Integrating the Space of Reflectance Spectra

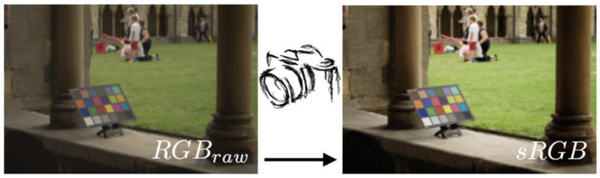
Graham D. Finlayson, *Member, IEEE*, Javier Vazquez-Corral, and Fufu Fang

***Abstract*—Color imaging algorithms - such as color correction, spectral estimation and color constancy - are developed and validated with spectral reflectance data. However, the choice of the reflectance data set - used in development and tuning - not only affects the results of these algorithms but it also changes the ranking of the different approaches. We propose that this *fragility* is because it is difficult to measure/sample enough data to statistically represent the large number of degrees of freedom apparent in spectral reflectances. In this paper, we propose that the space of reflectance data should not be sampled but, rather, integrated. Specifically, we advocate that the convex closure of a reflectance data set - all convex combinations of all spectra - should be used instead of discrete reflectance samples. To make the integration computation tractable, we approximate these convex closures by their enclosing hyper-cube in a privileged coordinate system. We use color correction as an exemplar color imaging problem to demonstrate the utility of our approach.**

***Index Terms*—Color imaging, integration, color correction, camera characterization.**

I. Introduction

UNSURPRISINGLY, many color algorithms ranging from color correction to color constancy to spectral estimation are developed from and tuned using measured reflectance data. In Figure 1, we show a typical color correction scenario. Here we take a picture of the Macbeth Color checker and then regress the captured-RGBs to corresponding display-RGBs here, using a linear transform, so that the captured image looks as close as possible to the correct colors (i.e. the colors we see ourselves when viewing the checker). In this example, the Macbeth color checker is used as training data for deriving the color correction linear transform. As well as making the colors in the checker to look correct, it is hoped that the derived transform also maps other unseen colors with low visual error.



$$\begin{bmatrix} R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 \\ \vdots & \vdots & \vdots \\ R_{24} & G_{24} & B_{24} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \approx \begin{bmatrix} r_1 & g_1 & b_1 \\ r_2 & g_2 & b_2 \\ \vdots & \vdots & \vdots \\ r_{24} & g_{24} & b_{24} \end{bmatrix}$$

Fig. 1. Left and right, top, shows respectively raw and color corrected image (both images with an sRGB gamma applied). The, ungammaed corrected linear RGBs, of the RAW are mapped to display counterparts by a 3 × 3 matrix. This matrix is chosen to best fit the data.

Clearly, the choice of reflectance ‘training data’ is an important consideration. Indubitably, if different reflectance training sets are used then different correction matrices would be calculated and, on unseen testing data, their performance would vary. A more ornery issue is that one algorithm might be ranked better than another - irrespective of how colors are corrected - simply because different training and testing data is used. In this paper, we are going to consider this *reflectance dependency* problem in detail. A visual explanation is presented in Figure 2.

First, we wonder whether the problem isn’t with the question of “which data set” but with the premise that that there exists an *ideal* dataset. While one might imagine that an ideal dataset exists for some target applications (e.g. correctly measuring skin color), it seems less obvious that one exists for color photography in general.

A central idea we test in this paper is that the number of reflectances in typical data sets is small given the intrinsic dimensionality of the data (reflectances are typically represented by 31 or more sampled measurements across the visible spectrum). To motivate our work, let us carry out a sampling ‘thought’ experiment which, while unrelated to color correction, highlights the problem of sampling when the data dimension is large. We ask “how many points do we need to generate -uniformly and randomly in a d-dimensional unit cube- to ‘cover’ the cube?” A priori we know that square, cube or d-dimensional hypercube have an area, volume or hyper-volume of 1 (by definition). So, for a given d-dimensional space, we randomly sample the unit hyper-cube, calculate the convex hull of the points and then we compute the corresponding volume. How many points do we need to randomly sample in order to approximate the volume?

Received 29 August 2024; revised 19 February 2025; accepted 28 March 2025. Date of publication 11 April 2025; date of current version 30 April 2025. The work of Graham D. Finlayson was supported in part by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S028730/1; and in part by York University, Toronto, Canada (where he was a VISTA Distinguished Visiting Scholar, funded by Canada First Research Excellence Fund and York University). The work of Javier Vazquez-Corral was supported in part by Spanish Ministry of Science and Innovation (MCIN/AEI)/10.13039/501100011033 and “ERDF/EU” under Grant PID2021-128178OB-I00, in part by the Departament de Recerca i Universitats from Generalitat de Catalunya under Grant 2021SGR01499, in part by the “Ayudas para la recualificación del sistema universitario español” financed by European Union-NextGenerationEU, and in part by the Generalitat de Catalunya CERCA Program to CVC’s General Activities. The associate editor coordinating the review of this article and approving it for publication was Prof. Jiaying Liu. *(Corresponding author: Javier Vazquez-Corral.)*

Graham D. Finlayson and Fufu Fang are with the School of Computing Sciences, University of East Anglia, NR4 7TJ Norwich, U.K. (e-mail: g.finlayson@uea.ac.uk; f.fang@uea.ac.uk).

Javier Vazquez-Corral is with the Computer Vision Center, Cerdanyola del Vallès, 08193 Bellaterra, Spain, and also with the Computer Sciences Department, Universitat Autònoma de Barcelona, Cerdanyola del Vallès, 08193 Bellaterra, Spain (e-mail: javier.vazquez@cvc.uab.cat).

Digital Object Identifier 10.1109/TIP.2025.3558443

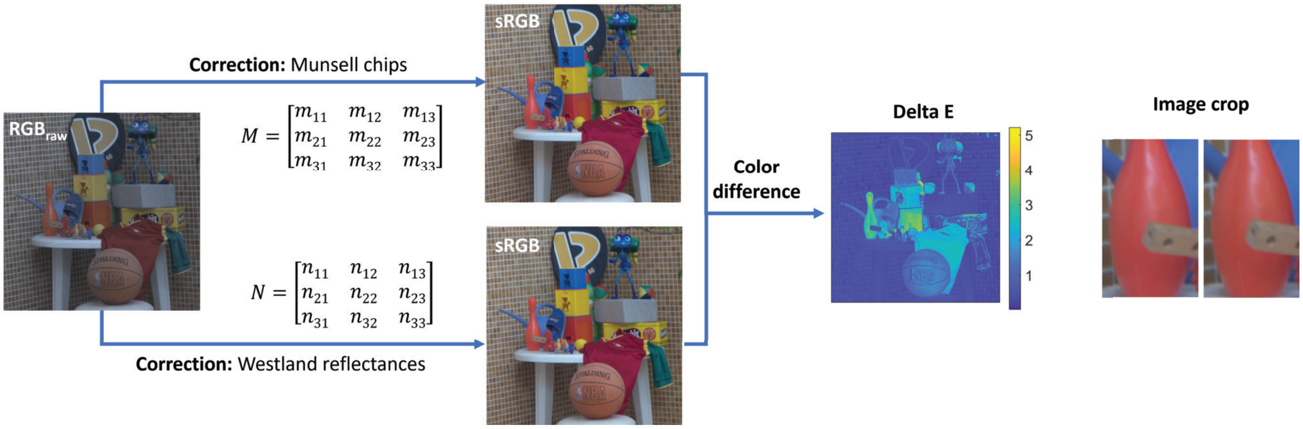

Correction: Munsell chips
RGB_raw
$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$
$N = \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix}$
Correction: Westland reflectances
sRGB
sRGB
Color difference
Delta E
5
4
3
2
1
Image crop

Fig. 2. Correcting with different training sets of reflectances leads to different color correction solutions. In this particular example, an original RAW image is corrected to sRGB using both the Munsell Chips reflectances [1] and the reflectances captured by Westland [2]. Then, the perceptual color difference between the two sRGBs images—that should theoretically be almost identical— is computed. However, we can see that these differences are large enough to be perceived by human observers (values of up to 5 Delta E, see the image crops). Even worst, the ranking of algorithms may therefore depend on the training and testing data that is used. In this paper, we propose a solution to this problem by proposing that reflectance sets should be represented by the set of all derived (e.g. from convex combinations) reflectances.

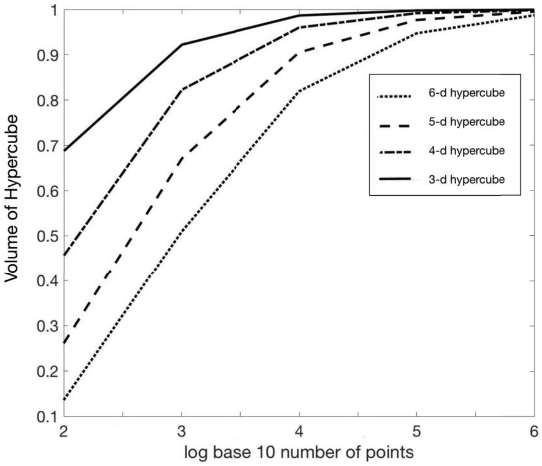


Fig. 3. Volume of the convex hull of uniformly and randomly selected points in dimensions 3, 4, 5 and 6. To cover 99% of the hypercube we need $O(10^d)$ points.

In Figure 3, we plot log_{10}(number of points) on the x axis against the volume of the convex hull of the points on the y-axis. As we can see in the Figure, our experiment indicates that to cover 99% of the actual volume we need on the order of $10^{\text{dimension}+1}$ points. That is, the number of sample points needed to cover a hypercube is exponential in the dimension of the data. This simple empirical observation turns out to be true in the limit when the dimension becomes large [3].

Let us return to the problem at hand: representing reflectance data. Spectral datasets, that are used in color imaging research, range from a few 10s of reflectance spectra measurements to a few thousands, e.g. the Macbeth Color checker (24 reflectances) [4], the Dupont reflectance set (120 reflectances), the object dataset (170 reflectances) [5] and the SFU composite reflectance set [6] (2000 reflectances including the Macbeth and Object dataset). Reflectances are often represented by their values measured in the interval 400 to 700 Nanometres (i.e. across the visible spectrum) with a 10 Nm sampling. As such reflectances are represented as 31-component vectors of measurements. It follows that each reflectance dataset - which comprise points in 31dimensional space, are modest in size. Given the high dimensionality the measurements (31)-vectors), can even 2000 reflectances adequately represent all reflectances we might encounter in the natural world?

If the intent is to represent all spectra that might occur in nature, a few thousand spectra will not be enough. However, in nature, reflectances often blend together when viewed from a distance. So even though a given reflectance set comprises a small number of measurements, we propose that the set can be usefully thought of as a sort of dictionary that might blend together to form new reflectances.

In detail, suppose we have a surface that is a patchwork of materials (e.g. a simple colored texture) and we view this texture from far enough away. The effective reflectance we see is a combination of the underlying reflectances. Or, to say the same thing mathematically, the combined reflectance is a convex combination of the individual reflectances (in proportion to the area that each reflectance covers). By making mosaics of colors by taking the reflectances in a given sampled data set we can -when viewing the mosaic from far enough away- make every reflectance that lies in the convex closure of the reflectance dataset [7].

In this paper, we propose that reflectance sets should be represented by the set of all derived (e.g. from convex combinations) reflectances. Moreover, as well as representing spectral data by its convex hull (more precisely, its convex closure), we also propose that the convex closure should be sampled uniformly and randomly. Of course, in doing so, we are changing the underlying distribution of the measured data. We argue that this is not a problem. Indeed, an argument can be made that we want to change the distribution. By uniformly and randomly sampling, we admit *all possible* physically plausible stimuli to be considered and every reflectance is equally as important as any other. In a sense, we are taking a sort of maximal ignorance stance (commonly used in color imaging research [8], [9], [10]) and being agnostic - save for the fact that it is statistically plausible - of the likelihood of saying that one measured spectrum is more or less likely than another.

Given the assumption that we can represent a reflectance set by its convex hull (and that all reflectances within this hull, its convex closure, are possible) then this begs the question of how do we make use of this representation. One might naively suppose that we could sample our data set but the dimensionality of the problem is against us. Rather, we will argue that we should integrate over the entire convex closure. And, for the problem of color correction, we show how this can be achieved.

Color correction algorithms attempt to map camera RGBs to a human vision system referenced color space, typically with a 3 × 3 matrix transform e.g. we map to XYZ tristimuly [11] or, as shown in Figure 1, to RGBs that drive a display (sRGB [12]). Figure 1 also teaches that the correction transform can be found by regression. There, we find the best linear map taking the 24 RGBs from the Macbeth color checker to the display sRGB counterparts.

The convex closure of a discrete set of surface measurements delimits an infinite set of reflectances. This seems to pose a problem for regression-based color correction. In Figure 1, it appears that one can only carry out a regression given a finite set of corresponding samples. To sidestep this problem, we will show that the least-squares regression depends on the spectral autocorrelation matrix which, assuming 31 sampling wavelengths, is a 31×31 matrix. We might still use regression if we can solve for the autocorrelation corresponding to an infitite set of reflectances. Unfortunately, for our convex closure of reflectances, calculating the autocorrelation is surprisingly laborious to compute (and, practically, is solved by random point generation which, again, brings us back to the question of 'how many samples?').

To make the computation feasible, we make two contributions. First, if a reflectance set is represented by its enclosing hypercube (which is itself a superset of dataset's convex closure) then we will show that it is easy to solve for a color correction matrix by integrating over a hypercube. However, it's important that the bounding hypercube fits our data well. Since, if the difference in volume between the convex closure of reflectance set and its bounding hyper cube is too large then we will have admitted many more reflectances just to make the integration problem easier to solve.

Our second contribution is to show how to characterize a reflectance set by computing the bounding hypercube with respect to a privileged basis. Importantly, in the privileged basis - related to a PCA decomposition of the data - the enclosing hypercube is a *tighter* fit (i.e. it is smaller and more tuned to the data) than if the hypercube were applied in the spectral domain.

Experiments will establish that integrating reflectance sets (not sampling) leads to good color correction - competitive with using the sampled reflectances themselves - when we train on a hyper-cube reflectance set and test on the individual sampled reflectances. This is an encouraging result since, as is always the case for least-squares, the best result in fitting data will be when the test and training sets are the same. That our method delivers almost as good results - training on our integrated reflectance set - but with the promise of better generalization to unseen data is a key contribution for our method. Further, viewed through the lens of the enclosing hypercube, we find different reflectance sets are much more similar. This is an important result since the question of 'what reflectance data should we use?' is often posed in color imaging research. The work we present here indicates the choice of which dataset is less important if it is integrated (i.e. all data sets become more similar to one another). Finally, a preliminary version of this work was presented in [13].

In section II, we review color correction and the idea of representing reflectances using a linear model. In sections III and IV we explain our core contribution solving for color correction by integrating over a sampled data set. Experiments validate our method in section V. The paper concludes in section VI.

II. Representing Reflectance and Linear Least-Squares Color Correction

Let us begin by recapitulating the integrated response equation that relates the eye's or camera's response to a spectrum of light $E(\lambda)$ striking a surface with the Lambertian spectral reflectance function $S(\lambda)$:

$$\underline{\rho} = \int_{\omega} E(\lambda)S(\lambda)\underline{Q}(\lambda)d\lambda. \tag{1}$$

In Equation 1, $\underline{Q}(\lambda)$ is a vector function of three camera or three human vision system referenced sensors and $\underline{\rho}$ is the trichromatic response vector. The integral is computed over the visible spectrum ω, which runs from 400 to 700 Nanometres. Here and thoughout this paper we will only consider the Lambertian part of surface reflectance —e.g. we will not consider properties such as Fluorescence. To our knowledge all prior-art research into color correction also make this simplifying assumption.

It is common to represent spectral quantities at 10 Nanometre sample points across the visible spectrum. This means that the spectral quantities $E(\lambda)$, $S(\lambda)$ and $Q_i(\lambda)$ $(i \in \{R, G, B\})$ in Equation 1 are represented by the corresponding 31-component vectors: $\underline{E}$, $\underline{S}$ and $\underline{Q}_i$ (where $\underline{Q}_i$ also incorporates the 10Nm sampling distance). Grouping the 3 sensors in a 31 × 3 matrix Q, Equation 1 can be rewritten [5] as:

$$\underline{\rho} = Q^t diag(\underline{E})\underline{S}^t, \tag{2}$$

where $diag()$ makes a diagonal matrix from the vector argument and ^{t} denotes vector/matrix transpose.

Let us represent a set of n surface reflectance spectra by the 31 × n matrix S. Each row of the matrix represents the reflectances at a single wavelength, and each column is one reflectance spectrum. We will denote camera sensors and the XYZ color matching functions (CMFs) by respectively R and

Q (both 31×3 matrices). The camera and XYZ trichromatic responses to all the reflectances in S are computed as:

$$P = R^t diag(\underline{E})S \tag{3a}$$

$$X = Q^t diag(\underline{E})S. \tag{3b}$$

Both P and X are $3 \times n$ matrices.

In Equations 3, we are calculating the XYZ response but we could also have calculated triplets for another human vision referenced color space such as sRGB [12]. But, here and henceforth in this paper we will calculate XYZs.

In color correction, we seek to map RGBs to XYZs, typically with a 3×3 matrix. Unless the camera sensitivities are a linear transform from the XYZs - the so called Luther conditions [10] are met - the correction is necessarily inexact [10]. The least-squares color correction optimization problem is written as:

$$\min_M \|MP - X\|_F \tag{4}$$

where $\|\cdot\|_F$ above denotes Frobenius norm.

Equation 4 can be solved in closed form using the Moore-Penrose inverse [14]:

$$M = XP^t(PP^t)^{-1}. \tag{5}$$

To simplify matters, it is useful to define a color signal matrix as

$$C = diag(\underline{E})S, \tag{6}$$

which means we can rewrite Equation 3 as:

$$P = R^t C \tag{7a}$$

$$X = Q^t C. \tag{7b}$$

Let's substitute Equation 7 into Equation 5

$$M = Q^t CC^t R(R^t CC^t R)^{-1}. \tag{8}$$

Equation 8 teaches that the correction matrix M only depends on the 31×31 color signal autocorrelation matrix CC^t and camera and XYZ spectral sensitivities. If we calculate Equation 8 using a fixed reflectance autocorrelation SS^t but with two lights: $\underline{E}$ and $k\underline{E}$ (where k is a scaling factor) we arrive at the same linear transform. This is a useful property since it implies that a 3×3 color correction matrix will work even if exposure changes.

It is sometimes useful to make the role of the illumination explicit in the autocorrelation:

$$CC^t = diag(\underline{E})SS^t diag(\underline{E}). \tag{9}$$

In [15] it was shown that we could rewrite Equation 9 as:

$$diag(\underline{E})SS^t diag(\underline{E}) = [\underline{E}\,\underline{E}^t] \otimes [SS^t]. \tag{10}$$

where the operator ⊗ 'means' the component-wise multiplication of two matrices.

In the same work it was also shown that if there were k measured lights in the $31 \times k$ matrix E then the color signal correlation matrix (all lights and all surfaces) equals:

$$CC^t = [EE^t] \otimes [SS^t]. \tag{11}$$

Again the central role of the spectral autocorrelation SS^t is clear

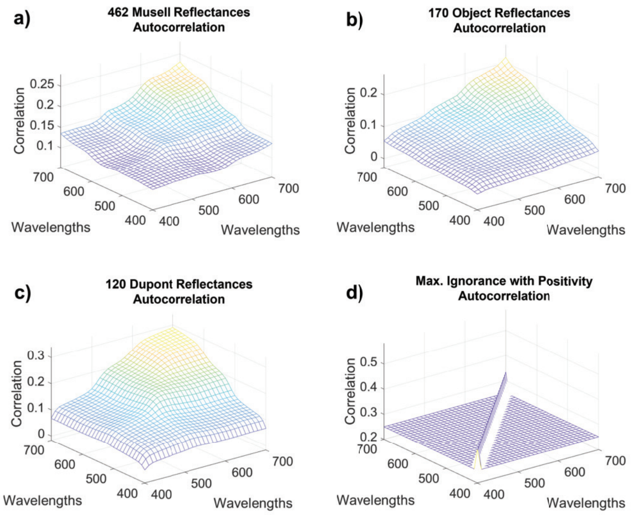


Fig. 4. In (a), (b) and (c) the autocorrelations, SS^t/n (n denotes the number of samples) of the Munsell, Object and Dupont reflectance datasets.

In Figure 4, respectively, in panels (a), (b) and (c), we plot the autocorrelation for the 170 object reflectances [5], the 462 Munsell's [16] and the 120 Dupont reflectances [5].

Encouragingly, the autocorrelations look, broadly, similar. As a function of increasing wavelength each of the reflectance datasets appear more correlated. And, the further apart two wavelengths are, the less they are correlated. This said, the autocorrelations are not the same and, as we will see, the different autocorrelations lead to significantly different color correction transforms (we will discuss further in the experiments section).

In panel d) of Figure 4, we show the autocorrelation corresponding to the Maximum ignorance with positivity (MIP) assumption, where any vector that is all positive between 0 and 1 is equally likely. Relative to this assumption, very jaggy spectra as well as smooth spectra can occur even though the former do not appear in nature. We include the MIP autocorrelation to show how different it is from the autocorrelations of real reflectance data. Previous work has shown that the MIP assumption (i.e. adopting the MIP autocorrelation matrix) leads to poor color correction performance.

The least-squares solution is only one way to calculate the 3×3 correction matrix. Indeed, various authors [17], [18] suggest that we should find a matrix that minimizes a perceptual error. In other research the color correction transform is not a matrix [19], [20], [21]. This said the venerable linear matrix and least-squares is widely deployed not least because image formation itself is linear. Here we will use linear regression not only because it is tried and tested but also because computing the best least-squares transform for our enclosing hypercube of reflectances - developed in this paper - can also be computed.

A. Representing Reflectance Data

It has been observed that because surfaces are smooth they can be represented by an s-dimensional linear model (where

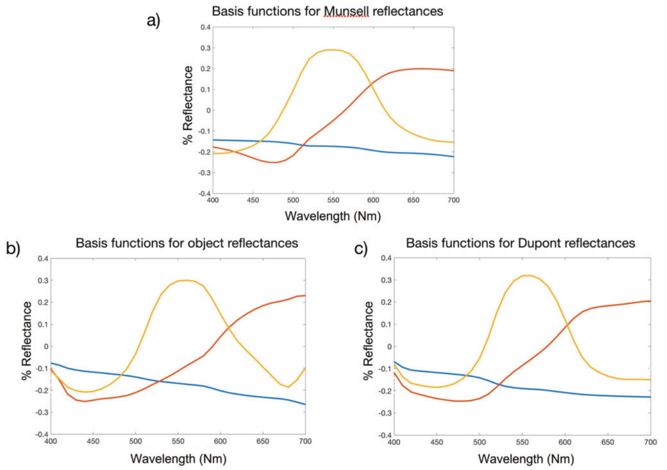


Fig. 5. In (a), (b) and (c) the best 3-dimensional bases for the Munsell, Object and Dupont reflectance datasets. Respectively, in blue, red and yellow are the 1st, 2nd and 3rd Characteristic vectors.

$s \ll 31$):

$$S(\lambda) \approx \sum_{i=1}^{m} U_i(\lambda)\sigma_i \;\equiv\; \underline{S} \approx U\underline{\sigma}, \tag{12}$$

here the ith column of U, U_i, (or $U_i(\lambda)$) is called a basis function and σ_i is a scalar weighting its contribution, $\underline{\sigma}$ is a $m \times 1$ vector. In Equation 12, U is a $31 \times m$ matrix and represents the discrete linear basis for modelling reflectances. The interesting question, of course, is what is the value of m. Different studies have concluded that the answer is between 6 and 9 basis functions [22], [23]. Though, for some applications even a 3-dimensional basis set suffices [24], [25].

For an n-reflectance data set S and a fixed dimension m we would like to find the basis U that best approximates:

$$min_{U,\Omega}\ \|S - U\Omega\|^2, \tag{13}$$

where U and Ω are, respectively, a $31 \times m$ and $m \times n$ and the ith column of Ω is $\underline{\sigma}_i$. Characteristic Vector Analysis (like PCA where the mean is not subtracted) is the apposite tool. CVA returns the optimal solution to Equation 13.

In Figure 5, we find the best 31×3 basis functions for the Munsell, Object and Dupont datasets. As for their autocorrelations, we find the bases have much in common. In blue we see the first characteristic vector which accounts for the greatest amount of variance in the respective datasets. Each function is broadly flat and represents a sort of DC function (in analogy to a Cosine basis expansion). The 2nd characteristic vectors are increasing functions from left to right and the 3rd are smooth functions with a single peak (at more or less the same position). These functions have a similar shape to $\cos 0.5x$ and $\cos x$ (the second and third terms in a cosine basis expansion).

The CVA solution has a number of attractive properties. First, the basis matrix U is orthonormal, $U^t U = I_{m\times m}$. Second, the basis is ordered. The first basis function is the direction (among all others) that best captures the variance in the spectral data S. The second basis function is the direction orthogonal to the first that captures the most variance and so on.

Further, in CVA, the $m \times m$ matrix $\Omega\Omega^t$ is the diagonal matrix D^2 with all positive diagonal components that decrease monotonically, from first to last diagonal component. Placing the diagonal components of D^2 in the vector $\underline{d}$, the ratio $\left(\sum_{i=1}^{k} d_i\right) / \left(\sum_{i=1}^{m} d_i\right)$ reports the % of variance captured by the first k basis functions. We can write Ω as DV^t (note D is the diagonal matrix that is the square-root of D^2) where V is also orthonormal.

It follows that we can decompose the autocorrelation of SS^t as:

$$SS^t \approx UD^2U^t. \tag{14}$$

Since U is orthonormal, it also follows that,

$$\underline{\sigma}_i = U^t \underline{S}_i, \tag{15}$$

and

$$U^t S = \Omega. \tag{16}$$

See [26] for a wider review of representing spectra by characteristic vector analysis.

III. Integrated Reflectance Sets

As presented in the last section, least-squares regression color correction is driven by the autocorrelation of the reflectance data set SS^t (or the autocorrelation of the color signal matrix that itself also depends on the reflectance autocorrelation).

In this section we make two contributions. First, we redescribe all reflectance datasets with a common privileged basis and we discuss how such a basis is computed. Second, we develop our convex closure representation: i.e. the idea that the convex hull of a reflectance set delimits all physically plausible reflectances. For applications like color correction we need to integrate over the convex closure e.g. to calculate the best color correction transform. But, integrating over convex sets is hard. Thus our second contribution, is directed towards making computations feasible. We represent reflectance data by the enclosing hypercube with respect to the common basis coordinates.

A. Deriving a Common Reflectance Basis

As illustrated in Figure 5, different reflectance data sets have different basis functions. This fact alone means it is difficult - if we use a low dimensional representation of reflectance - to compare one reflectance set against another. We can compare basis coordinates (the $\underline{\sigma}$) if and only if we use the same basis functions. To solve this problem we could solve for a *single* set of basis functions which is suitable to represent *all* reflectance datasets.

1) The Statistical Approach: One might start by combining the autocorrelation of the different datasets using a weighted sum:

$$SS^t = \sum_i \frac{S_i S_i^t}{n_i}, \tag{17}$$

where S_i is the set of reflectances in dataset i and n_i is the number of samples in this dataset. This weighted summation gives each reflectance dataset equal weight on its influence on the final basis functions, regardless the size of the reflectance dataset itself. Now, as described in the last section, single value decomposition (SVD) [27] is used to find the optimal $31 \times m$ basis matrix U. As written the magnitude of the autocorrelation would grow with the number of datasets. Thus we might calculate $\frac{SS^t}{\#datasets}$. Though, this is not required in least-squares regression which is independent of the magnitude of the autocorrelation.

2) The Cosine Series Expansion: The optimal basis functions (set of characteristic vectors) found for a reflectance set - at least from an abstract vantage point - looks somewhat like the first three terms in discrete cosine series expansion (see Figure 5). Thus, it is natural that we consider using the discrete cosine expansion as the common priveleged basis. Importantly, the discrete cosine basis has well known energy compaction properties [28]. It is a reasonable choice for the common basis if we wish to be a priori agnostic about the shape of reflectances.

As a cosine series, the basis matrix U is written in closed form. Here $i \in \{0, 1, 2, \cdots, 31\}$ and $j \in \{1, 2, \cdots, m\}$

$$U_{ij} = \cos\left(\frac{(j-1)\pi}{N}\left(i + \frac{1}{2}\right)\right) / k_j. \tag{18}$$

where the form of above equation is chosen so that U is orthogonal. The scalar k_j chosen so that the individual columns of U have magnitude 1 (it accounts for the Wavelength sampling).

Henceforth, for our common privileged basis, we will use cosine series basis in the following discussion and in the experimental section.

B. Convex- and Hyper-Cube Closures of Reflectances

Physically, if we have a checker-board pattern comprising two reflectances $S_1(\lambda)$ and $S_2(\lambda)$ in equal proportions then if we view this checker board from far enough away the checker will appear to have a single color (with an effective reflectance $0.5S_1(\lambda) + 0.5S_2(\lambda)$). More generally, viewed at distance a texture of colors blends to a single color (with the new effective reflectance being a convex sum of the individual reflectances in proportion to their % area coverage). That is, given a dataset of reflectances, if they are *allowed* appear in patterns of arbitrary shape and complexity, then all convex combinations of the reflectances in a dataset might be physically measured.

It follows that the set of plausible spectra is the convex closure of the reflectance samples found in a given data set. And, this convex closure in turn is delimited by the convex hull of the dataset. Unfortunately, calculating a convex hull in d dimensions (for reflectances d is 31) is computationally hard (complexity $O\left(n^{\lfloor \frac{d}{2} \rfloor}\right)$, [29]). Even for 16 dimensional data and a few 100s of reflectances the problem is intractable (at least given the current implementation in Matlab which in turn builds on the efficient Qhull algorithm [29]). In the 16-dimensional case, given 1000 points leads to on the order of 1 trillion operations to solve for the convex hull! (convex hull computation is $O(n^{\lfloor d/2 \rfloor})$, where n is the number of points, d denotes dimension and $\lfloor \ \rfloor$ denotes *floor*.

Suppose we can compute the convex hull of a reflectance set. For color correction we need to integrate over this set. Arguably, the cost of integrating over a convex hull is even greater than the cost of computing the hull itself. Perhaps the simplest integration problem is computing the volume of the convex hull. Here, the cost of computing an analytical solution is the same as the convex hull computation. Indeed, the cost is sufficiently high, that volumes are generally computed via a Monte Carlo simulation (i.e. by sampling).

To illustrate the Monte Carlo simulation suppose we are given a convex hull and we compute the bounding hypercube (the min and max coordinates in the hull per dimension, an easy computation). The volume of this enclosing hypercube is the product of the dimensions (of the hypercube). For the purposes of this example, suppose the volume of the hypercube is k. Now, we generate p points uniformly and randomly in the cube. For each point we check if it is inside or outside of the convex hull (a fast operation). If we find that 90% of the points are inside the hull then the computed volume - by random sampling- is $0.9k$. Of course, we have to choose *enough* random points to calculate the volume with good accuracy. From the introduction to this paper 'enough' can be very large (see Figure 2 and the associated discussion)

To sidestep the issue of the complexity of integration, we will find the enclosing hypercube of the data with respect to the common basis. For a d dimensional dataset S, the coordinates of S in basis U are equal to $\Omega = U^t S$ (see Equation 16). Denoting the kth row of Ω as Ω_k, the min and max values (μ_k and M_k) for the coefficient σ^k are calculated:

$$\mu_k = \min(\Omega_k) \leq \sigma_k \leq \max(\Omega_k) = M_k. \tag{19}$$

The $2d$ min and max coordinates in $\underline{\mu}$ and $\underline{M}$ delimit a hypercube. We illustrate this idea by making clear the bounding hyper-cube (in this case the 3 dimensional rectangle plotted in red in Figure 6).

Notice there is a significant 'gap' between the boundary of the points and the edge of the cube. For this example, the enclosing hypercube has more than twice the volume compared to the convex set. The physical meaning of the 'gap' is that under the hypercube model we admit reflectances that we both haven't seen before and may well be non-physically realizable. That is, they have either reflectivity less than 0 or greater than 1. This kind of extrapolation is common. The Maximum Ignorance, Maximum Ignorance with positivity and Toeplitz assumptions [9], [10], [30] all effectively admit non-physically realizable reflectances.

Under the Maximum ignorance with positivity assumption, we allow each spectral wavelength to be in the interval [0,1] and the value selected per wavelength to be independently of all other wavelengths. Here, all reflectances in the 31dimensional unit cube are equally likely. For our worked example, we can also compute the bounding hypercube - relative to the Dupont 3-dimensional basis - for the Maximum ignorance assumption with positivity. The corresponding MIP hypercube is shown in blue. The MIP cube is 8 times larger than the red hyper-cube based on real data (so, a substantially weaker assumption that admits many more infeasible reflectances).

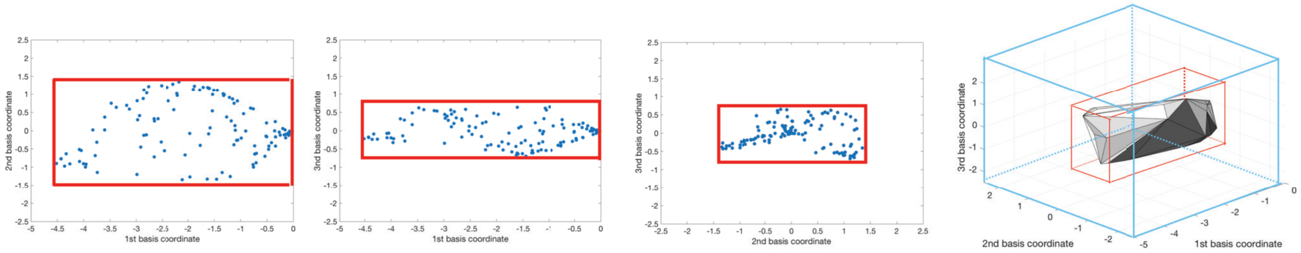


Fig. 6. For the Dupont data set we plot the coordinates with respect to an optimal basis. See the first three panels for plots of 1st against 2nd, 1st against 3rd and 2nd against 3rd basis pairs. On the last plot, we replot the data in 3D. The convex hull of the data points is shown as the shaded convex hull. The enclosing hypercube is shown in red. Assuming we projected all possible spectral data (the maximum ignorance with positivity assumption) on the same basis the corresponding enclosing hypercube is shown in blue.

C. The Importance of the Coordinate Space

The reader might wonder why all the analysis building to this point has represented spectra with respect to a basis. The reason is that we are seeking a representation where the bounding hypercube idea makes sense. While we are willing to allow some spectra that are not in the convex closure of the data set, we do not wish to admit too many.

Arguably, by decorrelating the data, CVA - or indeed the cosine basis that achieves a similar decomposition - finds a representation that is more cube-like. Indeed, data uniformly randomly distributed in a cube is completely decorrelated (knowledge of one coordinate tells you nothing about the other). And, of course by coding a reflectance with respect to a basis, we can choose to use fewer than 31 parameters (fewer than the total number of wavelengths).

Suppose instead we carried out a similar analysis in the primal domain (the primal domain here is the reflectance spectra themselves). For each wavelength, we can calculate the max and minimum reflectance values. In all likelihood this will give reflectances close to 0 and 1 respectively since there are very dark and very bright colors of all hues. If, per wavelength, we assume that reflectance lies in [0,1] and that one wavelength is not correlated with another then this is the definition of the Maximum ignorance with positivity assumption. This assumption leads us to accept many reflectances (think very jaggy spectra) which are not similar to any actual measured data.

We note that the volume of a geometric object does not change when we rotate the object (or more generally apply a unitary transform). When we project reflectances onto a basis, such as the cosine expansion we are re-describing the data with respect to new axes but we are not changing the shape of the data. As a test we projected the 120 Dupont data onto the full 31dimensional cosine expansion (i.e. we applied a full rank unitary 31 × 31 matrix) to the reflectances -see Figure 5-. Using the bounded hypercube we can calculate the volume before and after the unitary transform. Literally, we find the volume with respect to the discrete cosine basis to be orders of magnitude less than in the primal wavelength domain. That is the minimum bounding hypercube depends on the basis in which data is described.

We illustrate this concept in Figure 7 for the data shown (blue dots) in 2 dimensions. The rectangle at 45 degrees has a large bounding box (according to the primal axis). Rotating to a new basis returns a bounding box that more closely describes the shape of the underlying data.

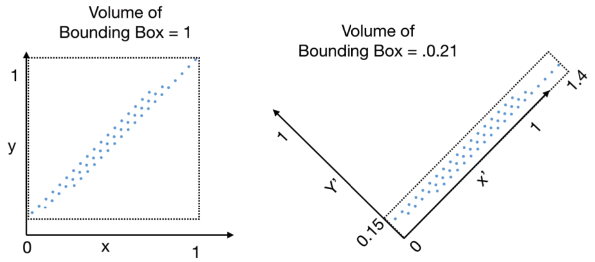


Fig. 7. In the left we see points scattered at 45 degrees. Relative to the x- and y- axes the bounding box - enclosing hypercube - of the data has area 1. Right, we rotate the axes 45 degrees (analogous to CVA). Now the bounding box fits the data better and has area 0.21.

IV. Calculating the Autocorrelation Matrix From the Reflectance Hypercube

We have already seen that the least-squares computation is carried out given knowledge of the spectral sensitivities of the camera and XYZ sensors, the autocorrelation and the light (or lights). In the *discrete* world the autocorrelation is simple to compute:

$$auto(S) = \frac{SS^t}{n}, \tag{20}$$

where n is the number of surfaces in the reflectance dataset. We would like to calculate the autocorrelation given the bounding hypercube. First, remember that in our basis representation we can write $S = U\Omega$ where U is the $31 \times m$ reflectance basis and each column of (the $m \times n$ matrix) Ω is the basis coordinates that define a single reflectance. Here, and henceforth - unless otherwise stated - we will assume $m = 31$ and that U is the cosine series basis. Moreover, we will set $m = 31$ so that U plays a decorrelating role - allows a bounding hypercube to better fit the data - but the transform is lossless.

This lossless property is important as even though reflectances are smooth, all the terms in the basis expansion are necessary for some reflectances. Indeed, for the Simon Fraser reflectance data set [6] (which also contains some of the reflectance data we use in this paper) we found that using only the first 30 components in a cosine basis expansion would occasionally result in a reflectance that had a non-negligible fitting error. Removing the highest frequency basis term resulted in the least well fitted spectrum being almost 1% different to the actual measured data. Put another way, removing a single term in the cosine basis expansion can result in a model of reflectance that cannot describe all the reflectances in a dataset. Henceforth we will always use the same number of terms in a cosine expansion as there are sampling wavelengths in a dataset.

Now, let us write

$$auto(S) = \frac{U\Omega\Omega^t U^t}{n} \tag{21}$$

Given that U is fixed, the autocorrelation for a reflectance set depends on $\Omega\Omega^t$ (the autocorrelation of the coordinates of the reflectances with respect to the basis).

$$auto(\Omega) = \frac{\Omega\Omega^t}{n}. \tag{22}$$

Let us rewrite Equation 22 as a summation:

$$[auto(\Omega)]_{ij} = \sum_{k=1}^{31} \Omega_{ik}\Omega_{jk}/n, \tag{23}$$

for the special case when $i = j$,

$$[auto(\Omega)]_{ii} = \sum_{k=1}^{31} \Omega_{ik}^2/n. \tag{24}$$

When we come to computing the autocorrelation over the hypercube, the special case of Equation 24 is important. In Equation 23 (when $i \neq j$) there are two statistical variables but in 24 there is only one. This an important detail.

Let us denote the ith row of Ω as Ω_i. The bounding hypercube Ω is defined by the min and max row coordinates in Ω.

$$Box(\Omega) = \{\underline{\mu}, \underline{M}\}\ , \quad \text{where } \mu_i = min(\Omega_i) \text{ and } M_i = max(\Omega_i). \tag{25}$$

We denote the autocorrelation of all the reflectances in a bounding hypercube for Ω as $auto(Box(\Omega))$. To calculate $auto(Box(\Omega))$ we need to integrate over the bounding box.

A. Integrating Over a Hypercube

Because the hypercube is a continuous set, the summations in Equations 23, 24 become integrals. We will use the notion σ_i to denote a sample in the ith coordinate (σ_i denotes a continuous variable). Given that the enclosing hypercube is defined by the min and max coordinates in each dimension (see Equation 22) then $\mu_i < \sigma_i < M_i$. The Expected value of the autocorrelation is computed as:

$$[E(auto(Box(\Omega)))]_{ij} = \begin{cases} \dfrac{\int_{\mu_i}^{M_i} \sigma_i^2 \ \mathrm{d}\sigma_i}{M_i - \mu_i} \text{when } i = j \\ \dfrac{\int_{\mu_j}^{M_j}\int_{\mu_i}^{M_i} \sigma_i\sigma_j \ \mathrm{d}\sigma_i \mathrm{d}\sigma_j}{(M_i - \mu_i)(M_j - \mu_j)} \\ \text{when } i \neq j. \end{cases} \tag{26}$$

We have different expressions for the diagonal and off-diagonal terms where they are the same in the discrete case (see Equations 23 and 24 and discussion). For the diagonal case there is only one statistical variable but for the off-diagonal there are two. This in turn means computing the expectation along the diagonal of the autocorrelation is a single integral problem and the off-diagonal computation involves solving a double integral.

Notice that the numerator terms of the continuous integral in Equation 26 look similar, as we would expect, to the discrete summations. The denominator terms are different from 'dividing' by n (of Eqns 23 and 24). When we compute expectations in the continuous domain, as commented above, we need to divide by the length of the interval (for the 1D case) and the area of integration for the 2D case. Equation 26 can be solved for in closed form with the autocorrelation defined as:

$$[E(auto(Box(\Omega)))]_{ij} = \begin{cases} \dfrac{M_i^3 - \mu_i^3}{3(M_i - \mu_i)} \text{when } i = j \\ \\ \dfrac{M_i^2 M_j^2 + \mu_i^2\mu_j^2 - \mu_i^2 M_j^2 - M_i^2\mu_j^2}{4(M_i - \mu_i)(M_j - \mu_j)} \\ \text{when } i \neq j. \end{cases} \tag{27}$$

For the continuous case (integrating over the bounding hypercube) the estimated autocorrelation is calculated as $UE(auto(Box(\Omega)))U^t$.

We remark that given that we know the enclosing hypercube (min-max coordinates), the computation of the autocorrelation is very rapid. Indeed, it will be quickler than calculating the autocorrelation of a real dataset. Yet, this said, color correction calculation is, in both cases, a very fast computation.

B. Integrating Over a Scaled Hypercube, sBox

Suppose we calculate a scaled hypercube that is k times the size in each dimension

$$sBox(\Omega, k) = \{\underline{\mu}, \underline{M}\}\ , \quad \text{where } \mu_i = k \cdot min(\Omega_i) \text{ and } M_i = k \cdot max(\Omega_i) \tag{28}$$

From Equation 27, it is easy to show that:

$$E(auto(sBox(\Omega, k))) = k^2 E(auto(Box(\Omega))) \tag{29}$$

That is, by scaling the bounding coordinates of the hypercube the corresponding autocorrelation also scales. From Section II we know that least-squares color correction is independent of the magnitude of the autocorrelation. Moreover, and perhaps more importantly, we have shown that the terms in the autocorrelation scale to the square of the scaling applied to the data.

We now refer the reader back to Equations 8 through 10 (how a reflectance autocorrelation is used in least-squares color correction). We simply substitute the expected autocorrelation, $E(auto(sBox(\Omega, k)))$, for SS^t, the autocorrelation from measured data in Equation 9. Given a measured illuminant spectrum, we calculate the color signal autocorrelation which can be substituted into Equation 8.

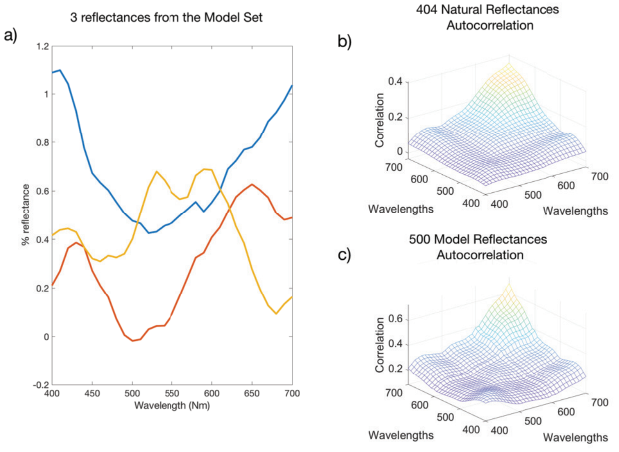


Fig. 8. In a), 3 reflectances from the Model reflectance dataset. Note there can be values larger than 1 and less than 0. Panels (b) and (c) show the autocorrelations for the **NAT** and **MOD** spectral datasets.

V. Experiments

A. Reflectance Datasets

Hitherto in this paper, we have considered the respectively 462, 170 and 120 reflectance **MUN**sell, **OBJ**ect and **DUP**ont reflectance sets. The Munsell reflectances are painted patches designed to have a large color gamut. The Object dataset contains spectral of typical objects including bricks, wood and pavement. The Dupont set contains the spectra of colorful dyed material. All of these datasets are similar in that they do not a priori place constraints on the shape of spectra.

To these three exemplar measured reflectance datasets, we add two more. First is the **NAT**ural dataset measured by Westland et al [2] which comprises 404 measured spectra of plants, foliage and flowers. Second, we add a **MOD**el that comprises 500 randomly selected spectra from the bounding hypercube of the bounding hypercubes of the **MUN**, **OBJ**, **DUP** and **NAT** datasets.

To construct the reflectance dataset **MOD**, using an index $k \in \{\mathbf{MUN}, \mathbf{OBJ}, \mathbf{DUP}, \mathbf{NAT}\}$, we calculate the enclosing hypercube - denoted by ENC - of these sets (where as before the spectra are represented by their Ω coordinates with respect to the Cosine Series Basis U):

$$\begin{aligned} box(\Omega_{\mathbf{ENC}}) &= \{\underline{\mu}_{\mathbf{ENC}}, \underline{M}_{\mathbf{ENC}}\}, \\ \underline{\mu}_{\mathbf{ENC}} &= \min_k \underline{\mu}_k, \\ \underline{M}_{\mathbf{ENC}} &= \max_k \underline{M}_k. \end{aligned} \tag{30}$$

For the ith of 500 trials we select a vector $\underline{\sigma}_i \in \mathcal{U}(\underline{\mu}_{\mathbf{ENC}}\ \ \underline{M}_{\mathbf{ENC}})$ (i.e. uniformly and randomly from the hypercube) and the corresponding reconstructed spectrum is calculated as $U\underline{\sigma}_i$.

In Figure 8, panel (a), we show 3 of the randomly generated reflectances in **MOD**. Notice that they look like plausible spectra but there are - as expected - values less than 0 and greater than 1. Notice also that the reflectances are not completely smooth. This non smoothness 'local scale' appears to occur in nature, especially in the spectra of flowers and plants [2]. Though, by construction, all the smooth spectra in Figure 8a (from the Macbeth Color Checker) must be in $box(\Omega_{\mathbf{MOD}})$. For our 5 data sets we found that more than 85% of our data are physically realizable: 85% of the volume of the hypercube correspond to reflectances with values in [0,1].

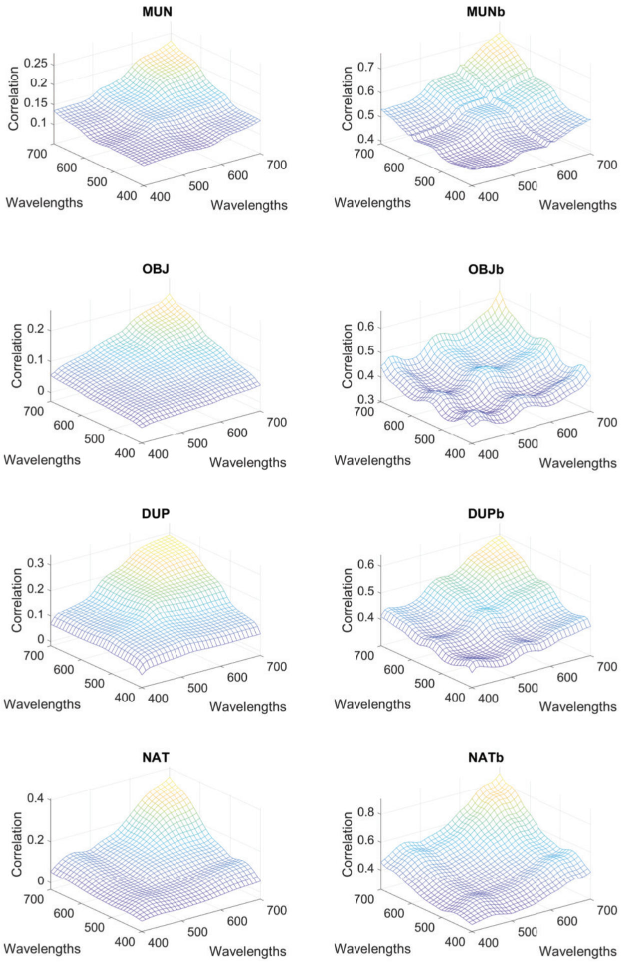


Fig. 9. On the left the 4 autocorrelation matrices and on the right the corresponding autocorrelation of the bounding box hypercubes.

Finally, we remark that we could of course simply ignore the 15% of reflectances that are infeasible (reflect more than 100% or less than 0% of the incident light). However, then the autocorrelation statistics for **MOD** would not be well described by our enclosing hypercube autocorrelation. Thus, even though they are not physically realiseable, we keep them in the **MOD** set to ensure congruence with our bounding hypercube model.

In Figure 8 in panels (b) and (c) we respectively show the autocorrelations of **NAT** and **MOD**. In Figure 9 (left column) we show the autocorrelations of our sets of measured reflectances. On the right are the autocorrelations computed from their enclosing hypercubes.

B. Comparing the Autocorrelations of Reflectance Datasets

Let us begin by comparing directly all pairs of autocorrelations using a % error metric, where $||.||_F$ denotes the Frobenius

TABLE I

$\frac{\%}{100}$ Reflectance Set Autocorrelation Errors

	MUN	**OBJ**	**DUP**	**NAT**	**MOD**	Mean
MUN	0	0.38	0.26	0.35	0.89	0.47
OBJ	0.57	0	0.76	0.44	1.82	0.90
DUP	0.23	0.44	0	0.30	0.66	0.41
NAT	0.38	0.32	0.37	0	1.07	0.54
MOD	0.48	0.65	0.41	0.53	0	0.52
Mean	0.42	0.45	0.45	0.41	1.11	0.57

TABLE II

% Enclosing Hypercube of Reflectance Set Autocorrelation Errors

	MUNb	**OBJb**	**DUPb**	**NATb**	**MODb**	Mean
MUNb	0	0.22	0.17	0.25	0.13	0.19
OBJb	0.27	0	0.12	0.42	0.32	0.28
DUPb	0.20	0.11	0	0.32	0.25	0.22
NATb	0.25	0.34	0.27	0	0.17	0.26
MODb	0.12	0.25	0.20	0.16	0	0.18
Mean	0.21	0.23	0.19	0.29	0.22	0.23

norm.

$$err_{A,B} = \frac{\|auto(\Omega_A) - auto(\Omega_B)\|_F}{\|auto(\Omega_A)\|_F}. \tag{31}$$

The results for the 5 data sets are summarized in Table I. The autocorrelations A and B from Equation 31 correspond to row and column respectively (the % autocorrelation error is not commutative). The last column and row are respectively the average errors of the first 5 rows and columns. Bottom right is the overall average error.

Looking at the Table I, errors range from 25% to almost 200% with the overall average as 57%. However, remembering that **MOD**, by construction is designed to be a superset of the other 4 datasets, it is perhaps unsurprising that the comparisons that involve the autocorrelation of **MOD** are the ones that, typically, have the highest errors. An interesting lesson we draw from the table is that the Dupont data -that comprises man-made dyed textiles- actually, has the lowest errors compared with the other data sets (see third row). That is, if we were to use only one of these datasets to represent the others (in terms of their autocorrelations), then we would choose the Dupont reflectances.

In Table II we repeat the experiment for the autocorrelations of the enclosing hypercubes of the different datasets. Here, the datasets are labelled as **MUNb**, **OBJb**, **DUPb**, **NATb** and **MODb**, the **'b'** stands for 'box' (i.e. the figures are for the autocorrelations of the enclosing hypercube). The error is calculated as:

$$err^{Box}_{A,B} = \frac{\|E(auto(box(\Omega_A))) - E(auto(box(\Omega_B)))\|}{\|E(auto(box(\Omega_A)))\|}. \tag{32}$$

With two exceptions (**NAT** compared with **OBJ**) and (**DUP** compared with **NAT**) the autocorrelations calculated over reflectance set's bounding hypercube are closer to each other (and usually much closer). Overall, the error is reduced from 57% to just 23% and the composite **MOD** set is now much closer to all other sets. Either the **MUN** and **MOD** would seem to be the best choice to represent all 5 datasets.

See again, in Figure 9, the autocorrelations of the sampled reflectance sets contrasted with corresponding enclosing hypercube autocorrelations on the right. According to our measure of similarity the autocorrelations on the right are more similar than those on the left, on average. Visually, they do look more similar: they appear flatter overall. Notice also that the characteristic contours of the Munsell data -see Figure 4- are now replicated in the enclosing hypercube autocorrelations for the Object and Dupont datasets.

TABLE III

Cross Validated Color Correction. Average for the 28 Cameras of the Median Delta E for the Standard Least-Squares Computation on the Auto-Correlation

	MUN	**OBJ**	**DUP**	**NAT**	Mean
MUN	0.90	1.49	1.87	2.30	1.64
OBJ	1.36	0.99	1.85	1.84	1.51
DUP	1.24	1.17	1.42	1.52	1.34
NAT	1.38	1.31	1.72	1.56	1.49
Mean	1.22	1.24	1.72	1.81	1.50

C. Color Correction Experiments

1) Reflectance Sets: In our color correction experiment we use the spectral sensitivities of 28 cameras in [31], the set of 102 illuminants in [6] and the **OBJ**, **DUP**, **NAT** and **MUN** reflectance data sets. For each light we also compute the reference XYZ tristimuli.

We wish to investigate how well least-squares color correction works when we train on one dataset and test on another. As an example we can train on the **MUN** reflectances to obtain a 3 × 3 correction matrix and then test on the **OBJ** reflectances. For one light and one camera, as there are 170 **OBJ** reflectances we map the 170 recorded RGBs to predict the corrresponding 170 XYZs. We calculate the CIELAB Delta E error for the 170 samples. We repeat this experiment 102 times (for the 102 lights). Ultimately, this yields $170 * 102 = 17340$ Delta E values. We compute the median of these entries, and then we repeat the operation for the other 27 cameras, obtaining a 28-d vector. We finally compute the average of this vector. The result of this procedure is shown in row 1, column 2, of Table III, and is 1.49. All pairs of training and testing surface sets -and their, analogously calculated respective Delta E values are also shown in Table III.

The averages over columns are shown in the rightmost column. These averages encode how well a reflectance set performs when it is used to determine the color correction transform. The average over rows speaks to the difficulty of correcting a given reflectance test set. Training with the **DUP** set gives the lowest error overall, on average 1.34 Delta E. On the other side, the Natural reflectances are hardest to color correct (an average of, 1.81).

We now repeat this experiment where we now train on the autocorrelation of all reflectances in the enclosing hypercube. We then test on the reflectance sets themselves. Here we do include the **MOD** dataset for training. Returning to Figure 6, the bounding hypercube of a reflectance set is many multiples in volume larger than the reflectances set itself (as defined by the convex closure). So, we are training on many more reflectances than the samples themselves. This said, we do not expect better color correction results (certainly we must

TABLE IV
CROSS VALIDATED COLOR CORRECTION, USING ENCLOSING HYPERCUBE TO TRAIN. AVERAGE FOR THE 28 CAMERAS OF THE MEDIAN DELTA E

	MUN	**OBJ**	**DUP**	**NAT**	Mean
MUNb	1.09	1.24	2.73	2.51	1.89
OBJb	1.44	1.61	2.31	1.88	1.81
DUPb	1.15	1.46	2.50	2.52	1.91
NATb	1.13	1.21	2.23	2.60	1.79
MODb	1.19	1.53	2.13	2.18	1.76
Mean	1.20	1.41	2.38	2.34	1.83

do worse than when we train and test on the same data). But if we obtain competitive performance then this means we can take a much more agnostic stance about which reflectance appear in the world and still get good color correction.

The color correction results for this second experiment are reported in Table IV. We obtain similar results (indeed a few are lower). On average, for corresponding entries, the error - already very low - remains low. The average over all the Table indicate a 22% increase than fitting with the actual reflectances. Four of the common entries are smaller.

D. Correction Performance Using Real-World Data

Here, we begin by building a large set of reflectances drawn from the hyperspectral images [32]. We select the set of "Stuff" images —composed of 15 images— and, for each of the images, we add one of each 16 pixels both horizontally and vertically to our set. We follow this approach because as one would expect, the same object in the same images has, more or less, the same spectrum. Following this approach, we end up with a set of 14336 real reflectances. We denote this reflectance set as **REP**.

Then, given a particular camera, we have synthesized all the RGBs for each reflectance and illumination. We have computed the DeltaE error as before for both the standard Least-Squares and our enclosing hypercube approach. In this case, we have computed for each camera the median error, the mean error, and the trimean error over the 14336 × 102 color signals. The mean of this process for the 28 cameras is presented in Table V.

In Table V, a 3 × 3 matrix is calculated for each of 5 training reflectance sets, **MUN**, **OBJ**, **DUP**, **NAT** and **MOD**. Then we report color correction performance in 3 tranches: see the first, second and third group of 3 columns. In the first 3 columns, we calculate the correction matrices calculated from the enclosing hypercube of the reflectance sets. The next 3 columns corresponds to using each reflectance set directly to drive least-squares. In the last tranche (last 3 columns) we report the performance of the Minimal Knowledge II method (MK-II) [33]. This method presents some similarities to ours, as it constructs the autocorrelation matrix given some *weak* statistical assumptions about the set of likely spectra.

In row 4, column 1 we see that defining a 3 × 3 matrix based on the enclosing hypercube of the **NAT** reflectances and testing on the **REP** reflectances results in a median error of 1.34. Respectively, training on the autocorrelation of the **NAT** itself or using the MK-II approach returns median errors of 1.49 and 1.85 respectively.

In Table V, the best results (in bold) for the Mean, Median and Trimean errors are obtained using enclosing hypercube autocorrelations, and the best overall is for the enclosing hypercube for the MUN dataset. The enclosing hypercube approach also works best overall. In Table V, green highlights the best result for a given statistic and reflectance training set.

It is remarkable how well colour correction based on the bounding hypercube works in this experiment. One might reasonably infer that for the **REP** reflectances the autocorrelations calculated using our bounding hypercubes are more representative than those generated by the raw statistics of the underlying measured spectral data. Of course, if the **REP** data had replicated, for example, the statistics of the **MUN** or any other measured data set then the corresponding autocorrelation (calculated for **MUN**) would necessarily deliver slightly better colour correction. However, this experiment has shown that our bounding hypercube generalizes well to unseen data.

Finally, we comment on the interesting question of what these numbers practically mean. We point the reader to Figure 10 which compares the results for our bounding hypercube correction and those of LS. Here, we can see how our method obtains lower errors than those of LS. In general, colors are mapped with low error for both cases. But, as we can see in the zoomed part of this image, our improvement is clear in different areas.

For completeness, we also computed correction transforms using the Maximum Ignorance with Positivity (MIP) [9] and the Minimal Knowledge-I (MK-I) [34] assumptions. See Table VI for summary statistics. [35]. Overall, computing the autocorrelation using bounding hypercubes works better than either of these two approaches. And, reflectance-based MK-II works better than MK-I (which works independently of any assumption about the reflectance set).

1) The Stability of Color Correction: We can ask a further question regarding Color Correction: Given a color signal, how stable is the solution computed when training with different datasets?

Here, we evaluate the stability of 3 color correction methods: Simple Least-Squares (LS), MKII, and our bounding hypercube approach. In particular, we want to consider the variance in the mapping of a set of RGB data. Specifically, if we train on a reflectance set (the same reflectance set drives LS, MKII, and *Our* approach) we will get a set of estimated XYZs. But, if we train these three color correction methods using different reflectance sets the estimated XYZs will vary. It follows, per method, we can calculate the estimated XYZs for different training reflectance sets and from these XYZ estimates we can calculate the variance in the estimated data. Ideally, we would like a color correction method to have a low *bias* (it is close to the actual XYZs) and a low variance (it works well given different training data sets).

The spectral data we use here comprises the 28 cameras x 102 illuminants x 14336 real world reflectances (**REP**) yielding 40943616 RGBs. As training data, we use one of the 5 reflectance sets: **MUN**, **OBJ**, **DUP**, **NAT**, or **MOD**. That is per reflectance training set we will estimate corresponding XYZs. Per reflectance, illuminant and camera (and per method) we will have 5 different estimates of the XYZs. We say a color

TABLE V

COLOR CORRECTION RESULTS WHERE THE CORRECTION TRANSFORMED IS CALCULATED USING THE ENCLOSING HYPERCUBE AUTOCORRELATION, THE LEAST-SQUARES(STANDARD AUTOCORRELATION), AND MINIMAL-KNOWLEDGE II; AND THEN TESTED ON THE REP REFLECTANCE SET. AVERAGE FOR 28 CAMERAS OF THE MEDIAN, MEAN, AND TRIMEAN DELTA E

	Enclosing Hypercube (Ours)			LS (standard autocorrelation)			MK-II		
	Median	Mean	Trimean	Median	Mean	Trimean	Median	Mean	Trimean
MUN	**1.12**	2	**1.26**	1.29	2.06	1.41	1.4	2.11	1.53
OBJ	1.19	**1.93**	1.32	1.4	2.03	1.5	1.64	2.24	1.73
DUP	1.14	2.02	1.29	1.45	2.06	1.55	1.67	2.26	1.77
NAT	1.34	2.21	1.47	1.49	2.16	1.59	1.85	2.39	1.94
MOD	1.21	2.02	1.35	1.21	2.01	1.34	1.48	2.18	1.6
MEAN	1.20	2.04	1.34	1.37	2.06	1.48	1.61	2.24	1.71

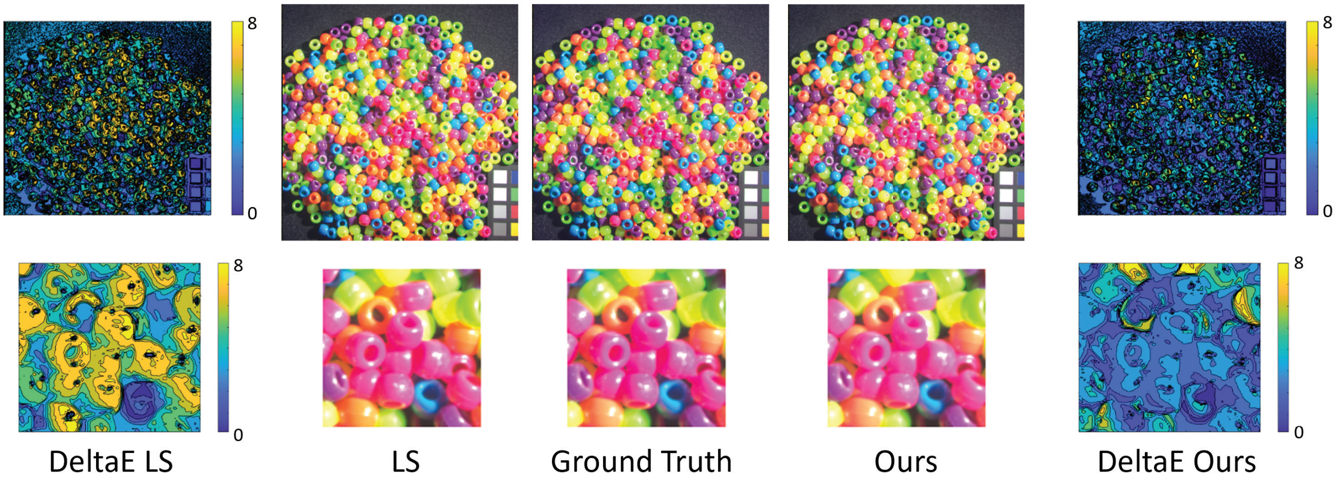


Fig. 10. Visual comparison between our approach and LS for a real image of the **REP** dataset. This result considers **OBJ** dataset as training reflectances and illuminant D65.

TABLE VI

COLOR CORRECTION RESULTS FOR MAXIMUM IGNORANCE WITH POSITIVITY (MIP) AND MINIMAL-KNOWLEDGE I (MK-I) TRANSFORMS (WHICH ARE DERIVED WITHOUT REFERENCE TO A TRAINING SET) THEN APPLIED TO THE REP REFLECTANCE SET. AVERAGE FOR THE 28 CAMERAS OF THE MEDIAN, MEAN, AND TRIMEAN DELTA E

	Median	Mean	Trimean
MIP	1.89	3.27	2.10
MKI	1.71	3.26	1.91

TABLE VII

STABILITY MEASURE—AS THE RADIUS CONTAINING ALL POSSIBLE ESTIMATES—FOR THE DIFFERENT METHODS. WE CAN CLEARLY SEE THAT OUR METHOD IS MORE STABLE THAN THE OTHERS

	Radius		
	Mean	Median	95%
LS	0.88	0.45	3.20
MKII	0.77	0.44	2.56
Ours	**0.60**	**0.33**	**2.10**

correction method is stable if these 5 estimates are close to one another.

To measure the stability of an algorithm we calculate the following measure. We find the minimum radius (in CIE Lab space) enclosing all 5 estimates—this means, the smallest DeltaE radius that will convey all the points—. Thus, we obtain 40943616 values (28 cameras x 102 illuminants x 14336 reflectances). Then, we calculate the mean, median and 95% from these.

Results for this computation are shown in Table VII. We can clearly see that our proposed hypercube approach -bottom row- is more stable than the standard Least-Squares autocorrelation approach, and the MK-II.

The code for integrating over a hypercube of reflectances in the privileged cosine basis coordinate space and an example least squares calculation can be found in *https://github.com/jvazquezcorral/Bounding_Hypercube*.

VI. DISCUSSION

The representation we developed in this paper —the enclosing hypercube in a privileged coordinate frame— is very much in the maximum ignorance direction [9], i.e., it is a rich enough model to account for all likely reflectances. Additionally, the enclosing hypercube also avoids accounting for many unlikely reflectances.

Another approach to combining datasets would have been simply to *stack* the 4 reflectance sets - **MUN**, **OBJ**, **DUP** and **NAT**) - together and use the combined set, denoted **STACK**, to calculate the reflectance autocorrelation. However, we do not believe this is a maximally ignorant way to proceed. Indeed, the researchers who compiled the 4 datasets did not claim that the reflectances were in any way representative of all the reflectances one might measure. The way to make them representative would be to allow convex combinations of the reflectances [36]. But, in making such an allowance, we follow the same path that led us to the enclosing hypercube of reflectances. In our view, given the dimensionality of reflectance data, it is not possible to sample all likely reflectances.

This being said, we repeat the color correction experiment for the real reflectances **REP** (28 cameras, 102 lights, and **REP** comprising 14336 reflectances). For each camera and light pair, we calculate least-squares using the autocorrelation calculated from **STACK**. The median, mean, and trimean errors are, respectively, 1.34, 2.02, and 1.45. Note these are a little worse than for **MOD** (1.21, 2.01 and 1.34). Of course, to generalize we could calculate the enclosing hypercube of **STACK**, but this simply gives us performance similar to the enclosing hypercube of **MOD**. In summary, for the experiment using real reflectances, color correction using a stacked dataset does not work as well as an enclosing hypercube approach.

But is our enclosing hypercube representative of *all* reflectance data? Almost certainly, there will exist reflectance that could increase the volume of our enclosing hypercube and, in so doing, change the related autocorrelation statistics. However, the reflectance sets used here have been measured over the last 50 years, often with the idea of measuring typical reflectances. For example, the natural reflectances measured in [2] are simply flowers, leaves, and bark found at the University of Keele campus. There is no evidence that the authors in that study or, indeed, in any previous studies have sought not to avoid *difficult* reflectances.

VII. Conclusion

In this paper, we have proposed that the space of reflectance data should not be sampled but, rather, integrated, introducing the idea of using the convex closure of a reflectance data set. To this end, we have shown how to approximate these convex closures by their enclosing hyper-cube (in a privileged coordinate system). We have also shown the utility of such an approach for the case of color correction by integrating over the border of the defined hypercube. Our proposed approach avoids the bias present in color correction due to the choice of reflectance datasets. Our results show that, under simulated real-life scenarios, i.e. when training and testing reflectance datasets are different, our method competes with and even outperforms state-of-the-art color correction algorithms. Additionally, we have also shown that the results of our approach are more stable despite the change in the training reflectance dataset, making our method more reliable for its deployment in the wild.

References

[1] Univ. Eastern Finland. *Spectral Database—Munsell Colors Matt (spectrofotometer Measured)*. Accessed: Feb. 15, 2025. [Online]. Available: http://www.uef.fi/web/spectral/munsell-colors-matt-spectrofotometer-measured

[2] S. Westland, J. Shaw, and H. Owens, "Colour statistics of natural and man-made surfaces," *Sensor Rev.*, vol. 20, no. 1, pp. 50–55, Mar. 2000.

[3] M. E. Dyer, Z. Füredi, and C. McDiarmid, "Volumes spanned by random points in the hypercube," *Random Struct. Algorithms*, vol. 3, no. 1, pp. 91–106, Jan. 1992.

[4] C. S. McCamy, H. Marcus, and J. Davidson, "A color-rendition chart," *J. App. Photog. Eng.*, vol. 2, no. 3, pp. 95–99, 1976.

[5] M. J. Vrhel, R. Gershon, and L. S. Iwan, "Measurement and analysis of object reflectance spectra," *Color Res. Appl.*, vol. 19, no. 1, pp. 4–9, Feb. 1994.

[6] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Res. Appl.*, vol. 27, no. 3, pp. 147–151, Jun. 2002.

[7] D. A. Forsyth, "A novel algorithm for color constancy," *Int. J. Comput. Vis.*, vol. 5, no. 1, pp. 5–35, Aug. 1990.

[8] M. J. Vrhel and H. J. Trussell, "Color correction using principal components," *Color Res. Appl.*, vol. 17, no. 5, pp. 328–338, Oct. 1992.

[9] G. D. Finlayson and M. S. Drew, "The maximum ignorance assumption with positivity," in *Proc. Color Imag. Conf.*, 1996, no. 1, pp. 202–205.

[10] B. K. P. Horn, "Exact reproduction of colored images," *Comput. Vis., Graph., Image Process.*, vol. 26, no. 2, pp. 135–167, May 1984.

[11] T. Smith and J. Guild, "The cie colorimetric standards and their use," *Trans. Opt. Soc.*, vol. 33, no. 3, p. 73, 1931.

[12] M. Anderson, R. J. Motta, S. Chandrasekar, and M. Stokes, "Proposal for a standard default color space for the internet–sRGB," in *Proc. Color Imag. Conf.*, Jan. 1996, pp. 238–245.

[13] G. D. Finlayson, J. Vázquez-Corral, and F. Fang, "The discrete cosine maximum ignorance assumption," in *Proc. Color Imag. Conf.*, vol. 29, Nov. 2021, pp. 13–18.

[14] C. F. Van Loan and G. H. Golub, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1983.

[15] S. D. Hordley, J. Paul, and G. Finlayson, "Colour correction in theory and practice," in *Proc. 5th IASTED Int. Conf. Visualizat., Imag., Image Process.*, Sep. 2005, pp. 320–325.

[16] S. M. Newhall, D. Nickerson, and D. B. Judd, "Final report of the OSA subcommittee on the spacing of the Munsell colors," *J. Org. Soc. Amer.*, vol. 33, no. 7, p. 385, Jul. 1943.

[17] J. Vazquez-Corral, D. Connah, and M. Bertalmío, "Perceptual color characterization of cameras," *Sensors*, vol. 14, no. 12, pp. 23205–23229, Dec. 2014.

[18] G. D. Finlayson and M. S. Drew, "White-point preserving color correction," in *Proc. Color Imag. Conf.*, Jan. 1997, vol. 5, no. 1, pp. 258–261.

[19] E. Garcia and M. R. Gupta, "Building accurate and smooth ICC profiles by lattice regression," in *Proc. Color Imag. Conf.*, Jan. 2009, vol. 17, no. 1, pp. 101–106.

[20] V. Cheung and S. Westland, "Color camera characterisation using artificial neural networks," in *Proc. Color Imag. Conf.*, Jan. 2002, vol. 10, no. 1, pp. 117–120.

[21] G. D. Finlayson, M. Mackiewicz, and A. Hurlbert, "Color correction using root-polynomial regression," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1460–1470, May 2015.

[22] D. H. Marimont and B. A. Wandell, "Linear models of surface and illuminant spectra," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 9, no. 11, pp. 1905–1913, 1992.

[23] M. S. Peercy, "Linear color representations for full speed spectral rendering," in *Proc. 20th Annu. Conf. Comput. Graph. Interact. Techn.*, Sep. 1993, pp. 191–198.

[24] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Color constancy: Generalized diagonal transforms suffice," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 11, no. 11, p. 3011, 1994.

[25] L. T. Maloney and B. A. Wandell, "Color constancy: A method for recovering surface spectral reflectance," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 3, no. 1, p. 29, 1986.

[26] L. T. Maloney, "Evaluation of linear models of surface spectral reflectance with small numbers of parameters," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 3, no. 10, p. 1673, 1986.

[27] J. E. Gentle, *Numerical Linear Algebra for Applications in Statistics*. Cham, Switzerland: Springer, 2012.

[28] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-100, no. 1, pp. 90–93, Jan. 1974.

[29] K. Fukuda. (2004). *Frequently Asked Questions in Polyhedral Computation*. [Online]. Available: http://www.ifor.math.ethz.ch/

[30] B. Mulgrew and C. F. Cowan, *Adaptive Filters and Equalisers*, vol. 56. Cham, Switzerland: Springer, 2012.

[31] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, "What is the space of spectral sensitivity functions for digital color cameras?," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 168–179.

[32] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range and spectrum," Dept. of Computer Science, Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-061-08, 2008.

[33] J. A. S. Viggiano, "Minimal-knowledge assumptions in digital still camera characterization. II: Non-uniform distribution," in *Proc. PICS Conf.*, Jan. 2003, pp. 435–440.

[34] J. A. S. Viggiano, "Minimal-knowledge assumptions in digital still camera characterization I: Uniform distribution, Toeplitz correlation," in *Proc. 9th Color Imag. Conf. (IST)*, vol. 9, Jan. 2001, pp. 332–336.

[35] G. D. Finlayson and J. Paul, "Minimal knowledge versus the real world," in *Proc. Color Imag. Conf.*, vol. 10, no. 1, Jan. 2002, pp. 133–138.

[36] G. D. Finlayson and P. Morovic, "Metamer sets," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 22, no. 5, pp. 810–819, 2005.

Graham D. Finlayson (Member, IEEE) is currently a Professor with the School of Computing Sciences, University of East Anglia, U.K. His research interests include color, physics-based computer vision, image processing, and the engineering required to embed technology in devices. He is a fellow of the Royal Photographic Society, the Society for Imaging Science and Technology, and the Institution for Engineering Technology.

Javier Vazquez-Corral is currently an Associate Professor with the Universitat Autònoma de Barcelona (UAB) and a Staff Scientist at the Computer Vision Center. His research interests are related to the use of color in image processing and computer vision problems. He is also interested in bridging the gap between color in the human brain and its use in computer-vision applications.

Fufu Fang is currently pursuing the Ph.D. degree with the School of Computing Science, University of East Anglia, with a research focus on color correction. He is also a Software Engineer at Arm working on a user-space GPU driver.