Metameric

Spectral Uplifting via Controllable Color Constraints

Van De Ruit, Mark; Eisemann, Elmar

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Metameric: Spectral Uplifting via Controllable Color Constraints

Mark van de Ruit
m.vanderuit-1@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Elmar Eisemann
e.eisemann@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Figure 1: Left. Spectral renderings of the same scene under four standard illuminants. This scene contains RGB textures which have been authored for spectral uplifting with our method. Top-right. Metameric mismatching between two objects, which are constrained to remain identical under D65 and A illuminants. Middle-right. Metamerism on a surface with exaggerated reflectances constrained to the theoretical boundaries of a metamer mismatch volume, such that a message becomes visible only under the FL11 illuminant. Bottom-right. A more realistic example of generated spectral reflectances using our method.

ABSTRACT

Spectral rendering is a crucial solution for photorealistic rendering. However, most available texture assets are RGB-only, and access to spectral content is limited. Uplifting methods that recover full spectral representations from RGB inputs have therefore received much attention. Yet, most methods are deterministic, while, in reality, there is no one-to-one mapping. As a consequence, the appearance of uplifted textures is fully determined under all illuminants. Hereby, metamers, which are materials with differing spectral responses that appear identical under a specific illumination, are excluded.

We propose a method which makes this uplifting process controllable. Hereby, a user can define texture appearance under various lighting conditions, leading to a greatly increased flexibility for content design. Our method determines the space of possible metameric manipulations and enables interactive adjustments, while maintaining a set of user-specified appearance constraints. To achieve this goal, we formulate the problem as a constrained optimization, building upon an interpolation scheme and data-based reflectance generation, which maintain plausibility. Besides its value for artistic control, our solution is lightweight and can be executed on the fly, which keeps its memory consumption low and makes it easy to integrate into existing frameworks.

CCS CONCEPTS

• **Computing methodologies → Reflectance modeling**.

KEYWORDS

spectral rendering, reflectance spectra, spectral uplifting, convex hull, linear system, metamer mismatch volume

1 INTRODUCTION

Physically-based rendering systems are able to generate increasingly photorealistic imagery. While early approaches encoded physical properties such as light and reflectance as RGB colors, spectral rendering has become an important component for realism. A trichromatic approximation, as RGB, cannot accurately predict realistic color phenomena under complex light sources [Borges 1991]. Spectral rendering, albeit at increased computational costs, addresses this shortcoming by performing light-transport for various wavelengths. It enables accurate color reproduction and support for phenomena such as chromatic dispersion and fluorescence.

For spectral rendering, scene assets need spectral material properties, e.g., reflectance. This is a challenge, as spectral material capture is laborious, difficult, and the related memory consumption can be significant. Further, content-authoring pipelines tend to target RGB color spaces and no solution allows an artist to easily interact with spectral definitions. Instead, a large body of work exists that seeks to sidestep these issues by uplifting RGB colors to full spectra. This is an inherently ill-posed problem, as a color can stem from an infinite number of metameric spectra. Typically, solutions opt for smooth and bounded shapes seen in many reflectances, often establishing a 1-to-1 mapping between RGB and resulting spectra. Such a conversion is restrictive, as it does not allow users to achieve metameric behavior. Further, color constancy is an issue; an uplifted material produces well-specified colors only under a single illuminant. In consequence, an artist may have to tweak materials to produce expected results under different illuminants.

We propose to make spectral uplifting controllable, such that a user can define simultaneous material appearances under different illuminants, hereby supporting metameric behavior. For a single input texture, we derive a simple polytope, forming a convex hull around the original RGB texture data in an $\mathcal{R}^3$ space. For each vertex of this convex hull, a smooth and bounded reflectance is generated, respecting user-provided color constraints that define how the reflectance stored in each vertex should appear under different illuminants. We then use generalized barycentric coordinates to transfer the spectra from the vertices of the convex hull to the enclosed RGB texture data. By employing prior work on metamer mismatch volume estimation [Mackiewicz et al. 2019], we can further ensure that user-provided color constraints are restricted to possible solutions, ensuring a minimal roundtrip error.

As the uplifting builds upon an interpolation scheme, our solution provides a simple, compressed format for spectral textures, leading to a practical representation for spectral-rendering contexts. For this reason, we have made our toolkit, which we call *Metameric*, as well as a *Mitsuba 3* [Jakob et al. 2022] plugin, openly available[1]. Specifically, we make the following contributions:

- An efficient uplifting technique supporting color-constancy;
- A solution for constrained artistic control;
- A compact representation for spectral textures.

---

[1] https://graphics.tudelft.nl/Publications-new/2023/VE23/

In the following, we cover the relevant background (Sec. 2), before presenting our method (Sec. 3) and results (Sec. 4). Finally, we discuss our findings (Sec. 5) and conclude (Sec. 6).

2 BACKGROUND

Color theory. Color as a phenomenon is determined by the signals of a set of observing sensors, which respond differently to the spectral distribution of incident light [Wyszecki and Stiles 1982]. Such sensors are described by observer functions. Given an illuminant's spectral power distribution (SPD) $e(\lambda)$ (equal energy E or $D65$ daylight illuminant are often-chosen whitepoints), we can express the response to a spectral surface reflectance $r(\lambda)$ as

$$\Phi(r) = \int_\Lambda \phi(\lambda)\, e(\lambda)\, r(\lambda)\, d\lambda, \tag{1}$$

where $\phi(\lambda)$ is a drop-in for the set of observer functions, and Λ the range of light (often the visible spectrum). This is formalized by the *International Commision on Illumination* (CIE) in the CIE XYZ color space with a trio of standard observer functions $\bar{x}$, $\bar{y}$ and $\bar{z}$. A variety of color spaces is defined as linear transformations of CIE XYZ, such as sRGB, which is prevalent in computer graphics.

The mapping Φ in Equation 1 describes a linear transformation from an essentially infinite-dimensional spectral space $\mathcal{X}$ to a (usually) three-dimensional color space. We call this combination of observer functions and illuminants a *color system*. The set of possible responses $\{r \in \mathcal{X} \mid \Phi(r) \neq 0\}$ in any given color system forms a convex region, referred to as its *object color solid* (OCS).

Metamerism. We briefly summarize Finlayson and Morovic [2005] and Logvinenko et al. [2013], who extensively discuss metamerism and related concepts. Given a known color system and signal, uplifting means inverting Equation 1 by finding a reflectance s.t.

$$\Phi^{-1}(\Phi(r)) = r. \tag{2}$$

This is an ill-posed problem due to the underdetermined nature of the linear system in Equation 1. Thus, there is a convex set of reflectances, being *metameric* with respect to this color system, i.e.,

$$\Phi^{-1}(\Phi(r)) = \{r' \in \mathcal{X} \mid \Phi(r) = \Phi(r')\}. \tag{3}$$

All spectra in the above *metamer set* are solutions to Equation 2. A secondary color system Ψ with differing observer or illuminant, applied to the reflectances in this metamer set, might have its signal responses differ. This is called respectively *observer-induced* and *illuminant-induced* mismatching. Formally, if the metamer set $\Phi^{-1}(\Phi(r))$ is mapped to Ψ, this results in a non-singleton OCS called a *metamer mismatch region*. This region represents the full range of colors that may be observed after a color-system change.

Much work focuses on estimating OCS boundaries. Given its novelty, we highlight the work of Logvinenko et al. [2013]. It uses a linear mapping $\Gamma : \mathcal{X} \to R^6$ s.t. $\Gamma(r) = (z, z')$, where $z = \Phi(r)$ and $z' = \Psi(r)$ form the corresponding color signals over the set of Equation 3. The authors show that, for a given z, the set of signals in a metamer mismatch volume is a cross-section of Γ:

$$\mathcal{M}(z, \Phi, \Psi) = \{z' \in R^3 \mid (z, z') \in \Gamma\}. \tag{4}$$

While points inside the mismatch volume are formed by different metamers, boundary points reduce to a single *optimal* spectrum.
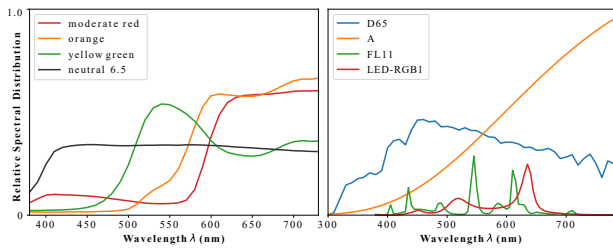
Figure 2: Reflectances (left) from a *BabelColor Average* dataset [Barber et al. 1996], and CIE standard illuminants (right). Note how reflectances are generally low-banded, while illuminants express a variety of shapes.

A notable property of optimal spectra is that they are elementary step-functions, consisting only of transitions between zeroes and ones. Knowledge on optimal spectra has expanded over the years [Logvinenko 2009; Schrödinger 1920; West and Brill 1983]. In our work, we apply the method of Mackiewicz et al. [2019] to find optimal spectra on mismatch volume boundaries, leading to a conservative approximate convex hull around $\mathcal{M}$. This approach establishes maximal theoretical boundaries to metameric mismatching. Empirically-established boundaries seem substantially smaller [Zhang et al. 2016], when ignoring structural colors.

Spectral distributions. We briefly expand on properties of natural spectra. Illuminants typically describe a quantity of energy per wavelength, while reflectances describe a surface's effectiveness in reflecting said energy. Illuminants are positively unbounded and express a variety of shapes dependent on the involved physical processes. In contrast, reflectances are bounded to $(0, 1)$, being energy-conserving. Fluorescents break this constraint, but should not be encoded as one-parameter functions. Further, reflectances are typically band-limited in the visible spectrum. This is considered a property of natural pigments, but not structural colors [Maloney 1986]. In this work, and most related work, we restrict ourselves to handling smooth reflectances. We depict several spectra in Fig. 2.

Spectral uplifting. While we briefly cover prior methods, we refer the reader to [Weidlich et al. 2021] for a broader categorization.

An early method for spectral uplifting, which nowadays has mostly theoretical implications, was proposed by MacAdam [1935]. The more recent and long-time standard method of Smits [1999] describes uplifting as an optimization problem, generating combinations of seven precomputed primary spectra. While this approach is fast, the resulting reflectances can break boundedness constraints. More recently, Meng et al. [2015] precompute sets of spectra on a grid spanning the xy chromaticity plane, recovering intermediates through interpolation. Their method produces smooth reflectances, but introduces round-trip errors for saturated colors. Otsu et al. [2018] improve on this approach by clustering measured spectra into a KD-tree over the xy plane. Inside each cluster, the authors interpolate weights applied to PCA-derived basis functions. While the resulting reflectances are inherently smooth, discontinuities arise at cluster boundaries, as different bases are used. Finally, Mallet and Yuksel [2019] describe convex combinations of three bases on sRGB gamut vertices, enabling reconstruction of input data
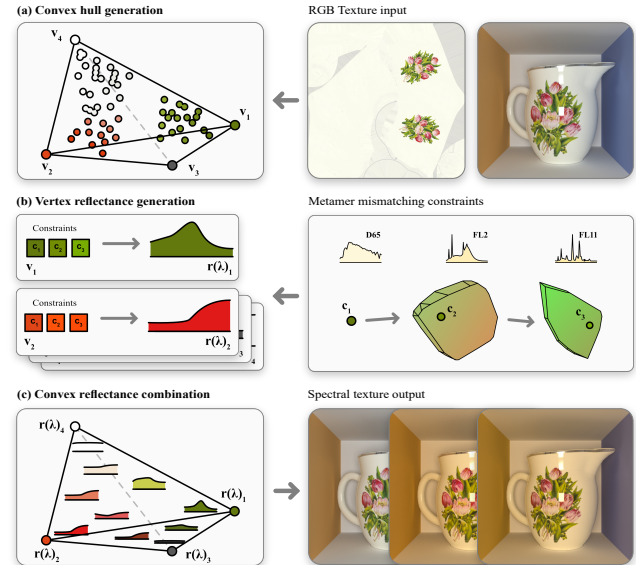

Figure 3: Method overview. As a preprocess, we generate a low-complexity convex hull around RGB inputs. For hull vertices, we generate reflectances satisfying artist-configurable color constraints. During rendering, we recover texture reflectances as convex combinations of vertex reflectances.

with little roundtrip error. However, due to the saturated shapes of their basis, uplifted spectra become arguably blocky. Further, their approach is restricted to uplifting of in-gamut sRGB data. Our method extends this concept, while avoiding such problems.

In their work, Jakob and Hanika [2019] use a low-dimensional parameterization of a sigmoidal function space. They precompute function coefficients inside a three-dimensional color lookup table. Function reconstruction from interpolated coefficients is inherently fast, and produces smooth reflectances with low round-trip error. Several extensions cover out-of-gamut spectra such as fluorescents [Jung et al. 2019; König et al. 2020]. A similar, Fourier-space based approach is demonstrated by Peters et al. [2019], addressing certain round-trip issues of the sigmoidal. Tódóva et al. [2021; 2022] expand on this method, to our knowledge being the first to introduce constrained spectral uplifting. They enable seeding of the method's coefficient generation, such that certain acquired spectra are reproduced accurately. Coefficient generation is costly, however, and their complex encoding of seeded constraints introduces overhead during rendering. Our approaches differ fundamentally. We generate reflectances from user-provided color constraints, implying we recover color-matched spectra, not reproductions. While enforcing color-constancy, this enables authoring of metamers for which spectral acquisition is difficult. Combined with our toolkit's interactive performance, this allows artistic expression in uplifting.

3 METHODOLOGY

Here, we present our constrained spectral uplifting, illustrated in Fig. 3. The core of our solution comprises three elements. First, we generate a low-complexity convex hull in $\mathcal{R}^3$ around the texel

colors of an RGB texture, and recover generalized barycentric coordinates to represent these texel colors as convex combinations of the hull's vertices. Second, we generate a small number of metamers with minimal round-trip error for the vertices of this convex hull, fulfilling artist-provided constraints that indicate the appearance of the vertex reflectances in different color systems. Third, during rendering, we spectrally uplift the input texture as a convex combination of the vertex metamers.

We first cover our method's foundation (Sec. 3.1), followed by convex-hull construction (Sec. 3.2). We then describe solving for vertex reflectances and texture uplifting (Sec. 3.3). Afterwards, we analyze user-guided uplifting and explain how to steer it via a set of RGB textures with known color systems (Sec. 3.4). We reserve implementation details for Sec. 4.

3.1 Foundation

As a color system describes a linear transformation (Equation 1), for any given color signal within that color system, there exists a set of metameric reflectances. Let n color signals $\Phi(r_1), \ldots, \Phi(r_n)$ with corresponding reflectances $r_1, \ldots, r_n$. We can then define a convex combination of these color signals using scalar weights $a_1, \ldots, a_n$, satisfying $\forall_i \ a_i \geq 0$ and $\sum a_i = 1$. Due to linearity:

$$\sum \Phi(r_i)a_i = \Phi\left(\sum r_i a_i\right). \tag{5}$$

In consequence, we see a direct relationship between linearly combined reflectances and linearly combined color signals. For example, the mean of two metamers is itself a metamer. This principle holds for arbitrary reflectances (Fig. 4), and is used in most prior work.

Meng et al [2015] apply this observation to interpolate spectra mapped to the xy chromaticity plane. Yet, a colorspace is trichromatic ($\mathcal{R}^3$). Any interpolation between three points leads to a triangle, while points outside the triangle's plane cannot be interpolated. Therefore, mapping multiple color signals to a plane is inherently problematic. Instead, we generate a convex hull with n vertices around the input colors in an $\mathcal{R}^3$ space such as CIE XYZ. This hull is minimally a 3-simplex (tetrahedron), though added vertices provide benefits such as a finer-grained control over uplifting. As all input colors of the texture to be uplifted lie within the hull, we can describe these colors as convex combinations of the hull's vertices, with scalar weights $a_1, \ldots a_n$ ($\forall_i a_i \geq 0, \sum a_i = 1$). As each vertex position is a color signal, we can find a suitable reflectance for each vertex from the signal's metamer set. Following Equation 5, we then recover valid reflectances for each input color, by computing the convex combination for these vertex reflectances.

3.2 Convex hull

In theory, the advantage of interpolating reflectances from the convex hull is that if vertex reflectances are bounded to $[0, 1]$, so will a resulting convex combination. Yet, the convex hull should not jut out of the color space, as, otherwise, natural reflectances cannot be reliably found for these vertices; they would not map to any color signal in the color system. This situation is common when only relying on a tetrahedron. For example, consider an RGB texture covering most of the gamut, and its $\mathcal{R}^3$ representation (Fig. 5). An enclosing tetrahedron's vertices would lie outside the color-system boundaries. We avoid this issue using a general polygonal hull.
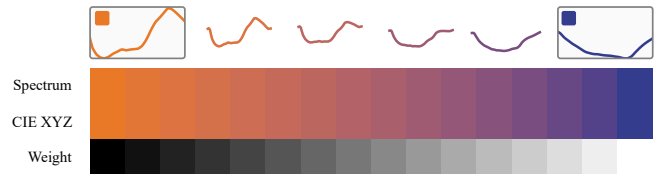

Figure 4: Following Equation 5, we show colors obtained from linearly mixing two reflectances, and corresponding results obtained from mixing their respective colors instead.
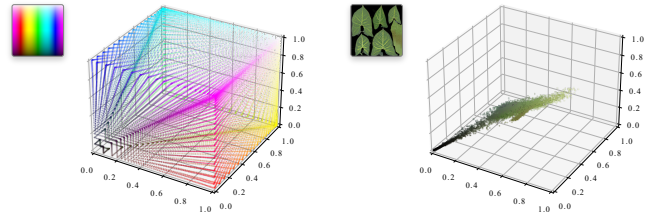

Figure 5: $\mathcal{R}^3$ mappings of texture data in linearized sRGB.

For a general polygonal hull, we can determine suitable weights to enable convex combinations. We rely on Mean Value Coordinates (MVCs) [Floater 2003], using the method of Ju et al. [2005] to adapt MVCs for triangle meshes. MVCs are in $[0, 1]$, sum to one, and are continuous within a hull's interior. The latter property ensures that recovered reflectances do not impress sudden discontinuities between upliftings, which is a problem Otsu et al. [2018] encounter.

Hull generation. Tan et al. [2018; 2016] show in their image segmentation work, that four to ten vertex structures encompass most input images, implying that a coarse hull typically suffices. We similarly generate an enclosing mesh around input colors using the *Quickhull* algorithm [Barber et al. 1996] and subsequently perform a progressive mesh simplification [Hoppe 1996] using repeated edge contractions. To generate a simplified convex hull, we should place a vertex resulting from a contraction strictly outside the existing hull. Sander et al. [2000] formulate such an optimization by ensuring that each new vertex strictly adds volume to a mesh. They then perform the contractions in an order that greedily minimizes this added volume. This approach might move contracted vertices outside the color system's gamut, in which case Tan et al. [2018; 2016] encounter reconstruction errors. To ensure valid vertex placement, we establish boundary constraints as follows.

First, we determine the region of possible color signals that a color system can produce, which is a convex hull; the OCS (Sec. 2). For example, a green illuminant will not allow any reflectance to produce a red color. We find this OCS hull using the recent sampling-based approach of Mackiewicz et al. [2019]. We then restrict vertex contractions to the hull's interior as otherwise, at a later stage, we cannot find reflectances for vertices outside the color system. Whenever a potential contraction falls outside the hull, the vertex is projected to the hull's surface. If this reprojection results in a contraction with negative volume, it is discarded and another is selected. Contractions are then repeated until the intended number of vertices is reached, or no further contractions can be performed
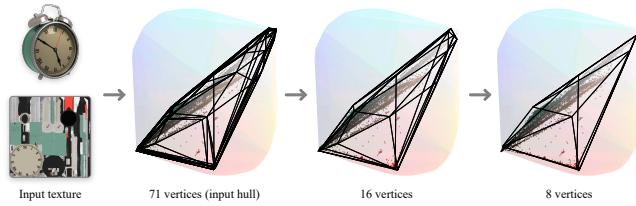
Figure 6: Progressive mesh simplification [Barber et al. 1996] of a convex hull, following the work of Tan et al. [2018; 2016], fitted around texture data (left). This results in simpler structures, but through volume-preserving constraints remains convex. Contracted vertices are bounded to a *object color solid*, the convex region of possible responses in a color system.

without breaking the above constraints. We show a convex hull generated by repeated progressive mesh simplification in Fig. 6.

3.3 Reflectance generation

Given the convex hull, we next generate reflectances for its vertices.

As in prior work, instead of working with spectra directly, we use weighted sums of basis functions obtained through PCA, which is an established representation for low-banded spectra based on measured data [Cohen 1964; Fairman and Brill 2004; Otsu et al. 2018; Parkkinen et al. 1989; Tzeng and Berns 2005]. We build our basis using a dataset of $\sim 41M$ reflectances [Zhang et al. 2016], which gathers earlier works [Foster et al. 2006; Le Moan et al. 2015; Li et al. 2014; Yasuma et al. 2010] representing a variety of natural and synthetic materials such as sediment, wood, plant-life, skin, food, paints, plastics, and textiles. Reflectances are measured over the $400 - 700$ nm spectral range at 10 nm intervals.

PCA reduces the dimensionality of a space. This leads to smooth reflectances, given a low number of basis functions; prior work has established that three functions are minimally required for most reflectances [Cohen 1964]. In short, we represent input reflectances as discretized, $k-$dimensional vectors r, resampling inputs if necessary. We then acquire m basis functions ($m \leq k$) using PCA, encoding the first m principal component eigenvectors in a $k \times m$ matrix B. Contrary to prior work, we retain $m > 3$ functions, ensuring that a single color system does not fully determine the resulting reflectance, leaving room for metameric mismatching.

We next formulate a discretization of Equation 1: $\Phi(r) = \Phi Bw$, where w is an unknown $m-$dimensional weight vector and $\Phi(r)$ is a known color signal. The corresponding color system Φ is encoded as a $3 \times k$ matrix of observer functions, multiplied with a measured illuminant. Solving for w then allows us to recover reflectances as $r = Bw$, which we store explicitly as a $k-$dimensional vector in the corresponding hull vertex. We solve via a linear-programming optimization [Dantzig 1982]. Hereby, we can enforce the solution's boundedness using additional constraints, i.e., $0 \leq Bw \leq 1$.

With a single color-system constraint (e.g, colors under *D65*), we can always find a reflectance (Equation 3). However, the system is underconstrained, thus, we can attempt to satisfy c secondary constraints $\Psi_1(r), \ldots, \Psi_c(r)$ under different color systems $\Psi_1, \ldots, \Psi_c$ simultaneously. In other words, for each vertex, we can specify its color signals in different color systems (e.g., colors under *D65* and

FL11). The metamer set, with respect to all constraints, is then:

$$\Phi^{-1}(\Phi(r)) = \{r' \in \mathcal{X} \mid \Phi(r) = \Phi(r') \land \forall_i \Psi_i(r) = \Psi_i(r')\}. \quad (6)$$

Involving all constraints, a solution w implies the existence of a reflectance $r = Bw$, which lies in the intersection of all color signals' respective metamer sets. The opposite also holds; no shared metamer exists if the sets' intersection is empty.

Determining high-dimensional metamer set boundaries is expensive, unless restrictions are imposed [Finlayson and Morovic 2005]. However, we can establish boundaries on mismatching between color signals in different color systems. To determine the region of valid weights w, we hence restrict the space of solutions by establishing input constraints in $\mathcal{R}^3$, one constraint at a a time. In each step, we produce a metamer mismatch volume (Equation 4), s.t. prior constraints are satisfied. We employ the method of Mackiewicz et al. [2019] to find this volume, clipping the constraint space as before. The order of constraints does not matter, as the resulting set of solutions lies within the insersection of all metamer sets.

Optimization then leads to a valid weight w among all possible solutions in the remaining set. We optimize for the smallest norm. If a metamer mismatch volume collapses to a point, a single weight forms the optimal solution. Upon finding w, we convert back to a spectrum ($r = Bw$) and store the result in the corresponding hull vertex. We store this spectrum and not the weight, as the number of vertices is low, and we avoid this multiplication during rendering. This further allows using an acquired reflectance, should this be available for the vertex.

Texture reconstruction. Once vertex reflectances are determined, uplifting of the input texture's color signals is straightforward by invoking the corresponding convex weights and Equation 5.

Indeed, uplifting a pixel p boils down to a matrix multiplication between MVC weights and vertex reflectances - for a single wavelength, only an inner product is needed. Specifically, given the MVC vector of pixel p; $A(p) := (\alpha_1, \cdots, \alpha_n)$, and vertex reflectances $R_1, \cdots R_n$, the uplifted spectrum is equal to $A(p)(R_1, \cdots, R_n)$.

3.4 User interaction

To enable constrained control, our toolkit provides an interface (Fig. 7). To begin uplifting, a user provides an input texture and primary color system (e.g. CIE XYZ and *D65*). As our method can uplift without secondary constraints, we immediately show reflectances and spectral renders of the input texture for different color systems. The interface then focuses on the convex hull in $\mathcal{R}^3$. Users can modify the hull to suit their needs, or add constraints to vertices.

As the user selects a vertex, a weight map illustrates how texels are affected by changes to this vertex. Next, the user can add a color system (e.g., *FL11*), and the interface shows a mismatch volume for the vertex in this system. The volume's interior describes all color signals that can be produced under *FL11*, while maintaining appearance in *D65*. The user can freely modify the vertex color in this second system, and a reflectance is generated fulfilling both constraints (*D65* and *FL11*). As the volume is color-coded, a user can intuit how a secondary constraint affects appearance of affected texels under *FL11*. Once satisfied, the user can add further constraints on other vertices, or via additional color systems.

The interface is interactive and provides direct visual feedback. We show an example session in the supplementary video.

Image seeding. Direct vertex manipulation gives full control and allows precise definitions. An additional option is to define constraints indirectly by providing measured images, which can be acquired or authored. The user then provides a color system per input image. The first image and color system (e.g., *D65*) is used as a reference. As input constraints can fall outside of the metamer mismatching boundaries of convex hull vertices, we cannot guarantee that a solution exist s.t. the input is reproduced without roundtrip error. In consequence, we minimize this error while preserving the primary input image. We proceed as follows.

We sample a random subset of texels from the primary image, and obtain secondary color signals at these texels from the secondary images. For each sample, we test whether a solution exists by attempting to recover a suitable reflectance, as authored textures may violate constraint boundaries, and otherwise sample again. We now have a set of texels with valid color constraints, and connect these to the convex hull using the previously computed MVCs. We next solve for vertex reflectances whose convex combinations satisfy the constraints at each sampled texel. This can again be specified as a linear programming optimization, in which basis function weights are found from which we can derive vertex reflectances.

If no solution exists, we reduce or relax the set of texel constraints. Hereby, we increase the space of possible solutions, but incur roundtrip error in secondary color systems. As the convex hull is unaffected, the primary will always be perfectly reproduced.

4 RESULTS

First, we evaluate accuracy, discuss implementation details, and establish used parameters. We then evaluate our method's per-vertex reflectance recovery (Sec. 4.1), followed by full-texture upliftings (Sec. 4.2). Finally, we detail rendering performance (Sec. 4.3).

Implementation. We implement most components as a preprocess in our uplifting toolkit. We operate on CIE XYZ values internally, enabling device-independent operations. We rely on *Open-Mesh* [Bischoff et al. 2002] for mesh simplification, and the *COIN-OR CLP* solver [Forrest et al. 2022] for linear-programming. We offload most work to *OpenGL*; MVC computation and rendering of uplifted textures are examples, though we note that MVCs are sensitive to numerical precision due to imprecise trigonometric functions. Finally, our user interface leverages *Dear ImGui* [Cornut 2022].

We define a straightforward exportable texture format consisting of two data blocks; vertex reflectances and MVCs. Disk storage is compressed using *zlib* [Deutsch and Gailly 1996]. We further implement a texture plugin for *Mitsuba 3* [Jakob et al. 2022]. Uplifting is implemented as four inner products, vectorized over four wavelengths. MVCs are shared across vector units, while vertex data is sampled per unit. We show example renders in *Mitsuba*, using authored spectral materials (Fig. 12).

Setup. We set a $400 - 700$ nm spectral range for all methods, and describe discrete spectra using $k = 64$ bins to handle high-frequency illuminants, resampling PCA inputs where necessary. Our choice of spectral range is motivated by our basis; our toolkit supports bases using a wider range such as the customary $360 - 380$ nm. Across

tests, we use $n = 8$ convex hull vertices; note that our method uses higher numbers if a hull cannot be simplified further.

While prior work employs $m = 3$ basis functions, Mackievicz et al. [2019] show that metamer mismatching boundaries are reduced by any linear model. The choice of m thus pivots on two factors; low values generate low-banded reflectances, while high values enable metameric mismatching. We generate mismatch volumes and interior metamer sets for $m = 6, 8, \ldots, 16$ (Fig. 8) but use $m = 12$ for further tests. As shown, mismatch boundaries are conservative approximations dependent on m. However, we do not consider this a problem, as Zhang et al. [2016] show that empirically measured mismatch volumes are smaller than the theoretical maximum.

4.1 Single-reflectance recovery

We sample reflectances from a *BabelColor Average* dataset [Barber et al. 1996], and measure color signals for standard illuminants *D65*, *A*, *E*, *FL2*, *FL11*, and *LED-RGB1*. We uplift for *D65* using the methods of Smits [1999], Meng et al. [2015], Otsu et al. [2018], and Jakob and Hanika [2019]. We then apply our method without and with $c = 1, 2, 3$ secondary constraints, for *FL2*, *FL11* and *LED-RGB1*.

Fig. 9 shows roundtrip colors and CIE LAB ΔE_{00} color difference for each method ($\Delta E_{00} \leq 1$ implies no perceptible difference; $\Delta E_{00} \leq 2$ implies minimal mismatching). With the exception of Smits [1999], all methods correctly recover for *D65*. Further, all methods except Otsu et al. [2018] display metameric mismatching for *FL11*; our method without additional color constraints is no exception here ($\Delta E_{00} = 7.85$). With constraints, results improves significantly; we measure $\Delta E_{00} \leq 0.72$ ($c = 2$) and $\Delta E_{00} \leq 0.14$ ($c = 3$), far below the perceptible limits for all illuminants.

4.2 Multiple-reflectance recovery

We demonstrate manual texture authoring in Fig. 10. We establish a baseline for *D65* using the sigmoidal [Jakob and Hanika 2019]. Without further constraints, our method produces smooth metamers to the baseline. We then show two mismatched upliftings under *FL11*, illustrating the variety our method enables (note the ΔE_{00} measures). We further show that a user can generate color-constant upliftings for *FL11*, while mismatching for *D65*.

We next test recovery of hyperspectral data, sampling textures from the *HyTexila* dataset [Khan et al. 2018] in *Leaf*, *Textile*, *Stone* and *Wood* categories. Each 1024^2 texture stores 186 spectral channels ($400 - 1000$ nm). We acquire color textures under the illuminants used in Sec. 4.1, and perform renders for *D65* using the sigmoidal [Jakob and Hanika 2019]. We seed our method with color images under *FL2*, *FL11*, and *LED-RGB1*. While we expect to avoid metameric failure, roundtrip error is likely for secondary color systems, as constraints are relaxed to preserve the convex hull.

Fig. 11 shows roundtrip error and recovered reflectances. Both methods correctly handle *D65*, but show mismatching in some situations. Our method's constraint fitting varies in quality between images; *Leaf* / *Wood* textures are recovered without perceptible error, while *Textile* / *Stone* uplifts show mismatching under *FL11*, as our method fails to correctly fit a number of outlier texels (e.g. *Textile*; $\Delta E_{00} \leq 4.56, \mu = 0.95$). The sigmoidal, in comparison, reconstructs the smooth reflectances of *Wood* / *Stone* textures, but produces visible metameric failure for *Leaf* / *Textile*.

Table 1: Runtime and texture memory usage during rendering; our method with $n = 4, 8, 16$ vertices, Jakob and Hanika's method [2019], and a hyperspectral ground with 186 spectral bins. Run on Ryzen 9 5950X, 32GB DDR4, NVIDIA RTX 3070.

	Sigmoid	n = 4	n = 8	n = 16	Ground
Runtime (s)	31.35	32.83	34.18	35.10	36.40
Memory (MB)	16.0	16.0	32.0	64.0	792.0

4.3 Memory and runtime

We compare memory and runtime of the methods used in Sec. 4.2. We render a 1024^2 px image at 256 *spp.* (averaging ten renders), s.t. a screen-filling render of the *Wood* texture is generated. All methods use single-precision, though we speculate that half-precision suffices for parts of each method. For our method, we measure for $n = \{4, 8, 16\}$ vertices. All choices gave a valid convex hull.

Table 1 lists results for each method. Results are as expected; the sigmoidal shares texture reads across four wavelengths, and is practically as fast as trichromatic rendering. Our method requires five texture reads per four wavelengths (one for MVCs, four for wavelength reflectances), whose size depends on vertex count. This implies a slight overhead for increased vertex counts. However, both methods compare favorably to the hyperspectral, as excessive memory consumption makes this intractable for large scenes.

5 DISCUSSION

In the following, we reflect on prior results and expand on several aspects of our method.

Reflectance recovery. Our method's ability to recover the right metamers depends on the number of secondary constraints (Sec. 4.1). While recovery for *D65* is expectedly without roundtrip error, results for $c \geq 2$ and the shapes of recovered spectra are similar to those of Otsu et al. [2018]. As their approach does not require secondary constraints, this bears discussion. In short, their method restricts the PCA basis to a representable set through clustering and use of $m = 3$ basis functions. This leads to recovery of low-banded reflectances which reproduce the *BabelColor Average* dataset [Barber et al. 1996] particularly well, but remains a 1-to-1 mapping. If we explicity select metamers in Fig. 9, their method incurs significant mismatching, arbitrarily favoring our method. This is not a failure of either method, instead demonstrating the necessity of constrained uplifting. To produce similar results to Otsu et al. [2018] without secondary constraints, we could apply their clustering approach, using a different basis set across convex hull vertices.

During hyperspectral texture recovery (Sec. 4.2), our method recovers the *D65* input, but mismatches in a number of texels especially for *FL11*. Vertex constraints are fitted on a sampled subset of the input image. We relax this fitting s.t. the convex hull is preserved. As the hull specifies a particular convex combination for each sample, this implies that constraints can at times not perfectly fit on all samples. The reconstruction is hence imperfect, though error is imperceptible for most texels. The sigmoidal [Jakob and Hanika 2019] provides good reconstruction of *Wood* and *Stone* textures, as the

function's shape matches the reflectances in these textures. However, this method shows full-image mismatching for vegetation, where recovered reflectances do not resemble the ground truth.

Basis function restrictions. We employ $m = 12$ basis functions in tests, which exceeds the minimum [Cohen 1964], but enables metameric control. This flexibility implies that a low number of constraints admits many solutions, making results nondeterministic. To avoid this behavior, one could vary m depending on the context. We leave this as future work, but if we can establish in which volume (Fig. 8) a constraint lies (preferably without solving for volumes), this enables reflectance recovery using the minimum required basis, while allowing artists to leverage more bases when necessary.

We further note that, as our basis uses common material data, reproduction is not guaranteed for all materials. If necessary, our toolkit supports loading bases that target specific material classes.

Convex hull restrictions. Our method has a notable limitation; image segmentation through a convex hull restricts the set of available upliftings. Consider an input image with two identical texels. No constraint can separate these texels into separate metamers, as they share weights and thus upliftings. If the input were separated into partial sets of independently uplifted texels, this can be handled. Alternatively, given an acquired image, one can start with a color system where texels are different. Then it is possible to produce metameric matching (e.g., the mug in Fig. 12). This indicates that the choice of input color system is influential, and can be restrictive.

6 CONCLUSION

Our novel method for spectral uplifting of RGB textures is controllable and allows a user to define material appearance under different illuminants simultaneously. It generates reflectances from a small number of metameric mismatching constraints and uplifts through a simple interpolation. The latter results in a compressed representation for spectral textures with minimal roundtrip error.

Previous uplifting without control cannot target metameric behavior or color constancy. Integrating our work into content pipelines can help with such issues, and we have published our toolkit to support widespread use.

REFERENCES

C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)* 22, 4 (1996), 469–483.

Botsch Steinberg Bischoff, M Botsch, S Steinberg, S Bischoff, L Kobbelt, and Rwth Aachen. 2002. OpenMesh–a generic and efficient polygon mesh data structure. In *In OpenSG Symposium*.

Carlos F. Borges. 1991. Trichromatic Approximation for Computer Graphics Illumination Models. *SIGGRAPH Comput. Graph.* 25, 4 (jul 1991), 101–104. https://doi.org/10.1145/127719.122729

Jozef Cohen. 1964. Dependency of the spectral reflectance curves of the Munsell color chips. *Psychonomic science* 1, 1 (1964), 369–370.

Omar Cornut. 2022. *Dear ImGui.* https://github.com/ocornut/imgui

George B. Dantzig. 1982. Reminiscences about the origins of linear programming. *Operations Research Letters* 1, 2 (1982), 43–48. https://doi.org/10.1016/0167-6377(82)90043-8

Peter Deutsch and Jean-Loup Gailly. 1996. *Zlib compressed data format specification version 3.3.* Technical Report.

Hugh S Fairman and Michael H Brill. 2004. The principal components of reflectances. *Color Research & Application* 29, 2 (2004), 104–110.

Graham D. Finlayson and Peter Morovic. 2005. Metamer sets. *J. Opt. Soc. Am. A* 22, 5 (May 2005), 810–819. https://doi.org/10.1364/JOSAA.22.000810

Michael S Floater. 2003. Mean value coordinates. *Computer aided geometric design* 20, 1 (2003), 19–27.

John Forrest, Stefan Vigerske, Ted Ralphs, Lou Hafer, John Forrest, jpfasano, Haroldo Gambini Santos, Matthew Saltzman, Jan-Willem, Bjarni Kristjansson, h-i gassmann, Alan King, pobonomo, Samuel Brito, and to st. 2022. *coin-or/Clp: Release releases/1.17.7*. https://doi.org/10.5281/zenodo.5839302

David H Foster, Kinjiro Amano, Sérgio MC Nascimento, and Michael J Foster. 2006. Frequency of metamerism in natural scenes. *Josa a* 23, 10 (2006), 2359–2372.

Hugues Hoppe. 1996. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 99–108.

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

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. https://mitsuba-renderer.org.

Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean value coordinates for closed triangular meshes. In *ACM Siggraph 2005 Papers*. 561–566.

A. Jung, A. Wilkie, J. Hanika, W. Jakob, and C. Dachsbacher. 2019. Wide Gamut Spectral Upsampling with Fluorescence. *Computer Graphics Forum* 38, 4 (2019), 87–96. https://doi.org/10.1111/cgf.13773

Haris Ahmad Khan, Sofiane Mihoubi, Benjamin Mathon, Jean-Baptiste Thomas, and Jon Yngve Hardeberg. 2018. HyTexiLa: High Resolution Visible and Near Infrared Hyperspectral Texture Images. *Sensors* 18, 7 (2018). https://doi.org/10.3390/s18072045

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

Steven Le Moan, Sony T George, Marius Pedersen, Jana Blahová, and Jon Yngve Hardeberg. 2015. A database for spectral image quality. In *Image Quality and System Performance XII*, Vol. 9396. SPIE, 225–232.

Changjun Li, M Ronnier Luo, MR Pointer, and Phil Green. 2014. Comparison of real colour gamuts using a new reflectance database. *Color Research & Application* 39, 5 (2014), 442–451.

Alexander D. Logvinenko. 2009. An object-color space. *Journal of Vision* 9, 11 (10 2009), 5–5. https://doi.org/10.1167/9.11.5

Alexander D Logvinenko, Brian Funt, and Christoph Godau. 2013. Metamer mismatching. *IEEE Transactions on Image Processing* 23, 1 (2013), 34–43. https://doi.org/10.1109/TIP.2013.2283148

David L. MacAdam. 1935. Maximum Visual Efficiency of Colored Materials. *J. Opt. Soc. Am.* 25, 11 (Nov 1935), 361–367. https://doi.org/10.1364/JOSA.25.000361

Michal Mackiewicz, Hans Jakob Rivertz, and Graham Finlayson. 2019. Spherical sampling methods for the calculation of metamer mismatch volumes. *J. Opt. Soc. Am. A* 36, 1 (Jan 2019), 96–104. https://doi.org/10.1364/JOSAA.36.000096

Ian Mallett and Cem Yuksel. 2019. Spectral Primary Decomposition for Rendering with sRGB Reflectance. The Eurographics Association. https://doi.org/10.2312/sr.20191216

Laurence T. Maloney. 1986. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A* 3, 10 (Oct 1986), 1673–1683.

https://doi.org/10.1364/JOSAA.3.001673

Johannes Meng, Florian Simon, Johannes Hanika, and Carsten Dachsbacher. 2015. Physically Meaningful Rendering using Tristimulus Colours. *Computer Graphics Forum* 34, 4 (2015), 31–40. https://doi.org/10.1111/cgf.12676

H. Otsu, M. Yamamoto, and T. Hachisuka. 2018. Reproducing Spectral Reflectances From Tristimulus Colours. *Computer Graphics Forum* 37, 6 (2018), 370–381. https://doi.org/10.1111/cgf.13332

Jussi PS Parkkinen, Jarmo Hallikainen, and Timo Jaaskelainen. 1989. Characteristic spectra of Munsell colors. *JOSA A* 6, 2 (1989), 318–322.

Christoph Peters, Sebastian Merzbach, Johannes Hanika, and Carsten Dachsbacher. 2019. Using Moments to Represent Bounded Signals for Spectral Rendering. *ACM Trans. Graph.* 38, 4, Article 136 (jul 2019), 14 pages. https://doi.org/10.1145/3306346.3322964

Pedro V Sander, Xianfeng Gu, Steven J Gortler, Hugues Hoppe, and John Snyder. 2000. Silhouette clipping. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 327–334.

Erwin Schrödinger. 1920. Theorie der Pigmente von grösster Leuchtkraft. *Annal. Der Phys.* 62 (1920), 603–622.

Brian Smits. 1999. An RGB-to-Spectrum Conversion for Reflectances. *Journal of Graphics Tools* 4, 4 (1999), 11–22. https://doi.org/10.1080/10867651.1999.10487511

Jianchao Tan, Jose Echevarria, and Yotam Gingold. 2018. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry. *ACM Transactions on Graphics (TOG)* 37, 6, Article 262 (Dec. 2018), 10 pages. https://doi.org/10.1145/3272127.3275054

Jianchao Tan, Jyh-Ming Lien, and Yotam Gingold. 2016. Decomposing Images into Layers via RGB-Space Geometry. *ACM Trans. Graph.* 36, 1, Article 7 (nov 2016), 14 pages. https://doi.org/10.1145/2988229

Di-Yuan Tzeng and Roy S Berns. 2005. A review of principal component analysis and its applications to color technology. *Color Research & Applications* 30, 2 (2005), 84–98.

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

L. Tódová, A. Wilkie, and L. Fascione. 2022. Wide Gamut Moment-based Constrained Spectral Uplifting. *Computer Graphics Forum* 41, 6 (2022), 258–272. https://doi.org/10.1111/cgf.14617

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

Gerhard West and Michael H. Brill. 1983. Conditions under which Schrödinger object colors are optimal. *J. Opt. Soc. Am.* 73, 9 (Sep 1983), 1223–1225. https://doi.org/10.1364/JOSA.73.001223

G. Wyszecki and W. S. Stiles. 1982. *Color science: Concepts and Methods, Quantative Data and Formulae*. Wiley.

Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. 2010. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing* 19, 9 (2010), 2241–2253.

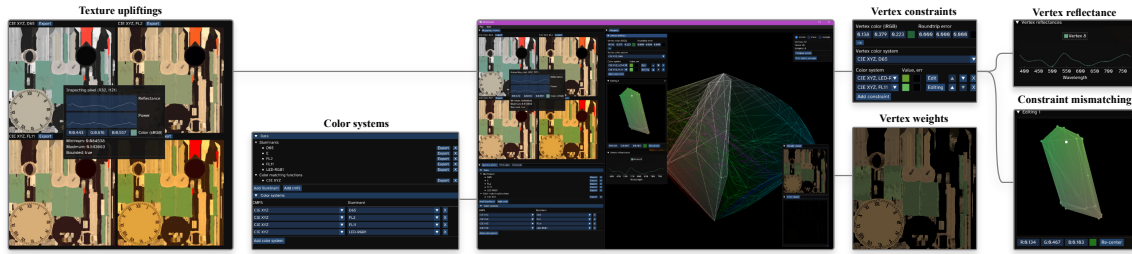Xiandou Zhang, Brian Funt, and Hamidreza Mirzaei. 2016. Metamer mismatching in practice versus theory. *J. Opt. Soc. Am. A* 33, 3 (Mar 2016), A238–A247. https://doi.org/10.1364/JOSAA.33.00A238

Figure 7: Our toolkit's user interface. A user can specify color systems (left) and see spectrally uplifted renders (far-left). The user can modify the convex hull (center), or select its vertices. Per vertex, the user can add and edit color-system constraints (right) within a metamer mismatch volume (far right). A vertex' affected texels and constrained reflectances are visualized.
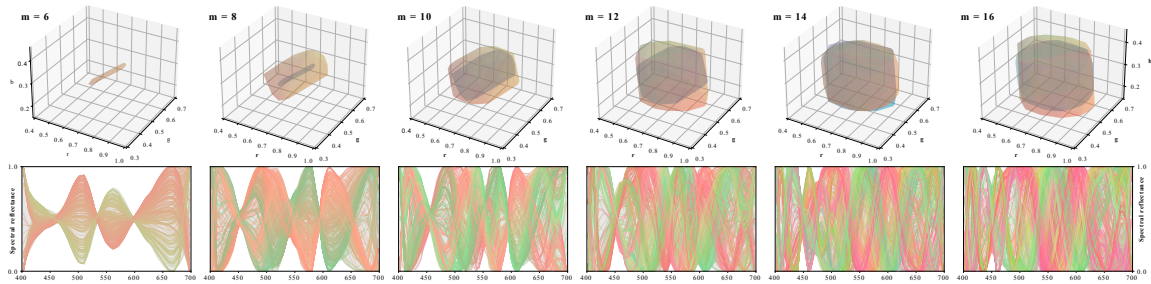

Figure 8: We generate metamer mismatch volumes (top) for illuminant-induced mismatching ($D65 \rightarrow FL11$) of neutral gray, using m basis funtions. We show sampled metamer reflectances for each volume (bottom) with $m > 6$. With less flexibility ($m < 6$) the volumes collapsed for this case.
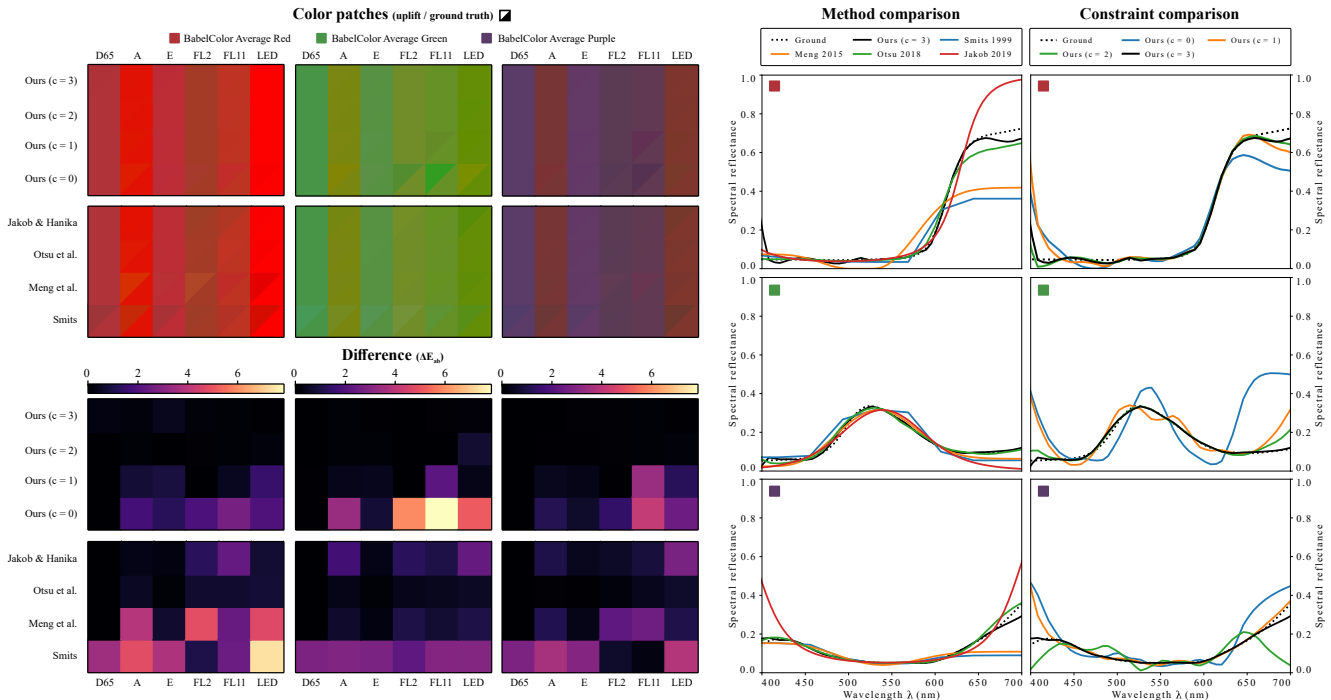

Figure 9: Single-reflectance recovery for reflectances from a *BabelColor Average* dataset [Barber et al. 1996]. We display roundtrip color and ground truth for uplifting methods (top-left). We constrain our method for *FL2*, *FL11*, and *LED-RGB1*, in addition to *D65*. We further show CIE LAB ΔE_{00} of each roundtrip (bottom-left). Resulting reflectances for each method are shown (middle-right), as well as reflectances resulting from different constraint sets in our method (right).

Figure 10: Artistic uplifting. Using sigmoidals [Jakob and Hanika 2019] (left) produces one reflectance. We can create metamers for *D65*, which mismatch for *FL11* (center) (we show direct fit and two possible choices). The primary can also be *FL11* (right). False colored textures illustrate difference to Sigmoidal, not error.
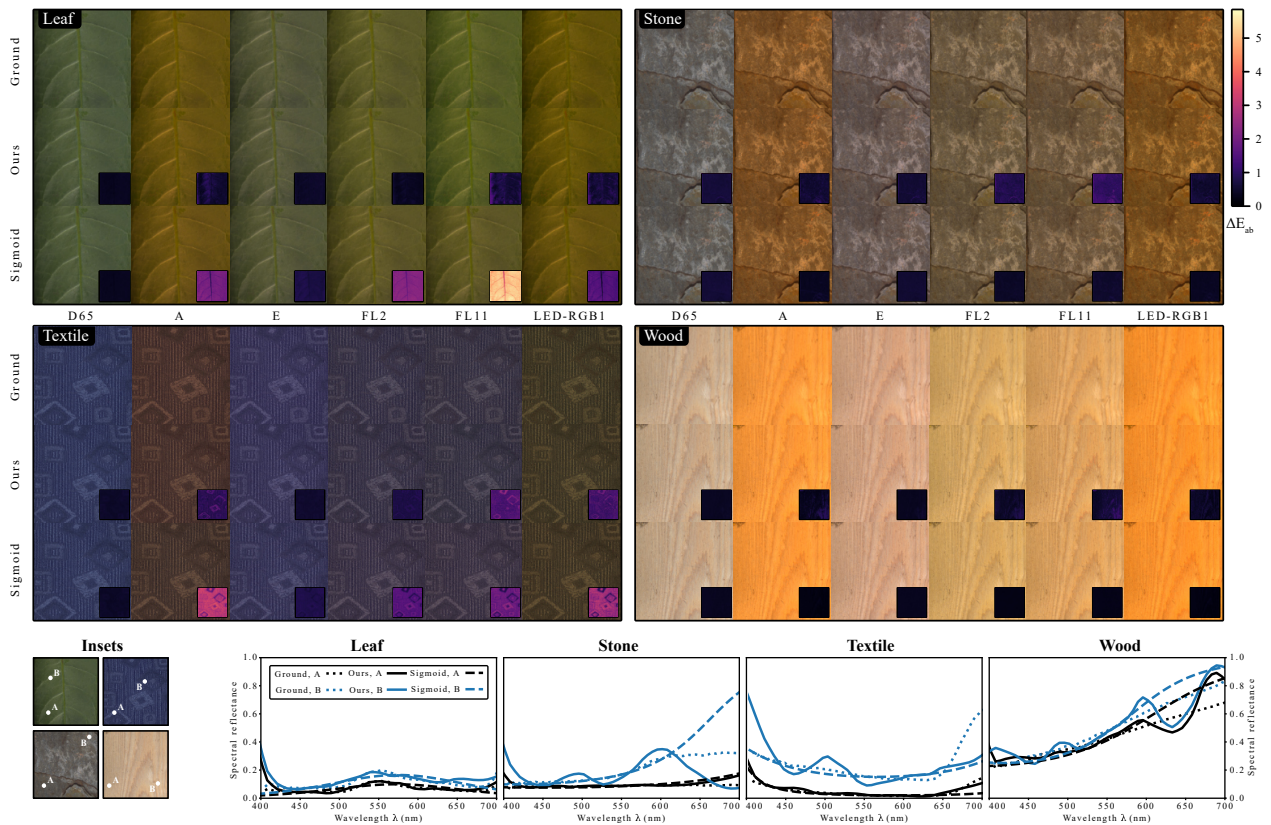

Figure 11: Recovery of hyperspectral images from the HyTexila dataset [Khan et al. 2018]. We display roundtrip and ΔE_{00} error for our method and that of Jakob and Hanika [2019] (top), and uplifted reflectances (bottom). We constrain our uplifting for *D65*, *FL2*, *FL11*, and *LED-RGB1*.

Figure 12: Example spectral renderings in Mitsuba 3 [Jakob et al. 2022], using spectral materials authored in our toolkit.