

Wide Gamut Spectral Upsampling with Fluorescence

A. Jung¹ and A. Wilkie^{2,3} and J. Hanika³ and W. Jakob⁴ and C. Dachsbacher¹

¹Karlsruhe Institute of Technology, Germany ²Charles University, Czech Republic

³Weta Digital ⁴École Polytechnique Fédérale de Lausanne (EPFL), Switzerland



Figure 1: A spectrally rendered scene with standard RGB to spectrum upsampling (left) cannot achieve reflectance spectra with arbitrary colour saturation and brightness due to energy conservation constraints. Our technique (centre) converts bright and deeply saturated input RGB colours to regular reflectance spectra and enhances them with a fluorescent component, reproducing colours from a considerably wider gamut. The input colours were specified in ACEScG—for instance, the top ring of the swimming pool has color (0.9, 0, 0.9) in ACEScG. To visualise colours of such high saturation, we interpret the rec2020 data of the rendered images as sRGB for display, and we also provide a display-independent difference image in CIE74 ΔE on the right (stopped down by 7ev, i.e. a saturated pixel value of 1.0 corresponds to $\Delta E = 128$). Similarly, the squares to the left visualise the ACEScG input colours reinterpreted as sRGB.

Abstract

Physically based spectral rendering has become increasingly important in recent years. However, asset textures in such systems are usually still drawn or acquired as RGB tristimulus values. While a number of RGB to spectrum upsampling techniques are available, none of them support upsampling of all colours in the full spectral locus, as it is intrinsically bigger than the gamut of physically valid reflectance spectra. But with display technology moving to increasingly wider gamuts, the ability to achieve highly saturated colours becomes an increasingly important feature.

Real materials usually exhibit smooth reflectance spectra, while computationally generated spectra become more blocky as they represent increasingly bright and saturated colours. In print media, plastic or textile design, fluorescent dyes are added to extend the boundaries of the gamut of reflectance spectra.

We follow the same approach for rendering: we provide a method which, given an input RGB tristimulus value, automatically provides a mixture of a regular, smooth reflectance spectrum plus a fluorescent part. For highly saturated input colours, the combination yields an improved reconstruction compared to what would be possible relying on a reflectance spectrum alone.

At the core of our technique is a simple parametric spectral model for reflectance, excitation, and emission that allows for compact storage and is compatible with texture mapping. The model can then be used as a fluorescent diffuse component in an existing more complex BRDF model. We also provide importance sampling routines for practical application in a path tracer.

1. Introduction

Physically based rendering has received considerable attention in recent years. Thanks to the availability of sophisticated material

models and Monte Carlo sampling techniques, it is now possible to create renderings of complex virtual worlds with stunning realism.

A number of physically based rendering systems throughout academia and industry simulate the underlying light transport spec-

trally, i.e. by considering the wavelength dependency of radiometric quantities. This has a number of advantages: in contrast to simple RGB-based simulations, it becomes considerably easier to support wave-optical effects, such as iridescence due to thin films, scratches, or other types of rough surface microstructure [BB17, WVJH17, YHW*18]. In the context of visual effects, spectral rendering is used to replicate the spectral sensitivity curves of cameras, thereby recreating the same metamerism as the onset objects to achieve a consistent look of real and virtual objects throughout all lighting conditions [FHF*17], [?]. Finally, spectra are the natural domain for computing products of colours (as is e.g. necessary to compute indirect illumination), and the approximation of this behaviour through RGB triplets is often poor [Pee93, WEV02].

From a practical point of view, transitioning to spectral rendering brings two clear drawbacks: the first is that scene content such as textures is typically created using tools that operate in various RGB colour spaces, which in turn necessitates an ambiguous conversion to spectral data before rendering can commence. This step is referred to as *spectral upsampling*.

The second issue is that energy conservation imposes a hard limit [Mac35b], *MacAdam's limit*, on reflectance colour saturation in any spectral renderer: no material can elastically scatter more light than it receives at any given wavelength. This limits materials to the *gamut of solid reflectances*, which intrinsically lacks colours beyond a certain level of saturation. In some cases, however, such extreme colours are desired. This issue is not restricted to simulations: designers of real-world materials (e.g. plastics, textiles and paint) routinely rely on optical brighteners or other fluorescent dyes to boost their overall colour saturation via *inelastic* scattering.

In contrast, in an RGB-based renderer it is simple to create very highly saturated colours, especially if a wide gamut RGB space is used as “working space” in which the multiplications of albedos and incident radiance are performed. While this approach is physically questionable at best (and downright wrong for colours outside the gamut of solid reflectances), it is noteworthy that within the limits of the spectral locus, a purely RGB-based workflow does not pose constraints on how bright or saturated objects can appear.

The intersection of these two issues motivates our work: we seek to develop a convenient method for spectral upsampling of arbitrary RGB colours, even those outside of the gamut of solid reflectances. In such cases, we model the material as a combination of a regular and a fluorescent dye that jointly reproduce the input colour—just as in real product design.

Our method builds on recent work for fast, minimal storage spectral upsampling [JH19] for the elastic component. For the fluorescent component of our model we derive a simple parametric form to describe the brighter and more saturated colours on the wide input gamut. In this work, we are mainly interested in the gamut extension capabilities offered by such an approach, and use a simple diffuse reflection model for the fluorescent component [JHD18].

Our technique is most relevant when working with assets created in wide gamut colour spaces as DCI P3, Rec. 2020, ACEScG, etc. Spectra for smaller spaces (e.g. sRGB) can generally be created directly, without the need for added fluorescence. Our technique

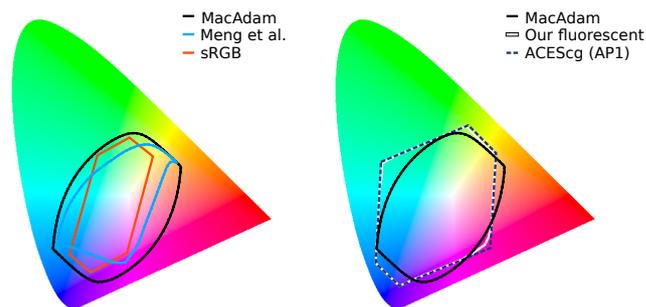


Figure 2: CIE chromaticity diagram showing all visible colours in a horseshoe shape together with slices through five gamuts: left, the sRGB gamut (red), the limits of the gamut of valid reflectances (MacAdam’s limit [Mac35a], black), and the gamut of natural (i.e. smooth [MSHD15], blue) reflectances. Right, the ACEScG gamut (dashed in dark blue) which we use as input colour space for our textures, as well as the approximate output colour space we cover with our fluorescent spectral upsampling (in the background in white). All gamuts are sliced at a brightness of $X+Y+Z=1.5$.

also enables spectral upsampling of RAW photographs that often have much wider colour gamut than sRGB. An additional benefit of our approach is that while all of sRGB can indeed be covered by conventional reflectance spectra, the highly saturated colours near the border of the sRGB gamut require very blocky spectral shapes near MacAdam’s limit of physically valid spectra. While physically plausible, such spectra are not very realistic, and in the context of Monte Carlo rendering are hard to sample. Therefore a moderate addition of fluorescence also makes sense near the boundaries of sRGB, in order to recreate the same colour with a smoother reflectance spectrum.

Our contributions are:

- the first spectral upsampling technique to include fluorescence
- a spectral model parameterise by three floating point values for the elastic component and three additional floating point values for the fluorescent part, making it small enough to be used in textured assets.
- analytical sampling routines for wavelength importance sampling at fluorescent scattering events.

2. Background and Previous Work

Colour Science. Given our understanding of the human visual system, colour is normally expressed as a vector of tristimulus values [WS00]. In the case of RGB spaces, all colours that can be expressed by values in $[0, 1]^3$ form the gamut of a given colour space. It has been shown that not all visible colours correspond to valid reflectances [Mac35b], due to energy conservation constraints. If one restricts oneself to natural reflectances, i.e. smooth reflectance spectra as they often appear in real materials, this gamut shrinks even more [MSHD15] (see also Figure 2).

Spectral Rendering. Physics tells us that the radiometric quantities associated with light depend on wavelength, and shall be computed by superposition. That is, transport may be simulated per wavelength separately, and the result added to the frame buffer in the end. In particular, the widely used practice of working with RGB values as units of computations in a renderer is actually wrong [WEV02]. To support bright colours outside the gamut of valid reflectances, we will need to extend the notion of independent transport per wavelength.

Fluorescence. Fluorescence is the process of light being absorbed at one wavelength and re-emitted at another, usually[†] longer wavelength. In our context we assume this process to be instantaneous. The most important parameters for this work are the absorption and the emission spectrum. The absorption spectrum $a(\lambda_i)$ determines how much light is absorbed for a given wavelength. This light is then re-emitted with a spectral distribution following the emission spectrum $e(\lambda_o)$ [Lak06]. The distance between the absorption and emission spectrum’s peak wavelength is known as the *Stokes shift*.

Fluorescence plays an important role in textile and packaging design where it is used to produce materials with bright colours and significant saturation. Furthermore, the colour of the scattered illumination due to fluorescence is typically relatively independent from the excitation wavelength, resulting in good colour constancy.

Similar to our setting, printing with fluorescent inks requires determining the fractions of regular and fluorescent ink. Rossier et al. [Ros13] determine the extended fluorescent gamut as combination of few fluorescent primary colours in the context of half toning, given an sRGB input image. Our model specifically targets larger gamuts and uses a parametric fluorescent additive.

Fluorescence also plays an essential role in the context of chemistry [Lak06] and biology [Kub17]. For instance, genetic modification of an organism’s DNA can cause it to express various fluorescent proteins in different cellular components, enabling detailed observations when excited with light of suitable wavelengths.

Rendering and acquisition of fluorescent materials in computer graphics has a long history [Gla94, WTP01, HHA*10]. For rendering we use a path tracer with next event estimation [MFW18], together with a simple diffuse BRDF [WWLP06]. In particular, we implement the BRDF model as described by Jung et al. [JHD18]:

$$f(\lambda_i, \omega_i, \omega_o, \lambda_o) = \frac{\delta_{\lambda_i, \lambda_o} (1 - ca(\lambda_i))r(\lambda_i) + ca(\lambda_i)Qe(\lambda_o)}{\pi}, \quad (1)$$

where the left summand in the numerator is the regular reflection. The Dirac delta function left ensures that this term only contributes when $\lambda_i = \lambda_o$. c is a scalar introduced for convenience to scale down the absorption spectrum $a(\lambda_i)$, and $r(\lambda)$ is the non-fluorescent reflectance spectrum. In the right summand, the fluorescent part, Q accounts for the quantum yield and $e(\lambda_o)$ is the emission spectrum mentioned above.

For any given input RGB colour, we are now interested in providing matching $r(\lambda)$, $a(\lambda)$, $e(\lambda)$ and c . We discuss modelling a ,

[†] Subjected to energetic radiation, some materials absorb multiple photons and emit light at shorter wavelengths [GM31]. We do not consider this case.

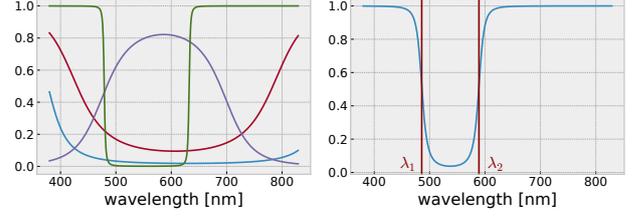


Figure 3: Left: Example spectra showing the expressivity of the sigmoid model. Right: Locations where we evaluate the derivative to penalise box-like spectra.

e , c and Q in Section 3.3. For r we rely on a standard model for spectral upsampling presented below.

Spectral upsampling. There are many techniques available to up-sample RGB values to a full spectrum. Other communities are interested in matching reference spectra (see for instance [CLC17]). We quickly recapitulate the most relevant methods for graphics:

MacAdam introduced a family of box spectra that can model the brightest and most saturated reflectances [Mac35b]. Smits [Smi99] uses optimisation to generate spectra which are smooth and constrained to be smaller than one. Meng et al. [MSHD15] provide a full colour gamut method for upsampling that also optimizes the smoothness of spectra. Energy conservation is not taken into account during the optimisation and enforced later on by scaling, hence bright spectra are not always achievable. Otsu et al. [OYH18] use a clustered PCA method to reconstruct spectra from tristimulus input. Their method does not obey the $[0, 1]$ constraint of reflectance spectra.

Our method builds on the parametric function space of Jakob and Hanika [JH19], which is able to express both saturated colours using blocky spectra as well as more moderate input with smooth spectra (cf. Figure 3). The model is a composition of a quadratic polynomial and a sigmoid that ensures energy-conserving reflectance on the interval $[0, 1]$:

$$r(\lambda) = S(P(\lambda)), \quad (2)$$

$$\text{where } P(\lambda) = c_0\lambda^2 + c_1\lambda + c_2, \text{ and } S(x) = \frac{1}{2} + \frac{x}{2\sqrt{1+x^2}}.$$

The coefficients $c^r = (c_0, c_1, c_2)$ are interpolated from a 3D look-up table generated using an one-time offline optimisation step. When upsampling RGB textures, the three coefficients c^r are typically stored for each texel, replacing the prior tristimulus values.

In our work, we want to avoid the blocky spectra at the boundary of the gamut of valid reflectances, but instead work with smoother spectra and mix in fluorescent dyes as optical brighteners to compensate the lack of saturation. This approach enables us to go outside the reflectance gamut and represent a richer set of colours. For this purpose, we extend the set of coefficients c^r with three additional fluorescence-specific coefficients c^f that represent the peak wavelength of the emission spectrum, a mixing factor, and the fluorophore’s stokes shift.

3. Fluorescent RGB upsampling

In this section we detail our upsampling technique. Like Jakob and Hanika [JH19], we compute a cube-shaped 3D-lookup table, where we store precomputed coefficients ($c^r \in \mathbb{R}^3$ for reflectance, $c^f \in \mathbb{R}^3$ for fluorescence) for discrete RGB values. The three dimensions of our lookup table correspond to the three dimensions of the RGB-[0, 1]³-cube of a given colour space, such that cube entries are spaced uniformly in RGB.

For each cube entry, we optimise the associated coefficients c^r and c^f such that we minimise the Euclidean distance inside the [0, 1]³ RGB cube between the target colour and the colour represented by the coefficients.

3.1. Overview

We begin by populating the table for the given RGB space with coefficients c^r as computed by the spectral upsampling method by Jakob and Hanika [JH19]. This process does not yield satisfactory reflectance spectra for all entries when working with a wide gamut colour space: some colours outside of MacAdam’s gamut simply cannot be reproduced by energy-conserving spectra, while others require unnaturally blocky spectra (a similar issue was reported by Meng et al. [MSHD15]). We detect the latter case using a simple analytical criterion detailed in Section 3.2. After this, we improve the spectra computed during this first stage by determining the parameters of our parametric model for fluorescence (Section 3.3).

3.2. Elastic scattering component

Figure 3 shows several example reflectance spectra created via non-fluorescent spectral upsampling using Equation (2). Note that some spectra become increasingly box-like, which is undesirable given the smoothness of natural reflectance spectra in the visible range.

To penalise such results during the optimisation, we require a metric that quantifies the smoothness of the resulting spectra. We choose the highest absolute derivative value that the model $r(\lambda)$ produces on the visible range $[\lambda_{\min}, \lambda_{\max}]$ in nanometer (nm). Since an analytical computation of the true maximum absolute derivative seems not feasible, we compute an approximation instead. First, we assemble the extremal values that the function reaches on the interval:

$$\mathcal{R} := \{r(\lambda_{\min}), r(\lambda_{\max}), r(\lambda_{\text{ext}})\},$$

where $\lambda_{\text{ext}} = -\frac{c_1}{2c_0}$ is the global minimum or maximum of the polynomial, which is only included if it lies between λ_{\min} and λ_{\max} . Exploiting the monotonicity of the sigmoid S , we then map the midpoint of this range through its inverse:

$$y_{\text{mid}} = S^{-1}((\max \mathcal{R} + \min \mathcal{R})/2).$$

To determine the associated wavelength, we must solve the quadratic equation $P(\lambda_{1,2}) = y_{\text{mid}}$. The final approximate derivative value is given by $|r'(\lambda_1)|$ or $|r'(\lambda_2)|$ —either solution is suitable. An example is shown in Figure 3.

The accuracy of this approximation increases with the maximum slope of r . For a box-like spectrum with values mostly close to 0 or 1, the corresponding internal polynomial P reaches extremely

low and high values, which get mapped to close-to-0 and close-to-1 by the sigmoid S , with the steepest derivative of $S(P)$ at $S = (\text{almost}0 + \text{almost}1)/2 \approx 0.5$ for symmetry reasons. Since we only want to penalise spectra above a certain steepness, we do not care about smooth spectra where the approximation is less accurate.

We then penalise spectra with an absolute derivative above a threshold t by adding the term $100 \cdot \max\{0, |r'(\lambda_1)| - t\}$ to the optimisation objective (i.e. the distance inside the RGB [0, 1]³ cube). We use $t = 0.016$ (when plotted over wavelengths in nm) as a threshold for smooth spectra, which is the maximum derivative observed on the classic Macbeth colour checker (see Figure 7). We use $|r'(\lambda_1)| - t$ in order to assure a continuous optimisation objective. Considering that colour distances inside the [0, 1]³ cube can reach values in $[0, \sqrt{3}]$, we then scale this difference by a factor of 100 in order to impact the target function even for slightly larger derivatives, causing the optimiser to prefer the fluorescent component to reduce colour reprojection error.

3.3. A parametric model for fluorescent spectra

To describe fluorescence in the BRDF model Equation (1) we must still specify concrete absorption and emission spectra. We use a simple low-dimensional parametric model instead of densely sampled spectra to ensure that our model is efficient to use at render time. Our model can also be importance sampled analytically, which is beneficial for Monte Carlo light transport simulation. The choice of an analytic model also enables a considerably simplified optimisation of model parameters for a given RGB input value.

In order to keep the number of parameters small, we fix the parameter $Q = 0.96$. We chose this value, as several real-world materials have quantum yields close to 100% [Lak06] and we want Q to be as high as physically reasonable in order to achieve higher saturations via reradiation. Considering the model in Equation (1), a lower Q would be mathematically equivalent to a larger Q coupled with lowering c and $r(\lambda)$, so we expect little benefit from using an additional parameter for Q .

Thus our model has three parameters $c^f = (\lambda_e, c, s)$: The peak wavelength of the emission spectrum λ_e in nm, a mixing factor $c \in [0, 1]$ (as in Equation (1)) to determine the amount of fluorescence, and the stokes shift s in nm, for describing the distance between the emission and absorption spectrum’s peak wavelengths.

We model the emission spectrum with a cubic b-spline

$$b(x) = \begin{cases} (x+3)^2/6 & \text{if } -3 < x \leq -1 \\ -x^2/3 + 1 & \text{if } -1 < x \leq 1 \\ (x-3)^2/6 & \text{if } 1 < x < 3 \end{cases} \quad (3)$$

which we stretch and translate to account for varying spectral widths and λ_e , respectively. We then define the absorption spectrum as the mirror image of the emission spectrum plotted over wavenumber, as literature suggests that this is often the case for real materials [Lak06]. We further model each pair of emission and absorption spectra such that they touch but do not overlap. In reality, such an overlap usually exists, but it has only a small impact on the resulting colour. In our case, this assumption simplifies our model because we can model the effects of excitation and emission

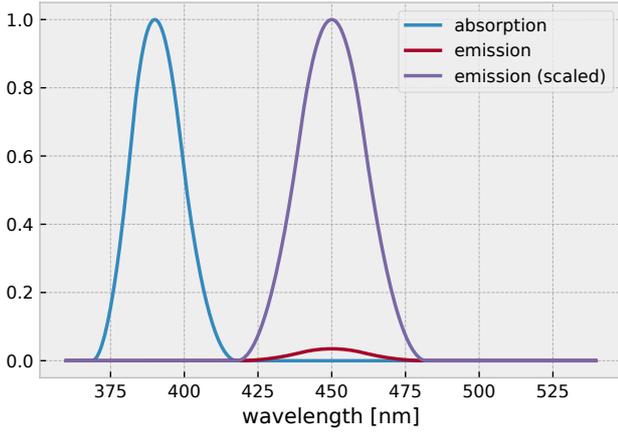


Figure 4: Our model for fluorescent spectra: Since the spectra mirror each other over wavenumbers, this plot over wavelengths shows the absorption spectrum as a distorted b-spline. This distortion increases with larger Stokes shift and smaller emission peak wavelength. In this example, the emission peak wavelength $\lambda_e = 400\text{nm}$, the Stokes shift is 60nm for both. The mixing coefficient c only is relevant when evaluating the full BRDF.

as separable without the danger of sampling events that transition to shorter wavelengths.

We compute a scaling factor α which is used to stretch the emission spectrum, such that mirroring it over wavelength results in an absorption peak wavelength at the specified distance s :

$$\alpha = \frac{\lambda_e s}{2\lambda_e - s}. \quad (4)$$

This leads us to the emission spectrum

$$e(\lambda_o) = b \left(\frac{3}{\alpha} (\lambda_o - \lambda_e) \right) \cdot \frac{9}{8\alpha}, \quad (5)$$

and the absorption spectrum

$$a(\lambda_i) = b \left(\frac{3}{\alpha} \cdot \left(\left(\frac{2}{\lambda_e - \alpha} - \frac{1}{\lambda_i} \right)^{-1} - \lambda_e \right) \right). \quad (6)$$

In contrast to the absorption spectrum, which takes on the value 1 at its peak, the emission spectrum is normalised so that it integrates to 1. This normalisation constant for $e(\lambda_o)$ has a closed-form solution and is given by $9/(8\alpha)$. An example of a typical resulting emission and absorption spectrum pair is given in Figure 4.

3.4. Parameter search

Target function. We minimise an objective function given by the Euclidean distance between the input colour and the colour represented by the coefficients. To compute the latter we evaluate the

reflectivity R for a given illuminant $I(\lambda)$ as

$$R(\lambda) = r(\lambda) \cdot (1 - ca(\lambda)) \cdot I(\lambda) + e(\lambda) \cdot \int_{-\infty}^{\infty} ca(\bar{\lambda}) I(\bar{\lambda}) d\bar{\lambda}, \quad (7)$$

and convert it to RGB using the colour matching functions associated with the desired colour space. We furthermore add a penalty for large derivatives described in Section 3.2. If no fluorescence is present, the above expression reduces to computing the colour of the reflectance spectrum r under a given illuminant I .

An outline of our algorithm is given in Algorithm 1. For the optimisation we use Ceres [AMO], both for the initial search for reflectance coefficients c^r , and for finding fluorescent coefficients c^f wherever reflectance alone does not lead to a satisfactory result. This can happen when there is no physically valid reflectance for the given input colour, or when a physically valid reflectance exists but is rejected for being too box-like.

In the initial search, Ceres finds zero-error c^r for colours inside the gamut of smooth spectra. In the following we only consider entries where those coefficients still have some error left. We will discuss finding fluorescence coefficients c^f wherever required, and refining c^r in combination with c^f wherever c^r alone is not enough.

We found that whenever fluorescence is required, the performance of the optimiser heavily depends on the quality of the input coefficients, and it easily gets stuck in local minima. We therefore support the optimiser in two ways: Initially, we use a priori knowledge about the fluorescence parameter space to find a "good" initialisation. Next, we iterate multiple calls to Ceres, while propagating information from neighbouring cube entries. We always cache the best coefficients found so far, in case one of these steps increases error.

Initialisation. After the initial search for c^r , we need to find a good initial guess for c^f to guide Ceres. We already know the meaning (emission peak wavelength, concentration, Stokes shift) of the fluorescence coefficients, which leads us to reasonable bounds for their value range (details in Section 4). We then use these bounds to perform a brute-force search over the three-dimensional fluorescent coefficient space to find initial values with low error, given the initial c^r . Figure 5 visualises a one-dimensional slice of such a brute-force search for fixed c^r , c and s , which reveals an intricate optimisation objective with several local minima.

Iterative refinement. We found it most effective to call Ceres multiple times, alternately optimising both c^r and c^f jointly, and only optimising c^r (Algorithm 1). If smooth reflectances are desired, the target function inside these calls penalises large derivatives as described in Section 3.2. We terminate this process after a maximum number of iterations, or when the error no longer improves.

Information from neighbours. While the procedure described so far finds optimal solutions for some RGB values, for many others it gets stuck in local minima. To improve convergence globally, we propagate solutions from adjacent cube entries: for each entry, whose coefficients c_j^r, c_j^f have not yet converged below some error threshold, we look at the coefficients c_k^r, c_k^f of adjacent entries $k = 1, \dots, 26$. Coefficients that would result in a lower error for the

centre entry than c_j^r, c_j^f are then used to initialize the starting guess of the subsequent iteration.

We then resume optimising c^r and c^f jointly and optimising c^r only, reseeding using neighbour information, and so on. This process repeats a number of times to ensure that information can propagate to more distant entries.

We found that results further improve when decreasing the error thresholds over iterations (implemented by comparing neighbouring error to an upscaled centre error), and also starting from the entry's own coefficients if no better neighbours are available (hence, the loop over k in Algorithm 1 begins at $k = 0$, which represents the centre entry j).

Algorithm 1 Parameter Optimisation

```

for all cube entries do
  optimise  $c^r$ 
for all cube entries with remaining error do
  brute-force search  $c^f$ 
for  $i = 1, \dots, 50$  do
  for all cube entries  $j$  with remaining error do
    for all  $k = 0, \dots, 26$  neighbours with error  $\leq i \cdot$  own error do
       $c_j^r \leftarrow c_k^r, c_j^f \leftarrow c_k^f$ 
      for  $l = 1, \dots, 31$  do
        if  $c_j^r, c_j^f$  still have error then
          if  $l \bmod 2$  then
            optimise  $c^r$  and  $c^f$  jointly
          else
            only optimise  $c^r$ 

```

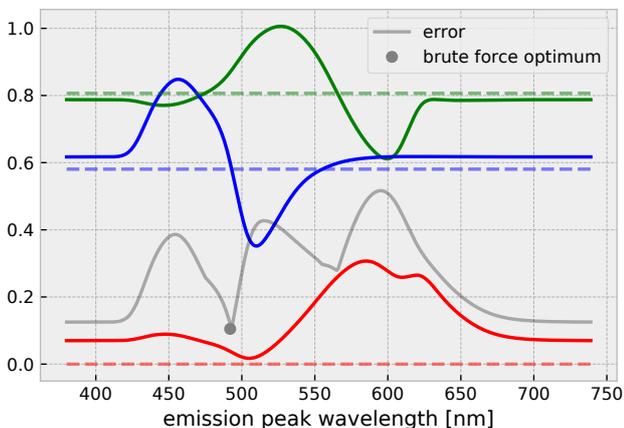


Figure 5: One-dimensional brute-force search for a low-error emission peak wavelength λ_e , given a fixed Stokes shift (50nm), mixing factor ($c = 1.0$) and reflectance coefficients. The goal is to match RGB=(0,0.80,0.58) (dotted lines). The coloured lines represent the achieved colour depending on λ_e . The grey line is the RGB-distance between goal colour and the colour represented by the coefficients. The grey dot marks the best value found with a step size of 1nm.

3.5. Importance sampling

Our model admits a simple importance sampling technique for use in Monte Carlo rendering systems. Given a material interaction, we must first decide whether to scatter elastically or inelastically, for which we use the probabilities specified in [JHD18]. In case of inelastic scattering, the next step involves importance sampling a new wavelength from the absorption spectrum $a(\lambda_i)$ on a camera path, or from the emission spectrum $e(\lambda_o)$ on a light path.

We can sample $e(\lambda_o)$ precisely using the following expression:

$$\lambda_o = \lambda_e + \alpha \left(\frac{2}{3}(\xi_1 + \xi_2 + \xi_3) - 1 \right). \quad (8)$$

ξ_1, \dots, ξ_3 are uniformly distributed random numbers on the interval $[0, 1]$. This relies on an idea by Stark et al. [SSA05], which exploits the convolutional nature of b-splines: in particular, any b-spline can be expressed as a repeated convolution of box functions, which can hence be sampled by summing a suitable number of uniformly distributed variates.

The same approach unfortunately cannot directly be applied to the absorption spectrum $a(\lambda)$, since the underlying spline curve is mapped through a nonlinear transformation. We currently sample a triangular density function, whose peak and support match those of $a(\lambda)$, though we note this scheme could be significantly improved e.g. by sampling a normal distribution or suitably parameterized b-spline. A code snippet for the triangular sampling scheme is provided in the supplemental material. The integral of the absorption spectrum required for computing the probability of a fluorescent event on a camera path as specified in [JHD18] can be approximated with the integral of a b-spline with the same support.

In the elastic case, we simply preserve the current wavelength and multiply the path throughput by the reflectance spectrum $r(\lambda)$ divided by the probability for elastic scattering.

4. Results

We implemented the simple diffuse fluorescent BSDF from Equation (1) in the context of a standard path tracer with next event estimation in an existing spectral rendering framework. All our experiments were run on an Intel i7-6700 3.4GHz CPU.

Optimisation Parameters. We tested our optimisation for the ACEScg colour space. The cube entries are placed in a regular grid at $[k \cdot 1/32]^3$, $k = 0, \dots, 32$. After the initial search for c^r , the brute-force search evaluates all combinations of emission peak wavelengths in $[300, 800]$ in 10nm-steps, Stokes shifts in $[5, 100]$ in 10nm-steps, and mixing factors in $[0, 1]$ in 0.1 steps to initialise c^f wherever required. While a search with smaller step sizes (e.g. 1nm for emission peak and Stokes shift) initially yields lower-error coefficients, it takes considerably more time (46min for 1nm steps versus 3min for 10nm steps) and the initial benefit vanishes after several rounds of seeding with adjacent coefficients. We performed 50 iterations considering adjacent coefficients as seed, each followed by up to 31 calls to Ceres, alternating joint optimisation of c^r and c^f , and optimisation of c^r only.

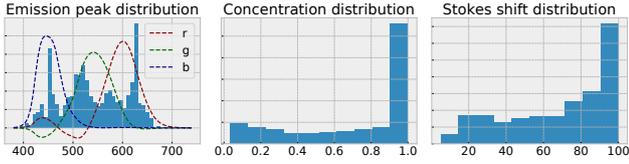


Figure 6: Distribution of λ_e , c and s for entries using fluorescence with $t = \infty$ and no UV light. The ACEScG RGB response curves are overlaid on the emission peak distribution.

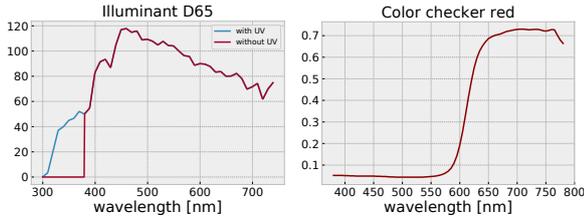


Figure 7: Left: Illuminant D65 with and without ultraviolet component. Right: colour checker red reflectance spectrum.

Cube statistics. For $t = \infty$ (i.e. arbitrarily large reflectance derivatives), both with and without UV light in the illuminant, the whole optimisation took 2h 43min. In this case 5728 entries required fluorescence, 1486 of which still have non-negligible error when the optimisation is complete. For $t = 0.016$ this number increases to 1908, with a total of 16333 entries requiring fluorescence, leading the optimisation to last 4 hours.

We visualise the distribution of optimised λ_e , c and s in Figure 6 for all entries with $c > 0$, i.e. all entries actually using fluorescence, for the $t = \infty$ cube. The emission peaks are somewhat correlated to the colour matching functions' peaks at roughly 450, 530 and 630nm. The concentration and Stokes shift histograms show that larger values are preferred, which makes sense as those increase the amount of reradiated energy.

Jointly optimising both c^r and c^f in every call to Ceres takes twice as long to achieve the same quality as alternating joint optimisation of c^r and c^f , and isolated optimisation of c^r . Alternating isolated optimisation of either c^r or c^f leads to a worse result, with 2782 entries still having a non-negligible error after almost 4 hours for $t = \infty$ without UV light.

HSV test card. In a first experiment, we directly convert texels of a 2D ACEScG texture (Figure 8 right) to colours that can be represented by our model. To do so, we perform lookups into the pre-computed coefficient cube and use trilinear interpolation to interpolate reflectivities of the cube's entries. Following this, we evaluate the resulting RGB reflectance using the same illuminant that was also used to create the coefficient cube.

We compare the error of the input texture to the resulting colour for a cube without fluorescence (i.e. only optimising reflectance coefficients), and two cubes with fluorescence in Figure 8. In order to show what is theoretically possible we compare arbitrary derivatives with and without fluorescence, and using a derivative

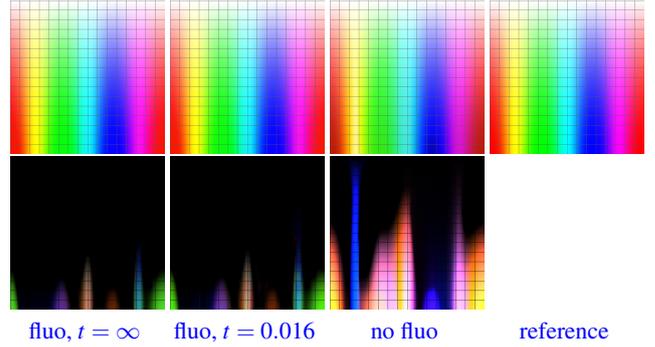


Figure 8: Texture reconstruction comparison. From left to right: fluorescence and $t = \infty$, fluorescence and $t = 0.016$, no fluorescence, input reference. The image values are meant for ACEScG but are displayed as sRGB. The bottom row shows 10 times the absolute difference in RGB. RMSE: no fl: 0.10443, fl ($t = \infty$): 0.01549

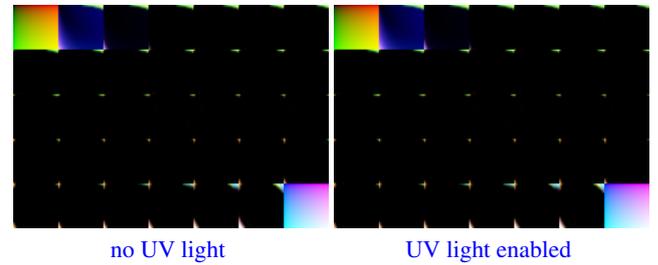


Figure 9: Overexposed absolute error of individual cube entries at 1/32 slices throughout the cube. The upper left and lower right square show the true color of the first and last slice of the cube instead of the error. The blue axis increases from slice to slice, the red (green) axis from left to right (top to bottom) within each slice. At the borders and first and last slice, the corresponding channels are 0 or 1. An animation of this is part of the supplemental material.

threshold of $t = 0.016$ with fluorescence. Both the input texture as well as the reconstructed textures store ACEScG tristimulus values, which on screen are re-interpreted as sRGB, thus visualising a desaturated version of the true colours. The addition of fluorescence increases both the number of colours that can be reconstructed with zero error, as well as the error for the remaining colours. However, we were not able to achieve zero error on the full ACEScG gamut, which is most obvious in the dark blue and red regions.

Cube entry errors. Figure 9 visualises the component-wise absolute error of the colour for each entry of the coefficient cube. The black region in the centre corresponds to the colours that can be reached by either reflectance only or reflectance plus fluorescence. The first layer corresponds to the blue channel being 0, which poses the biggest issue for the optimiser. The red and green channel being zero correspond to the upper row and left column respectively.

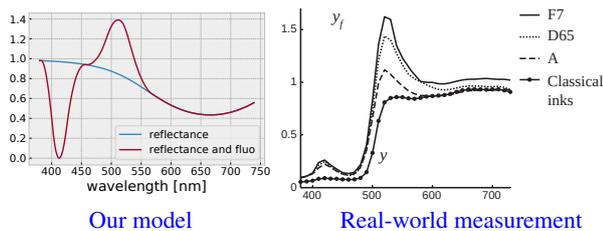


Figure 10: Phenomenological comparison of reflectivity plotted over wavelength for saturated colours. Left: our result ($t = \infty$, no UV light, $RGB=(0.52, 0.97, 0.84)$). Right: real life example (day-light fluo yellow colourant) reproduced after [Ros13, Fig. 1.2]. Both spectra exhibit a base reflectivity which stays below 1.0. On top, there is a fluorescent peak at the rising edge of the spectrum, which for the real material is even stronger in illumination with UV light.

Gamut. We evaluated the approximate gamut of our fluorescent upsampling model Figure 2 (right). This figure has been drawn based on data assuming a D65 illuminant, and results may vary for other illuminants. As can be seen, the fluorescent upsampling model is able to reach outside MacAdam’s limit, and almost fills all corners of the wide ACEScG gamut. There are some slight differences remaining in cyan and more obvious gaps in the magenta corner. As our search space is limited by ACEScG, we do not reach full coverage of MacAdam’s limit. To evaluate the limits of our method, we would need an even larger input colour space.

Spectral comparison. We compare the phenomenological appearance of the reflectivity created by our model with the reflectivity of a real-world material that was enhanced with a fluorescent dye in Figure 10. Both spectra are mostly below 100% reflectivity except for the fluorescent peak. The synthetic and real fluorescent peak are of roughly similar shape, intensity and width.

UV light. Figure 9 reveals that adding ultraviolet light to the illuminant as shown in Figure 7 does not decrease the overall error. This might be partly because the ultraviolet component of D65 is much weaker than the visible component. Another possible reason is that due to the limited Stokes shift, adding UV light would mostly influence blue-ish colours. However, Figure 9 shows that colours with large blue and small red and green values inside the cube can already be realised well without fluorescence. More comparisons with and without UV light are presented in the supplemental material.

Reradiation Matrices. The reradiation matrix corresponding to the spectra can be created by evaluating the BRDF in discrete steps. Examples of such matrices are presented in Figure 13. Since the absorption and emission spectrum do not overlap, the area corresponding to fluorescence is entirely above the diagonal.

Cube Interpolation. We evaluate three different strategies for converting an RGB texture to a coefficient texture. We always interpret the texel’s RGB value as cube coordinates in $[0, 1]^3$. We compare using coefficients from the coefficient cube’s nearest entry to

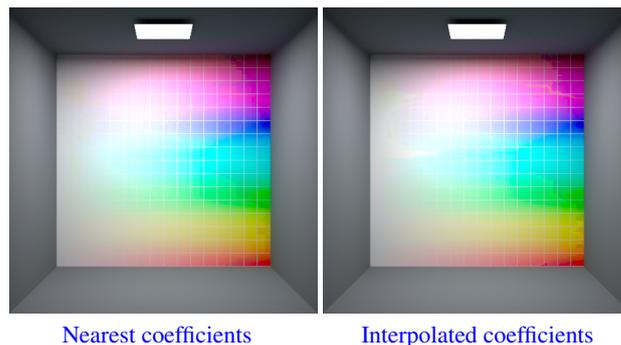


Figure 11: A colourful input texture converted to coefficient textures using the nearest (left) and interpolated (right) coefficients. While using nearest coefficients produces many seams with low contrast, interpolating coefficients results in a mostly smooth image, but with few, more disruptive artefacts. These are most noticeable along the transition to where fluorescence is required, but can also be spotted in the white regions on the left.

interpolating coefficients of the 8 surrounding cube entries in Figure 11.

As both these strategies come with drawbacks, we also present a rendering based on true interpolation of the reradiation matrices represented by the 8 surrounding cube entries in Figure 12. Strictly speaking this does not result in a low-storage texture representation anymore, since each texel now stores a full reradiation matrix, which is why this figure uses a lower resolution input texture. Examples for interpolated reradiation matrices are presented in Figure 13. Note that a material corresponding to such an interpolated matrix would contain several fluorophores mixed together, while at discrete cube entries the corresponding material contains only one fluorophore.

When interpolating fluorescence coefficients we have to take care to only interpolate emission peaks λ_e and stokes shifts s of entries that actually incorporate fluorescence, i.e. entries with $c > 0$, since otherwise their coefficients stored for λ_e and s are meaningless. The mixing coefficient c itself however needs to be interpolated with regular interpolation weights.

It turns out that when transitioning from the region that does not require fluorescence to the region that does, the reflectance spectra’s brightness drops significantly. Due to the nonlinearity of the sigmoid mapping, a trilinear interpolation of reflectance coefficients does not result in a linear drop of the reflectance spectrum intensity. All the while, the fluorescent component’s intensity increases linearly with the interpolation of c . This results in a glow-like effect in this transition region, caused by the overly bright interpolated reflectance spectra. In conclusion, while for smooth regions inside the cube interpolating coefficients produces reasonable results, it is not a good idea in general due to the abrupt change in reflectance once fluorescence is incorporated. On the other hand, using the nearest entry’s coefficients results in a larger number of hard transitions, but less noticeable ones.

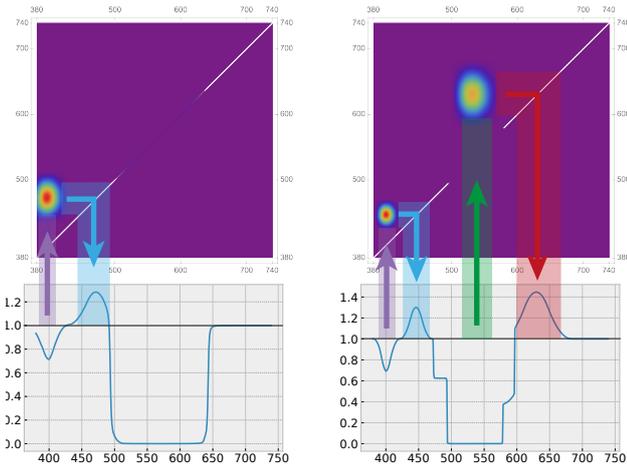


Figure 12: Reradiation matrices with input from all 8 voxel corners (top), and corresponding reflectance spectra under illuminant E (bottom). The example on the left is typical: the 8 corners of the voxel for a particular RGB value place several almost identical reradiation lobes in close proximity, so that the resulting overall reradiation is still confined to a single area in the matrix. This is the sort of scenario where interpolating coefficients yields reasonable results. But in some cases, the scenario seen on the right occurs: two or more clearly distinct reradiation components corresponding to different emission colours are introduced, so a full matrix reconstruction is needed.

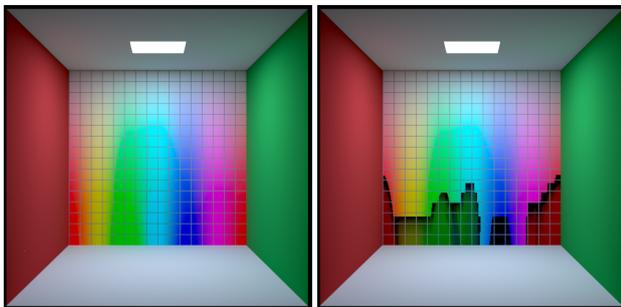


Figure 13: In this rendering, we evaluate all 8 reradiation matrices involved in a given coefficient cube voxel, and analytically sample the discrete reradiation matrix defined by the interpolated coefficients. While this requires a lot of storage (we have to explicitly generate the entire reradiation matrix for each texel), this is the proper way to do it, and leads to a smooth result. The image on the right has reflectance set to 0 wherever fluorescence is present, and shows how it increases as chroma goes up. Some hues, like e.g. orange, do not seem to require fluorescence as much as others, as the gamut of reflectance spectra reaches further in some regions (see Figure 2).

5. Conclusion and future work

We presented a first method capable of automatically adding a fluorescent component to reflectance spectra that are synthesised based on RGB values. A typical use case for this are wide gamut input textures, such as textures in the ACEScG space which we use for our examples. In such a large RGB space, not all colour values have corresponding pure reflectance spectra, as parts of it lie outside the gamut of solid colours. We solve this problem by adding fluorescent components to reflectance spectra, which allows us to generate highly saturated solid colours without violating any laws of physics. We will upload the coefficient cubes as well as an open source implementation that can be used to read and query them.

Our simple parametric model for the absorption and emission spectrum of the fluorescent component is suitable for texturing as it is based on three floating point coefficients. It matches the sigmoid spectra used for normal reflectance in our system insofar as it is also physically plausible, without directly corresponding to any particular set of real fluorescent dyes: instead, it is designed to phenomenologically match the mirrored appearance of real absorption and emission spectra.

In the future, we will analyse actual fluorescent dyes and their absorption and emission spectra with a PCA-based method, and/or directly use a database of a few relevant dyes instead. This will improve our ability to match the appearance of real highly saturated materials. For instance, to generate spectral data that plausibly mimics the appearance of printed materials, the reradiation matrices of only very few real fluorescent inks can be used.

Interpolation of our model's coefficients, while producing reasonable results in most cases, produces unintuitive results on the border where the spectral reconstruction switches from plain to fluorescence enhanced. Adding lower amounts of fluorescence to less saturated colours, even if it would not yet be necessary, might result in a smoother drop of standard reflectance intensity, and yield more reasonable interpolation results. Additionally, a method to enforce adjacent cube entries to have similar emission peaks, even at a slightly increased error, might be necessary to improve interpolation results further.

In this initial version, our model does not produce overlapping absorption and emission spectra. This way, we dodge the question whether fluorescence will ever result in a shift to shorter wavelengths or not. Dealing with this question is orthogonal to the purpose of this work, so we deemed it best to skirt the problem entirely. If reradiation were to occur in both directions, evaluation of indirect lighting effects might become more interesting if a certain amount of re-absorption of fluorescent emission takes place.

References

- [AMO] AGARWAL S., MIERLE K., OTHERS: Ceres solver. <http://ceres-solver.org.5>
- [BB17] BELCOUR L., BARLA P.: A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Transactions on Graphics* 36, 4 (July 2017), 65. 1
- [CLC17] CAO B., LIAO N., CHENG H.: Spectral reflectance reconstruction from rgb images based on weighting smaller color difference group. *Color Research & Application* 42, 3 (2017), 327–332. 3

- [FHF*17] FASCIONE L., HANIKA J., FAJARDO M., CHRISTENSEN P., BURLEY B., GREEN B.: Path tracing in production – part 1: Writing production renderers. In *SIGGRAPH 2017 Courses* (2017). 2
- [Gla94] GLASSNER A.: A model for fluorescence and phosphorescence. In *Fifth Eurographics Workshop on Rendering* (1994), Eurographics, pp. 57–68. 3
- [GM31] GÖPPERT-MAYER M.: Über Elementarakte mit zwei Quantensprüngen. *Annalen der Physik* 401, 3 (1931), 273–294. 3
- [HHA*10] HULLIN M., HANIKA J., AJDIN B., KAUTZ J., SEIDEL H.-P., LENSCH H.: Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. *Transactions on Graphics (Proceedings of SIGGRAPH)* 29, 4 (2010), 1–7. 3
- [JH19] JAKOB W., HANIKA J.: A low-dimensional function space for efficient spectral upsampling. *Computer Graphics Forum (Proceedings of Eurographics)* 38, 2 (Mar. 2019). 2, 3, 4
- [JHD18] JUNG A., HANIKA J., DACHSBACHER C.: A simple diffuse fluorescent BBRRDF model. In *MAM2018: Eurographics Workshop on Material Appearance Modeling* (2018). 2, 3, 6
- [Kub17] KUBITSCHKE U.: *Fluorescence microscopy: from principles to biological applications*. John Wiley & Sons, 2017. 3
- [Lak06] LAKOWICZ J. R.: *Principles of Fluorescence Spectroscopy*. 3rd ed. Springer, 2006. 3, 4
- [Mac35a] MACADAM D. L.: Maximum visual efficiency of colored materials. *Journal of the Optical Society of America* 25, 11 (1935), 361–367. 2
- [Mac35b] MACADAM D. L.: The theory of the maximum visual efficiency of colored materials. *Journal of the Optical Society of America* 25, 8 (1935), 249–249. 2, 3
- [MFW18] MOJŽIČ M., FICHET A., WILKIE A.: Handling fluorescence in a uni-directional spectral path tracer. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations* (2018). 3
- [MSHD15] MENG J., SIMON F., HANIKA J., DACHSBACHER C.: Physically meaningful rendering using tristimulus colours. *Proc. Eurographics Symposium on Rendering* 34, 4 (June 2015), 31–40. 2, 3, 4
- [OYH18] OTSU H., YAMAMOTO M., HACHISUKA T.: Reproducing spectral reflectances from tristimulus colours. *Computer Graphics Forum* 37, 6 (2018), 370–381. 3
- [Pee93] PEERCY M. S.: Linear color representations for full speed spectral rendering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (1993), SIGGRAPH '93, pp. 191–198. 2
- [Ros13] ROSSIER R.: *Framework for printing with daylight fluorescent inks*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2013. 3, 8
- [Smi99] SMITS B.: An RGB-to-spectrum conversion for reflectances. *Journal of Graphics Tools* 4, 4 (1999), 11–22. 3
- [SSA05] STARK M. M., SHIRLEY P., ASHIKHMIN M.: Generation of stratified samples for b-spline pixel filtering. *J. Graphics Tools* 10, 1 (2005), 39–48. 6
- [WEV02] WARD G., EYDELBERG-VILESHIN E.: Picture Perfect RGB Rendering Using Spectral Prefiltering and Sharp Color Primaries. In *Eurographics Workshop on Rendering* (2002), The Eurographics Association, pp. 117–124. 2
- [WS00] WYSZECKI G., STILES W. S.: *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley classics library. John Wiley & Sons, 2000. 2
- [WTP01] WILKIE A., TOBLER R., PURGATHOFER W.: Combined rendering of polarization and fluorescence effects. In *Rendering Techniques* (2001), pp. 197–204. 3
- [WVJH17] WERNER S., VELINOV Z., JAKOB W., HULLIN M.: Scratch iridescence: Wave-optical rendering of diffractive surface structure. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 36, 6 (Oct. 2017). 1
- [WWLP06] WILKIE A., WEIDLICH A., LARBOULETTE C., PURGATHOFER W.: A reflectance model for diffuse fluorescent surfaces. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia* (2006), pp. 321–331. 3
- [YHW*18] YAN L.-Q., HAŞAN M., WALTER B., MARSCHNER S., RAMAMOORTHY R.: Rendering specular microgeometry with wave optics. *ACM Trans. Graph.* 37, 4 (July 2018), 75:1–75:10. 1