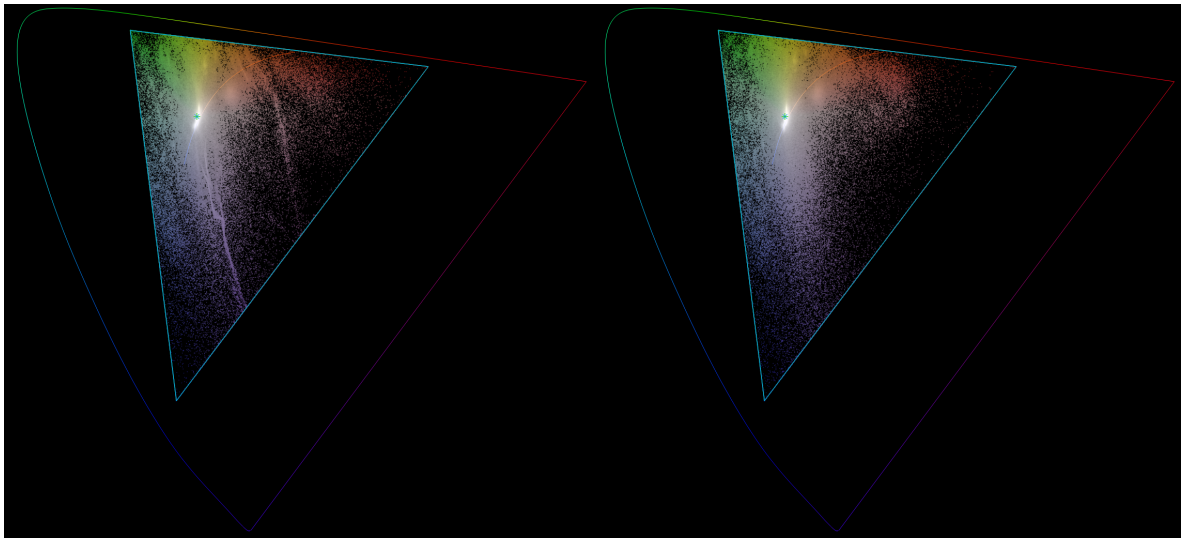
Figure 5.14: All shots are matched to 5600 K using X-Rite Colorchecker SG as reference scene.

5.7 Discussion

The most considerable challenge for this technology is the convoluted and opaque input state of the images. Cameras have unknown spectral sensitivities, and images reaching post-production have already undergone several unknown affine transformations like chromatic adaptation matrices or matrices to an encoding colour space. The *spectral intermediate*

(a) 2800K (b) 5600K (c) 10000K (d) White Only (e) RGB Only

Figure 5.15: All shots are matched to 5600 K (image b) using smooth spectra as reference scene.



(a) Colorchecker based transform (b) Smooth Spectra based transform

Figure 5.16: Comparison: uv chromaticity diagram of X-Rite Colorchecker SG vs smooth spectra

pipeline would benefit from a *generic smooth camera sensitivity* which could be used for material already converted and adapted to a specific intermediate colour space. One could think that the *Academy Reference Input Capture Device (RICD)* could be used as a reference observer. But the RICD is colourimetric to the 1931 Standard Observer, which means it provides no additional benefit in spectrally defining the actual non-colourimetric cameras.

We have shown that it is possible to seamlessly blend spectral and RGB workflows within a professional motion picture pipeline, even if the acquisition is still trichromatic. With the smooth spectra reference scene, we can rest assured that we get both smooth and accurate results within reason. We can give the creatives easy-to-use tools which operate in the spectral domain. In this lecture we did not discuss potential other use cases like *filter transfer* or *observer transfer*. The latter, especially, offers a lot of creative possibilities.