



Wide Gamut Moment-based Constrained Spectral Uplifting

L. Tódová,¹ A. Wilkie¹ and L. Fascione^{2,*}

¹Faculty of Mathematics and Physics, Department of Software and Computer Science Education, Charles University, Prague, Czech Republic
todova@gmail.com, wilkie@cgg.mff.cuni.cz

²NVIDIA, Santa Clara, California, USA
l.fascione@gmail.com

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. In a recent publication, we have proposed 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. We extend this previous work, which supported the sRGB gamut only, by describing a method that is able to constrain any spectrum from within the gamut of realisable reflectances. 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.

Keywords: rendering, reflectance and shading models rendering

CCS Concepts: • Computing methodologies → Reflectance modelling

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. 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.

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

*Work conducted while at Weta Digital.

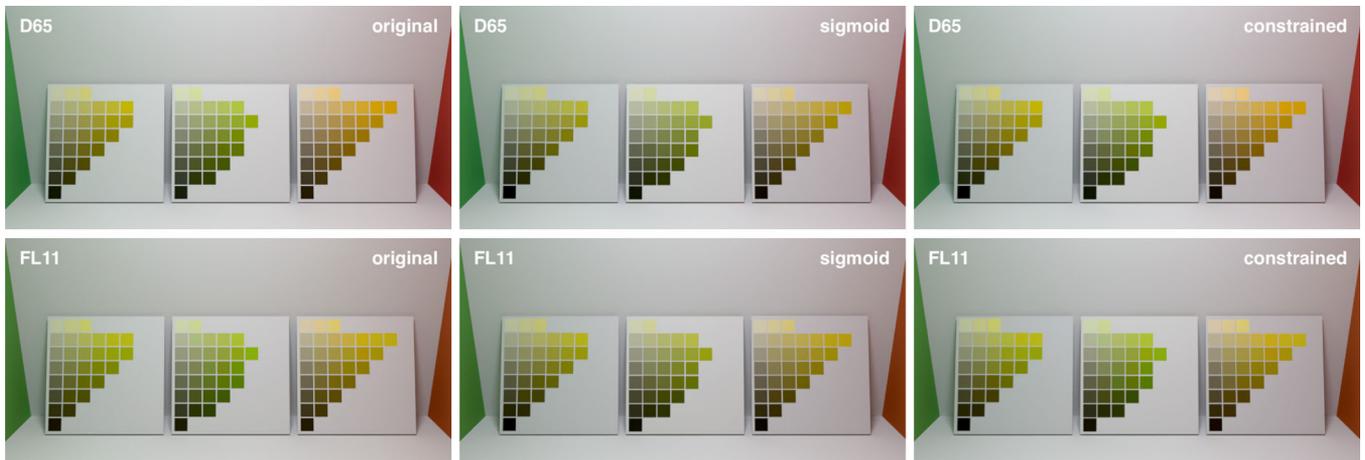


Figure 1: Pages from the Munsell Book of Colour under illuminant CIE D65 (top row) and CIE FL11 (bottom row). **Left column:** the materials in the scene are defined via their spectral properties, so no uplifting is needed. **Middle column:** the materials are defined as wRGB values and uplifted with the sigmoid method. Note that no metamerism failures occur in the bottom row: the colour gradient stays ordered. **Right column:** The materials are defined as wRGB 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 metamerism disruptions of the colour sequence as real Munsell Book of Colour pages do. Note that white balance was applied to both sets of images.

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 digitized 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 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 with spectral data directly. 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.

In previous work [TWF21], we proposed an uplifting method that allowed sRGB textures to be uplifted so that user-selected key sRGB colours would evaluate to pre-measured spectra. We termed these pre-defined mappings between specific sRGB colours and spectra of the *constraints* of the uplifting system. Using this approach, sRGB colours in close vicinity to the constraints uplift to spectral shapes similar to the original, so that minor texture edits in terms of sRGB colour difference do not cause noticeable visual distortions. All the other parts of the sRGB 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.

However, while our previous publication proved that the basic approach works, and its contribution was valuable insofar as we addressed a problem with no previous solution, it was capable of uplifting within the sRGB gamut only. But in order to fully utilize our approach in the VFX industry, we must be able to constrain any value within the gamut of realisable reflectances.

In this paper, we extend our previous publication by addressing this aspect. We propose a method supporting larger gamuts, and demonstrate it on Adobe Wide Gamut RGB. We describe the technical challenges this extension poses, and discuss possible solutions. Our approach exhibits similar performance as in our previous work, with the additional benefit of providing implicit gamut mapping even outside the gamut of realisable reflectances, i.e. plausibly uplifting all values within Adobe Wide Gamut RGB.

Furthermore, we perform an extended evaluation of our method, which now also explores the behaviour of constraints that previously had to be omitted due to sRGB limitations. These include saturated colours along with colours from the edges of the chromaticity diagram. Even for values such as these, the performance of our method is satisfactory.

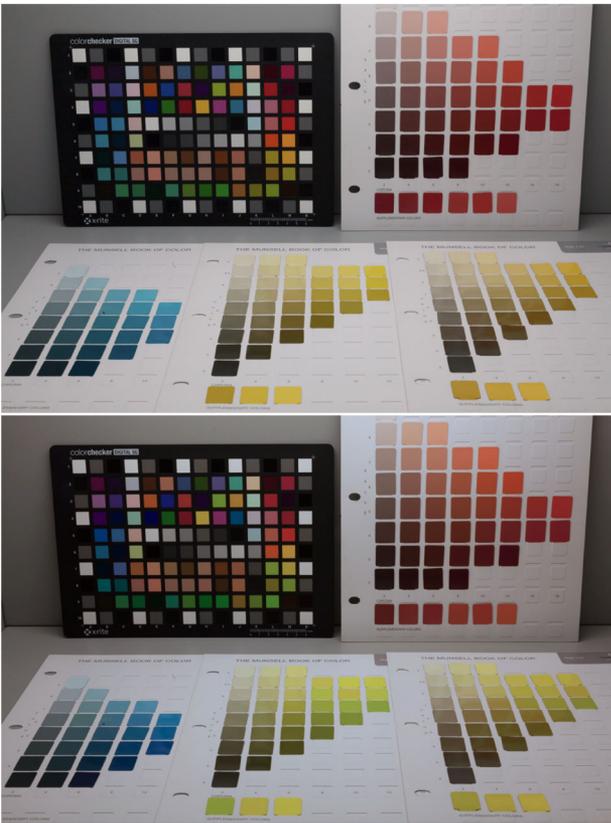


Figure 2: Photos of four Munsell Book of Colour pages (050B, 075Y, 100Y and 075R) in an xRite Judge QC viewing booth. **Top:** daylight. **Bottom:** fluorescent light. The photos are white balanced so that the neutral patches on the colour checker match. Note the distortions in the colour gradient on 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.

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 smooth and simple.

Although multiple techniques for spectral uplifting exist, not all of them satisfy the listed criteria. They also differ in the RGB gamut they are capable of uplifting with negligible error, and the round-trip error they obtain both outside of this gamut.

The technique by MacAdam [Mac35] is capable of creating only blocky spectra, which are not representative of the smooth re-

flectances usually found in nature. The widely used proposal by Smits [Smi99] is prone to small 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. While the errors are usually not perceivable for the sRGB gamut, these inaccuracies are accentuated when uplifting other RGB values and the method is rendered unstable for larger gamuts.

One of the first approaches which both produced smooth spectra and also adequately uplifted larger gamuts was proposed by Meng *et al.* [MSHD15]. However, since the spectra were created without taking energy conservation into account, they had to be re-scaled in order to reproduce colours with real physical counterparts. This resulted in colour shift, and therefore, round-trip errors. 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 and a subsequent inability to correctly uplift larger gamuts.

Our method builds mainly on the technique proposed by Jakob and Hanika [JH19], which utilizes 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 three 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 centre 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 is targeted for uplifting the sRGB gamut and produces smooth spectra that satisfy spectral range restrictions with negligible round-trip error for the sRGB gamut, and acceptable error for a variety of other colour spaces. 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 shapes 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. However, as stated by Peters *et al.* [PMHD19], although the round-trip accuracy for reflectance spectra is reasonably satisfactory, the resulting spectra do not always have a physical counterpart, as the reconstruction does not obey the $[0,1]$ restriction needed for physically plausible reflectance spectra.

In addition to linear function spaces, non-linear approaches to spectral representations have also been proposed. These representations are, however, incompatible with linear pre-filtering of textures [PMHD19].

To eliminate these shortcomings, Peters *et al.* [PMHD19] 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.

2.3. Gamut mapping

As opposed to the sRGB gamut, wide gamut RGB spaces contain values that cannot be realized as reflectances and therefore cannot be uplifted to viable reflectance spectra that would evaluate to their respective RGB values. In order to support the uplifting of such values, they must first be mapped into the gamut.

Gamut mapping is the process of modifying colour values to become displayable in a given limited target colour space. To do so, multiple methods exist: in the context of ICC profiles [Int10], these are referred to as *rendering intents*. They differ not only in their approach towards the out-of-gamut values, but some also affect the overall appearance of the entire image.

A *perceptual* rendering intent scales the entire source colour space in order to match the destination space. While this preserves the relative visual relationships between the colours, the resulting image is often noticeably less saturated than the source. The *saturation* rendering intent works in a similar manner, it also enhances the saturation of the new colours in order to provide more vibrant results.

Colorimetric rendering intents, on the other hand, only modify out-of-gamut values—although strictly speaking, this is only true with the *absolute colorimetric* intent. The *relative colorimetric* intent also adjusts white and optionally also black point of the target space.

However, as can be seen in Figure 3, even if restricted to only modifying out-of-gamut values, different trajectories can be used to move out-of-gamut colours to the border of the displayable space. As reviewed by Mokrzycki *et al.* [MF98], the colour values can either be moved along the axis connecting them to the centre of the destination gamut (*node clipping*), or they can decrease their

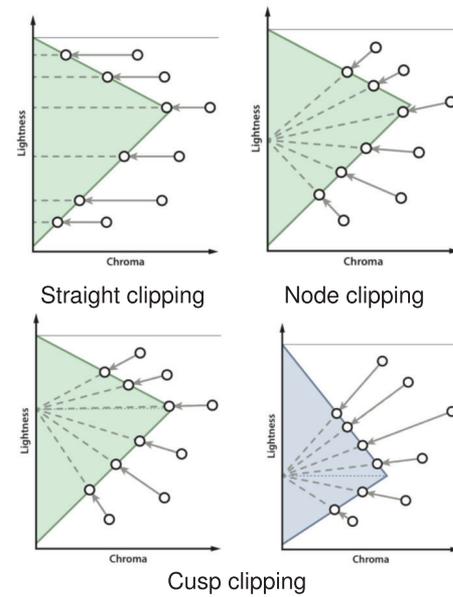


Figure 3: Possible approaches to the absolute colorimetric rendering intent. While the straight clipping approach preserves the luminance and decreases chroma, the node and cusp clipping approaches project the colours towards the neutral axis, retaining chroma but losing highlight information. Note that the cusp clipping is visualized for two different hues.

chroma component with constant luminance until they fall inside the gamut (*straight clipping*). While node clipping is susceptible to losing highlight information, straight clipping tends to create artificial highlights. Additionally, in order to improve the node clipping approach, *cusp clipping* has been proposed. There, the focus point is at the luminance of the most saturated colour for each hue. Although its results are better than those obtained with the node clipping approach, it is still susceptible to similar problems.

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. This cannot be done with the sigmoid approach as proposed by Jakob and Hanika [JH19], as it lacks enough degrees of freedom to match complex spectral shapes. Instead, we utilize the moment-based spectral representation [PMHD19]. 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.

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.* [PMHD19] 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 550 nm, 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 $|m|$ trigonometric moments for spectral representation implies that $|c| = |m| + 1$ 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 [PMHD19], which utilizes 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 2000 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 stabilizes at around $|m| = 20$, i.e. $|c| = 21$.

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 $|c| = 4$, 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 $|c| = 21$. The adequacy of the representation is determined

by its round-trip error under an error-prone illuminant (specifically, FL11). In contrast to our previous publication, we utilized the CIE Delta E 2000 metric instead of the RGB Euclidean distance. This eliminated our previous issues with inaccuracies in the dark regions of the cube.

Since the maximum allowed CIE Delta E 2000 error can be defined by the user, we cannot, in advance, determine the average number of coefficients needed for a moment-based representation. However, as concluded later (see Section 4.4.1), the memory overhead resulting from increased accuracy is negligible.

3.3. Seeding a moment-based uplift coefficient cube

In a similar manner that Jakob and Hanika use the centre point of the cube as a starting point for the fitting process, we utilize 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 eight 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 utilized 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 minimisation 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 utilize 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 discretized 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 three 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 neighbours) additionally allow us to eliminate the distance cost function—computing only three 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 three 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.

The last detail we discuss is how to perform coefficient recalculation of lattice points that contain multiple moment representations. Let us assume lattice point P and its two seeded neighbour voxels, A and B . After seeding of the cube, P contains two moment representations, one from the metameric family of A , the other from B . Let us then assume a non-seeded lattice point Q , and let Q be a neighbour of P . Now, let us use the information provided by P as prior for fitting the moment representation of Q .

If we choose to employ only the representation corresponding to A , we can expect visible artefacts between Q and the voxel B . Similarly, recalculating only the representation of B would result in colour artefacts between Q and A . Therefore, in order to keep the colour transitions within all voxel pairs smooth, we first interpolate the spectra reconstructed from all coefficient representations of P and we use the result of this interpolation as a basis of coefficient recalculation. In this specific case, this translates to the reconstruction of 2 spectra stored at point P (one belonging to moment repre-

sentation of A , the other of B) and their linear interpolation resulting in a new spectrum that is subsequently stored with only three coefficients.

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 of 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 eight 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 eight 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 eight 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 utilized in order to properly reconstruct the seeds.

3.6. Support of larger RGB gamuts

Since the RGB spaces are bounded orthogonal systems, the shape of their gamut is a cube. In the case of sRGB, the RGB cube is at the same time also the gamut of possible reflectances *within sRGB*. However, with other larger RGB gamuts, this is no longer the case. For example, for the Adobe Wide Gamut RGB (wRGB) gamut, the subset of colours that correspond to valid surface reflectances is a convex, curved shape and, as Figure 4 shows, considerable parts of the wRGB cube do not correspond to colours that can be realized by plain reflectance spectra. These can only be realized via emission, or reflectance with a fluorescence component [JWH*19].

In this section, we describe our method of supporting targeted uplifting within larger RGB gamuts, and specifically demonstrate it for the Adobe Wide Gamut RGB gamut.

It is obvious that if a coefficient cube approach like in our previous work [TWF21] is used, a significant number of the cube’s lattice points fall out of gamut, so no exactly matching reflectance spectra can be synthesized for them. Although a simple solution would be to discard the out-of-gamut points and flag their voxels as ‘empty’, this would cause issues during uplifting in the voxels containing the gamut’s boundary. We call these voxels the *boundary* voxels, since a part of their values lies within the gamut and the rest is outside. If, during uplift, a wRGB value falls into the boundary voxel, our system has no means of interpolating between its corners, since not all of them contain reflectance values. Therefore, even these values would have to be discarded, which in turn results in incomplete support of larger gamuts.

We address this issue by applying a gamut mapping technique, which assigns the out-of-gamut lattice points coefficients of a boundary RGB value closest to them. This results in plausible, al-

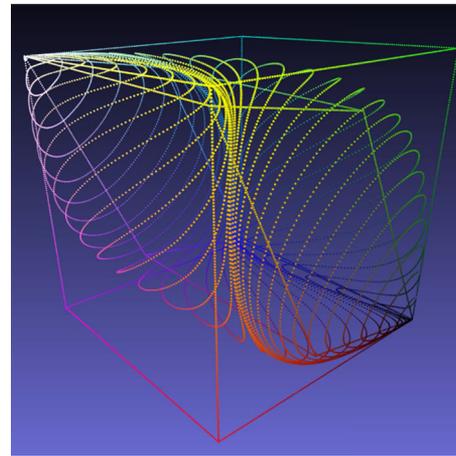


Figure 4: The gamut of realisable reflectances, the sRGB and the wRGB coefficient cubes. While the sRGB cube only covers a part of the gamut, the wRGB cube encompasses most of it (note that small parts are sticking out of the cube, mainly in the yellow and cyan regions). The transparent parts of the cube do not correspond to colour values that can be realized as plain reflectance spectra: these RGB values need to be mapped to the gamut of realisable reflectances.

beit slightly incorrect colours in the boundary voxels, and has the additional benefit of also supporting other out-of-gamut RGB values (which is valuable if e.g. an artist paints a texture that is too colourful). Following, we explain the applied gamut mapping technique in more detail.

3.6.1. Our approach to gamut mapping

Gamut mapping is performed as the final step of the cube creation, after the fitting of the lattice points. For its purposes, we utilize an absolute colorimetric intent—specifically, we use the node clipping approach described in Section 2.3). Since the goal of this paper is to preserve the exact colours of the constraints, employing any kind of global mapping approach is not an option, and, although we believe straight clipping would also provide satisfactory results, we have opted for node clipping in order to preserve saturation. Cusp clipping was not used due to being more complex and offering only moderately better results (if that).

In order to map an RGB value into the gamut, we first find the line from the entry to the centre of the cube. The RGB value we wish to map our out-of-gamut entry to lies on this line. To obtain it and its respective coefficients, we apply a recursive approach as described in Algorithm 1.

The recursive division of the line is only meaningful while its endpoints are in different voxels. Since we can expect the boundary voxel to already show colour inconsistencies due to some of its corners being out of gamut, once we reach this voxel in our division process, we can terminate the search. The initial *depth* parameter is, therefore, set to $depth = \log_2(diagonal)$, where *diagonal* is the number of voxels on the diagonal of the cube.

Algorithm 1. Gamut mapping

```

1: function MOVETOGAMUT(center, outer, depth)
2:   middle.rgb  $\leftarrow$  (center.rgb + outer.rgb)/2
3:   if depth > 0 then
4:     if middle is in gamut then
5:       MoveToGamut(middle, outer, depth - 1)
6:     else
7:       MoveToGamut(center, middle, depth - 1)
8:     end if
9:   else
10:    if middle is in gamut then
11:      final.rgb  $\leftarrow$  middle.rgb
12:    else
13:      final.rgb  $\leftarrow$  center.rgb
14:    end if
15:    final.coeff  $\leftarrow$  fit from voxel corner of final.rgb
16:    return final
17:  end if
18: end function

```

Once we find the RGB value we want our entry to map to, we obtain its coefficients (see step 15 of Algorithm 1). As prior, we use the coefficients of the corner of the voxel the RGB value falls into. If this lattice points have multiple coefficient representation, we perform coefficient recalculation in the same manner as we did during the fitting (see Section 3.3.1).

3.7. Wide gamut sigmoid-based uplift

Although the paper by Jakob and Hanika [JH19] focuses mainly on spectral uplifting within the sRGB gamut, we extend their work by creating a wRGB sigmoid-based uplifting cube. We do this solely in order to provide comparisons of a non-constrained approach to ours and to showcase the benefits of constraining the uplifting process.

The extension is done in exactly the same manner as ours for the moment-based cube. For lattice points outside of the supported gamut, we apply gamut mapping by recursively moving the points into the gamut, and subsequently running the CERES solver in order to obtain their coefficients. The only difference to our approach is that, since the points have only one fit, no coefficient recalculation needs to be performed.

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 assess the plausibility of our gamut mapping technique for wide gamut uplifting. Lastly, we provide measurements of both memory utilisation and time performance. All the data of colour atlases and illuminants used in our experiments were provided by ART [Wil18].

4.1. Accuracy of constrained uplifting

We determined the accuracy of our proposed constrained uplifting approach by measuring the round-trip error of a large constraint set: the Munsell Book of Colour (MBOC). As difference metric, we used CIE Delta E 2000.

As expected, since the cube was fitted with regard to the daylight CIE D65 illuminant, the round-trip error under D65 was negligible. However, examining the behaviour of our model under more complex, spiky illuminants, similar to the fluorescent illuminant used in the xRite Judge QC viewing booth (see Figure 2), proved to be more interesting.

In our experiments, we utilized all 24 non-LED CIE standard illuminants, including the five high pressure discharge lamp ones. For all constraints and illuminants, we measured both the average round-trip ΔE error and the percentage of round-trips that resulted in $\Delta E \geq 1$. In order to assess the effect of the input parameters on the performance of our uplifting model, we ran the experiments for multiple cubes with different resolutions, as well as different error allowed during the storage of spectra with moments. We also ran the same experiments for sRGB as well as wRGB. Note that not all constraints were utilized for every experiment—especially for smaller cubes, constraints with similar RGB values fall into the same voxels, so that only one of them can be used (see section 4.4.1 for more detail). Additionally, for sRGB, constraints falling outside of the gamut had to be omitted.

For the purposes of comparing the constrained uplifting model to the non-constrained approaches, we also ran the same experiments for the sigmoid cubes as proposed by Jakob and Hanika [JH19].

The results, presented in Table 1, show slight improvement in both the average error and the percentage of erroneous uplifts (i.e. uplifts that resulted in $\Delta E \geq 1$) with increasing cube resolution and also decreasing permitted error during constraint storage. We attribute the higher errors in cubes with lower resolution to the higher distance between the original constraints' positions and the corners of their respective voxels during the seeding process. Because of this, the prior coefficients passed to the optimizer are further away from the desired coefficients, allowing the optimizer more room for error. The maximum allowed error parameter only accentuates this deficiency, as the coefficients are already stored with less precision.

While the results do not show significant differences between sRGB and wRGB cubes, we can observe that the wRGB cubes perform slightly better for almost all cases. This may initially seem contradictory, mainly due to our statement in the previous paragraph. As the wRGB cube is noticeably larger than the sRGB cube, its voxels are also larger than those of the sRGB cube, subsequently resulting in larger distances between the constraints and their seeded voxel corners. Additionally, especially for cubes with lower resolutions, the fitting process of seeded points in the wRGB cubes occasionally fails and the spectra need to be stored with three coefficients only, which does not happen at all for the sRGB gamut.

We attribute this behaviour to the shapes of the spectra: according to our observations, constant spectral shapes are more prone to failure. This is because the trigonometric moments are, in their nature, Fourier coefficients, and are therefore not well-suited for

Table 1: The average Delta E 2000 error and the percentage of constraints' round-trips with $\Delta E \geq 1$, measured both for the sRGB and the wRGB gamut, for different sets of input parameters as well as for sigmoid cubes as defined by Jakob and Hanika [JH19]. The constraint set utilized was the Munsell Book of Colour, and the illuminants under which the experiments were performed were the CIE standard illuminants and the HP illuminants. Note that not all entries of MBOC were utilized for the experiments due to collisions of constraints in cube voxels.

| sRGB | | | | | | | | | | |
|--------------------|-------|-------------------|-------|-------------------|-------|-------------------|-------|-------------------|-------|-------------------|
| Max ΔE Dim | 8 | | 16 | | 32 | | 64 | | 128 | |
| | Avg | $\Delta E \geq 1$ |
| 2.0 | 0.52 | 0.1461% | 0.52 | 0.1520% | 0.53 | 0.1547% | 0.51 | 0.1425% | 0.50 | 0.1373% |
| 1.0 | 0.40 | 0.0866% | 0.41 | 0.0912% | 0.41 | 0.0919% | 0.39 | 0.0853% | 0.38 | 0.0805% |
| 0.5 | 0.32 | 0.0541% | 0.27 | 0.0379% | 0.25 | 0.0308% | 0.24 | 0.0275% | 0.24 | 0.0268% |
| 0.25 | 0.26 | 0.0325% | 0.17 | 0.0148% | 0.14 | 0.0086% | 0.13 | 0.0061% | 0.13 | 0.0050% |
| 0.1 | 0.20 | 0.0330% | 0.11 | 0.0083% | 0.07 | 0.0027% | 0.06 | 0.0010% | 0.06 | 0.0009% |
| 0.05 | 0.17 | 0.0282% | 0.07 | 0.0066% | 0.06 | 0.0026% | 0.05 | 0.0006% | 0.04 | 0.0002% |
| Sigmoid | 46.73 | 0.9581% | 47.89 | 0.9581% | 50.87 | 0.9581% | 56.9 | 0.9581% | 68.05 | 0.9581% |
| wRGB | | | | | | | | | | |
| Max ΔE Dim | 8 | | 16 | | 32 | | 64 | | 128 | |
| | Avg | $\Delta E \geq 1$ |
| 2.0 | 0.65 | 0.1716% | 0.50 | 0.1282% | 0.45 | 0.1114% | 0.43 | 0.1074% | 0.42 | 0.1010% |
| 1.0 | 0.58 | 0.1377% | 0.42 | 0.0885% | 0.37 | 0.0733% | 0.35 | 0.0651% | 0.34 | 0.0605% |
| 0.5 | 0.50 | 0.1115% | 0.31 | 0.00472% | 0.25 | 0.0304% | 0.23 | 0.0239% | 0.22 | 0.0221% |
| 0.25 | 0.51 | 0.1140% | 0.25 | 0.0315% | 0.15 | 0.0177% | 0.13 | 0.0054% | 0.12 | 0.0042% |
| 0.1 | 0.52 | 0.1205% | 0.20 | 0.0324% | 0.09 | 0.0093% | 0.06 | 0.0021% | 0.05 | 0.0005% |
| 0.05 | 0.52 | 0.1360% | 0.18 | 0.0280% | 0.08 | 0.0086% | 0.05 | 0.0021% | 0.04 | 0.0003% |
| Sigmoid | 57.79 | 0.9559% | 54.21 | 0.9580% | 56.28 | 0.9555% | 57.51 | 0.9555% | 61.18 | 0.9558% |

representing a constant signal: a deficiency the optimizer further amplifies. As the shapes of colours with low saturation tend to resemble constant lines, as opposed to the more complex spectra of saturated colours, it is expected that their reconstruction is more prone to error.

While the optimizer's failures during the fitting process may appear concerning, this rarely happens—for cubes with $res = 32$ (i.e. of size 32^3), we observed only around four failures during the seeding of the whole constraint set (containing 891 entries, i.e. $891 * 8 = 7128$ seeds). Additionally, if the fitting does fail, it is always for the voxel corners that are farthest away from the constraints—in our case, these usually had the interpolation weight lower than 0.1, meaning that their influence on the uplifted spectral shape was minimal. Realistically, this deficiency can only be observed for the wRGB cubes with $res = 8$, which demonstrate even worse results than their sRGB counterparts. Therefore, in case wide gamut support is required, we advise on setting the resolution parameter to at least $res = 32$.

However, overall, utilising cubes with even smaller resolutions and higher allowed error appears promising. As the maximum error perceivable by a standard observer is $\Delta E < 1$ [MT11], even the average Delta E of 0.53 is highly sufficient for reproduction purposes, and while the model cannot guarantee no human perceptual discrepancies, these can be minimized by tweaking the input parameters.

With regard to the performance of the sigmoid model, the improvement is significant, even for cubes with low resolutions. While our model results in only an extremely small portion of the constraints with the round-trip error of $\Delta E \geq 1$, for the sigmoid cubes, this number is as high as 96%. Closer examination has revealed that

the round-trip measurements resulting in $\Delta E < 1$ were all obtained under the CIE D65 illuminant, i.e. the illuminant under which the sigmoid cubes were fitted. However, note that while the average Delta E error for the sigmoid cubes is rather high, this does not make the cubes unusable for regular uplifting purposes: for our experiments, we specifically chose complex, spiky spectra that expose the constraints' susceptibility to metameric artefacts.

We compare the effect of our uplifting technique on the resulting curve shapes in Figure 5, where we provide comparisons of curves of the seeded and uplifted spectra. We can observe that the differences are negligible, and that the slight deviations from original curves are mainly due to the imperfections of the optimizer rather than the moment representation.

In Figure 6, we present the results of comparison between the sigmoid-based uplifts and our constrained uplifts on specific pages of the Munsell Book of Colour. We set the resolution of our cubes to 64, because it is a sufficient enough size for all constraints to be utilized, and we set the maximum allowed Delta E to 0.1. While the results are satisfactory enough even for higher Delta E parameter, we consider the slight memory and performance overhead resulting from higher coefficient representations (see Section 4.4) to be worth the higher constraint uplifting accuracy. For this reason, we also set the maximum allowed Delta E parameter to 0.1 for other experiments performed in this paper. Due to our experiments presented in Table 1, setting the parameter to even lower values does not noticeably improve the constraint uplifting accuracy.

We can clearly see that our technique provides reasonable results throughout, while images uplifted with the sigmoid-based cube demonstrate rather significant colour errors. Although our model

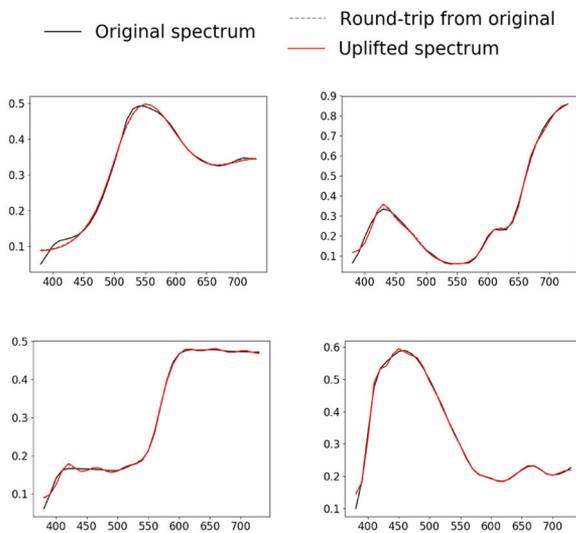


Figure 5: Accuracy of constrained uplifting demonstrated on examples of input spectra. The spectral curve labelled round-trip is a direct reconstruction of the spectrum from the moments used to seed the fitting process. The red curve labelled uplift is the result of querying the final coefficient cube, including interpolation within the voxel the RGB value falls into. 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. As one can see, the uplifted spectra follow the spectra reconstructed from the original coefficients quite precisely, and also match the input spectra pretty well. Note that the demonstrated range on the y-axis is not $[0,1]$, i.e. the deviations are even less perceptible than demonstrated.

does not completely eliminate human-perceivable errors, these occur exclusively for extreme illuminants, such as HP2, which excessively amplify even the smallest deviations that occur due to the moment-based representation.

Other small deviations in the uplifting process are expected both because of the imprecision of the moment-based representation and due to the failures of the optimizer during the seeding process. These can be observed mainly for the darker colours, since their spectra tend to resemble constant lines close to zero. However, note that the error caused by this ‘failure’ is still barely (if at all) perceptible.

4.2. Uplift consistency across RGB space

In order to assess how our technique uplifts the entire supported RGB gamuts (and not just the regions around the seeds), we created multiple uplift 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 highly saturated sRGB colours, and we uplifted it

for all selected uplift cubes under a spiky, error-prone illuminant, specifically HP2 (under D65, there are, as per the fitting process, practically zero differences). The choice for the HP2 illuminant, the sRGB gamut and the saturated texture was motivated by the fact that under these conditions, the differences between individual uplifts are most perceivable. While the wRGB gamut shows similar differences, some of the values of the texture may fall inside the boundary voxels and therefore, their uplifts are less pronounced than if they were properly inside the gamut. Similarly, the differences are present even under other illuminants (or for other textures), but are less noticeable.

In Figure 7, we compare the resulting uplifts. While the distinctions between them are barely perceivable by the human eye, the difference images between individual pairs 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 MBOC constraint set, especially in the light green region, and for the RAL and MCCSG constraint sets in the saturated red region. We contribute this to the distinct, complex input spectra (i.e. spectra from different metamer families) in these regions 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 8 demonstrates that our approach can properly uplift large regions of the RGB gamut without showing artefacts under varying illumination. We provide multiple renderings of a rainbow texture, uplifted with various constraint sets under distinct illuminants both for the sRGB and the wRGB gamut. 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 exhibit significant artefacts.

4.3. Performance of gamut mapping

While some of the wRGB textures shown in the preceding sections already contained colours outside the gamut of realisable reflectances, the performance of the gamut mapping technique has not yet been explicitly analysed. In order to do so, we created a texture with values falling out of wRGB, uplifted it under the D65 illuminant and subsequently applied gamut mapping. We present the results in Figure 9.

As anticipated, there is a visible difference between the original texture and the gamut-mapped uplift in the places of the out-of-gamut values. Since the RGB values in close proximity to each other in terms of hue are mapped to extremely similar values, the gradients in the texture’s out-of-gamut areas tend to fade out, giving rise to areas with constant colours. Therefore, the farther away the out-of-gamut value is from the gamut boundary, the higher the difference between the original and the uplifted RGB value (see also Figure 9). This is an expected and desired behaviour of local gamut mapping techniques. Additionally, while the mapped values in

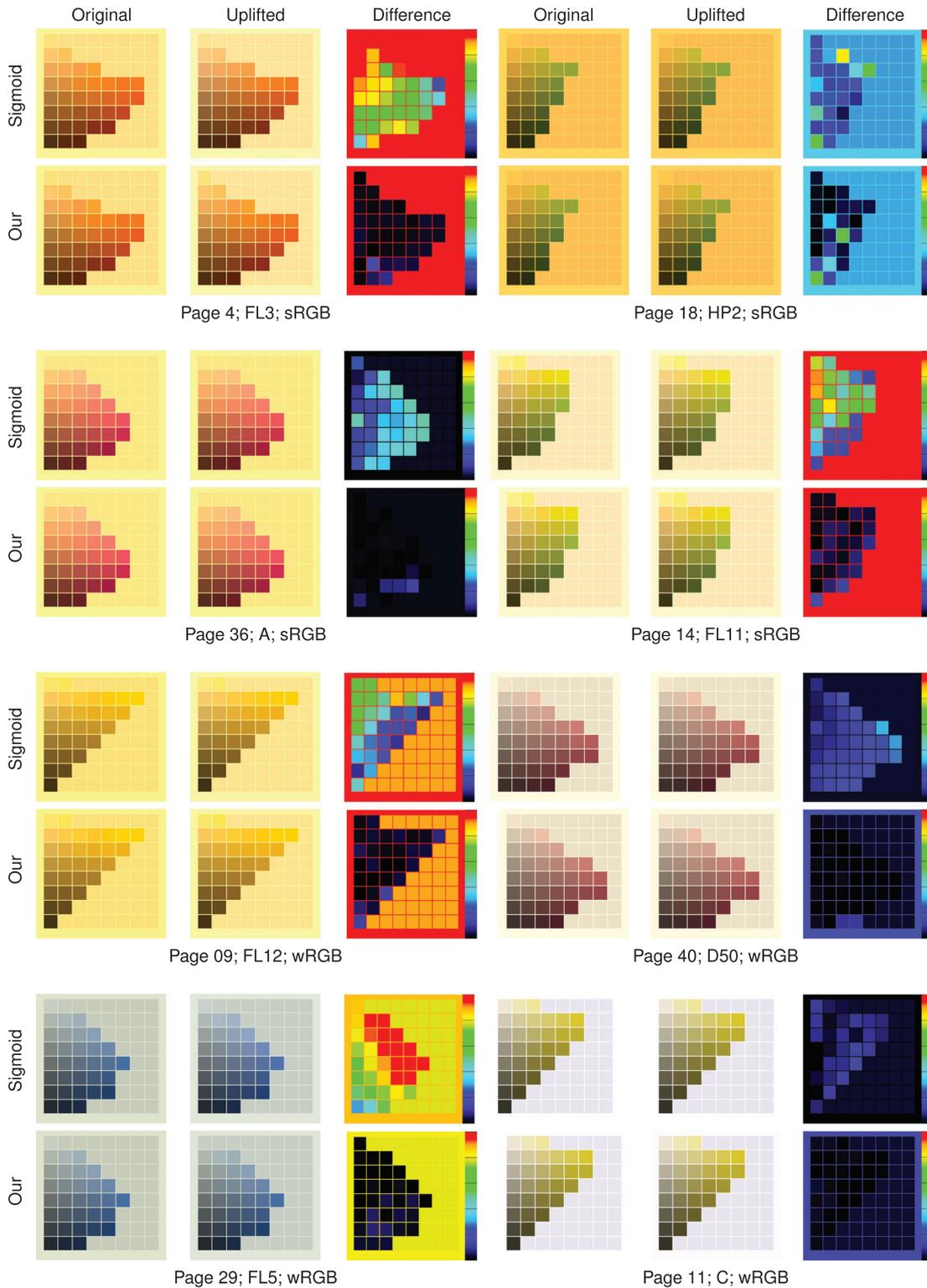


Figure 6: Comparison of our uplifting model with the sigmoid-based technique [JH19] for pages of the Munsell Book of Colour under different illuminants. For each of the constrained uplifts, we utilize a different moment-based cube seeded with the colour patches from that specific page, with the maximum delta E parameter set to 0.1 and the resolution parameter set to 64. The difference images are relative to maximum $\Delta E_{00} = 3$. Note that patches that fall outside sRGB have been omitted for the first four visualized pages.

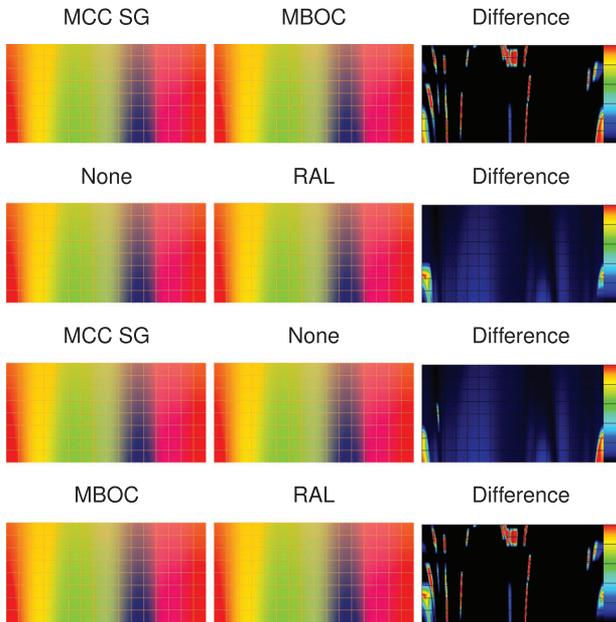


Figure 7: A region of the sRGB gamut uplifted with different sets of input spectra, illuminated by HP2. The difference images between selected pairs serve as a tool for demonstrating the effects of different constraint sets on the resulting uplift. Note that the difference images are relative to maximum $\Delta E = 2$, i.e. the differences are perceivable by the human eye.

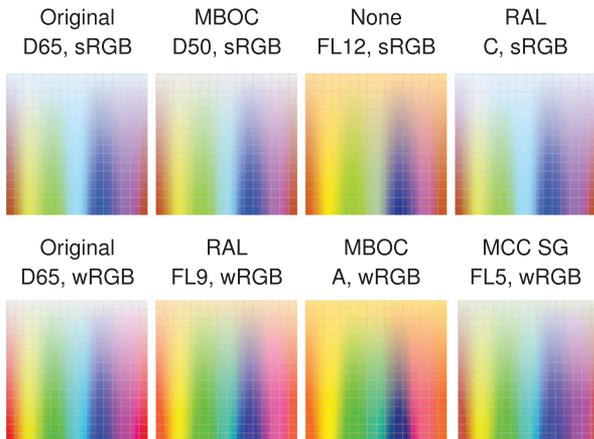


Figure 8: Constrained spectral uplifting of a colourful texture for various constraint sets under different illuminants. Note that all uplifts were performed with 64^3 -sized cubes, i.e. some of the constraints may not have been utilized due to collisions during seeding. Also note that the colour gamuts of the uplifts are scaled so that wRGB fits into the standard sRGB gamut that can be shown on screen. This results in the desaturated appearance of the sRGB uplifts.

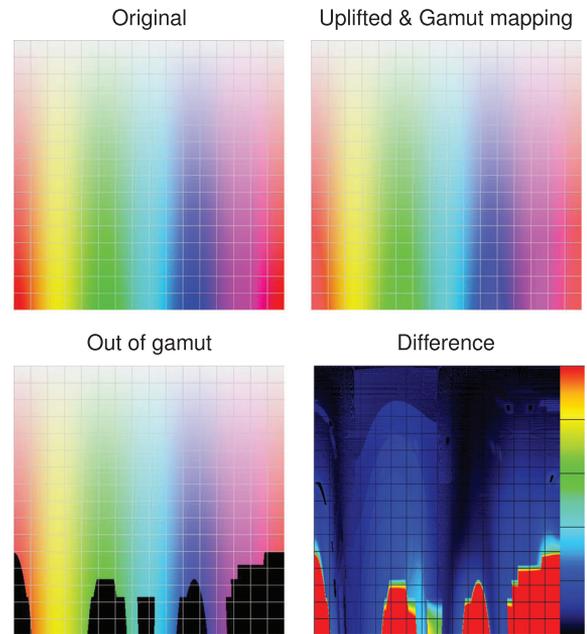


Figure 9: The behaviour of our gamut mapping strategy shown on a wRGB texture with out-of-gamut values. The image labelled Out of gamut highlights the values from Original texture that fall outside of wRGB, while the image labelled Uplifted & Gamut mapping was obtained by uplifting the Original texture under the D65 illuminant with a cube of size 64^3 and by subsequently applying our gamut mapping strategy. The Difference image shows the difference between the Original and the Uplifted textures. Note that it is relative to $\Delta E = 1$, and that the differences in the images are also perceivable by the human eye. Also note that not all values shown in the Out of gamut image necessarily fall out of gamut—some of them fall inside the boundary voxels, which also require gamut mapping.

areas of similar hues are similar, they are not identical, which prevents undesirable artefacts such as banding.

However, while the performance of our gamut mapping strategy is satisfactory, it is often utilized in cases where it should not be necessary, i.e. inside the gamut in the boundary voxels. This is not as problematic for cubes with higher resolutions, where the boundary voxels take up a small portion of the cube. However, if we were to use a cube with e.g. $res = 8$ in order to uplift the texture from Figure 9, we would find that 31% of the image's voxels fall out of gamut in contrast to the 11% for a 64^3 -sized cube utilized for the purposes of Figure 9. To give a better example, for a cube of size 8^3 , we estimate that around 22% of the wRGB gamut is not supported and needs to be mapped, in comparison to roughly 3% for a cube of size 64^3 .

For future work, we propose solving this problem by either creating a dynamic structure that is capable of encompassing the whole wRGB gamut, or by utilising the fitted lattice points on the corners of the boundary voxels to determine the desired uplifted spectrum (e.g. by employing the optimizer). Currently, if high precision in

saturated colours is needed, we recommend utilising a cube with at least $res = 64$.

4.4. 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.

4.4.1. 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 spectra within the RGB cube. For example, 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 for both the sRGB and the wRGB gamut. 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, and, for its 1597 wRGB entries, the cube resolution would need to be as high as 420 and even then, one constraint would remain unseeded. Additionally, due to the rigid nature of an evenly spaced voxel grid, using a higher cube resolution does not necessarily imply more points that can be successfully seeded. Due to voxel edges being in different positions for different cube resolutions, increasing cube size might even have an adverse effect—for example, while an sRGB cube of size 90^3 is sufficient for the sRGB entries of 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 three floating point values for all non-seeded points, and, for a cube with maximum allowed ΔE set to 0.1, an average of 16 floating point values per constraint. For the 340^3 -sized sRGB cube required for the proper coverage of the Munsell Book of Colour, this would yield a size of over 450.35MB. Although decreasing the maximum allowed ΔE parameter and therefore using less coefficients for storing constraints is possible, it would not noticeably improve the size of the cube—even if we were to use three 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 all sRGB entries of the 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 utilized, there exist some that have all of their voxels fitted, which might result in a lack of smooth colour transitions within these regions.

We, therefore, conclude that, for the purposes of constrained spectral uplifting for large sets of user-supplied target spectra (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 e.g. kD-trees, which can split the RGB space into variably sized voxels according to the number of constraints.

Table 2: Fitting time of a 32^3 -sized sRGB coefficient cubes with $max_{\Delta E} = 0.1$ for multiple colour atlases. As the chosen cube resolution may be insufficient for the utilisation of all constraints in a given atlas (see Section 4.4.1), we also provide the number of seeds that were placed on lattice points. Note that as the performance metric, we use the wall clock time.

| Colour atlas | Number of seeds | Seeding time | Fitting time | Overall |
|--------------|-----------------|--------------|--------------|-------------|
| MBOC | 7672 | 33 min 53 s | 42 s | 34 min 35 s |
| RAL | 1072 | 4 min 59 s | 43 s | 5 min 42 s |
| Macbeth SG | 560 | 2 min 48 s | 45 s | 3 min 33 s |
| Macbeth | 184 | 1 min 43 s | 1 min 15 s | 2 min 58 s |
| None | 1 | 0.03 s | 4 min 00 s | 4 min 00 s |
| Sigmoid | 1 | 0.01 s | 24 s | 24 s |

However, it has to be noted that the approach presented in this paper works perfectly fine for constraint sets with up to several dozen, or even low numbers of hundred, data points. This is sufficient for typical usage in VFX scenarios, where only a few key assets (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.

4.4.2. 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 creation 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 obtained for the sRGB gamut in Table 2. The size of each cube is 32^3 , and the maximum allowed Delta E is set to 0.1. All the experiments are performed on an Intel Xeon CPU E5-2680 v3 (48 logical cores).

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 significantly longer than the fitting of the latter (on average, a seeded point takes 0.27 s to fit, in comparison to the 0.004 s 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.

Other factors affecting the performance of cube creation are the user-defined parameters, i.e. the resolution of the cube and the maximum allowed Delta E. While it is obvious that the fitting time will increase with increasing cube resolution, it is noteworthy that this does not happen for the seeding time. For example, for a cube of size 8^3 constrained with MBOC ($\Delta E = 0.1$), the average fitting time of a seeded point is 0.64 s, while for its 128^3 -sized variant, it is only 0.11 s. This is due to larger voxels in cubes with lower resolutions. As already observed when examining the decreased accuracy of constraint uplifting in cubes with smaller resolutions (see Section 4.1), larger voxels imply increased distance between the constraints' RGB values and the RGB values of the voxel corners. This causes greater difference between the constraints' spectral shapes

and the desired spectral shapes of the seeded points, subsequently resulting in more iterations of the optimizer.

Decreasing the maximum allowed Delta E parameter also results in higher seeding time, due to higher number of coefficients per representation and higher required precision, both of which strain the optimizer. However, for $\Delta E = 1$, representations with only four coefficients are generally sufficient, and therefore, setting the parameter above the value 1 does not noticeably improve performance.

For wRGB cubes, the ratio between their seeding and the fitting time is almost identical to their sRGB counterparts. However, in their case, the performance of cube creation is also affected by gamut mapping.

The time it takes to move a point inside the gamut is identical for each lattice point, and is only dependent on the number of recursive iterations, i.e. on the resolution of the cube. Since the gamut always takes up the same portion of the cube (in case of wRGB, roughly 20% of voxels fall out of gamut—however, this number decreases with higher resolutions due to smaller portion of cube voxels being on the gamut boundary), the overall performance of gamut mapping is, therefore, dependent only on the cube size and on the time spent in one iteration of the recursive gamut mapping process. By performing experiments similar to those presented in Table 2 on the wRGB cubes, we have observed that the overall time of gamut mapping is 12 s for 32^3 -sized cubes. From this knowledge and the knowledge of how many lattice points fall out of gamut, we can compute this value for any other cube resolution—for example, for cubes with $res = 16$, the time spent on gamut mapping is only 3 s, and for the cubes of size 128^3 , this is 155 s. We have performed additional experiments for wRGB cubes of other resolutions to support this claim.

In terms of overall performance, for the unconstrained cubes, the gamut mapping process takes up around 7% of the whole cube creation process. While this might be slightly higher for their constrained variants (since their fitting benefits from multiple starting points), overall, the time spent on gamut mapping is acceptable and, in contrast to the seeding process, even negligible.

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 [PMHD19], not to mention the interpolation of metamers for lattice points with multiple representations and a non-trivial check on whether the current voxel has been seeded, 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 second, we provide measurements of the execution time of the constrained uplift of the renders in Figure 6. The testing conditions are identical to the previous experiment—the resolution of the tested cubes is set to $res = 64$, and the maximum allowed $\Delta E = 0.1$. How-

ever, we have observed that cubes with different user-defined settings perform similarly.

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 four 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 8 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.

In terms of colour accuracy, the results of our uplifting model are noteworthy, as the uplifted curves describe the original ones with negligible differences. We also propose an approach with which to apply our method to larger gamuts, where we achieve similar results.

However, neither the memory, nor the execution time of the uplifting process 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]. Additionally, our support of wRGB is not complete, since, for our recommended settings, an estimated 3% of the wRGB gamut is uplifted to less saturated values.

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. We believe that such a structure would also enable us to cover larger gamuts more efficiently. Second, 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 S., MIERLE K.: Ceres solver. <http://ceres-solver.org> (2022).
- [Bur74] BURG J. P.: Maximum entropy spectral analysis. *Astronomy and Astrophysics Supplement 15* (1974), 383.
- [GNS09] GRIVA I., NASH S., SOFER A.: Nonlinear least squares datafitting. In *Linear and Nonlinear Optimization*. SIAM - Society for Industrial and Applied Mathematics (2009), pp. 743–758.
- [Int10] International Color Consortium: Specification ICC.1:2010, profile version 4.3.0.0. 2010. https://www.color.org/specification/ICC1v43_2010-12.pdf (2010).
- [JH19] JAKOB W., HANIKA J.: A low-dimensional function space for efficient spectral upsampling. In *Computer Graphics Forum38* (2019), pp. 147–155.
- [JWH*19] JUNG A., WILKIE A., HANIKA J., JAKOB W., DACHSBACHER C.: Wide gamut spectral upsampling with fluorescence. In *Computer Graphics Forum38* (2019), pp. 87–96.
- [Mac35] MACADAM D. L.: The theory of the maximum visual efficiency of colored materials. *JOSA* 25, 8 (1935), 249–252.
- [MF98] MONTAG E. D., FAIRCHILD M. D.: Gamut mapping: Evaluation of chroma clipping techniques for three destination gamuts. In *Proceedings of the Color and Imaging Conference* (1998), vol. 1998, Society for Imaging Science and Technology, pp. 57–61.
- [MSHD15] MENG J., SIMON F., HANIKA J., DACHSBACHER C.: Physically meaningful rendering using tristimulus colours. In *Computer Graphics Forum34* (2015), pp. 31–40.
- [MT11] MOKRZYCKI W., TATOL M.: Colour difference delta E—A survey. *Machine Graphics and Vision* 20, 4 (2011), 383–411.
- [OYH18] OTSU H., YAMAMOTO M., HACHISUKA T.: Reproducing spectral reflectances from tristimulus colours. In *Computer Graphics Forum37* (2018), pp. 370–381.
- [PMHD19] PETERS C., MERZBACH S., HANIKA J., DACHSBACHER C.: Using moments to represent bounded signals for spectral rendering. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- [RVHN03] ROMERO J., VALERO E., HERNÁNDEZ-ANDRÉS J., NIEVES J. L.: Color-signal filtering in the Fourier-frequency domain. *JOSA A* 20, 9 (2003), 1714–1724.
- [Smi99] SMITS B.: An rgb-to-spectrum conversion for reflectances. *Journal of Graphics Tools* 4, 4 (1999), 11–22.
- [TWF21] TÓDOVÁ L., WILKIE A., FASCIONE L.: Moment-based constrained spectral uplifting. In *Proceedings of the EGSR (DL)* (2021), A. Bousseau and M. McGuire (Eds.), Eurographics Association, pp. 215–224.
- [Wil18] WILKIE A.: The Advanced Rendering Toolkit. <http://cgg.mff.cuni.cz/ART> (2018).