

Moment-based Constrained Spectral Uplifting

L. Tódová¹  and A. Wilkie¹  and L. Fascione²

¹Charles University, Czech Republic

²Weta Digital, New Zealand

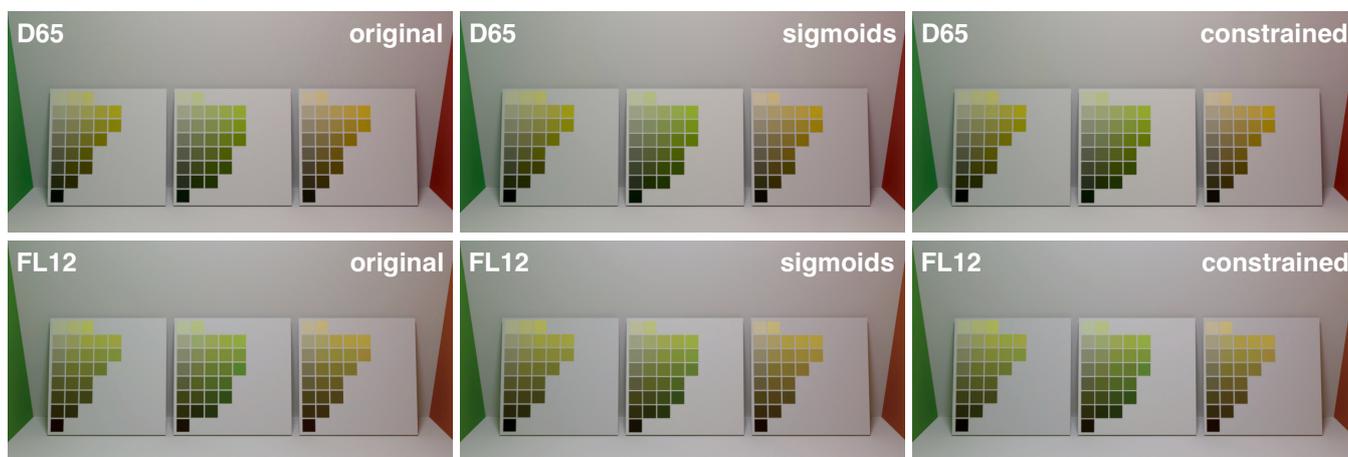


Figure 1: Pages from the Munsell Book of Colour under illuminant CIE D65 (top row) and CIE FL12 (bottom row). **Left column:** the materials in the scene are defined via their spectral properties, so no uplifting is needed. **Middle column:** the materials are defined as RGB values, and uplifted with the sigmoid method. Note that no metameric failures occur in the bottom row: the colour gradient stays ordered. **Right column:** The materials are defined as RGB values, and uplifted with our method. The spectra of the Munsell Book of Colour were used to constrain the uplift: as one can see, particularly in the middle page (050GY), the spectra show the same metameric disruptions of the colour sequence as real Munsell Book of Colour pages do. Note that only the colour chips which fall into sRGB are shown on the pages: so there are fewer than in the photos of real pages in Figure 2. Also note that under FL12, the more saturated chips are also outside the gamut of sRGB: the metameric gradient failure is actually considerably more noticeable in real life (e.g. in a viewing booth), or on a wide gamut cinema screen, than in these renderings. White balance was applied to both sets of images.

Abstract

Spectral rendering is increasingly used in appearance-critical rendering workflows due to its ability to predict colour values under varying illuminants. However, directly modelling assets via input of spectral data is a tedious process: and if asset appearance is defined via artist-created textures, these are drawn in colour space, i.e. RGB. Converting these RGB values to equivalent spectral representations is an ambiguous problem, for which robust techniques have been proposed only comparatively recently. However, other than the resulting RGB values matching under the illuminant the RGB space is defined for (usually D65), these uplifting techniques do not provide the user with further control over the resulting spectral shape. We propose a method for constraining the spectral uplifting process so that for a finite number of input spectra that need to be preserved, it always yields the correct uplifted spectrum for the corresponding RGB value. Due to constraints placed on the uplifting process, target RGB values that are in close proximity to one another uplift to spectra within the same metameric family, so that textures with colour variations can be meaningfully uplifted. Renderings uplifted via our method show minimal discrepancies when compared to the original objects.

CCS Concepts

• **Computing methodologies** → **Reflectance modeling**;

1. Introduction

Over the last few years, the demand for physical accuracy in rendering has grown substantially. The main advantage of providing renderers with the capability to simulate light transport in a physically accurate fashion is that one can obtain an intrinsically realistic scene appearance via global illumination. But there are also other useful capabilities of physically correct rendering, such as colour accuracy: working with spectral data allows one to predict object appearance under varying illuminations.

However, almost all VFX and game engine assets, in particular their textures and material data, are defined in RGB, as this is easier for the artists creating them. Defining genuinely spectral assets would require artists to either specify spectral reflectances from reference data collections (e.g. colour atlas data), or to measure them on a real asset with a spectrometer. Both options are more or less tedious, and the second one is not possible for fully virtual assets. Additionally, painting spectral textures is an open problem.

To make spectral rendering possible while utilising RGB textures as input, a reliable conversion from RGB to spectral data, called *spectral uplifting* (or *spectral upsampling*), is needed. However, as RGB colour spaces are intrinsically smaller than the space of all possible spectra, multiple different spectral representations of the same RGB colour, called *metamers*, exist[†]. Typically, metamers no longer match in appearance under different light sources. This may cause issues with spectra created by current uplifting technology, which, while robust and reliable, creates arbitrary metamers. Therefore, the appearance of assets defined via RGB can be somewhat unpredictable under changing illumination.

Figure 2 shows an example of this: the Munsell Book of Colour is an old colour atlas, which was defined before fluorescent lights were common. The yellow-green pages of the atlas are known to exhibit noticeable distortions in the colour gradient of the samples when viewed under fluorescent or LED light sources. If an RGB texture of an atlas page were used as input, a naive uplifting technique would have no chance of reproducing the fact that under one type of light source there is a smooth gradient, while under others the gradient breaks down.

Accurately reproducing colour atlas pages under varying illumination is of course not a very practically useful goal in itself: we merely use them as an example of a problem which the VFX industry faces in practice. There, one must frequently deal with the task of matching plate footage of real objects with rendered images of their digitised asset counterparts. In such scenarios, it is crucial to preserve visual continuity: the viewer must not be made aware that a real asset is replaced with a virtual one. However, in order for the viewer to be oblivious to such an asset switch, not even the slightest colour differences between them must be visible.

For such work, 3D artists currently use standard colourspace modelling tools to carefully craft elaborate virtual doubles of a real asset, and obtain a perfect appearance match under the main target

[†] From a mathematical viewpoint, there are infinitely many metamers: but in reality the number of achievable metamers is limited by the properties of real colourants and pigments.

illumination (e.g. for daylight). If the same asset is later used in another scene with different illumination (e.g. fluorescent lamps, in an indoor setting), the entire virtual asset appearance has to be manually fine-tuned again. Such a process is, as expected, both tedious and time-consuming.

If the virtual double were modelled using spectral data, appearance continuity under varying illumination would be guaranteed in a spectral rendering system: but, as stated earlier, artists cannot easily work directly with spectral data. Even though the core rendering technology in the VFX industry is now moving towards spectral systems, asset textures will continue to be painted in colour space, as this is the intuitive way for artists to perform such work.

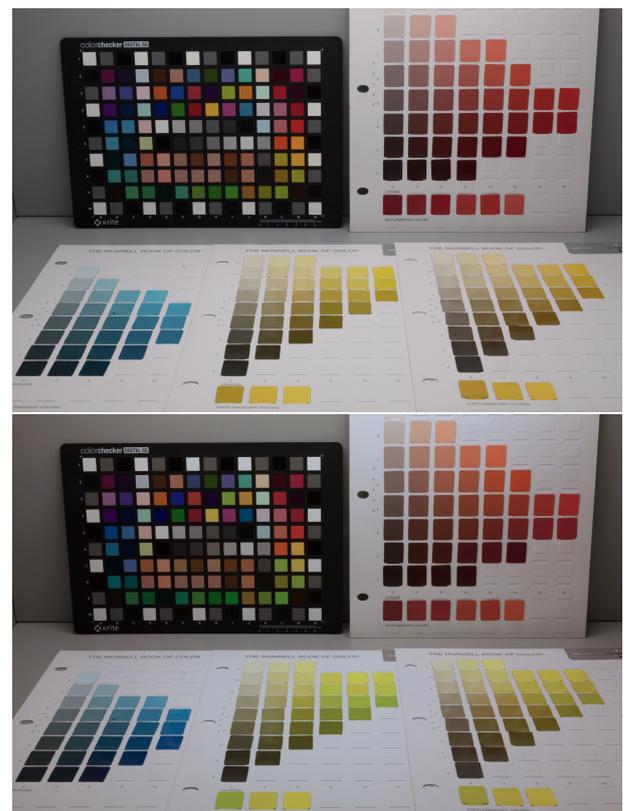


Figure 2: Photos of four Munsell Book of Colour pages (050B, 075Y, 100Y and 075R) in an xRite Judge QC viewing booth. **Top:** daylight **Bottom:** fluorescent light. The photos are white balanced so that the neutral patches on the colour checker match. Note the distortions in the colour gradient on the yellow pages under fluorescent light: this same sort of effect can also be seen in Figure 1, in the images obtained with our uplifting.

Our approach now offers the key advantage that it allows RGB textures to be uplifted so that user-selected key RGB colours evaluate to specific spectra that can be measured on a real asset. These pre-defined mappings between specific RGB colours and spectra are called *constraints* of the uplifting system. RGB colours in close vicinity to the constraints uplift to spectral shapes similar to the original, so that minor texture edits in terms of RGB colour do not cause noticeable visual distortions. All the other parts of the RGB

space return plausible, smooth spectra. This offers the new functionality of exact appearance match of renders with plate footage under varying illumination, and thereby eliminates a substantial amount of tedious appearance fine-tuning work.

2. Previous work

There are two areas of prior work relevant to our technique: spectral uplifting itself, and efficient representations of spectral shapes.

2.1. Spectral Uplifting

The reflectance spectra created by a spectral uplifting process need to satisfy multiple constraints. Their values should fall within the $[0, 1]$ range, the round-trip error caused by uplift and subsequent conversion to RGB should be negligible, and the resulting spectral shapes should qualitatively correspond to real-life materials, which are generally fairly smooth and simple[‡].

Although multiple techniques for spectral uplifting exist, not all of them satisfy the listed criteria.

The technique by MacAdam [Mac35] is capable of creating only blocky spectra, which are not representative of the smooth reflectances usually found in nature. The widely used proposal by Smits [Smi99] is prone to minor round-trip errors, which arise from slightly out-of-range spectra: the process is based on a set of blocky basis functions, and the result is, while usually fairly close, not guaranteed to be in the $[0, 1]$ range. One of the first approaches which produced smooth spectra was proposed by Meng et al. [MSHD15]. Unfortunately, it did not take energy conservation into account, which resulted in colours with no real physical counterpart, i.e. materials that violate the conservation of energy. Otsu et al. [OYH18] introduced a technique that is capable of outperforming most of the existing approaches under specific conditions. Its drawback is its inability to satisfy the $[0, 1]$ spectral range restrictions, which again causes colour errors upon round trips.

Our method builds mainly on the technique proposed by Jakob and Hanika [JH19], which utilises a low-dimensional parametric model for spectral representation in order to create a pre-built uplifting model, which can then be employed during rendering. The structure of the uplifting model is a cube-shaped 3D lookup table, consisting of evenly spaced lattice points representing RGB values. Each of the points contains a mapping to its respective spectral representation, which in the model by Jakob and Hanika consists of 3 sigmoid coefficients. The acquisition of the coefficient sets for individual points is performed during the creation of the model by an optimisation tool, CERES solver [AM*]. Requiring only a set of prior coefficients and the definition of functions which to minimize (in case of Jakob and Hanika, the difference between the reconstructed and the target RGB), the CERES solver is capable of modifying the coefficients to the point where they reconstruct a spectrum that evaluates to the desired RGB value. This process

is referred to as lattice point *fitting*. Jakob and Hanika [JH19] initially fit the coefficients for the center of the cube (i.e. the lattice point with $RGB = (0.5, 0.5, 0.5)$), and let the fitting process gradually fill in all lattice points by using the coefficients of already fitted neighbours as prior for non-fitted points. Their approach produces smooth spectra that satisfy spectral range restrictions with negligible error. Jung et al. [JWH*19] further improve this technique for wide gamut spectral uplifting by introducing new parameters for fluorescence.

None of the existing techniques, however, propose a way in which to constrain the uplifting process to deliver specific spectral shapes, and they also cannot trivially be extended in this direction. The main obstacle is that their spectral representations are simple, and unable to reproduce all possible user-defined spectra.

2.2. Moment-based Spectral Representation

A trivial way to store spectral information is via regular sampling. While being easy to implement and handle, this approach is not very efficient in terms of memory utilisation, especially when high accuracy is desired.

The simple and smooth shape of most real-life reflectance spectra indicate that using a lower-dimensional linear function space, such as the Fourier series, could be the key to their reliably efficient storage. Techniques based on this observation have been studied for representation of both emission spectra [RVHN03] and reflectance spectra [PMHD19a]. However, although the round-trip accuracy is reasonably satisfactory, the resulting spectra do not always have a physical counterpart, as the reconstruction does not obey the $[0, 1]$ constraint needed for physically plausible reflectance spectra.

In addition to linear function spaces, non-linear approaches to spectral representations have also been proposed. The representations are, however, incompatible with linear pre-filtering of textures [PMHD19b].

To eliminate these shortcomings, Peters et al. [PMHD19b] proposed a novel approach to spectral representation. While their representation uses Fourier coefficients (which makes it compatible with linear filtering), the reconstruction is based on the theory of moments (i.e. it is non-linear in that regard). We base our work on this approach, and discuss it in more detail in sections 3.1 and 3.2.

3. Constrained Spectral Uplifting

Similarly to Jakob and Hanika [JH19] and Jung et al. [JWH*19], our uplifting model is also based on a pre-computed RGB coefficient cube that provides mappings of RGB values to their corresponding spectral representations. However, the spectral representation we employ must be capable of accurately representing the user-supplied spectra which we need to reproduce, and this cannot be done with the original sigmoid approach: sigmoids do not have enough degrees of freedom to match complex spectral shapes. Instead, we utilise the moment-based spectral representation [PMHD19b]. In the following subsections, we first provide a brief overview of the principle behind this spectral representation, and then discuss how we fit a moment-based uplift coefficient cube based on the user-supplied target spectra.

[‡] There are exceptions to this, most notably the reflectance spectra of structure colour. But in our work, we concern ourselves with the comparatively smooth spectra found in normal colourants and pigments.

3.1. Obtaining Moment-Based Coefficients

As the shapes of spectra are aperiodic, storing them via Fourier coefficients requires their conversion to a periodic signal for which the Fourier coefficients can then be computed. Although the mapping of wavelength range to the signal can be performed linearly, Peters et al. [PMHD19b] advise against this approach and propose two improvements to the mapping — mirroring of the signal (which eliminates artefacts at the boundaries) and warping (which focuses the reconstruction accuracy on the area around 550nm, i.e. the most important part in terms of colour reconstruction). After testing the proposed methods for the purposes of our uplifting model, we choose to mirror (due to more accurate round-trips), but not warp the signal (due to its incompatibility with our model).

The moment-based coefficients stored for a periodic signal $g(\varphi)$ are then computed as:

$$c = 2\Re \int_{-\pi}^0 g(\varphi) \mathbf{c}(\varphi) d\varphi, \quad (1)$$

where $\mathbf{c}(\varphi)$ is the Fourier basis.

The coefficients c are referred to as *trigonometric moments*. Note that using n_m trigonometric moments for spectral representation implies that $n_c = n_m + 1$ coefficients are actually stored. The additional factor is the zeroth moment c_0 .

3.2. Reconstruction from Moment-Based Coefficients

The process of spectral reconstruction from an input set of trigonometric moments is based on the theory of moments, specifically on Maximum Entropy Spectral Estimate (MESE) [Bur74], which has been shown to produce impressive results when used for the reconstruction of emission spectra. However, as the MESE is not bounded, it cannot be directly used for the reconstruction of reflectance spectra, which must satisfy the $[0, 1]$ range constraint. Therefore, *bounded MESE* has been introduced [PMHD19b], which utilises a duality between bounded and unbounded moment problems formulated in terms of the Herglotz transform. Based on this duality, trigonometric moments can be converted to their corresponding unbounded *exponential moments*, so that the bounded problem represented by the trigonometric moments has a solution if and only if the dual unbounded problem represented by the exponential moments has a solution.

Although this method has been shown to perform rather accurately for even a low number of moments (e.g. 5), reconstruction of complex reflectance spectra with sharp edges often requires a substantially higher coefficient count. By computing the Delta E round-trip error (specifically, the CIE 1976 Delta E) under multiple illuminants for a large ($> 12k$) database of spectra from multiple colour atlases, we empirically determined that for typical reflectance spectra the error stabilises at around $n_m = 20$, i.e. $n_c = 21$.

Naturally, we wish for the reconstruction to be as precise as possible, however, we want to prevent unnecessarily high coefficient counts due to both memory consumption and, as discussed later, also time performance. Therefore, we opt for using a variable number of coefficients for each spectrum. This number is computed with a heuristically-based iterative method — starting from

$n_c = 4$, we check whether the coefficient count is sufficient, and, if not, we increase it and move on to the next iteration, repeating this process up to $n_c = 21$. The adequacy of the representation is determined by its round-trip error under an error-prone illuminant (specifically, FL11).

On average, moment-based representations satisfying our model require 16 coefficients. For most of the constraints, such precision is hardly necessary, and could be decreased if we were to focus on memory utilisation. However, as our goal is to properly assess the accuracy of the uplifting process, we do not concern ourselves with the slight memory overhead.

3.3. Seeding a Moment-Based Uplift Coefficient Cube

In a similar manner than Jakob and Hanika use the center point of the cube as a starting point for the fitting process, we utilise the user-provided constraints. We call this process the *seeding* of the cube. For each of the constraints, it works as follows:

First, we obtain the constraint's trigonometric moment representation c . Then, we find the constraint's position within the RGB cube according to its RGB value. As it is likely that it does not lie exactly on a lattice point, but rather falls inside one of the cube voxels, we assign the obtained coefficients c to all 8 voxel corners. This is done in order to ensure proper reconstruction upon uplifting.

To support larger sets of user-provided spectra, e.g. whole colour atlases, we extend the cube by allowing multiple moment representations per lattice point. We distinguish them by constraint-specific IDs, which are then utilised during the uplifting process for the purposes of identifying the original seed of the voxel.

By seeding the cube, we have placed initial coefficients at some of the lattice points which do not necessarily reconstruct curves that evaluate to the target RGB. Therefore, the coefficients must be modified so that the resulting colour difference is as low as possible.

Our problem of improving the coefficients satisfies the definition of the *Non-linear Least Squares* problem [GNS09]. Non-linear Least Squares is an unconstrained minimization problem in the following form:

$$\underset{x}{\text{minimize}} \quad f(x) = \sum_i f_i(x)^2, \quad (2)$$

where $x = \{x_0, x_1, x_2, \dots\}$ is a parameter block that we are improving (i.e. our coefficients) and f_i are so-called *cost functions* which we want to minimize.

Similarly to Jakob and Hanika [JH19] and Jung et al. [JWH*19], we utilise the CERES solver [AM*] for solving our problem. We define the cost functions as follows:

$$\begin{aligned} f_0(x) &= |target_rgb.R - current_rgb.R| \\ f_1(x) &= |target_rgb.G - current_rgb.G| \\ f_2(x) &= |target_rgb.B - current_rgb.B| \\ f_3(x) &= \sum_{i=0}^s |constraint[i] - current_spectrum[i]|, \end{aligned} \quad (3)$$

where $target_rgb$ is the RGB value of the lattice point, *constraint*

represents a discretised reflectance spectrum of the input constraint, *current_spectrum* represents the spectrum the current coefficients x reconstruct, *current_rgb* is the RGB value of *current_spectrum* and s is the number of samples used for internal representation of reflectance curves.

While the $f_0(x)$, $f_1(x)$ and $f_2(x)$ residuals are identical to the previous approaches, we also add the $f_3(x)$ residual (called the *distance* residual) in order to preserve the input spectral shapes.

3.3.1. Reduced Coefficient Count in Unseeded Cube Parts

As the properties of the input spectra correspond to the initial fits of the lattice points, their number of coefficients is bound to be comparatively high. However, after the initial fitting round that deals with the RGB cube voxels that contain seeds, we do not need to maintain such high coefficient counts: this would be inefficient and memory consuming, in addition to propagating specific spectral shape features beyond the area of the RGB cube where they are actually wanted. Instead, once we leave the initial fitting regions that contain exemplar spectra we want to reproduce, we switch to lower-dimensional moment representations that intrinsically yield smooth spectra not unlike the sigmoids of the original technique [JH19]: the remainder of the lattice points are fit with 3 coefficients only.

The loosened requirements on the spectral shapes of the unseeded lattice points (i.e. we only want them to be smooth and to be within the same metameric family as their neighbors) additionally allow us to eliminate the distance cost function — computing only 3 RGB residuals has the added benefit of lower time complexity.

The conversion of a moment representation c_i of a seeded point to a fitting prior c_j for a non-seeded lattice point is performed by spectral reconstruction of c_i and its subsequent storage with only 3 coefficients. Although this process, called *coefficient recalculation*, causes loss of spectral information, it preserves the rough outline of the curve — essentially, it can be viewed as a low-pass filter. This works to our benefit — it reduces the likelihood of significant colour artefacts between the seeded and non-seeded points, while keeping the spectra smooth.

A problem arises if the lattice point that is being recalculated (denoted P) for the purposes of fitting a non-seeded point (denoted Q) contains multiple moment representations from different metameric families: this is the case if several of its neighbour voxels have been seeded with different spectral shapes. To demonstrate the problem, let us assume two seeded neighbour voxels of P , A and B . Choosing to recalculate the representation corresponding to A would result in visible colour artefacts between Q and the voxel seeded with B , and, vice versa, recalculating only the representation of B would result in colour artefacts between Q and A . As it is our intention to keep the colour transitions within all voxel pairs smooth, we use a spectrum interpolated from all neighbouring seeds as basis of the recalculation instead.

3.4. Interpolation of Metameric Spectra

In the following, we show that the linear combination of two spectra that are metameric under a given light source results in another metameric spectrum. To our best knowledge, this insight, while not

particularly mathematically complex, has not been explicitly stated in graphics literature before.

Let us assume the spectral power distributions of two metamers saved at a lattice point, $P_1(\lambda)$ and $P_2(\lambda)$, that satisfy the conditions

$$\begin{aligned} \int P_1(\lambda)\bar{r}(\lambda)d\lambda &= \int P_2(\lambda)\bar{r}(\lambda)d\lambda \\ \int P_1(\lambda)\bar{g}(\lambda)d\lambda &= \int P_2(\lambda)\bar{g}(\lambda)d\lambda \\ \int P_1(\lambda)\bar{b}(\lambda)d\lambda &= \int P_2(\lambda)\bar{b}(\lambda)d\lambda \end{aligned} \quad (4)$$

where $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$ are the RGB colour matching functions.

Let us express the R component of the RGB value resulting from the linear combination of $P_1(\lambda)$ and $P_2(\lambda)$ as follows:

$$R = \int a \cdot P_1(\lambda)\bar{r}(\lambda)d\lambda + b \cdot P_2(\lambda)\bar{r}(\lambda)d\lambda, \quad (5)$$

where $a + b = 1$

By rewriting this expression and utilising the equality from Equation 4, we get

$$R = a \cdot \int P_1(\lambda)\bar{r}(\lambda)d\lambda + (1 - a) \cdot \int P_1(\lambda)\bar{r}(\lambda)d\lambda, \quad (6)$$

So

$$R = \int P_1(\lambda)\bar{r}(\lambda)d\lambda \quad (7)$$

The same proof can be equivalently applied to the G and B components of the resulting RGB value. Therefore, we conclude that the resulting spectral distribution is also a metamer.

3.5. Using a Moment-Based Coefficient Cube in a Renderer

Discretisation of the RGB space in terms of a cube-like structure poses the problem how to uplift RGB query values for which no direct mapping to a moment representation exists (i.e. which do not directly lie on a lattice point). In such a case, it is reasonable to employ a weighted trilinear interpolation of the data stored at the 8 voxel corners: both Jakob and Hanika [JH19] and Jung et al. [JWH*19] propose interpolating the (in their case: sigmoid) coefficients, mainly due to this being faster than interpolating the reconstructed spectra. In our own implementation of the original sigmoid fitting, we noted that such an interpolation of coefficients indeed works: but interpolation of complete spectra is more accurate even for sigmoids. Additionally, as our technique uses variable coefficients counts, interpolating coefficients within a voxel is generally not an option. Therefore, we always interpolate the reconstructed spectra.

Due to the potential presence of multiple moment representations per lattice point, reconstruction of spectra at such lattice points is not straightforward. If a voxel has been seeded during the construction of the uplifting model, we force exclusive use of the metameric family of the original spectral seed, in order to achieve our goal of matching the input spectra. In all other voxels, for each of its 8 lattice points, we reconstruct spectra of all moment representations that have accumulated there, and interpolate between them in equal ratios: as discussed in section 3.4, this is permissible,

and also yields a metamer for the RGB coordinates of the lattice point. This "hybrid" metamer is then used as input for the 8 corner trilinear interpolation that yields the actual result spectrum for the RGB query value. The reason for this strategy is the same as for the coefficient recalculation, i.e. to provide a smooth transition between metameric families.

Whether the voxel has been seeded or not is determined by means of the unique IDs assigned to the moment representations during the seeding process. If the voxel corners share a common ID, it is the representations corresponding to this ID that must be utilised in order to properly reconstruct the seeds.

4. Results

In the following, we evaluate the accuracy of our technique and compare its results to the sigmoid-based uplift as defined by Jakob and Hanika [JH19]. We start by assessing the quality of our implementation in terms of both accuracy upon uplifting constraints and the colorimetric properties of the uplifting system as a whole. We then provide measurements of both memory utilisation and time performance. All the data of colour atlases and illuminants used in our experiments was provided by ART [Wil18].

4.1. Accuracy of Constrained Uplifting

We tested the accuracy of our proposed constrained uplifting approach on the Munsell Book of Colour (MBOC), which we utilised as a constraint set for a 32^3 -sized coefficient cube for the sRGB colour space. Of the 1598 entries in the MBOC, 1396 are in the sRGB gamut: so our experiments only included those. If a larger RGB space (such as e.g. Adobe RGB) were used as input space, the full number of atlas entries could be used: working with sRGB was not due to inherent restrictions in our proposed technique, but only to stay within the standard RGB space of graphics. Also, in a 32^3 -sized coefficient cube, 82 of the atlas entries can not be used due to the limitation that only one seed can be present in each voxel. Increasing the cube resolution would allow to use those 82 missing entries: using a 32^3 cube was only done to keep memory usage and pre-computation times reasonable, not because the cube dimension specifically needs to be a power of two.

We compared the difference between the uplifted spectra to the original ones via the standard CIE Delta E colour difference metric. In the following, we do not quote the results for the original fitting illuminant CIE D65, as accuracy is excellent in that case anyway (negligible Delta E throughout). The interesting case is how well colour appearance matches under CIE FL11, which we used as an example of a spiky fluorescent illuminant that is qualitatively similar (but not identical) to the fluorescent lamp used in the xRite Judge QC viewing booth (see Figure 2).

Even under this illuminant with low colour rendering index (read: with a high propensity to bring out metameric failures in materials), the average round-trip error is just a Delta E of 0.21. As the maximum error perceivable by a standard observer is $\Delta E < 1$ [MT11], this value is negligible and can be regarded as highly satisfactory. Of the 1314 entries, only 22 were found to return a $\Delta E > 1$: all of these were RGB values extremely close to (0, 0, 0).

An explanation of this behaviour in the dark region of the RGB cube can be seen in Figure 3: the reason is an inadequate fitting of seeded points. As the trigonometric moments are, in their nature, Fourier coefficients, they are mainly suitable for representing sinusoidal-like signals. When the optimiser works with them, it tends to amplify these shapes rather than create a constant signal (see Figure 4). As the spectral shapes of dark colours often resemble almost constant lines close to zero, it makes sense that the optimiser would primarily fail there. But note that the error caused by this "failure" is still barely (if at all) perceptible in most cases, and that special case handling could probably be introduced to eliminate this behaviour.

We compare the effect of our uplifting technique on the resulting curve shapes in Figures 4 and 5, where we provide comparisons of curves of the seeded and uplifted spectra. While for darker colours (see Figure 4), the errors are mainly due to inadequacies of the optimiser, slight curve deviations for brighter colours are usually due to imperfections of the moment representation.

But as already stated, these deficiencies do not significantly degrade the accuracy of the uplifting system. In Figure 3, we can clearly see that our technique provides reasonable results throughout, while images uplifted with the sigmoid-based cube demonstrate rather significant colour errors. Except for slight, barely visible discrepancies, results of our uplifting process are almost identical to the original spectral render.

4.2. Uplift Consistency Across RGB Space

In order to assess how our technique uplifts the entire RGB gamut (and not just the regions around the seeds), we created multiple coefficient cubes that were seeded with different colour atlases. This included an atlas with a single starting constraint at RGB(0.5, 0.5, 0.5), i.e. fitted with the same sole starting constraint as the sigmoid-based approach by Jakob and Hanika [JH19]).

We first compared their performance in terms of colour reconstruction with regard to uplifting a gradient texture. We selected a gradient with saturated colours in the red-yellow-green region, as that is where the differences are most perceivable. The behaviour of these gradient textures is shown in Figure 6, again illuminated by CIE FL11 (under D65, there are, as per the fitting process, practically zero differences). While the distinctions between individual uplifts are barely perceivable by the human eye, the difference images demonstrate some variations — mainly around the locations of seed points, which is precisely what is intended by constraining the uplifting process. To be specific, the most evident artefacts can be observed for the Pantone constraint set, especially in the orange region. We attribute this to its large number of distinct, complex input spectra (i.e. spectra from different metameric families) which, however, evaluate to RGB values in close vicinity. This makes them exhibit metameric discontinuities similar to that of the 075Y and 100Y pages of the Munsell Book of Colours (see Figure 2).

None of the gradient textures, however, exhibit any significant discontinuities, which indicates that our interpolation approach works properly in the presence of multiple metameric families of reflectance spectra.

Figure 7 demonstrates that our approach can properly uplift large

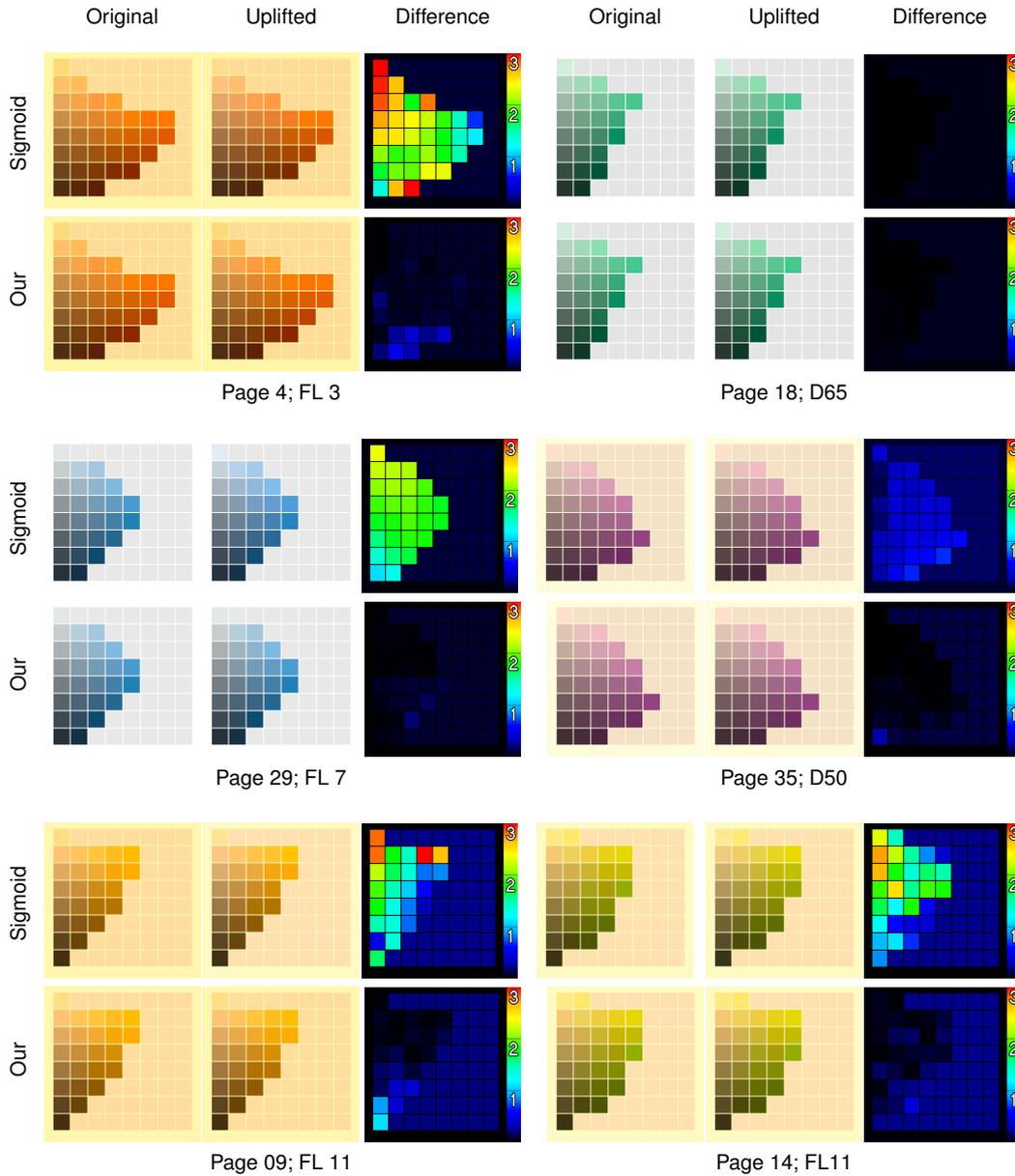


Figure 3: Comparison of our uplifting model with the sigmoid-based technique [JH19] for pages of the Munsell Book of Colour under different illuminants. For each of the constrained uplifts, we utilise a different moment-based cube seeded with the colour patches from that specific page. Maximum Delta E in the difference images is 3. Note that patches that fall outside sRGB have been omitted from the pages.

regions of the RGB gamut without showing artefacts under varying illuminations. We provide multiple renderings of a rainbow texture that covers most of the RGB gamut, uplifted with various constraint sets under distinct illuminants. Even though a considerable subset of all voxels was seeded and the remainder was filled in with smoother spectra (as described in section 3.3.1), none of the renderings exhibits significant artefacts.

4.3. Performance and Future Work

In this section, we evaluate the performance of our method in terms of both memory and execution time, and propose possible future work for their improvement.

4.3.1. Memory Usage

The memory necessary for storing our cube depends on its resolution (i.e. the number of lattice points), which is, in turn, dependent on both the size of our constraint set and on the position of its spec-

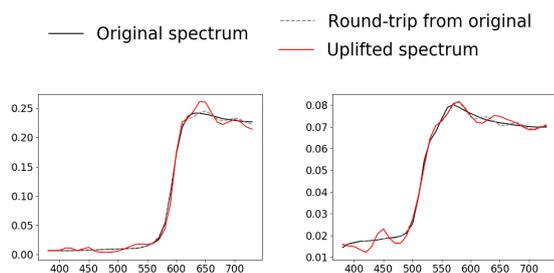


Figure 4: Problematic cases of constrained uplifting: reflectance spectra of darker colours. The spectral curve labelled round-trip is a direct reconstruction of the spectrum from the moments used to seed the fitting process. The red curve labelled uplift is the result of querying the final coefficient cube, including interpolation within the voxel the RGB value lies in. All three curves should be identical: separating "round-trip" and "uplift" shows whether any error that is present is caused by the moment-based representation, or the interpolation in the coefficient cube. While the round-trip from the moment-representation used for the purposes of seeding is reasonable (in the areas where the round-trip plot is barely visible, it mimics the original spectrum), the uplifted spectrum demonstrates visible deviations due to the optimiser's failure to properly fit at least one of the voxel corners. Note that the demonstrated range on the y-axis is not $[0, 1]$, i.e. the deviations are ultimately still quite small, and do not lead to visible colour shifts.

tra within the RGB cube. The Macbeth Colour Checker (MCC), which contains only 24 entries that are spaced quite far apart from each other in the RGB space (with one of them falling outside of sRGB), requires as little as a 13^3 -sized cube. Due to the close proximity of some seeds, the 1396 sRGB entries of the Munsell Book of Colour would require as much as 340 lattice points per axis for all seeds to fall into a unique voxel. Additionally, due to the rigid nature of an evenly spaced voxel grid, using a higher cube dimension does not necessarily imply more points that can be successfully seeded. Due to voxel edges being in different positions for different cube dimensions, increasing cube size might even have an adverse effect — for example, while a cube of size 90^3 is sufficient for the RAL Design atlas, in a 300^3 -sized cube, 1 point remains unfitted due to a voxel collision.

To store the coefficients of cube entries, we require 3 floating point values for all non-seeded points, and, on average, 16 floating point values per constraint. For the 340^3 -sized cube required for the proper coverage of the Munsell Book of Colour, this would yield a size of over 450.35MB. Although using less coefficients for storing constraints is possible, it would not noticeably improve the size of the cube — even if we were to use 3 coefficients for all coefficient representations within the cube, the overall size would still be over 449.8MB. That is a negligible improvement, and the overall size remains excessive — after all, the seeding of the whole Munsell Book of Colour requires only 1396 voxels, which sums up to a maximum of $8 \cdot 1396$ lattice points. Additionally, while most of the regions of the cube are barely utilised, there exist some that have all of their voxels fitted, which might result in a lack of smooth colour transitions within these regions.

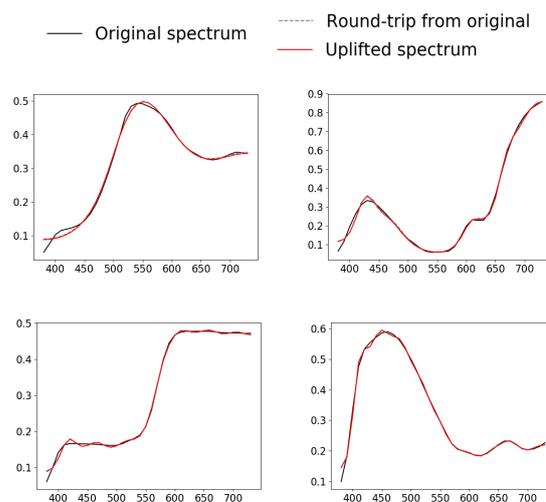


Figure 5: Accuracy of constrained uplifting demonstrated on examples of input spectra that correspond to saturated RGB colours. The types of spectral curves shown in the plots are the same as in Figure 4. As one can see, and as opposed to Figure 4, the uplifted spectra follow the spectra reconstructed from the original coefficients quite precisely, and also match the input spectra quite well.

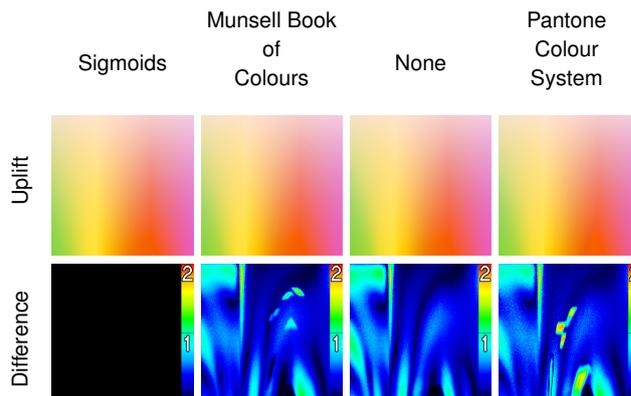


Figure 6: A region of the RGB gamut uplifted with different sets of input spectra, illuminated by FL11. The difference images are a result of comparison with the sigmoid uplift and serve as a tool for demonstrating the effects of different constraint sets on the resulting uplift. Note that the difference images are relative to maximum $\Delta E = 2$.

We therefore conclude that, for the purposes of constrained spectral uplifting for large sets of user-supplied target spectra (like e.g. entire colour atlases), a coefficient cube with evenly spaced lattice points is distinctly sub-optimal, in terms of both memory requirements and its resulting colorimetric properties. As future work, we will investigate a dynamic structure like e.g. kD-trees, which can split the RGB space into variably-sized voxels according to the number of constraints.

However, it has to be noted that the approach presented in this paper works perfectly fine for constraint sets with up to several

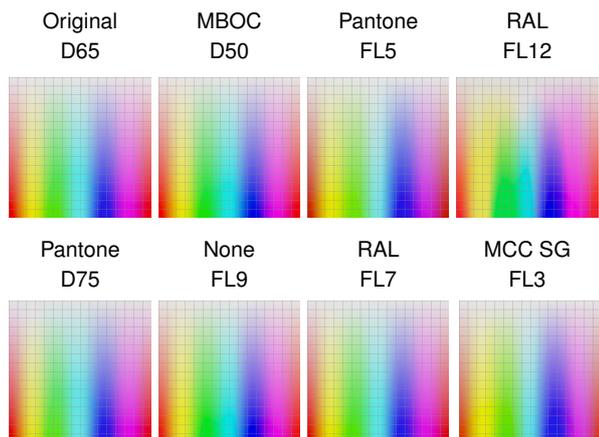


Figure 7: Constrained spectral uplifting of a colourful texture for various constraint sets under different illuminants. Note that all uplifts were performed with 32^3 -sized cubes, i.e. some of the constraints may not have been utilised due to collisions during seeding. Under D65, all uplifts match.

dozen, or even low numbers of hundred, data points. This is sufficient for typical usage in VFX scenarios, where only a few key assets (like e.g. the main colours of the costume of a lead character) are measured on set, in order to later constrain the spectral uplift of virtual doubles.

4.3.2. Execution time

Since our uplifting model is created prior to the rendering process, we divide the evaluation of the execution time into two parts — the cube fitting process, and the rendering speed when utilising our cube for uplifting purposes.

We test the execution time of cube fitting on multiple sets of constraints in forms of colour atlases, and present the results in Table 1. All the experiments are performed on an Intel Core i7-8750H CPU (12 logical cores), and the size of each cube is 32^3 .

| Colour atlas | Number of seeds | Seeding time | Fitting time | Overall |
|--------------|-----------------|--------------|--------------|------------|
| MBOC | 10512 | 7h 32m 18s | 6m 59s | 7h 39m 17s |
| RAL | 1336 | 43m 11s | 9m 15s | 52m 26s |
| Macbeth SG | 576 | 27m 14s | 8m 8s | 35m 22s |
| Macbeth | 184 | 9m 29s | 14m 17s | 23m 46s |
| None | 1 | 0.08s | 45m 5s | 45m 6s |
| Sigmoid | 1 | 0.01s | 7m 16s | 7m 16s |

Table 1: Fitting time of a 32^3 -sized coefficient cube for multiple colour atlases. As this cube resolution may be insufficient for the utilisation of all constraints in a given atlas (see section 4.3.1), we also provide the number of seeds that were placed on lattice points.

Due to the higher coefficient count and the strict requirements placed upon the shapes of the reconstructed spectral curves, the fitting of seeded lattice points takes a lot longer than the fitting of the latter (on average, a seeded point takes 2.4 seconds to fit, in comparison to the 0.03 for those which are not seeded). However, as

the cube fitting process is multi-threaded, it benefits from multiple starting points evenly positioned across the RGB cube. This is particularly obvious when comparing the performance of the fitting of the Macbeth charts and the fitting without constraints.

None of these usage cases outperform the sigmoid-based cube in terms of fitting time. While our technique needs to use various complex mathematical operations, such as the application of Levinson’s algorithm, Herglotz transform and multiple other conversion processes [PMHD19b], not to mention the interpolation of metamers for lattice points with multiple representations, the sigmoid-based approach evaluates the spectral curve from a coefficient set with as little as six floating point operations for any given wavelength [JH19].

This drawback of our technique also carries over to using our method during rendering. In order to properly evaluate the execution time, we perform two tests — first, we compare the performance of the sigmoid-based cube with the performance of our non-constrained cube (in order to avoid the overhead of spectral reconstruction from higher-dimensional coefficient representations), and secondly, we provide measures of the execution time of the constrained uplift of the renders in Figure 3. All experiments are performed on an Intel Xeon CPU E5-2680 v3 (48 logical cores), with 32^3 -sized cubes.

For the latter experiments, the sigmoid-based approach performed, on average, 2.1 times better than our constrained cubes. The performance overhead arising from constraining the uplifting process was expected. When used in a spectral renderer — specifically, ART [Wil18] — on a closely viewed texture of one of the pages of the Munsell Book of Colour (that is, when pretty much all the pixels in the image show the constrained texture), rendering times slow down by about a factor of 4 when compared to the sigmoid uplift.

For non-constrained uplifting, our method provides no benefits compared to the sigmoid-based approach, except perhaps that it creates slightly more varied spectral shapes. Even when not constrained, uplifting of the colourful textures shown in Figure 7 is 2.3 times slower with our cube than with the sigmoid-based technique.

That having been said, we wish to point out that so far, our focus was placed on the correctness and accuracy of the constrained uplift, i.e. our implementation of both the model creation and its utilisation in a renderer does not include any real optimisations yet. In the future, these could be applied to the fitting process (by further exploiting the possibilities of the CERES solver, or, possibly, another optimisation technique) and to the actual uplifting, which currently does not cache any intermediate values (such as the exponential moments) and therefore requires them to be unnecessarily re-computed during each uplift. We estimate that such optimisations could improve the performance up to a factor of two, both during fitting and during rendering.

We also note that our approach, while being both slower to fit and slower to render than the sigmoid technique, offers the unique capability of targeted uplifting, which was simply not available before. As such, we deem a somewhat slower performance compared to the sigmoid-based uplift to be an acceptable price to pay.

5. Conclusion

We presented the first method capable of constraining the spectral uplifting process with an arbitrary set of target spectra. By utilising a trigonometric moment-based approach for spectral representation, the RGB values of the target spectra are accurately uplifted to their original spectral shapes, while the rest of the RGB gamut uplifts to smooth spectra. This results in smooth transitions between the various metameric families that originate from the constraining process.

Our model shows a slight weakness when uplifting very dark colours, which we attribute to the inability of moment-based spectral representations to represent constant spectra, and to the CERES solver, which only amplifies this deficiency. However, even including these minor drawbacks, the results in terms of colour accuracy are noteworthy, as the uplifted curves describe the original ones with negligible differences (even the errors for very dark colours are acceptable).

However, neither the memory, nor the execution time of the uplifting processes of our model are optimal: the new and so far unique capability to perform targeted uplifts comes at the cost of some overhead that is not present in e.g. the unconstrained sigmoid uplift technique of Jakob and Hanika [JH19].

In the future, we will primarily focus on utilising a more suitable and memory efficient structure for storing the constraints, such as a kD-tree or an octree. Secondly, we will improve the time execution by optimising the moment reconstruction process.

Acknowledgements

We acknowledge funding by the Czech Science Foundation grant 19-07626S, the EU H2020-MSCA-ITN-2020 grant number 956585 (PRIME), and the Charles University grant number SVV-260588.

References

- [AM*] AGARWAL, SAMEER, MIERLE, KEIR, et al. *Ceres Solver*. <http://ceres-solver.org> 3, 4.
- [Bur74] BURG, JOHN PARKER. “Maximum entropy spectral analysis”. *Astronomy and Astrophysics Supplement* 15 (1974), 383 4.
- [GNS09] GRIVA, IGOR, NASH, S, and SOFER, A. “Nonlinear Least Squares Data Fitting”. *Linear and Nonlinear Optimization* (2009), 743–758 4.
- [JH19] JAKOB, WENZEL and HANIKA, JOHANNES. “A Low-Dimensional Function Space for Efficient Spectral Upsampling”. *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, 147–155 3–7, 9, 10.
- [JWH*19] JUNG, ALISA, WILKIE, ALEXANDER, HANIKA, JOHANNES, et al. “Wide gamut spectral upsampling with fluorescence”. *Computer Graphics Forum*. Vol. 38. 4. Wiley Online Library. 2019, 87–96 3–5.
- [Mac35] MACADAM, DAVID L. “The theory of the maximum visual efficiency of colored materials”. *JOSA* 25.8 (1935), 249–252 3.
- [MSHD15] MENG, JOHANNES, SIMON, FLORIAN, HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. “Physically meaningful rendering using tristimulus colours”. *Computer Graphics Forum*. Vol. 34. 4. Wiley Online Library. 2015, 31–40 3.
- [MT11] MOKRZYCKI, WS and TATOL, M. “Colour difference delta E-A survey”. *Mach. Graph. Vis* 20.4 (2011), 383–411 6.
- [OYH18] OTSU, HISANARI, YAMAMOTO, MASAFUMI, and HACHISUKA, TOSHIYA. “Reproducing spectral reflectances from tristimulus colours”. *Computer Graphics Forum*. Vol. 37. 6. Wiley Online Library. 2018, 370–381 3.
- [PMHD19a] PETERS, CHRISTOPH, MERZBACH, SEBASTIAN, HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. “Spectral Rendering with the Bounded MESE and sRGB Data”. *Workshop on Material Appearance Modeling*. Vol. 2019. 2019, 07–09 3.
- [PMHD19b] PETERS, CHRISTOPH, MERZBACH, SEBASTIAN, HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. “Using moments to represent bounded signals for spectral rendering”. *ACM Transactions on Graphics (TOG)* 38.4 (2019), 1–14 3, 4, 9.
- [RVHN03] ROMERO, JAVIER, VALERO, EVA, HERNÁNDEZ-ANDRÉS, JAVIER, and NIEVES, JUAN L. “Color-signal filtering in the Fourier-frequency domain”. *JOSA A* 20.9 (2003), 1714–1724 3.
- [Smi99] SMITS, BRIAN. “An RGB-to-spectrum conversion for reflectances”. *Journal of Graphics Tools* 4.4 (1999), 11–22 3.
- [Wil18] WILKIE, ALEXANDER. *The Advanced Rendering Toolkit*. <http://cgg.mff.cuni.cz/ART>. 2018 6, 9.